IBM Tivoli Storage Manager
Version 7.1.6

Optimizing Performance

IBM

IBM Tivoli Storage Manager
Version 7.1.6

Optimizing Performance

IBM

Note:
Before you use this information and the product it supports, read the information in "Notices" on page 251.

Third edition (June 2016)

This edition applies to version 7, release 1, modification 6 of IBM Tivoli Storage Manager products (product numbers 5608-E01, 5608-E02, and 5608-E03), and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this publication

This information helps you to optimize the performance of IBM® Tivoli® Storage Manager servers and clients, and identify and solve performance problems.

Standard subscription and support services from IBM do not include extensive performance analysis and tuning. The comprehensive analysis of a performance problem is a billable service that is offered to Tivoli Storage Manager customers. For more information, see the IBM Software Support Handbook.

Who should read this guide

The guide is intended for administrators who want to improve the performance of Tivoli Storage Manager servers and clients.

Before using this information, ensure that you are familiar with your Tivoli Storage Manager solution:

- How the Tivoli Storage Manager servers and clients are used and monitored
- The operating systems on which your Tivoli Storage Manager servers and clients run
- The networks that are in use for Tivoli Storage Manager server and client operations
- The storage devices that are used for Tivoli Storage Manager operations

Publications

The Tivoli Storage Manager product family includes IBM Tivoli Storage FlashCopy® Manager, IBM Tivoli Storage Manager for Space Management, IBM Tivoli Storage Manager for Databases, and several other storage management products from IBM.

To view IBM product documentation, see IBM Knowledge Center.

Part 1. Where to start

This information helps you to optimize the performance of IBM Tivoli Storage Manager servers and clients, and identify and solve performance problems.

Where you start in this information depends on what your goal is:

- If you are installing or upgrading a new server and clients, start with Part 2, "Configuration best practices," on page 9.
- If you need to investigate performance degradation, start with Part 3, "Solving performance problems," on page 45.

Before using this information, ensure that you are familiar with your Tivoli Storage Manager solution:

- How the Tivoli Storage Manager servers and clients are used and monitored
- The operating systems on which your Tivoli Storage Manager servers and clients run
- The networks that are in use for Tivoli Storage Manager server and client operations
- The storage devices that are used for Tivoli Storage Manager operations

Standard subscription and support services from IBM do not include extensive performance analysis and tuning. The comprehensive analysis of a performance problem is a billable service that is offered to Tivoli Storage Manager customers. For more information, see the IBM Software Support Handbook.

Chapter 1. Operating system and performance information cross-reference

Most of the performance information applies to any client or server, on any operating system. For certain operating systems, specific information about configuration of clients and servers for performance is available.

Table 1. Topics for Tivoli Storage Manager servers, by operating system

Server operating system	**Key topics**	**Topics specifically for the operating system**
AIX®	Chapter 3, “Configuring the server for optimal performance,” on page 11 Chapter 5, “Monitoring and maintaining the environment for performance,” on page 41 Chapter 8, “Identifying performance bottlenecks,” on page 61 Chapter 11, “Tuning server performance,” on page 121 Chapter 12, “Tuning disk storage for the server,” on page 159 Chapter 14, “Tuning network performance,” on page 231	“Tuning AIX systems for Tivoli Storage Manager server performance” on page 153 “Configuring AIX systems for disk performance” on page 170 “Monitoring performance with operating system tools” on page 42 “Setting network options for Tivoli Storage Manager on AIX systems” on page 233
HP-UX	Chapter 3, “Configuring the server for optimal performance,” on page 11 Chapter 5, “Monitoring and maintaining the environment for performance,” on page 41 Chapter 8, “Identifying performance bottlenecks,” on page 61 Chapter 11, “Tuning server performance,” on page 121 Chapter 12, “Tuning disk storage for the server,” on page 159 Chapter 14, “Tuning network performance,” on page 231	“Monitoring performance with operating system tools” on page 42

Table 1. Topics for Tivoli Storage Manager servers, by operating system (continued)

Server operating system	Key topics	Topics specifically for the operating system
Linux	Chapter 3, "Configuring the server for optimal performance," on page 11 Chapter 5, "Monitoring and maintaining the environment for performance," on page 41 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 11, "Tuning server performance," on page 121 Chapter 12, "Tuning disk storage for the server," on page 159 Chapter 14, "Tuning network performance," on page 231	"Tuning Linux systems for Tivoli Storage Manager server performance" on page 154 "Tuning Linux on System z systems for Tivoli Storage Manager server performance" on page 155 "Configuring Linux systems for disk performance" on page 170 "Monitoring performance with operating system tools" on page 42
Oracle Solaris	Chapter 3, "Configuring the server for optimal performance," on page 11 Chapter 5, "Monitoring and maintaining the environment for performance," on page 41 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 11, "Tuning server performance," on page 121 Chapter 12, "Tuning disk storage for the server," on page 159 Chapter 14, "Tuning network performance," on page 231	"Tuning Oracle Solaris systems for Tivoli Storage Manager server performance" on page 156 "Configuring Oracle Solaris systems for disk performance" on page 171 "Monitoring performance with operating system tools" on page 42
Windows	Chapter 3, "Configuring the server for optimal performance," on page 11 Chapter 5, "Monitoring and maintaining the environment for performance," on page 41 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 11, "Tuning server performance," on page 121 Chapter 12, "Tuning disk storage for the server," on page 159 Chapter 14, "Tuning network performance," on page 231	"Tuning Windows systems for Tivoli Storage Manager server performance" on page 156 "Monitoring performance with operating system tools" on page 42

Table 2. Topics for Tivoli Storage Manager clients, by operating system

Client operating system or environment	Key topics	Topics specifically for the operating system
AIX	Chapter 4, "Configuring clients for optimal performance," on page 39 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 13, "Tuning client performance," on page 173 Chapter 14, "Tuning network performance," on page 231	"Tuning journal-based backups" on page 213 "File space tuning" on page 221 "Tuning for Tivoli Storage Manager for Space Management" on page 239
HP-UX	Chapter 4, "Configuring clients for optimal performance," on page 39 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 13, "Tuning client performance," on page 173 Chapter 14, "Tuning network performance," on page 231	"File space tuning" on page 221 "Tuning for Tivoli Storage Manager for Space Management" on page 239
Linux	Chapter 4, "Configuring clients for optimal performance," on page 39 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 13, "Tuning client performance," on page 173 Chapter 14, "Tuning network performance," on page 231	"Tuning journal-based backups" on page 213 "File space tuning" on page 221 "Tuning for Tivoli Storage Manager for Space Management" on page 239
Mac OS X	Chapter 4, "Configuring clients for optimal performance," on page 39 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 13, "Tuning client performance," on page 173 Chapter 14, "Tuning network performance," on page 231	
Oracle Solaris	Chapter 4, "Configuring clients for optimal performance," on page 39 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 13, "Tuning client performance," on page 173 Chapter 14, "Tuning network performance," on page 231	"File space tuning" on page 221 "Tuning for Tivoli Storage Manager for Space Management" on page 239

Table 2. Topics for Tivoli Storage Manager clients, by operating system (continued)

Client operating system or environment	Key topics	Topics specifically for the operating system
VMware	"Tuning virtual machine backup operations" on page 223 "Resolving common performance problems with virtual machine backup operations" on page 192	
Windows	Chapter 4, "Configuring clients for optimal performance," on page 39 Chapter 8, "Identifying performance bottlenecks," on page 61 Chapter 13, "Tuning client performance," on page 173 Chapter 14, "Tuning network performance," on page 231	"Tuning journal-based backups" on page 213 "Windows system state backups" on page 223

Resources for operating system information

User groups and other sites can be good sources of information for tuning and solving problems for your operating system. The following list gives some examples.

AIX
Search resources from the AIX Virtual User Group at https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Power Systems/page/AIX Virtual User Group - USA.

Search for performance management and tuning information for AIX in the AIX product information.

Linux
Search for performance for Linux on System z® systems at http://www.ibm.com/developerworks/linux/linux390/perf/index.html.

Windows
Search for performance information for Windows hardware at http://msdn.microsoft.com/windows/hardware.

Chapter 2. Best practices for Version 7.1.6

Learn about best practices that you can implement for the Tivoli Storage Manager Version 7.1.6 server and client.

Inline data deduplication and directory-container storage pools

Use directory-container storage pools to protect backup and archive data. Data that is stored in a directory-container storage pool can use both inline data deduplication and inline compression.

Inline data deduplication or inline compression reduces data at the time it is stored. By using directory-container storage pools, you remove the need for volume reclamation, which improves server performance and reduces the cost of storage hardware. You can protect and repair data in directory-container storage pools at the level of the storage pool.

With Tivoli Storage Manager Version 7.1.6, you can convert a storage pool that uses a FILE type device class or a virtual tape library (VTL) to a directory-container storage pool.

Part 2. Configuration best practices

Typically, hardware configuration and selection have the most significant effect on the performance of a Tivoli Storage Manager solution. Other factors that affect performance are the operating system selection and configuration, and the configuration of Tivoli Storage Manager.

Procedure

- The following best practices are the most important for optimal performance and problem prevention.
- Review the table to determine the best practices that apply to your environment.

Best practice	More information
Use fast disks for the server database. Enterprise-grade solid-state disks (SSD), with Fibre Channel or SAS interface, offer the best performance.	Use fast, low-latency disks for the database. Using SSD is essential if you are using data deduplication and node replication. Avoid Serial Advanced Technology Attachment (SATA) and Parallel Advanced Technology Attachment (PATA) disks. For details and more tips, see the following topics: • “Checklist for server database disks” on page 17 • Choosing the correct type of storage technology
Ensure that the server system has enough memory.	Review operating system requirements in technote 1243309. Heavier workloads require more than the minimum requirements. Advanced features such as data deduplication and node replication can require more than the minimum memory that is specified in the system requirements document. If you plan to run multiple instances, each instance requires the memory that is listed for one server. Multiply the memory for one server by the number of instances that are planned for the system.
Separate the server database, the active log, the archive log, and disk storage pools from each other.	Keep all Tivoli Storage Manager storage resources on separate disks. Keep storage pool disks separate from the disks for the server database and logs. Storage pool operations can interfere with database operations when both are on the same disks. Ideally, the server database and logs are also separated from each other. For details and more tips, see the following topics: • “Checklist for server database disks” on page 17 • “Checklist for server recovery log disks” on page 19 • “Checklist for storage pools on DISK or FILE” on page 25
Use at least four directories for the server database. For larger servers or servers that use advanced features, use eight directories.	Place each directory on a LUN that is isolated from other LUNs and from other applications. A server is considered to be large if its database is larger than 2 TB or is expected to grow to that size. Use eight directories for such servers. See “Checklist for server database disks” on page 17.

Best practice	More information
If you are using data deduplication, node replication, or both, follow the guidelines for database configuration and other items.	Configure the server database according to the guidelines, because the database is extremely important to how well the server runs when these features are being used. For details and more tips, see the following topics: • "Checklist for container storage pools" on page 20 • "Checklist for data deduplication" on page 30 • "Checklist for node replication" on page 35
For storage pools that use `FILE` type device classes, follow the guidelines for the size of storage pool volumes. Typically, 50 GB volumes are best.	Review the information in "Optimal number and size of volumes for storage pools that use disk" on page 130 to help you to determine volume size. Configure storage pool devices and file systems based on throughput requirements, not only on capacity requirements. Isolate the storage devices that are used by Tivoli Storage Manager from other applications that have high I/O, and ensure that there is enough throughput to that storage. For more details, see "Checklist for storage pools on DISK or FILE" on page 25.
Schedule Tivoli Storage Manager client operations and server maintenance activities to avoid or minimize overlap of operations.	For more details, see the following topics: • "Tuning the schedule for daily operations" on page 132 • "Checklist for Tivoli Storage Manager server configuration" on page 27
Monitor operations constantly.	By monitoring, you can find problems early and more easily identify causes. Keep records of monitoring reports for up to a year to help you identify trends and plan for growth. See Chapter 5, "Monitoring and maintaining the environment for performance," on page 41.

Related concepts:

"Samples of data protection solutions" on page 57

Chapter 3. Configuring the server for optimal performance

Evaluate the characteristics and configuration of the system where the server is installed to ensure that the server is set up for good performance.

Before you begin

First, review basic requirements for a server. Then, review the following information for more details.

Procedure

1. Review the "Checklist for the server hardware and the operating system" on page 12. Correct items as needed.
2. Review the "Checklist for server database disks" on page 17. Correct items as needed.
3. Review the "Checklist for server recovery log disks" on page 19. This checklist covers the active log, archive log, and other logs. Correct items as needed.
4. Review the "Checklist for container storage pools" on page 20. Correct items as needed.
5. Review the "Checklist for storage pools on DISK or FILE" on page 25. Correct items as needed.
6. If you are getting new storage, test the storage system before you implement it. You can use tools to evaluate the characteristics of storage systems before you use them for the Tivoli Storage Manager database or storage pools. For more information, see "Analyzing the basic performance of disk systems" on page 73.
7. Review the tips for disk systems on specific operating systems. Operating systems can require different techniques for optimizing disk operations. For details, see "Configuring the operating system for disk performance" on page 170.
8. Review the "Checklist for Tivoli Storage Manager server configuration" on page 27 for tips about configuring schedules and other operations.
9. If you use data deduplication, review the "Checklist for data deduplication" on page 30.
10. If you use node replication, review the "Checklist for node replication" on page 35.

Related tasks:

"Grouping data by using collocation in server storage pools" on page 126

"Tuning the schedule for daily operations" on page 132

Checklist for the server hardware and the operating system

Use the checklist to verify that the system where the server is installed meets requirements for hardware and software configuration.

Question	Tasks, characteristics, options, or settings	More information
Does the operating system and hardware meet or exceed requirements? • Number and speed of processors • System memory • Supported operating system level	If you are using the minimum required amount of memory, you can support a minimal workload. You can experiment by adding more system memory to determine whether the performance is improved. Then, decide whether you want to keep the system memory dedicated to the server. Test the memory variations by using the entire daily cycle of the server workload. If you run multiple servers on the system, add the requirements for each server to get the requirements for the system. **Restriction:** For servers on IBM AIX operating systems, do not use Active Memory™ Expansion (AME). When you use AME, DB2® software uses 4 KB pages instead of 64 KB pages. Each 4 KB page must be decompressed when accessed, and compressed when not needed. When the compression or decompression occurs, DB2 and the server wait for access to the page, which degrades the server performance.	Review operating system requirements at Tivoli Storage Manager Supported Operating Systems. Additionally, review the guidance in Tuning tasks for operating systems and other applications. For more information about requirements when these features are in use, see the following topics: • Checklist for data deduplication • Checklist for node replication To help you determine whether processor or memory characteristics are the cause of performance problems, see Identifying server performance problems. For more information about sizing requirements for the server and storage, see the Tivoli Storage Manager Blueprint.
Are disks configured for optimal performance?	The amount of tuning that can be done for different disk systems varies. Ensure that the appropriate queue depths and other disk system options are set.	For more information, see the following topics: • Checklist for server database disks • Checklist for server recovery log disks • Checklist for storage pools on DISK or FILE

Question	Tasks, characteristics, options, or settings	More information
Does the server have enough memory?	Heavier workloads and advanced features such as data deduplication and node replication require more than the minimum system memory that is specified in the system requirements document. For databases that are not enabled for data deduplication, use the following guidelines to specify memory requirements: • For databases less than 500 GB, you need 16 GB of memory. • For databases with a size of 500 GB - 1 TB, you need 24 GB of memory. • For databases with a size of 1 TB - 1.5 TB, you need 32 GB of memory. • For databases greater than 1.5 TB, you need 40 GB of memory. Ensure that you allocate extra space for the active log and the archive log for replication processing.	For more information about requirements when these features are in use, see the following topics: • Checklist for data deduplication • Checklist for node replication • Memory requirements To help you determine whether processor or memory characteristics are the cause of performance problems, see Identifying server performance problems.
Does the system have enough host bus adapters (HBAs) to handle the data operations that the Tivoli Storage Manager server must run simultaneously?	Understand what operations require use of HBAs at the same time. For example, a server must store 1 GB/sec of backup data while also doing storage pool migration that requires 0.5 GB/sec capacity to complete. The HBAs must be able to handle all of the data at the speed required.	See Tuning HBA capacity.

Question	Tasks, characteristics, options, or settings	More information
Is network bandwidth greater than the planned maximum throughput for backups?	Network bandwidth must allow the system to complete operations such as backups in the time that is allowed or that meets service level commitments. For node replication, network bandwidth must be greater than the planned maximum throughput.	For more information, see the following topics: • Tuning network performance • Checklist for node replication

Question	Tasks, characteristics, options, or settings	More information
Are you using a preferred file system for Tivoli Storage Manager server files?	Use a file system that ensures optimal performance and data availability. The server uses direct I/O with file systems that support the feature. Using direct I/O can improve throughput and reduce processor use. The following list identifies the preferred file systems for each operating system: • AIX: Use the JFS2 file system with the `rbrw` option. • HP-UX: Use the VxFS file system. • Linux: Use either the ext3 or ext4 file system for the database, recovery log, and storage pool data. Use the following file system that is appropriate for your operating system and level: – For Red Hat Enterprise Linux x86_64, use the ext3 or ext4 file system. If Red Hat Enterprise Linux 6.4 or later is installed, use the ext4 file system. – For SUSE Linux Enterprise Server and for Red Hat Enterprise Linux ppc64, use the ext3 file system. • Oracle Solaris: Use the ZFS file system. • Windows: Use New Technology File System (NTFS) without compression.	For more information, see Configuring the operating system for disk performance.

Question	Tasks, characteristics, options, or settings	More information
Did you configure enough paging space?	Paging space, or swap space, extends the memory that is available for processing. When the amount of free RAM in the system is low, programs or data that is not in use are moved from memory to paging space. This action releases memory for other activities, such as database operations. For AIX, Linux, HP-UX, and Oracle Solaris systems, use a minimum of 32 GB of paging space or 50% of your RAM, whichever value is larger. On Windows systems, paging space is automatically configured.	
Did you tune kernel parameters after installation of the server?	You must tune kernel parameters for servers that are on HP-UX, Linux, or Oracle Solaris systems.	See the information about tuning kernel parameters for your operating system: • HP-UX: Tuning kernel parameters for HP-UX systems • Linux: Tuning kernel parameters for Linux systems • Solaris: Tuning kernel parameters for Oracle Solaris systems Not all operating systems require that kernel parameters be tuned.

Checklist for server database disks

Use the checklist to verify that the system where the server is installed meets requirements for hardware and software configuration.

Question	Tasks, characteristics, options, or settings	More information
Is the database on fast, low-latency disks?	Do not use the following drives for the Tivoli Storage Manager database: • Nearline SAS (NL-SAS) • Serial Advanced Technology Attachment (SATA) • Parallel Advanced Technology Attachment (PATA) Do not use internal disks that are included by default in most server hardware. Enterprise-grade solid-state disks (SSD), with Fibre Channel or SAS interface, offer the best performance. If you plan to use the data deduplication functions of Tivoli Storage Manager, focus on disk performance in terms of I/O operations per second (IOPS).	For more information, see Checklist for data deduplication
Is the database stored on disks or LUNs that are separate from disks or LUNs that are used for the active log, archive log, and storage pool volumes?	Separation of the server database from other server components helps reduce contention for the same resources by different operations that must run at the same time. **Tip:** The database and the archive log can share an array when you use solid-state drive (SSD) technology.	
If you are using RAID, did you select the optimal RAID level for your system? Did you define all LUNs with the same size and type of RAID?	When a system must do large numbers of writes, RAID 10 outperforms RAID 5. However, RAID 10 requires more disks than RAID 5 for the same amount of usable storage. If your disk system is RAID, define all your LUNs with the same size and type of RAID. For example, do not mix 4+1 RAID 5 with 4+2 RAID 6.	
If an option to set the strip size or segment size is available, did you optimize the size when you configured the disk system?	If you can set the strip size or segment size, use 64 KB or 128 KB sizes on disk systems for the database.	The block size that is used for the database varies depending on the table space. Most table spaces use 8 KB blocks, but some use 32 KB blocks.

Question	Tasks, characteristics, options, or settings	More information
Did you create at least four directories, also called storage paths, on four separate LUNs for the database? Create one directory per distinct array on the subsystem. If you have fewer than three arrays, create a separate LUN volume within the array.	Heavier workloads and use of some features require more database storage paths than the minimum requirements. Server operations such as data deduplication drive a high number of input/output operations per second (IOPS) for the database. Such operations perform better when the database has more directories. For server databases that are larger than 2 TB or are expected to grow to that size, use eight directories. Consider planned growth of the system when you determine how many storage paths to create. The server uses the higher number of storage paths more effectively if the storage paths are present when the server is first created. Use the *DB2_PARALLEL_IO* variable to force parallel I/O to occur on table spaces that have one container, or on table spaces that have containers on more than one physical disk. If you do not set the *DB2_PARALLEL_IO* variable, I/O parallelism is equal to the number of containers that are used by the table space. For example, if a table space spans four containers, the level of I/O parallelism that is used is 4.	For more information, see the following topics: • Checklist for data deduplication • Checklist for node replication For help with forecasting growth when the server deduplicates data, see technote 1596944. For the most recent information about database size, database reorganization, and performance considerations for Tivoli Storage Manager Version 6 and Version 7 servers, see technote 1452146. For information about setting the *DB2_PARALLEL_IO* variable, see Recommended settings for IBM DB2 registry variables.
Are all directories for the database the same size?	Directories that are all the same size ensure a consistent degree of parallelism for database operations. If one or more directories for the database are smaller than the others, they reduce the potential for optimized parallel prefetching. This guideline also applies if you must add storage paths after the initial configuration of the server.	
Did you raise the queue depth of the database LUNs on AIX systems?	The default queue depth is often too low.	See Configuring AIX systems for disk performance.

Related tasks:

"Choosing the correct type of storage technology for Tivoli Storage Manager" on page 161

Checklist for server recovery log disks

The recovery log for the server consists of the active log, the archive log, and optional logs for mirroring and failover. Use the checklist to verify that disk systems that are being used for the logs have the characteristics and configuration that are key to good performance.

Question	Tasks, characteristics, options, or settings	More information
Are the active log and archive log stored on disks or LUNs that are separate from what is used for the database and storage pool volumes?	Ensure that the disks where you place the active log are not used for other server or system purposes. Do not place the active log on disks that contain the server database, the archive log, or system files such as page or swap space.	Separation of the server database, active log, and archive log helps to reduce contention for the same resources by different operations that must run at the same time.
Are the logs on disks that have nonvolatile write cache?	Nonvolatile write cache allows data to be written to the logs as fast as possible. Faster write operations for the logs can improve performance for server operations.	
Did you set the logs to a size that adequately supports the workload?	If you are not sure about the workload, use the largest size that you can. **Active log** The maximum size is 512 GB, set with the **ACTIVELOGSIZE** server option. Ensure that there is at least 8 GB of free space on the active log file system after the fixed size active logs are created. **Archive log** The size of the archive log is limited by the size of the file system on which it is located, and not by a server option. Make the archive log at least as large as the active log.	• For log sizing details, see the recovery log information in technote 1421060. • For information about sizing when you use data deduplication, see Checklist for data deduplication.
Did you define an archive failover log? Did you place this log on a disk that is separate from the archive log?	The archive failover log is for emergency use by the server when the archive log becomes full. Slower disks can be used for the archive failover log.	Use the **ARCHFAILOVERLOGDIRECTORY** server option to specify the location of the archive failover log. Monitor the usage of the directory for the archive failover log. If the archive failover log must be used by the server, the space for the archive log might not be large enough.

Question	Tasks, characteristics, options, or settings	More information
If you are mirroring the active log, are you using only one type of mirroring?	You can mirror the log by using one of the following methods. Use only one type of mirroring for the log. • Use the **MIRRORLOGDIRECTORY** option that is available for the Tivoli Storage Manager server to specify a mirror location. • Use software mirroring, such as Logical Volume Manager (LVM) on AIX. • Use mirroring in the disk system hardware.	If you mirror the active log, ensure that the disks for both the active log and the mirror copy have equal speed and reliability. For more information, see Configuring the recovery log.

Related tasks:

"Choosing the correct type of storage technology for Tivoli Storage Manager" on page 161

Checklist for container storage pools

Review how your directory-container and cloud-container storage pools are set up to ensure optimal performance.

Question	Tasks, characteristics, options, or settings	More information
Measured in terms of input/output operations per second (IOPS), are you using fast disk storage for the Tivoli Storage Manager database?	Use a high-performance disk for the database. Use solid-state drive technology for data deduplication processing. Ensure that the database has a minimum capability of 3000 IOPS. For each TB of data that is backed up daily (before data deduplication), add 1000 IOPS to this minimum. For example, a Tivoli Storage Manager server that is ingesting 3 TB of data per day would need 6000 IOPS for the database disks: `3000 IOPS minimum + 3000 (3 TB x 1000 IOPS) = 6000 IOPS`	For recommendations about disk selection, see "Checklist for server database disks" on page 17 For more information about IOPS, see the Tivoli Storage Manager Blueprints.

Question	Tasks, characteristics, options, or settings	More information
Do you have enough memory for the size of your database?	Use a minimum of 40 GB of system memory for Tivoli Storage Manager servers, with a database size of 100 GB, that are deduplicating data. If the retained capacity of backup data grows, the memory requirement might need to be higher. Monitor memory usage regularly to determine whether more memory is required. Use more system memory to improve caching of database pages. The following memory size guidelines are based on the daily amount of new data that you back up: • 128 GB of system memory for daily backups of data, where the database size is 1 - 2 TB • 192 GB of system memory for daily backups of data, where the database size is 2 - 4 TB	Memory requirements
Have you properly sized the storage capacity for the database active log and archive log?	Configure the server to have a minimum active log size of 128 GB by setting the **ACTIVELOGSIZE** server option to a value of 131072. The suggested starting size for the archive log is 1 TB. The size of the archive log is limited by the size of the file system on which it is located, and not by a server option. Ensure that there is at least 10% extra disk space for the file system than the size of the archive log. Use a directory for the database archive logs with an initial free capacity of at least 1 TB. Specify the directory by using the **ARCHLOGDIRECTORY** server option. Define space for the archive failover log by using the **ARCHFAILOVERLOGDIRECTORY** server option.	For more information about sizing for your system, see the Tivoli Storage Manager Blueprints.

Question	Tasks, characteristics, options, or settings	More information
Is compression enabled for the archive log and database backups?	Enable the `ARCHLOGCOMPRESS` server option to save storage space. This compression option is different from inline compression. Inline compression is enabled by default with Tivoli Storage Manager V7.1.5 and later. **Restriction:** Do not use this option if the amount of backed up data exceeds 6 TB per day.	For more information about compression for your system, see the Tivoli Storage Manager Blueprints.
Are the Tivoli Storage Manager database and logs on separate disk volumes (LUNs)? Is the disk that is used for the database configured according to best practices for a transactional database?	The database must not share disk volumes with Tivoli Storage Manager database logs or storage pools, or with any other application or file system.	For more information about server database and recovery log configuration, see "Server database and recovery log configuration and tuning" on page 121.
Are you using a minimum of eight (2.2 GHz or equivalent) processor cores for each Tivoli Storage Manager server that you plan to use with data deduplication?	If you are planning to use client-side data deduplication, verify that client systems have adequate resources available during a backup operation to complete data deduplication processing. Use a processor that is at least the minimum equivalent of one 2.2 GHz processor core per backup process with client-side data deduplication.	• Effective planning and use of deduplication • Tivoli Storage Manager Blueprints
Is there enough storage space for the database?	For a rough estimate, plan for 100 GB of database storage for every 50 TB of data that is to be protected in deduplicated storage pools. *Protected data* is the amount of data before data deduplication, including all versions of objects stored. As a best practice, define a new container storage pool exclusively for data deduplication. Data deduplication occurs at the storage-pool level, and all data within a storage pool, except encrypted data, is deduplicated.	

Question	Tasks, characteristics, options, or settings	More information
Have you estimated storage pool capacity to configure enough space for the size of your environment?	You can estimate capacity requirements for a deduplicated storage pool by using the following technique: 1. Estimate the base size of the source data. 2. Estimate the daily backup size by using an estimated change and growth rate. 3. Determine retention requirements. 4. Estimate the total amount of source data by factoring in the base size, daily backup size, and retention requirements. 5. Apply the deduplication ratio factor. 6. Apply the compression ratio factor. 7. Round up the estimate to consider transient storage pool usage.	For an example of using this technique, see Effective planning and use of deduplication.
Have you distributed disk I/O over many disk devices and controllers?	Use arrays that consist of as many disks as possible, which is sometimes referred to as wide striping. Ensure that you use one database directory per distinct array on the subsystem. Set the *DB2_PARALLEL_IO* registry variable to enable parallel I/O for each table space used if the containers in the table space span multiple physical disks. When I/O bandwidth is available and the files are large, for example 1 MB, the process of finding duplicates can occupy the resources of an entire processor. When files are smaller, other bottlenecks can occur. Specify eight or more file systems for the deduplicated storage pool device class so that I/O is distributed across as many LUNs and physical devices as possible.	For guidelines about setting up storage pools, see "Checklist for storage pools on DISK or FILE" on page 25. For information about setting the *DB2_PARALLEL_IO* variable, see Recommended settings for IBM DB2 registry variables.
Have you scheduled daily operations based on your backup strategy?	The best practice sequence of operations is in the following order: 1. Client backup 2. Storage pool protection 3. Node replication 4. Database backup 5. Expire inventory	• "Scheduling data deduplication and node replication processes" on page 136 • "Daily operations for directory-container storage pools" on page 133

Question	Tasks, characteristics, options, or settings	More information
Do you have enough storage to manage the DB2 lock list?	If you deduplicate data that includes large files or large numbers of files concurrently, the process can result in insufficient storage space. When the lock list storage is insufficient, backup failures, data management process failures, or server outages can occur. File sizes greater than 500 GB that are processed by data deduplication are most likely to deplete storage space. However, if many backup operations use client-side data deduplication, this problem can also occur with smaller-sized files.	For information about tuning the DB2 **LOCKLIST** parameter, see "Tuning server-side data deduplication" on page 147.
Is sufficient bandwidth available to transfer data to a Tivoli Storage Manager server?	To transfer data to a Tivoli Storage Manager server, use client-side or server-side data deduplication and compression to reduce the bandwidth that is required. Use a V7.1.5 server or higher to use inline compression and use a V7.1.6 or later client to enable enhanced compression processing.	For more information, see the **enablededup** client option.
Have you determined how many storage pool directories to assign to each storage pool?	Assign directories to a storage pool by using the **DEFINE STGPOOLDIRECTORY** command. Create multiple storage pool directories and ensure that each directory is backed up to a separate disk volume (LUN).	

Checklist for storage pools on DISK or FILE

Use the checklist to review how your disk storage pools are set up. This checklist includes tips for storage pools that use DISK or FILE device classes.

Question	Tasks, characteristics, options, or settings	More information
Can the storage pool LUNs sustain throughput rates for 256 KB sequential reads and writes to adequately handle the workload within the time constraints?	When you are planning for peak loads, consider all the data that you want the server to read or write to the disk storage pools simultaneously. For example, consider the peak flow of data from client backup operations and server data-movement operations such as migration that run at the same time. The Tivoli Storage Manager server reads and writes to storage pools predominantly in 256 KB blocks. If the disk system includes the capability, configure the disk system for optimal performance with sequential read/write operations rather than random read/write operations.	For more information, see Analyzing the basic performance of disk systems.
Is the disk configured to use read and write cache?	Use more cache for better performance.	
For storage pools that use FILE device classes, have you determined a good size to use for the storage pool volumes?	Review the information in “Optimal number and size of volumes for storage pools that use disk” on page 130. If you do not have the information to estimate a size for FILE device class volumes, start with volumes that are 50 GB.	Typically, problems arise more frequently when the volumes are too small. Few problems are reported when volumes are larger than needed. When you determine the volume size to use, as a precaution choose a size that might be larger than necessary.
For storage pools that use FILE device classes, are you using preallocated volumes?	Scratch volumes can cause file fragmentation. To ensure that a storage pool does not run out of volumes, set the **MAXSCRATCH** parameter to a value greater than zero.	Use the **DEFINE VOLUME** server command to preallocate volumes in the storage pool. Use the **DEFINE STGPOOL** or **UPDATE STGPOOL** server command to set the **MAXSCRATCH** parameter.
For storage pools that use FILE device classes, did you compare the maximum number of client sessions to the number of volumes that are defined?	Always maintain enough usable volumes in the storage pools to allow for the expected peak number of client sessions that run at one time. The volumes might be scratch volumes, empty volumes, or partly filled volumes.	For storage pools that use FILE device classes, only one session or process can write to a volume at the same time.

Question	Tasks, characteristics, options, or settings	More information
For storage pools that use FILE device classes, have you set the **MOUNTLIMIT** parameter of the device class to a value that is high enough to account for the number of volumes that might be mounted in parallel?	For storage pools that use data deduplication, the **MOUNTLIMIT** parameter is typically in the range of 500 - 1000. Set the value for **MOUNTLIMIT** to the maximum number of mount points that are needed for all active sessions. Consider parameters that affect the maximum number of mount points that are needed: • The **MAXSESSIONS** server option, which is the maximum number of Tivoli Storage Manager sessions that can run concurrently. • The **MAXNUMMP** parameter, which sets the maximum number of mount points that each client node can use. For example, if the maximum number of client node backup sessions is typically 100 and each of the nodes has MAXNUMMP=2, multiply 100 nodes by the 2 mount points for each node to get the value of 200 for the **MOUNTLIMIT** parameter.	Use the **REGISTER NODE** or **UPDATE NODE** server command to set the **MAXNUMMP** parameter for client nodes.
For storage pools that use DISK device classes, have you determined how many storage pool volumes to put on each file system?	How you configure the storage for a storage pool that uses a DISK device class depends on whether you are using RAID for the disk system. If you are not using RAID, then configure one file system per physical disk, and define one storage pool volume for each file system. If you are using RAID 5 with *n* + 1 volumes, configure the storage in one of the following ways: • Configure *n* file systems on the LUN and define one storage pool volume per file system. • Configure one file system and *n* storage pool volumes for the LUN.	For an example layout that follows this guideline, see Figure 28 on page 168.
Did you create your storage pools to distribute I/O across multiple file systems?	Ensure that each file system is on a different LUN on the disk system. Typically, having 10 - 30 file systems is a good goal, but ensure that the file systems are no smaller than approximately 250 GB.	For details, see the following topics: • Chapter 12, "Tuning disk storage for the server," on page 159 • "Tuning and configuring storage pools and volumes" on page 125

Related tasks:

"Tuning and configuring storage pools and volumes" on page 125

"Choosing the correct type of storage technology for Tivoli Storage Manager" on page 161

Checklist for Tivoli Storage Manager server configuration

Evaluate key configuration settings and scheduling that can affect performance for the Tivoli Storage Manager server.

Question	Tasks, characteristics, options, or settings	More information
Have you set up server schedules so that critical operations do not interfere with each other?	Schedule operations that might otherwise be automatically started: • Disable automatic expiration by setting the **EXPINTERVAL** server option to 0. • Configure storage pools so that storage-pool migration, reclamation, and duplicate-identification operations are not automatically started. • Schedule each type of server data maintenance task with controlled start times and durations so that they do not overlap with each other. **Restriction:** You cannot use expiration, migration, reclamation, or duplicate-identification operations with container storage pools. Schedule storage pool protection before replication processing.Schedule node replication to avoid or minimize overlap with client backups.	"Tuning the schedule for daily operations" on page 132

Question	Tasks, characteristics, options, or settings	More information
Are you running enough processes to handle data operations in your environment?	Verify that the number of processes for an operation is enough to complete the workload. For example, if performance for reclamation seems slow, tune the number of parallel processes that are allocated for that operation. Use the following commands and parameters to control processes for different operations: • Storage pool backup processes: **MAXPROCESS** parameter on the **BACKUP STGPOOL** command • Duplicate identification processes: **NUMPROCESS** parameter on the **IDENTIFY DUPLICATES** command. • Migration activity: **MIGPROCESS** parameter on the **DEFINE STGPOOL** command • Parallel expiration activity: **RESOURCES** parameter on the **EXPIRE INVENTORY** command • Reclamation processes: **RECLAIMPROCESS** parameter on the **DEFINE STGPOOL** command Continue increasing parallel processes to the point where a resource on the server becomes saturated. **Restriction:** You cannot identify duplicates, migrate data, expire data, reclaim data, export data, or import data with container storage pools. Use the **PROTECT STGPOOL** command to protect data in container storage pools. Schedule storage pool protection before replication processing.	For more information, see the following topics: • "Improving the speed of database backups" on page 124 • "Multiple session backup and restore" on page 209

<table>
<tr><th>Question</th><th>Tasks, characteristics, options, or settings</th><th>More information</th></tr>
<tr><td>Are client backup schedules configured so that backups are spread across the available time?</td><td>Schedule client backups in a backup window that is isolated from all data maintenance processes, such as storage-pool migration, reclamation, and duplicate identification processing.

If possible, schedule client backups so that they are not all started at one time. You might not have to stagger schedules if sufficient server resources are available to process all client backups.

Also, if you are using client-side data deduplication and there is commonality in the data that is being backed up, you might not have to stagger schedules.</td><td>"Avoiding contention for server resources during client operations" on page 142</td></tr>
<tr><td>Are server option values updated from the defaults for optimum performance?</td><td>Set the EXPINTERVAL server option to 0 and schedule inventory expiration processing.

Set the MAXSESSIONS server option to a value no higher than 1000, which is the maximum that was tested in IBM labs. Setting the value higher than what is required for the maximum number of sessions that you expect can unnecessarily consume memory on the server.</td><td>"Limits for the server database size and peak client sessions" on page 56</td></tr>
<tr><td>Have you set up a schedule for your database backups?

Have you configured backup operations appropriately for the size of your database?</td><td>When you set up a schedule for database backup, you have more control over when server resources are engaged. Schedule database backups to run after both the client backup, and if used, the storage pool backup.

Perform only full database backups, not incremental backups.

For databases over 500 GB, use multistreaming for database backups to improve performance.

Make the archive log directory for the database large enough so that you do not run out of space between database backups and so that only one database backup is required every 24 hours. Under normal conditions, do not back up the database at unscheduled times.</td><td>For more information, see the following topics:
• "Tuning the schedule for daily operations" on page 132
• "Improving the speed of database backups" on page 124</td></tr>
</table>

Question	Tasks, characteristics, options, or settings	More information
Have you formatted disk storage pool volumes sequentially if they are placed on the same file system?	Formatting volumes sequentially helps avoid disk fragmentation and improves sequential read and write performance. To format multiple sequential disk pool volumes, use the **DEFINE VOLUME** command and specify a value for the **NUMBEROFVOLUMES** parameter. Each volume is allocated sequentially to avoid fragmentation.	"Checklist for storage pools on DISK or FILE" on page 25

Checklist for data deduplication

Data deduplication requires more processing resources on the server or client. Use the checklist to verify that hardware and your Tivoli Storage Manager configuration have characteristics that are key to good performance.

Question	Tasks, characteristics, options, or settings	More information
Are you using fast disk storage for the Tivoli Storage Manager database as measured in terms of input/output operations per second (IOPS)?	Use a high-performance disk for the Tivoli Storage Manager database. At a minimum, use 10,000 rpm drives for smaller databases that are 200 GB or less. For databases over 500 GB, use 15,000 rpm drives or solid-state drives. Ensure that the Tivoli Storage Manager database has a minimum capability of 3000 IOPS. For each TB of data that is backed up daily (before data deduplication), include an extra 1000 IOPS to this minimum. For example, a Tivoli Storage Manager server that is ingesting 3 TB of data per day would need 6000 IOPS for the database disks: `3000 IOPS minimum + 3000 (3 TB x 1000 IOPS) = 6000 IOPS`	"Checklist for server database disks" on page 17 For more information about IOPS, see the Tivoli Storage Manager Blueprint at IBM Spectrum Protect Blueprints

Question	Tasks, characteristics, options, or settings	More information
Do you have enough memory for the size of your database?	Use a minimum of 64 GB of system memory for Tivoli Storage Manager servers that are deduplicating data. If the retained capacity of backup data grows, the memory requirement might need to be higher. Monitor memory usage regularly to determine whether more memory is required. Use more system memory to improve caching of database pages. The following memory size guidelines are based on the daily amount of new data that you back up: • 128 GB of system memory for daily backups of data, where the database size is 1 - 2 TB • 192 GB of system memory for daily backups of data, where the database size is 2 - 4 TB	"Memory requirements" on page 131
Have you properly sized the storage capacity for the database active log and archive log?	The suggested starting size for the active log is 16 GB. Configure the server to have an maximum active log size of 128 GB by setting the **ACTIVELOGSIZE** server option to a value of 131072. The suggested starting size for the archive log is 48 GB. The size of the archive log is limited by the size of the file system on which it is located, and not by a server option. Make the archive log at least as large as the active log. Use a directory for the database archive logs with an initial free capacity of at least 500 GB. Specify the directory by using the **ARCHLOGDIRECTORY** server option. Define space for the archive failover log by using the **ARCHFAILOVERLOGDIRECTORY** server option.	
Are the Tivoli Storage Manager database and logs on separate disk volumes (LUNs)? Is the disk that is used for the database configured according to best practices for a transactional database?	The database must not share disk volumes with Tivoli Storage Manager database logs or storage pools, or with any other application or file system.	See "Server database and recovery log configuration and tuning" on page 121

Question	Tasks, characteristics, options, or settings	More information
Are you using a minimum of eight (2.2 GHz or equivalent) processor cores for each Tivoli Storage Manager server that you plan to use with data deduplication?	If you are planning to use client-side data deduplication, verify that client systems have adequate resources available during a backup operation to complete data deduplication processing. Use a processor that is at least the minimum equivalent of one 2.2 GHz processor core per backup process with client-side data deduplication.	Effective Planning and Use of IBM Tivoli Storage Manager V6 and V7 Deduplication.
Have you properly sized disk space for storage pools?	For a rough estimate, plan for 100 GB of database storage for every 10 TB of data that is to be protected in deduplicated storage pools. *Protected data* is the amount of data before deduplication, including all versions of objects stored. As a best practice, define a new container storage pool exclusively for data deduplication. Data deduplication occurs at the storage-pool level, and all data within a storage pool, except encrypted data, is deduplicated.	"Checklist for container storage pools" on page 20
Have you estimated storage pool capacity to configure enough space for the size of your environment?	You can estimate capacity requirements for a deduplicated storage pool by using the following technique: 1. Estimate the base size of the source data. 2. Estimate the daily backup size by using an estimated change and growth rate. 3. Determine retention requirements. 4. Estimate the total amount of source data by factoring in the base size, daily backup size, and retention requirements. 5. Apply the deduplication ratio factor. 6. Round up the estimate to consider transient storage pool usage.	Effective Planning and Use of IBM Tivoli Storage Manager V6 and V7 Deduplication.

Question	Tasks, characteristics, options, or settings	More information
Have you distributed disk I/O over many disk devices and controllers?	Use arrays that consist of as many disks as possible, which is sometimes referred to as wide striping. When I/O bandwidth is available and the files are large, for example 1 MB, the process of finding duplicates can occupy the resources of an entire processor during a session or process. When files are smaller, other bottlenecks can occur. Specify eight or more file systems for the deduplicated storage pool device class so that I/O is distributed across as many LUNs and physical devices as possible.	See "Checklist for storage pools on DISK or FILE" on page 25.
Have you scheduled data deduplication processing based on your backup strategy?	If you are not creating a secondary copy of backup data or if you are using node replication for the second copy, client backup and duplicate identification can be overlapped. This can reduce the total elapsed time for these operations, but might increase the time that is required for client backup. If you are using storage pool backup, do not overlap client backup and duplicate identification. The best practice sequence of operations is client backup, storage pool backup, and then duplicate identification. For data that is not stored with client-side data deduplication, schedule storage-pool backup operations to complete before you start data deduplication processing. Set up your schedule this way to avoid reconstructing objects that are deduplicated to make a non-deduplicated copy to a different storage pool. Consider doubling the time that you allow for backups when you use client-side data deduplication in an environment that is not limited by the network. Ensure that you schedule data deduplication before you schedule compression.	See "Scheduling data deduplication and node replication processes" on page 136.

Question	Tasks, characteristics, options, or settings	More information
Are the processes for identifying duplicates able to handle all new data that is backed up each day?	If the process completes, or goes into an idle state before the next scheduled operation begins, then all new data is being processed. The duplicate identification (IDENTIFY) processes can increase the workload on the processor and system memory. If you use a container storage pool for data deduplication, duplicate identification processing is not required. If you update an existing storage pool, you can specify 0 - 20 duplicate identification processes to start automatically. If you do not specify any duplicate-identification processes, you must start and stop processes manually.	
Is reclamation able to run to a sufficiently low threshold?	If a low threshold cannot be reached, consider the following actions: • Increase the number of processes that are used for reclamation. • Upgrade to faster hardware.	
Do you have enough storage to manage the DB2 lock list?	If you deduplicate data that includes large files or large numbers of files concurrently, the process can result in insufficient storage space. When the lock list storage is insufficient, backup failures, data management process failures, or server outages can occur. File sizes greater than 500 GB that are processed by data deduplication are most likely to deplete storage space. However, if many backup operations use client-side data deduplication, this problem can also occur with smaller-sized files.	For information about tuning the DB2 **LOCKLIST** parameter, see "Tuning server-side data deduplication" on page 147.

Question	Tasks, characteristics, options, or settings	More information
Is deduplication cleanup processing able to clean out the dereferenced extents to free disk space before the start of the next backup cycle?	Run the **SHOW DEDUPDELETE** command. The output shows that all threads are idle when the workload is complete. If cleanup processing cannot complete, consider the following actions: • Increase the number of processes that are used for duplicate identification. • Upgrade to faster hardware. • Determine whether the Tivoli Storage Manager server is ingesting more data than it can process with data deduplication and consider deploying an extra Tivoli Storage Manager server.	
Is sufficient bandwidth available to transfer data to a Tivoli Storage Manager server?	Use client-side data deduplication and compression to reduce the bandwidth that is required to transfer data to a Tivoli Storage Manager server.	For more information, see the **enablededupcache** client option.

For planning and best practice information, see *Effective Planning and Use of IBM Tivoli Storage Manager V6 and V7 Deduplication* at Effective Planning and Use of IBM Tivoli Storage Manager V6 and V7 Deduplication.

Related tasks:

"Evaluating data deduplication results" on page 67

"Tuning server-side data deduplication" on page 147

"Tuning client-side data deduplication" on page 200

Checklist for node replication

A successful implementation of node replication relies on sufficient, dedicated hardware resources. Increased amounts of memory and processor cores are required. The database and its logs must be appropriately sized to ensure that transactions can complete. A dedicated network, with enough bandwidth to handle the amount of data you intend to replicate, is required.

Use the checklist to verify that hardware and your Tivoli Storage Manager configuration have characteristics that are key to good performance.

Question	Tasks, characteristics, options, or settings	More information
Are you using high-performance disk for the Tivoli Storage Manager database?	Ensure that the disks that are configured for the Tivoli Storage Manager database have a minimum capability of 3,000 I/O operations per second (IOPS). For each TB of data that is backed up daily (before data deduplication), add 1,000 IOPS to this minimum. For example, a Tivoli Storage Manager server that is ingesting 3 TB of data per day would need 6,000 IOPS for the database disks: `3,000 IOPS minimum + 3,000 (3 TB x 1,000 IOPS) = 6,000 IOPS`	"Checklist for server database disks" on page 17
Are you using enough processor cores and memory for node replication and optionally, data deduplication?	If you are using node replication without deduplication, use a minimum of 4 processor cores and 64 GB of RAM for both the source and the target servers. For any server that is configured for node replication and data deduplication, use a minimum of 8 processor cores and 128 GB of RAM.	
Have you properly sized your disk space for the database, logs, and storage pools?	To determine whether your database can handle the additional space requirements, you must first estimate how much more database space node replication uses. For the active log, use a minimum size of 64 GB for node replication. Use the maximum allowed size for the active log, which is 128 GB if you are also using data deduplication. Make the archive log space at least as large as the space defined for the active log. Also, specify a directory for the archive failover log in case it is needed.	Determining server database requirements for node replication (V7.1.1)
Is your network capable of handling the additional traffic for the amount of data that you intend to replicate between source and target servers?	For node replication, network bandwidth must be greater than the planned maximum throughput. You can estimate network bandwidth that is based on the amount of data that you are replicating.	Estimating network bandwidth for node replication (V7.1.1)
Are you using data deduplication with node replication?	By using data deduplication with node replication, you reduce the bandwidth that is required for replication operations. Data deduplication reduces the amount of data that is sent to the target of the replication operation.	Measuring effects of data deduplication on node replication processing (V7.1.1)

Question	Tasks, characteristics, options, or settings	More information
Have you scheduled node replication in the optimum order for the daily schedule?	Ensure that you are running node replication after client backup. Complete data deduplication processing before replication processing. Schedule compression after replication.	For more information, see the following topics: • "Scheduling data deduplication and node replication processes" on page 136 • "Compatibility and resource usage for server processes" on page 138
Have you optimized the number of sessions that are used for sending data to the target replication server?	You can improve replication performance by using the **MAXSESSIONS** parameter on the **REPLICATE NODE** command to specify data sessions. The number of sessions that are used for replication depends on the amount of data that you are replicating.	Managing the number of replication sessions (V7.1.1)
Do you have enough mount points to avoid stalled replication servers and other server processes?	Determine the number of logical and physical drives that can be dedicated to the replication process. For example, if a library has ten tape drives and four of the drives are used for another task, there are six available drives for node replication. Specify the number of mount points you require and ensure that there are drives available to complete node replication.	Typically tape is not used for node replication except for the initial replication.
Do the node replication processes complete replicating all newly ingested data before the beginning of the next backup cycle?	If replication processes cannot finish before the start of the next backup cycle, consider the following actions: • Ensure that there are sufficient mount points and drives available for node replication processes to complete. • Increase the number of data sessions that are used for node replication. • Upgrade to faster hardware and more bandwidth for the source and target servers.	
If you are using data deduplication with node replication, do the processes for identifying duplicates complete before the start of node replication processing so that data deduplication is used to its full advantage?	If the process completes, or goes into an idle state before node replication begins, then all new data is being processed.	

For more information about planning for node replication, see Planning for node replication (V7.1.1).

Chapter 4. Configuring clients for optimal performance

You can configure the client system to ensure that the client is set up for good performance.

Procedure

Use the information in the following table to help you configure the client for optimal performance.

Action	More information
Ensure that the client system meets the minimum hardware and software requirements.	For information about client requirements, see Client environment requirements.
Ensure that you are using the appropriate method to back up the data in your environment.	See "Selecting the optimal client backup method" on page 173.
If client options were changed from the default values, note them for further analysis. Some problems can be resolved by setting the client option values to the default values.	For information about client options, see Processing options.
Look for solutions for common client performance problems.	For information about solving common client performance problems, see "Common client performance problems" on page 191.
Fine-tune the client by adjusting the values of client options that affect performance.	See Chapter 13, "Tuning client performance," on page 173

Chapter 5. Monitoring and maintaining the environment for performance

By monitoring server and client operations constantly, you can find problems early and more easily identify causes. Keep records of monitoring reports for up to a year to help you identify trends and plan for growth.

Procedure

- Use the server monitoring script, `servermonV6.pl`. This script includes server and system commands that help you monitor a server. To download the script and see more details, see technote 1432937.
- Use monitoring tools to verify that client and server operations are completed successfully and within a time that meets your business requirements. For monitoring tips, see the Daily monitoring checklist.
  1. Verify that server maintenance processes such as database backup and storage pool backup are completed successfully. Investigate failed or incomplete processes.
  2. Verify that client backups are successful and complete. Investigate failed or incomplete client backups, especially for clients that are most critical to your overall operations.
- If you are using data deduplication, node replication, or both, verify that processes that are related to these features are finishing. For example:
  – For server-side data deduplication, verify that the duplicate identification processes can handle all of the new data that the server stores each day. If the processes finish or go to an idle state before subsequent operations such as reclamation begin, you know that the processes can handle the new data.
  – For node replication, verify that replication processes finish replicating all newly stored data before the start of client backups on the next day.
  – If you are using both data deduplication and node replication, verify that the duplicate identification processes can finish before the start of node replication. This sequence ensures that you are taking full advantage of data deduplication.
- Keep your Tivoli Storage Manager software up to date. Go to the IBM Support Portal and search for fix packs that might be useful to apply to your server, clients, or both.
- Keep other software and hardware products in your environment up to date. For software and hardware products in your environment other than Tivoli Storage Manager, review service levels and firmware periodically and update them as needed. Completing such a review every six months is a typical goal. However, review and apply security fixes monthly or as needed (for example, on an emergency basis).

Related concepts:

"Checklist for data deduplication" on page 30

"Checklist for node replication" on page 35

Related tasks:

"Server monitoring script" on page 80

Monitoring performance with operating system tools

Monitor your Tivoli Storage Manager solution so that you know when you must investigate performance changes. Operating systems have different tools that are available for monitoring performance. Simulating workloads to test performance is another useful task to learn.

Procedure

- To monitor system processors and storage for usage and the effects of Tivoli Storage Manager activities, you can use the following commands and tools.

  Tip: The server monitoring script, `servermonV6.pl`, includes some of the operating system commands that are in the following lists.

  AIX systems
  For information about a command, search for the command in the product information for the version of AIX that you are using.

Command or tool	Purpose	More information
iostat command	Statistics about input/output for the entire system and for devices that are attached to the system	
lparstat command	Reports about logical partition (LPAR) configuration and statistics	
nmon command	System monitoring reports	For information about nmon Analyzer and other tools that help you analyze the data from the **nmon** command, search at AIX performance analysis and tuning.
nstress package of tools	Stress testing of the system	Search for the latest **nstress** package at AIX performance analysis and tuning.
perfpmr script	A data collection script, typically used before you report a problem to IBM Software Support	Search for information about the script in the product information for the version of AIX that you are using.
sar command	System activity monitoring	
vmstat command	Virtual memory statistics	
Iometer, an open source tool	Measurement and characterization of the input/output capabilities of a system	For information, see www.iometer.org.
Netperf benchmark	Tools to help you measure bandwidth and latency of networks	For information, see www.netperf.org.

  HP-UX systems
  For information about the commands, see the operating system documentation.

Command or tool	Purpose
iostat command	Statistics about input/output for the entire system and for devices that are attached to the system
sar command	System activity monitoring

Command or tool	Purpose
svmon command	Memory usage monitoring
vmstat command	Virtual memory statistics
Iometer, an open source tool	Measurement and characterization of the input/output capabilities of a system For information, see www.iometer.org.

Linux systems
For information about the commands, see the operating system documentation.

Command or tool	Purpose
iostat command	Statistics about input/output for the entire system and for devices that are attached to the system
nmon command	System monitoring reports
sar command	System activity monitoring
Iometer, an open source tool	Measurement and characterization of the input/output capabilities of a system For information, see www.iometer.org.
Netperf benchmark	Tools to help you measure bandwidth and latency of networks For information, see www.netperf.org.

Oracle Solaris systems
For information about the commands, see the operating system documentation.

Command or tool	Purpose
iostat command	Statistics about input/output for the entire system and for devices that are attached to the system
sar command	System activity monitoring
svmon command	Memory usage monitoring
vmstat command	Virtual memory statistics
Iometer, an open source tool	Measurement and characterization of the input/output capabilities of a system For information, see www.iometer.org.

