Improve Oracle Database performance with IBM Systems server-side caching

A technical report

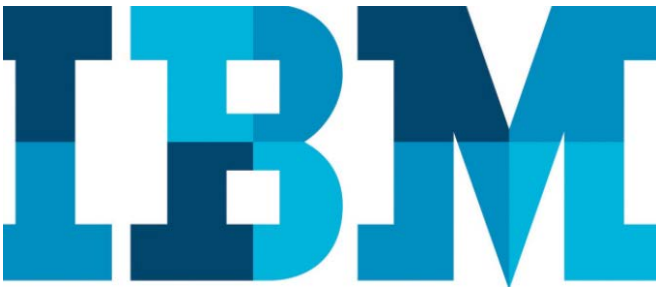
Overview

Challenge

You have a well-balanced storage system and fully tuned Oracle Database instance. How do you still improve the database throughput beyond its current limits?

Solution

Starting with AIX Version 7.1 with Technology Level 4 Service Pack 2 (7100-04-02), IBM Storage Systems support a new feature called *server-side caching* that uses SAN attached flash storage to improve read performance.



This white paper shows the benefits of using *server-side caching*, a feature supported by IBM® System Storage® to improve the performance of an Oracle Database hosted on a spinning disk storage system with little or no disruption to the existing infrastructure. The paper also discusses the configuration of server-side cache with an Oracle Database and its benefits for existing workloads.

The use of server-side caching at the operating system level is not only storage agnostic but also eliminates the need for data migration to a newer storage system to achieve better performance reason.

The target audience for this paper is technical leaders, system database and storage administrators planning to implement server-side caching feature with an Oracle Database.

This paper does not discuss the configuration aspects of the existing storage system. It also does not replace any official documentation provided by Oracle for deploying databases.

Note that the server-side caching feature is only recommended for use with a single instance Oracle Database and is not meant for use in an Oracle Real Application Clusters (RAC) environment.

The screen captures and graphs listed in this paper show improvements to the transactions per second (TPS) from the performance data captured in the lab setup. Results might differ in other environments depending on data, workload type, and resource availability.

Assumptions and prerequisites

This paper assumes that an Oracle Database host is running at least IBM AIX® version 7100-04-02 and later. The paper also assumes that a midrange external storage system, such as IBM Storwize® V7000, is used for storing the database and is configured for optimal database performance. Many customers already have storage systems like this configured with spinning disk and would like to gain the advantage of solid state or flash storage devices, but cannot upgrade the storage system without disrupting existing systems.

In addition to the adequate memory to run the existing database workload, the database host must have additional free memory for the caching software to manage metadata on each read block. A minimum of 4 GB memory is required for any IBM POWER® processor-based server logical partition (LPAR) that has caching enabled.

Several perquisites must be met at the operating system level before the installation of Oracle Database software. Refer to the database installation guide from the Oracle documentation website at: docs.oracle.com/database/121/AXDBI/toc.htm.