Windows systems

Command or tool	Purpose
Windows Performance Monitor (**perfmon** command)	Performance monitoring of the system and attached devices For information, see the operating system documentation.
Iometer, an open source tool	Measurement and characterization of the input/output capabilities of a system For information, see www.iometer.org.

Command or tool	Purpose
Netperf benchmark	Tools to help you measure bandwidth and latency of networks For information, see www.netperf.org.

- To help you understand the performance of the Tivoli Storage Manager environment, consider the following tools. These tools can be useful to determine performance under ideal conditions. However, they simulate only some of the operations that occur during Tivoli Storage Manager activities.

  FTP FTP is available on almost any system. You can use FTP to estimate the throughput that Tivoli Storage Manager might get on a backup or restore operation. The results of the test are only an approximation.

  To use FTP to evaluate performance, create or use an existing file and use FTP to transfer it. Use a single file larger than 200 MB for the test. If the operation involves smaller files, then the results from a test with FTP might not be accurate.

  You might have to manually time how long the transfer takes to calculate throughput. Include these operations in the time estimates:

  – Read from disk

  – Send over network

  – Write to disk

  An alternative to FTP is SCP. However, SCP encrypts the data so it might not perform as well as FTP.

  dd The command is available on systems such as AIX and Linux to initiate disk reads or writes.

Related tasks:

"Analyzing data flow with the **dd** command" on page 73

Part 3. Solving performance problems

When you notice degradation in performance of a Tivoli Storage Manager solution, start by verifying that conditions outside of the server and client are not the cause. Learn about problem symptoms and causes, and how to use tools to identify them.

Chapter 6. Performance tuning and bottleneck identification

When you tune a Tivoli Storage Manager solution, you must look at all of its components and their configuration. Performance degradation in key operations, including client backups, storage pool migration, and expiration, can be the result of insufficient resources, poor configuration, or both.

The following variables must be examined:

- Server hardware and its configuration
  - Processors, memory, cache, and storage backplane
  - Internal and external storage resources, including disk systems for the server database, recovery logs, and storage pools

  Any hardware component that is in the data path might be the bottleneck. For illustrations of the data paths and the possible components, see "Potential bottlenecks in the data flow for Tivoli Storage Manager operations" on page 49.
- The network that is used for communications and data transfers among the components
- Client hardware and its configuration, and the characteristics of the client data that is being protected

The best way to begin performance tuning is to provide sufficient resources for and optimally configure the server and clients. For example, for a server provide enough system memory, disk systems that are properly sized and configured to handle the workload, database and logs that are properly separated, and correct operating system settings. For backup-archive clients, key items include enough memory, adequate network bandwidth, and careful choice and configuration of backup methods.

To find bottlenecks and identify ways to improve performance, you can use both built-in tools for systems and storage devices, and Tivoli Storage Manager tools.

This performance information provides guidelines for the best performance. Also included are procedures and information about analysis tools to identify performance problems.

Standard subscription and support services from IBM do not include extensive performance analysis and tuning. The comprehensive analysis of a performance problem is a billable service that is offered to Tivoli Storage Manager customers. For more information, see the IBM Software Support Handbook.

Performance tuning guidelines and expectations

Performance tuning is not a one-time task but an ongoing effort. Because the system environment and client workloads change, you must continually monitor and periodically tune the solution.

Because the performance of a Tivoli Storage Manager solution can be affected by so many factors, make changes in a controlled way. Evaluate the effects of the changes that you introduce by collecting appropriate measurements before and after changes.

For example, the following approach can be effective:

1. Track performance of the solution over time by establishing an initial baseline of measurements of operational performance. Periodically collect the same measurements and compare the results.
2. Implement a method to track all changes that are made to the Tivoli Storage Manager solution.

   Use strict change control to help you understand the performance impact of any change.

   Limit the changes that you make at one time so that you can more easily determine what makes a difference.
3. After a change and before you make further changes, observe system operations and performance over enough time to evaluate the full effect of the changes.

   Observe the system over a time span that is based on typical cycles of operations. For example, if you always have a once-a-week peak in client backup operations, be sure that peak time is included in your observations.
4. Evaluate the results before you make further changes.

Most performance tuning yields limited improvements. Carefully consider how much time is reasonable to spend on improving system performance. Service Level Agreements are an excellent way to set performance goals.

Running a system near its limits can have negative consequences. In such a case, 10 percent more workload might degrade response times dramatically more than an expected 10 percent. In this situation, you must determine which component or process is the bottleneck and eliminate it.

After a system is properly tuned, improving system performance can be achieved only by reducing the workload or adding appropriate resources. You might have to revise your goals and expectations. For significant performance improvements, you must find the bottleneck and then consider one or more of the following actions:

- Use faster processors
- Add processors
- Add system memory
- Use faster communication links

  Consider a dedicated backup network, if client backup operations use a LAN that is shared with many other types of operations.
- Add disk storage
- Create a new server on a different system

Symptoms and causes of performance problems

When the performance of a Tivoli Storage Manager environment is less than expected, there can be one or more causes. Identifying the bottleneck in your environment can explain the performance degradation.

The following symptoms can indicate poor Tivoli Storage Manager performance:

- Processes or client backups take more time to run than normal
- Slow response times occur for commands that are issued
- Slow response times occur, and the system or process might seem hung
- Unexpected changes occur in response times or resource usage
- Throughput on the system is not as expected

- Processor usage is higher than normal for a certain process
- Network problems occur that are related to load, firewall, or routers

Performance problems can occur when changes are made in the environment. For example, changes to any of the following items can affect performance:

- Hardware configuration: Adding, removing, or changing configurations such as how disks are connected
- Operating system: Installing or updating a file set, installing fix packs, and changing parameters
- Applications: Installing new versions and fixes, configuring or changing data placement, or installing or upgrading device drivers or firmware
- Network: Any changes to the network, packet loss, or intermittent connectivity
- Disk units that are aging, or are damaged
- Options that are used to tune the operating system or an application
- Scheduling of processes or backups during times of high usage
- Unexpected increase in usage of a shared resource like the network or disks

You can collect data on the Tivoli Storage Manager server, client, or both at the same time to help diagnose where the problem is occurring in the environment and what the problem is.

Potential bottlenecks in the data flow for Tivoli Storage Manager operations

In operations such as client backup and storage pool migration, data moves through many physical components that can affect the speed of the operations. Understanding the characteristics of these components can help when you are working to improve performance.

Data flow for client backup operations over a LAN

Figure 1 shows data flow in a typical configuration for client backup operations over a local area network (LAN). For a client backup operation, the data flow starts at the client disk (item 1 in the graphic and table) and ends at one of the devices for the server storage pools (item 10 or 12).

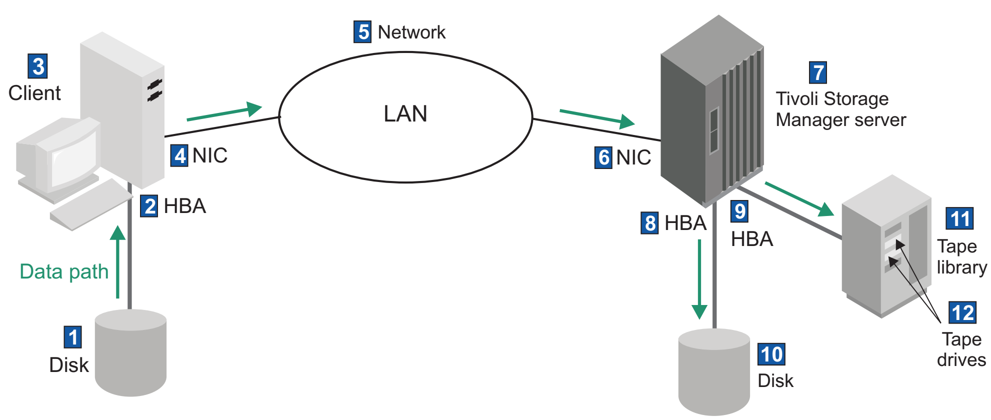

Figure 1. Data flow for client backup operations over a LAN

The data for backup operations flows through many hardware components, any of which are potential bottlenecks. The following table describes characteristics of the hardware that can affect performance.

Item in Figure 1 on page 49	Item	Key characteristics that can affect performance	Details
1	Disk for the client system	Disk type and rotational speed	
2	Host bus adapter (HBA) that connects the disk to the client system	HBA type and its capabilities	
3	Client system	The speed of the I/O bus, the number of processors, the speed of the processors, and the amount and speed of RAM	Use of data compression, data deduplication, and encryption, including Secure Sockets Layer (SSL) protocol, can affect processor performance on the client system. If processor usage is too high on the system, consider adding more processors, or turning off the options that enable compression, data deduplication, or encryption. Then, see whether performance improves. For information about tuning for client memory limits, see "Reduce client memory use" on page 196. Software such as firewalls and antivirus programs might affect the efficiency of client operations. For example, during a restore operation, an antivirus program might scan the contents of each restored object, checking for virus signatures. If you suspect that a firewall or antivirus program is slowing client operations, consider temporarily turning the firewall or antivirus program off to see whether performance improves. For tips to minimize the impact of firewall or antivirus programs on other applications, see the documentation for those programs.

Item in Figure 1 on page 49	Item	Key characteristics that can affect performance	Details
4	Network interface card (NIC) that connects the client system to the LAN	NIC type and its capabilities	A fast network interface card (NIC) improves network throughput. If you cannot use the latest NIC in your configuration, consider adjusting the client **TCPWINDOWSIZE** option to improve network throughput, particularly on client systems that are geographically distant from the server. Adjust the **TCPWINDOWSIZE** options in small increments; a window size that is larger than the buffer space on the network interface adapter might actually degrade throughput. For more network considerations, see Chapter 14, "Tuning network performance," on page 231.
5	Network	The many components on a network, and the effective speed of data transfer over the network, which is limited by its slowest component	
6	NIC that connects the server to the LAN	NIC type and its capabilities	
7	Server system	The speed of the I/O bus, the number of processors, the speed of the processors, and the amount and speed of RAM	
8	HBA that connects the server to the disk	HBA type and its capabilities	See "Tuning HBA capacity" on page 152.
9	HBA that connects the server to the tape library	HBA type and its capabilities	
10	Disk for the server storage pool	Disk type and rotational speed	
11	Tape library for the server storage pool	Number of drives and mount point availability for the operation	
12	Tape drive for the server storage pool	Tape type and sustainable speed	

Data flow for client backup operations over SAN

Figure 2 on page 52 shows data flow in a typical configuration for client backup operations over a storage area network (SAN). Metadata for the backup operation flows over the LAN. For a client backup operation, the data flow starts at the client disk (item 1 in the graphic and table) and ends at one of the devices for the server storage pools (item 11 or 13).

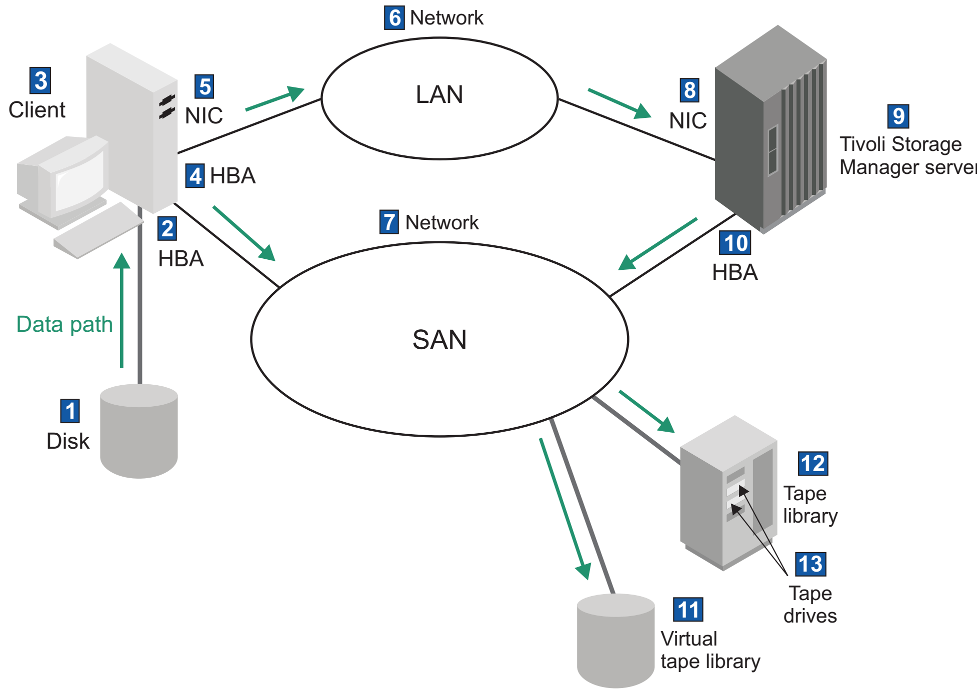

Figure 2. Data flow for client backup operations over the SAN

The data for backup operations flows through many hardware components, any of which are potential bottlenecks. The following table describes characteristics of the hardware that can affect performance.

Item in Figure 2	Item	Key characteristics that can affect performance	Details
1	Disk for the client system	Disk type and rotational speed	
2	Host bus adapter (HBA) that connects the disk to the client system	HBA type and its capabilities	

Item in Figure 2 on page 52	Item	Key characteristics that can affect performance	Details
3	Client system	The speed of the I/O bus, the number of processors, the speed of the processors, and the amount and speed of RAM	Use of data compression, data deduplication, and encryption, including Secure Sockets Layer (SSL) protocol, can affect processor performance on the client system. If processor usage is too high on the system, consider adding more processors, or turning off the options that enable compression, data deduplication, or encryption. Then, see whether performance improves. For information about tuning for client memory limits, see "Reduce client memory use" on page 196. Software such as firewalls and antivirus programs might affect the efficiency of client operations. For example, during a restore operation, an antivirus program might scan the contents of each restored object, checking for virus signatures. If you suspect that a firewall or antivirus program is slowing client operations, consider temporarily turning the firewall or antivirus program off to see whether performance improves. For tips to minimize the impact of firewall or antivirus programs on other applications, see the documentation for those programs.
4	HBA that connects the client system to the SAN	HBA type and its capabilities	
5	Network interface card (NIC) that connects the client system to the LAN	NIC type and its capabilities	A fast network interface card (NIC) improves network throughput. If you cannot use the latest NIC in your configuration, consider adjusting the client **TCPWINDOWSIZE** option to improve network throughput, particularly on client systems that are geographically distant from the server. Adjust the **TCPWINDOWSIZE** options in small increments; a window size that is larger than the buffer space on the network interface adapter might actually degrade throughput. For more network considerations, see Chapter 14, "Tuning network performance," on page 231.
6	Network: LAN	The many components on a network, and the effective speed of data transfer over the network, which is limited by its slowest component	

Item in Figure 2 on page 52	Item	Key characteristics that can affect performance	Details
7	Network: SAN	The many components on a network, and the effective speed of data transfer over the network, which is limited by its slowest component	
8	NIC that connects the server to the LAN	NIC type and its capabilities	
9	Server system	The speed of the I/O bus, the number of processors, the speed of the processors, and the amount and speed of RAM	
10	HBA that connects the server to the SAN	HBA type and its capabilities	See "Tuning HBA capacity" on page 152.
11	Virtual tape library (VTL) for the server storage pool	VTL model characteristics that affect the performance of the operation	
12	Tape library for the server storage pool	Number of drives and mount point availability for the operation	
13	Tape drive for the server storage pool	Tape type and sustainable speed	

Server storage data flow

Figure 3 shows data flow in the storage backplane in a server system. The data flow might be for an operation such as migration of storage pool data from disk storage pools to other storage pools. For a migration operation, the data flow starts at the source storage pool (item 1 in the graphic and table) and ends at the device for the target storage pool (item 6 or 8).

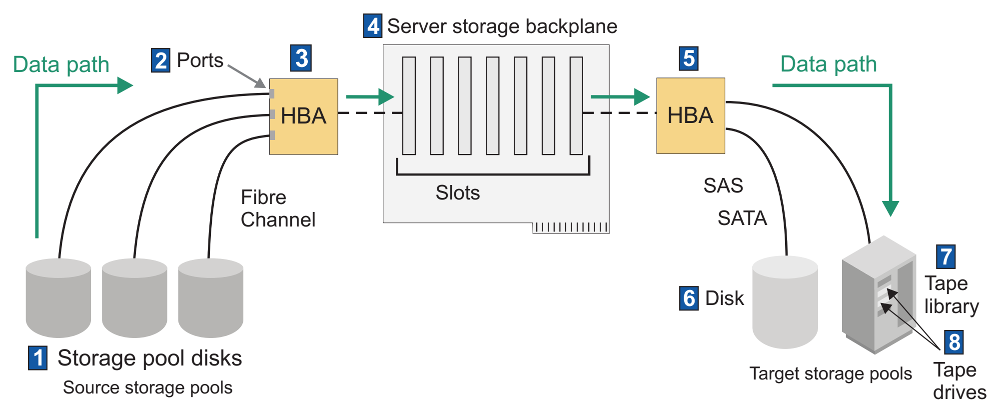

Figure 3. Data flow through the server storage backplane

The following table describes characteristics of the hardware that can affect performance of the operation.

Item in Figure 3 on page 54	Item	Key characteristics that can affect performance
1	Disks for source storage pools	Disk type and rotational speed
2	Ports	Multiple connection points to devices
3	HBA	These devices can have multiple ports. The total amount of data that is being transferred by disks at one time cannot exceed the overall throughput of the HBA.
4	Server storage backplane	The total of the speeds of all cards that are attached to the backplane cannot exceed the speed of the bus.
5	HBA	These devices can have multiple ports. The total amount of data that is being transferred by disks at one time cannot exceed the overall throughput of the HBA.
6	Disks for target storage pools	Disk type and rotational speed
7	Tape library for target storage pools	Number of drives and mount point availability for the operation
8	Tape drives for target storage pools	Tape type and sustainable speed

Related concepts:

"Reduce client data flow with compression" on page 198

Related tasks:

"Analyzing data flow with the **dd** command" on page 73

Workloads for the server

The ability of a server to handle workload is directly related to the server's resources, including system processors, memory, and I/O bandwidth. The ability of a server to efficiently process daily tasks determines how large a server can be.

Because any system has finite resources, the maximum workload on the server is limited by the recovery objectives. For example, if you decrease the frequency of server database backups to lessen the workload, you increase the time between recovery points for the system. Less frequent server database backups might cause the system to miss the recovery point objectives (RPO).

Ensure that a server can complete essential tasks in a 24-hour cycle:

- Complete the client workload.

  The client workload is the amount of data that is backed up or archived during the backup window. The backup window is typically a span of time at night or in the early morning. The ability of the server to store this data within the backup window can be limited by various factors:
  - Server storage capacity
  - I/O throughput to the storage devices
  - Network bandwidth
  - Other system attributes such as available memory or processors for the server
  - Characteristics of the client systems that are being backed up, including the following characteristics:
    - Processor speeds and memory for the client systems

- Disk speeds on client systems
- The total amount of data from all clients
- The total number of clients that request services from the server at one time

- Complete the essential server maintenance operations.

  Daily completion of the following operations keeps the server environment in good working condition and helps you prepare for disaster recovery of the server. These operations are key to effective data maintenance and management:
  - Expiration
  - Database backup
  - Reclamation

  Additional daily operations are required depending on the configuration of the solution and the features that are used:
  - Storage pool migration
  - Storage pool backup
  - Duplicate identification processes
  - Node replication processes

For examples of how a solution can be configured to handle workloads, see the sample architecture documents in the Tivoli Storage Manager wiki on Service Management Connect at http://www.ibm.com/developerworks/community/wikis/home/wiki/Tivoli Storage Manager/page/Sample Architectures.

Limits for the server database size and peak client sessions

IBM tests the Tivoli Storage Manager server to a specific database size and peak number of concurrent client sessions. However, you must consider the tested values along with other operational factors in your specific environment. Experiences that are reported by other users are also helpful.

Database size

Testing shows that databases with utilization as much as 4 TB are possible.

The practical limit for the database size depends on the performance characteristics of the server system and the time that is required to back up or restore the database. For many users, a 1 - 2 TB server database allows completion of database backup and database restore operations in a time that fits their maintenance window.

Consider deploying another server if the following conditions occur:

- Performance degrades to an unacceptable level as the database grows
- The time that is required to complete server maintenance such as database backup exceeds the total window of time for server maintenance

When you add a server, either balance the existing workload across the servers, or assign any new workload to the new server.

Peak number of client sessions

Testing shows that the Tivoli Storage Manager server can handle up to 1000 concurrent client sessions. If this value is exceeded, depending on memory or other system limitations, the server performance might degrade or operations might become unresponsive.

The actual number of concurrent sessions where performance problems arise depends on the resources that are available to the server. What the sessions are doing also influences the practical limit on sessions. For example, sessions that move data have a larger effect on the amount of I/O to the target storage pool compared to incremental backup sessions that mostly send queries without sending many files. Also, sessions that perform client-side deduplication drive more I/O to the server database than other sessions.

To reduce the peak-session workload, it might be appropriate to deploy another server or to adjust client scheduling.

Set the **MAXSESSIONS** server option no higher than the tested limit of 1000. Setting the maximum number of sessions higher than necessary uses extra RAM on the server system, but might not have a significant impact.

Related reference:

"Running concurrent client sessions" on page 208

Samples of data protection solutions

Samples of data protection solutions that use Tivoli Storage Manager for selected scenarios are available on the Service Management Connect wiki. The samples describe specific hardware and software configurations and give performance measurements that were obtained in IBM test labs.

The samples can help you plan new deployments or evaluate a current deployment of Tivoli Storage Manager. For more information, see IBM Tivoli Storage Manager Sample Architectures.

Related concepts:

"Resolving common client performance problems" on page 191

Related tasks:

"Selecting the optimal client backup method" on page 173

Chapter 7. Taking the first steps for solving performance problems

Start by looking for problems that are indicated by system errors or obvious deficiencies outside of Tivoli Storage Manager. After you are satisfied that no such problems exist, continue by reviewing information about identifying performance problems in the Tivoli Storage Manager environment. Throughout the process, monitor for errors and changes in performance.

Procedure

1. Review the best practices that are described in Part 2, "Configuration best practices," on page 9. Make changes as needed, based on the information.
2. Verify that there are no problems or errors outside of the server software. Problems in the server hardware, the operating system, the network, and attached storage devices can severely affect operations. Fix all errors outside of the server software before you diagnose server performance issues.
   a. Review error logs for the operating system to find errors that can affect the server. For example, for AIX systems use the **errpt** command to view errors. For Linux systems, look in the `/var/log` path.
   b. Verify that attached storage devices such as disk systems are operational and have no errors.
   c. Verify that storage and local area networks do not have frequent port errors.
3. Review the server activity log and client error logs.
4. Review the log for the server database, the `db2diag.log` file. To find the file, see Locating DB2 diagnostic log files.

What to do next

Implement modifications and fix any problems as described in the preceding steps, and then determine whether you must further analyze performance. Use techniques that are described in Chapter 8, "Identifying performance bottlenecks," on page 61 to analyze your system for performance bottlenecks.

Chapter 8. Identifying performance bottlenecks

When you suspect that you have a performance problem or want to improve performance, you can use the provided flowcharts to help identify sources of performance issues. The charts suggest the use of tools and scripts to help you measure performance on your system.

Procedure

- For client problems, or if you are not sure what performance issue you have, start at "Diagnosing backup and restore performance" on page 62.
- For server problems, start at "Identifying server performance problems" on page 64.

Diagnosing backup and restore performance

Use the flowchart to diagnose issues with backup and restore performance. The table provides more information and links to diagnostic tasks.

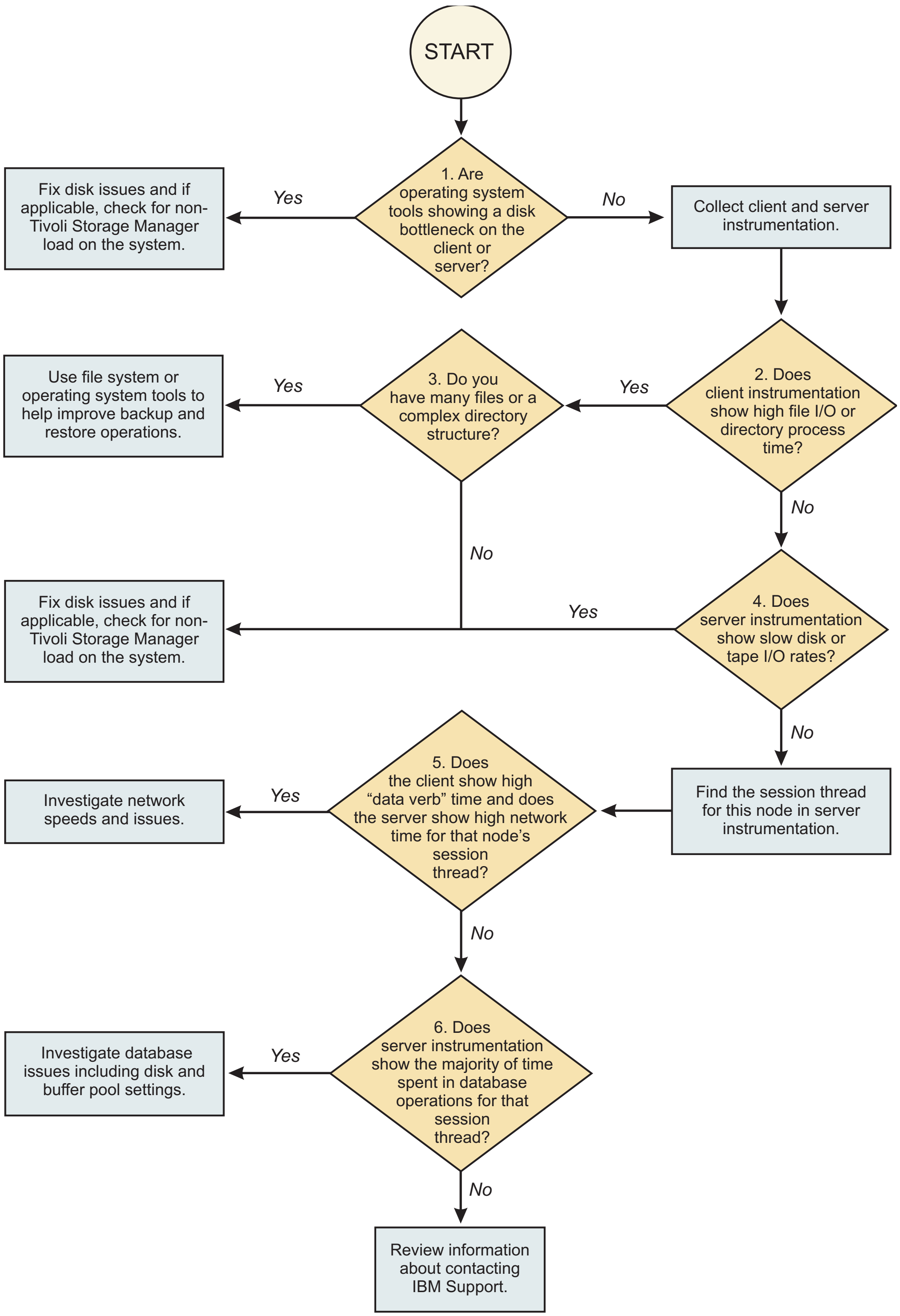

Figure 4. Flowchart to diagnose backup and restore performance. Use this chart with Table 3 on page 63.

Table 3. Explanation of the flowchart decisions and tasks for backup and restore performance

Step	Question	Diagnostic tasks
1	Do operating system tools show a disk bottleneck on the client or server? For more information, see "Identifying disk bottlenecks for Tivoli Storage Manager servers" on page 68.	**Yes** Fix any disk issues. If applicable, check the load on the system for applications that are not related to Tivoli Storage Manager. See Chapter 12, "Tuning disk storage for the server," on page 159. **No** Collect client and server instrumentation. See: • "Client instrumentation report" on page 90 • "Server monitoring script" on page 80 Go to question 2.
2	Does client instrumentation show high file I/O or directory process time?	**Yes** Go to question 3. **No** Go to question 4.
3	Do you have many files or a complex directory structure?	**Yes** Use file system or operating system tools to help improve backup and restore operations. See "File space tuning" on page 221. **No** Fix any disk issues. If applicable, check the load on the system for applications that are not related to Tivoli Storage Manager. See Chapter 12, "Tuning disk storage for the server," on page 159.
4	Does server instrumentation show slow disk or tape I/O rates?	**Yes** Fix any disk issues. If applicable, check the load on the system for applications that are not related to Tivoli Storage Manager. See Chapter 12, "Tuning disk storage for the server," on page 159. **No** Find the session thread for this node in server instrumentation. The thread might include the node name or you can see if the session thread ID is listed in the Activity Log to determine which node is affected. Go to question 5.
5	Does the client show high `Data Verb` time and does the server show high network time for that node's session thread?	**Yes** Investigate network speeds and issues and fix any problems. See Chapter 14, "Tuning network performance," on page 231. **No** Go to question 6.
6	Does server instrumentation show that most time is spent in database operations for that session thread?	**Yes** Investigate database issues, including disk and buffer pool settings. Fix any problems. See "Identifying server performance problems" on page 64. **No** IBM support personnel can help in diagnosing performance problems by requesting certain traces and other information from the environment. The comprehensive analysis of a performance problem is a billable service that is offered to Tivoli Storage Manager customers. For information about IBM Support and gathering data for problems, see the Software Support Handbook See also Chapter 9, "Collecting and analyzing data for performance problems," on page 75.

Related concepts:

"Potential bottlenecks in the data flow for Tivoli Storage Manager operations" on page 49

Identifying server performance problems

Use the flowchart to diagnose issues with server operations. The table after the flowchart provides more information and links to diagnostic tasks and tools.

Tip: Before you review the flowchart, ensure that you answer all questions and fix any issues that are described in the "Checklist for the server hardware and the operating system" on page 12 and the "Checklist for Tivoli Storage Manager server configuration" on page 27.

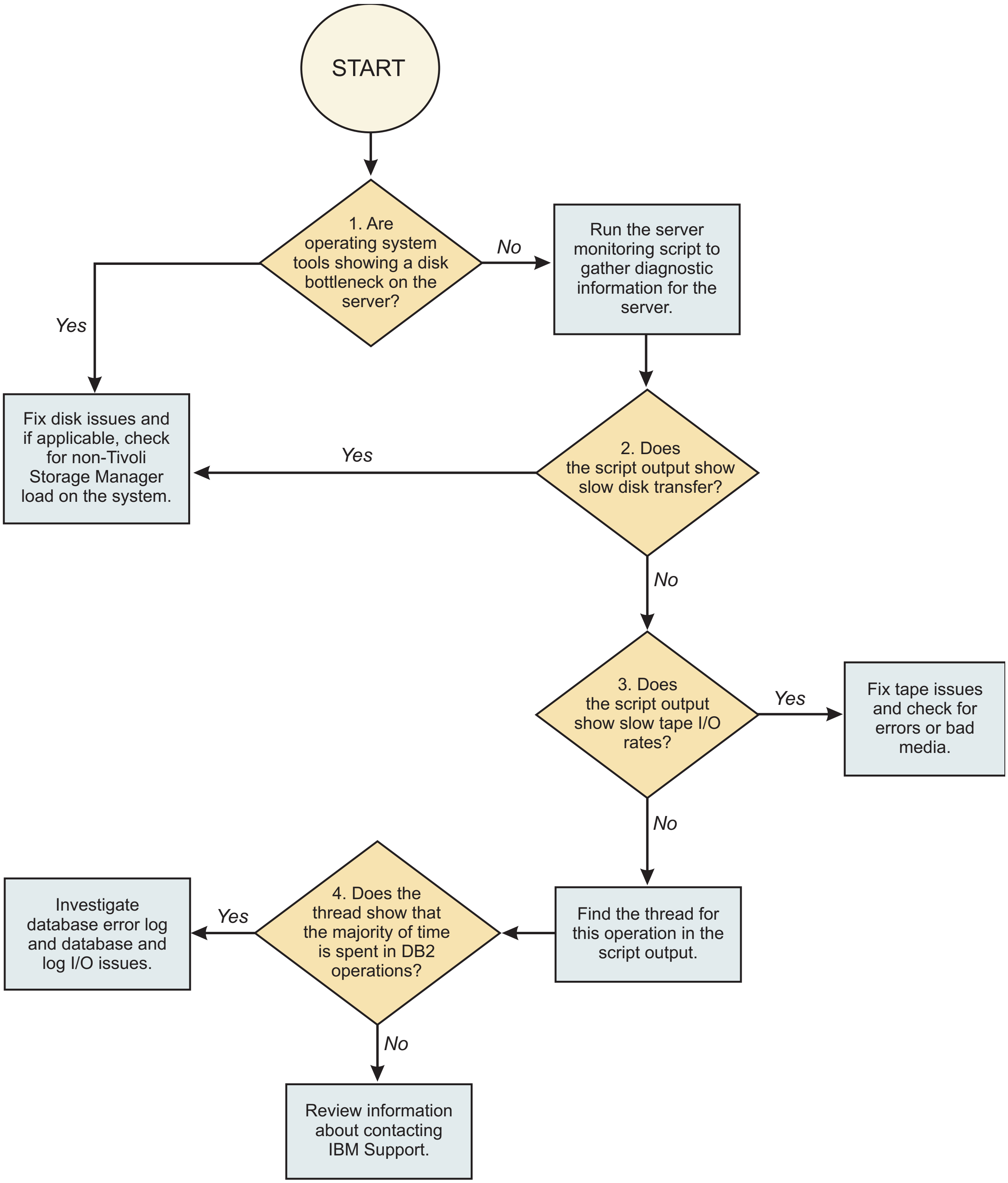

Figure 5. Flowchart to solve issues with server performance

Table 4. Flowchart questions and tasks to help you diagnose and fix server performance problems

Step	Question	Diagnostic tasks
1	Do operating system tools show a disk bottleneck on the server? For more information, see "Identifying disk bottlenecks for Tivoli Storage Manager servers" on page 68.	**Yes** Fix any disk issues. If applicable, check the load on the system for applications that are not related to Tivoli Storage Manager. See Chapter 12, "Tuning disk storage for the server," on page 159. **No** Run the server monitoring script to gather diagnostic information for the server. See "Analyzing disk performance by using the server monitoring script" on page 69. Go to question 2.
2	Does the script output show slow disk transfer?	**Yes** Fix any disk issues. If applicable, check the load on the system for applications that are not related to Tivoli Storage Manager. See "Analyzing the basic performance of disk systems" on page 73. **No** Go to question 3.
3	Does the script output show slow tape I/O rates?	**Yes** Fix tape issues and check for errors or bad media, if applicable. See "Tuning tape drive performance" on page 150. **No** Find the thread for the operation in the script output. Go to question 4.
4	Does the thread show that most of time is spent in DB2 operations?	**Yes** Investigate database error log and database and log I/O issues. See "Server database and recovery log configuration and tuning" on page 121. **No** IBM support personnel can help in diagnosing performance problems by requesting certain traces and other information from the environment. The comprehensive analysis of a performance problem is a billable service that is offered to Tivoli Storage Manager customers. For information about IBM Support and gathering data for problems, see the Software Support Handbook See also Chapter 9, "Collecting and analyzing data for performance problems," on page 75.

Related concepts:

"Potential bottlenecks in the data flow for Tivoli Storage Manager operations" on page 49

Related tasks:

"Reporting performance problems" on page 77

Evaluating data deduplication results

You can evaluate the effectiveness of Tivoli Storage Manager data deduplication by examining various queries or reports. Actual data reduction results can show whether the expected storage savings are achieved. You can also evaluate other key operational factors, such as database utilization, to ensure that they are consistent with expectations.

Before you begin

Consider the following factors when you are evaluating data deduplication results:

- When you are using data deduplication, you might not see immediate space savings on the server.
- Because data deduplication includes multiple backup operations across multiple clients, processing will become more effective over time.

Therefore, it is important to collect results at regular intervals to record valid results.

Procedure

Use the following commands and tools to help you evaluate data deduplication effectiveness:

Action	Explanation

Use the **QUERY STGPOOL** server command to quickly check deduplication results.	The `Duplicate Data Not Stored` field shows the actual reduction of data, in megabytes or gigabytes, and the percentage of reduction of the storage pool. For example, issue the following command:
	`query stgpool format=detailed`
	If the query is run before reclamation of the storage pool, the `Duplicate Data Not Stored` value is not accurate because it does not reflect the most recent data reduction. If reclamation did not yet take place, issue the following command to show the amount of data to be removed:
	`show deduppending backkuppool-file`
	Where `backkuppool-file` is the name of the deduplicated storage pool.

Use the **QUERY OCCUPANCY** server command.	This command shows the logical amount of storage per file space when a file space is backed up to a deduplicated storage pool.
Examine the Tivoli Storage Manager client backup reports to see the data reduction for a backup operation that is run with client-side data deduplication and compression.	The backup reports are available upon the completion of backup operations. Over time, if the backup reports repeatedly show little to no data reduction after many backups, consider redirecting the client node to a non-deduplication storage pool if one is available. This way, the client is not wasting time by processing data that are not good candidates for data deduplication.

Action	Explanation
Run the deduplication report script to show information about the effectiveness of data deduplication.	The report provides details of deduplication-related utilization of the Tivoli Storage Manager database. You can also use it to gather diagnostic information when the deduplication results are not consistent with your expectations. To obtain the script and usage instructions for the script, see technote 1596944.

What to do next

For more information, see Effective Planning and Use of IBM Tivoli Storage Manager V6 and V7 Deduplication.

Related concepts:

"Checklist for data deduplication" on page 30

Related tasks:

"Tuning server-side data deduplication" on page 147

"Tuning client-side data deduplication" on page 200

Identifying disk bottlenecks for Tivoli Storage Manager servers

Tools can help you identify where disk storage that is used for Tivoli Storage Manager servers might have bottlenecks.

Before you begin

Before you begin this task, review information about optimal disk storage configuration for the server database, recovery logs, and storage pools.

Procedure

To identify disk bottlenecks, you can use one or both of the following methods:

- Use the server monitoring script. The `servermonV6.pl` script collects monitoring data that can help you evaluate Tivoli Storage Manager server activities (performance, processes, sessions). The script runs server instrumentation and in some cases system tools.

  See "Analyzing disk performance by using the server monitoring script" on page 69.
- Use analysis tools that are provided by third parties. Such tools can be effective to analyze storage systems for basic performance characteristics before they are used for Tivoli Storage Manager storage.

  See "Analyzing disk performance by using system tools" on page 71.

Related concepts:

"Potential bottlenecks in the data flow for Tivoli Storage Manager operations" on page 49

Related reference:

"Checklist for server database disks" on page 17

"Checklist for server recovery log disks" on page 19

"Checklist for storage pools on DISK or FILE" on page 25

"Server instrumentation for performance analysis" on page 80

Analyzing disk performance by using the server monitoring script

Use the Perl script, `servermonV6.pl`, to collect monitoring data that is needed to evaluate Tivoli Storage Manager server activities (performance, processes, and sessions). You can use the script to collect performance statistics about the disk storage that is used by DB2, the database manager for Tivoli Storage Manager.

Procedure

1. Download the latest copy of the Perl script from technote 1432937. Review the details in the technote carefully. For example, some specific versions or fix packs for the server might have a problem with certain parts of the script.
2. Download the script into a new directory.
   - For convenience, place the script in the directory where you plan to run it. The output is created where you run the script.
   - The output from the script requires some storage space. Plan for at least 100 MB for the directory.
3. Log on with the server instance user ID and run the script. Respond to the prompts from the script. You must provide an administrator ID for the server.

   Windows servers: For a Windows server, you must open a DB2 command window. For example, click **Start** > **Run** and enter `db2cmd`. From the command window that opens, change to the script directory and run the script from there.
4. Allow the script to run long enough to collect data that is related to operations that you are concerned about. If you are looking for a particular problem, run the script for the expected duration of the problem. If a specific problem is not yet identified, run the script for 24 hours to get a view of what is happening in the system.

   The script runs until you stop it.
5. Stop the script by pressing Ctrl+C.
6. To learn about the performance of the database and recovery log disks, find the instrumentation file in the output. The file has a name like *timestamp*`-instr.txt`, where *timestamp* is the date and time. View the section of the report that has the label, `DB2 STATISTICS DURING INSTRUMENTATION INTERVAL`. The following example shows part of a report.

```
Deadlocks detected: 0 --> 0.0/sec
Number of lock escalations: 0 --> 0.0/sec
Lock waits: 0 --> 0.0/sec
Time waited on locks(*): 0.000 sec
Locks held: 3 before, 3 after
Intern Rollbacks Due To Dlock: 0 --> 0.0/sec
Total sorts: 1108 --> 0.9/sec, 0.001 sec/sort
Total sort time(*): 967 --> 0.8/sec
Sort overflows: 1 --> 0.0/sec
Direct reads from database: 19740 --> 16.2/sec, 0.000 sec/read
Direct read time: 0.154
Direct writes to database: 31166 --> 25.6/sec, 0.000 sec/write
Direct write time: 0.221
Number of Log Pages Written: 2011 --> 1.7/sec, 0.0001 sec latency
Log Write Time: 0.217 sec
Number of Log Writes: 898 --> 0.7/sec
```

The following table describes key items to examine in the report. Over time, repeat running the script and reviewing the reports to see whether performance is consistent or varies.

Report item	Description	Values to look for
`Direct reads from database`	The number of read operations that did not use the buffer pool. The report also shows average time for read to the database during the interval of the server instrumentation.	Typically 4 ms is a good value for average time for reads. A value of 10 ms or greater indicates a possible problem. • Verify that the disk system is running and has no errors. • Check whether other applications or systems are using the disk system and causing a conflict with Tivoli Storage Manager operations. If problems or a conflict are not the cause, you might need faster disks to improve the performance of Tivoli Storage Manager operations.
`Direct writes to database`	The number of write operations that did not use the buffer pool. The report also shows average time for write operations to the database during the interval of the server instrumentation.	Typically 2 ms is a good value for average write time. An average write time that is greater than 5 ms indicates a problem.
`Number of Log Pages Written`	The number of log pages that are written to disk during the interval of the server instrumentation. The report also shows the *average latency*, which is the average time delay for each write operation to the log.	Typically 2 ms is a good value for the latency for writes to the log. If log write latency is more than 5 ms, the report includes a warning message.

7. For servers that are running on AIX, HP-UX, Linux, or Oracle Solaris systems, find the file that has the **iostat** command output. The file has a name like *timestamp*-iostat.txt, where *timestamp* is the date and time.
   a. Look at how busy the disks are. If you see that the disks are more than 80% busy, then you have a potential disk bottleneck. For AIX systems, the column is labeled `% tm_act`. For Linux systems, the column is labeled `%util`.
   b. Look for the column that indicates what queues are full. (For AIX systems, the column is labeled `serv qfull`.) The count in the queues-full column indicates the number of times per second that the service queue becomes full. When the service queue becomes full, the disk is not accepting any more service requests. The value represents a measurement since the last time iteration. Consistently high numbers are a sign of a problem. The queue depth might be too low; disk systems from manufacturers other than IBM often have a default value of 1.

Examine the queue depth for the disks. If the queue depth is less than 32, consult the documentation for the disk system or contact the manufacturer about guidelines for setting the queue depth.

For an LPAR environment, also examine the queue depth for the Virtual I/O Servers (VIOS) on the AIX host system. You might have to adjust the queue depth for not only the Tivoli Storage Manager LPAR, but also the VIOS.

c. Look at the column for transfers per second that were issued to the physical disk. (For AIX systems, the column is labeled `tps`.) This column is the indicator of I/O operations per second (IOPS).

   Also, look at the average service time. (For AIX systems, the column is labeled `avg serv`.) High averages of read/write or service times can indicate a problem. In general, for Tivoli Storage Manager, aim for service values that are less than 5 ms for log and database read/write operations.

Analyzing disk performance by using system tools

You can use system tools to monitor I/O for disk storage to help you identify disk bottlenecks. For example, use tools such as **nmon** for AIX and Linux operating systems, and the Performance Monitor for Windows systems.

About this task

Other tools can be used but are not documented here. For example, for operating systems such as AIX and Linux, you can use the **sar** command to collect information about system activity.

Related tasks:

"Analyzing data flow with the **dd** command" on page 73

"Server monitoring script" on page 80

Related reference:

"Server instrumentation for performance analysis" on page 80

Analyzing disk performance by using the nmon command

You can use the **nmon** command on AIX and Linux operating systems. Use the command to display local system statistics in interactive mode and to record system statistics in recording mode.

Procedure

1. Run the command as root. You can run the command from any directory, but you must be logged in as the root user. The command starts a background process to capture the specified number of snapshots of system statistics at the specified intervals, and writes the output file.

   Important: Do not use the **kill** command to end the process because that causes the file to be incomplete and unusable for analysis.

   To run the command, use the following parameters:

   -f
   Specifies that the output is written to a file. The file is created in the directory where you run the command, with the name *hostname_YYMMDD_HHMM*.nmon.

   -s nnnn
   Specifies the seconds between snapshots of statistics.

-c nnn
Specifies the number of snapshots.

Typically to create a report for performance analysis, you specify 15 minutes between snapshots (900 sec) over 24 hours (96 snapshots). For example, issue the following command:

```
nmon -f  -s 900 -c 96
```

To collect a snapshot of the system every hour for seven days, issue the following command:

```
/home/hm12857/netmon/nmon -f -s 3600 -c 168
```

To collect a snapshot of the system every 30 minutes for five days, issue the following command:

```
/home/hm12857/netmon/nmon -f -s 1800 -c 240
```

2. Analyze the data by using the **nmon** Analyzer spreadsheet tool. Focus on the disk-busy statistics (`Disk %Busy`). Look for disks that are consistently over 80% busy (weighted average). The weighted average is shown in red in the chart on the `diskbusy` tab.

Analyzing disk performance by using Windows Performance Monitor (perfmon)

Examine disk statistics by using performance counters that are available in the Performance Monitor.

Procedure

1. Start the Performance Monitor. From a command prompt, enter: `perfmon`.
2. Create a Data Collector Set to collect data about disks. Select the following performance counters from the **Physical Disk** category:
   - **Avg. Disk Sec./Transfer**
   - **Avg. Disk Queue Length**
   - **Avg Disk Bytes/Transfer**
   - **Disk Bytes/sec**
   - **Split IO/sec**
3. Run the **perfmon** tool while you are experiencing the performance problem. Compare the results with the guidance in the following table.

Performance counter	Guidance
Physical Disk: Avg. Disk Sec./Transfer	Values less than 25 ms are good.
Physical Disk: Avg Disk Queue Length	A value that is 2 or 3 times the number of disks in the array is optimal.
Physical Disk: Avg Disk Bytes/Transfer	The goal is that the stripe size for the array is at least the average of this counter.
Physical Disk: Disk Bytes/sec	Optimal result is that the sum of values for all disks that are attached to a single controller is less than 70% of the theoretical throughput.
Physical Disk: Split IO/sec	A nonzero value for this counter indicates possible disk fragmentation.

Analyzing the basic performance of disk systems

To verify that a storage system can meet the workload requirements for Tivoli Storage Manager operations, run stress tests. You might also want to analyze disk performance when changes are made to the host or the network backbone.

About this task

Various tools are available for analysis and stress tests for disk characteristics such as I/O operations per second (IOPS).

Procedure

- For AIX, you can use the **ndisk64** command. Search for the latest **nstress** package at AIX performance analysis and tuning.
- You can use third-party tools such as Iometer, which is available for Windows and other operating systems. For information about the Iometer tool, see http://www.iometer.org.
- For operating systems such as AIX and Linux, you can use the **dd** command for simple tests of capabilities.

Analyzing data flow with the dd command

You can use the **dd** command as a quick test to estimate the best possible results for data flow to disks. The command is available for operating systems such as AIX or Linux.

About this task

The **dd** command can be useful if you do not have or do not want to install more powerful tools. To estimate peak performance under ideal conditions, use the **dd** command to time how long a write to a device takes. Then, time how long a read from the device takes.

Procedure

1. To run a write test, issue the following command.

   ```
   time dd if=/dev/zero of=/device_path/filename bs=262144 count=40960
   ```

   where *device_path* is the name of the file system that you want to test, and *filename* is the name of a file.

   Important: The *filename* file must not exist in the file system. If the file does exist, the command overwrites it with zeros.

   The output of this command gives you the time that is required to write a 10 GB file in 256 KB blocks.
2. To run a read test of the file that was written, issue the following command.

   ```
   time dd if=/device_path/filename of=/dev/null bs=262144 count=40960
   ```

   When you evaluate the command results, keep in mind that if you just ran the write test that the data might still be in the disk cache. The time reported by the command for the read operation is therefore less than what you can expect for typical Tivoli Storage Manager server operations. For typical Tivoli Storage Manager server operations, data is not likely to be in the cache and is read from the disk itself.

Chapter 9. Collecting and analyzing data for performance problems

Capturing specific performance metrics, as the problem occurs in your environment, is essential to help IBM support personnel with the analysis.

Most performance problems appear as unacceptable response times or resource usage. Performance problems can slowly develop over time, as a result of dwindling resources, or suddenly as the result of a hardware or software change in the environment.

As part of the standard product support package, IBM will help to determine whether a performance problem is a result of a product defect. Gathering key performance metrics, from the customer environment will be a key part of this activity. The comprehensive analysis of a performance problem is a billable service that is offered to Tivoli Storage Manager customers. For more information, see the Software Support Handbook at http://www.ibm.com/support/customercare/sas/f/handbook/home.html.

Measuring the baseline

Performance problems are often reported immediately following some change to system hardware or software. Unless there is a pre-change baseline measurement with which to compare post-change performance, it can be difficult to qualify the problem.

About this task

Any change in the environment, including software, hardware, or network changes, can affect the performance of operations in your environment.

As a best practice, monitor the environment before and after each change. The alternative is to take the baseline measurements at regular intervals, for example, once a month, and save the output. When a problem is found, you can use the previous measurements for comparison. It is a good idea to collect a series of measurements, which can help you diagnose a possible performance problem.

To maximize performance diagnosis, collect data for various periods of the day, week, or month when performance is likely to be an issue. For example, you might have workload peaks during these times:

- In the middle of the mornings for online users
- During a late-night batch run
- During the end-of-month processing
- During major data loads

Collect data for each peak in workload because a performance problem might cause problems only during one of these periods and not during other times.

Restriction: Using any tool to collect the baseline data can impact the performance of the system that is being measured.

Procedure

To collect baseline data, you can use the following tools:

- On any operating system, you can use the Tivoli Storage Manager server monitoring script. For more information, see "Server monitoring script" on page 80.
- On AIX and Linux operating systems, you can use the `nmon` utility. For more information about this and other tools, see AIX performance analysis and tuning.
- On Windows operating systems, you can use the `perfmon` utility to gather a set of performance counters. For more information about the utility, see technote 7002840.

Periodically record your baseline measurements so that you can use the data for comparisons after an unexpected performance degradation. If you collect baseline data before a performance problem is detected, IBM Support can use the data to help you resolve performance problems.

Describing performance problems

Support personnel often receive information that is insufficient to accurately determine the nature of a performance problem. You must be able to describe the problem in as much detail as possible.

About this task

Always obtain as much detail as possible before you collect or analyze data, by asking the following questions about the performance problem:

- Can the problem be demonstrated by running a specific command or reconstructing a sequence of events? What is the least complex example of the problem.
- Is the slow performance intermittent? Does it get slow at certain times, but then returns to normal? Does it occur at certain times of the day or in relation to some specific activity?
- Is everything slow or only some things? What aspect is slow? For example, time to run a command, or elapsed time to complete a process, or time to paint the screen?
- When did the problem start occurring? Was the situation the same since the system was first installed or went into production? Did anything change on the system before the problem occurred (such as adding more users or migrating more data to the system)?
- If the issue is client and server, can the problem be demonstrated locally on the server (network versus server issue)?
- If network related, how are the network segments configured (including bandwidth such as 100 Mb/sec or 10 Mb/sec)? Are there any routers between the client and server?
- What vendor applications are running on the system, and are those applications involved in the performance issue?
- What is the impact of the performance problem on the users?

Reporting performance problems

Before you report a problem, you can collect information in advance to facilitate the problem investigation.

About this task

When you report a performance problem, it is not enough just to gather data and analyze it. Without knowing the nature of the performance problem, you might waste time and resources when you analyze data that might have nothing to do with the problem that is being reported.

Your local support personnel can use this information to help solve the performance problem with you.

For information about IBM Support and gathering data for problems, see the Software Support Handbook

Procedure

To help get your problem resolved more quickly, complete the following tasks:

1. Gather information about the performance issue to help you prepare a problem description:
   - For backup-archive client performance issues, run the client instrumentation and the server monitoring script simultaneously. See "Collecting instrumentation data with the client" on page 90 and "Server monitoring script" on page 80.
   - For server performance issues, run the server monitoring script. See "Server monitoring script" on page 80.
   - Gather detailed information about LUN layout, cache size and setting information, disk system information, file systems type, RAID type, and other setup details. Because many performance issues are I/O related, this information is important.
   - Collect a list of the hardware information such as host bus adapter type, processor type, and amount of RAM you have on the client and server.
   - Gather network and SAN zoning information.
2. Gather more information about your performance problem and your environment by completing the performance questionnaire at technote 1623261.
3. Provide a statement of a simple specific instance of the problem. Separate the symptoms and facts from the theories, ideas, and your own conclusions. Problem Management Records that report `the system is slow` statements can require extensive investigation to determine what is meant by slow, how it is measured, and what is acceptable performance.
4. Gather information about everything that changed on the system in the weeks before the problem. Missing something that changed can block a possible investigation path and might delay finding a resolution. If all the facts are available, IBM Support can eliminate the unrelated facts.

   Tip: Be sure that you collect the information from the correct system. In large sites, it is easy to accidentally collect the data on the wrong system, which makes it difficult to investigate the problem.
5. Provide the following information:

- A problem description that can be used to search the problem-history database to see whether a similar problem was reported.
- Describe the aspect of your analysis that led you to conclude that the problem is caused by a defect in the operating system.
- Describe the hardware and software configuration in which the problem is occurring:
  - Is the problem confined to a single system, or does it affect multiple systems?
  - What are the models, memory sizes, and number and size of disks on the affected systems?
  - What kinds of LAN and other communications media are connected to the systems?
  - Does the overall configuration include other operating systems?
- Describe the characteristics of the program or workload that is experiencing the problem.
  - Does an analysis with operating system tools indicate that it is processor-limited or I/O-limited?
  - What is the workload that is run on the affected systems?
- Describe the performance objectives that are not being met.
  - Is the primary objective console or terminal response time, throughput, or real-time responsiveness?
  - Were the objectives derived from measurements on another system? If so, what was its configuration?

6. If this report is the first report of the problem, you will receive a PMR number for use in identifying any additional data that you supply and for future reference. Include all of the following items when the supporting information and the performance data are gathered:
   - A means of reproducing the problem:
     - If possible, include a program or shell script that demonstrates the problem.
     - At a minimum, a detailed description of the conditions under which the problem occurs is needed.
   - The application that is experiencing the problem:
     - If the application is, or depends on, any software product, identify the exact version and release of that product.
     - If the source code of a user-written application cannot be released, document the exact set of compiler parameters that are used to create the executable program.

Collecting instrumentation data for the Tivoli Storage Manager server, client, and API

Tivoli Storage Manager instrumentation can collect data to help isolate performance bottlenecks in the Tivoli Storage Manager client, server, or the network.

Tivoli Storage Manager instrumentation is available for the Tivoli Storage Manager server, client, and API. It is intended to be used for performance tuning and problem determination. You can use the instrumentation as an alternative to collecting data from the traditional Tivoli Storage Manager **trace** command.

The following steps are a basic approach to working on performance bottlenecks:

1. Use Tivoli Storage Manager instrumentation data to determine which backup component (client, server, or network) accrues the most time during the Tivoli Storage Manager process.
2. After you isolate the component that takes the most time, try to determine whether the operation is restricted by a hardware or software resource.
3. Change how that resource is used, or augment it. For example, upgrade processors, or increase memory, disks, or tape drives.
4. Repeat this process as needed to reduce the bottleneck to an acceptable level.

Benefits of Tivoli Storage Manager instrumentation

There are benefits that are associated with using Tivoli Storage Manager instrumentation function over using the traditional Tivoli Storage Manager **trace** command.

The advantages of using the Tivoli Storage Manager instrumentation function are as follows:

- Tivoli Storage Manager traces potentially produce large trace files, frequently causing out-of-space conditions on file systems, and can cause significant performance degradation. With the Tivoli Storage Manager instrumentation function, no huge trace files are created, causing minimal performance impact.
- Tivoli Storage Manager instrumentation generates concise reports that collate and summarize significant performance information. The report files are typically small in size, often less than 1 MB, and are designed to have minimal impact on performance. The data is stored in memory until the instrumentation session ends.

How processes are tracked

Instrumentation tracks operations that can affect performance.

For example, the following operations are tracked:

- Disk I/O
- Network I/O
- Tape I/O

Each Tivoli Storage Manager process can have multiple threads. All threads can operate on different processors. The Tivoli Storage Manager server can have hundreds of active threads at a time. You can use the **show threads** command to see a snapshot of the active threads.

For example, a backup operation uses at least two threads. A **SessionThread** thread receives data from the client and sends it to an **SsAuxSinkThread** thread. When you back up data to a sequential device, the **AgentThread** thread moves the data from the **SsAuxSinkThread** thread and writes the data to tape. When you back up data to random disk on IBM AIX, Linux, and UNIX systems, a **DiskServerThread** writes the data to the device. When you back up data to disk on Microsoft Windows systems, the data moves directly to the random disk from the **SsAuxSinkThread** thread.

Tivoli Storage Manager instrumentation tracks processes in the following manner:

- Operations are tracked on a thread-by-thread basis
- Most sessions and processes use more than one thread

- Results are stored in memory until instrumentation is ended

Server instrumentation for performance analysis

You can use server instrumentation to track operations, such as backup and restore, and to help identify where performance problems originate.

The server monitoring script, `servermonV6.pl`, runs the server instrumentation commands to collect data. Typically, you can use the script instead of the server instrumentation commands by themselves. You can download the server monitoring script from http://www.ibm.com/support/docview.wss?uid=swg21432937.

Server monitoring script

Collecting a complete set of data for Tivoli Storage Manager server is essential for analyzing performance problems.

About this task

The server monitoring script is a Perl script that can be used to collect performance data during Tivoli Storage Manager server activities. This script is a standard tool that is used to help diagnose performance problems. After the script is enabled, the script logs on to the target Tivoli Storage Manager server. The script starts a server instrumentation trace. At set intervals, several **SHOW** commands, **query** commands, and DB2 commands are run to capture point-in-time records of information. On UNIX systems, operating system disk and processor information are also collected. A series of output files are created that can help the support team to diagnose performance problems.

Procedure

1. Obtain the server monitoring script. The Perl script can be downloaded at http://www.ibm.com/support/docview.wss?uid=swg21432937. The document contains all the information that you need including:
   - The variables that you must update to connect to your Tivoli Storage Manager server
   - Instructions on how to start the Perl script on your Tivoli Storage Manager server
   - List of commands that are run on the Tivoli Storage Manager server
2. Leave the script on to capture the performance problem and collect data. The length of time to leave the script running can vary depending on the performance problem. Generally, you can leave the script running for about an hour while it captures the performance problem. If you are running the client instrumentation at the same time, have the Perl script on until you turn off the client instrumentation.

What to do next

After the performance files are collected, you can report the problem to IBM support.

Related tasks:

"Reporting performance problems" on page 77

Related reference:

"Server instrumentation for performance analysis"

Server instrumentation categories

Tivoli Storage Manager server instrumentation can report on the elapsed times for the process categories that are documented in the table. Server instrumentation tracks all input and output on a thread-by-thread basis for the categories.

Table 5 lists the server instrumentation categories that are tracked and the activity that is timed.

Table 5. Server instrumentation categories

Category	Activity
Acquire Latch	The amount of time to acquire a database page from disk or buffer pool
Acquire XLatch	The amount of time to acquire a database page for update (from disk or buffer pool)
CRC Processing	The amount of time to compute or compare Cyclic Redundancy Check (CRC) values in storage pools
Data Copy	The amount of time to copy data to various buffers in memory
DB2 Commit	The amount of time to commit the DB2 transaction
DB2 Connect	The amount of time connected to DB2
DB2 CR Exec	The amount of time to run an SQL statement that counts rows
DB2 CR Prep	The amount of time to prepare an SQL statement that counts rows
DB2 Delet Exec	The amount of time for DB2 to run an SQL statement that deletes a row
DB2 Delet Prep	The amount of time for DB2 to parse an SQL statement that deletes a row
DB2 Fetch	The amount of time to prepare an SQL statement that retrieves one row from DB2
DB2 Fetch Exec	The amount of time for DB2 to run an SQL statement that returns one row
DB2 Fetch Prep	The amount of time for DB2 to prepare an SQL statement that returns one row
DB2 Inser Exec	The amount of time for DB2 to run an SQL statement that inserts a row
DB2 Inser Prep	The amount of time for DB2 to parse an SQL statement that inserts a row
DB2 MFetch	The amount of time to prepare an SQL statement that retrieves many rows from DB2
DB2 MFtch Exec	The amount of time for DB2 to run an SQL statement that returns many rows
DB2 MFtch Prep	The amount of time for DB2 to prepare an SQL statement that returns many rows
DB2 Reg Exec	The amount of time for DB2 to run complex SQL statements
DB2 Reg Fetch	The amount of time for DB2 to retrieve rows for a complex SQL statement

Table 5. Server instrumentation categories (continued)

Category	Activity
DB2 Reg Prep	The amount of time for DB2 to prepare complex SQL statements
DB2 Updat Exec	The amount of time for DB2 to run an SQL statement that updates a row
DB2 Updat Prep	The amount of time for DB2 to parse an SQL statement that updates a row
Disk Commit	The amount of time that it takes to run the **FSYNC** command or other system call to ensure that writes are completed to disk
Disk Read	The amount of time used to read from disk
Disk Write	The amount of time used to write to disk. You can combine this amount with the Disk Commit amount to get the total write time
Fingerprint	The amount of time used to find extent boundaries for data deduplication
ICC Digest	The amount of time that an algorithm takes for data deduplication extents
Namedpipe Recv	The amount of time to receive data on a named pipe
Namedpipe Send	The amount of time to send data on a named pipe
Network Recv	The amount of time to receive data on a network from a client
Network Send	The amount of time to send data on a network to a client
Shmem Copy	The amount of time to copy data to and from a shared memory segment
Shmem Read	The amount of time to read data from shared memory buffer
Shmem Write	The amount of time to write data to shared memory buffer
Tape Commit	The amount of time to synchronize tape, to ensure that data is written from device buffers to media
Tape Data Copy	The amount of time to copy data to tape buffers in memory
Tape Locate	The amount of time to locate a tape block for read/write operations
Tape Misc	The amount of time to process tape that is not tracked in another tape category (operations such as open or rewind)
Tape Read	The amount of time to read from tape
Tape Write	The amount of time to write to tape
Thread Wait	The amount of time to wait on some other thread
Tm Lock Wait	The amount of time to acquire the transaction manager lock

Table 5. Server instrumentation categories (continued)

Category	Activity
Uncompress	The amount of time used to uncompress data
Unknown	The amount of time for something that is not tracked by another category

Server threads in instrumentation output

The server program divides its operations into threads. In instrumentation output, the names of the threads identify the operations.

Only some of the threads in the instrumentation output are useful for diagnosing performance problems. The most important threads are for reclamation of storage pool volumes, migration of data from random-access storage pools, and backup of storage pools.

Reclamation of storage pool volumes

The main thread for a reclamation operation for a storage pool volume is called AfRclmVolumeThread. The main thread starts one or two child threads. Each child thread controls a thread that is called AgentThread. Data movement operations start with an AgentThread that reads an object from a volume that is being reclaimed. See Figure 6.

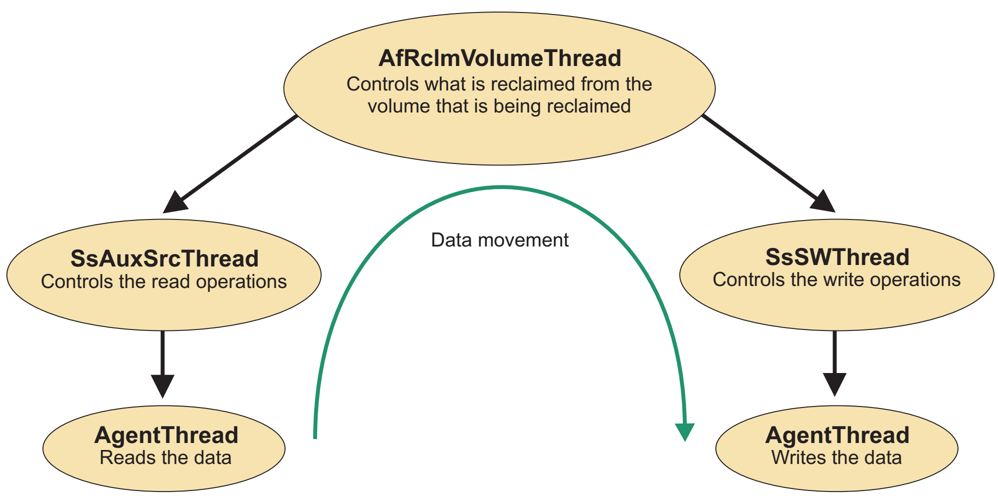

Figure 6. Threads for reclamation of storage pool volumes

A typical data movement operation starts with an AgentThread that reads an object from a volume that is being reclaimed. That data is processed through the SsAuxSrcThread, AfRclmVolumeThread, and SsSWThread threads. Data movement ends when the data is written on the target volume by the AgentThread thread that writes the data.

Migration of data from random-access storage pools

The main thread for a migration operation for a random-access storage pool is DfMigrationThread. The child threads to complete the migration operation differ by operating system.

AIX, Linux, HP-UX, and Solaris

The main thread, DfMigrationThread, does the work of selecting the data for migration and the volumes that are read from and written to. The thread starts two child threads: SsAuxSrcThread, which controls the read operations, and SsSWThread, which controls the write operations. See Figure 7.

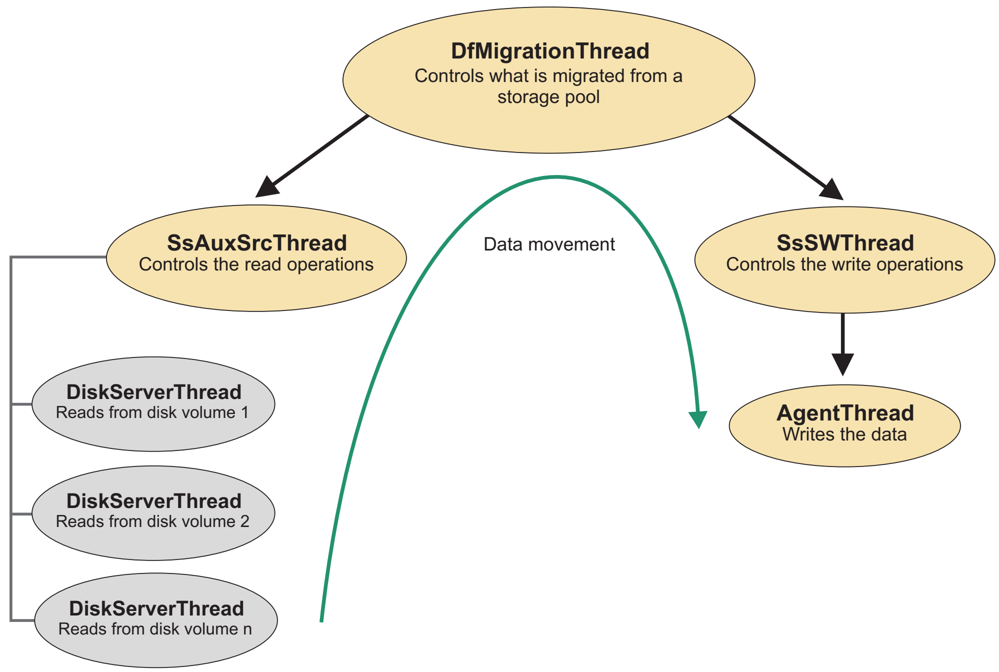

Figure 7. Threads for storage pool migration on AIX, Linux, HP-UX, and Solaris systems

To read the data, the SsAuxSrcThread thread uses a DiskServerThread thread for each volume that must be read. The SsAuxThread thread uses multiple DiskServerThread threads if the data that is being migrated is on more than one volume.

DiskServerThread threads are independent of the SsAuxSrcThread thread. For each volume in a random-access storage pool, a DiskServerThread runs constantly to read and write from that particular volume. For example, if the storage pool has 10 disk volumes, then 10 DiskServerThread threads are always running. Because the SsAuxThread is not a parent for the DiskServerThread threads, you cannot use the ID of the SsAuxThread thread to find a DiskServerThread that is being used.

To write the data, the SsSWThread thread controls a child thread that is called AgentThread, which writes the data to the target volume.

Data movement starts with the DiskServerThread that reads the data from the volume that has the data to be migrated. That data is processed through the SsAuxSrcThread, DfMigrationThread, and SsSWThread threads. Data movement ends when the data is written on the target volume by the AgentThread thread that writes the data.

Windows

The main thread, DfMigrationThread, does the work of selecting the data for migration and the volumes that are read from and written to. The thread starts two child threads: SsAuxSrcThread, which controls the read operations, and SsSWThread, which controls the write operations. The

SsAuxSrcThread thread reads data directly from the disks, without using other threads. For writing the data, the SsSWThread thread controls a separate child thread that is called AgentThread, which writes the data to the target volume.

See Figure 8.

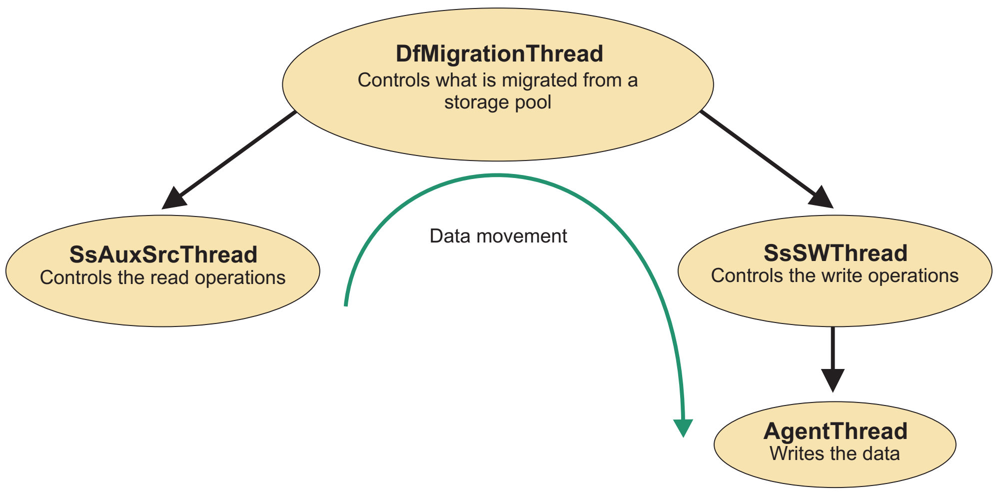

Figure 8. Threads for storage pool migration on Windows systems

Data movement starts with the SsAuxSrcThread that reads the data from the volume that has the data to be migrated. That data is processed through the DfMigrationThread and SsSWThread threads. Data movement ends when the data is written on the target volume by the AgentThread thread that writes the data.

Backups for random-access storage pools

The main thread for a backup operation for a random-access storage pool is DfBackupPoolThread. The threads for reading from the random-access storage pool differ by operating system.

AIX, Linux, HP-UX, and Solaris
The main thread, DfBackupPoolThread, controls the work for the backup operation, including selection of volumes and reading and writing the data. The thread starts two child threads: SsAuxSrcThread, which controls the read operations, and SsSWThread, which controls the write operations. See Figure 9 on page 86.

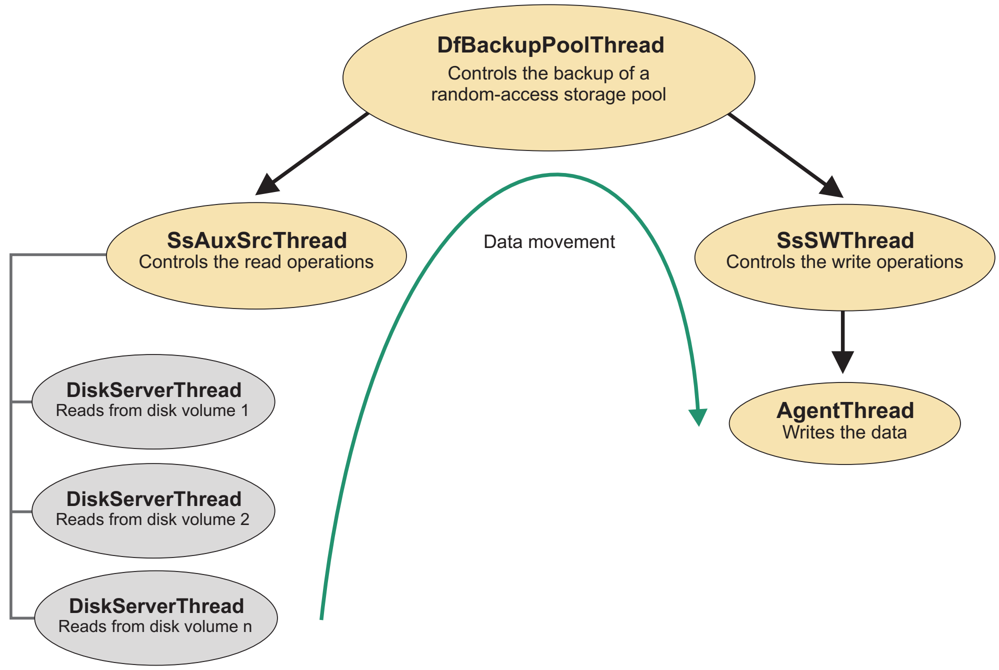

Figure 9. Threads for backup of random-access storage pools on AIX, Linux, HP-UX, and Solaris systems

To read the data, the SsAuxSrcThread thread uses a DiskServerThread thread for each volume that must be read. The SsAuxThread thread uses multiple DiskServerThread threads if the data that is being migrated is on more than one volume.

DiskServerThread threads are independent of the SsAuxSrcThread thread. For each volume in a random-access storage pool, a DiskServerThread runs constantly to read and write from that particular volume. For example, if the storage pool has 10 disk volumes, then 10 DiskServerThread threads are always running. Because the SsAuxThread is not a parent for the DiskServerThread threads, you cannot use the ID of the SsAuxThread thread to find a DiskServerThread that is being used.

To write the data, the SsSWThread thread controls a child thread that is called AgentThread, which writes the data to the target volume.

Data movement starts with the DiskServerThread that reads the data from the volume that has the data to be backed up. That data is processed through the SsAuxSrcThread, DfBackupPoolThread, and SsSWThread threads. Data movement ends when the data is written on the target volume by the AgentThread thread that writes the data.

Windows

The main thread, DfBackupPoolThread, controls the work for the backup operation, including selection of volumes and reading and writing the data. The thread starts two child threads: SsAuxSrcThread, which controls the reading of data, and SsSWThread, which controls the writing of data. The SsAuxSrcThread thread reads the data directly from the disks, without using other threads. For writing the data, the SsSWThread thread controls a separate child thread that is called AgentThread, which writes the data to the target volume. See Figure 10 on page 87.

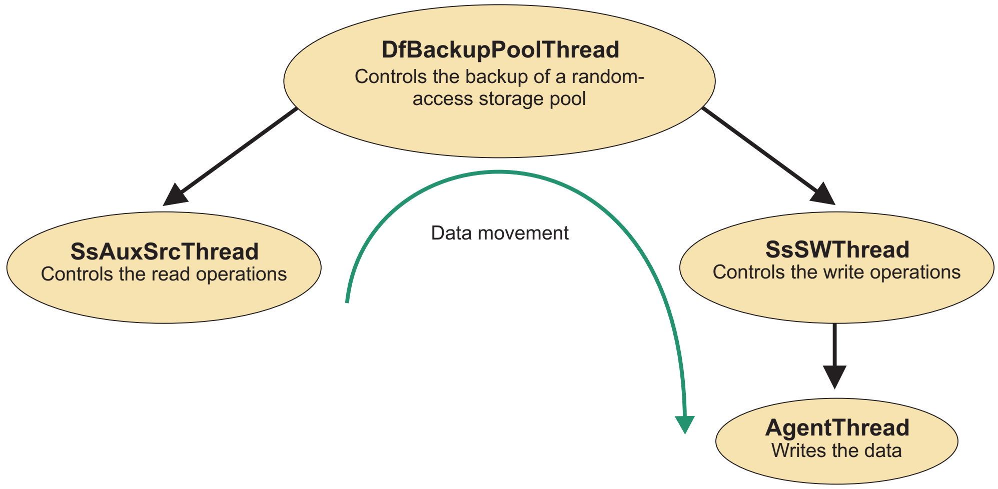

Figure 10. Threads for backup of random-access storage pools on Windows systems

Data movement starts with the SsAuxSrcThread that reads the data from the volume that has the data to be backed up. That data is processed through the DfBackupPoolThread and SsSWThread threads. Data movement ends when the data is written on the target volume by the AgentThread thread that writes the data.

Backups for sequential-access storage pools

The main thread for a backup operation for a sequential-access storage pool is AfBackupPoolThread. This thread controls the work for the backup operation, including selection of volumes and reading and writing the data. The main thread starts two child threads: SsAuxSrcThread, which controls the read operations, and SsSWThread, which controls the write operations. Each of these child threads controls a separate child thread that is called AgentThread, which either reads or writes the data. See Figure 11.

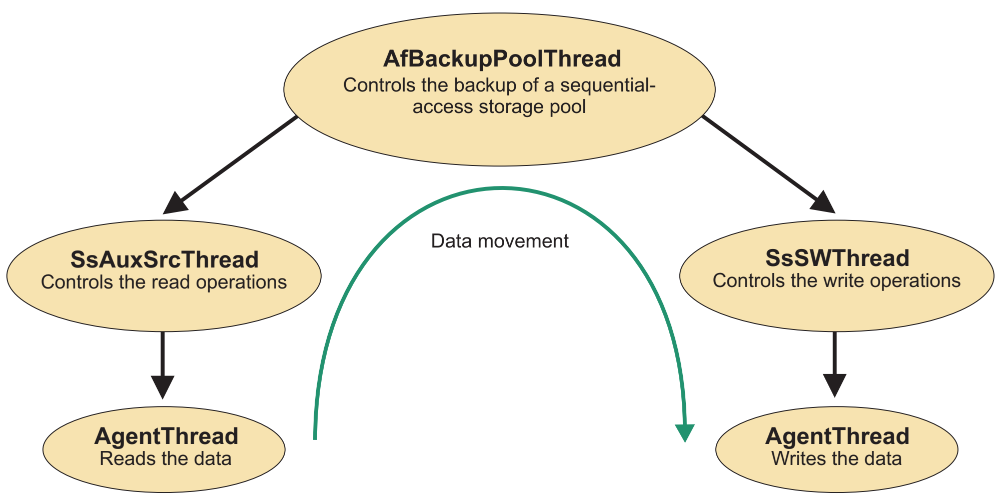

Figure 11. Threads for backup of sequential-access storage pools

Data movement starts with the AgentThread that reads the data from the volume that is being backed up. That data is processed through the SsAuxSrcThread, AfBackupPoolThread, and SsSWThread threads. Data movement ends when the data is written on the target volume by the AgentThread thread that writes the data.

Copying active data for storage pool volumes

The main thread for a copy operation for a storage pool volume is called DfCopyActiveDataThread. The main thread starts one or two child threads. Each child thread controls a thread that is called AgentThread. See Figure 12.

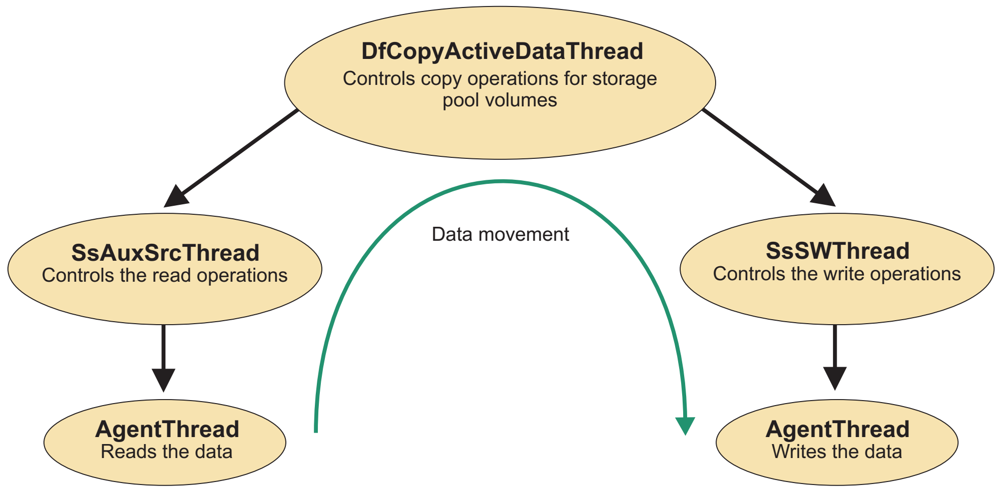

Figure 12. Threads for copying active data for storage pools

A typical data movement operation starts with an AgentThread that reads an object from a volume that is being copied. That data is processed through the SsAuxSrcThread, DfCopyActiveDataThread, and SsSWThread threads. Data movement ends when the data is written on the target volume by the AgentThread thread that writes the data.

Replicating data from a source server

The main threads for replicating data from a source server to a target server are NrReplicateFilespace. This thread determines the data that must be replicated, which is a database heavy task and it is expected that database activities are dominant. NrReplicateBatch threads then send the data to the target server over the network. To read the data, the NrReplicateBatch threads start a child thread, SsAuxSrcThread, which controls the read operations. The NrReplicateBatch thread sends the data that is identified by the NrReplicateFilespace threads to the target server. See Figure 13 on page 89.

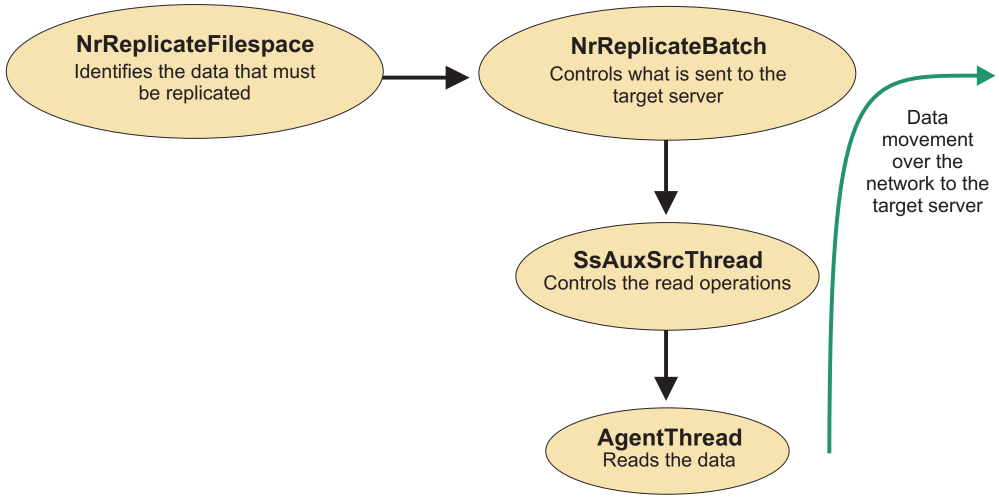

Figure 13. Threads for replicating data from a source server

Time spent in "Thread Wait" by the NrReplicateBatch threads can be spent in one of the following ways:

- Waiting for the NrReplicateFilespace thread to provide lists of files to replicate
- Waiting for the SsAuxSrcThread to read the source data from storage

The NrReplicateBatch threads control the network messages that are sent to the target server and the database.

Expire inventory

The main thread for expire inventory is ExpirationProcessThread. Expire inventory does not move data and is a database intensive operation. It is expected that database operations are dominant in this thread. There might be several of these threads active, depending on the RESOURCE option that is used. See Figure 14 on page 90.

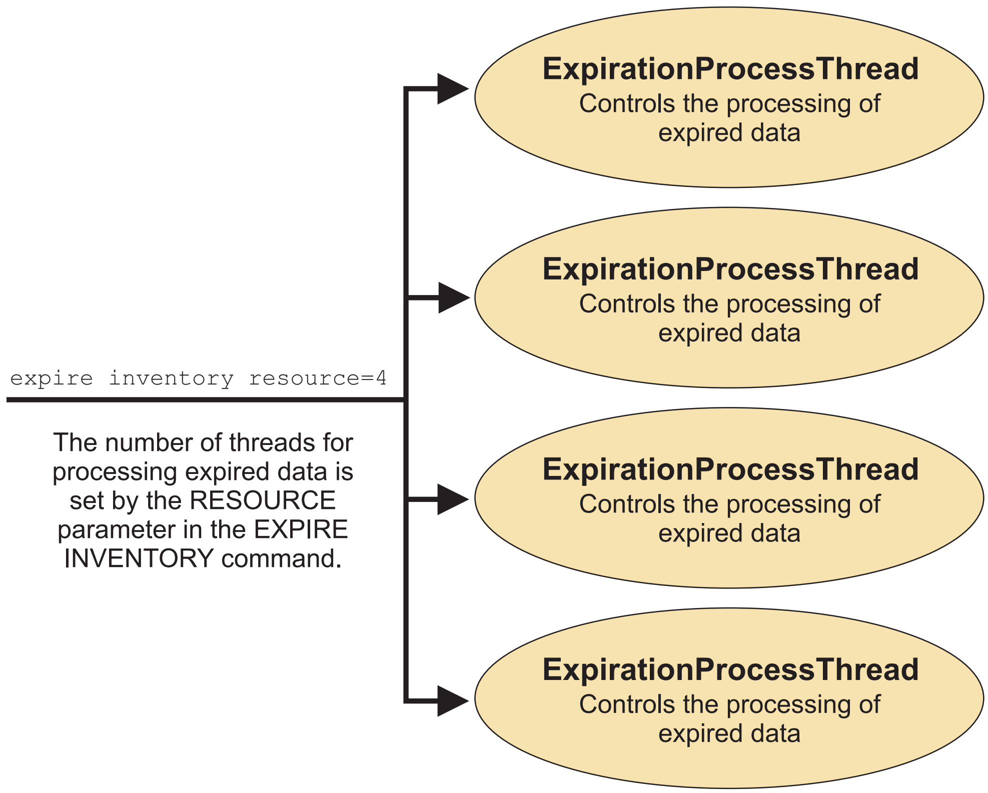

Figure 14. Threads for expire inventory

Client instrumentation report

Use client instrumentation to collect performance data on the Tivoli Storage Manager backup-archive client.

Collecting instrumentation data with the client

Tivoli Storage Manager client instrumentation identifies the elapsed time that is spent performing specific activities. By default, instrumentation data is automatically collected by the backup-archive client during backup or restore processing.

About this task

To disable or later enable instrumentation, use the `enableinstrumentation` option.

With this option enabled, you do not have to wait for a customer service representative to direct you to collect performance data when a problem occurs. Instead, the data is collected whenever you run a backup or restore operation. This feature can be helpful because you do not have to re-create the problem just to collect performance data. The information is already collected by the client.

The default setting of this option is *yes*, which means that instrumentation data is collected even if you do not specify this option. Typically, collecting instrumentation data by default has no measurable impact on regular performance.

By default, the output is appended to the instrumentation log file (`dsminstr.log`) in the directory that is specified by the DSM_LOG environment variable. If you did not set the DSM_LOG environment variable, the instrumentation log file is stored in the current directory (the directory where you started the **dsmc** command).

You can optionally change the name and location of the instrumentation log file by using the `instrlogname` option. You can also control the size of the log file by specifying the `instrlogmax` option.

Instrumentation data is not collected for the backup-archive client GUI or web client GUI.

The `enableinstrumentation` option replaces the `-TESTFLAG=instrument:detail`, `-TESTFLAG=instrument:API`, and `-TESTFLAG=instrumentation:detail/API` options that are used in previous versions of the client. For information about using these option, see Starting client instrumentation.

Procedure

To collect client instrumentation data, use one of the following methods:

- By default, client instrumentation data is collected during backup or restore processing so you do not need to update the client options file.

  However, if you need to turn off the collection of instrumentation data, set the `enableinstrumentation no` option in the client options file (`dsm.opt` on Windows or `dsm.sys` on UNIX and Linux clients).

  To later enable instrumentation, set `enableinstrumentation yes` or remove the option from the client options file.
- If the `enableinstrumentation no` option is set in the client options file, you can start client instrumentation when you run a backup or restore operation by including the `-enableinstrumentation=yes` option in a command.

  For example, on Windows clients, start a selective backup and client instrumentation by issuing the following command:

  ```
  dsmc sel c:\mydir\* -subdir=yes -enableinstrumentation=yes
  ```

  For example, on UNIX and Linux clients, start a selective backup and client instrumentation by issuing the following command:

  ```
  dsmc sel /home/mydir/* -subdir=yes -enableinstrumentation=yes
  ```

  Similarly, you can turn off client instrumentation when you run a backup or restore operation by including the `-enableinstrumentation=no` option in a command.

Example

The following example shows the type of data that is collected:

```
PROCESS[4428] Starting TSM Instrumentation Report: Mon Apr 18 10:58:05 2016

========================>PROCESS[4428] NEW COMMENCE REPORT<========================

-----------------------------------------------------------------------------------
PROCESS[4428] TSM Client final instrumentation statistics: Mon Apr 18 10:58:05 2016

Instrumentation class: Client detail
Completion status: Success

-----------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
No instrumented activity reported for thread 4420

--------------------------------------------------------------------------------

Detailed Instrumentation statistics for

Thread: 5076  Elapsed time =   510.979 sec

Section                  Actual(sec) Average(msec) Frequency used
--------------------------------------------------------------------------------
Compute                      0.218       0.0      27535
BeginTxn Verb                0.000       0.0         32
Transaction                  0.374      11.7         32
File I/O                     2.668       0.1      20702
Compression                 32.105       1.2      27520
Data Verb                  445.225      64.3       6927
Confirm Verb                 0.000       0.0          1
EndTxn Verb                  0.000       0.0         32
TCP Read                    29.422     198.8        148
Thread Wait                  0.905     904.8          1
Other                        0.062       0.0          0

--------------------------------------------------------------------------------

Detailed Instrumentation statistics for

Thread: 5532  Elapsed time =   438.018 sec

Section                  Actual(sec) Average(msec) Frequency used
--------------------------------------------------------------------------------
Process Dirs                 0.140       9.4         15
Solve Tree                   0.000       0.0          1
Sleep                        0.062      62.4          1
TCP Read                     0.546      39.0         14
Thread Wait                437.206     950.4        460
Other                        0.062       0.0          0

--------------------------------------------------------------------------------

Detailed Instrumentation statistics for

Thread: 5620  Elapsed time =   512.383 sec

Section                  Actual(sec) Average(msec) Frequency used
--------------------------------------------------------------------------------
Sleep                        0.125      62.4          2
TCP Read                     0.796      44.2         18
Thread Wait                510.495    1012.9        504
Other                        0.967       0.0          0

--------------------------------------------------------------------------------
No instrumented class associated with thread 6108
Current command:
sel c:\fileLoad\* -subdir=yes

IBM Tivoli Storage Manager
Command Line Backup-Archive Client Interface
  Client Version 7, Release 1, Level 6.18 20160418A
  Client date/time: 04/18/2016 10:58:05

Options settings:
                BACKUPREGISTRY: YES
               CHANGINGRETRIES: 4
           COLLOCATEBYFILESPEC: NO
                    COMMMETHOD: TCP/IP
                COMPRESSALWAYS: YES
```

```
                    COMPRESSION: YES
                 DEDUPCACHEPATH: c:\Program Files\Tivoli\tsm\baclient
                 DEDUPCACHESIZE: 256
                  DEDUPLICATION: NO
                   DISKBUFFSIZE: 32
               ENABLEDEDUPCACHE: YES
                  ENABLELANFREE: NO
                 ENCRYPTIONTYPE: AES128
                 FOLLOWSYMBOLIC: CLC
                   IMAGEGAPSIZE: 32
              LANFREECOMMMETHOD: NAMED PIPE
                 MAKESPARSEFILE: YES
                  MAXCMDRETRIES: 2
          MEMORYEFFICIENTBACKUP: NO
                       NODENAME: OEMTEST10
                 PASSWORDACCESS: TRUE
         PRESERVELASTACCESSDATE: NO
           PROCESSORUTILIZATION: 0
                        REPLACE: TRUE
            RESOURCEUTILIZATION: 2
                   SKIPMIGRATED: NO
              SKIPNTPERMISSIONS: NO
              SKIPNTSECURITYCRC: NO
              SNAPSHOTCACHESIZE: 1
                         SUBDIR: TRUE
                  SUBFILEBACKUP: NO
               SUBFILECACHESIZE: 10 MB
                     TAPEPROMPT: NO
                    TCPBUFFSIZE: 32 KB
                     TCPNODELAY: YES
                TCPSENDBUFFSIZE: 0 KB
                  TCPWINDOWSIZE: 63 KB
                   TXNBYTELIMIT: 25600K
                        VERBOSE: VERBOSE

-------------------------

Session established with server ARC1: AIX
  Server Version 7, Release 1, Level 4.100
  Server date/time: 04/18/2016 08:54:40  Last access: 04/18/2016 08:37:01

Total number of objects inspected:           79
Total number of objects backed up:           79
Total number of objects updated:              0
Total number of journal objects:              0
Total number of objects rebound:              0
Total number of objects deleted:              0
Total number of objects expired:              0
Total number of objects failed:               0
Total number of objects encrypted:            0
Total number of bytes transferred:       212.71 MB
LanFree data bytes:                           0  B
Data transfer time:                      445.11 sec
Network data transfer rate:              489.35 KB/sec
Aggregate data transfer rate:            426.23 KB/sec
Total number of bytes pre-compress:  671,102,861
Total number of bytes post-compress: 222,963,689
Total number of objects grew:                 0
Total number of retries:                      0
Objects compressed by:                       67%
Total data reduction ratio:               66.77%
Elapsed processing time:               00:08:31
Average file size:                      8.10 MB

PROCESS[4428] Ended TSM Instrumentation Report: Mon Apr 18 11:06:38 2016

-----------------------------------------------------------------------------------
```

Related information:

Enableinstrumentation

Instrlogname

Instrlogmax

Client instrumentation categories

Tivoli Storage Manager client instrumentation reports the elapsed times for many process categories.

The following table lists the client instrumentation categories that are tracked and the activity that is timed.

Table 6. Client instrumentation categories

Category	Activity
Query Server Dirs	Receiving the server inventory directories for incremental backup
Query Server Files	Receiving the server inventory files for incremental backup
Process Dirs	Scanning for files to back up
Cache Examine	Scanning the local disk cache database for files to expire
Solve Tree	Determining directory structure
Compute	Computing throughput and compression ratio
BeginTxn Verb	Building transactions
Transaction	File open, close, and other miscellaneous operations
File I/O	File read and write
Compression	Compressing and uncompressing data
Encryption	Encrypting and decrypting data
CRC	Computing and comparing CRC values
Delta	Adaptive subfile backup processing
Data Verb	Sending and receiving data to and from the server (points to the network or Tivoli Storage Manager server)
Confirm Verb	Response time during backup for server confirm verb
EndTxn Verb	Server transaction commit and tape synchronization (points to the Tivoli Storage Manager server)
Other	Everything else that is not tracked already

Cloud instrumentation processes

AIX Linux Solaris Windows

IBM Tivoli Storage Manager reports the time that it takes for certain processes that run in a cloud environment to complete.

The following table lists the cloud instrumentation processes that are tracked and the activity that is timed.

Table 7. Cloud instrumentation processes

Process	Time that is spent on this activity
INST_CLOUD_CONNECT	Connecting to the cloud.
INST_CLOUD_CONT	Creating, deleting, or managing cloud containers.
INST_CLOUD_DELETE	Deleting objects from cloud containers.
INST_CLOUD_ATCH	Attaching to the Tivoli Storage Manager server Java™ virtual machine (JVM).
INST_CLOUD_DTCH	Detaching from the Tivoli Storage Manager server JVM.
INST_CLOUD_STATS	Collecting and reporting cloud statistics for the Operations Center.
INST_CLOUD_READ	Read operations from the specified cloud provider.
INST_CLOUD_WRITE	Write operations to the specified cloud provider.

Virtual machine instrumentation categories

IBM Tivoli Storage Manager virtual machine (VM) instrumentation reports the elapsed times for many process categories.

The following table lists the virtual machine instrumentation categories that are tracked and the activity that is timed.

Table 8. Virtual machine instrumentation categories

Category	Activity
VM Snapshot	Time that is spent generating and removing a VM guest snapshot by using the VMware Infrastructure Software Development Kit (VI SDK). Some of the work is completed asynchronously, for example, snapshot delete.
VM Send Data	Time that is spent sending data to the Tivoli Storage Manager server. Data processing includes client-side data deduplication and the `Network Send` phase.
VM Get Data	Time that is spent retrieving data from the Tivoli Storage Manager server. This category includes the following activities: • Retrieving control files from the Tivoli Storage Manager server during incremental backup. • Buffering data that is received during VM guest restore, and is then written out by using VM I/O writes to VMware.
VM Query	Time that is spent querying the Tivoli Storage Manager server to determine the following: • Whether data deduplication and compression, or both, are enabled. • Whether file space queries for nodes on virtual machines are used.

Table 8. Virtual machine instrumentation categories (continued)

Category	Activity
VM Query VE	Time that is spent querying whether VMware VM guest disks are changed. This category uses the VMware VI SDK to identify a set of changed blocks.
VM Assign	Time that is spent assigning file groups to the Tivoli Storage Manager server, by using the `dsmGroupHandler` function.
VM VCM Lock	Time that is spent locking a semaphore during Volume Control Manager (VCMLIB) API calls. The time is spent on the following functions: • Locking for volume control data reads or updates • Manipulating data by block control • Retrieving megablocks in a volume
VM Transaction	Time that is spent processing transactions with the Tivoli Storage Manager server.
VM I/O	Time that is spent reading and writing data to and from the VIX Disk Library for Virtual Disk Development Kit (VDDK) disks for a VM. Performance can vary depending on whether thin or thick-provisioned disks are used and if the disks are lazy zeroed.
VM Control File I/O	Time that is spent reading and writing VM guest control (CTL) files during VM guest backup and restore operations.
Thread Wait	Time that is spent opening and closing VDDK disks on IBM Tivoli Storage Manager for Virtual Environments clients. Opening and closing of VM disks is serialized for a Tivoli Storage Manager client instance. The time that it takes to open and close the VM disks includes mounting the disk on the client or proxy system.

API instrumentation report

Use API instrumentation to collect performance data on applications that use the Tivoli Storage Manager API.

API instrumentation identifies the time that elapsed during application activities. It is used for applications and products that use the API. The following products use the API:

- Tivoli Storage FlashCopy Manager
- Tivoli Storage Manager for Mail
- Tivoli Storage Manager for Databases
- Tivoli Storage Manager for Virtual Environments
- Tivoli Storage Manager for Enterprise Resource Planning

Collecting instrumentation data with the API

API instrumentation identifies the time that elapsed during application activities. It is used for applications and products that use the Tivoli Storage Manager API. By default, instrumentation data is automatically collected by the API during backup or restore processing.

About this task

To disable or later enable instrumentation, use the `enableinstrumentation` option.

With this option enabled, you do not have to wait for a customer service representative to direct you to collect performance data when a problem occurs. Instead, the data is collected whenever you run a backup or restore operation. This feature can be helpful because you do not have to re-create the problem just to collect performance data. The information is already collected by the API.

The default setting of this option is *yes*, which means that instrumentation data is collected even if you do not specify this option. Typically, collecting instrumentation data by default has no measurable impact on regular performance.

By default, the output is appended to the instrumentation log file (`dsminstr.log`) in the directory that is specified by the DSM_LOG environment variable (or the DSMI_LOG environment variable for API-dependent products such as Tivoli Storage Manager for Databases: Data Protection for Microsoft SQL Server and Tivoli Storage Manager for Mail: Data Protection for Microsoft Exchange Server). If you did not set the DSM_LOG environment variable, the instrumentation log file is stored in the current directory (the directory where you started the **dsmc** command).

You can optionally change the name and location of the instrumentation log file by using the `instrlogname` option. You can also control the size of the log file by specifying the `instrlogmax` option.

The `enableinstrumentation` option replaces the `-TESTFLAG=instrument:API` option that is used in previous versions of the API. For information about using the `-TESTFLAG=instrument:API` option, see Starting API client instrumentation.

Procedure

To collect API instrumentation data, use one of the following methods:

- By default, API instrumentation data is automatically collected during backup or restore processing so you do not need to update the client options file.

  However, if you need to turn off the collection of instrumentation data, set the `enableinstrumentation no` option in the client options file (`dsm.opt` on Windows or `dsm.sys` on UNIX and Linux clients).

  To later enable instrumentation, set `enableinstrumentation yes` or remove the option from the client options file.
- To turn API instrumentation on the command-line interface, append the following option to the end of a command:

  ```
  -enableinstrumentation=yes
  ```

  To turn off API instrumentation on the command-line interface, append the following option to the end of a command:

  ```
  -enableinstrumentation=no
  ```

Results

The categories for tracking API activities are different from the client instrumentation categories.

Example

The following example shows the type of data that is collected:

```
PROCESS[4120] Starting TSM Instrumentation Report: Mon Apr 18 10:43:13 2016

========================>PROCESS[4120] NEW COMMENCE REPORT<========================

-----------------------------------------------------------------------------------
PROCESS[4120] TSM Client final instrumentation statistics: Mon Apr 18 10:43:13 2016

Instrumentation class: API
Completion status: Success

-----------------------------------------------------------------------------------

-----------------------------------------------------------------------------------

Detailed Instrumentation statistics for

Thread: 5472  Elapsed time =     3.354 sec

Section                  Actual(sec) Average(msec) Frequency used
-----------------------------------------------------------------------------------
Waiting on App             3.354     838.5         4
API Send Data              0.000       0.0         3
Other                      0.000       0.0         0

-----------------------------------------------------------------------------------

Detailed Instrumentation statistics for

Thread: 4208  Elapsed time =     9.703 sec

Section                  Actual(sec) Average(msec) Frequency used
-----------------------------------------------------------------------------------
Waiting on App             4.009     167.1        24
API Send Data              4.914     614.3         8
API Query                  0.062      31.2         2
API End Txn                0.499     166.4         3
API Misc                   0.218      72.8         3
Other                      0.000       0.0         0

-----------------------------------------------------------------------------------

Detailed Instrumentation statistics for

Thread: 2268  Elapsed time =    10.109 sec

Section                  Actual(sec) Average(msec) Frequency used
-----------------------------------------------------------------------------------
Waiting on App             9.532    1361.7         7
API Query                  0.312      52.0         6
API End Txn                0.187     187.2         1
API Misc                   0.078      78.0         1
Other                      0.000       0.0         0

-----------------------------------------------------------------------------------

Detailed Instrumentation statistics for
```

```
Thread: 4276  Elapsed time =    18.502 sec

Section                 Actual(sec) Average(msec) Frequency used
---------------------------------------------------------------------------------
Waiting on App           16.193     476.3        34
API Query                 0.842      49.6        17
API Misc                  1.466     209.5         7
Other                     0.000       0.0         0

---------------------------------------------------------------------------------

PROCESS[4120] Ended TSM Instrumentation Report: Mon Apr 18 10:43:32 2016

---------------------------------------------------------------------------------
```

Related information:

`Enableinstrumentation`

`Instrlogname`

`Instrlogmax`

API instrumentation categories

Tivoli Storage Manager API client instrumentation reports the elapsed times for many process categories.

The following table lists the API client instrumentation categories that are tracked and the activity that is timed.

Table 9. API instrumentation categories

Category	Activity
Waiting on App	Time that the Tivoli Storage Manager API is waiting on the application that is sending Tivoli Storage Manager data. For example, the time spent waiting for a database application to send Tivoli Storage Manager data. If this value is high, focus your performance analysis on the application that is sending the data as well as the disk performance.
API Send Data	Time that is spent sending data to the Tivoli Storage Manager server. If the value is high, there might be a network problem or a storage pool performance problem on the Tivoli Storage Manager server.
API Query	Time that is spent querying the Tivoli Storage Manager server for information.
API Get Data	Time spent retrieving data from the Tivoli Storage Manager server. A high value might represent network problems between the server and client or server storage pool performance problems. For example, slow disk speeds or tape mount times.
API End Txn	Time that is spent in committing the current transaction to the Tivoli Storage Manager server. If the value is high, consider changing the settings that might use larger client transactions, or examine the server active log write performance.

Table 9. API instrumentation categories (continued)

Category	Activity
API Dedup fingerprint	Time that is spent calculating the segmentation sizes of the incoming data. This category is a CPU intensive operation.
API ICC Digest (dedup)	Time that is spent computing the hash for the deduplication segments. This category is a CPU intensive operation.
API Query Dedup Cache	Time that is spent querying the deduplication cache on local disk for deduplication segments.
API Query Server Dedup	Time that is spent querying the Tivoli Storage Manager server for data deduplication segments. If this value is high, examine database performance on the Tivoli Storage Manager server.
API Misc	Other minor activities of the Tivoli Storage Manager API client.

Scenarios for analyzing instrumentation data

Scenarios can help to illustrate how to use and interpret instrumentation reports.

Scenario: Improving performance of client backups

The scenario illustrates how to interpret client instrumentation reports to resolve a problem with backup operations.

Problem

Dave, a Tivoli Storage Manager administrator, is experiencing a performance degradation of backup operations on a file server. He is backing up the file server from an AIX backup-archive client to an AIX server, and the performance has degraded from an average throughput of 32 MB per second to 15 MB per second in the last few days.

Goal

Dave expects to restore his previous throughput level during backups.

Data collection

Dave collects the following data:

- Client instrumentation data
- Server monitoring data, which is generated by running the server monitoring script (`servermonv6.pl`)

Analysis and determination of the bottleneck

During the backup operation, the data flows from the client, over the network, to the Tivoli Storage Manager server. The data is transferred from the disk system through a host bus adapter (HBA) that connects the disk to the Tivoli Storage Manager client. The client backs up the data to the server through a local area network (LAN) connection. A Network Interface Card (NIC) connects the client to the LAN and a separate NIC connects the LAN to the server. The data is backed up from the server to disk and to a tape library through separate HBA devices.

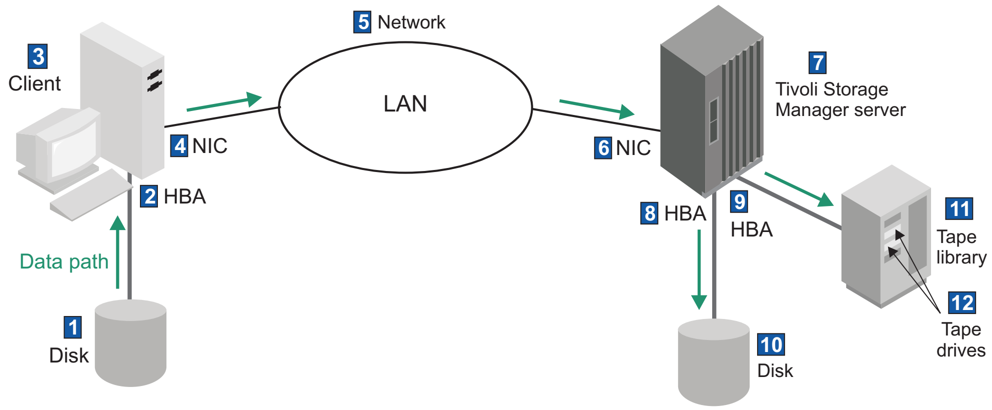

Figure 15. Data flow for client backup operations over a local area network

Following the flow of data, Dave looks at the client data first.

The most recent backup operation finished with the following statistics:

```
Total number of objects inspected:       1
Total number of objects backed up:       1
...
Total number of bytes transferred:    11.80 GB
LanFree data bytes:                   11.80 GB
Server-Free data bytes:                   0  B
Data transfer time:                  216.01 sec
Network data transfer rate:        57,294.91 KB/sec
Aggregate data transfer rate:      16,542.69 KB/sec
Elapsed processing time:           00:12:28
Average file size:                 11.66 GB
```

To determine where the slowdown occurs, Dave runs a test backup of the file server with the `testflag=instrument:detail` option. By setting this option, Dave collects client instrumentation data about the backup operation. He reviews the instrumentation output. The client instrumentation report shows that, during the backup, a large amount of time was spent on `File I/O` operations. The elapsed time of the backup was 746.666 seconds, and 524.380 seconds were spent in `File I/O`. By looking at the client instrumentation categories in Table 6 on page 94, Dave knows that the `File I/O` category represents the elapsed time for file read and write operations.

```
Thread: 2571  Elapsed time   746.666 sec
Section      Actual (sec) Average(msec)  Frequency used
------------------------------------------------------------------
Process Dirs         0.000           0.0              0
Solve Tree           0.000           0.0              0
Compute              0.234           0.0          48345
BeginTxn Verb        0.000           0.1              2
Transaction          0.715         357.5              2
File I/O           524.380          10.8          48346
Compression          0.000           0.0              0
Encryption           0.000           0.0              0
CRC                128.042           2.6          48398
Delta                0.000           0.0              0
Data Verb           87.912           1.8          48345
Confirm Verb         0.136           8.5             16
```

```
EndTxn Verb          2.234       1117.0            2
Other                4.513          0.0            0
------------------------------------------------------------------
```

The rate of data transfer is calculated by dividing the file size by the time that is recorded in the `File I/O` field in the instrumentation output:

```
(11.66 GB x 1024 MB/GB)/524.380s = 22.8 MB/s
```

The backup reads the data only from the client disk, so Dave suspects a problem with the disk on the client system.

Problem resolution

Upon further investigation, Dave discovers that the AIX file system was recently mounted with the `cio` option, which enabled concurrent I/O in the file system. He concludes that mounting the file system with the AIX concurrent I/O option caused a degradation in backup performance. Concurrent I/O prevents file system read-ahead operations.

Dave modified the system settings to ensure that the AIX file system is not mounted with the `cio` option. As a result, the performance of backup operations is restored to its previous level.

Scenario: Improving performance of migration operations

This scenario illustrates how to interpret server monitoring data to improve migration operations.

Problem

Kate is a Tivoli Storage Manager administrator, and she noticed that the disk-to-tape storage pool migrations run slowly on her Windows server.

Goal

Kate expects the write time to be a value close to the capabilities of the drive.

Data collection

Kate collects the following data:

- Server monitoring data, which is generated by running the server monitoring script (`servermonv6.pl`).

  Tip: Kate runs the migration process for about 30 minutes before she starts the script. In this way, she avoids collecting data for tape mounts. She allows the script to run for about 1 hour.

Analysis and determination of the bottleneck

During the migration operation, data is read from the disk and written to tape. The data is transferred from the disk through a host bus adapter (HBA) to the Tivoli Storage Manager server. The data is transferred from the server through a separate HBA to the tape system.

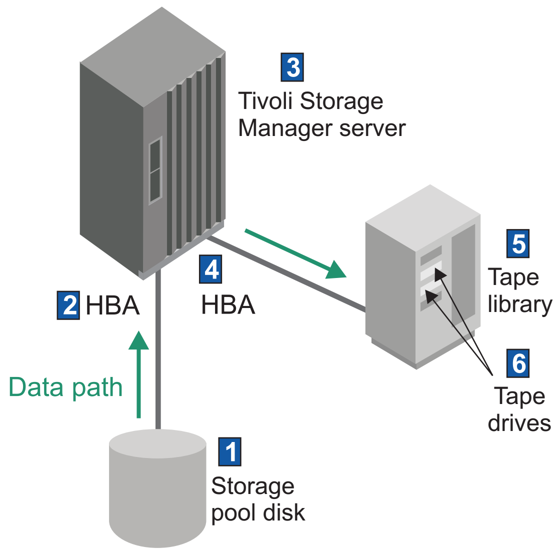

Figure 16. Data flow for migration operations

Following the flow of data, Kate looks at the disk read activity first.

As Kate investigates, she notices that all tape operations are slow, including client backups, reclamation, and database backups to tape. Kate suspects a problem with the tape system, which might be causing slow write times. The tape system is an LTO library with SCSI-attached tape drives.

To determine where the slowdown occurs, Kate runs a disk-to-tape storage pool migration test and collects server instrumentation data, by using the server monitoring script. She locates the files that were collected by the server monitoring script, and finds the files that are time-stamped during a time when migration was slow. She looks for the files that are in the form: `YYYYMMDD-HHMM-YYYYMMDD-HHMM-instr.txt`, where the time stamps represent the start and end times when the output was gathered.

Kate reviews the file, and finds Thread 61, which processed a total of 966912 KB. Then, she finds the corresponding thread that has a `Tape Write` category in it, which processed about the same amount of data. She found Thread 34, which processed 968192 KB.

```
Thread 61 DfMigrationThread (Win Thread ID 4436)  17:39:076-->17:47:38

Operation       Count  Tottime  Avgtime  Min-   Max-    Inst     Total
                                         time   time    Tput     KB
---------------------------------------------------------------------
Disk Read        3777   22.680    0.006  0.000  0.031  42632.8  966912
Thread Wait      3778  487.450    0.129  0.016  0.313
Unknown                  0.061
----------------------------------------------------------------------
Total                  510.191                          1895.2  966912

Thread 34 AgentThread (Win Thread ID 5340)  17:39:07.816-->17:47:38.007

Operation       Count  Tottime  Avgtime  Min-   Max-    Inst     Total
                                         time   time    Tput     KB
---------------------------------------------------------------------
Tape Write      30257  508.816    0.017  0.000  0.141   1902.8  968192
Tape Data Copy  31661    0.863    0.000  0.000  0.016
Thread Wait      3777    0.220    0.000  0.000  0.016
Unknown                  0.292
----------------------------------------------------------------------
Total                  510.191                          1897.7  968192
```

Kate used the server instrumentation categories in Table 5 on page 81 to understand the meaning of the `Disk Read`, `Tape Write`, and `Thread Wait` categories. The server instrumentation output shows the following characteristics:

- The Thread 61 data shows a high value for the `Thread Wait` field (487.450 seconds), and low value for the `Disk Read` field (22.680 seconds). This result indicates that Thread 61 is not a concern because the instantaneous throughput is acceptable and the thread was in a waiting phase. The disk is not a problem.

  Tip: The instantaneous throughput, `Inst Tput`, is calculated by dividing the `Total KB` value by `Tottime` value.
- The Thread 34 data shows that there is a low value for the `Thread Wait` field (0.220 seconds), and a high value for the `Tape Write` field (508.816 seconds). The output also shows that the `Inst Tput` value for the tape write is very slow (1902.8 KB per second). She concluded that the problem is related to the tape system, as shown by the high `Tape Write` value.

Problem resolution

Kate investigates the following probable sources of the tape system problem:

- The tape attachment path
- The device driver level of the tape drive
- The SCSI adapter driver level
- The SCSI adapter settings

After the investigation, Kate upgrades the SCSI adapter device driver. The disk-to-tape storage pool migration improves to 75% of the native capacity. Client backups to tape are also much faster.

Tip: All server speeds depend on the hardware that is used. The values in this scenario might not apply to your system. Refer to the performance characteristics of your tape drive or disk system to determine an acceptable level of performance.

Scenario: Improving performance of database backups

This scenario illustrates how to interpret API client instrumentation data and server monitoring data to improve database backup operations.

Problem

Dave, a system administrator, notices a performance degradation in server database backups after a Tivoli Storage Manager upgrade.

Goal

Dave expects the time to complete a database backup to be the same as it was before the upgrade.

Data collection

Dave collects the following data:

- API client instrumentation data
- Server monitoring data, which is generated by running the server monitoring script (`servermonv6.pl`)

Dave runs the API client instrumentation, and the server monitoring script for the entire duration of the database backup.

Analysis and determination of the bottleneck

During database backup operations, the database is read and then backed up to the target storage pool. The data is backed up from the disk where the Tivoli Storage Manager database is located through a host bus adapter (HBA) to the Tivoli Storage Manager server. The data is transferred from the server through a separate HBA to the tape system.

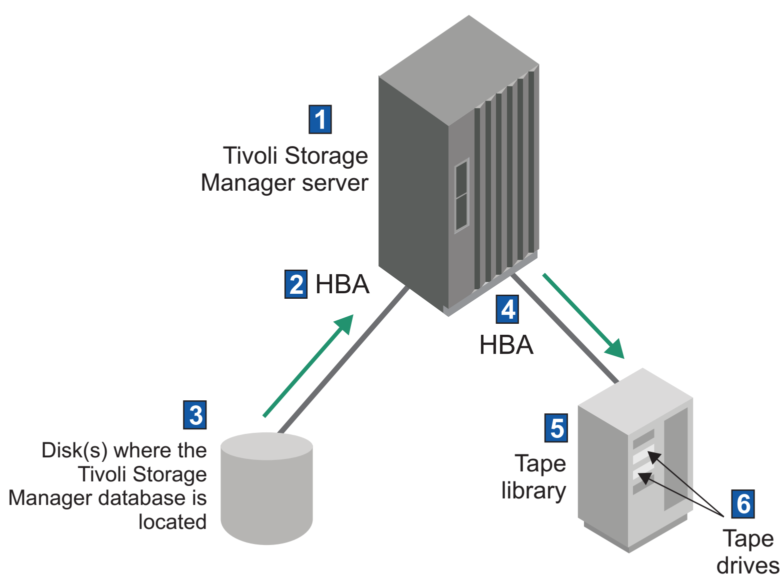

Figure 17. Data flow for database backup operations

Dave starts by reviewing data from the API client instrumentation. He finds that Thread 1 shows a backup period of almost 5 hours (17954.687 seconds). About 99% of the time was spent in the `API send data` category. By reviewing the API client categories in Table 9 on page 99, Dave determines that most of the time was spent either sending data to the Tivoli Storage Manager server through the network, or writing the data to the backup device.

```
Thread: 1  Elapsed time = 17954.687 sec (1)

Section                  Actual(sec) Average(msec)  Frequency used
------------------------------------------------------------------
Waiting on App            168.155       2.8          59246
API Send Data           17786.518     300.3          59236 (2)
API Query                   0.001       0.1          4
API End Txn                 0.004       4.1          1
API Misc                    0.009       3.0          3
Other                       0.000       0.0          0
------------------------------------------------------------------
```

Dave reviews the server instrumentation data. He determines the time when the database backup was started by examining the Tivoli Storage Manager activity log. He locates the files that were collected by the server monitoring script, and finds the files that are time-stamped during the database backup. The file names have the following structure: `YYYYMMDD-HHMM-show.txt`. By examining a file that was time-stamped during the database backup, Dave finds the thread number for the database backup.

Tip: For backups of server databases, the associated node name is always `$$_TSMDBMGR_$$`.

Dave searches the data for a node that is named `$$_TSMDBMGR_$$`, and finds the information for the database backup session:

```
Session 22486:    Type=Node,   Id=$$_TSMDBMGR_$$
    Platform=DB2/AIX64, NodeId=1, Owner=tsminst
    SessType=4, Index=0, TermReason=0
    threadId=24431
    ProxyByAgent False
    RecvWaitTime=0.000  (samples=0)
    Backup  Objects ( bytes )  Inserted: 0 ( 0.0 )
    Backup  Objects ( bytes )  Restored: 0 ( 0.0 )
    Archive Objects ( bytes )  Inserted: 0 ( 0.0 )
    Archive Objects ( bytes ) Retrieved: 0 ( 0.0 )
    Last Verb ( ConfirmResp ), Last Verb State ( Sent )
    Global id reports 0 mount points in use
    Write MP count 0, read MP count 0 keepUsedMP = No.
```

The information shows that Thread 24431 is associated with the database backup. Dave opens the server instrumentation report from the same time frame. The file names for server instrumentation reports have the format `YYYYMMDD-HHMM-YYYYMMDD-HHMM-instr.txt`, where the time stamps represent the start and end times when the output was gathered. He searches for Thread 24431, and finds the following output:

```
Thread 24431   00:21:34.695-->00:43:20.577
Operation       Count  Tottime  Avgtime  Mintime  Maxtime InstTput  Total KB
----------------------------------------------------------------------------
Network Recv   660678 1190.148    0.002    0.000   64.847  15556.7  18514797(3)
Network Send       21    0.000    0.000    0.000    0.000                  0
Thread Wait     72323  112.404    0.002    0.000   33.003
Unknown                  3.328
----------------------------------------------------------------------------
Total                 1305.881                            14178.0  18514797
```

Dave finds that most of the time was spent in the `Network Recv` phase. By reviewing the server instrumentation categories in Table 5 on page 81, he determines that most of the time was spent receiving data from the network.

Dave determines that the network is causing performance degradation. The client and server reports show lengthy times for sending and receiving data over the network

Problem resolution

Dave identifies network settings that were incorrectly set as a result of the upgrade. Dave corrects the settings, and the performance of the database backup time achieves the same level as before the upgrade.

Scenario: Improving performance of restore operations for database applications

This scenario illustrates how to interpret API client instrumentation data and server monitoring data to improve database application restore operations.

Problem

Kate, a Tivoli Storage Manager administrator, notices a performance degradation in restore operations for an SAP application database

Goal

Kate expects to improve the performance of database restore operations.

Data collection

Kate collects the following data:

- API client instrumentation data
- Server monitoring data, which is generated by running the server monitoring script (`servermonv6.pl`)

Kate runs the API client instrumentation and the server monitoring script for the entire duration of the restore operation. She runs a restore operation and collects the instrumentation data during a time of little or no activity on the Tivoli Storage Manager server. Kate uses IBM Tivoli Storage Manager for Enterprise Resource Planning to back up the database for an SAP application.

Analysis and determination of the bottleneck

During the restore operation, data is read from the tape drives, sent over the network to the Tivoli Storage Manager client, and is written to the client disk. The data is transferred from the tape system through a host bus adapter (HBA) that connects the tape library to the server. The server transfers the data to the client system through a local area network (LAN) connection. A Network Interface Card (NIC) connects the server to the LAN and a separate NIC connects the LAN to the client. The data is restored to the disk from the client through a separate HBA device.

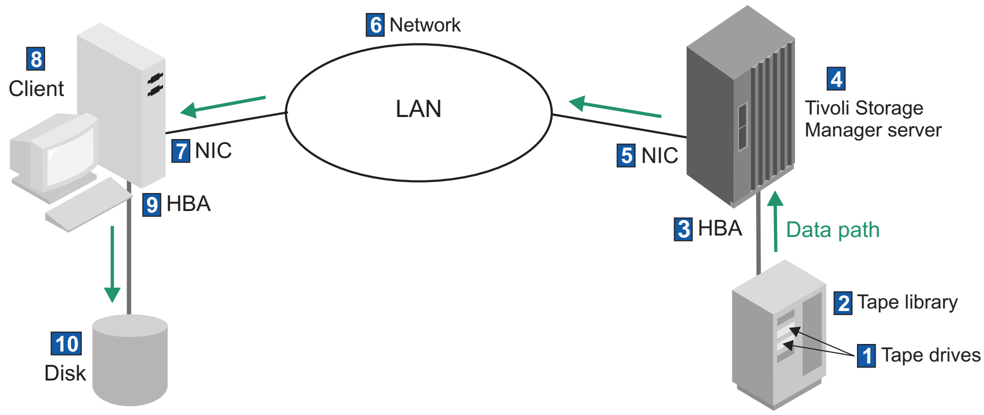

Figure 18. Data flow for restore operations for database applications

Following the flow of data, Kate starts to review the server instrumentation data. First, Kate notes the time when the restore was started by reviewing the Tivoli Storage Manager activity log. Then, she locates the files that were collected by the server monitoring script, and finds the files that are time-stamped around the time when the restore operation was slow. The instrumentation file names have the following structure: `YYYYMMDD-HHMM-YYYYMMDD-HHMM-instr.txt`.

Kate examines the instrumentation output to find a thread that was reading from tape. She finds Thread 131787. Data for the thread shows that 9.100 seconds was spent in the `Tape Read` category and has a fast instantaneous throughput (`InstTput`) of 269584.5 KB per sec. She reviews the server instrumentation categories in Table 5 on page 81 and finds that there is no problem reading from tape. The instantaneous throughput and high amount of thread wait time indicate that tape read is not the bottleneck. Kate sees that 99% of the time was spent in `Thread Wait` (1199.192 seconds):

```
Thread 131787 AgentThread parent=131782  13:16:25.938-->13:36:34.274
 Operation       Count  Tottime  Avgtime  Mintime  Maxtime InstTput  Total KB
 ----------------------------------------------------------------------------
 Tape Read        9583    9.100    0.001    0.000    0.354 269584.5   2453248
 Thread Wait      9585 1199.192    0.125    0.000  267.561
 Unknown                  0.042
 ----------------------------------------------------------------------------
 Total                 1208.335                              2030.3   2453248
```

Following the data flow, she looks for a thread with a `Network Send` category and a thread that processed approximately the same amount of data (2453248 KB). She finds Thread 131781, which processed a total of 2452368 KB. Thread 131781 is a `psSessionThread` thread, which shows the timing of the data as it was sent from the server, through the network to the client. Kate sees that the `InstTput` for the thread was 2052.8 KB per second, which was slower than expected. The `Thread Wait` phase took 0.786 seconds, which was less than 1% of the total time. The data indicates that most of the time was spent in the `Network Send` phase. The findings suggest performance degradation of the network on the client side.

```
Thread 131781 psSessionThread parent=299  13:16:25.938-->13:36:34.274
 Operation       Count  Tottime  Avgtime  Mintime  Maxtime InstTput  Total KB
 ----------------------------------------------------------------------------
 Data Copy           1    0.000    0.000    0.000    0.000
 Network Recv        5   12.778    2.556    0.001   12.719      0.0         0
 Network Send    19170 1194.666    0.062    0.000  267.561   2052.8   2452368
 DB2 Fetch Prep      1    0.003    0.003    0.003    0.003
 DB2 Fetch Exec      8    0.003    0.000    0.000    0.002
 DB2 MFtch Exec      3    0.008    0.003    0.000    0.004
 DB2 Fetch           8    0.000    0.000    0.000    0.000
 DB2 MFetch          5    0.000    0.000    0.000    0.000
 DB2 Commit          2    0.000    0.000    0.000    0.000
 DB2 Reg Prep        2    0.003    0.002    0.000    0.003
 DB2 Reg Exec        2    0.000    0.000    0.000    0.000
 DB2 Reg Fetch       2    0.000    0.000    0.000    0.000
 Thread Wait      9584    0.786    0.000    0.000    0.351
 Unknown                  0.085
 ----------------------------------------------------------------------------
 Total                 1208.335                              2029.5   2452368
```

Next, Kate looks at the API client instrumentation report.

```
Detailed Instrumentation statistics for

Thread: 2911  Elapsed time =  1669.061 sec

Section                  Actual(sec) Average(msec) Frequency used
-----------------------------------------------------------------
Waiting on App         1435.153      94.8     15131
API Get Data            233.909      15.5     15131
Other                     0.000       0.0         0

-----------------------------------------------------------------

Detailed Instrumentation statistics for

Thread: 2902  Elapsed time =  1913.868 sec

Section                  Actual(sec) Average(msec) Frequency used
-----------------------------------------------------------------
Waiting on App         1681.437     110.4     15224
API Get Data            232.432      15.3     15224
Other                     0.000       0.0         0

-----------------------------------------------------------------

Detailed Instrumentation statistics for

Thread: 2893  Elapsed time =  2093.726 sec

Section                  Actual(sec) Average(msec) Frequency used
-----------------------------------------------------------------
Waiting on App         1926.681     127.1     15153
API Get Data            167.045      11.0     15153
Other                     0.000       0.0         0

-----------------------------------------------------------------
```

By reviewing the data from the API client instrumentation, Kate sees that the longest amount of time was spent on the `Waiting on App` category. This category shows the time that was spent by the application to process the restore data. After Kate reviews the API report, she determines that most of the restore time is spent waiting for the application database. As a next step, Kate could verify the performance of the disk system.

Problem resolution

Kate contacts the storage administrator and the SAP administrator to help resolve the problem. After further investigation, the storage administrator determines that the database layout is the source of the problem, and corrects it.

Part 4. Tuning the components

After you verify that you are following the guidelines for optimal configuration, learn about options and parameters that can help you to tune performance.

Chapter 10. Tuning Operations Center performance

AIX Linux Windows

The performance of the Tivoli Storage Manager Operations Center depends on the system resources of the hub and spoke servers and the processor and memory on the computer where the Operations Center is installed. Performance also depends on the number of client nodes and virtual machine file spaces that are being monitored by all servers.

Before you begin

Review the system requirements for the Operations Center before you install it.

About this task

Use the Operations Center System Requirements Calculator at technote 1641684 to estimate the system requirements for running the Operations Center and the hub and spoke servers that are monitored by the Operations Center.

Resource usage on the Operations Center computer

AIX Linux Windows

The number of administrators who are logged in to the Operations Center affects resource usage on the computer where the Operations Center runs.

Resource usage for each logged-in administrator

The number of administrators who are logged in at the same time, and the number of tasks that each is working on, affects the resource usage by the Operations Center. For example, the following situation can be typical:

- Running the Operations Center uses about 3% of a processor core (based on an Intel X7550 2.00 GHz or equivalent processor).
- Each administrator who is logged in to the Operations Center uses 2% - 3% of a processor core on the computer where the Operations Center runs. This usage level assumes that each administrator completes about 40 tasks per hour.
- The Operations Center might have 8 - 10 administrators who are logged in and completing tasks at the same time. A task might be one of the following activities:
  - Opening and viewing a page, window, or wizard in the interface, for example, a storage pool Properties page, a server Quick Look window, an Alert Details window, or the Add Client wizard.
  - Configuring settings on the Settings page
  - Issuing a command from the Operations Center command line

  A typical user might complete 20 - 120 of these tasks per hour.

Memory for the JVM heap size

The Operations Center requires at least 128 MB of memory for the maximum Oracle Java virtual machine (JVM) heap size for up to eight concurrently logged-in administrators. This memory requirement assumes that each administrator completes about 40 tasks per hour within the user interface. The JVM uses a maximum of 1 GB, or 25% of the physical memory of the system (with systems that have more than 192 MB of memory). The heap size starts at a smaller amount but can increase up to the maximum.

If the system that is running the Operations Center has more than 1 GB of memory, no adjustments are needed. If you must adjust the memory for the JVM heap size, use the `-Xmx` option in the `jvm.options` file for the web server JVM. The `jvm.options` file is in the following directory, where *installation_dir* represents the directory where the Operations Center is installed:

- AIX Linux *installation_dir*`/ui/Liberty/usr/servers/guiServer/`
- Windows *installation_dir*`\ui\Liberty\usr\servers\guiServer\`

If the `jvm.options` file is not in the directory, and you must adjust the memory, create the file in the directory.

Effect of the network on performance

AIX Linux Windows

The network that connects the hub server and the system where the Operations Center is installed can affect the performance of the Operations Center.

To achieve better system performance, use one of the following configurations:
- The Operations Center is on the same system as the hub server.
- The Operations Center is on a system that is located physically near the hub server system.

Also, consider facilitating upgrades and maintenance by designating a physical or virtual system that is separate from the production environment as the hub server.

Network latency

Network latency is the time interval between the following operations:
- The initiation of a send operation from a source system
- The completion of the matching receive operation by the target system

Latency between the Operations Center web server and web browsers
For the best responsiveness when logged in to the Operations Center, use a network connection with a round-trip latency that is no greater than 5 ms. This latency can typically be achieved when the systems are on the same local area network (LAN). Higher latencies might be acceptable but can cause degraded responsiveness. For example, the responsiveness across a wide area network (WAN) might not be acceptable to users of the Operations Center.

Latency between the Operations Center web server and the hub server
For the best responsiveness, use a network connection with a round-trip

latency that is no greater than 10 ms. Lowest latency is achieved when both of these servers are installed on the same system or on the same LAN.

Latency between the hub server and spoke servers
Round-trip latency can be high, for example, 100 ms, or low, for example, 10 ms. However, with high latency, Operations Center pages that show details about an individual client, policy domain, server, or storage pool might take a longer time to be displayed. Also, if connection timeout issues occur, you might be required to adjust the **ADMINCOMMTIMEOUT** value on the hub and spoke servers. For information about what to do if connection timeout issues occur, see the following known issue on the IBM Support Portal at technote 1651428.

Estimating the network latency

You can use a **ping** command to estimate the network latency. To use a **ping** command, complete the following steps:

1. From the source system, ping the target system by issuing the following commands, where *remoteHost* represents the address for the target system:
   - AIX Linux `ping –c 20 remoteHost`
   - Windows `ping –n 20 remoteHost`
2. Calculate the average of the intervals for all successful responses from the remote host. This calculation is an estimate of the round-trip latency.

Tip: You can also use more sophisticated network tools that measure network bandwidth and latency between systems. An example is Netperf (http://www.netperf.org/netperf/).

Effect of status monitoring on performance

AIX Linux Windows

When you configure a server as a hub or spoke server, status monitoring is automatically enabled. Status monitoring requires extra resources on each server on which it is enabled.

Tip: In this topic, the term *client* represents both client nodes and virtual machine file spaces. Also, the resource estimates are approximate.

Server resource requirements for status monitoring

The resources that are required depend primarily on the number of clients that are managed by the hub and spoke servers.

Also, the hub server requires fewer resources if the spoke servers are running Tivoli Storage Manager V7.1 or later, than it does if the spoke servers are running V6.3.4 or a later modification of V6.3.

Table 10 on page 116 summarizes the resource requirements for a server on which status monitoring is enabled.

Table 10. Resource requirements for a server on which status monitoring is enabled

Resource requirement	Resource usage for the base level of up to 1000 clients	Resource usage for every 1000 clients over the base level	Example: Resource usage for a spoke server with 2000 clients
Processor usage The value is based on lab measurements that used the Intel X7550 2.00 GHz core.	1.1 processor cores	0.1 processor cores	1.2 processor cores
More space in the server database	2 GB if the server is at V7.1 1 GB if the server is at V7.1.1 or later	2 GB if the server is at V7.1 1 GB if the server is at V7.1.1 or later	4 GB if the server is at V7.1 2 GB if the server is at V7.1.1 or later
More space for the server archive log The value assumes that a full database backup is completed every 24 hours.	10 GB	10 GB	20 GB
Spoke server at V6.3.4 or a later modification of V6.3: Data transfer to the hub server over the network	30 - 60 MB per hour	30 - 60 MB per hour	60 - 120 MB per hour
Spoke server at V7.1 or later: Data transfer to the hub server over the network	5 - 10 MB per hour	5 - 10 MB per hour	10 - 20 MB per hour

Consider adding a buffer of 25% - 50% to the database and log requirements for a server that has a heavy workload. For example:

- A server that is scheduled to back up hundreds of client nodes or virtual-machine file spaces daily
- A server that has many I/O operations per second (IOPS) due to operations such as data deduplication

Extra resource requirements for a hub server

The hub server must have adequate resources for the number of clients that it manages directly, according to the estimates in Table 10. For managing spoke servers, the hub server must also have the extra resources that are described in Table 11 on page 117. Use the table to monitor items that are defined on the hub server and the spoke servers that are connected to it.

Table 11. Extra resource requirements for a hub server.

Resource requirement	For managing spoke servers at V7.1 or later	For managing spoke servers at V6.3.4 or a later modification of V6.3
Processor usage The value is based on lab measurements that used the Intel X7550 2.00 GHz core.	Negligible	More processor resources, equal to 0.1 processor cores for every 1000 clients on all monitored servers (all clients on all spoke servers at V6.3.4 or a later modification of V6.3).
More space in the server database	Negligible	If the hub server is at V7.1: More disk space for the database, equal to 2 GB for every 1000 clients on all monitored spoke servers at V6.3.4 or a later modification of V6.3. If the hub server is at V7.1.1 or later: More disk space for the database, equal to 1 GB for every 1000 clients across all monitored spoke servers at V6.3.4 or a later modification of V6.3.
More space for the server archive log The value assumes that a full database backup is completed every 24 hours.	More disk space for the archive log, equal to 600 MB for every 1000 clients on all monitored spoke servers at V7.1 or later.	More disk space for the archive log, equal to 10 GB for every 1000 clients on all monitored spoke servers at V6.3.4 or a later modification of V6.3.
IOPS capacity for the server database on the hub server	More I/O capability for the database volumes, to support 50 IOPS for every 1000 clients on spoke servers at V7.1 or later. The estimate is based on an average I/O size of 8 KB.	More I/O capability for the database volumes, to support 200 IOPS for every 1000 clients on spoke servers at V6.3.4 or a later modification of V6.3. The estimate is based on an average I/O size of 8 KB.

For a hub server that manages spoke servers, you get optimal performance if the server database is on disks that can process 8 KB operations at the rate of at least 1000 IOPS. To get this IOPS capacity, use a single enterprise-level solid-state drive (SSD). If SSD is not an option, you might want to use a SAN-attached array of 15000-rpm serial-attached SCSI (SAS) hard disk drives, each capable of handling hundreds of 8 KB IOPS. The choice depends on the overall workload of the hub server.

Example of resource requirements for a hub server

Table 12 on page 118 shows a resource estimate for a hub server with spoke servers that are at V6.3.4 or a later modification of V6.3. Table 13 on page 119 shows a resource estimate for a hub server with spoke servers that are at V7.1 or later. In both examples, the hub server has 1000 clients, and each of the five spoke servers has 2000 clients.

Table 12. Example of resource requirements for a hub server with spoke servers that are at V6.3.4 or a later modification of V6.3

Resource requirement	Resource usage for 1000 clients that are managed on the hub server	Resource usage on the hub server for the 10,000 clients that are managed on five spoke servers that are at V6.3.4 or a later modification of V6.3 (2000 clients on each)	Total estimated resource usage
Processor usage The value is based on lab measurements that used the Intel X7550 2.00 GHz core.	1.1 processor cores	1 processor core The estimate is based on 0.1 processor core for every 1000 clients on the spoke servers.	2.1 processor cores
More space in the server database	2 GB if the hub server is at V7.1 1 GB if the hub server is at V7.1.1 or later	20 GB if the hub server is at V7.1 10 GB if the hub server is at V7.1.1 or later	22 GB if the hub server is at V7.1 11 GB if the hub server is at V7.1.1 or later
More space for the server archive log The value assumes that a full database backup is completed every 24 hours.	10 GB	100 GB	110 GB
Spoke server: Data transfer to the hub server over the network	Not applicable	300 - 600 MB per hour The estimate is based on 30 - 60 MB per hour for every 1000 clients on the spoke servers.	300 - 600 MB per hour
IOPS capacity for the server database on the hub server	200 IOPS	2000 IOPS The estimate is based on 200 IOPS for every 1000 clients on the spoke servers.	2200 IOPS

Table 13. Example of resource requirements for a hub server with spoke servers that are at V7.1 or later

Resource requirement	Resource usage for 1000 clients that are managed on the hub server	Resource usage on the hub server for the 10,000 clients that are managed on five spoke servers at V7.1 or later (2000 clients on each)	Total estimated resource usage
Processor usage The value is based on lab measurements that used the Intel X7550 2.00 GHz core.	1.1 processor cores	Negligible	1.1 processor cores
More space in the server database	2 GB if the hub server is at V7.1 1 GB if the hub server is at V7.1.1 or later	Negligible	2 GB if the hub server is at V7.1 1 GB if the hub server is at V7.1.1 or later
More space for the server archive log The value assumes that a full database backup is completed every 24 hours.	10 GB	6 GB The estimate is based on 600 MB for every 1000 clients on the spoke servers.	16 GB
Spoke server: Data transfer to the hub server over the network	Not applicable	50 - 100 MB per hour The estimate is based on 5 - 10 MB per hour for every 1000 clients on the spoke servers.	50 - 100 MB per hour
IOPS capacity for the server database on the hub server	200 IOPS	500 IOPS The estimate is based on 50 IOPS for every 1000 clients on the spoke servers.	700 IOPS Consider establishing a baseline capacity of 1000 IOPS for the hub server database if the hub server manages any spoke servers.

Effect of the status refresh interval on performance

AIX Linux Windows

In the Operations Center, the status refresh interval is the number of minutes between status collection refreshes. Changing the default value of this interval can affect the performance of the Operations Center and the servers that are using status monitoring.

You can set this interval on the Settings page of the Operations Center or by issuing the **SET STATUSREFRESHINTERVAL** command on each hub or spoke server. Use the same interval on the hub and spoke servers. Using different intervals can reduce the accuracy of the information that is shown in the Operations Center.

Effect of decreasing the interval

If you decrease the interval to get more frequent refreshes of collected data, more data is processed and maintained, which uses more space in the server database. More frequent refreshes can also mean higher processor usage. The disks where the server database is located might also require higher input/output operations per second (IOPS).

If you decrease the interval by half, the server database and the archive log space that is required for status monitoring is doubled.

Also, do not decrease the interval to less than 5 minutes. An interval of less than 5 minutes can cause the following issues:

- Operations Center data that is supposed to be refreshed after the defined interval takes a longer time to be refreshed.
- Operations Center data that is supposed to be refreshed almost immediately when a related change occurs in the storage environment also takes a longer time to be refreshed.

Effect of increasing the interval

If you increase the interval to get less frequent refreshes of collected data, resource requirements are reduced. However, the data that is shown in some Operations Center views might not be current.

If you double the interval, the server database and the archive log space that is required for status monitoring is reduced by half.

If a hub server is connected to spoke servers over a network with high latency, consider increasing the interval to reduce the data that is sent over the network for status monitoring.

Chapter 11. Tuning server performance

Many factors must be considered when you are tuning the configuration of your Tivoli Storage Manager server for optimum performance. Review this information to evaluate settings for your operating system, key Tivoli Storage Manager operations, scheduling for the server and client workloads, and configuration for functions that require more server resources.

Server database and recovery log configuration and tuning

How you configure and size the database and recovery logs is essential to Tivoli Storage Manager performance.

Before you begin

The server records changes that are made to the database in the recovery log. The recovery log is used to maintain the database in a transactionally consistent state, and to maintain consistency across server startup operations. The recovery log is composed of an active log, an archive log, and optional logs, including the active log mirror and archive failover log. The following figure shows the Tivoli Storage Manager server, database, and recovery log in relation to each other.

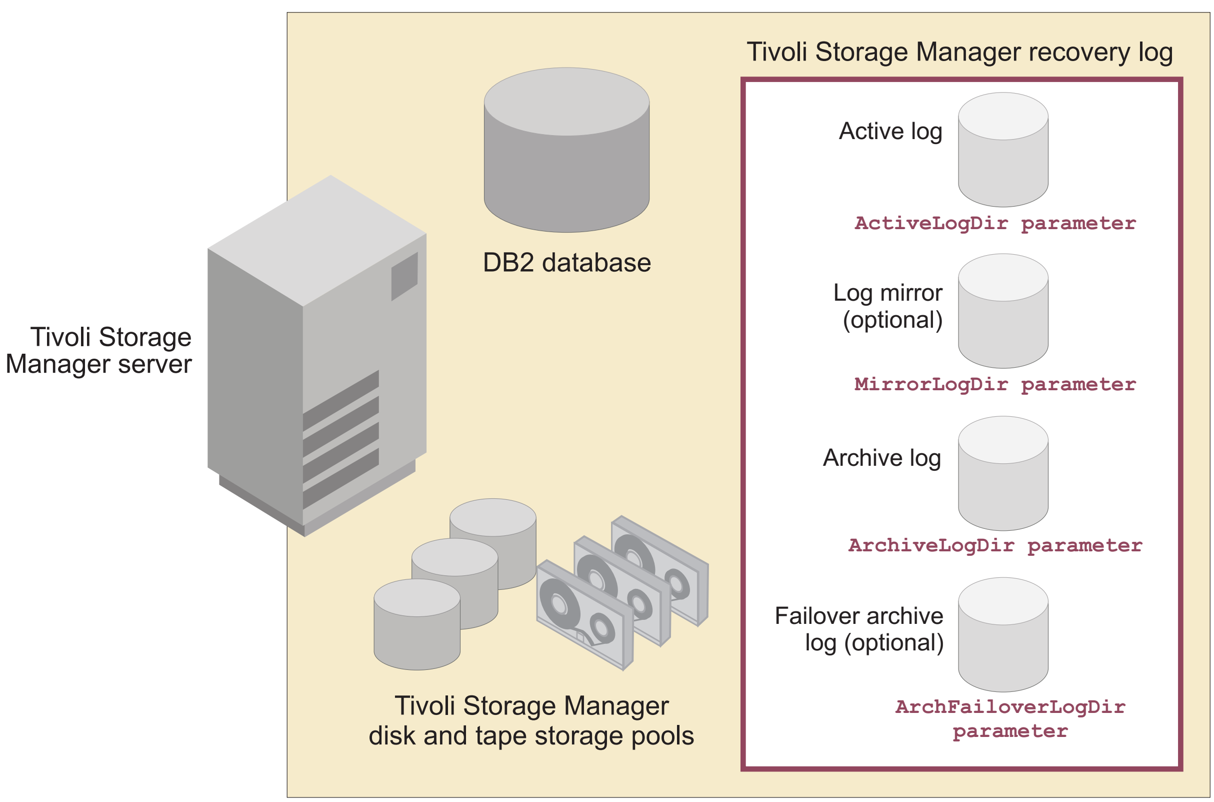

Figure 19. Components of the Tivoli Storage Manager database and recovery log

Database configuration and tuning

The Tivoli Storage Manager database contains information that is needed for server operations and information about client data. Ensure that you configure adequately for your space requirements and plan ahead for future database growth.

Before you begin

Tip: Complete the "Checklist for server database disks" on page 17 to verify that your server database hardware is configured for good performance.

Configuring the server database directories for optimal performance

The *database containers* are the directories that DB2 uses to store the server database. Overall database size and the amount of data that is handled on a daily basis are the most important factors for how you configure the database directories.

Plan for growth with more directories at the start

When you are setting up your database space, ensure that you configure for future growth by including extra directories at the outset. If you are starting out with a moderate amount of data but plan to manage more, make sure that you plan for the larger amount of data to avoid reconfiguration work later.

If you must add directories after your initial configuration, make sure that you create directories that are equally sized. You can use the **EXTEND DBSPACE** command to add new directories for the database to use.

Plan ahead when you want to add space to the database. After new directories are added, data is redistributed and space is reclaimed for the system. This process can take considerable server resources. For more information, see the **EXTEND DBSPACE** command.

Use multiple database directories

How you spread the database directories across available disk storage has a strong effect on performance. Follow these guidelines for the database directories that you use:

- Use at least four directories initially for the database, spread across four LUNs or physical disks. For large Tivoli Storage Manager servers, use eight directories or more. You can use up to 128 directories for the server database.

  For 2 TB servers for which data deduplication is planned, use eight directories or more for the server database. With data deduplication, the load on the database becomes greater because there are frequent queries to the database to determine what deduplicated extents are on the server.
- Make all directories that are used for the database the same size to ensure parallelism.
- Place each database directory in a different file system. This placement improves performance because DB2 stripes the database data across the various directories.
- Place the directories on disks that have the same capacity and I/O characteristics. For example, do not mix 10000 rpm and 15000 rpm drives for the database directories.

- For most disk systems, performance is best if one database directory is on one LUN, which has one logical volume.

The following image illustrates how to distribute database directories for data deduplication, by using eight disks.

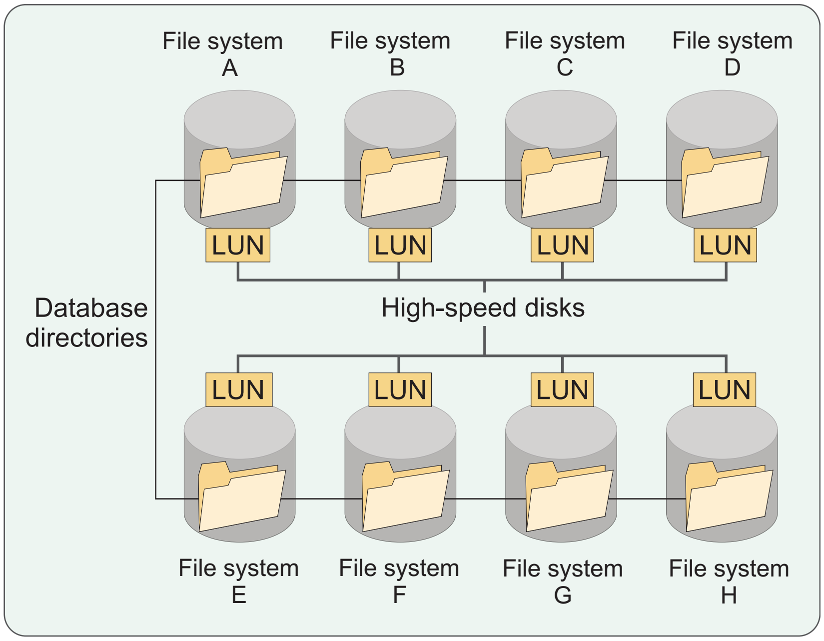

Figure 20. Distribution of Tivoli Storage Manager database directories

Reorganizing database tables and indexes

Server performance can degrade when database tables and table indexes are not reorganized.

About this task

Over time, database information becomes fragmented, causing unexpected database growth and increased space requirements for the active and archive logs. When you reorganize tables and table indexes, it reduces fragmentation and reclaims space.

If you are not running data deduplication on the Tivoli Storage Manager server, you might have to reorganize only the tables. However, you must monitor the database growth to avoid server degradation. If you are running data deduplication, reorganize both tables and indexes for best results.

You can reorganize tables and indexes online when the Tivoli Storage Manager server is running, or offline when the server is halted. Depending on your server workloads, you might have to disable both table and index reorganization to maintain server stability, and to complete daily server activities. If you experience unacceptable database growth or server performance degradation when reorganization is disabled, reorganize the tables offline.

Tivoli Storage Manager reorganizes tables and indexes by default. Allow server-initiated reorganization of both tables and indexes to run. If automatic reorganization is affecting server performance, you can manually schedule reorganization.

Improving the speed of database backups

You can use multiple, concurrent data streams for database backups, which reduce the time that is required for a backup or restore operation to complete.

About this task

The amount of time that you can save depends on the size of the database. In general, the best performance benefits occur when you update your configuration from one to two streams and when the database is more than 100 GB.

A potential disadvantage when you use multistream processing is that more volumes and drives are required than for single-stream processing. For example, if the backup of an 850 GB database requires a single Linear Tape-Open (LTO) volume, switching to four data streams requires four drives. Volumes might be partially filled, especially if you use high-capacity volumes and device compression. If a backup of the database uses only 30% of a tape volume after compression, then four-stream processing results in even larger amounts of wasted space.

In some cases, particularly on AIX, problems with the TCP loopback interface can cause database backups to be unacceptably slow. Try configuring shared memory for database backups to see whether the speed of the database backup improves.

Configuring and tuning the recovery log

The management of the recovery log is important to normal operations for the server.

Before you begin

Tip: Complete the checklist for server recovery log disks so that the disk systems that are being used for the logs are optimally configured. Then, review the following information about how to configure the recovery log.

Procedure

- For the active log and active log mirror, follow these guidelines:
  - Ensure that the directory that holds the active log is as large as, or larger than, the size of the active log. A directory that is larger than the active log can accommodate failovers if they occur.

    Creating an active log mirror provides higher reliability, but comes at a cost. When the log is mirrored, it can affect performance because of the doubled I/O activity that is required to maintain the mirror. The additional space that the log mirror requires is another factor to consider. If you are using separate physical disks for the active log, use the **MIRRORLOGDIR** parameter in the **DSMSERV FORMAT** command to find the mirror log directory. After you install the server, change the mirror log directory location by changing the value of the **MIRRORLOGDIR** option in the server options file and restarting the server.
  - Use disk system read-ahead processing to archive active logs more quickly.
  - If there are active log performance issues, you can set the `LOGBUFSZ` server option in the `dsmserv.opt` file to 256 and restart the server.
- For the archive log and failover archive log, follow these practices:
  - You can create an archive failover log to store archive log files when the archive log directory is full. For best results, maintain an archive failover log.

- Do not share archive log space with other applications, including other Tivoli Storage Manager servers. Other applications can write to the archive log directory and use the space that is required by the archive log. Ensure that each server has a separate storage location that is owned and managed by that specific server.
- Set aside space in the failover archive log for possible emergency use.
- Compress the archive log to reduce the need to run full database backups to clear the archive log. By compressing archived logs, you can reduce the amount of disk space that is required for the database environment. In high performance environments, such as large blueprint configurations, failure to compress archive logs can cause bottlenecks and performance degradation.

Related reference:

"Optimization of disk system read-ahead processes" on page 161

Recovery log sizing

Ensure that the size of the active and archive logs meets the requirements of the server workload.

The active log is limited to a maximum size of 128 GB. The archive log is limited in size only by the size of the file system where it is located. It is better to create the active and archive log directories too large as opposed to too small. Consider the minimum values in the following table. Using these values or higher values can prevent log space problems for a server.

Storage pool deduplication enabled?	Active log directory: Minimum size	Archive log directory: Minimum size
No	16 GB	48 GB
Yes	Use the maximum size of 128 GB	128 GB

For more information about sizing the recovery log and examples that are based on different server loads, see Upgrading the server.

Related reference:

"Checklist for server recovery log disks" on page 19

Tuning and configuring storage pools and volumes

Logical storage pools and storage volumes are the principal components in the Tivoli Storage Manager model of data storage. By manipulating the properties of these objects, you can optimize the use of storage devices.

Compressing data to save storage space

You can use server-side data compression to increase the amount of available space in a storage pool.

About this task

Review the following table to compare client-side and server-side compression:

Type of compression	Advantages	Disadvantages
Client-side	• Reduced load on the network • Reduces data that is stored in the storage pool	• Higher processor usage by the client • Potentially longer elapsed time for client operations such as backup
Server-side	• Uses inline compression to compress data as it is written to a container storage pool • Reduces the amount of space that is required to store the data • Does not affect on client operations such as backup	• No load reduction on the network • Higher processor usage by the server

Procedure

Inline compression is enabled by default. To disable compression, issue the **UPDATE STGPOOL** command and specify the **COMPRESSION=NO** parameter.

Related concepts:

"Reduce client data flow with compression" on page 198

Optimizing data organization for restore and retrieval operations

How you organize data affects how efficiently and quickly Tivoli Storage Manager can access it and perform retrieval operations.

Grouping data by using collocation in server storage pools

Use collocation to improve Tivoli Storage Manager performance and maintain optimal data organization.

Before you begin

Tip: The following information does not apply to container storage pools.

When you use collocation, the performance of restore operations for large amounts of data can be significantly improved because fewer mounts are required to locate the necessary files. Generation of backup sets and export operations are also faster. In addition, collocation decreases the chance for media contention with other clients. While performance is improved by using collocation, enabling it increases both the amount of server time that is needed to collocate files for storing and the number of volumes that are required for data storage.

You can enable collocation by node, group, or file space. Collocation by group is the default. Each option provides different benefits and considerations for performance.

Table 14. Collocation trade-offs

Type	Volume usage	Volume mounts	Restore time
No collocation	Low volume usage	Few number of mounts for migration and reclamation	Longest restore time
Collocated by node	High volume usage	High number of mounts for migration and reclamation	Good restore time, but not optimized for multi-session restore
Collocated by group	Low volume usage	Few mounts for migration and reclamation	Good restore time
Collocated by file space	High volume usage	High number of mounts for migration and reclamation	Good restore time, but not optimized for multi-session restore

About this task

Consider the following information when you are determining which type of collocation you want to use:

- Collocation by group provides the best balance of restore performance versus tape volume efficiency and it is the best practice choice for most situations. Collocation by group results in a reduction in unused tape capacity, which allows more collocated data on individual tapes. If collocation is needed to improve restore performance, use collocation by group. Manage the number of nodes in the groups so that backup data for the entire group is spread over a manageable number of volumes.
- For primary storage pools on tape, use collocation by group:
  - To get the full benefit of collocation by group, you must define the collocation groups and their nodes.
  - Nodes that are not grouped are collocated by node.
- For nodes with two or more large file spaces that might get close to filling a tape volume, use collocation by file space.
- Use an active data pool to collocate active data.
- Group nodes that have a low chance of being restored at the same time to avoid volume contention.
- Group nodes that are backed up to disk at the same time.

To enable collocation, use the **COLLOCATE** parameter on the **DEFINE STGPOOL** command when you are defining a primary sequential-access, copy, or active-data storage pool. You can use the **UPDATE STGPOOL** command to enable collocation for an existing storage pool.

Maintaining active-data storage pools

Setting up active-data storage pools can be useful for fast client restore operations. By keeping only active data in a storage pool, you can reduce the number of onsite or offsite storage volumes that you use, or reduce bandwidth when you copy or restore files that are vaulted electronically in a remote location.

Before you begin

One of the main benefits of using active-data storage pools is that it requires less disk space because the active-data pool contains only active file versions. Inactive data can be moved to tape.

The benefits of an active-data pool depend on the specific device type that is associated with the pool. For example, active-data pools that are associated with a FILE device class are ideal for fast client restore operations for the following reasons:

- FILE volumes do not have to be physically mounted
- The server does not position past inactive files that do not have to be restored

In addition, client sessions that are restoring from FILE volumes in an active-data pool can access the volumes concurrently, which also improves restore performance.

Active-data pools that use removable media, such as tape or optical, offer similar benefits. Although tapes must be mounted, the server does not have to position past inactive files. However, the primary benefit of using removable media in active-data pools is the reduction of the number of volumes that are used for onsite and offsite storage. If you vault data electronically to a remote location, you can save bandwidth by copying and restoring only active data. Use an active-data pool that is associated with a SERVER device class to copy and restore data when you want to save bandwidth.

For optimal efficiency during point-in-time restores and to avoid switching between active-data pools and primary or copy storage pools, the server retrieves both active and inactive versions from the same storage pool and volumes.

Improving file restore performance with caching in disk storage pools

You can use caching for Tivoli Storage Manager random-access (DISK) storage pools to improve restore performance of files.

When you enable caching for a storage pool, the migration process leaves a cached copy of a file in the storage pool after migrating files to the next storage pool. If your disk storage pool is large enough to hold the backed-up data from one day, then caching is a good method. When you restore files that were recently backed up to a disk storage pool, the benefit of caching is evident. The time that is required to retrieve files is reduced.

Trade-offs for the use of caching include the following performance impacts:

- Disk caching can affect backup throughput. When cached files must be deleted to create space for file backups, the deletion operations require database updates.
- If you use caching with larger storage pools, they might become fragmented and response time can increase.

Use the **DEFINE STGPOOL** or **UPDATE STGPOOL** command with the `CACHE=YES` parameter to enable caching. If you enable disk storage pool caching, and then suspect that it is affecting performance, try turning off caching. To turn off caching, issue the **UPDATE STGPOOL** command with `CACHE=NO` to see whether there is a positive effect.

Using file system cache for storage pools

By default, the Tivoli Storage Manager server reads and writes storage pool data with non-buffered I/O, which bypasses the file system cache. Using file system cache can be beneficial in certain situations but it is not a best practice.

About this task

If you switch to file system cache, change the default with care. The use of file system cache can decrease overall server throughput and increase server processor usage. Use file system cache for storage pool I/O only if it results in a significant performance improvement.

You might see benefits from using file system cache in the following environments:

- A disk storage system that you are using for storage pools has a relatively small cache memory size.
- The disk storage system does not provide read-ahead capability. You must improve storage pool migration throughput from the disk volumes by using the operating system read-ahead capability.
- Data that is stored on the Tivoli Storage Manager server is expected to be immediately retrieved through some other process.

Restriction: Enabling the server to use file system cache is not a preferred practice. If you contact IBM Software Support for a performance issue, you must disclose that you are using this option. You might be asked to disable this option before IBM can resolve the issue.

Procedure

To use file system cache for storage pool I/O, complete the following steps:

1. Specify one of the following options in the `dsmserv.opt` file:
   - Container storage pools: `DIOENABLED NO`
   - Other types of storage pools: `DIRECTIO NO`

   Tips: For better performance, make the following adjustments:
   - Disable direct I/O for IBM Spectrum Scale™
   - Enable IBM Elastic Storage™ Server
2. Restart the server.
3. Observe operations for the effects of the change and determine whether to keep the `DIRECTIO NO` entry in the options file or remove it.

   Tip: To display the current value for the **`DIRECTIO`** option, specify the option name when you issue the **`QUERY OPTION`** command.

File system fragmentation for storage pools on disk

Fragmentation problems can occur when you are writing to scratch volumes in multiple storage pools simultaneously, in one file system. Because unfragmented volumes generally perform better than fragmented volumes, use the **DEFINE VOLUME** command to preallocate volumes for sequential disk storage pools (**DEVTYPE** is FILE).

In some operating system environments, preallocating several FILE-device class volumes or random-disk pool volumes in parallel in the same file system can also result in fragmentation. If you are running a Windows system or using JFS2 file systems on AIX or ext4 file systems on Linux, fragmentation is not a problem with preallocated volumes. Tivoli Storage Manager uses operating system tools to allocate files without fragmentation, even when created in parallel.

If you encounter slow performance, you can use operating system tools to help determine how fragmented a storage pool volume file might be. The tools report the number of fragments per file. To achieve high read performance from a volume, the number of fragments per megabyte should not exceed 0.25. For example, if a 50 GB volume has more than 12,800 fragments, then read performance for the volume might be poor because of fragmentation.

Use the following tools to determine fragmentation level:
- On AIX, issue the **fileplace** command.
- On Linux, issue the **filefrag** command.
- On Windows, use the **contig** utility. You can find details and usage information about the **contig** utility on the Microsoft TechNet site.

To avoid fragmentation, use preallocated volumes and use more file systems per storage pool. By using a greater number of file systems, you can usually avoid fragmentation because writing to different volumes is more likely to occur in different file systems.

Optimal number and size of volumes for storage pools that use disk

When you calculate the size and number of volumes for storage pools that are defined with a FILE or DISK device class, consider the type of storage, how much storage is available, and other variables.

The optimal size for storage pool volumes depends on these elements:
- The type of storage pool, whether random access (DISK) or sequential access (FILE)
- How much total storage is available
- How many concurrent writes are expected (client sessions and server processes)
- How many storage pool volumes can be managed
- The storage pool collocation settings
- Whether you use preallocated or scratch volumes
- The average object size that is stored in the storage pool (having the volume size larger than the average object)

If you do not have the information to estimate a size for FILE device class volumes, start with volumes that are 50 GB.

The FILE device class allows only one session or process to write to a volume at a time. Therefore, you must have at least as many volumes as you have sessions and processes that you expect to write to the pool at the same time.

If you are using collocation, a minimum number of volumes is required. For example, if you are running collocation by node, use at least one volume per node. When you use collocation with preallocated volumes, consider the potential unused storage within those volumes. Collocation with preallocated volumes requires more space. If you use collocation by group, you can reduce the amount of unused storage by designing groups to minimize this waste.

You can use preallocated volumes to avoid file system fragmentation, but you must estimate how much space is required for the storage pool and allocate enough volumes to handle that load. This estimate assumes that the amount of data you are backing up does not fluctuate. For example, processing that is done at the end of the month might use more storage pool space than regular daily processing. If you use scratch volumes, fluctuation is not a factor because Tivoli Storage Manager allocates what is needed for each day's processing. If you are using preallocated volumes at a recovery site, some additional time is required for the recovery process because volumes must be preallocated and formatted.

Example: Choosing volume size for a sequential file storage pool

In this example, you must determine a volume size for a sequential-access FILE storage pool with 100 TB of available space. Collocation is not used, therefore you do not have to consider the empty space that is required in preallocated volumes. The maximum number of simultaneous backup sessions that are used during the backup window is 250. To prepare for growth of the system and the occurrences where 250 sessions are exceeded, double that amount to 500. Using these values, the maximum volume size that is needed is 204 GB.

Related tasks:

"Grouping data by using collocation in server storage pools" on page 126

Configuring and tuning the server

How you configure and tune different Tivoli Storage Manager server settings and operations can affect performance of the system.

Memory requirements

You must have sufficient memory available for server operations such as replication and data deduplication. Memory requirements are based on the projected maximum database size. Extra memory improves database query performance by increasing the amount of memory that is available for the database buffer pools.

Use the following table to determine the minimum amount of memory you must allocate on the server to complete operations, based on the database size. The size that is required depends on the functions that you use.

Table 15. Minimum size requirements based on database size

Database size	Standard server operations (no data deduplication or replication required)	Either data deduplication or replication required	Both data deduplication and replication required
500 GB	16 GB	24 GB	32 GB
1000 GB	24 GB	40 GB	56 GB
1500 GB	32 GB	56 GB	80 GB
2000 GB	40 GB	72 GB	104 GB
2500 GB	48 GB	88 GB	128 GB
3000 GB	56 GB	104 GB	152 GB
3500 GB	64 GB	120 GB	176 GB
4000 GB	72 GB	136 GB	200 GB

Tuning the schedule for daily operations

Typically, backup operations must be completed every day for all clients. Certain server maintenance processes must also run every day. Ensuring that resources for these critical operations are available when needed requires planning and tuning.

About this task

During the client workload phase, server resources support client operations. Client workloads are predominantly client backup and archive activities. Typically, these operations are completed during the nightly schedule window. During the server workload phase, server resources are dedicated to managing the recently received data from the client workload and performing the following activities, which are necessary to manage, protect, and maintain the server:

- Protecting client data by backing up storage pools
- Allocating data appropriately in the storage hierarchy
- Keeping the database, storage hierarchy, and server operations working efficiently
- Preparing for the next schedule cycle

Carefully schedule client and server workloads to achieve the best performance for your environment. If the client and server operations overlap or are not given enough time and resources to be processed, the environment might be impacted in the following ways:

- Less processing power and memory available to support an operation
- Performance degradation
- Insufficient space for data storage
- Issues with data placement
- Failed operations

For optimum performance, split the tasks of backing up and archiving client data, and performing server data maintenance into separate time windows. Most operations on the server have an optimal order, and in some cases must be completed without overlap to avoid resource contention problems.

What to do next

In addition to tasks that are completed in all Tivoli Storage Manager environments, you might need to plan for optional processes.

Daily operations for directory-container storage pools

Schedule daily operations for the server depending on the type of storage pool that you use. You can complete specific tasks with directory-container storage pools.

About this task

The following image illustrates how Tivoli Storage Manager tasks fit into the daily schedule.

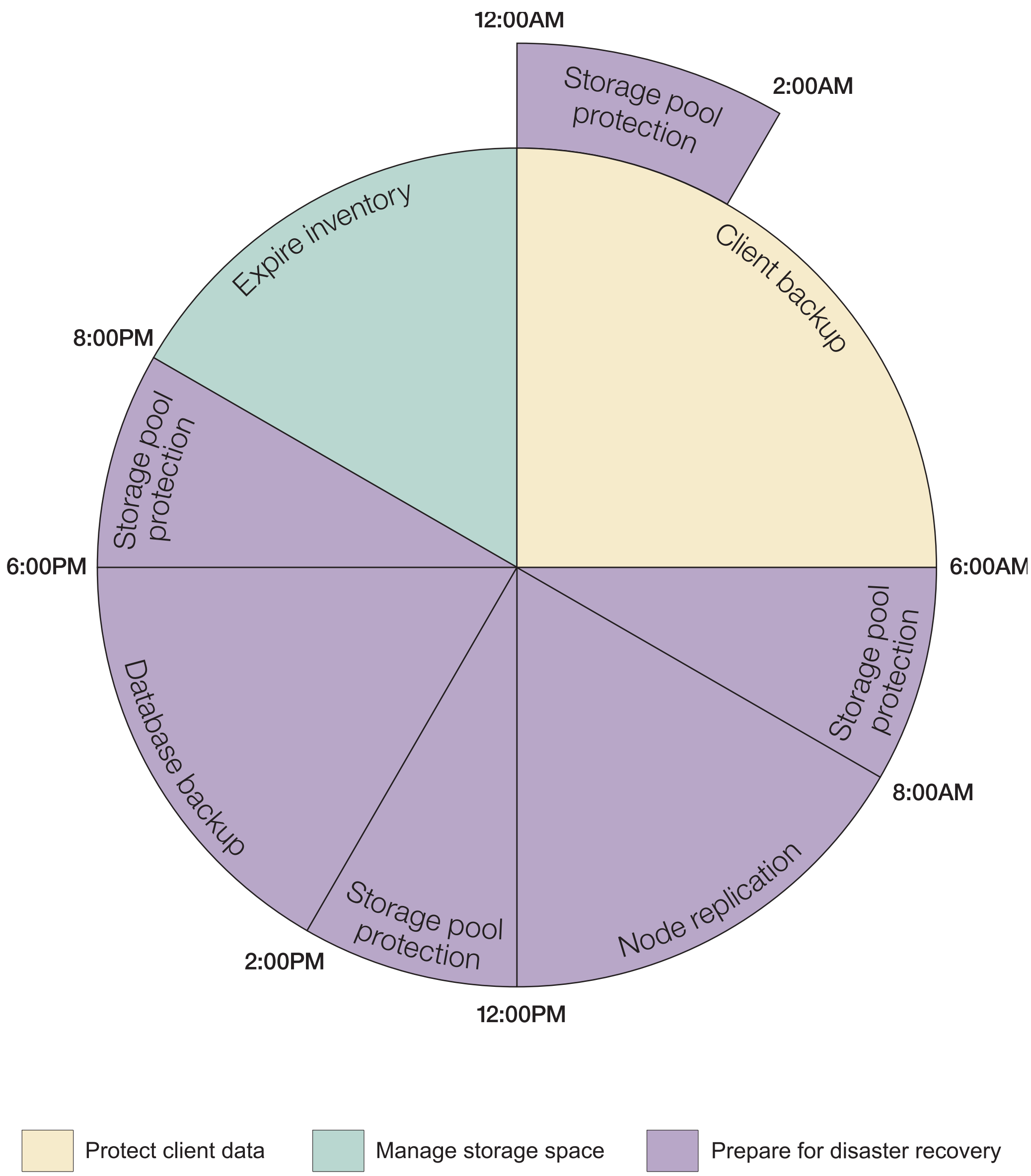

Figure 21. Daily schedule of operations for directory-container storage pools

You can schedule daily activities for Tivoli Storage Manager by using the Operations Center. The Operations Center creates the storage pool protection schedules when you use the wizards to configure replication or add a directory-container storage pool. You can also use the Operations Center to schedule client backups.

To manually create a schedule for daily operations, use the **DEFINE SCHEDULE** command.

Procedure

1. Perform an incremental backup of all clients on the network by using the **incremental** client command or use another supported method for client backup operations.
2. Create a DR copy of the Tivoli Storage Manager database by using the **BACKUP DB** command.
3. Protect data in directory-container storage pools to reduce node replication time by using the **PROTECT STGPOOL** command. Protect storage pools at regular intervals during the daily schedule.
4. Perform node replication to create a secondary copy of the client data on another Tivoli Storage Manager server by using the **REPLICATE NODE** command.
5. Remove objects that exceed their allowed retention period by using the **EXPIRE INVENTORY** command.

Daily operations for storage pools on FILE and DISK devices

Schedule daily operations for the server depending on the type of storage pools you use. You can complete specific tasks with storage pools on FILE and DISK devices.

About this task

The following image illustrates how Tivoli Storage Manager operations fit into the daily schedule.

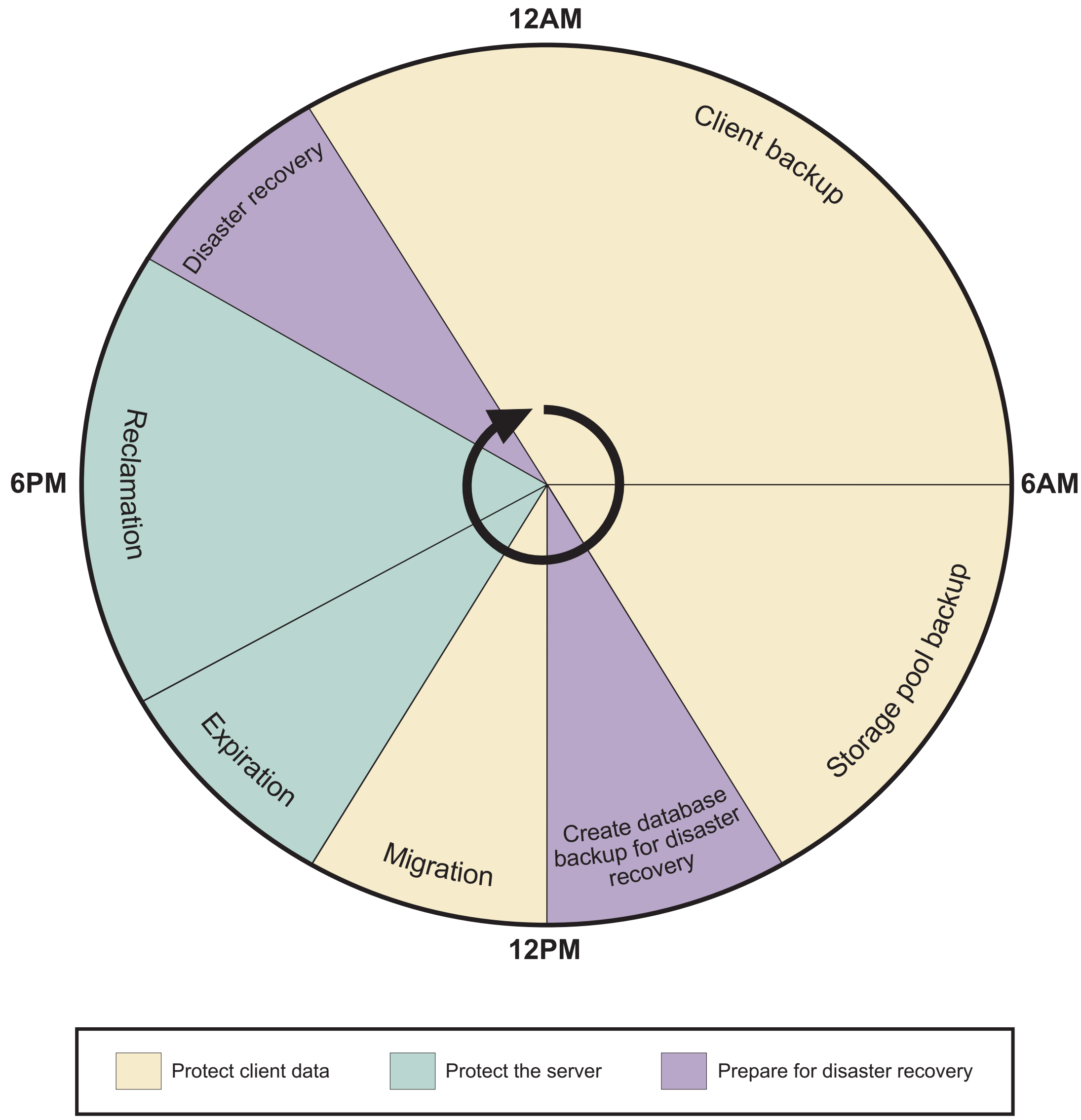

Figure 22. Daily schedule of server operations for storage pools on FILE or DISK devices

To follow best practices, schedule daily operations for Tivoli Storage Manager. Sample commands are provided to implement each step. All listed commands are server commands unless otherwise noted.

Procedure

1. Perform an incremental backup of all clients on the network by using the **incremental** client command or use another supported method for client backup operations.
2. Create a secondary disaster recovery (DR) copy of your client data by using the **BACKUP STGPOOL** command. If you are copying active data, complete this operation during the storage pool backup window.
3. Create a DR copy of the Tivoli Storage Manager database by using the **BACKUP DB** command. In addition, use the **BACKUP VOLHISTORY** and **BACKUP DEVCONFIG** commands to create DR copies of the volume history and device configuration files.

4. Migrate data from disk storage pools to tape storage pools with the **MIGRATE STGPOOL** command.
5. Remove objects that exceed their allowed retention period by using the **EXPIRE INVENTORY** command.
6. Reclaim unused space from storage pool volumes that are released through processes like data deduplication and inventory expiration by using the **RECLAIM STGPOOL** command.
7. Complete disaster recovery preparations. For example, if you are using the Tivoli Storage Manager disaster recovery manager function (DRM), issue the following commands:
   - **DELETE VOLHISTORY** to remove older versions of database backups, which are no longer required.
   - **MOVE DRMEDIA** to track database backup and copy storage pool volumes that are to be moved offsite and to identify the expired or empty volumes that are to be moved onsite.
   - **PREPARE** to create a recovery plan file.

Scheduling data deduplication and node replication processes

Data deduplication and node replication are optional functions that can be used with Tivoli Storage Manager. They provide added benefits but also require additional resources and consideration for the daily schedule.

About this task

Depending on your environment, using data deduplication and node replication can change the tasks that are required for the daily schedule. If you are using node replication to create the backup copy of your data, then storage pool backups are not needed. Likewise, you do not need to migrate your data to tape storage pools for the creation of offsite backup media.

The following image illustrates how to schedule data deduplication and node replication processes to achieve the best performance. Tasks that overlap in the image can be run at the same time.

Restriction: The amount of duplicate identification processes that can be overlapped is based on the processor capability of the Tivoli Storage Manager server and the I/O capability of the storage pool disk.

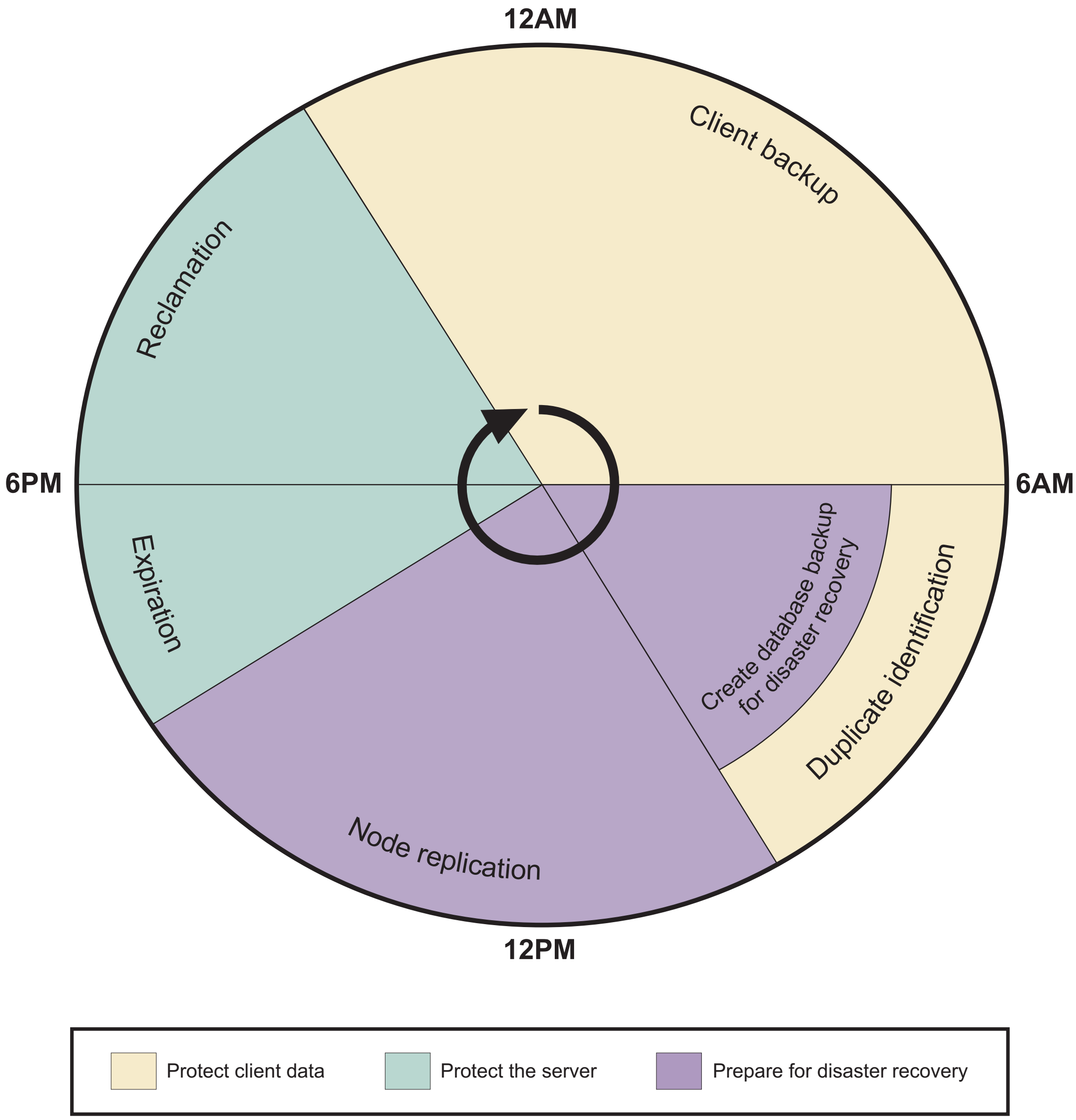

Figure 23. Daily schedule when data deduplication and node replication are used

The following steps include commands to implement the schedule that is shown in the image. For this example, tape is not used in the environment.

Procedure

1. Perform an incremental backup of all clients on the network to a deduplicated file storage pool by using the **incremental** client command or use another supported method for client backup.
2. You can run the following tasks in parallel:
   a. Perform server-side duplicate identification by running the **IDENTIFY DUPLICATES** command. If you are not using client-side data deduplication, this step processes data that was not already deduplicated on your clients.
   b. Create a disaster recovery (DR) copy of the Tivoli Storage Manager database by running the **BACKUP DB** command. In addition, run the **BACKUP VOLHISTORY** and **BACKUP DEVCONFIG** commands to create DR copies of the volume history and device configuration files.

3. Perform node replication to create a secondary copy of the client data on another Tivoli Storage Manager server by using the **REPLICATE NODE** command. By performing node replication after duplicate identification processing, you can take advantage of data reduction during replication.
4. Remove objects that exceed their allowed retention by using the **EXPIRE INVENTORY** command.
5. Reclaim unused space from storage pool volumes that are released through data deduplication and inventory expiration by using the **RECLAIM STGPOOL** command.

Related concepts:

"Checklist for data deduplication" on page 30

"Checklist for node replication" on page 35

Compatibility and resource usage for server processes

Review this information about resource requirements and compatibility issues to help plan your daily schedule and run processes in the optimal order.

The table contains the following information for server tasks:

Process
: Lists the process or operation that is performed by the Tivoli Storage Manager server.

Requirements and recommendations
: Lists any requirements that must be met before a process can be performed. Best practice information is also covered where it is applicable.

Compatibility issues
: Lists any compatibility issues that might arise when you are running processes together.

Prerequisite tasks
: Lists tasks that must be completed before the process is performed.

Resource implications
: Lists resources that are required to run the process and provides guidance on how much usage can be expected:

  Low Resource usage is low. Running the process does not affect other operations.

  Moderate
  Resource usage is moderate. Running the process might affect other operations.

  High Resource usage is high. Dedicate the resource to running the process until it completes.

  Tip: Mount points and volumes are used for most server processes. Because the use of these resources is highly variable, depending on environment configuration, the table does not include a usage designation.

  For operations that use file mount points with a device class of type FILE, set the mount limit parameter of the device class high enough to accommodate all simultaneous mounts. For example, the number of parallel backup sessions for a database backup are typically not more than 5, but for client backup, the mount point requirement can be in the range of 500 - 1000.

For operations that use physical tape mounts, the mount points are limited by the number of actual tape drives. When you are backing up storage pools to tape, plan to use parallel storage pool backup processes that do not exceed the number of available tape drives, and possibly leave some drives unused to remain available for client restores.

Table 16. Server process requirements

Process	Requirements and recommendations	Compatibility issues	Prerequisite tasks	Resource implications
Backing up the database	None	None	Backing up storage pools	– Mount points and volumes – Processor (low) – Memory (low) – Disk or tape I/O (moderate) – Database I/O (high)
Backing up or archiving client data	Requirement: Define and configure client nodes within the Tivoli Storage Manager server. Recommendation: Back up storage pools immediately after the main client backup or archive operation is finished to ensure that a complete copy is created for the primary storage pool.	Expiring inventory Running inventory expiration while you are backing up clients can cause resource contention problems. If expiration is processing a node that is being backed up, performance degradation is usually the result. Backing up storage pools Wait for client backups to finish before you start a storage pool backup. Otherwise, the storage pool backup copy does not include the entire client backup.	None	– Mount points and volumes – Locking (moderate) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate) – Network (moderate to high)
Backing up storage pool	Requirement: Store new data in the primary storage pool.	None	Backing up client data	– Mount points and volumes – Locking (low) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)
Copying active data	Requirement: Store new active data in the primary storage pool.	None	Backing up client data	– Mount points and volumes – Locking (low) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)

Table 16. Server process requirements (continued)

Process	Requirements and recommendations	Compatibility issues	Prerequisite tasks	Resource implications
Expiring inventory	Requirement: Deactivated data must exist on the server. Recommendation: Run inventory expiration within its own processing window as much as possible. In addition, run inventory expiration before the reclamation process to ensure that the process reclaims as much space as possible, considering policy definitions.	Backing up client data Expiring inventory while you are backing up clients can cause resource contention problems. If expiration is processing a node that is being backed up, performance degradation is usually the result.	None	– Locking (high) – Processor (high) – Memory (moderate) – Database I/O (high)
Generating backup sets	Requirement: Store data in at least one primary storage pool.	None	None	– Mount points and volumes – Locking (low) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)
Identifying duplicates	Requirement: Store new data that is not deduplicated from client-side deduplication in a primary storage pool that is enabled for server-side deduplication. Recommendation: Run duplicate identification before reclamation (as much as possible).	None	Potential prerequisite: If you are backing up storage pools, the process might not run at optimal speed against objects that are already identified. In heavy deduplication environments, it can be beneficial to back up storage pools before you run duplicate identification.	– Mount points and volumes – Locking (moderate) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)

Table 16. Server process requirements (continued)

Process	Requirements and recommendations	Compatibility issues	Prerequisite tasks	Resource implications
Migrating storage pools	Requirement: Store data in at least one primary storage pool.	None	Potential prerequisite: If data deduplication is being used in the storage pool that is being migrated, and the target storage pool is deduplicated, run duplicate identification before you move or migrate that data.	– Mount points and volumes – Locking (high) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)
Moving data	Requirement: Store data in at least one primary storage pool.	None	Potential prerequisite: If data deduplication is being used in the storage pool that is being migrated, and the target storage pool is deduplicated, run duplicate identification before you move or migrate that data.	– Mount points and volumes – Locking (high) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)
Moving data by node	Requirement: Store data in at least one primary storage pool.	None	Potential prerequisite: If data deduplication is being used in the storage pool that is being migrated, and the target storage pool is deduplicated, run duplicate identification before you move or migrate that data.	– Mount points and volumes – Locking (high) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)
Reclaiming volumes in an onsite storage pool	Requirement: Store data on storage pool volumes that are expired. In addition, put data on storage pool volumes that are identified as duplicated (through the identify duplicates process).	None	Expire inventory before you reclaim volumes in an onsite storage pool. Potential prerequisite: If deduplication is being used for the storage pool that is being reclaimed, complete duplicate identification and a storage pool backup before deduplicating data.	– Mount points and volumes – Locking (high) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)

Table 16. Server process requirements (continued)

Process	Requirements and recommendations	Compatibility issues	Prerequisite tasks	Resource implications
Reclaiming volumes in an offsite storage pool	Requirement: Store data on storage pool volumes that are expired. In addition, data is on storage pool volumes that are identified as duplicated (through the identify duplicates process). The data must be in a copy storage pool that is flagged as offsite.	None	Expire inventory before you reclaim volumes in an offsite storage pool. Potential prerequisite: If deduplication is being used for the storage pool that is being reclaimed, complete duplicate identification and a storage pool backup before deduplicating data.	– Mount points and volumes – Locking (high) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate)
Replicating nodes	Requirement: Store data in at least the primary storage pools and define and prepare a target server for replication. Recommendation: If you are using data deduplication for the replication process, run identify duplicates to completion in the primary storage pools before you run replication. This recommendation can be ignored if you are using client-side data deduplication for your entire environment.	None	Back up client data before you replicate nodes Potential prerequisite: If the replication process relies on data that is being deduplicated, run duplicate identification against all data that is being replicated.	– Mount points and volumes – Locking (moderate) – Processor (moderate) – Memory (moderate) – Disk or tape I/O (moderate) – Database I/O (moderate) – Network (moderate to high)

Avoiding contention for server resources during client operations

Tivoli Storage Manager requires many resources to complete client backup, archive, or hierarchical storage management migration operations. If multiple sessions are in contention for server resources, system performance can be affected.

Server resources that are used during client operations include database logs, server locks, drives, mount points, or volumes. For example, a client backup session might need a mount point, tape drive, and tape volume on which to store data. After these resources are allocated to the backup session, another client session to restore data that is on the tape volume might start. The restore session is delayed until the backup session unmounts the volume and releases it.

Resource contention has a direct effect on performance and the ability to complete an operation in a timely manner. The problem of resource contention is more critical with long-running client sessions or server processes. Because new database entries are stored in the recovery log until they are committed to the database, long-running sessions or processes can insert many of these entries under a single transaction, pinning the recovery log. A pinned recovery log prevents all transactions from applying database changes and causes server processes to run slowly.

You can schedule client sessions and server processes at different times to avoid resource contention and delays. When you are setting up schedules, you might need to prevent some server processes from starting automatically. For example, disable expiration, migration, reclamation, and duplicate-identification so that they can be scheduled later. Use administrator-command schedules to run these operations daily.

Related reference:

"Compatibility and resource usage for server processes" on page 138

Disabling automatic processes and setting schedules

Disable automatic processes like inventory expiration, migration, reclamation, and identification of duplicate data and set up schedules so that you can control when these operations are completed during the daily schedule.

About this task

Review the following sections for examples of disabling automatic operations and setting up schedules. Some general setup steps are omitted and the example uses a particular order for the server processes. However, you can reorder the processes to better fit your environment.

Procedure

1. Disable automatic inventory expiration by setting the EXPINTERVAL server option to zero.

   ```
   setopt expinterval 0
   ```

2. Disable automatic migration and reclamation processes by using the **DEFINE STGPOOL** command to set the **HIGHMIG** and **RECLAIM** parameters to a value of *100*. You might have to increase the number of allowed processes for migration and reclamation in order for them to complete in a reasonable amount of time. The actual number of processes depends on available tape drives. If you have already defined storage pools, you can change values for the **MIGPROCESS** and **RECLAIMPROCESS** parameters by using the **UPDATE STGPOOL** command.

   ```
   def devc LARGEFILE devt=file mountlimit=500 maxcap=20480m dir=/tsmfile
   def stg FILEPOOL LARGEFILE maxscratch=200 reclaim=100 hi=100 lo=0 migpr=4
   reclaimpr=20 next=tapepool
   ```

3. If you have storage pools that are defined with data deduplication enabled, disable the duplicate identification processes:

   ```
   def stg FILEPOOL LARGEFILE maxscratch=200 reclaim=100 hi=100 lo=0  dedup=yes
   identifypr=0 migpr=4 reclaimpr=4
   ```

Example: Setting a schedule for the client backup window:

This example initiates an incremental backup of all associated nodes in the STANDARD domains.

The schedule starts daily at 8:00 PM by using server-prompted scheduling mode. The long-running schedules continue past the duration, so a shorter duration can be used to force those schedules to start close to the beginning of the start window.

```
def schedule standard nightly_backups description="Nightly backups of nodes in
domain standard" starttime=20:00 duration=5 durunits=hours period=1 perunits=days
```

Example: Setting up the server maintenance schedule:

Schedule server maintenance operations to run outside of the client backup window, with as little overlap as possible.

You can control the timing of schedules for maintenance tasks by setting the start time in combination with the duration time for each operation. Here is an example of how you might time each process:

08:00 - end
Storage pool backup.

11:00 to 13:00
Identification of duplicates.

13:00 - 15:00
Inventory expiration.

14:00 - 16:00
Storage pool migration.

16:00 - 18:00
Reclamation processing.

18:00 - end
Database backup, including volume history and device configuration backup.

20:00 - end
Client backup.

After you determine a timeline, use the **DEFINE SCHEDULE** command to create schedules for each process. You can include scripts in each schedule so that commands are processed automatically. Use the **DEFINE SCRIPT** command to create a script and the **UPDATE SCRIPT** command to add lines.

The following scripts are examples of how to define each server task:

Storage pool backup

```
/*--------------------------------------*/
/* Storage Pool Backup                  */
/*--------------------------------------*/
def script STGBACKUP "/* Run stg pool backups */"
upd script STGBACKUP "backup stg archivepool copypool maxproc=4
wait=yes" line=005
upd script STGBACKUP "backup stg backuppool copypool maxproc=4
wait=yes" line=010
upd script STGBACKUP "backup stg filepool copypool maxproc=4 wait=yes"
line=020
upd script STGBACKUP "backup stg filepool2 copypool maxproc=4 wait=yes"
line=025
```

```
upd script STGBACKUP "backup stg tapepool copypool maxproc=3 wait=yes"
line=030
def sched STGBACKUP type=admin cmd="run STGBACKUP" active=yes desc="Run
all stg pool backups." \
startdate=today starttime=08:00:00 dur=45 duru=minutes per=1 peru=day
commit
```

Identification of duplicates

```
/*--------------------------------------*/
/* Deduplication                        */
/*--------------------------------------*/

def script DEDUP "/* Run identify duplicate processes. */"
upd script DEDUP "identify duplicates FILEPOOL numpr=4 duration=120" \
line=010
upd script DEDUP "identify duplicates FILEPOOL2 numpr=2 duration=120" \
line=015
def sched  DEDUP type=admin cmd="run DEDUP" active=yes desc="Run identify
duplicates." \
startdate=today starttime=11:00:00 dur=45 duru=minutes per=1 peru=day
commit
```

Inventory expiration

```
/*--------------------------------------*/
/* Expiration
/*--------------------------------------*/
def script EXPIRE "/* Run expiration processes. */"
upd script EXPIRE "expire inventory wait=yes duration=120" line=010
def sched  EXPIRATION type=admin cmd="run expire" active=yes desc="Run
expiration." \
  startdate=today starttime=13:00:00 dur=45 duru=minutes per=1 peru=day
commit
```

Storage pool migration

```
/*--------------------------------------*/
/* Storage Pool Migration               */
/*--------------------------------------*/

def script MIGRATE "/* Run stg pool migration */"
upd script MIGRATE "migrate stg archivepool duration=30 wait=yes" line=005
upd script MIGRATE "migrate stg backuppool duration=30 wait=yes" line=010
upd script MIGRATE "migrate stg filepool2 duration=60 wait=yes" line=015
def sched  MIGRATE type=admin cmd="run MIGRATE" active=yes desc="Migrate
data to tape pools ." \
  startdate=today starttime=14:00 dur=45 duru=minutes per=1 peru=day
commit
```

Reclamation processing

```
/*--------------------------------------*/
/* Storage Pool Reclamation             */
/*--------------------------------------*/

def script RECLAIM "/* Run stg pool reclamation */"
upd script RECLAIM "reclaim stg filepool threshold=40 duration=120
wait=yes" line=005
upd script RECLAIM "reclaim stg filepool2 threshold=40 duration=120
wait=yes" line=008
upd script RECLAIM "reclaim stg tapepool threshold=60 duration=60 wait=yes"
line=010
def sched  RECLAIM type=admin cmd="run RECLAIM" active=yes desc="Reclaim
space from FILEPOOL and TAPEPOOL." \
  startdate=today starttime=16:00 dur=45 duru=minutes per=1 peru=day
commit
```

Database backup, including volume history and device configuration backup

```
/*--------------------------------------*/
/* Database Backup                      */
/*--------------------------------------*/
```

```
def script DBBACKUP "/* Run DB backups */"
upd script DBBACKUP "backup db devc=ts3310devc type=full wait=yes" line=005
upd script DBBACKUP "backup volhist" line=010
upd script DBBACKUP "backup devconf" line=015
def sched  DBBACKUP type=admin cmd="run DBBACKUP" active=yes desc="Run
database backup." \
  startdate=today starttime=18:00:00 dur=45 duru=minutes per=1 peru=day
commit
```

Tuning node replication

After you are replicating data, you can measure the effectiveness of your configuration and tune the speed of replication processes.

About this task

You can use commands that are specific to node replication to tune performance.

Measuring the effectiveness of a replication configuration
: A replication configuration is optimal if the number of replicated files that are stored on a target server is equal to the number of files that are stored on the source server. Use the **QUERY REPLNODE** command to display the number of files that are stored on source and target replication servers.

Increasing the speed of a node replication process
: You can set the **REPLBATCHSIZE** server option together with the **REPLSIZETHRESH** server option to increase the speed of processing between two replicating servers. These options specify how many files to include in a batch transaction and define a threshold for the size of the batch, in megabytes.

The default value for each option, which is 4096, is the best practice setting. If you want to improve the performance of a node replication process, try tuning the **REPLBATCHSIZE** and the **REPLSIZETHRESH** server options. Change the default values only after you monitor the node replication performance over several sessions. When the default values of the options are increased, the server requires more space in the active log. You might need to allocate space for the active log that is two or more times larger than an active log size that uses the default size of 4096. In addition, the server might require a longer initialization time when you start.

Use a trial and error method to increase the server options. You can increase the server options in any order. Start by increasing one of the options incrementally by 10%. If replication performance does not improve, revert the setting to its original value. Increase the other option incrementally by 10%. Ensure that you monitor active log usage during the first few replication operations to ensure sufficient active log space is available. Larger transactions run longer and use more active log space that causes other server processes to run slowly. If server processes run slowly, reduce the options until the replication and other server processes can complete.

Tuning server-side data deduplication

Tune settings and configuration for different operations to ensure that the performance of server-side data deduplication is efficient.

Procedure

| **Tip:** The following steps do not apply to container storage pools.

1. Control processor resources by setting the number of duplicate identification processes that you want to use. Do not exceed the number of processor cores available on your Tivoli Storage Manager server when you set the **NUMPROCESS** value. Define a duration limit for the **IDENTIFY DUPLICATES** command, otherwise, processes that are running after the command is issued run indefinitely.
2. Determine the threshold for reclamation of a deduplicated storage pool. A deduplicated storage pool is typically reclaimed to a threshold that is less than the default of 60 to allow more of the identified duplicate extents to be removed. Experiment with the setting of this value to find a threshold that can be completed within the available time.
3. Determine how many reclamation processes to run.

   Tip: A reclamation setting of more than 25 and less than 40 is sufficient.
4. Schedule data deduplication processing that is based on how you create a second copy of your data. If you are backing up your storage pool, do not overlap client backup and duplicate identification. Complete the storage pool backup before the identify process. If the storage pool backup is not complete, the copy process takes longer because it requires the deduplicated data to be reassembled before the backup.

   You can overlap duplicate identification and client backup operations in the following scenarios:
   - You are not backing up your storage pool.
   - You are using node replication to create a secondary copy of your data.

   Running these operations together can reduce the time that is required to finish processing, but might increase the time for client backup.
5. To prevent deadlocks in the Tivoli Storage Manager server, you might need to modify the DB2 **LOCKLIST** parameter before you deduplicate a large amount of data. When the amount of concurrent data movement activity is high, deadlocks can occur in the server. If the amount of concurrent data that is moved exceeds 500 GB at a time, adjust the DB2 **LOCKLIST** parameter as follows:

*Table 17. Tuning DB2 **LOCKLIST** parameter values.*

Amount of data	LOCKLIST parameter value
500 GB	122000
1 TB	244000
5 TB	1220000

Related concepts:

"Checklist for data deduplication" on page 30

Related tasks:

"Scheduling data deduplication and node replication processes" on page 136
"Evaluating data deduplication results" on page 67

"Tuning client-side data deduplication" on page 200

Restoration of deduplicated data

Restore operations that require data to be reassembled from a sequential-access disk (FILE) storage pool that is set up for data deduplication have different performance characteristics than restore operations from a FILE storage pool that is not set up for deduplication.

In a FILE storage pool that is not set up for data deduplication, files are typically restored in a sequential process. In a FILE storage pool that is set up for data deduplication, however, data is distributed throughout the storage pool. As a result, the input/output (I/O) is more random, which can lead to slower restore times. In addition, more server processor resources are consumed when data is restored from a deduplicated storage pool. This occurs because the data is checked to ensure that it has been reassembled properly by using MD5 algorithms.

Although restore operations of small files from a deduplicated storage pool might be relatively slow, these operations are still typically faster than restore operations of small files from tape because of the added tape mount and locate time.

Improving read performance for deduplicated storage pools

To obtain the different extents that make up a file from a deduplicated storage pool, client restore operations and certain server processes might require opening and closing FILE volumes multiple times. The frequency with which FILE volumes are opened and closed during a session can severely affect performance.

About this task

Opening and closing volumes multiple times can affect the following server processes that read data from a deduplicated storage pool:

- Volume reclamation
- **MOVE DATA** or **MOVE NODEDATA**
- **EXPORT**
- **AUDIT VOLUME**
- Storage-pool restore operation
- Volume restore operation
- Data migration

To reduce the number of times a volume is opened and closed, Tivoli Storage Manager allows multiple input FILE volumes in a deduplicated storage pool to remain open at the same time during a session. To specify the number of open FILE volumes in deduplicated storage pools that can remain open, use the **NUMOPENVOLSALLOWED** server option. Set this option in the server options file or by using the **SETOPT** command.

During a client-restore operation, volumes can remain open for as long as the client session is active. During a no-query restore operation, the volumes remain open until the no-query restore completes. Then, all volumes are closed and released. However, for a standard restore operation started in interactive mode, the volumes might remain open at the end of the restore operation. The volumes are closed and released when the next classic restore operation is requested.

Procedure

This option can significantly increase the number of volumes and mount points in use at any one time. To optimize performance, complete the following tasks:

- To set **NUMOPENVOLSALLOWED**:
  1. Select a beginning value. The default is 10. A small increase to this option can be beneficial but that might not be true in all environments.
  2. Monitor client sessions and server processes.
  3. Note the highest number of volumes open for a single session or process. If the highest number of open volumes is equal to the value specified by **NUMOPENVOLSALLOWED**, increase the setting of **NUMOPENVOLSALLOWED**.
- To prevent sessions or processes from having to wait for a mount point:
  1. Increase the value of the **MOUNTLIMIT** parameter in the device-class definition.
  2. Set the value of the **MOUNTLIMIT** parameter high enough to allow all client sessions and server processes that are using deduplicated storage pools to open the number of volumes that are specified by the **NUMOPENVOLSALLOWED** option.
  3. Check the following results:
     - For client sessions, check the destination in the copy group definition to determine how many nodes are storing data in the deduplicated storage pool.
     - For server processes, check the number of processes that are allowed for each process for the storage pool.
- For any node that is backing up or archiving data into a deduplicated storage pool, set the value of the **MAXNUMMP** parameter in the client-node definition to a value at least as high as the **NUMOPENVOLSALLOWED** option. Increase this value if you notice that the node is failing client operations because the **MAXNUMMP** value is being exceeded.

Results

Each session within a client operation or server process can have as many open FILE volumes as specified by this option. A session is initiated by a client operation or by a server process. Multiple sessions can be started within each.

Tuning server operations for client backups

When possible, limit the number of versions of any backup file to the minimum required.

About this task

File backup performance is degraded when there are many versions of an object. Use the **DEFINE COPYGROUP** command and modify the VEREXISTS parameter to control the number of versions, or use the **UPDATE COPYGROUP** command. The default number of backup versions is 2.

If the retention requirements in your environment differ among client systems, use different copy groups rather than taking the lowest common denominator. For example, if your accounting systems require records to be kept for seven years, but other systems need data to be kept for only two years, do not specify seven for all of them. Instead, create two separate copy groups. Not only are backups potentially faster, but you also use less storage because you are not keeping data that you do not need.

Similarly, you can set up a separate copy group for system state backups to avoid keeping unnecessary operating system files. For example, if you want to retain system state data for a week and all other data for a year, create a separate copy group for the system state data.

Tuning operations for automatic backup-archive client deployment

You can take actions that might improve backup-archive client deployment performance.

Procedure

- Deploy the client when the clients, server, and network have minimal activity. Do not deploy during client backup operations.
- To avoid the cost of retrieving a wrong package, use separate schedules for each client architecture (for example, x86, x64, ia64).
- If multiple clients are being upgraded concurrently, store the deployment packages in a random-access `DISK` or in a sequential-access `FILE` storage pool. Both types of storage pools support read access to the same object from multiple sessions concurrently. If the storage pool uses tape, the server serializes read access to the storage pool volume that contains the upgrade package data. Serial access is also used for a storage pool volume in a virtual tape library (VTL), even if the data is stored on physical disks.
- Provide sufficient cache memory on the disk system that is used by the random-access `DISK` or sequential-access `FILE` storage pools that contain the deployment packages. Deployment packages are read from the storage pool volumes during retrieval by using direct input/output (I/O). This means that the server operating system does not cache the data in memory, and each I/O must come from the disk system. When the server is deploying many clients at the same time, the disk system sees a high read-hit ratio on these data blocks, which leads to faster throughput.
- Balance multiple clients across the network interfaces on the server. Such balancing might be done to optimize backup performance as well.

Tuning tape drive performance

There are a few basic procedures for maintaining the performance of your tape drives.

Configuring enough tape drives

You must configure enough tape drives for operations that occur at the same time in your environment:

- The maximum number of Tivoli Storage Manager client sessions that are backing up directly to tape at any time during the peak backup window.
- Extra tape drives for other functions that run during the backup window. For example, storage pool migration, storage pool backup, and reclamation.

Cleaning tape drives

Cleaning the tape drive according to the manufacturer's specifications is important to ensure maximum tape drive performance. Failure to clean tape drives can cause read and write errors, drive failures, and poor performance.

Enabling tape compression

In most cases, the preferred method of enabling compression at the tape drive improves Tivoli Storage Manager throughput.

You can use the **FORMAT** parameter of the **DEFINE DEVCLASS** command to specify the appropriate recording format to be used when you write data to sequential access media. The default is DRIVE, which specifies that Tivoli Storage Manager selects the highest format that can be supported by the sequential access drive on which a volume is mounted. This setting usually allows the tape control unit to perform compression.

Tip: Avoid specifying the DRIVE value when a mixture of devices is used in the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the **FORMAT** parameter with the DRIVE value.

If you do not use compression at the client and your data is compressible, you should achieve higher system throughput if you use compression at the tape control unit unless your network is slow.

Tape drive transfer rate

Many factors affect the sustained transfer rate of Tivoli Storage Manager when you use tape drives. The sustained transfer rate takes into account the net effect of all these factors.

The following factors affect the sustained transfer rate:

- Native transfer rate
- Compression ratio
- File size
- Server attachment
- Server attachment host bus adapter (HBA) type
- Disk transfer rate
- Network bandwidth
- Server utilization
- Start/stop performance
- Application control file activity
- Tivoli Storage Manager transaction size
- Bus bandwidth
- Quality of the media
- Use of collocation for restore operations
- Distribution of data on the tape volume

Tape drive streaming rate performance

Streaming rate is the rate at which a tape drive can read and write, not including any start and stop operations. Most uses of tape include some start and stop operations, which slow down the sustained rate at which the drive operates.

When you are writing to a tape drive, the drive normally returns control to the application when the data is in the tape drive buffer but before the data is written to tape. This mode of operation provides all tape drives a significant performance improvement. However, the drive's buffer is volatile. For the application to ensure that the data is written to tape, the application must flush the buffer. Flushing the buffer causes the tape drive to back hitch (start/stop). The Tivoli Storage Manager parameters **TXNBYTELIMIT** and **TXNGROUPMAX** control how frequently Tivoli Storage Manager issues this buffer flush command.

When you are writing to a tape drive, you must consider network bandwidth. For example, 1-gigabit Ethernet can sustain 60 - 70 MB per second. Therefore, you cannot back up to a tape drive faster than that rate.

Using high performance tape drives

When you use high performance tape drives with Tivoli Storage Manager, it is important to use the appropriate server and client options to enhance performance.

Consider these settings for the best performance:

Server options

```
TXNGROUPMAX 4096
MOVESIZETHRESH 32768
MOVEBATCHSIZE 1000
```

Client options

```
TXNBYTELIMIT 10G
```

If on average, Tivoli Storage Manager clients have files smaller than 100 KB, back these clients up to a disk storage pool for later migration to tape. This allows more efficient data movement to tape.

Tuning HBA capacity

The server system must have enough host bus adapters (HBAs) to handle data operations that Tivoli Storage Manager runs simultaneously.

About this task

Ensure that you have enough HBA bandwidth to handle peak loads on the system. When you are planning for peak loads, consider all operations that can occur simultaneously in your environment.

For example, if you are backing up to a disk pool, you need enough network bandwidth for the client backup operation. You also need a similar amount of outgoing bandwidth to disk through fibre, SAS, or other HBA for that backup data. More bandwidth is required if storage pool migration tends to run during the backup window. In addition to the bandwidth needed for the backup operation, you need bandwidth to read the data from disk and to write to tape. If you consider only the bandwidth that is needed for the backup operation, your available bandwidth is limited when storage pool migration starts.

Related concepts:

"Potential bottlenecks in the data flow for Tivoli Storage Manager operations" on page 49

Related reference:

"Compatibility and resource usage for server processes" on page 138

Tuning tasks for operating systems and other applications

Review this information for guidance on improving operating system performance for the server and for impacts that are associated with applications other than Tivoli Storage Manager.

Tuning AIX systems for Tivoli Storage Manager server performance

There are a number of actions that can improve performance for a Tivoli Storage Manager server that is running in an AIX environment.

About this task

You can use a Tivoli Storage Manager server in System p logical partitions (LPARs).

Procedure

- Use the `rbrw mount` option to release memory from the file system cache. For more information about the release-behind sequential read and write (**`rbrw`**) option, see the AIX product information.

  AIX systems can cache much file system data, which can take away memory that is needed for Tivoli Storage Manager server and DB2 processes. To avoid paging with the AIX server, use the `rbrw mount` option for the JFS2 file system. Less memory is used for the file system cache and more is available for Tivoli Storage Manager.

  Concurrent I/O (CIO) options are not necessary to access or mount database or log file systems. Tivoli Storage Manager completes mounts automatically. Also, CIO disables the read-ahead feature of JFS2 file systems, decreasing the read performance of the database during backup. Do not use the file system mount options, CIO and Direct I/O (DIO), for file systems that contain the Tivoli Storage Manager database, logs, or storage pool volumes. These options can cause performance degradation of many server operations. Tivoli Storage Manager and DB2 can still use DIO where it is beneficial to do so, but Tivoli Storage Manager does not require the mount options to selectively take advantage of these techniques.
- Use the Portable Operating System Interface (POSIX) time zone specification for best system performance.

  If the server is installed on an AIX Version 6.1 operating system, you must use the Portable Operating System Interface (POSIX) time zone specification. The alternative time zone specification, which is based on the Olson database, severely degrades system performance. For information about setting time zone specifications, see the documentation for the AIX operating system. If the server is installed on AIX Version 7.1, this issue does not occur.
- Tivoli Storage Manager supports sharing Fibre Channel ports through NPIV in a pSeries LPAR. While you can share these ports, be sure that the port has adequate bandwidth for all LPARs that use that port. Sharing resources with other logical partitions can affect the performance of the Tivoli Storage Manager

server. When there are other logical partitions on the system, you can dedicate resources to the Tivoli Storage Manager server partition.
- Tivoli Storage Manager can use 10-gigabit Ethernet ports that are shared through the VIO server. However, when you use a shared VIO port, it does not always provide full 10-gigabit throughput. If full 10-gigabit Ethernet bandwidth is needed, you can use the logical Host Ethernet Adapter (LHEA) method of sharing.

Tuning AIX virtual memory

The AIX virtual address space is managed by the Virtual Memory Manager (VMM). Monitor the paging statistics to identify problems with virtual memory usage.

Procedure

To monitor paging statistics and identify potential issues, complete the following steps:

1. Run the **vmstat** command. Review the paging statistics in the pi and po columns. An occasional nonzero value is not a concern because paging is the main principle of virtual memory. If the values are constantly nonzero, there might be a memory bottleneck.
2. If paging occurs constantly, verify whether the problem is caused by heavy use of the file system cache. Review the output from the following command:
   `vmstat -I 5`
3. If the values for pi and po columns are high, and the values are similar to the values for fi and fo columns, consider using the `rbrw mount` option on all active JFS2 file systems to reduce or eliminate the paging problem.
4. If the paging problem persists after you use the `rbrw mount` option for JFS2 file systems, run the AIX **vmo** command to tune the virtual memory system. For more information about using the **vmo** command, see the documentation for the AIX operating system.

Related tasks:

"Monitoring performance with operating system tools" on page 42
"Configuring AIX systems for disk performance" on page 170

Tuning Linux systems for Tivoli Storage Manager server performance

There are a number of actions that can improve performance for a Tivoli Storage Manager server that is running in a Linux environment.

Before you begin

Review system requirements for the installation of a Tivoli Storage Manager server to ensure that you have the required specifications for your operating system. For more information, see technote 1243309.

Procedure

- Most enterprise distributions come with many features, however most of the time only a small subset of these features are used. Disable features that are not used.
- Use the **vm.pagecache_limit_mb** and **vm.swappiness** options to release memory from the file system cache.

Linux systems can cache much file system data, which can take away memory that is needed for Tivoli Storage Manager server and DB2 processes. As a root user on Linux, you can limit the amount of memory that is allowed for caching file data by setting the **vm.pagecache_limit_mb** kernel parameter to 1024. Also, set the **vm.swappiness** kernel parameter to 0. For example:

```
linuxbox:/ # sysctl vm.pagecache_limit_mb          # to display current value
vm.pagecache_limit_mb = 0                          # (0 means no limit)
linuxbox:/ # sysctl -w vm.pagecache_limit_mb=1024   # to change at runtime
vm.pagecache_limit_mb = 1024
linuxbox:/ # sysctl vm.swappiness
vm.swappiness = 60
linuxbox:/ # sysctl -w vm.swappiness=0
vm.swappiness = 0
```

To apply these changes for all restarts of the operating system, edit the `/etc/sysctl.conf` file and add `vm.pagecache_limit_mb=1024` and `vm.swappiness=0`.

Tuning Linux on System z systems for Tivoli Storage Manager server performance

You can use a number of methods to improve the performance of a Tivoli Storage Manager server on a Linux on System z system.

About this task

For information about performance for Linux on System z systems, see Tuning hints and tips.

Procedure

The following steps can help to improve the performance of a Tivoli Storage Manager server on a Linux on System z system:

- Upgrade to SUSE Linux Enterprise Server 11 Service Pack 1 (SLES11 SP1) for better disk and network performance.
- Use Fibre Channel SCSI disks instead of fiber-connected DASD (direct access storage device), if possible, for greater throughput.
- Dedicate all Tivoli Storage Manager database, log, and storage disks to the Linux guest if the system is running under z/VM®.
- Use logical volume manager (LVM) for disk storage-pool logical volumes. Using LVM striping improves throughput on operations such as storage pool backup and migration.
- Use the ext4 file system for improved performance for defining storage pool volumes.
- For the Tivoli Storage Manager database and logs, use either the ext3 or ext4 file system. As a best practice, use the following file system that is appropriate for your operating system and level:
  - For Red Hat Enterprise Linux x86_64, use the ext3 or ext4 file systems. Use the ext4 file system only if Red Hat Enterprise Linux 6.4 or later is installed.
  - For SUSE Linux Enterprise Server and for Red Hat Enterprise Linux ppc64, use the ext3 file system.
- Use OSA-Express3 network adapters instead of OSA-Express2 for greater throughput and reduced processor use. See the networking suggestions for OSA-Express3 at IBM z Systems™ - Networking features.

- For external network connections to a Linux guest under z/VM, attach the OSA adapter directly to the Linux guest.

Tuning Oracle Solaris systems for Tivoli Storage Manager server performance

There are a number of actions that can improve performance for a Tivoli Storage Manager server running in a Solaris environment.

Procedure

- Use raw partitions for disk storage pool volumes on the Solaris platforms. Raw logical volumes offer better client backup and restore throughput than ZFS, UFS, or Veritas file system volumes.
- Use the most recent version of Solaris and apply the latest kernel patch.
- If you use ZFS file systems, see "Configuring Oracle Solaris systems for disk performance" on page 171 for tips.

  Restriction: ZFS does not work with direct I/O.
- If you use UFS file system volumes, specify the `forcedirectio` flag when you mount these file systems. If the file system is mounted with this option, data is transferred directly between the user address space and the disk. If you specify the `noforcedirectio` option when the file system is mounted, data is buffered in kernel address space when data is transferred between user address space and the disk. Large sequential data transfers benefit from the `forcedirectio` performance option. The default behavior is `noforcedirectio`.
- If you use VxFS for disk or file storage pool volumes, mount the VxFS file systems with the direct I/O option:

  ```
  mincache=direct,convosync=direct
  ```

Tuning Windows systems for Tivoli Storage Manager server performance

You can take a number of actions to improve performance for a Tivoli Storage Manager server that is running in a Windows environment.

Procedure

The following actions can help improve performance:

- Disable NTFS file compression on disk volumes. Because of the potential for performance degradation, do not use NTFS file compression on disk volumes that are used by the Tivoli Storage Manager server.
- Use the shared memory communication method when you are using a local client. For optimal backup and restore performance when you are using a local client on a Windows system, use the shared memory communication method. The method is used by including the **`COMMMETHOD`** option set to `SHAREDMEM` in the server options file and in the client options file.
- Use the VMXNET 3 network adapter type when the Tivoli Storage Manager server is in a VMware guest environment. Provide all disks that are used for the server database, log file, and storage as mapped raw LUNs rather than using virtual disks in a VMware datastore.
- Additional actions can affect Tivoli Storage Manager client and server performance.
  - Windows 8 Defender can significantly degrade Tivoli Storage Manager backup and restore throughput, especially for smaller files. To improve

backup and restore performance on Windows 8, which might increase the security risk to the system, use one of the following methods:

  - Disable Windows 8 Defender. Click **Start** > **Administrative Tools** > **Computer Management** > **Services and Applications** > **Services**. Locate Windows 8 Defender in the list of services. Right-click **Windows Defender** and then select **Properties**. Change the Startup Type attribute to **Disabled**.
  - Without disabling Windows 8 Defender Service, exclude a specific drive that has backup or restore errors. Use this method when there are multiple logical drives on the system. Excluding a drive is less of a security risk than disabling Windows 8 Defender Service.

- Antivirus software can negatively affect backup performance.
- Disable or do not install unused services.
- Disable or do not install unused network protocols.
- Give preference to background application performance.
- Avoid screen savers.
- Ensure that the paging file is not fragmented.
- Ensure that any device driver is current, especially for new hardware.

Secure Sockets Layer (SSL) effects on server performance

Secure Sockets Layer (SSL) provides secure communications between the Tivoli Storage Manager client and server, but it can affect system performance.

If SSL is needed, use it only for sessions where it is necessary and add processor resources on the Tivoli Storage Manager server system to handle the increased requirements. Or, try other options, such as networking devices like routers and switches, that provide SSL function instead.

LDAP directory server usage: effects on performance

If you are using an LDAP server to authenticate administrator and node passwords, there might be some performance impacts.

More processor resources are used when you are authenticating with a Lightweight Directory Access Protocol (LDAP) server instead of using local authentication. Tests in IBM labs show that LDAP has about a 5% impact.

If you are using Secure Sockets Layer (SSL) sessions in combination with LDAP server authentication, the additional performance impact for sessions that transfer small amounts of data is negligible. For sessions that transfer large amounts of data, you can expect a significant performance impact because SSL must encrypt all of the data.

Chapter 12. Tuning disk storage for the server

Disk storage systems have different operating characteristics that can be configured and tuned for improved performance for a Tivoli Storage Manager server.

About this task

Review the information about how to configure your disk storage systems, and the operating system for your Tivoli Storage Manager server.

Principles for tuning disk systems for Tivoli Storage Manager

Many aspects of disk storage can be considered for optimizing operations. For most systems, the separation of server database, logs, and storage pools is key to a Tivoli Storage Manager configuration that has good performance.

The following principles are key to achieving better disk storage performance:

- Select and configure disk storage for both performance and capacity. Adequate capacity is not the only factor to consider.
- For most disk systems, separate the primary Tivoli Storage Manager server components from each other. Ensure that the server database, active log, archive logs, and storage pools are each in separate locations.
- Monitor the systems. Workloads on systems often increase, and such increases might trigger the need for more storage or for configuration changes. Enforce strict change control to help in troubleshooting any performance degradation.
- Limit mirroring to one type of mirroring. For example, if the operating system is set up to do mirroring, do not configure the Tivoli Storage Manager server to do mirroring of the active log (**MIRRORLOGDIRECTORY** server option).
- Ensure that server maintenance operations are running, such as expiration and database table and index reorganization. See "Tuning the schedule for daily operations" on page 132.

Understand the entire picture of disk storage and how it relates to operations in your environment. You must examine not only on the configuration of the disks, but the entire configuration as it relates to performance. All of the following items have an impact:

Disk storage system choice and configuration

- Disk type and speed. Larger disks might not be better if the rotational speed is not also higher.
- Disk layout.
- Type of RAID.
- File system type and mount options.

Server system hardware and its configuration

- Speed and number of processors and amount of memory.
- Whether multiple instances of Tivoli Storage Manager run on the same system and use the same disk storage systems.
- Host bus adapter (HBA) speed.

- Whether HBAs are dedicated to disk operations. An HBA that is shared by disk and tape can have performance problems.
- Whether disks are shared with other systems or applications.

Tip: Look for updated information about performance in the Tivoli Storage Manager wiki.

Related reference:

"Checklist for server database disks" on page 17

"Checklist for server recovery log disks" on page 19

Disk system types

The choice of disk system affects the configuration options that are available. How the disk system is configured affects the resulting performance of the Tivoli Storage Manager server. For example, disk systems vary by how the individual disk units can be organized to create RAID arrays.

The following list shows the variety of system types that can be used for Tivoli Storage Manager disk storage:

Solid-state drive (SSD) systems
Solid-state drive technology (sometimes called *flash memory*) provides the highest level of performance, with much higher rates of I/O operations per second (IOPS) than other storage systems. An SSD works at speeds much closer to that of memory than disk drives. SSD has no delay from a spinning platter or from an arm that must move to the correct position.

If you use SSD for Tivoli Storage Manager storage, ensure that you are using enterprise-level quality SSD.

IBM DS8000® series
The DS8000 series are high-performance disk systems that accept several different RAID types, including RAID 5 and RAID 10. The size of the arrays, in terms of the quantity of disk units, is fixed. Therefore, a DS8000 series unit has a fixed number of RAID arrays (ranks).

IBM DS5000 series
The DS5000 series of midrange disk systems can manage a wide variety of disk configurations. You can create RAID arrays with as few as two and as many as several dozen disks. You can have smaller RAID arrays for the Tivoli Storage Manager database and larger arrays for the Tivoli Storage Manager disk storage pool.

IBM Storwize® V7000
The Storwize V7000 system is a midrange system that virtualizes RAID storage. A system consists of a set of drive enclosures. You configure the drives into arrays, and create volumes from those arrays. You can configure the system with multiple device types. With multiple device types, you can use fast disk or SSD for the server database and lower-cost, higher-capacity disk for storage pools.

IBM Storwize V3700 is an entry-level system that has similar characteristics as the Storwize V7000.

Optimization of disk system read-ahead processes

Most advanced disk systems can automatically optimize the performance of read operations when the disk system can detect sequential reads. When the disk system detects sequential reads, it can have the data for the next read in cache, or at least have the read in progress.

Disk systems detect sequential reads on a LUN-by-LUN basis. However, a sequential read might not be detected if more than one read is in progress for the same LUN. Disk systems do not perceive file systems or files within a LUN and distinguish only the blocks that are being accessed. When two sequential reads are in progress on one LUN, the blocks that are being accessed no longer respond as sequential blocks. The blocks appear to come from different places, and the read-ahead optimizations are typically stopped.

Choosing the correct type of storage technology for Tivoli Storage Manager

Storage devices have different capacity and performance characteristics. These characteristics affect which devices are better for use with Tivoli Storage Manager.

Procedure

Review the following table to help you to choose the correct type of storage technology for the storage resources that the server requires.

Table 18. Storage technology types for Tivoli Storage Manager storage requirements

<table>
<tr><th>Storage technology type</th><th>Database</th><th>Active log</th><th>Archive log and archive failover log</th><th>Storage pools</th></tr>
<tr><td>Solid-state disk (SSD)</td><td>Place the database on SSD in the following circumstances:
You are using Tivoli Storage Manager data deduplication.
You are backing up more than 8 TB of new data daily.</td><td>If you place the Tivoli Storage Manager database on an SSD, as a best practice, place the active log on an SSD. If space is not available, use high-performance disk instead.</td><td>Save SSDs for use with the database and active log. The archive log and archive failover logs can be placed on slower storage technology types.</td><td>Save SSDs for use with the database and active log. Storage pools can be placed on slower storage technology types.</td></tr>
<tr><td>High-performance disk with the following characteristics:
15k rpm disk
Fibre Channel or serial-attached SCSI (SAS) interface</td><td>Use high-performance disks in the following circumstances:
The server does not do data deduplication.
The server does not do node replication.
Isolate the server database from its logs and storage pools, and from data for other applications.</td><td>Use high-performance disks in the following circumstances:
The server does not do data deduplication.
The server does not do node replication.
For performance and availability, isolate the active log from the server database, archive logs, and storage pools.</td><td>You can use high-performance disks for the archive log and archive failover logs. For availability, isolate these logs from the database and active log.</td><td>Use high-performance disks for storage pools in the following circumstances:
Data is frequently read.
Data is frequently written.
For performance and availability, isolate storage pool data from the server database and logs, and from data for other applications.</td></tr>
</table>

Table 18. Storage technology types for Tivoli Storage Manager storage requirements (continued)

Storage technology type	Database	Active log	Archive log and archive failover log	Storage pools
Medium-performance or high-performance disk with the following characteristics: • 10k rpm disk • Fibre Channel or SAS interface	If the disk system has a mix of disk technologies, use the faster disks for the database and active log. Isolate the server database from its logs and storage pools, and from data for other applications.	If the disk system has a mix of disk technologies, use the faster disks for the database and active log. For performance and availability, isolate the active log from the server database, archive logs, and storage pools.	You can use medium-performance or high-performance disk for the archive log and archive failover logs. For availability, isolate these logs from the database and active log.	Use medium-performance or high-performance disk for storage pools in the following circumstances: • Data is frequently read. • Data is frequently written. For performance and availability, isolate storage pool data from the server database and logs, and from data for other applications.
SATA, network-attached storage	Do not use this storage for the database. Do not place the database on XIV storage systems.	Do not use this storage for the active log.	Use of this slower storage technology is acceptable because these logs are written once and infrequently read.	Use this slower storage technology in the following circumstances: • Data is infrequently written, for example written once. • Data is infrequently read. .
Tape and virtual tape				Use for long-term retention or if data is infrequently used.

Tuning System Storage DS8000 series storage systems

IBM System Storage® DS8000 series storage systems are designed to be fast and manage heavy I/O.

About this task

A DS8000 series storage system can be configured in RAID 5 and RAID 10 arrays. The quantity of disk units in an array is fixed. Stripe or segment sizes are set automatically. You cannot tune cache for a DS8000 series storage system; for example, you cannot disable cache for the LUNs that are used for the server database. The inability to tune cache is usually not a problem because these systems typically have large amounts of cache.

Procedure

For best performance, follow these guidelines:

- Place the Tivoli Storage Manager database, active log, archive log, and disk storage pools on separate extent pools.

This arrangement uses more storage, but achieves better performance.

- Distribute the server components over as many ranks as possible. Ranks are on different device adapter pairs.
- Minimize other applications that are sharing the ranks that Tivoli Storage Manager uses.
- Use as many adapters as possible to access LUNs.
- If you are using a DS8000 series storage system for multiple Tivoli Storage Manager servers, put all the server databases on one set of ranks. Put all the storage pools for the servers on another set of ranks.
- Test and monitor the results for the disk system configuration in your environment.

Tuning System Storage DS5000 series and other IBM midrange storage systems

IBM System Storage DS5000 series and other IBM midrange storage systems offer great flexibility as to how they can be configured for use with Tivoli Storage Manager.

About this task

The systems have these characteristics:

- Many types of RAID can be used.
- The number of disks per LUN is flexible.
- Segment or stripe sizes and cache can be set for each LUN.
- Different models have different disk types (Fibre Channel or SATA).
- Different amounts of system cache are available, though typically less than in systems such as System Storage DS8000 series.

Procedure

For best performance with IBM midrange disk systems, separate the Tivoli Storage Manager database, recovery log, and storage pools so that they are on different physical spindles. Samples show how to configure these types of disk systems:

- “Sample layouts for a server database on DS5000 series disks” on page 164
- “Sample layouts for server recovery logs on DS5000 series disks” on page 166
- “Sample layout of server storage pools on DS5000 series disks” on page 168

Disk I/O characteristics for Tivoli Storage Manager operations

Generally, Tivoli Storage Manager storage pools are written to and read from by using I/O sizes of 256 KB.

With page sizes of 8 KB and 32 KB, the I/O that is used by the Tivoli Storage Manager database varies. The database manager might, at times, prefetch larger amounts.

While Tivoli Storage Manager might request the 8 KB and 32 KB I/O sizes, the operating system might choose to run I/O differently. Running the I/O differently can result in smaller or larger I/O being sent to the disk system.

Tivoli Storage Manager tries to use direct I/O in most situations, which avoids the file system cache. The result of avoiding the cache is better processor efficiency and performance. If you use operating system parameters to tune the file system cache, you might notice no effect for this reason.

Sample layouts for a server database on DS5000 series disks

Samples illustrate several ways to follow configuration guidelines for the server database when you use DS5000 series Fibre Channel disks. The samples illustrate advantages and disadvantages of configuration choices.

Remember: Ensure that the server database, recovery logs, and storage pools are on different disks.

Sample 1: Good layout for a small server

By using five disks for the database, you can set up the disks with the following characteristics. See Figure 24.

- Configure disks in a 4 + 1 RAID 5 array.
- Set the stripe size to 256 KB.
- Define one directory (also called a container) and one logical volume for the database.
- Set the **DB2_Parallel_IO** environment variable:

  ```
  DB2_Parallel_IO=*:4
  ```

  The IBM DB2 program that is the server's database manager uses this value when it balances load across the disks.

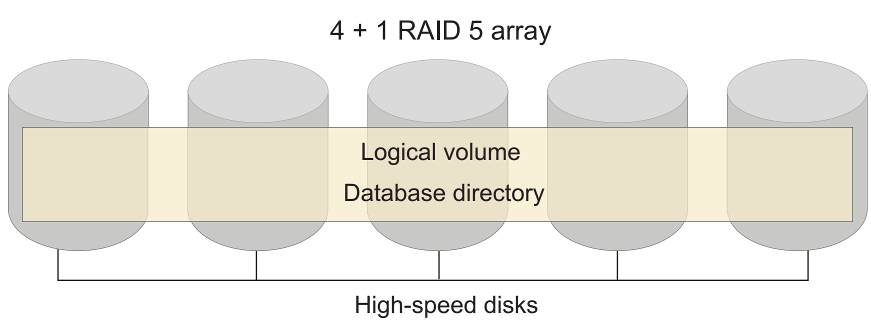

Figure 24. Database layout for a small server

This layout has the following advantages:

- The layout follows the guidelines for optimal stripe size.
- The layout follows the guidelines for having a one-to-one correspondence between logical volumes and containers.

The layout has the following disadvantages:

- Write operations for parity bits can affect performance if the system has a small cache.
- The layout has only one container for the database, which is typically not optimal but might not be a problem for small server workloads.
- Database and database indexes are spread across only five disks.

Sample 2: Better layout that uses RAID 10

By using eight disks for the database, you can set up the disks with the following characteristics. See Figure 25.

- Configure disks in a 4 + 4 RAID 10 array.
- Set the stripe size to 256 KB.
- Define one directory (also called a container) and one logical volume for the database.
- Set the **DB2_Parallel_IO** environment variable:

```
DB2_Parallel_IO=*:4
```

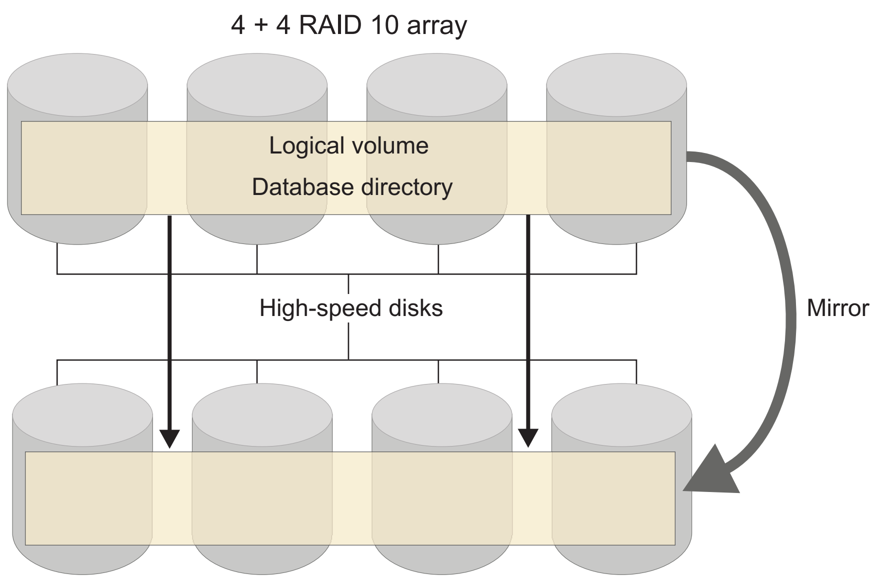

Figure 25. Database layout that uses eight disks in a RAID 10 array

This layout has the following advantages:

- The layout follows the guidelines for optimal stripe size.
- The layout follows the guidelines for having a one-to-one correspondence between logical volumes and containers.
- The system does not have the performance cost of parity write operations.
- RAID 10 is an optimal choice for the server database. With its two sets of disks, this configuration results in faster reads for the database.

The layout has the following disadvantages:

- The layout has only one container for the database, which is typically not optimal but might not be a problem for small server workloads.
- The layout requires twice as many disks as in sample 1 because of the use of RAID 10 instead of RAID 5.

Sample 3: Better layout that uses RAID 10 and more containers

By using 16 disks for the database, you can set up the disks with the following characteristics. See Figure 26 on page 166.

- Configure the disks in two 4 + 4 RAID 10 arrays.

- Set the stripe size to 256 KB.
- Define two directories (also called containers) and two logical volumes for the database.
- Set the **DB2_Parallel_IO** environment variable:

  ```
  DB2_Parallel_IO=*:4
  ```

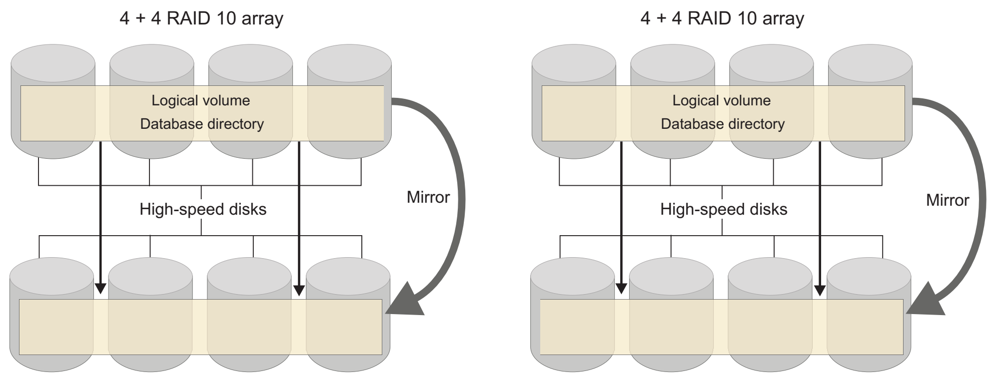

Figure 26. Database layout that uses 16 disks in two RAID 10 arrays

This layout has the following advantages:
- The layout follows the guidelines for optimal stripe size.
- The layout follows the guidelines for having a one-to-one correspondence between logical volumes and containers.
- The system does not have the performance cost of parity write operations.
- RAID 10 is an optimal choice for the server database. With its two sets of disks, this configuration results in faster reads for the database.
- Many more physical spindles means better performance for database read and write operations.
- More database containers means that more data can be prefetched by the DB2 software.

The disadvantage is that this layout requires more disks than the other samples.

Sample layouts for server recovery logs on DS5000 series disks

Samples illustrate several ways to follow configuration guidelines for server recovery logs when you use DS5000 series Fibre Channel disks. The active log is placed on disk with the fastest speed characteristics.

Remember: Ensure that the server database, recovery logs, and storage pools are on different disks.

Sample 1: Good layout without RAID

In this sample, disks that are not configured as a RAID array are used for the recovery logs. This type of arrangement is called *just a bunch of disks* (JBOD). The active log, the archive log, and the failover archive log are placed on separate disks.

The layout has the following advantages:

- Cache read-ahead is used for the disks.
- The separation of the active log, the archive log, and the failover archive log follows guidelines.
- The active log is placed on the fastest disk.

The layout has the following disadvantages:
- This layout has a single point of failure. For example, if the disk for the active log failed, you have no RAID to help you recover.
- All files for the active log are on one disk, which might be slower than if you spread out the files.

Sample 2: Better layout with RAID 1

In this sample, RAID 1 disks are used for the active log and the archive log. This sample has the following features:
- RAID 1 acts as a mirror for the active log. As an alternative, you can use the Tivoli Storage Manager server option for mirroring the active log, **MIRRORLOGDIRECTORY**.
- RAID 1 is used for the archive log.
- The disk for the archive failover log is not RAID 1 because it is not as critical to server operations as the other logs.

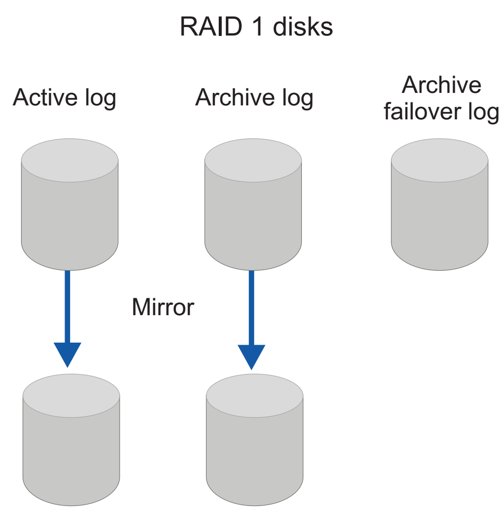

Figure 27. Logs layout with RAID 1

The layout has the following advantages:
- Fast Fibre Channel disks are used.
- Cache read-ahead is used for the disks.
- RAID 1 is used for the active log and archive log locations, which removes the single point of failure for those logs.

Performance might be slower because all of the files for the active log are on one disk.

Sample layout of server storage pools on DS5000 series disks

A sample illustrates how to follow configuration guidelines for storage pools that use the DISK device class and DS5000 series disks.

Remember: Ensure that the server database, recovery logs, and storage pools are on different disks.

Sample layout

In this sample, the Fibre Channel or Serial Advanced Technology Attachment (SATA) disks in a DS5000 series system are configured with these characteristics:

- The disks are configured in a 4 + 1 RAID 5 array. Stripe size is 256 KB.
- Four logical volumes are defined on the disks. In Tivoli Storage Manager, these volumes are defined as four storage pool volumes for a random-access (DISK) storage pool.

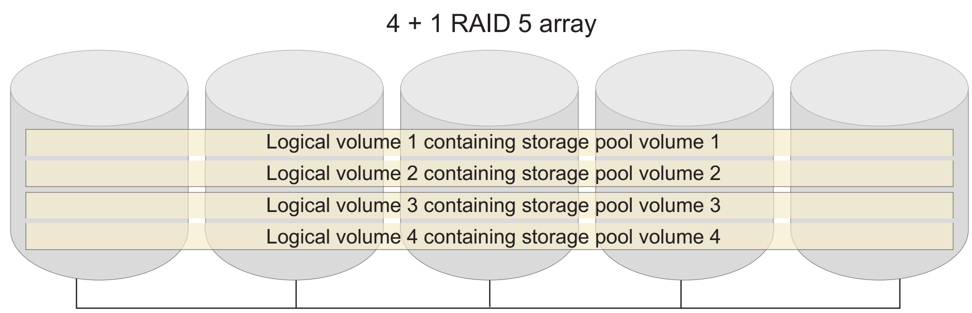

Figure 28. Storage pool layout with 4 + 1 RAID 5

The layout has the following advantages:

- The layout follows the stripe size recommendation (full stripe write).
- The layout follows the guideline that an *n* + 1 RAID array has no more than *n* volumes.

The disadvantage is that there is a performance cost for the write-parity operations. The write-parity operations might not cause a performance problem if the cache for the disk system is adequate.

Tuning Storwize V7000 and V3700 systems

The IBM Storwize V7000 is an ideal disk system for use with Tivoli Storage Manager. You can configure the system with multiple device types, so that you can use fast disk or SSD for the server database and lower-cost, higher-capacity disk for storage pools.

About this task

The Storwize V7000 includes the Easy Tier® function, which automatically and nondisruptively moves frequently accessed data from HDD to SSD disks. With this function, you can configure the system to gain most of the benefits of SSD speed for the server database without having the entire database on SSD.

The IBM Storwize V3700 is an entry level system with capabilities that are similar to the Storwize V7000. The Storwize V3700 is also a good disk system to use with Tivoli Storage Manager.

Example configuration with Storwize V7000 systems

The example illustrates how you can configure IBM Storwize V7000 systems for a Tivoli Storage Manager server. In this example, the server needs 2 TB for its database and 58 TB for its storage pool on disk.

The following components are used in the configuration:

- One Storwize V7000 Disk Control Enclosure, which can hold 24 2.5-inch disk units
- Three Storwize V7000 Disk Expansion Enclosures, each of which can hold 24 2.5-inch disk units

Together, these enclosures can hold up to 96 disk units. See the following table for the disk specifications and configuration. All disks are 2.5-inch disks.

Server storage requirement	Disk capacity	Disk speed and type	Disk quantity	RAID type	RAID array quantity	Usage
Database	300 GB	15k rpm SAS HDD	12	4 + P RAID 5	2	These disks provide space for the database on two 4 + P RAID 5 arrays, with two spare disks. For optimal performance, use the Easy Tier function with SSD disk units for the database. Use the Easy Tier function for the database only, not for the logs or the storage pool. If you are using data deduplication, node replication, or both, either substitute SSD disk units for the database or add some SSD units and use the Easy Tier function.
Active log and archive log	300 GB	15k rpm SAS HDD	4	RAID 0	2	These disks provide space for the archive log and active log on two RAID 0 arrays. Spare disks are shared with the disks for the database.
Storage pool	900 GB	10k rpm SAS HDD	80	6 + P RAID 5	11	These disks provide space for a 58 TB storage pool. The disks are configured in eleven 6 + P RAID 5 arrays, with three spare disks. If the throughput for the storage pool does not require the higher speed of the 10k rpm drives, you can use 7.2k rpm nearline-SAS HDD drives. Verify that the slower disks can meet throughput requirements. If greater storage pool capacity is required, add more enclosures.

Configuring the operating system for disk performance

The operating system configuration and the types of file systems in use affect the performance of disks. Learn about how to configure these items for the best performance for Tivoli Storage Manager server.

About this task

Parameters that affect disk performance vary by operating system.

Configuring AIX systems for disk performance

Use JFS2 file systems for Tivoli Storage Manager in most cases. Examine the queue depth for disks that are in use.

Procedure

- Use JFS2 file systems for the Tivoli Storage Manager database, recovery logs, and disk storage pools, unless you are using disk for LAN-free operations. If you are using disk for LAN-free operations, then use General Parallel File System (GPFS™) for the shared storage pools.

  Use the JFS2 **rbrw** option on the **mount** command, especially if Tivoli Storage Manager database backups are stored to a `FILE` device class.
- The default queue depths for non-IBM disks that are used for Tivoli Storage Manager are often low by default. If the queue depth is less than 32, see the documentation for the disk system or contact the manufacturer about guidelines for the queue depth. To change the queue depth, see the AIX product information.

Configuring Linux systems for disk performance

When you configure disks for Tivoli Storage Manager, use the Logical Volume Manager (LVM).

Procedure

- Use Linux Logical Volume Manager (LVM) to create logical volumes on the disk LUNs for all disks that are used for Tivoli Storage Manager components.

  Set the LVM read-ahead to 0 for all logical volumes on disk systems that provide adaptive read-ahead capabilities, for example, enterprise-type disk systems.

  If more space is needed, the logical volumes provide an easy way to extend the volumes and file systems. LVM also provides striping, which can be used to improve sequential I/O performance.
- For the Tivoli Storage Manager database and logs, use either the ext3 or ext4 file system. As a best practice, use the following file system that is appropriate for your operating system and level:
  - For Red Hat Enterprise Linux x86_64, use the ext3 or ext4 file system. Use the ext4 file system only if Red Hat Enterprise Linux 6.4 or later is installed.
  - For SUSE Linux Enterprise Server and for Red Hat Enterprise Linux ppc64, use the ext3 file system.
- For Tivoli Storage Manager storage pools, use the ext4 file system. The ext4 file system has the following advantages for use with storage pools:
  - You do not have to write out each block I/O to allocate the storage pool volume, which improves the performance of the **DEFINE VOLUME** command.
  - You can avoid file and free space fragmentation, which improves read and write performance.

– When you define new volumes, the Tivoli Storage Manager server activities that are running are not negatively affected.

Configuring Oracle Solaris systems for disk performance

For best performance for disks that are used for Tivoli Storage Manager, configure the system according to a few guidelines.

Procedure

- Use the latest Solaris version and release, and apply the latest kernel patch. By using the latest release, you can take advantage of the latest improvements to functions such as ZFS.
- Tune the record size. The record size property specifies a suggested block size for files in the file system. This property is designed only for use with database workloads that access files in fixed-size records. Specifying a record size value greater than or equal to the record size of the database can result in significant performance gains. Changing the record size of the file system affects only files that are created after the change. Specify a number that is a power of 2, in the range of 512 bytes - 128 KB.
  1. For the file system where the Tivoli Storage Manager database is stored, set the record size equal to the database page size, which is 8 - 32 KB.

     When the database page size is less than the page size of the operating system, set the record size as the page size of the operating system. Page size is 8 KB on SPARC systems and 4 KB on x64 systems.
  2. For the recovery logs for the Tivoli Storage Manager server, use a distinct file system or preferably a distinct pool with the default record size of 128 KB.
- Keep sufficient free space in the zpool. For the Tivoli Storage Manager database, set a quota on the zpool to ensure that 20% of the zpool space is always available. Performance can be affected when more than 80% of the zpool capacity is in use.
- Use a separate, dedicated zpool for recovery log files. When you use a dedicated LUN on dedicated disks for log files, ZFS can handle the latency-sensitive log activity in priority. Dedicated LUNs help the system handle the database workload with a high transaction rate.
- Set a maximum limit for the Adjustable Replacement Cache (ARC) size, which is used by ZFS to cache the data from all pools. The cache dynamically shrinks or grows based on memory usage on the system. If the server requires a large amount of memory, set a limit on the maximum size for the ARC. With the limit on the ARC size, ZFS is less likely to shrink its cache in periods of peak memory demand.

  To estimate the maximum size for the ZFS ARC, complete the following calculation:
  1. Add the memory usage requirements for applications that use a large amount of memory on the system.
  2. From the size of the physical memory of the system, subtract the memory usage that you obtained in step 1.
  3. Multiply the result of step 2 by 80%. Use this number as the value for the **zfs_arc_max** parameter.

  For example, to limit the ARC size to 30 GB, set the **zfs:zfs_arc_max** parameter in the /etc/system file:

  ```
  set zfs:zfs_arc_max = 32212254720
  ```

  You can use the **kstat** command to verify the size of the ZFS cache.

- For the file system where the Tivoli Storage Manager database is stored, set `primarycache=metadata`. The **`primarycache`** parameter controls what is cached in the ARC, and the value `metadata` means that the ARC is used for metadata only.
- Adjust the ZFS I/O queue size. You control the I/O queue depth for a ZFS LUN with the **`zfs_vdev_max_pending`** parameter.

  When there are many LUNs in a ZFS pool, and each LUN contains many physical disks, the ZFS I/O queue can limit the read operations. The ZFS I/O queue size is set to 35 by default, which means that there are 35 concurrent I/O streams per LUN. A high number of LUNs in a zpool can cause contention on the storage channel and lead to longer response times. To relieve the contention and reduce the time, you can adjust the ZFS I/O queue size dynamically. In the following example, the queue size is reduced to 10:

  ```
  echo zfs_vdev_max_pending/W0t10 | mdb -kw
  ```

  You can check the average queue depth by using the **`iostat`** command and observing the number of entries in the `actv` and `wait` queues.

Chapter 13. Tuning client performance

You can optimize the performance of Tivoli Storage Manager clients. Review the methods for backing up data and select the best methods for your environment. Review the information about client options and adjust configuration settings as needed.

Selecting the optimal client backup method

You can use several techniques with the backup-archive client to help you ensure the best performance during various types of backup processing.

About this task

For most situations, incremental backup is the most comprehensive backup method to use. It is the best method for detecting file changes in your local system, and provides the capability to run individual file restores.

However, there are two major factors that can prevent you from completing incremental backups:

- The amount of available memory
- The backup window duration

Incremental backups can be memory intensive because the memory for an incremental backup holds the list of files that are known to the Tivoli Storage Manager server. Therefore, the memory that is required for an incremental backup is proportional to the number of files in the file system that are being backed up. If the system does not have enough memory, the incremental backups can fail. The time that it takes the client to scan the file system and the amount of changed data can also contribute to backup processing that does not complete within the scheduled backup window. Memory-efficient and journal-based backups do not maintain the whole file list in memory.

Use the following general guidelines to resolve memory and backup window issues.

Guideline	More information
1. Resolve any memory issues first. You must resolve memory issues before you can resolve a backup window issue. For example, you can use the `memoryefficientbackup yes` or `memoryefficient diskcachemethod` options to reduce the memory requirements for backing up file systems that contain millions of files.	See "Reduce client memory use" on page 196
2. Resolve any backup window issues. For example, if the number of files that change daily is relatively small, you can use journal-based backup.	See "Tuning journal-based backups" on page 213
3. If you cannot otherwise resolve memory issues, consider using image backups.	See Image backup

Deciding which backup method to use

Many backup techniques are available in the backup-archive client. Begin with progressive incremental backups and move to other types of incremental backups or image backups if necessary.

Procedure

Use the following table to determine the backup method that you want to use. The table contains common backup scenarios that you might encounter and the suggested backup methods to use.

Scenario	Use this backup method
I want to run the most comprehensive type of file backup on my client system.	• "Progressive incremental backup" on page 175 • "Incremental-by-date backup" on page 179
I want to take advantage of the benefits of progressive incremental backups, but I am running into memory issues.	• "Memory-efficient backup" on page 177 • "Memory-efficient backup with disk caching" on page 178
I have many small files in my file system with a few changes, but incremental backup processing does not complete within the allotted time.	"Journal-based backup" on page 176
On operating systems such as AIX or Linux, I have large file systems that can be divided into logical partitions. I want to be able to provide a direct path to the files that I want to back up.	"Backup of virtual mount points" on page 178
Scheduled backups do not complete within the allotted time. I have an application that can provide a list of files that changed since the previous backup. I want to improve the speed of the backup process by backing up just this list of changed files.	"File list backup" on page 180
I want to increase the throughput of incremental backup operations and restore operations.	"Multiple session backup" on page 180
I might be traveling to a location where a high-speed network is not available, and I might have to rely on a communication device that has limited bandwidth.	"Adaptive subfile backup" on page 181
I tried to use several types of incremental backups but scheduled backup processing does not complete within the allotted time.	• "Image backup" on page 182 • "Image plus incremental-by-date image backup" on page 183 • "Image plus incremental backup" on page 184
I want to streamline the backups of the volumes on my NetApp vFiler.	"Snapshot differential backup" on page 185
I want to back up my virtual machines that are hosted on a Hyper-V server.	"Windows Hyper-V backup" on page 186

Scenario	Use this backup method
I want to back up my VMware virtual machines, but I do not have a license to use Tivoli Storage Manager for Virtual Environments.	• "Full backups for virtual machines" on page 187 • "VMware file-level backup on Windows" on page 189
I want to back up my VMware virtual machines and I want to use my licensed Tivoli Storage Manager for Virtual Environments software.	• "Full backups for virtual machines" on page 187 • "VMware incremental backup" on page 187 • "Virtual machine incremental-forever-full backup" on page 188 • "Virtual machine incremental-forever-incremental backup" on page 188
I want to back up my Microsoft Hyper-V virtual machine, but I do not have a license to use Tivoli Storage Manager for Virtual Environments.	For more information, see Back up virtual machines on a Hyper-V system.
I want to back up my Microsoft Hyper-V virtual machine and I want to use my licensed Tivoli Storage Manager for Virtual Environments software.	For more information, see Back up virtual machines on a Hyper-V system.
I want to improve the speed of my VMware virtual machine backups by backing up several virtual machines at the same time.	Back up virtual machines on a Hyper-V system

File backup techniques

If you are backing up your system on a file-by-file basis, you can use several backup techniques.

Use the following information to determine which file backup technique to use that best meets your needs.

Progressive incremental backup

Progressive incremental backup is the standard method of backup that is used by Tivoli Storage Manager. Incremental backup processing backs up only those files that changed since the last full or incremental backup, unless the files are excluded from backup.

How it works

The following processes occur during an incremental backup:

- The client queries the Tivoli Storage Manager server for active backup version metadata.
- The server returns a list of active backup versions for the entire file system.
- The client scans and compares the list with the local file system to determine which files are new or changed since the last backup.
- The client backs up the new and changed files.

When to use

Use incremental backup when the system is not constrained by memory, backup window duration, or other operational issues. Incremental backup is the default backup method.

Advantages

Incremental backup processing has the following advantages:

- This method is the most comprehensive backup method for Tivoli Storage Manager.
- No redundant backups are made. You back up only what is changed.
- There is less network utilization because unchanged files do not have to be sent over the network.
- This method is a form of single-instance storage because a file is not backed up again if it does not change. Incremental backups are more efficient and save space on the server storage pools.
- Files are easier to restore because you do not have to restore the base backup version first and apply incremental or differential changes.

Disadvantages

Incremental backups processing has the following disadvantages:

- The client system might run out of memory if the number of active backup versions is too large.
- The time that it takes to scan file systems that contain millions of files can exceed the duration of the backup window.

If incremental backup operations do not complete successfully, try other variations of incremental backup:

- "Journal-based backup"
- "Memory-efficient backup" on page 177
- "Memory-efficient backup with disk caching" on page 178
- "Backup of virtual mount points" on page 178
- "Incremental-by-date backup" on page 179
- "File list backup" on page 180
- "Multiple session backup" on page 180

Journal-based backup

Journal-based backup is an alternative form of incremental backup that uses a change journal that is maintained by the Tivoli Storage Manager journal process. On Windows clients, the change journal is maintained by a journal service. On AIX and Linux clients, the change journal is maintained by a journal daemon process.

How it works

The following processes occur during journal-based backup processing:

- Journal-based backup processing uses real-time monitoring of a file system for changed files.
- The names of the changed files are logged to the journal database.
- During backup processing, the client queries the journal for the list of changed files, and then backs up the changed files.

When to use

Use journal-based backup in the following situations:

- The scheduled backups are not completed within the allotted time.
- There are less than 1,000,000 files and a small number of changes between backups (less than 1,000,000).

- There are less than 10,000,000 objects with 10-15% velocity of change. The velocity of change means the rate at which files are changed over a short amount of time (such 1 or 2 seconds).

Advantages
Journal-based backup can often greatly reduce the time that it takes to determine which files changed.

Disadvantages
Journal-based backup processing has the following limitations:
- You must still run incremental backups periodically.
- Journal-based backups are not suitable for file systems where large numbers of files can change over a short time interval, such as changing hundreds or thousands of files in 1 or 2 seconds.
- This method is available only on Windows, AIX, and Linux clients.

Related task: "Tuning journal-based backups" on page 213

Memory-efficient backup

The performance of incremental backups can degrade if the system is memory-constrained before the backup begins. Run incremental backup with the **memoryefficientbackup** yes option in the client options file. This setting causes the client to process only one directory at a time during incremental backups, which reduces memory consumption but increases backup time.

How it works
The following processes occur during an incremental backup with the memory-efficient setting:
- The client queries the server for the metadata of active backup versions for the first directory to be backed up.
- The server returns a list of active backup versions for the directory.
- The client scans the list and compares it with the local file system, and backs up the new and changed files.
- The client queries the server for the next directory and repeats the process for all directories.

When to use
Use memory-efficient backup when your system has a low amount of memory available for incremental backups.

Advantages
Memory-efficient backup is a comprehensive incremental backup with a smaller backup memory footprint.

Disadvantages
Memory-efficient backup processing has the following disadvantages:
- The backup run time is increased.
- This method does not work for a single directory that contains a large number of files.
- If the system is not memory-constrained, running memory-efficient backup can degrade the backup performance.

Related task: "Reduce client memory use" on page 196

Memory-efficient backup with disk caching

If your client system is memory-constrained and incremental backups still cannot complete successfully with the **memoryefficientbackup** yes setting, run incremental backups with the **memoryefficientbackup** diskcachemethod option. This setting causes the client to use less memory but requires more disk space on the client system.

How it works
This method is similar to incremental backup processing but the client temporarily stores active backup version metadata on disk instead of memory.

When to use
Use memory-efficient backup with disk caching in the following situations:

- The client is running out of memory with incremental backups and memory-efficient backup is not sufficient.
- Journal-based backup is not available on the operating system.

Advantages
Memory-efficient backup with disk caching is a comprehensive incremental backup operation with a smaller backup memory footprint.

Disadvantages
Memory-efficient backup processing with disk caching has the following disadvantages:

- The backup processing time might be longer because the active backup inventory is on disk instead of in memory.
- Gigabytes of free disk space are required to temporarily cache the active backup inventory.

Related task: "Reduce client memory use" on page 196

Backup of virtual mount points

You can save processing time when you define a virtual mount point within a file system because it provides a direct path to the files that you want to back up.

How it works
The following processes occur during the backup of virtual mount points:

- Instead of backing up an entire file system to a single file space on the server, you can logically partition a large file system into smaller file systems, and then define mount points for backup processing.
- The file system that are represented by the mount points can be managed as separate file spaces on the server.

When to use
Use virtual mount points to back up large, balanced, AIX, HP-UX, Linux, and Solaris file systems that can be efficiently divided into logical partitions.

Advantages
Backup processing of virtual mount points provides a balanced approach to the backup of large file systems by effectively dividing them into smaller file systems. It is more efficient than defining the file system with the **domain** option, and then specifying the **exclude** option to exclude the files you do not want to back up.

Disadvantages

Backup processing of virtual mount points has the following limitations:

- This method of backup processing is not appropriate for a single directory that contains a large number of files.
- Virtual mount points are static and cannot be changed.
- This method requires monitoring to ensure that new directories are still backed up in one of the virtual mount points, along with other processing that is required to maintain the virtual mount point definitions.
- Command-line restore operations require the use of braces ({ }) to delimit the virtual mount point name in the file specification.
- This method is only available for AIX, HP-UX, Linux, and Solaris operating systems.

Related concept: "File space tuning" on page 221

Incremental-by-date backup

This backup method backs up new and changed files that have a modification date later than the date of the last incremental backup that is stored at the server, unless the files are excluded from backup.

How it works

The following processes occur during an incremental-by-date backup:

- The client queries the server for the most recent backup of the entire file system.
- The server returns the time stamp of the most recent backup of the entire file system.
- The client scans and compares the list from the server with the local file system and backs up the new and changed files that are based on the time stamp of the most recent backup.

When to use

Use incremental-by-date backup in the following situations:

- The scheduled backups are not completed within the allotted time.
- The changes to the file system are additive or changing, but not deleted.
- You also run weekly (or periodic) full incremental backups.

Advantages

Incremental-by-date backup processing has the following benefits:

- This method reduces the time that it takes to determine which files changed.
- This method removes the processing time on the server that is used to query the database for changed files.
- This method removes the network traffic that is used to communicate the query results.

Disadvantages

Incremental-by-date backup processing has the following disadvantages:

- This method reduces the flexibility of the scope of the backup operation. You must back up the entire file system.
- The files are not backed up if the changes do not affect the date (for example, attribute, mode, ACL, rename, copy, move, and security changes).

- The deleted files are not expired on the server.
- Policy rebinding does not take place.
- The entire file system must be scanned.
- This method cannot be used if the client and server clocks are set to different times or are not in the same time zone.

Related information: Incremental-by-date backup

File list backup

You can control which files are backed when run a backup with the **filelist** option.

How it works
File list backup can be used in the following manner:
- An application creates a list of files for backup and passes the list to the client.
- The client runs a selective backup of the files that are specified in the list.

When to use
Use file list backup in the following situations:
- The scheduled backups are not completing within the allotted time.
- The list of changed files is known.

Advantages
Selective backup eliminates the query of the server database and the scan of local file system.

Disadvantages
File list backup has the following disadvantages:
- You must find a way to create the file list.
- You must explicitly specify the files. You cannot use wildcard characters or directory recursion in the file list.
- Large file lists have an affect on memory requirements during client restore and retrieve operations.

Related reference: Filelist

Multiple session backup

The backup-archive client can run concurrent sessions to back up and restore data to help improve performance. During incremental backup processing, the client can process multiple objects in parallel by opening more than one session with the Tivoli Storage Manager server.

How it works
Multiple sessions are used when you specify multiple file specifications on a **backup**, **restore**, **archive**, or **retrieve** command. For example, you can start a multiple session backup with the following command:
- On the AIX, HP-UX, Linux, Mac OS X, or Solaris client:

  ```
  incr /Volumes/filespace_A /Volumes/filespace_B
  ```
- On the Windows client:

  ```
  incr c: d:
  ```

The **resourceutilization** option is used to regulate the level of resources that the Tivoli Storage Manager server and client can use during processing. The default is to use a maximum of two sessions, one session to query the server and one session to send file data.

When to use
Use multiple backup sessions when you want to increase client performance, and you have sufficient client and server resources and processing capacity. For example, the server and client hardware must have sufficient memory, storage, and processor capacity to support multiple sessions. The network bandwidth must also be sufficient to handle the increased amount of data that flows across the network.

Advantages
Using more than one backup session can often lead to overall improvements in throughput.

Disadvantages
Running multiple backup sessions has the following disadvantages. Some workarounds are included.

- During a multiple session backup operation, files from one file specification might be stored on multiple tapes on the server and interspersed with files from different file specifications. This arrangement can decrease restore performance.

  To avoid the performance degradation in restore operations, set the **collocatebyfilespec** option to yes. This setting eliminates the interspersing of files from different file specifications by limiting the client to one server session for each file specification. Therefore, if the data is stored to tape, the files for each file specification are stored together on one tape, unless another tape is required for more capacity.
- The client might produce multiple accounting records.
- The server might not start enough concurrent sessions. To avoid this situation, the **maxsessions** server parameter must be reviewed and possibly changed.
- A **query node** command might not summarize the client activity.

Related task: "Improving client performance by using multiple sessions" on page 208

Adaptive subfile backup

If you plan to back up your files over a network device with limited bandwidth, you can reduce network traffic by using adaptive subfile backup.

How it works
Adaptive subfile backup processing backs up the changed portion of a file on Windows clients.

When to use
Use adaptive subfile backup in the following situations:

- The network is constrained.
- The file sizes are small (less than 2 GB in size).

Advantages
Adaptive subfile backup processing has the following benefits:

- Faster throughput.
- Reduced storage pool consumption.

Disadvantages

Adaptive subfile backup processing has the following disadvantages:

- This method uses a large amount of local cache space.
- Some processing time is required during the backup.
- The restore operations can take longer because of the base file and delta files are restored.
- The client can run out of disk space during the restore if disk space is constrained because of how files are reconstructed from the base files and the delta files.

Related task: Performing a backup with limited bandwidth

Image backup techniques

If the different variations of progressive incremental backup and file backup operations do not complete successfully, consider running an image backup to reduce the backup window.

Image backup

Image backup processing backs up your file system as a single object.

How it works

During image backup processing, the client sends a logical block image of a file system to the Tivoli Storage Manager server.

When to use

Use image backup processing in the following situations:

- You cannot resolve system memory problems or progressive incremental backup is otherwise unusable.
- There are too many changes in the file system (greater than 1,000,000 objects) for journal-based backup.
- Most of the file system contains small files (average size less than 1 MB).
- You must have a faster recovery time than what can be achieved with file-level restore.
- For AIX, HP-UX, Linux, and Solaris clients:
  - The file system is at least 60% full.
  - Online image backup is not available, and you can unmount the file system.

Advantages

Image backup processing has the following benefits:

- Backups are faster.
- No scan time is required to determine what changed.
- Overall data movement is faster.
- Restore times are faster.

Disadvantages

You cannot restore individual files directly from the Tivoli Storage Manager server.

The following variations of image backup are available:

- Offline (static) image backup processing
  - The volumes to be backed up are mounted read-only.

- This method is available for AIX, HP-UX, Linux x86, Solaris, and Windows operating systems.
- This method is the most effective backup method for FlashCopy operations.

- Online (dynamic) image backup processing
  - The volumes to be backed up remain online.
  - Fuzzy backup processing occurs when the data is changed during the image backup processing.
- Online image backup by using snapshots
  - The volumes to be backed up remain online.
  - The image backup is made at a single point in time.
  - It is available only for AIX JFS2, Linux x86, and Windows operating systems.

Image plus incremental-by-date image backup

Image backup plus incremental-by-date image backup processing is one of two methods that you can use to run efficient incremental backups of your file system.

How it works

The following processes occur during image plus incremental-by-date image backup processing:

- During a full image backup (for example, when you issue the `dsmc backup image` command), the client sends a logical block image of a file system to the server.
- Subsequent backups are incremental-by-date image backups (for example, when you issue the `dsmc backup image -mode=incremental` command), in which the client queries the server for the last backup of the entire file system.
- The server sends the time stamp of last backup of the entire file system to the client.
- The client scans and compares the time stamp with the local file system, and backs up the new and changed files.

During an image plus incremental-by-date restore operation, the following processes occur:

- The client requests an incremental image restore.
- The server sends the base image to the client.
- The server returns more files that must be applied to the base image to satisfy the recovery point.

When to use

Run image plus incremental-by-date image backup processing in the following situations:

- You need faster backups.
- You must be able to restore files to a specific point in time.

Tip: Periodically run full image backups to maintain a file system image that is close to what existed at the time of the last incremental-by-date image backup. When you periodically run a full image backup, it can also improve restore time.

Advantages

Image plus incremental backup processing has the following advantages:

- Backups are faster.
- No scan time is required to determine what changed.
- Overall data movement is faster.
- Restore times are faster.
- Protection of files that changed after the image backup was created.
- In some cases, recovery time and recovery point objectives are improved.

Disadvantages

Image plus incremental-by-date image backup processing has the following limitations:

- This method reduces the flexibility of the scope of the backup operation. You must back up the entire file system.
- The files are not backed up if the changes do not affect the date (for example, attribute, mode, ACL, rename, copy, move, and security changes).
- The deleted files are not expired on the server.
- Policy rebinding does not take place.
- The entire file system must be scanned.
- This method cannot be used if the client and server clocks are set to different times or are not in the same time zone.
- Deleted files are not reconciled. Deleted files are not expired on the server. Therefore, when you restore an image with the incremental option, files that were deleted after the original image backup are present after the restore.
- More storage space is required on the Tivoli Storage Manager server.

Image plus incremental backup

Image backup plus file system incremental backup processing is the second method that you can use to run efficient incremental backups of your file system.

How it works

The following processes during image plus incremental backup processing:

- During a full image backup (for example, when you issue the `dsmc backup image` command), the client sends a logical block image of a file system to the server.
- Subsequent backups are progressive incremental backups in which the client queries server for the active backup version metadata.
- The server returns list of active backup versions for the entire file system.
- The client scans and compares the list with the local file system.
- The client backs up the new and changed files.

During an image plus progressive incremental restore operation, the following processes occur:

- The client requests an incremental image restore.
- The server returns the base image.
- The server returns more files that must be applied to the base image to satisfy the recovery point.
- The server optionally returns the list of files that must be deleted from the base image.

When to use

Use image plus incremental backup processing in the following situations:

- You need faster backups.
- You want to restore files to a specific point in time.
- You want to be able to reconcile deleted files.

Tips:

- Run incremental backups of the file system periodically to ensure that the server records additions and deletions accurately.
- Run an image backup periodically to ensure faster restores.

Advantages

Image plus incremental backup processing has the following advantages:

- Backups are faster.
- No scan time is required to determine what changed.
- Overall data movement is faster.
- Restore times are faster.
- Protection of files that changed after the image backup was created.
- In some cases, recovery time and recovery point objectives are improved.

Disadvantages

Image plus incremental backup processing has the following disadvantages:

- More time is required to periodically create image backups.
- More storage space is required on the Tivoli Storage Manager server.

Snapshot differential backup

If you are backing up NetApp filer or vFiler volumes or N-Series file server volumes, you can use a snapshot differential backup to streamline the incremental backup process.

How it works

The following processes occur during snapshot differential backup processing:

- The first time that you run an incremental backup with the **snapdiff** option, a snapshot is created (the base snapshot) and a traditional incremental backup is run by using this snapshot as the source. The name of the snapshot that is created is recorded in the Tivoli Storage Manager database.
- The second time an incremental backup is run with the **snapdiff** option, a newer snapshot is either created, or an existing one is used to find the differences between these two snapshots. The second snapshot is called `diffsnapshot`. The client then incrementally backs up the files that are reported as changed by NetApp to the Tivoli Storage Manager server.

When to use

Use this method to back up NetApp filer or vFiler volumes or N-Series file server volumes on Windows, AIX 64-bit, and Linux x86/86_64 clients.

Advantages

Snapshot differential backup processing can save you time by not having to scan the whole volume for changed files.

Disadvantages

Snapshot differential backup processing has the following limitations:

- On Windows systems, it does not work for any of the NetApp predefined shares, including `C$`, because the client cannot determine their mount points programmatically.
- You must periodically take a new base snapshot with the **`createnewbase`** option to back up any files that might have been skipped.

Virtual machine backup techniques

You can use several backup techniques to back up virtual machines.

Use the following information to determine which virtual machine backup technique to use that best meets your needs and environment.

Windows Hyper-V backup

The client can back up virtual machines that are hosted by a Hyper-V server on Windows systems.

How it works

For Hyper-V servers that run on Windows Server 2008 or Windows Server 2012 systems, full virtual machine backup processing backs up snapshots of Hyper-V virtual machines by using the Microsoft Volume Shadow Copy Services (VSS) interface. You can restore a virtual machine by using the files that are created by VSS.

Advantages

Hyper-V backup processing has the following advantages:

- This method can back up data without stopping the virtual machine or stopping any running applications within the virtual machine when the guest virtual machine is running a VSS-enabled Windows operating system.
- This method can restore either individual virtual machines or a group of virtual machines that run on a Hyper-V server for disaster recovery purposes.
- This method adds backup and restore functions for guest operating systems without the need for you to install a Tivoli Storage Manager client on the guest virtual machine.
- This method can be used for both disaster recovery and long-term data backup support.

Disadvantages

Hyper-V backup processing has the following disadvantages:

- Backups are not granular.
- You cannot run individual file restore operations from a full virtual machine backup.

When a Hyper-V server runs on a Windows Server 2012 R2, or a newer version of a Windows Server operating system, you can also use the incremental-forever backup techniques. To use incremental-forever backup methods, you must have a license for Tivoli Storage Manager Data Protection for Hyper-V. With this license, file-level recovery is also possible. For complete information about creating incremental-forever backups of Hyper-V virtual machines, or for information about using the Tivoli Storage Manager recovery agent to restore files from a backed up Hyper-V virtual machine, see the IBM Knowledge Center topics at http://www.ibm.com/support/knowledgecenter/SS8TDQ_7.1.1.

Full backups for virtual machines

Full virtual machine backup processing backs up a virtual machine from a virtual machine-based host. For backups that use Hyper-V servers, you must use one of the following operating systems:

- Windows Server 2008
- Windows Server 2012

How it works
Full virtual machine backup processing stores a backup copy of all virtual disk images and configuration information for a virtual machine.

Advantages
With full virtual machine backup processing, you get faster data movement than a file-level backup.

Disadvantages
Full virtual machine backup processing has the following disadvantages:

- Backups are not granular.
- Full virtual machine backup operations enable a complete restore of a virtual machine, but they take more time and more server space than a file-level or incremental backup.
- You can restore individual files from a full virtual machine backup only with IBM Tivoli Storage Manager for Virtual Environments.
- This method is only available on Linux and Windows clients.

VMware incremental backup

You can run an incremental backup of a virtual machine from a VMware ESX or ESXi-based host. Incremental backup processing requires a Tivoli Storage Manager for Virtual Environments license.

How it works
An incremental backup of a virtual machine backs up all changes that occurred since the previous backup of the virtual machine, whether the backup was a full backup, or another incremental backup.

Advantages
Incremental backup processing backs up changes to virtual machines between full virtual machine backups.

Disadvantages
Incremental backup processing of a virtual machine has the following disadvantages:

- The size of incremental backups can increase if you do not run a full backup regularly.
- It is inefficient to restore data from incremental backups because the process must automatically complete the following tasks:
  - Restore the most recent full backup.
  - Restore each incremental backup up to the specified recovery point.
- This method is available only on Linux and Windows clients.

Virtual machine incremental-forever-full backup

Incremental-forever-full virtual machine backup processing backs up all the used blocks on a virtual machine's disks. To run this type of backup, you must have a license for one of the following products:

- Tivoli Storage Manager for Virtual Environments: Data Protection for VMware V6.4 or later.
- Tivoli Storage Manager for Virtual Environments: Data Protection for Hyper-V V7.1.1 or later. The system that hosts the Hyper-V server must run Windows Server 2012 R2 or a newer Windows Server operating system.

How it works

The following processes occur during incremental-forever-full virtual machine backup processing:

- A full virtual machine backup is required only one time.
- Data from incremental backups is combined with data from the full backup to create a synthetic full backup image. This type of full backup is called a synthetic backup because it is created from the data that is stored on the server and not from reading the used blocks on the production disks.
- Each incremental-forever-full virtual machine backup operation reads and copies all of the used blocks, whether the blocks are changed or not since the previous backup.

Advantages

Incremental-forever-full virtual machine backup processing has the following advantages:

- Periodic full backups are no longer necessary.
- During a restore operation, you can specify options for a point in time and date to recover data. The data is restored from the original full backup and all of the changed blocks that are associated with the data.

Disadvantages

Incremental-forever-full virtual machine backup processing has the following disadvantages:

- If one or more of the progressive incremental backups is corrupted on the server, you might not be able to fully recover a virtual machine. To ensure that you can fully recover a virtual machine, periodically run a full virtual machine backup.
- This method is available only on Linux and Windows clients.

Virtual machine incremental-forever-incremental backup

Incremental-forever-incremental backup processing backs up only the disk blocks that have changed since the last backup. To run this type of backup, you must have a license to use one of the following products:

- Tivoli Storage Manager for Virtual Environments: Data Protection for VMware V6.4 or later.
- Tivoli Storage Manager for Virtual Environments: Data Protection for Hyper-V V7.1.1 or later. The system that hosts the Hyper-V server must run Windows Server 2012 R2 or a newer Windows Server operating system.

How it works

The following processes occur during incremental-forever-incremental backup processing of a virtual machine:

- A full virtual machine backup is required only one time.
- A full virtual machine backup operation copies all of the used disk blocks that are owned by a virtual machine to the Tivoli Storage Manager server.
- After the initial full backup, all subsequent backup operations of the virtual machine are incremental-forever-incremental backups.
- This method copies only the blocks that changed since the previous backup, regardless of the type of the previous backup.
- The server uses a grouping technology that associates the changed blocks from the most recent backup with data that is already stored on the server from previous backups.
- A new full backup is then effectively created each time changed blocks are copied to the server by an incremental-forever-incremental backup.

Advantages

Incremental-forever-incremental backup processing has the following advantages:

- Periodic full virtual machine backups are no longer necessary.
- This method reduces the amount of data that goes across the network.
- This method reduces data growth because all incremental backups contain only the blocks that changed since the previous backup.
- No comparison with the backup target is required since only changed blocks are identified.
- Impact to the client system is minimized.
- The length of the backup window is reduced.
- Data restore operations are simplified.
- This method optimizes data restore operations.

Disadvantages

Incremental-forever-incremental backup processing has the following disadvantages:

- If one or more of the progressive incremental backups is corrupted on the server, you might not be able to fully recover a virtual machine. To ensure that you can fully recover a virtual machine, periodically run a full virtual machine backup.
- It is available only on Linux and Windows clients.

VMware file-level backup on Windows

On Windows systems, you can use the backup-archive client to create file-level backups of VMware virtual machines.

How it works

The following processes occur during file-level backup processing of a virtual machine:

- A VMware snapshot is taken of the virtual machine to be backed up.
- The file systems of the virtual machine are remotely mapped to the vStorage backup server.
- A file-level progressive incremental backup is run for all of the file systems.
- The data is stored under the node name that matches the host name of the virtual machine.

- The data for each virtual machine is stored in the node that is associated with the virtual machine.
- The file system mappings are removed and the snapshot is removed.

When to use
Use file-level virtual machine backup processing if you want to restore individual files from a virtual machine but you do not have a license for Tivoli Storage Manager for Virtual Environments.

Advantages
File-level virtual machine backup processing has the following advantages:
- You can use include and exclude rules to identify the files to back up.
- Files are backed up as individual files rather than as an image backup.

Disadvantages
File-level restores must be made from a backup-archive client that is installed directly on a virtual machine guest.

Parallel backups of virtual machines

You can improve performance of virtual machine backups by running parallel backups of multiple virtual machines by using a single instance of the backup-archive client. This feature is available only in the Tivoli Storage Manager V6.4 or later client.

How it works
The following processes occur during parallel backup processing of virtual machines:
- A single Tivoli Storage Manager data mover node can be used to concurrently back up multiple virtual machines.
- When the backups are initiated, the client establishes parallel sessions to copy the data to the Tivoli Storage Manager server.

Advantages
Parallel virtual machine backup processing has the following advantages:
- The backup window is reduced.
- You can optimize the backups so that they do not adversely affect the servers that are hosting the virtual machines.

Disadvantages
You must optimize the parallel backups. The number of virtual machines that you can back up in parallel depends on the following factors:
- The processing power of the server that the Tivoli Storage Manager data mover node runs on.
- The performance of I/O between the client and the Tivoli Storage Manager server.

Common client performance problems

Typical client performance issues are often related to backup operations that do not complete within the backup window or that send too much data over the network.

Resolving common client performance problems

The table contains common client problems and actions that can help to improve client performance.

Scenario	Solution	More information
During incremental backups, the client gets out of memory errors that cause the operating system to use more virtual memory or use more RAM than the client system can handle. How can I reduce these memory errors and have my backups complete within the backup window?	Update the client system hardware by increasing the system memory. If it is not possible to update the hardware, try running journal-based backups. If more memory is needed, try memory-efficient incremental backups.	For more information, see the following topics: • "Tuning journal-based backups" on page 213 • "Reduce client memory use" on page 196
Journal-based backups do not complete within the backup window. What alternatives can I use?	Try one or more of the following actions: • Use image backups to back up an entire volume as a snapshot • Examine the design of the file system on AIX, HP-UX, Linux, and Solaris operating systems	For information about image backups, see Image backup. For information about tuning client file spaces, see "File space tuning" on page 221.
How can I reduce the amount of client data that is sent across the network to the Tivoli Storage Manager server?	Try one or more of the following methods: • Use compression during backup operations • Use include-exclude options to exclude files from the backup operation • Use client-side deduplication • Use LAN-free backups	For more information, see the following topics: • "Reduce client data flow with compression" on page 198 • "Reducing client data flow with include and exclude options" on page 202 • "Checklist for data deduplication" on page 30 • "Performance tuning for LAN-free environments" on page 230
Some backup-archive clients back up much of the same data daily. How can I prevent the data that is a duplicate of the data from the previous day from being resent to the server?	Run incremental backups, use client-side data deduplication, or both.	"Checklist for data deduplication" on page 30
We have limited network bandwidth. How can I improve the communication between the client and Tivoli Storage Manager server?	Fine-tune the network and communication settings.	Chapter 14, "Tuning network performance," on page 231

Scenario	Solution	More information
What other methods can I use to reduce the time that it takes to back up a client?	Try one of the following actions: • Use multiple client sessions for backup operations • Set the **resourceutilization** option to optimize the number of multiple sessions	For more information, see the following topics: • "Running concurrent client sessions" on page 208 • "Multiple session backup and restore" on page 209 • "Optimizing the number of multiple sessions to run" on page 210

Resolving common performance problems with virtual machine backup operations

The table contains common questions and situations about virtual machine backup operations and solutions that can help to improve the performance.

The information in the following table, unless otherwise stated, applies to virtual machine backup operations on both the standard backup-archive client and Tivoli Storage Manager for Virtual Environments.

Scenario	Solution	More information
Excluding Tivoli Storage Manager settings, what settings can improve the efficiency of VMware operations with virtual guests?	The VMware Changed Block Tracking (CBT) feature for virtual machines on the vSphere client is enabled automatically by Tivoli Storage Manager. This feature can reduce the data that is sent to the Tivoli Storage Manager server during incremental backups. CBT can reduce the total backup time and the storage pool size on the Tivoli Storage Manager server. CBT might increase the resource utilization slightly on vSphere hosts.	Review information about CBT. Go to the http://www.vmware.com/ and search for information about Changed Block Tracking (CBT) on virtual machines.
What is the best network adapter type to use for a system that is running as a VMware guest?	When you are running a Tivoli Storage Manager Windows client in a VMware guest environment, use the VMXNET Generation 3 (VMXNET 3) network adapter type. The VMXNET 3 adapter is a virtual network device from VMware that is optimized to provide enhanced hardware and software performance in a virtual environment.	Review information about the VMXNET 3 adapter. Go to the http://www.vmware.com/ and search for information about the VMXNET 3 network adapter.

Scenario	Solution	More information
I am running parallel backups of virtual machines. How do I reduce the processor load during parallel backups, and also improve the throughput from the Tivoli Storage Manager backup-archive client to the Tivoli Storage Manager server?	Optimize parallel backups by using the following client options: • **vmmaxparallel** can be used with VMware and Microsoft Hyper-V virtual machines • **vmlimitperhost** can be used with VMware only • **vmlimitperdatastore** can be used with VMware only	For more information, see the following topics: • "Optimizing parallel backups of virtual machines" on page 224 • Vmmaxparallel client option • Vmlimitperhost client option • Vmlimitperdatastore client option
How do I select the best transport mode for virtual backups?	The optimal transport mode to use depends on the composition of the backup environment. Use the **vmvstortransport** option to specify the preferred order of transport modes to use during backup or restore operations of VMware virtual machines.	For more information, see the following topics: • "Selecting a transport mode for VMware backups" on page 227 • Vmvstortransport client option
For Tivoli Storage Manager for Virtual Environments, how can I fine-tune the scalability of incremental-forever backups of virtual guests?	Virtual machine disk files are stored on the Tivoli Storage Manager as blocks of data called megablocks. When a change occurs on a disk in an area that is represented by a megablock, a Tivoli Storage Manager object is created. When a large number of Tivoli Storage Manager objects exist for the same virtual machine data, excessive demands are placed on the Tivoli Storage Manager server database. Use the following options to control the number of Tivoli Storage Manager objects that are created on the server: • **mbobjrefreshthresh** • **mbpctrefreshthresh**	For more information, see the following topics: • "Adjusting the scalability of virtual machine backup operations" on page 228 • Mbobjrefreshthresh client option • Mbpctrefreshthresh client option • Fine-tuning Tivoli Storage Manager server database scalability

Client restart options

The **commrestartduration** and **commrestartinterval** options affect how aggressively the client attempts to reconnect with the server if a network outage or other condition interrupts client and server communications.

Defaults are provided for both options and the defaults work well in most environments. You can experiment with these options to see whether smaller values speed up the reconnection process.

Neither option directly improves client performance, but either option, or both options, can be set to reduce reconnection times so that client data can be processed as soon as possible if the client is disconnected from the server.

commrestartduration client option

The **commrestartduration** option sets the number of seconds that the client waits after a communications failure before it tries to reconnect with a server.

You can set this value to any integer in the range 0 - 9999; the default is 60 seconds.

commrestartinterval client option

The **commrestartinterval** option sets the number of seconds that the client waits between each reconnection attempt.

You can set this value to any integer in the range 0 - 65535; the default is 15 seconds.

Tuning memory

You can tune the client to use less memory during incremental backup operations.

Client memory requirements and ulimit settings for incremental backups

AIX | HP-UX | Linux | Mac OS X | Solaris

The amount of memory that is used for normal incremental backups is proportional to the number of files that are being examined for backup. When you start an incremental backup operation that backs up numerous files, one way to prevent the client from running out of usable memory is by setting the operating system ulimit data value (`ulimit -d`) to **unlimited**.

If your business policies do not support a ulimit value of **unlimited**, you can estimate the system memory that is required for incremental backups by using the following calculations, and then set the ulimit to an appropriate value. Alternatively, you can set the `MEMORYEFFICIENT DISKCACHEMETHOD` client option or use journal-based backups to reduce memory requirements.

Tip: Your file system configuration can affect memory use.

To estimate the memory that is needed for an incremental backup, which is based on the number of objects (files and directories) that exist on the file system, complete the following steps:

1. Multiply the total number of objects by 700 to get an estimated number of bytes in the file system. The 700 is an estimate for the number of bytes in each path. For example, if the number of objects is 500000, then 500000 x 700=350000000.

   Tip: The multiplier that is used in these calculations (700) is an estimate of the amount of memory that is needed per object. If your files and directories have file names that are longer than 80 characters, you might need extra memory.
2. Round up the value by 33% or to the next 100 MB from the previous step. In this example, round the 350 MB value to 500 MB. Convert this value to KB (400 x 1024=409600 KB).
3. If you have more than one file system, estimate the memory requirements for each file system, and use the highest of these estimates to set the ulimit data value.

This estimate works when the client RESOURCEUTILIZATION option is set to 4 or less. The estimation is part of a *producer session*. A producer session is a producer thread that scans the file system to search for changed, new, or deleted files. A producer session uses memory, and the number of producer sessions is important for calculating random access memory (RAM).

If you use a RESOURCEUTILIZATION value of 5, 6, or 7, you can have up to two concurrent producer sessions. RESOURCEUTILIZATION values of 8 and 9 can have up to three concurrent producer sessions. If RESOURCEUTILIZATION 10 is set, you can have up to four concurrent producers. You must base the ulimit value on the sum of the number of objects in each of the two, three, or four file systems that have the highest number of objects.

For example, in the following scenario, you have:

- /fs1 contains 500000 objects
- /fs2 contains 400000 objects
- /fs3 contains 50000 objects
- /fs4 contains 40000 objects

If you specify RESOURCEUTILIZATION 6, you can have up to two concurrent producer sessions. Therefore, you must calculate the ulimit value for the file systems with the highest number of objects (500000 and 400000):

(500000 + 400000) * 700 = 630000000 ~= 630 MB. Round up to 700 MB, then convert to KB = 716800. Set the ulimit value to 716800.

Tip: If the number of objects on the file system increases, you must readjust the ulimit value to accommodate the growth.

If you do not set the ulimit value to **unlimited**, or to an estimated value, you can use the following methods to reduce the memory that is needed for incremental backups:

MEMORYEFFICIENTBACKUP DISKCACHEMETHOD
This method uses disk space as if it were system memory. You might be able to use the default ulimit value, but you need free disk space that is available to process objects. For more information about estimating the disk space that is needed for this option, see the backup-archive client documentation. If disk space is limited, you can use the **memoryefficientbackup yes** option setting. This option uses less disk space that the DISKCACHEMETHOD option, but it does decrease incremental backup performance.

Journal-based backup
This method uses journal-based backups. The journal daemon records changes to an object or its attributes in a journal database. During a journal-based backup, the client obtains a list of files that are eligible for backup from the journal database instead of by scanning the entire file system. Journal-based backups reduce the memory that is needed to process incremental backups.

Reduce client memory use

The **memoryefficientbackup** client option determines how much memory the client uses during incremental backup operations. Restricting the amount of memory that the client can use during incremental backups reduces the efficiency of incremental backup processing. The default setting for the **memoryefficientbackup** option is `no`, which does not limit the memory that the client can use.

During an incremental backup, the client determines which objects are new or changed since the last backup, and which objects must be expired on the server. An object, in this context, is a file or a directory.

By default, the client uses memory to create the list of new, changed, or expired objects to be updated by an incremental backup. Using available memory for this process makes incremental backups more efficient by reducing the time that is needed to prepare the list of objects to include in the backup. On client systems that have either limited memory, or that have applications that are not adversely affected if the client uses up the available memory, you can limit how much memory the client uses during incremental backups.

The following settings are available for the **memoryefficientbackup** option:

memoryefficientbackup no
: The client uses an algorithm that does not limit the amount of memory that is used to process an incremental backup. This setting is the default, and it is the most efficient setting for incremental backups in many configurations.

memoryefficientbackup yes
: The client uses an algorithm that requires less memory when it is processing incremental backups. This setting can increase the server workload, which, in turn, increases the time that is required to complete incremental backups. This setting can adversely affect incremental backup performance in configurations where there are many clients that are backing up files to the same server, and where each of the client systems has many file system objects.

memoryefficientbackup diskcachemethod
: The client uses an algorithm that requires even less memory than `memoryefficientbackup yes`.

  With this setting, the client keeps the list of objects to back up or expire on disk, so it uses less memory.

In many configurations, the most efficient setting for incremental backups is `memoryefficientbackup no`. However, if memory is limited on the client systems, or if you see Tivoli Storage Manager error messages that indicate memory errors, consider using a different setting. To determine the setting, review the following guidelines and select the first setting that applies:

- `memoryefficientbackup no`

  To determine whether you have enough memory to use `memoryefficientbackup no`, complete the following steps:

  1. Determine the number of objects in the client file systems.
  2. Round up the number of objects in the file systems, to the next million and divide that number by 1,000,000. Multiply the quotient by 300 MB to determine how to set the **memoryefficientbackup** option.

For example, if the client file systems have 5,202,131 objects, round up this number to 6,000,000. Divide the result by 1 million and assign it to a temporary variable that is called *numfsobjs*. In this example, `numfsobjs=6` (6,000,000/1,000,000=6). Use the value of *numfsobjs* as described in the following calculations:

32-bit clients
: If the value of the *numfsobjs* variable is less than or equal to 5, multiply *numfsobjs* by 300 MB. If the amount of physical memory on the client system is equal to or greater than the product of `numfsobjs x 300 MB`, specify `memoryefficientbackup no` (the default).

64-bit clients
: If the amount of physical memory on your client is equal to or greater than the product of `numfsobjs x 300 MB`, specify `memoryefficientbackup no` (the default).

- `memoryefficientbackup diskcachemethod`

  If the client has at least the following amount of fast, temporary disk storage available for use by the client process, specify `memoryefficientbackup diskcachemethod`.

  – On UNIX and Linux systems, temporary disk space must meet or exceed `numfsobjs x 300 MB`.

  – On Windows systems, temporary disk space must meet or exceed `numfsobjs x 600 MB`.

  – On Mac OS X systems, temporary disk space must meet or exceed `numfsobjs x 1200 MB`.
- If none of the previous conditions apply, use `memoryefficientbackup yes`.

Alternatives to using the memoryefficientbackup client option

To reduce client memory consumption, you can use the following alternatives instead of the setting `memoryefficientbackup yes`.

- Use the client include and exclude options to back up only what is necessary.
- Use journal-based incremental backup on Windows (NTFS), AIX (JFS2), or Linux (all supported file systems) clients.
- Use the **virtualmountpoint** option to define multiple virtual mount points within a single file system, and back up these mount points, sequentially. Virtual mount points can be used on UNIX and Linux systems, but not on Mac OS X.
- Spread the data across multiple file systems and back up these file systems sequentially.
- Use the image backup function to back up the entire volume. Image backups can take less time and resources than incremental backups, on file systems that have many small files.

Tuning client data throughput

Use client options to improve the throughput of client data to IBM Tivoli Storage Manager.

Reduce client data flow with compression

The backup-archive client can compress data before it sends the data to the server. Enabling compression on the client reduces the amount of data that is sent over the network and the space that is needed to store it on the server and storage pools. Two client options determine when and if the client compresses data: **compression** and **compressalways**.

In addition to compressing objects, to reduce the amount of data, you can also consider enabling client-side data deduplication. For information about configuring client-side data deduplication, see "Tuning client-side data deduplication" on page 200.

Related tasks:

"Compressing data to save storage space" on page 125

compression client option

The **compression** client option specifies whether compression is enabled on the Tivoli Storage Manager client. For optimal backup and restore performance with many clients, consider enabling client compression.

Compressing the data on the client reduces demand on the network and the Tivoli Storage Manager server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. If you use node replication, compressed data remains compressed during the transfer from the source server to the target replication server. The data is then stored on the target replication server in the compressed format.

Client compression reduces the performance of each client, and the reduction is more pronounced on the slowest client systems. For optimal backup and restore performance when you have fast clients and a heavily loaded network or server, use client compression. For optimal backup and restore performance when you have a slow client, or a lightly loaded network or server, do not use compression. However, you must consider the trade-off of greater storage requirements on the server when you are not using client compression. The default for the **compression** option is `no`.

Compression can cause severe performance degradation if attempts to compress a file fail. Compression fails when the compressed file is larger than the original file. The client detects this size difference and stops the compression process, fails the transaction, and resends the entire transaction uncompressed. Compression failure occurs because the file type is not suitable for compression or the file is already compressed. Short of turning off compression, there are two options that you can use to reduce or eliminate compression failures:

- Use the **compressalways yes** option. This default option prevents compression reattempts if the compressed file is larger than the uncompressed file.
- Use the **exclude.compression** option in the client options file. This option disables compression for specific files, for example, all `*.gif` files or other files that get larger during attempted compression. Excluding these files saves processor cycles because there is no attempt to compress files that cannot be

compressed. Look in the client output (dsmsched.log) for files that are causing compression reattempts and exclude those file types.

Use the following values for the **compression** option:

- For a single fast client, a fast network, and a fast server:
  `compression no`
- For multiple clients, a slow network, or a slow server:
  `compression yes`

Do not enable the client **compression** option if a client has a built-in file compression feature. For example, if hardware compression is in use for the media where Data Protection for Oracle data is stored, do not enable client compression. Compression, on these types of clients, reduces the amount of data that is backed up to the server.

Restriction: Windows Data can be compressed by using NTFS. However, the data must be decompressed before it can be accessed by the Tivoli Storage Manager server. Therefore, slower backups and higher processor usage might occur if you use NTFS compression.

compressalways client option

The **compressalways** option specifies whether to continue compressing an object if it grows during compression, or resend the object, uncompressed. This option is valid when client compression is enabled by the **compression** option.

The **compressalways** option is used with the **archive**, **incremental**, and **selective** commands. This option can also be defined on the server. If this option is set to yes, the default value, files compression continues even if the file size increases. To stop compression if the file size grows, and resend the file uncompressed, specify compressalways no. This option controls compression only if the administrator specifies that the client node determines the selection. To reduce the impact of repeated compression attempts if the compressed file is larger than the original, specify **compressalways** yes.

To prevent unsuccessful compression attempts, you can list files that cannot be compressed on one or more client **exclude.compression** statements. Exclude files that contain graphics; even exclude word-processing files if they contain embedded graphics. Also, exclude audio files, video files, files that are already encrypted, and files that are saved in an archive format, such as .jar files, .zip files, and other compressed file formats.

Using Tivoli Storage Manager client compression and encryption for the same files is valid. The client first compresses the file data and then encrypts it, so that there is no loss in compression effectiveness that is caused by encryption, and encryption is faster if there is less data to encrypt.

The following example shows how to exclude objects that are already compressed or encrypted, by using **exclude.compression** statements:

```
exclude.compression ?:\...\*.gif
exclude.compression ?:\...\*.jpg
exclude.compression ?:\...\*.zip
exclude.compression ?:\...\*.mp3
exclude.compression ?:\...\*.cab
exclude.compression ?:\...\*.aes
exclude.compression ?:\...\*.rsa
```

The preferred setting is `compressalways yes`, and then use **`exclude.compression`** statements to omit files that cannot be compressed.

Tuning client-side data deduplication

The performance of client-side data deduplication can be affected by processor requirements and deduplication configuration.

About this task

Data deduplication is a method of reducing storage needs by eliminating redundant data. Client-side data deduplication is the process of removing the redundant data during a backup operation on the client system. Client-side data deduplication is especially effective when you want to conserve bandwidth between the Tivoli Storage Manager client and server.

Procedure

To help you enhance the performance of client-side data deduplication, take the following actions based on the task that you want to complete.

Table 19. Actions for tuning client-side data deduplication performance

Action	Explanation
Ensure that the client system meets the minimum hardware requirements for client-side data deduplication.	Before you decide to use client-side data deduplication, verify that the client system has adequate resources available during the backup window to run the deduplication processing. The preferred minimum processor requirement is the equivalent of one 2.2 GHz processor core per backup process with client-side data deduplication. For example, a system with a single-socket, quad-core, 2.2-GHz processor that is used 75% or less during the backup window is a good candidate for client-side data deduplication.
Use a combination of deduplication and compression to obtain significant data reduction.	When data is compressed after it is already deduplicated, it can give you more savings in data reduction as compared to running data deduplication alone. When data deduplication and compression are both enabled during a backup operation on the backup-archive client, the operations are sequenced in the preferred order (data deduplication followed by compression).
Avoid running client compression in combination with server-side data deduplication.	When you use client compression in combination with server-side data deduplication, it is typically slower and reduces data volume less than the preferred alternatives of server-side data deduplication alone, or the combination of client-side data deduplication and client-side compression.

Table 19. Actions for tuning client-side data deduplication performance (continued)

Action	Explanation
Increase the number of parallel sessions as an effective way to improve overall throughput when you are using client-side deduplication. This action applies to client systems that have sufficient processor resources, and when the client application is configured to perform parallel backups.	For example, when you use IBM Tivoli Storage Manager for Virtual Environments, it might be possible to use up to 30 parallel VMware backup sessions before a 1 Gb network becomes saturated. Rather than immediately configuring numerous parallel sessions to improve throughput, increment the number of sessions gradually, and stop when you no longer see improvements in throughput. For information about optimizing parallel backups, see "Optimizing parallel backups of virtual machines" on page 224.
Configure the client data deduplication cache with the **enablededupcache** option.	The client must query the server for each extent of data that is processed. You can reduce the processor usage that is associated with this query process by configuring the cache on the client. With the data deduplication cache, the client can identify previously discovered extents during a backup session without querying the Tivoli Storage Manager server. The following guidelines apply when you configure the client data deduplication cache: • For the backup-archive client, including VMware virtual machine backups, always configure the cache for client-side data deduplication. • For Tivoli Storage Manager for Virtual Environments operations, if you configure multiple client sessions to back up a vStorage backup server, you must configure a separate cache for each session. • For networks with low latency that process a large amount of deduplicated data daily, disable the client deduplication cache for faster performance. **Restriction:** • For applications that use the Tivoli Storage Manager API, do not use the client data deduplication cache because backup failures can occur if the cache becomes out of sync with the Tivoli Storage Manager server. This restriction applies to the Tivoli Storage Manager Data Protection applications. Do not configure the client data deduplication cache when you are using the data protection products. • If you use image backups, do not configure the client data deduplication cache.
Decide whether to use client-side data deduplication or server-side data deduplication.	Whether you choose to use client-side data deduplication depends on your system environment. In a network-constrained environment, you can run data deduplication on the client to improve the elapsed time for backup operations. If the environment is not network-constrained and you run data deduplication on the client, it can result in longer elapsed backup times. To evaluate whether to use client-side data or server-side data deduplication, see the information in Table 20 on page 202.

Use the following checklist to help you choose whether to implement client-side or server-side data deduplication.

Table 20. Checklist for choosing client-side versus server-side data deduplication

Question	Response
Does the speed of your backup network result in long backup times?	**Yes** Use client-side data deduplication to obtain both faster backups and increased storage savings on the Tivoli Storage Manager server. **No** Determine the importance of storage savings versus faster backup process.
What is more important to your business: The amount of storage savings that you achieve through data reduction technologies, or how quickly backups complete?	Consider the trade-offs between having the fastest elapsed backup times and gaining the maximum amount of storage pool savings: • For the fastest backups in an unconstrained network, choose server-side data deduplication. • For the largest storage savings, choose client-side data deduplication that is combined with compression.

What to do next

For more information about using Tivoli Storage Manager deduplication, see Effective Planning and Use of IBM Tivoli Storage Manager V6 and V7 Deduplication.

Related concepts:

"Checklist for data deduplication" on page 30

Related tasks:

"Evaluating data deduplication results" on page 67

"Tuning server-side data deduplication" on page 147

Reducing client data flow with include and exclude options

You can use client **include** and **exclude** options to reduce the amount of data that is backed up, which shortens the backup and restore windows.

Almost every file system collects files that are not critical to your applications and users. Examples of such files include operating system files or updates that can be easily downloaded and be reinstalled if you do not have a local copy, core files, log files, and cached data. Use the Tivoli Storage Manager client **include** and **exclude** options to ensure that the client operations are protecting only what is important.

Adjusting the I/O buffer size of the client

The **diskbuffsize** client option specifies the maximum disk input/output (I/O) buffer size (in KB) that the client uses when it is reading files.

Optimal client performance during backup, archive, or HSM migration processes might be achieved if the value for this option is equal to or smaller than the amount of file read ahead that is provided by the client file system. A larger buffer requires more memory and might not improve performance.

The default value is 32 KB for all clients except clients that run on AIX. For clients that run on AIX operating systems, the default value is 256 KB, except when `enablelanfree yes` is specified. When `enablelanfree yes` is specified on AIX, the default value is 32 KB. API client applications have a default value of 1023 KB, except for Windows API client applications, version 5.3.7 and later, which have a default value of 32 KB.

The default value is the preferred value for the client operating system.

If the performance of restore operations seems slow, consider resizing the **diskbuffsize** option:

1. Stop the current restore operation.
2. Set this option in the appropriate server stanza in the `dsm.sys` file: `diskbuffsize 32`.
3. Restart the restore operation.

Optimizing the transaction size

A transaction is a unit of work that is exchanged between a client and the server.

A client program can transfer more than one file or directory between the client and server before the client commits the data in the transaction to server storage. If a transaction contains more than one file or directory, it is called a transaction group.

Data in a transaction is sent from the client to the server during backup operations. Data in a transaction is sent from the server to the client during a restore operation.

You can control the amount of data that is sent between the client and server by setting the **txnbytelimit** client option. The server administrator can also limit the number of files or directories that are contained within a transaction group by setting the **TXNGROUPMAX** option.

These two options work together such that the size of any transaction is based on which of these parameter values is reached first. For example, if the **TXNGROUPMAX** option is set to `4096` and the **txnbytelimit** option is set to `25600` KB (25 MB), then up to 4096 small files can be included in a transaction if the sum of their sizes does not exceed 25600 KB. If objects are 25 MB in size, or larger, they are sent as one file in one transaction.

Changing how much data can be sent in a transaction affects the speed at which the client performs work. The default values are sufficient in most environments, except when the data is being written directly to a tape device. At the end of each transaction, the tape buffers must be written to physical media, which is a slow process; transferring more data with each transaction can improve performance when the server is writing data directly to tape.

Consider the following tips when you set a value for **txnbytelimit**:

- Increasing the amount of data per transaction increases the size of logs and log pools space on the server. Verify that you have enough free disk space to contain larger transaction logs and log pool space. Increasing log sizes can also cause longer server start times.
- Increasing the amount of data per transaction causes more data to be retransmitted if an error occurs. Resending data reduces performance, and resending larger transactions reduces performance even more.
- The benefits of changing the setting for the **txnbytelimit** option depend on the configuration and the type of workloads. In particular, increasing this value benefits tape storage pool backup more so than disk storage pool backup, especially if you are protecting many small files.

Consider setting a smaller **txnbytelimit** value if error conditions cause repeated retransmissions of transactions, when you specify static, shared static, or shared dynamic as the copy serialization attribute in the standard management class. A smaller **txnbytelimit** value applies to `static` and `shared` attributes. If a file changes during a backup operation and the client does not send it, the client still must resend the other files in that transaction.

To enhance performance, set the **txnbytelimit** option to 2 GB, and on the server, set the **TXNGROUPMAX** option to 256 KB. Additionally, for small file workloads, stage the backups to a disk storage pool and then migrate the files to tape.

For the **txnbytelimit** option, you can specify a value in the range 300 KB - 32 GB. The default value is 25600 KB.

Suggested settings for txnbytelimit

txnbytelimit setting when backing up objects to disk before migrating them to tape: `txnbytelimit 25600K`

txnbytelimit setting when backing up objects directly to tape:
`txnbytelimit 10G`

If you are using Tivoli Storage Manager with the IBM Content Manager application and you notice slow server data movement operations, see the support article at technote 1246443 for information about using the `CM VOL_AGGREGATESIZE` setting in the `CM RMVOLUMES` table to improve transaction performance.

Effects of management classes on transactions

Each copy of a file or directory that is backed up by Tivoli Storage Manager is bound to (associated with) a management class.

A management class includes a backup copy group. A backup copy group defines how Tivoli Storage Manager manages objects that were backed up. Management class attributes include such things as the storage pool where the object is stored, how many versions of each object are created, and how long the versions are retained.

During backup operations, Tivoli Storage Manager bundles file and directory backup copies into transactions. That is, the client opens a transaction with the server database, backs up one or more objects, and then closes the transaction. If the Tivoli Storage Manager server database successfully commits the transaction, the client repeats the process until all eligible objects are backed up.

Transaction-based processing provides for reliable backups, but each commit operation also increases processing time. In general, the best performance is achieved by grouping as many objects as possible into a single transaction.

The maximum size of a transaction is governed by the following two parameters:

TXNGROUPMAX
: This option is set on the server. It specifies the maximum number of objects that can comprise a transaction

txnbytelimit
: This option is set on each client. It specifies the maximum size of a transaction, in KB.

The size of any transaction is based on which of these parameter values is reached first. For example, if the **TXNGROUPMAX** option is set to `4096` and the **txnbytelimit** option is set to `25600` KB (25 MB), then up to 4096 small files can be included in a transaction if the sum of their sizes does not exceed 25600 KB. If objects are 25 MB in size, or larger, they are sent as one file in one transaction.

Another factor that can influence the transaction size is the destination storage pool for the backed up objects. The objects within a transaction must all be directed to the same storage pool. As the transaction is processed, if one of the objects is directed to a different storage pool, then the current transaction is committed, and a new transaction is opened for the object that is destined for another storage pool.

If the destination storage pool is frequently changed during a backup operation, performance is reduced because new transactions must be created. For example, assume that you have a directory structure that contains many media files of different media file formats, such as the following files:

```
/media/vid001.jpg
/media/vid001.wmv
/media/vid002.jpg
/media/vid002.wmv.
.
.
.
/media/vid9999.wmv
```

Assume also that you have **include** statements that bind these files types to different management classes, such as the following examples:

```
include /media/*.jpg diskclass
include /media/*.wmv tapeclass
```

The management classes named DISKCLASS and TAPECLASS each specify a different storage pool: one writes to disk and the other writes to tape. When the media files are backed up, the `/media/vid001.jpg` file is backed up in one transaction and is directed to the disk storage pool. The next object, `/media/vid001.wmv`, is backed up in another transaction that is directed to the tape storage pool. Then, the `/media/vid002.jpg` file is backed up in yet another new transaction and is directed to the disk storage pool. This behavior adversely affects the performance of backup operations. In addition to the additional processing time from the inefficient transactions, more delays can occur if you must wait for tapes to be mounted.

Consider revising your management classes, or management class bindings, to reduce or eliminate the number of different storage pools that the client uses for backed up objects.

A similar situation can occur when directory objects are backed up. By default, directory objects are bound to the management class with the longest **RETONLY** (retain only version) value. If multiple management classes in the active policy set have the same value for **RETONLY**, then the management class that is last, when sorted alphabetically, is used. For example, if the management classes named DISKCLASS and TAPECLASS both have the same **RETONLY** setting, and they are both in the active policy set, then the default management class for directory objects is TAPECLASS.

If file objects are directed to a disk storage pool and directory objects are directed to a different storage pool, such as tape, that also reduces transaction efficiency and slows performance. One way to avoid the inefficiencies of using a different management class for directory objects is to use the **dirmc** option and specify the same management class that you use to back up files. Using the example management classes named DISKCLASS and TAPECLASS, set `DIRMC DISKCLASS` to bind directory objects to the management class, and storage pool, that you use for file objects.

Setting options to minimize processor usage

You can set several client options to reduce the time that is required for client processing tasks and improve performance. The client options to consider are **quiet**, **virtualnodename**, **ifnewer**, **incrbydate**, and **tapeprompt**.

For Mac OS X file systems, limiting the length of extended attributes can help improve the client performance.

For clients on any operating system, turning off antivirus programs, or other programs that compete with the client for system resources, can also improve client performance.

quiet client option

Two client options determine whether messages are displayed during backup operations: **quiet** and **verbose**. The **verbose** client option is the default option; it causes messages to be displayed in the output during client operations. The **quiet** client option can be set to suppress messages from being displayed.

When you set the **quiet** option, messages and summary information are still written to the log files, but they are not displayed in GUI or command-line output. The **quiet** option provides two main benefits that can improve client performance:

- For tape backups, the first transaction group of data is always resent. To avoid the transaction from being resent, use the **quiet** option to reduce retransmissions at the client.
- If you are using the client scheduler to schedule backups, the **quiet** option reduces entries in the schedule log, which might improve client throughput.

Although the **quiet** option can offer some modest performance improvements, consider using the default (**verbose**) option. The benefits of having messages displayed and logged can outweigh the performance gains that the **quiet** option provides.

virtualnodename client option

When you restore, retrieve, or query objects that are owned by another node, consider using the client **virtualnodename** option instead of the **fromnode** option.

The **fromnode** option uses more system resources than the **virtualnodename** option. By using the **virtualnodename** option instead of the **fromnode** option, you might improve client performance.

ifnewer client option

The **ifnewer** client option is used only with restore commands. This option can reduce network traffic during restore operations. This option ensures that files are restored only if the date of the file that is stored on the server is newer than the date of the same file that is stored on the client node.

The **ifnewer** option can be set only on the command line, and, like all options that are specified on the command line, it must be preceded by a hyphen (-) character. For example:

```
dsmc restore "/home/grover/*" -sub=y -rep=y -ifnewer
```

incrbydate client option

The **incrbydate** client option can shorten backup windows because the option causes the client to back up objects only if they are new, or changed, since the last incremental backup was run. As its name implies, this option can be used only to perform incremental backups.

Incremental backups that use the **incrbydate** option, have limitations that regular incremental backups do not have. You must understand these limitations to correctly use this option. For incremental by date backups, consider the following limitations:

- Files that are created or modified after a directory was processed by the Tivoli Storage Manager client, but before the backup completes, are skipped at the next **incrbydate** backup.
- **incrbydate** backups do not cause expired files to be deleted from the server.
- If a management class changes for a file or a directory after an **incrbydate** is run, the stored objects are not rebound to the new management class.
- After an **incrbydate** backup runs, if only the attributes of an object are changed, the file is not included in the next **incrbydate** backup.

During an incremental backup operation, where the **incrbydate** option is not used, the server reads the attributes of all the client files that are in the server file system and passes this information to the client. The client then compares the server attribute list to all of the files in the client file system. This comparison can be time-consuming, especially on clients that have limited memory.

With an incremental-by-date backup, the server passes only the date of the last successful incremental backup operation to the client and the client backs up only files that are new or changed since the previous incremental backup. The time savings can be significant. However, regular, periodic incremental backups are still needed to back up files workstation files that are excluded from backups, by the limitations of incremental by date backups.

For example, if a new file in your file system has a creation date that is earlier than the last successful backup date, future incremental-by-date backups do not back up this file because the client assumes that the file has already been backed up. Also, files that were deleted are not detected by an incremental-by-date backup and these deleted files are restored if you perform a full system restore.

For a full list of restrictions about the **incrbydate** option, see incrbydate option.

The **incrbydate** option can be specified only on the command line with the **incremental** command. Like all options that are specified on the command line, it must be preceded by a hyphen (-) character. For example:

```
dsmc incremental -incrbydate
```

Consider journal-based backups as an alternative to incremental by date backups. Journal-based backups perform a traditional incremental backup of the file system when the first backup occurs. A journal file records which file system objects are changed after the initial backup, and the journal is used to determine which objects to include in subsequent backups. Journal-based backup is most appropriate for file systems that do not change many objects, often. For more information about journal-based backups, see "File backup techniques" on page 175.

tapeprompt client option

The **tapeprompt** client option specifies whether you want to be prompted to wait for a tape to be mounted, if a tape is required to back up or restore objects.

Backup and restore, and archive and retrieve operations can be processed with fewer interactive input delays if you specify `tapeprompt no`. With `tapeprompt no` specified, the client still waits for tapes to be mounted if they are required, but the prompts that ask whether you want to wait for a tape, or skip objects that require a tape, are suppressed.

Improving client performance by using multiple sessions

You can set up the client to use multiple backup sessions to improve client performance.

Running concurrent client sessions

Running two or more client program instances at the same time on the same system might provide better overall throughput than a single client instance, depending on the available resources.

You can schedule backups for multiple file systems concurrently on one Tivoli Storage Manager client system with any of the following methods:

- By using one node name, and running one client scheduler, and setting the **resourceutilization** client option to 5 or greater, and including multiple file systems in the schedule, or in the domain specification. This method is the simplest way to run concurrent sessions.
- By using one node name, running one client scheduler, and scheduling a command that runs a script on the client system, where the script includes multiple client (**dsmc**) commands.
- By using multiple node names and running one client scheduler for each node name, and where each scheduler uses its own client options file.

Multiple session backup and restore

A multiple session restore operation allows backup-archive clients to start multiple sessions that use no-query restore operations, thus increasing the speed of restore operations. A multiple session restore operation is similar to a multiple session backup operation.

Multiple session restores can be used under the following conditions:

- The data to be restored is stored on several tape volumes or file device class volumes.
- Sufficient mount points are available.
- The restore is done by using the no-query restore protocol.

When you request a backup or archive, the client can establish more than one session with the server. The default is to use two sessions: one to query the server and one to send file data.

Parallel (concurrent) backup and restore operations that work with sequential file or tape storage pools require multiple mount points. A mount point is a tape or a file device class volume. The **resourceutilization** client option governs the maximum number of concurrent backup or restore sessions that the client can use. The **MAXNUMMP** server parameter, on the **UPDATE NODE** or **REGISTER NODE** commands, and the **MOUNTLIMIT** setting in the **DEFINE DEVCLASS** and **UPDATE DEVCLASS** commands, determines how many mount points a client node can use, at one time.

Configure these settings according to your requirements and available hardware. Take into account the number of mount points that all nodes might need, at any one time. For example, if you have four client nodes and only eight tape drives, if you configure all four nodes with MAXNUMMP 8, one node can seize all of the tape drives, leaving no tape drives for other nodes to use.

If all the files are on random disk, only one session is used. There is no multiple session restore for a random-access disk-only storage pool restore. However, if you are restoring files and the files are on four sequential disk volumes (or on four tape volumes) and other files are on random access disk, you can use up to five sessions during the restore.

Server settings take precedence over client settings. If the client **resourceutilization** option value exceeds the value of the server **MAXNUMMP** setting for a node, you are limited to the number of sessions that are specified by the **MAXNUMMP** parameter.

Multiple restore sessions are allowed only for *no-query* restore operations. A no-query restore is started by using an unrestricted wildcard in the file specification on the **restore** command. The following is an example of a no-query restore.

```
dsmc restore /home/*
```

The wildcard character (*) is unrestricted because it does not filter on object names or extensions. For example, dsmc restore /home/????.* is unrestricted. No-query restores also cannot use any of the object-filtering options. Specifically, you cannot use the **inactive**, **latest**, **pick**, **fromdate**, or **todate** options. For details about running a no-query restore, see the Restore command.

The server sends the **MAXNUMMP** value to the client during sign-on. During a no-query restore operation, if the client receives a notification from the server that another volume that contains data to be restored was found, the client checks the **MAXNUMMP** value. If another session would exceed the **MAXNUMMP** value, the client does not start the session.

Backup considerations

Only one producer session per file system compares attributes for incremental backup. Incremental backup throughput does not improve for a single file system with a small amount of changed data.

Data transfer sessions do not have file system affinity; each consumer session could send files from multiple file systems, which helps balance the workload. Sending files from multiple file systems is not beneficial if you are backing up directly to a tape storage pool that is collocated by file space. Do not use multiple sessions to back up objects directly to a storage pool collocated by file space. Use multiple commands, one per file space.

The setting of the **resourceutilization** option and internal heuristics determine whether new consumer sessions are started.

When you backup objects directly to tape, you can prevent multiple sessions, so that data is not spread across multiple volumes, by setting the **resourceutilization** option to 2.

Restore considerations

Only one session is used when files are restored from random access disk storage pools.

Only one file system can be restored at a time with the command line, but multiple sessions can be used on a single file system.

Even small clients can realize improved throughput of restore operations if the data to be restored is on multiple tapes. One session can be restoring data while another might be waiting for tapes to be mounted, or be delayed while it is reading a tape, looking for the data to restore.

Tape cartridge contention might occur, especially when files are not restored from a collocated pool. Collocating files reduces the likelihood of tape cartridge contention.

Optimizing the number of multiple sessions to run

Tivoli Storage Manager clients can establish concurrent sessions to back up and restore data. The creation of concurrent sessions is controlled by an algorithm within the client software; you cannot directly control this algorithm. The default behavior is to use two sessions: one to query the server and one to send file data. You can set the **resourceutilization** option to cause the client to use additional concurrent sessions to query and send data.

Multiple sessions are used when you specify multiple file specifications on a backup command, **restore** command, **archive** command, or a **retrieve** command. For example, if you enter the following command and you specify `resourceutilization 5`, the client might start a second session to query the server for a list of files that were backed up on file space B:

```
inc /Volumes/filespaceA /Volumes/filespaceB
```

Whether the second session starts depends on how long it takes to query the server about files that are backed up on file space A. The client might also try to read data from the file system and send it to the server on multiple sessions.

The value that you specify for the **resourceutilization** option is an integer in the range 1 - 10. The value that you specify does not correlate directly to the number of sessions the client can create. For example, setting resourceutilization 5 does not mean that the client can have only five concurrent sessions running. What it does indicate is that this client might create more concurrent sessions than a client that has **resourceutilization** set to 1, but fewer concurrent sessions than a client that has **resourceutilization** set to 10. The **resourceutilization** option setting increases or decreases the ability of each client to create multiple sessions.

The following factors affect the performance of concurrent sessions:

Available server resources and processing capacity
The hardware that the Tivoli Storage Manager server runs on must have sufficient memory, storage, and processor capacity to efficiently support multiple sessions.

Available client resources and processing capacity
The hardware that the Tivoli Storage Manager client runs on must also have sufficient memory, storage, and processor capacity to efficiently support multiple sessions.

Configuration of the client storage subsystem
File systems that are spread across multiple disks, either by software striping, RAID-0, or RAID-5, can accommodate the increases in random read requests that concurrent sessions generate with better efficiency than a single-drive file system can. In fact, a single-drive file system might not show any performance improvement if you set the **resourceutilization** option.

For file systems spread across multiple physical disks, setting the **resourceutilization** to 5 or higher can produce optimal performance in configurations where the server has sufficient processing capacity and memory to handle the load.

Network bandwidth
Concurrent sessions increase the amount of data that flows over the network. In particular, LANs might be adversely affected by the increased data traffic.

If you set the **resourceutilization** option and you are backing up client files directly to a sequential device, update the server **MAXNUMMP** setting to accommodate the additional mount points that concurrent sessions might need.

Before you change any settings, consider the potential disadvantages of concurrent sessions:

- Concurrent sessions could produce multiple accounting reports.
- The server might not be configured to support all of the potential concurrent sessions. Review the server **MAXSESSIONS** setting and change it if the client-initiated sessions can exceed its current value.
- A **QUERY NODE** command might not accurately summarize client activity.

During restore operations, the default client behavior is to use a single session, unless the **resourceutilization** option was specified to a value greater than 2. When you are restoring files for a critical client system from tape, and the files are on many tape volumes, set the **RESOURCEULTIILZATION** value to 10. If four tape drives are available, and you want restore operations to use all four tape volumes concurrently, set the **MAXNUMMP** value for the node to 4. If all of the client files that are being restored are in random-access disk storage pools, only one restore session is used, regardless of the **resourceutilization** option value.

The default value for the **resourceutilization** option is 1, and the maximum value is 10.

For example, if the data to be restored is on five different tape volumes, and the maximum number of mount points for the node that is requesting the restore is 5, and **resourceutilization** option is set to 3, three sessions are used for the restore. If the **resourceutilization** setting is increased to 5, five sessions are used for the restore. There is a one-to-one relationship between the number of restore sessions that are allowed and the **resourceutilization** setting.

The following values are the preferred settings:

For workstations
```
resourceutilization 1
```

For a small server
```
resourceutilization 5
```

For a large server
```
resourceutilization 10
```

The following table shows the maximum number of concurrent sessions that are possible, for each of the values in the **resourceutilization** range. A producer thread is a session that scans the client system for eligible files. The remaining sessions are consumer threads and are used to transfer data. Subtract the producer sessions that are listed in the table from the maximum number of sessions to determine the number of consumer threads. In the table, the threshold column shows how soon a subsequent thread can be started after the previous thread is started, for each of the values that are specified for the **resourceutilization** option.

resourceutilization value	Maximum number of sessions	Unique number of producer sessions	Threshold (seconds)
1	1	0	45
2	2	1	45
3	3	1	45
4	3	1	30
5	4	2	30
6	4	2	20
7	5	2	20
8	6	2	20
9	7	3	20
10	8	4	10
0 (default)	2	1	30

Tuning journal-based backups

To help improve the performance of incremental backups, you can run journal-based backups.

About this task

Journal-based backups have the following advantages over standard incremental backups:

- Journal-based backups can complete faster than standard incremental backups because they do not compare file system object attributes with information that is stored on the server. Instead, on a file system that supports journaling, changes to a file system are recorded in a locally stored journal database. The locally stored journal database entries are used to determine which objects to include in backup operations.

  The benefits of using a journal-based backup do diminish if the file systems have many file changes. Journal-based backups perform best on large file systems in which many of the files do not often change.
- Journal-based backups require less memory and less client disk I/O than full incremental backups.

Procedure

Use the information in the following table to help you tune journal-based backups.

Action	Explanation
Ensure that there is enough disk space on the client system to contain the journal database.	The amount of disk space that is required for the journal database depends on the number of files and directories that change between each successive journal-based backup operation.
Use default settings.	The default settings for journal size, log names and locations, file system check intervals, and other journal settings work well in most environments.

Action	Explanation
Change the default settings. Edit the `tsmjbbd.ini.smp` file to include or exclude file systems to monitor for changes, set the size of the journal database, and specify notification options and other settings. Save the changes to a file named `tsmjbbd.ini` (without the `smp` extension).	If the default settings do not work well in your system environment, change them. For example, by excluding file systems, you can limit the amount of data to monitor for journal-based backups. This action can improve the performance of the backups. Configuration settings for the journal service (on Windows) or journal daemon (on Linux and AIX) are copied to the client disk when you install the backup-archive client. The default settings are in the `tsmjbbd.ini.smp` file. Comments in the `tsmjbbd.ini.smp` file provide the documentation for the journal settings. The journal service or journal daemon uses the `tsmjbbd.ini` file when the journal service is started. **Tips for the Windows client:** • You can use the configuration wizard to edit the default settings. • Changes that are made to the `tsmjbbd.ini` file are applied dynamically. When changes are made to the settings in the file, the journal service applies the changes automatically without needing to restart the service.

What to do next

For more information about when to use journal-based backups, see "Journal-based backup" on page 176.

Optimizing restore operations for clients

Standard Tivoli Storage Manager progressive incremental backup operations are optimized to restore individual files or small numbers of files.

Progressive incremental backup minimizes tape usage, reduces network traffic during backup operations, and eliminates the storage and tracking of multiple copies of the same data. Progressive incremental backup might reduce the impact to client applications during backup. For a balanced level of backup and restore performance, try running progressive incremental backup with collocation set on, in the storage pool.

If restore performance is more important than a balance between backup and restore operations, you can optimize based on your goals for restore performance. When you optimize for restore operations, there are often costs in tape usage and backup performance.

When you optimize restore operations, the performance depends on the type of media that you use. For more information about the media that you can use to restore data, see Table 21 on page 215.

Table 21. Advantages and disadvantages of the different device types for restore operations

Device type	Advantages	Disadvantages
Random access disk	• Quick access to files • No mount point needed	• No reclamation of unused space in aggregates • No deduplication of data
Sequential access disk (FILE)	• Reclamation of unused space in aggregates • Quick access to files (disk based) • Allows deduplication of data	Requires mount point but not as severe an impact as real tape
Virtual tape library	• Quick access to files because of disk-based media • Existing applications that were written for real tape do not have to be rewritten	• Requires mount point but not as severe an impact as real tape • No deduplication of data
Active data pools	• No sorting through inactive files to get to active data • Can be defined on any type of storage pool • Tapes can be taken offsite for disaster recovery	Cannot be used with random access disk pools
Tape	• A large amount of data can be stored on a tape • Tapes can be taken offsite for disaster recovery	• Requires mount point and physical tape mounting/dismounting • No deduplication of data • Slower access to files because of sequential access of tapes

The following tasks can help you balance the costs against the need for optimized restore operations:

- Identify systems that are most critical to your business. Consider where your most important data is, what is most critical to restore, and what needs the fastest restore. Identify which systems and applications you want to focus on, optimizing for restore.
- Identify your goals and order the goals by priority. The following list has some goals to consider:
  - Disaster recovery or recovery from hardware crashes, requiring file system restores
  - Recovery from loss or deletion of individual files or groups of files
  - Recovery for database applications (specific to the API)
  - Point-in-time recovery of groups of files

The importance of each goal can vary for the different client systems that you identified as being most critical.

For more information about restore operations for clients, see "Concepts for client restore operations" on page 218.

Environment considerations

Tivoli Storage Manager performance depends upon the environment.

The environment includes network characteristics, storage hardware, and time constraints for backup and restore operations. Consider the following items when deciding on the storage hardware:
- Types of tape drives used
- The availability of snapshot functions
- The availability of disk drives
- The availability of fiber-channel adapters

Consider sequential-access disk (FILE) to store data that requires quick restoration. For data that is less critical, store the data to random access disk, then allow or force the data to migrate to tape.

You can also use active-data pools to store active versions of client backup data. Archive and space-managed data is not allowed in active-data pools. Inactive files are removed from the active-data pool during expiration processing. Active-data pools that are associated with a FILE device class do not require tape mounts, and the server does not have to position past inactive files. In addition, FILE volumes can be accessed concurrently by multiple client sessions or server processes. You can also create active-data pools that use tape media, which can be moved off-site, but which require tape mounts.

If you do not use FILE or active-data pools, consider how restore performance is affected by the layout of data across single or multiple tape volumes. You can have multiple simultaneous sessions when you use FILE to restore, and mount overhead is skipped with FILE volumes. Major causes of performance problems are excessive tape mounts and needing to skip over expired or inactive data on a tape. After a long series of incremental backups, perhaps over years, the active data for a single file space can be spread across many tape volumes. A single tape volume can have active data that is mixed with inactive and expired data.

Restoring entire file systems

Using a file system image backup optimizes restore operations when an entire file system must be restored. For example, in disaster recovery or recovery from a hardware failure.

Restoring from an image backup minimizes concurrent mounts of tapes and positioning within a tape during the restore operation.

Consider the following information when you run file system restore operations:
- Combine image backups with progressive incremental backups for the file system to allow for full restore to an arbitrary point-in-time.
- To minimize disruption to the client during backup, use either hardware-based or software-based snapshot techniques for the file system.
- Perform image backups infrequently. More frequent image backups give better point-in-time granularity, but there is a cost. The frequent backups affect the tape usage, there is an interruption of the client system during backup, and there is greater network bandwidth needed.

As a guideline you can run an image backup after a percentage of data is changed in the file system, since the last image backup.

Image backup is not available for all clients. If image backup is not available for your client, use file-level restore as an alternative.

Restoring parts of file systems

Progressive incremental backups optimize restore operations for small numbers of files or groups of files. These backups also make optimal use of network bandwidth for backup operations, and can minimize elapsed backup time and tape usage.

To optimize for restoring a file or a group of files, or for a system on which an image backup cannot be made, consider the following methods:

- Use collocation by group, by a single client node, or by client file space for primary sequential pools that clients back up to. For large file spaces for which restore performance is critical, consider creating mount points on the client system. The mount points would allow collocation of data under the file space level.
- Specify the client option `COLLOCATEBYFILESPEC`. This option helps limit the number of tapes that are written to by objects from one file specification. For more information about this option, see Collocatebyfilespec.
- Create backup sets that can be taken to the client system and can be used to restore from, directly. This method is effective if there is sufficient lead time before the restore, and can save network bandwidth.

  Backup set creation can also be done periodically when resources are available, for example, on weekends.
- Use progressive incremental backups, but periodically force a backup of all files.

  Some users reported it effective to define multiple Tivoli Storage Manager client nodes on a system. One client node runs the incremental backups and uses policies which retain multiple versions. Another client node runs either full backups or incremental backups with collocation, but uses policies that retain a single version. One node can be used for restoring older versions of individual files. You can use the other client node for restoring a complete file system or directory tree to the latest version.

  Another effective way to optimize restore operations is to occasionally create a backup image.
- Create multiple storage pool hierarchies for clients with different priorities. For the most critical data, the best choice might be to use only disk storage. You can use different storage hierarchies to set collocation differently in the hierarchies.
- Although it might affect server performance, issue the **MOVE NODEDATA** command to consolidate critical data in tape storage pools. You can even issue the command in storage pools that have collocation that is turned on. It might be important to consolidate data for certain nodes, file spaces, and data types more often than for others. If you do not use collocation or are limited by tape quantity, you can consolidate data more often. You can also consider the rate of data turnover.

For more information about collocation, see Keeping client files together using collocation (V7.1.1).

Restoring databases for applications

Doing more frequent full backups leads to faster restores for databases. For some database products, you can use multiple sessions to restore, you can restore just the database, or restore just the database log files.

For information about data protection for databases, see IBM Tivoli Storage Manager for Databases

Restoring files to a point-in-time

Keeping many versions is not essential for restoring to a point-in-time. But by increasing the number of versions that you keep, you might restore from an earlier point-in-time and still find the versions corresponding to that time.

If you also schedule incremental backups regularly, you might have greater granularity in restoring to a discrete point-in-time. However, keeping many versions can degrade restore operation performance. Setting policy to keep many versions also has costs, in terms of database space and storage pool space. Your policies might have overall performance implications.

If you cannot afford the resource costs of keeping the large numbers of file versions and must restore to a point-in-time, consider the following options:

- Use backup sets
- Export the client data
- Use an archive
- Take a volume image, including virtual machine backups

You can restore to the point-in-time when the backup set was generated, the export was run, or the archive was created. Remember, when you restore the data, your selection is limited to the time at which you created the backup set, export, or archive.

Tip: If you use the archive function, create a monthly or yearly archive. Do not use archive as a primary backup method because frequent archives with large amounts of data can affect server and client performance.

See “Restoring parts of file systems” on page 217.

Concepts for client restore operations

The client restore includes the following operations:

“No-query restore operations” on page 219

“Running multiple commands with backup and restore” on page 219

“Running multiple sessions on clients for a restore” on page 220

“Controlling resource utilization by a client” on page 220

No-query restore operations

The client uses two different methods for restore operations: Standard restore (also called classic restore), and no-query restore.

The no-query restore requires less interaction between the client and the server, and the client can use multiple sessions for the restore operation. The no-query restore operation is useful when you restore large file systems on a client with limited memory. The advantage is that no-query restore avoids some processing that can affect the performance of other client applications. In addition, it can achieve a high degree of parallelism by restoring with multiple sessions from the server and storage agent simultaneously.

With no-query restore operations, the client sends a single restore request to the server instead of querying the server for each object to be restored. The server returns the files and directories to the client without further action by the client. The client accepts the data that comes from the server and restores it to the destination named on the restore command.

The no-query restore operation is used by the client only when the restore request meets both of the following criteria:

- You enter the restore command with a source file specification that has an unrestricted wildcard.

  An example of a source file specification with an unrestricted wildcard is:

  ```
  /home/mydocs/2002/*
  ```

  An example of a source file specification with a restricted wildcard is:

  ```
  /home/mydocs/2002/sales.*
  ```

- You do not specify any of the following client options:

  inactive
  latest
  pick
  fromdate
  todate

To force classic restore operations, use ?* in the source file specification rather than *. For example:

```
/home/mydocs/2002/?*
```

For more information about restore processes, see the Restore command.

Running multiple commands with backup and restore

You can run multiple commands instead of multiple sessions to speed up the backup and restore of client nodes with critical data.

When you use multi-sessions to back up data, the sessions might be contending for the same underlying hard disk. The contention for resources can cause delays in processing.

An alternative method is to manage backups by starting multiple client commands, where each command backs up a predetermined number of file systems. Using this method, with collocation at the file space level, can improve backup throughput and allow for parallel restore processes across the same hard disk drives.

You must issue multiple commands when you are restoring more than one file space. For example, when you are restoring both a `C` drive and a `D` drive on a Windows system you must issue multiple commands.

You can issue the commands one after another in a single session or window, or issue them at the same time from different command windows.

When you enter multiple commands to restore files from a single file space, specify a unique part of the file space in each restore command. Be sure that you do not use any overlapping file specifications in the commands. To display a list of the directories in a file space, issue the **QUERY BACKUP** command on the client. For example:

```
dsmc query backup -dirsonly -subdir=no /usr/
```

Running multiple sessions on clients for a restore

To use multiple sessions, data for the client must be on multiple sequential access volumes in a file or tape storage pool. Or the data can be contained in a random disk storage pool (with a device class with a device type of DISK). The data for a client usually becomes spread out over some number of volumes over time.

To potentially benefit from multi-session restores, consider collocating client data by group. Collocation by group can cause data for a node to be distributed on more than one volume. The distribution happens while the group's total data is kept on as few volumes as possible.

Restore operations can be restricted on mount points. The **MAXNUMMP** parameter in the **REGISTER NODE** or **UPDATE NODE** command applies to restore operations. The client can restrict the number of sessions, which are based on the combination of the **MAXNUMMP** value and the client **RESOURCEUTILIZATION** value. Unlike tape, you can mount FILE volumes in multiple sessions simultaneously, for restore or retrieve operations.

Set the client option for resource utilization to one greater than the number of sessions that you want. Use the number of drives that you want that single client to use. The client option can be included in a client option set.

Issue the restore command so that it results in a no-query restore process.

Controlling resource utilization by a client

You can control the number of mount points (equivalent to drives) allowed to a client by setting the **MAXNUMMP** parameter on either the **UPDATE NODE** or **REGISTER NODE** command.

At the client, the option for resource utilization also affects how many drives (sessions) the client can use. The client option, resource utilization, can be included in a client option set. If the number specified in the **MAXNUMMP** parameter is too low and there are not enough mount points for each of the sessions, it might not be possible to achieve the benefit of the multiple sessions that are specified in the resource utilization client option.

- For backup operations, prevent multiple sessions if the client is backing up directly to tape so that data is not spread among multiple volumes. Multiple sessions can be prevented at the client by using a value of 2 for the resource utilization option on the client.

- For restore operations, set the resource utilization option to one greater than the number of sessions that you want. Use the number of drives that you want that single client to use.
- With file spaces, a session is limited to processing a single file space. You cannot have multiple backup or restore sessions processing a single file space. However, if you have multiple file spaces on a client, you can have multiple sessions processing those file spaces.

File space tuning

Using Tivoli Storage Manager virtual mount points can enhance the performance of backup and restore operations on file systems that contain millions of files.

On many operating systems that Tivoli Storage Manager supports, you might be able to use file system or operating system tools to divide the file systems into manageable units, such that each file system can be protected in an acceptable backup or restore window.

On AIX, HP-UX, Linux, and Solaris operating systems, the Tivoli Storage Manager **virtualmountpoint** option can be used to logically divide a large file system into smaller increments. *Virtual mount points* are a Tivoli Storage Manager construct. Virtual mount points are not recognized by the operating system as mount points; they are only recognized and used by Tivoli Storage Manager.

When it protects objects that are contained under a virtual mount point, Tivoli Storage Manager treats each virtual mount point as a separate file space. Creating virtual mount points can enhance performance in the following ways:

- Less memory is needed for client operations because the virtual mount points divide a large file system into smaller increments, and processing fewer objects uses less memory.
- Tivoli Storage Manager can do more work in parallel, by running concurrent backup or restore operations on objects that are under two or more of the virtual mount points.

Using virtual mount points to improve performance works best if each of the virtual mount points has approximately the same number of files. If you cannot divide your file system in this way, virtual mount points might not be a suitable means to improve performance.

To illustrate how to use virtual mount points, assume that a client has a large file system called `/data`. Also, assume that the `/data` file system has multiple subdirectories that you want to frequently protect.

You can use the **virtualmountpoint** option to create virtual mount points that divide the `/data` file system into manageable logical units, such as shown in the following example:

```
virtualmountpoint /data/dir1
virtualmountpoint /data/dir2
.
.
virtualmountpoint /data/dir19
virtualmountpoint /data/dir20
```

These example **virtualmountpoint** options create 20 virtual mount points for the `/data` file system. The objects (`dir1`, `dir2`, and so on) that are shown on the example **virtualmountpoint** statements are directory objects on the file system.

When objects in these directories are stored on the server, they are stored in a file space that matches name of the objects that are included on each **virtualmountpoint** statement. That is, objects in dir1 are stored in file space named dir1, and so on.

You can back up and restore objects in each virtual mount point, independently of the others, and independently of other objects that are not in a virtual mount point. Any objects that get added to the /data file system, but that are not under a virtual mount point, are protected when you back up the objects in the /data file system. Objects that are under a virtual mount point are protected when you back up the virtual mount point.

If you use **virtualmountpoint** options, monitor the growth of the physical file system. If many new objects are added to the physical file system in locations that are not defined as logical mount points, it might eventually be easier to abandon the use of the virtual mount points and just backup the entire file system.

If you intend to use virtual mount points to divide the contents of a large file system, be aware that adding virtual mounts after a file system has been backed up can change the command syntax that is needed to restore objects.

For example, assume that you back up the /data/dir1/file1 object before you create any virtual mount points. The /data/dir1/file1 object is stored on the server in the /data file space. Assume that you later create a virtual file space by setting virtualmountpoint /data/dir1 and that you create and back up a file1 object in it. This new file1 object is stored on the server in the /dir1 file space (the file space matches the virtual mount point name).

Running dsmc restore /data/dir1/file1 restores the file1 object from the copy that is stored on the server in the virtual mount point (dir1) file space.

To restore the file1 object that was saved in the /data file space, you must use the following syntax:

```
dsmc restore {/data}/dir1/file1
```

The brace ({ and }) characters force the server to search the /data file space for the file1 object.

Consider the following items if you use virtual mount points to create addition file spaces on the server:

- For applications that use the Tivoli Storage Manager API, limit the number of file spaces to no more than 100 per client. Examples of programs that use the API are IBM Tivoli Storage Manager for Virtual Environments, IBM Tivoli Storage Manager for Mail,IBM Tivoli Storage Manager for Enterprise Resource Planning, and IBM Tivoli Storage Manager for Databases.
- For sequential-accessed storage pool volumes, collocate files by node or group, rather than by file space. For example, 100 small file systems require 100 volumes if they are collocated by file space, but fewer volumes are needed if the files are collocated by node or group.

Windows system state backups

The most recently released versions of the Tivoli Storage Manager backup-archive client and Tivoli Storage Manager server software include updates that improve the performance of Windows system state backup and restore operations.

There are no user-configurable options or settings that can be adjusted to improve the efficiency of Windows system state protection. Backing up or restoring the Windows system state is a resource-intensive and time consuming operation. If you decide that you must back up the Windows system state, consider whether you can define server policy settings that retain fewer versions of system state backups. For example, your organization might require you to retain data files for 60 days, but require only 10 days of retention for system state information. The client **include.systemstate** option can be used to specify a different management class to be used for system state backups.

The decision to explicitly back up the Windows system state depends on how you plan to restore a node, after a node fails. The following factors can influence your decision to back up the Windows system state data:

- If you plan to restore a node by reinstalling the operating system from the Windows installation media, or from a repair disk and image backup, you do not need to back up the Windows system state data.
- If you plan to restore a physical machine from an image backup or from a snapshot image backup, back up all volumes because system state data might exist on other disks, and not just on the C drive.
- If you plan to restore a Windows virtual machine, system state objects are backed up when you perform a full backup of the virtual machine. A separate backup of the system state data is not required to restore a Windows virtual machine from a full virtual machine backup.
- If you plan to perform a bare metal restore of a client node, you must explicitly back up the system state files so they are available to restore the system state objects to the same, or a different, system. To reduce storage requirements, associate system state backups with policies that limit the number of backup copies that are retained on the server or in storage pools.

  Restriction: Bare metal restore of Microsoft Windows servers and workstations that adhere to the Unified Extensible Firmware Interface (UEFI) specification is possible only from Tivoli Storage Manager backup-archive clients that are V7.1, or later.
- Performing a system state backup of a Windows system by using a V6.2 or later Tivoli Storage Manager client can cause issues on a V5.5 Tivoli Storage Manager server. For more information about this configuration, see technote 1470662.

Tuning virtual machine backup operations

You can improve the performance of backup operations for virtual machines by adjusting client options.

About this task

To help improve the performance of backup operations for virtual machines with the backup-archive client or Tivoli Storage Manager for Virtual Environments, adjust the settings for the following options:

- Options for optimizing parallel backups of virtual machines

- Options for the transport mode for VMware backups
- Options for adjusting the scalability of virtual machine backup operations (applicable only to Tivoli Storage Manager for Virtual Environments Version 6.4 or later)

Optimizing parallel backups of virtual machines

The Tivoli Storage Manager V6.4 and later backup-archive client provides parallel backup processing for backing up multiple virtual machines at the same time with one Tivoli Storage Manager data mover node.

About this task

The data mover node is the node that represents a specific Tivoli Storage Manager backup-archive client that moves data from one system to another.

If you have a license to use Tivoli Storage Manager for Virtual Environments, you can run parallel backup processing of full, incremental, incremental-forever full, and incremental-forever incremental virtual machine backups. If you use the standard backup-archive client, you can run only full virtual machine backups in parallel.

To help optimize parallel virtual machine backups for Tivoli Storage Manager for Virtual Environments: Data Protection for VMware, adjust the settings for the **vmmaxparallel**, **vmlimitperhost**, and **vmlimitperdatastore** options. These options can also help reduce the processor load that parallel backups can create on a host in the vSphere infrastructure.

To optimize parallel virtual machine backups for Tivoli Storage Manager for Virtual Environments: Data Protection for Hyper-V, adjust the settings for the **vmmaxparallel** option.

For more information option, see Parallel backups of virtual machines.

vmmaxparallel client option

The **vmmaxparallel** option specifies the maximum number of virtual machines that can be backed up to a Tivoli Storage Manager server at one time per client process.

Before you set a value for the **vmmaxparallel** option, review the following information:

Preferred setting

The preferred value for the **vmmaxparallel** option depends on the following factors:

- Resource availability on the Tivoli Storage Manager server and client
- Network bandwidth between the server and the client
- The load tolerance on the participating VMware infrastructure

To determine the preferred setting for this option, experiment with parallel backups of the virtual machines. In that way, you can select a setting that is appropriate for the backup window and the hardware and system configuration in the environment.

The default value is 1, which might be too restrictive. The maximum value is 50, which might be ineffective.

In general, set this option to the highest value that still maintains a tolerable processor load on the vSphere hosts and a tolerable I/O load on the data stores for a virtual machine backup workload.

Effects on performance

Increasing this value can result in more parallelism in terms of backup processing and can improve aggregate throughput to the Tivoli Storage Manager server from a backup-archive client.

Potential trade-offs for this setting

Setting a value too low might limit the potential of a backup environment by restricting the aggregate throughput to the Tivoli Storage Manager server. However, a low value might be necessary to throttle the amount of data that is sent from the client to the Tivoli Storage Manager server, or to reduce the processor or I/O load on the vSphere hosts and data stores.

Setting a value too high might result in oversaturating the link between the client and the Tivoli Storage Manager server, or elevating the processor load on certain vSphere hosts. Beyond a certain value, you might not experience improvements in aggregate throughput performance, depending on the network bandwidth and proxy or vSphere host processor resources that are available.

`vmlimitperhost` client option

The **vmlimitperhost** option specifies the maximum number of virtual machines on an ESX server that can be included in a parallel backup operation.

Before you set a value for the **vmlimitperhost** option, review the following information:

Preferred setting

The preferred value is the default value of 0. By using the default value, you set no limits on the maximum number of virtual machines on an ESX server that can be included in a parallel backup operation. Verify that the value for the **vmlimitperhost** option is compatible with the value for the **vmmaxparallel** option.

Set the **vmlimitperhost** option to the highest value that can still maintain a tolerable processor load on any single vSphere host for a virtual backup workload. Ensure that backup sessions are distributed equally across the affected vSphere hosts.

When you are specifying the setting, consider the set of virtual machines that are being backed up.

For example, if a set of 10 virtual machine guests is hosted on 5 vSphere hosts, and **vmmaxparallel** is set to 10, set the **vmlimitperhost** option to 2. In this way, you can distribute the parallel backup sessions across the hosts during a 10-guest parallel backup operation.

Effects on performance

The **vmlimitperhost**, **vmmaxparallel**, and **vmlimitperdatastore** options limit the number of parallel backup operations that occur overall and for any single vSphere host. You can use these options to reduce the processor load that parallel backups can create on a vSphere host.

For a set of virtual machine guests that you back up, the order that Tivoli Storage Manager uses for creating backup sessions is random. Depending on the setting for the **vmmaxparallel** option, it might be possible that too many backup sessions would involve too few vSphere hosts at any one time during a backup operation.

The **vmlimitperhost** option can be used to ensure that no more than the number of backup sessions that are indicated by the **vmlimitperhost** option value affect any one host.

Potential trade-offs for this setting

Setting a value too low might artificially limit the maximum number of concurrent virtual machine backups in an environment to less than what is feasible. However, a low value might be necessary to throttle the amount of data that is sent to the Tivoli Storage Manager server, or to reduce the processor load on the involved vSphere hosts.

Setting a value too high might result in elevated processor loads on certain vSphere hosts.

vmlimitperdatastore client option

The **vmlimitperdatastore** option specifies the maximum number of virtual machines in a data store that can be included in a parallel backup operation.

Before you set a value for the **vmlimitperdatastore** option, review the following information:

Preferred setting

The preferred value is the default value of 0. By using this value, you set no limit on the maximum number of virtual machines in a data store that can be included in a parallel backup operation. However, ensure that the selected value is compatible with the value that is being used for the **vmmaxparallel** option.

Set the **vmlimitperdatastore** option to the highest value that can still maintain a tolerable processor load on any single vSphere host for a virtual machine backup workload. In addition, adjust this value so that the backup workload is spread across as many vSphere data stores as possible.

When you are specifying the setting, consider the set of virtual machines that are being backed up.

For example, if a set of 10 virtual machine guests is hosted on 5 vSphere data stores, and **vmmaxparallel** is set to 10, set the **vmlimitperdatastore** option to 2. In this way, you can distribute the parallel backup sessions across the data stores during a 10-guest parallel backup operation.

Effects on performance

The **vmlimitperdatastore**, **vmmaxparallel**, and **vmlimitperhost** options limit the number of parallel backups that occur overall and for any single vSphere datastore. You can set these options to reduce the processor load that parallel backups can create on a vSphere host or hot spots on vSphere datastore LUNs.

For a set of virtual machine guests that you back up, the order that Tivoli Storage Manager uses to create backup sessions is random. Depending on the setting for the **vmmaxparallel** option, it might be possible that too many backup sessions would involve too few vSphere datastores.

The **vmlimitperdatastore** option can be used to ensure that no more than the number of backup sessions that are indicated by the **vmlimitperdatastore** option affect any one datastore.

Potential trade-offs for this setting

Setting a value too low might artificially limit the maximum number of concurrent virtual machine backups in an environment to less than what is feasible. However, a low value might be necessary to throttle the amount of data that is sent to the Tivoli Storage Manager server, or to reduce the processor load on the vSphere hosts or the I/O load on vSphere data stores.

Setting a value too high might result in elevated processor loads on certain vSphere hosts, depending on the mapping of VMware data stores to hosts. Setting too high a value might also result in elevated loads on certain vSphere data stores. This outcome might lead to inefficiencies because the underlying LUNs of those data stores must handle excessive I/O processes compared to others.

Selecting a transport mode for VMware backups

To set the preferred transport order or hierarchy for backup or restore operations on VMware virtual machines, specify the **vmvstortransport** option.

Procedure

Before you set the **vmvstortransport** option, review the following information:

- In most cases, set the **vmvstortransport** option to `default` (`san:hotadd:nbdssl:nbd`). If you do not specify a value for the **vmvstortransport** option, the default value is used.

  The optimal transport mode depends on the composition of the backup environment. Table 22 specifies the transport mode to use for specific backup environments.

Table 22. Preferred transport modes for specific backup environments.

Backup environment	Value for the vmvstortransport option
You want to offload backup traffic from the LAN, and instead, move it over a SAN.	*san*
You are using a data mover node that is installed on a virtual machine to back up other virtual machines. The backups can be moved over a SAN or a LAN.	*hotadd*
You are backing up virtual machines over an Ethernet LAN; you either do not have, or do not want to use, a SAN to offload backup traffic from the LAN.	*nbd*
You are backing up virtual machines over an Ethernet LAN and you want to use SSL to encrypt the data. Note that encrypting data can degrade backup performance.	*nbdssl*

- The values that are listed in Table 22 are the preferred transport methods, but you might not want to specify a single transport method as the value. You might specify multiple transport methods so that you can fail over to another transport method when the first one fails; otherwise, the operation might fail. However, you might want to restrict this option so that only a certain set of transport methods is used. If you leave an entry out of the colon-separated value list, that entry is no longer available and is skipped.
- Consider the effects of the setting on performance.

  It is typically preferable to use the fastest available transport mode. However, in some environments, it might be necessary to avoid certain transport modes or emphasize others for resource management purposes.
- Consider the potential trade-offs for this setting.

Specifying a slower transport mode might reduce the aggregate throughput of the backup environment.

What to do next

For more information about the **vmvstortransport** option, see the Vmvstortransport client option.

Adjusting the scalability of virtual machine backup operations

You can improve performance by adjusting the scalability within Tivoli Storage Manager for incremental-forever backups of virtual machines.

Before you begin

You must have a license to use Tivoli Storage Manager for Virtual Environments V6.4 or later.

About this task

Virtual machine disk files are stored on the Tivoli Storage Manager server as data blocks called megablocks. Each of these megablocks contains 128 MB of data. When a change occurs on a disk in an area that is represented by a megablock, a Tivoli Storage Manager object is created. For every subsequent incremental backup, if a change is detected, an extra Tivoli Storage Manager object is created on the server. When a large amount of objects exists for the same virtual machine data, excessive demands are placed on the Tivoli Storage Manager server database.

Procedure

To fine-tune these Tivoli Storage Manager server scalability conditions, use either the **mbobjrefreshthresh** or **mbpctrefreshthresh** option, but not both.

mbobjrefreshthresh client option

Use this option when you estimate the Tivoli Storage Manager objects that represent production data for each virtual machine backup.

For example, when the number of Tivoli Storage Manager objects exceeds this value, the megablock is refreshed. This action means that the entire 128 MB block is backed up to the Tivoli Storage Manager server and is represented as a single Tivoli Storage Manager object.

Before you set a value for the **mbobjrefreshthresh** option, review the following information:

Preferred setting

The preferred value is the default, 50. When the number of Tivoli Storage Manager objects that are needed to describe a 128 MB megablock for a virtual machine guest disk exceeds this value, the entire megablock is refreshed.

If you are backing up to a tape storage pool on the server, you might decrease this value so that megablock refreshes occur more often. This way, the data that was backed up for each virtual machine guest disk is more likely to be collocated on tape volumes. This setting might improve restore performance in such cases.

Effects on performance

When a megablock is refreshed, the objects that were used to represent the megablock area in previous backups are expired. This option can affect the quantity of data that is copied to the Tivoli Storage Manager server and the server database-related processor utilization during incremental forever backups.

Potential trade-offs for this setting

Setting this option close to its maximum value of 8192 can result in sending less data to the Tivoli Storage Manager server during an average incremental forever backup operation. However, the number of database entities that the Tivoli Storage Manager server must track increases. This result can increase server processor utilization during incremental forever backups by a small degree.

Setting this option close to its minimum value of 2 can result in marginal database processing savings during incremental forever backups. However, the quantity of data that is copied to the Tivoli Storage Manager server might be higher and might approach the size of a full backup.

`mbpctrefreshthresh` client option

The **`mbpctrefreshthresh`** option defines a threshold for the percentage of a megablock that can change before a full refresh is initiated. Use this option when you estimate the amount of extra data that is backed up for each virtual machine.

For example, when a 128 MB block of a production disk changes more than the percentage specified by the **`mbpctrefreshthresh`** option, the entire 128 MB block is copied to the Tivoli Storage Manager server. The block is represented as a single Tivoli Storage Manager object.

Before you set a value for the **`mbpctrefreshthresh`** option, review the following information:

Preferred setting

The preferred value is the default, 50. When a 128 MB megablock changes by a percentage that exceeds this value since its last refresh (a full copy to the Tivoli Storage Manager server), the entire megablock is refreshed.

If you are backing up to a tape storage pool on the server, you might decrease this value so that megablock refreshes occur more often. This way, the data that was backed up for each virtual machine guest disk is more likely to be collocated on tape volumes. This setting might improve restore performance in such cases.

Effects on performance

When a megablock is refreshed, the objects that were used to represent the megablock area in previous backups are expired. This option can affect the quantity of data that is copied to the Tivoli Storage Manager server and the server database-related processor utilization during incremental forever backups.

Potential trade-offs for this setting

Setting this option close to its maximum value of 100 can result in sending less data to the Tivoli Storage Manager server during an average incremental forever backup operation. However, the number of database entities that the Tivoli Storage Manager server must track increases. This result can increase server processor utilization during incremental forever backups by a small degree.

Setting this option close to its minimum value of 1 can result in marginal database processing savings during incremental forever backups. However, the quantity of data that is copied to the Tivoli Storage Manager server might be higher and might approach the size of a full backup.

Performance tuning for LAN-free environments

LAN-free backup can improve performance because the backup traffic can be routed over the SAN instead of the LAN. LAN-free data movement can make LAN bandwidth available for other uses and decrease the load on the Tivoli Storage Manager server, allowing it to support a greater number of concurrent client connections.

Backing up data to tape or disk over the SAN, or restoring data from tape or disk over the SAN, has the following advantages over equivalent operations that are performed only over the LAN:

- Metadata is sent to the server over the LAN; sending metadata over the LAN has negligible impact on LAN performance. Client data bypasses the potentially busy and slower LAN and is sent over the faster SAN. Backing up or restoring data over a SAN is generally faster than the same operation over a LAN.
- Sending client data over the SAN frees the Tivoli Storage Manager server from the task of handling data, which leads to more efficient use of server resources because the data goes directly to storage.
- Using a SAN is more efficient than a LAN when you are protecting large files, or databases; Tivoli Data Protection products generally benefit from SAN efficiencies.

When you configure Tivoli Storage Manager in a SAN environment, consider the following points:

- Ensure that you provide a sufficient number of data paths to tape drives.
- Backing up many small files directly to a real tape device can be inefficient. For file systems that have many small files, consider sending the files over the LAN to a disk storage pool, and migrate the files to tape later.
- Optimize the transaction size for writing files to tape or disk; for information, see "Optimizing the transaction size" on page 203.
- To improve backup and restore performance, include `lanfreecommmethod shardemem` in the client options file if the storage agent and client are on the same system. Setting this option allows the Tivoli Storage Manager client and Tivoli Storage Manager storage agent to communicate by using RAM, instead of using TCP/IP.
- Set the **`tcpnodelay`** option to `YES`, in the server or client options. This setting allows packets that are smaller than the maximum transmission unit (MTU) to be sent immediately.

Do not use LAN-free backup and restore if you are using Tivoli Storage Manager server-side data deduplication. You cannot bypass the server if server-side data deduplication is used to reduce redundant objects processing.

Chapter 14. Tuning network performance

If possible, use a dedicated local area network (LAN) or a storage area network (SAN) for backup operations. Keep device drivers for network components updated to take advantage of the latest fixes and improvements. Consider the Tivoli Storage Manager options that can help you tune how the clients and the server use the network. Ensure that you understand how TCP/IP works and review the information about TCP flow control and the sliding window.

Tuning TCP/IP settings for clients and servers

Typically, the default values for the client and server options for TCP/IP work well.

Before you begin

Review the information in "TCP flow control and the sliding window" on page 233. Ensure that you observe system performance before and after changes.

Procedure

- If you set the **TCPWINDOWSIZE** option to a value that is greater than 63 on either the Tivoli Storage Manager server or client, you must enable TCP window scaling (as defined by RFC 1323) in the TCP/IP settings. See the documentation for your operating system for how to set the TCP receive window size.
- Typically the default values for the **TCPWINDOWSIZE** options on the client and server are preferred. A larger window might improve communication performance, especially on fast networks with high latency, such as a long-distance wide area network (WAN) connection.

  If you decide to tune TCP/IP window sizes, review the following guidance.
  - If you increase the size of the TCP/IP window, do so in increments. For example, try doubling the value for the **TCPWINDOWSIZE** option and observing the results before you increase the value again. A larger value for the **TCPWINDOWSIZE** option does not always give better performance.

    Tip: Do not set the sliding window size to be larger than the buffer space on the network adapter. The window acts as a buffer on the network. A window size that is larger than the buffer space on the network adapter might cause packets to be lost on the network adapter. Because packets must be sent again when they are lost, throughput might degrade.
  - If the operating system automatically tunes the TCP receive window size, consider setting the Tivoli Storage Manager server **TCPWINDOWSIZE** option to `0`. Setting the option to `0` means that server sessions use the receive window for the operating system.

    If the operating system cannot automatically tune TCP window size, do not set **TCPWINDOWSIZE** option to `0`.
  - The window size that you set with the **tcpwindowsize** client option might have to be a compromise for different operations on the system. For example, you might have to use a value that is a compromise between what works best for backup-archive client operations and what works best for Tivoli Storage Manager for Virtual Environments operations.

- For the **TCPNODELAY** option, use the default value of YES. This setting disables the Nagle algorithm and allows packets that are smaller than the MTU size to be immediately sent.
- Windows systems have a TCP auto-tuning feature that monitors session transfer statistics and then adjusts the receive window as needed for optimal performance. For Tivoli Storage Manager servers and clients that run on these systems, you might consider setting the Tivoli Storage Manager **TCPWINDOWSIZE** option to 0 to use auto-tuning.

  Tip: The TCP auto-tuning feature is enabled by default on some versions of Windows, and disabled by default on others. If you intend to use auto-tuning, ensure that it is enabled for the Windows system.

  Always monitor operations after such changes to verify that the performance is not degraded.

Controlling network traffic from client schedules

You can control the network traffic from scheduled client operations by using certain Tivoli Storage Manager server **SET** commands. The commands control how frequently clients contact the server, and how the sessions are spread across the schedule window.

Procedure

- Use the **SET RANDOMIZE** command to randomize start times within the startup window of each client schedule. Communication errors can happen when many clients contacts the server simultaneously. If concurrent scheduled operations for clients have communication errors, you can increase the randomization percentage so that client contact is spread out. Increased randomization decreases the chance for communication overload and failure. The randomized start times apply only to clients that use the client-polling scheduling mode.
- Set how frequently a client can contact the server to obtain scheduled work by using the **SET QUERYSCHEDPERIOD** command. This command overrides the client setting, and applies when the client-polling mode is used for schedules. A shorter time period means more network traffic because of client polling. Use longer settings (6 - 12 hours) to reduce network traffic. Alternately, use the server-prompted scheduling mode to eliminate network traffic because of client polling.
- Set a global limit on the number of times that a scheduled command is tried again on a client by using the **SET MAXCMDRETRIES** command. This command overrides the client setting. A smaller number reduces network traffic that is caused by scheduled commands that are tried again.

  If you use the **SET MAXCMDRETRIES** command, be sure to consider schedule startup windows. If a retry is attempted outside of the schedule startup window, it fails.
- Set the number of minutes between retries of a scheduled command after a failed attempt to contact the server. Use the **SET RETRYPERIOD** command. This command overrides the client setting. A larger value reduces the network traffic that is caused by retries and increases the chance of a successful retry.

  If you use the **SET RETRYPERIOD** command, be sure to consider schedule startup windows. If a retry is attempted outside of the schedule startup window, it fails.

Setting network options for Tivoli Storage Manager on AIX systems

The default values for network options on the AIX operating system can be used for most Tivoli Storage Manager configurations. Review the preferred network-related settings.

Procedure

- If you use 10 Gbit Ethernet adapters, enable flow control on the switch or router port that the AIX system is connected to. See your network administrator, or see the manuals for your router or switch to determine how to enable flow control.
- If you set the **TCPWINDOWSIZE** option to greater than 63, set the `rfc1323` network option to 1. To ensure that the new setting is used whenever the system is restarted, use the **-p** option on the command. For example, issue the following command:

  ```
  no -o rfc1323=1 -p
  ```
- If you see nonzero values for the `no mbuf errors` field in the output from the **entstat**, **fddistat**, or **atmstat** commands, increase the value for the `thewall` option. Set the `thewall` option to at least 131072 and the `sb_max` option to at least 1310720. Newer versions of the AIX operating system have larger defaults for these options, and modifying them might not be necessary.

TCP/IP and network concepts for advanced tuning

If you plan to tune TCP/IP settings for the Tivoli Storage Manager client or server, first ensure that you understand key concepts.

TCP/IP sends and receives data for applications on a system. TCP/IP is composed of two protocols: Transmission Control Protocol (TCP) and Internet Protocol (IP).

Applications such as the Tivoli Storage Manager client and server interact with TCP. By changing the **TCPWINDOWSIZE** client and server options, you affect the flow control function in TCP.

Applications do not interact with IP or lower-level protocols that control how one system communicates its receive window size to another, retransmission of lost data, or receipt of data from a sending system.

The following factors can affect network operations:

- System resources, such as memory and processors.
- Communications adapters. Link utilizations and the limitations of various communication layer implementations affect the use of resources.
- Data sizes and load on the network.

TCP flow control and the sliding window

Transmission Control Protocol (TCP) uses a *sliding window* for flow control. Before you tune any TCP/IP settings, first understand how the TCP sliding window works.

The TCP sliding window determines the number of unacknowledged bytes, *x*, that one system can send to another. Two factors determine the value of *x*:

- The size of the send buffer on the sending system
- The size and available space in the receive buffer on the receiving system

The sending system cannot send more bytes than space that is available in the receive buffer on the receiving system. TCP on the sending system must wait to send more data until all bytes in the current send buffer are acknowledged by TCP on the receiving system.

On the receiving system, TCP stores received data in a receive buffer. TCP acknowledges receipt of the data, and *advertises* (communicates) a new *receive window* to the sending system. The receive window represents the number of bytes that are available in the receive buffer. If the receive buffer is full, the receiving system advertises a receive window size of zero, and the sending system must wait to send more data. After the receiving application retrieves data from the receive buffer, the receiving system can then advertise a receive window size that is equal to the amount of data that was read. Then, TCP on the sending system can resume sending data.

The available space in the receive buffer depends on how quickly data is read from the buffer by the receiving application. TCP keeps the data in its receive buffer until the receiving application reads it from that buffer. After the receiving application reads the data, that space in the buffer is available for new data. The amount of free space in the buffer is advertised to the sending system, as described in the previous paragraph.

Ensure that you understand the TCP window size when you use sliding window for flow control. The window size is the amount of data that can be managed. You might need to adjust the window size if the receive buffer receives more data than it can communicate. For more information about optimizing the TCP window size, see "Optimization of window size for different operations on the same system" on page 236.

How the send and receive buffers interact has the following consequences:

- The maximum number of unacknowledged bytes that a system can send is the smaller of two numbers:
  - The send buffer size on the sending system
  - The receive window size that the receiving system advertises to the sending system
- When the receiving application reads data as fast as the sending system can send it, the receive window stays at or near the size of the receive buffer. The result is that data flows smoothly across the network. If the receiving application can read the data fast enough, a larger receive window can improve performance.
- When the receive buffer is full, the receiving system advertises a receive window size of zero. The sending system must pause and temporarily cannot send any more data.
- In general, more frequent occurrences of zero size for the receive window results in overall slower data transmission across the network. Every time the receive window is zero, the sending system must wait before sending more data.

Typically, you set the send window and the receive window sizes separately for an operating system. In AIX, for example, the `tcp_sendspace` and `tcp_recvspace` parameters of the **no** command can be used to set the send and receive window sizes.

The sliding window that is used by Tivoli Storage Manager operations is controlled with the **TCPWINDOWSIZE** option.

Related concepts:

"TCP window controls in Tivoli Storage Manager"

TCP window controls in Tivoli Storage Manager

The **TCPWINDOWSIZE** options for Tivoli Storage Manager server and clients override the operating system settings for the size of send and receive windows for TCP/IP sessions. The **TCPWINDOWSIZE** option is available as a server option and a client option. With each option, you specify one value, which is used as the size for both the send and receive windows.

During incremental backup operations for files, both client and server act as receivers of data:

- The server sends metadata about the inventory of active backup versions to the client. The metadata consists of file names and attributes. For file systems that contain millions of files, this data can be a substantial amount, as much as hundreds of megabytes or even gigabytes.
- The client sends backup copies of new and changed files to the server.

Typically the default value for the **TCPWINDOWSIZE** option works well. A larger window might improve communication performance, especially on fast networks with high latency, such as a long-distance wide area network (WAN) connection.

Specifying the **TCPWINDOWSIZE** option with a value of 0 causes Tivoli Storage Manager to use the operating system default for the TCP window size. If the operating system does not automatically tune the TCP window size, avoid using the operating system default. The operating system default might be optimized for other applications, which might not be the optimal setting for Tivoli Storage Manager.

If Tivoli Storage Manager clients and server are on the same subnet, a larger TCP window size is unlikely to improve throughput. Also, you might need more kernel memory if you set a large TCP receive window size. The risk of the increased memory requirements might be greater than the benefit of a larger TCP window size.

Modern operating systems provide TCP/IP stacks that commit the requested memory as it is needed. Therefore, these systems have less risk of increased kernel memory for the send and receive buffers. These operating systems also automatically tune the receive buffer size by observing the session transfer statistics, and either increasing or decreasing the receive window, as appropriate. For these operating systems only, you might set the Tivoli Storage Manager server **TCPWINDOWSIZE** option to 0 and use the automatic tuning feature. These settings are especially useful when clients that connect to the server are remote.

The **TCPWINDOWSIZE** option is not related to the **TCPBUFSIZE** server option or the **tcpbuffsize** client option. The **TCPWINDOWSIZE** option is also not related to the send and receive buffers that are allocated in client or server memory.

Related concepts:

"TCP flow control and the sliding window" on page 233

Optimization of window size for different operations on the same system

The sizes of TCP send and receive windows that work well for one application might not work well for another application, even for another Tivoli Storage Manager application.

Finding the correct balance for the window sizes between the server and the client is also important. For example, if you reduce the **TCPWINDOWSIZE** option on the client from 2000 to 63 and set the option on the server to 1024, slower backup performance is likely the result for the following reasons:

- Tivoli Storage Manager uses the **TCPWINDOWSIZE** to configure both the send buffer size and the receive buffer size. Thus on the client, the send buffer size is 63 KB.
- The maximum number of unacknowledged bytes that can be sent is limited by the smaller of the send buffer size and the receive window size that is communicated by the receiving system. Therefore, although the receiver (Tivoli Storage Manager server) has a window size of up to 1024 KB, the effective window size is 63 KB.

For backup operations to the Tivoli Storage Manager server, typically the server takes in data fast enough so that larger TCP receive windows do not cause the window size to become zero. If the server takes in data fast enough, larger window sizes can improve performance, and smaller window sizes can decrease performance.

Restore operations by a backup-archive client have different characteristics than restore operations by a Tivoli Storage Manager for Virtual Environments client. The backup-archive client performs the following actions for a restore operation:

1. The backup-archive client reads data that is sent by the Tivoli Storage Manager server from the TCP receive buffer.
2. The backup-archive client writes the data directly to files on disk.

   If this write operation is slow and the Tivoli Storage Manager server sends data faster than the client can write it, the TCP receive buffer becomes full. When the TCP receive buffer becomes full, the receiver advertises a zero window size more frequently and the operation slows.
3. The client repeats steps 1 and 2 until all data is restored.

Typically, the restore operation by a Tivoli Storage Manager for Virtual Environments client can be slower than a restore operation by a backup-archive client because of the operations that occur for writing the data. The Tivoli Storage Manager for Virtual Environments client performs the following actions for a restore operation:

1. The Tivoli Storage Manager for Virtual Environments client reads data that is sent by the Tivoli Storage Manager server from the TCP receive buffer.
2. The Tivoli Storage Manager for Virtual Environments client writes the data to the vStorage API. Then, more operations and resources might be required, including communications with VMware, VMware processing of the data, and allocation of new blocks as the virtual machine disk (VMDK) grows.

   If this write operation is slow and the Tivoli Storage Manager server sends data faster than the client can write it, the TCP receive buffer becomes full. When the TCP receive buffer becomes full, the receiver advertises a zero window size more frequently and the operation slows.
3. The Tivoli Storage Manager for Virtual Environments client repeats steps 1 and 2 until all data is restored.

In step 2 on page 236, the Tivoli Storage Manager for Virtual Environments restore operation has more possible operations and might need more resources than a restore operation by a backup-archive client. Therefore, a larger receive window size is more likely to cause the TCP receive window size to become zero for the Tivoli Storage Manager for Virtual Environments restore operation. When both Tivoli Storage Manager for Virtual Environments and the backup-archive client are in use on a system, you must find a window size that balances the needs of the two types of operations. For example, in one case a window size of 1008 gave better overall throughput for such a system.

Chapter 15. Performance tuning for products that are used with Tivoli Storage Manager

Performance tuning information is available for products that are used with Tivoli Storage Manager products and components.

IBM Tivoli Storage FlashCopy Manager
: Look for the most recent performance tuning information in the wiki at http://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Tivoli Storage FlashCopy Manager.

IBM Tivoli Storage Manager for Space Management
: See "Tuning for Tivoli Storage Manager for Space Management."

IBM Tivoli Storage Manager for Virtual Environments
: Look for the most recent performance tuning information in the wiki at http://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Tivoli Storage Manager/page/Tivoli Storage Manager for Virtual Environments.

  See also "Tuning virtual machine backup operations" on page 223.

IBM Tivoli Storage Manager for z/OS® Media
: See "Configuration tips for z/OS media server performance" on page 240.

Content Management
: See "Content Management environments" on page 242.

Tuning for Tivoli Storage Manager for Space Management

Tivoli Storage Manager for Space Management provides techniques that you can use to optimize object migration to, and recall from, tape storage.

Optimizing migration to tape

If you must migrate many small files to the server, performance is better if the data moves to disk rather than to tape. After the files are migrated to disk, you can use storage pool migration to move the files to tape.

If you attempt to migrate many files directly to tape, performance of the migration operation might be unsatisfactory, particularly if each file is relatively small (<100 MB). By default, Tivoli Storage Manager for Space Management operates on one file at a time, so there is one transaction for each file that is migrated (or recalled). Setting the **hsmgroupedmigrate** option to YES can improve performance because it causes the selected files to be grouped into a single transaction. This option can be especially helpful, if you use a wildcard pattern on the file specification parameter on the **dsmmigrate** command. You might not know, in advance, how many files might match the wildcard pattern.

The number of objects that are grouped in a single transaction is limited by the Tivoli Storage Manager server option named **txngroupmax**. The number of bytes that are sent in a single transaction is limited by the Tivoli Storage Manager client option named **txnbytelimit**.

Optimizing recall from tape

Tape processing is automatically optimized when you use the **dsmrecall** command and include both the **-filelist** option, which specifies the file that contains the list of files to recall, and the destination file system. If you do not specify a file system, the recall process does not optimize tape processing.

For example, the following command optimizes tape recall because it specifies both the **-filelist** option and a file system:

```
dsmrecall -filelist=myFileList myFileSystem
```

The **-preview** option on the **dsmrecall** command produces file lists that list the files that are in the file that is specified by the **-filelist** option and orders the file lists by the tapes that contain the files. You can specify each of these cartridge-centric file lists, on a separate **dsmrecall** command, to recall the files from tapes in a particular order.

If the list file that is specified on the **filelist** option is a collection file that was created by using the **dsmrecall** command with the `-preview` option, the recall begins immediately. The files are recalled in the order that they are specified in the collection file. To change the order in which files are recalled, you can edit the entries in the collection file.

If the list file is not a collection file, and the list file is correctly formatted, the file entries are sorted for optimized tape processing and they are then recalled.

In the following example, the recall begins immediately because the file that is identified by the **-filelist** parameter is a collection file:

```
dsmrecall -filelist=/HsmManagedFS/.SpaceMan/tapeOptimizedRecall/node_ID/PID/
FileList.ordered.collection myFileSystem
```

For more information about optimizing tape recall processing, see Optimized tape recall processing.

Configuration tips for z/OS media server performance

When you migrate a Tivoli Storage Manager V5 server on a z/OS system to a V6.3 server or later on an AIX or Linux on System z system, network bandwidth requirements can differ. After the migration, you might see an increase in the amount of data that is transferred over networks that are connected to the Tivoli Storage Manager server.

Before you set up a network connection between the Tivoli Storage Manager server and the z/OS media server, follow these suggestions:

- For client connections to back up, archive, restore, and retrieve data, the V6.3 server or later requires the same network bandwidth as the V5 server that was running on a z/OS system. This information is based on the assumption that the migrated V6.3 server or later protects the same clients as the V5 server.
- More network bandwidth is required for operations that store or retrieve data by using the z/OS media server than for operations that use a local disk or tape.
- For best results, use dedicated networks for connections between the V6.3 server or later and the z/OS media server. Use technologies that optimize network performance and efficiency, such as jumbo frames and HiperSockets™. You can use HiperSockets for connections between Linux and z/OS logical partitions that are located on the same System z central processor complex.

- To increase the network bandwidth between the V6.3 server or later and the z/OS media server, set up multiple interfaces that you specify on a single media server definition. You can use a comma-delimited expression for the TCP/IP address, as in this example:

  ```
  define server ... hladdress=127.0.0.1,192.0.2.0 lladdress=1555 ...
  ```

  If you use this method, data can be stored or retrieved by using any available network interface. You are not required to distribute storage requirements over several device classes. This method increases the available network bandwidth and supports network connection failover and load balancing.
- To reduce network bandwidth requirements, store backup and archive data to a local disk pool for the V6.3 server or later. Then, use storage pool backup and storage pool migration to copy and move the data to z/OS media server tape storage. More network bandwidth is required if you back up or archive data directly to the z/OS media server FILE device-class storage, and then move the data to z/OS tape storage.
- To optimize network performance for the z/OS media server, ensure that both the z/OS system and the Tivoli Storage Manager server systems can use a large TCP/IP window size. On z/OS, this means that the `TCPIP.PROFILE TCPCONFIG` statement includes the **`TCPMAXRCVBUFRSIZE`** parameter, which is set to the default value of 256 K or greater. On AIX, this means that the network parameter **`rfc1323`** is set to 1, which is not the default value. On Linux, this means that the kernel parameter **`net.ipv4.tcp_window_scaling`** is set to the default value of 1.
- If the z/OS media server stores the data, the single-session backup throughput might be 80% of the throughput that is expected when local storage devices are used. For example, assume the throughput for backing up a large amount of data is 30 MB per second for a V5 server on z/OS. After migration to the V6.3 server or later on AIX or Linux on System z and with a z/OS media server, the throughput might be reduced to 24 MB per second.
- Install and configure a Tivoli Storage Manager storage agent on the client system to back up or archive data to the z/OS media server directly. This method might reduce the network bandwidth requirements on the V6.3 server or later and increase backup throughput.
- For best performance, set the Maximum Transmission Unit (MTU) to 64 K. If the z/OS media server provides access to many volumes simultaneously, there might be socket errors because of failed MTU allocation requests by the TCP transport provider. The socket errors can cause a virtual IO error on the Tivoli Storage Manager server that accesses the z/OS media server. The Tivoli Storage Manager server marks affected volumes as "read only", due to virtual IO errors. If failed MTU allocation requests create socket errors, you can reduce the MTU size from 64 K to 32 K. You can reduce or eliminate socket errors and increase successful TCP socket activity by reducing the MTU size. However, a low MTU size might or might not improve performance. You can also reduce the mount limit in the device class to control the number of concurrently accessed volumes.

Content Management environments

Server performance can be affected by Content Management applications that interface with Tivoli Storage Manager through the Tivoli Storage Manager client application programming interface (API).

Minimizing time for each transaction

While most Tivoli Storage Manager operations process many files per transaction, Content Management applications tend to process few, or just one, file per transaction. With as few as one file per transaction, the time for each such transaction becomes critical. Key to Content Management application performance is the time that it takes to write to the storage pool and the active log.

- To minimize the time for writing to the storage pool and the active log, use disk systems that use write cache, which hides the latency of writing to physical disk.
- Consider avoiding the use of Tivoli Storage Manager features such as simultaneous write or active log mirroring. When you use these features, the server must perform more write operations at the end of each transaction. The additional write operations might cause slow performance in Content Management environments.
- Be careful when you are mirroring to storage over long distances. The time that is involved in the I/O process grows with the distance.

Reducing wasted space in `FILE` storage pools

If the average file that Content Management sends to Tivoli Storage Manager for backup is smaller than 256 KB, and you use `FILE` device classes for the storage pools, a substantial amount of space might be wasted in the storage pools.

Transactions that are 256 KB or less can waste space because the server writes a minimum of one block, or 256 KB, to a volume in a storage pool that uses a `FILE` device class. For example, if a transaction is only 64 KB, the space that is used on disk for the transaction is still 256 KB.

You can consider using the `NONBLOCK` data format for `FILE` storage pools that are used for Content Management data. Using the `NONBLOCK` data format instead of the `NATIVE` data format might save space under these conditions.

The data format for an existing storage pool cannot be changed. If your storage pools use the `NATIVE` data format and you want to try the `NONBLOCK` data format, you must define new storage pools.

Part 5. Appendixes

Appendix A. Server instrumentation reference

You can use server instrumentation to track operations, such as backup and restore, and to help identify where performance problems originate.

The server monitoring script, `servermonV6.pl`, runs the server instrumentation commands to collect data. Typically, you can use the script instead of the server instrumentation commands by themselves. You can download the server monitoring script from technote 1432937.

Related tasks:

"Server monitoring script" on page 80

Selecting a server instrumentation strategy

Follow the usage strategies to get the best results when you use server instrumentation.

Procedure

You can select any of the following server instrumentation strategies:

- Start and stop server instrumentation around the operation. An operation can be any procedure that affects performance, such as backup or restore operations.
  1. Start server instrumentation, and start the operation that you want to monitor.
  2. End server instrumentation immediately after the operation completes. If a thread is started when instrumentation is active, session and process statistics are included in the output. A thread is a sequence of actions that are managed by an operating system scheduler. A process might require more than one thread. For example, a backup operation uses at least two threads. You can issue a Tivoli Storage Manager administrative client macro command to start server instrumentation before you start the operation.
- Set a time limit when you run server instrumentation.
  - The optimum length of time to run server instrumentation for most cases is 5 - 15 minutes. You can run server instrumentation for up to 30 minutes.
  - If server instrumentation is active for 30 minutes, hundreds of threads are traced and the output can be overwhelming. Reports with that many threads can make it more difficult to diagnose a problem.
  - Do not run server instrumentation on a busy server for the maximum amount of time. When possible, limit instrumentation on the server. If the system workload is the problem, instrumentation results might not help you resolve the source of system performance problems.
- Find a match for the multiple threads of a particular session or process. Look for the parent and child relationships between the threads. In the instrumentation output per thread, use the thread ID and the parent thread ID to find the other threads that are associated with the operation.
  - Find the thread in the instrumentation data. For example, look in the Tivoli Storage Manager activity log file for a session ID that matches a particular client session in the instrumentation data.

– During the operation, take the output from the **SHOW THREADS** command to see the session or process ID that a particular thread is working on. Use the thread ID from the output to find that same thread ID in the instrumentation.
– Look for related threads, which are based on the amount of data that is moved.

Related reference:

"INSTRUMENTATION BEGIN"

"INSTRUMENTATION END" on page 247

Starting and stopping server instrumentation

You can start server instrumentation from an administrative command line or from an administrative client. After you stop server instrumentation, you can use the results to determine where performance problems are occurring.

About this task

You must have system privilege to start or stop server instrumentation.

Procedure

Complete the following steps to use server instrumentation:

1. Issue the **INSTRUMENTATION BEGIN** command to start instrumentation from the administrative client:

   ```
   dsmadmc -id=id -password=password instrumentation begin
   ```

   The *id* must be a Tivoli Storage Manager administrator ID that has system privilege.
2. Start an operation for which you want to analyze performance.
3. Issue the **INSTRUMENTATION END** command to stop server instrumentation. Specify an output file for the data. If you do not specify an output file, data is written only to the screen. If you issue any remote commands from an administrative client and redirect the output to a file, that file is saved to the administrative client. The following command can be issued from the administrative client:

   ```
   dsmadmc -id=id -password=password instrumentation end > filename
   ```

Related concepts:

"Server instrumentation categories" on page 81

Related tasks:

"Server monitoring script" on page 80

INSTRUMENTATION BEGIN

Use this command to start server instrumentation.

Privilege class

You must have system privilege to start server instrumentation.

Syntax

```
                                                    .-COLLECTStmt--=--False---------.
>>--INSTrumentation Begin--+--------------------+--+------------------------------+--------><
                           '-MAXThread=--nnnnn--'  '-COLLECTStmt--=--+-False-+----'
                                                                     '-True--'
```

Parameters

MAXThread
Specifies the maximum number of threads that you want to trace. The default is 4096. If more than 4096 threads might be running during the instrumentation interval, raise this value. The maximum number of threads that you can run is 100,000. The minimum number of threads that you can run is 512.

COLLECTStmt
Collects detailed information on SQL commands, especially SQL commands that take a long time to run. The default value is **FALSE**. If you suspect that the DB2 program that is the database manager for the server is causing the problem, change this value to **TRUE**.

Example: Start server instrumentation at the administrative command-line client

Start an administrative client session in command-line mode, and begin collecting data. Use the administrator ID, ralph_willson.

```
dsmadmc -id=ralph_willson -password=Ka1amaz0p instrumentation begin
```

Example: Use command redirection to start server instrumentation for a storage agent

Start server instrumentation on storage agent StgAgnt_375.

```
dsmadmc -id=ralph_willson -password=Ka1amaz0p
StgAgnt_375:instrumentation begin
```

INSTRUMENTATION END

Use this command to stop server instrumentation and save the output.

Privilege class

You must have system privilege to stop server instrumentation.

Syntax

```
>>--INSTrumentation End--+------------+------------------------------><
                         '-filename---'
```

Parameters

filename
Specifies the file name for the output. Instrumentation output is generated when instrumentation is stopped. The file that you specify for instrumentation output is saved to the server directory. If you are running the administrative client from another system to issue this command, the output is not stored on

the local system, but on the server system. You can specify a path to a different location if you cannot write to the server directory.

The preferred approach is to instead redirect the output to a file. See the examples that follow.

If you do not either specify a file name or redirect the output to a file, the output is only displayed on the screen and is not saved.

Example: Stop server instrumentation and redirect the output to a file

Stop server instrumentation and send the output to the `instr_041413.ods` file.

```
dsmadmc -id=ralph_willson -password=Ka1amaz00pa$$w0rd
instrumentation end > instr_041413.ods
```

Example: Use command redirection to stop server instrumentation for a storage agent

Stop server instrumentation on storage agent `StgAgnt_375` and send the output to the `instr_041413.ods` file.

```
dsmadmc -id=ralph_willson -password=Ka1amaz2p StgAgnt_375:instrumentation
end > instr_041413.ods
```

Related concepts:

"Server instrumentation categories" on page 81

Server instrumentation for different operating platforms

Server instrumentation differs on the various Tivoli Storage Manager server operating systems.

The operating systems differ for server instrumentation in these respects:

- On operating systems such as AIX, HP-UX, Linux, and Solaris, only one thread does I/O to any disk storage-pool volume (called **DiskServerThread**). This thread provides a disk volume-centric view and can be difficult to get complete operation disk statistics.
- On Windows servers, the following processes occur:
  - Any thread can do I/O on a disk storage-pool volume (called **SsAuxThread** for backup)
    - These threads provide a process or session-oriented view
    - It can be more difficult to identify disk contention issues
  - Windows timing statistics have only about a 15-millisecond granularity

Appendix B. Accessibility features for the Tivoli Storage Manager product family

Accessibility features help users who have a disability, such as restricted mobility or limited vision to use information technology products successfully.

Accessibility features

The IBM Tivoli Storage Manager family of products includes the following accessibility features:

- Keyboard-only operation using standard operating-system conventions
- Interfaces that support assistive technology such as screen readers

The command-line interfaces of all products in the product family are accessible.

Tivoli Storage Manager Operations Center provides the following additional accessibility features when you use it with a Mozilla Firefox browser on a Microsoft Windows system:

- Screen magnifiers and content zooming
- High contrast mode

The Operations Center and the Tivoli Storage Manager server can be installed in console mode, which is accessible.

The Operations Center help system is enabled for accessibility. For more information, click the question mark icon on the help system menu bar.

Vendor software

The Tivoli Storage Manager product family includes certain vendor software that is not covered under the IBM license agreement. IBM makes no representation about the accessibility features of these products. Contact the vendor for the accessibility information about its products.

IBM and accessibility

See the IBM Human Ability and Accessibility Center (http://www.ibm.com/able) for information about the commitment that IBM has to accessibility.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Linear Tape-Open, LTO, and Ultrium are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability
These terms and conditions are in addition to any terms of use for the IBM website.

Personal use
You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use
You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies,” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Glossary

A glossary is available with terms and definitions for the IBM Tivoli Storage Manager family of products.

See Tivoli Storage Manager glossary.

To view glossaries for other IBM products, see IBM Terminology.

Index

D

E

F

H

I

S

T

U

V

W

Z

Product Number: 5608-E01
5608-E02
5608-E03

Printed in USA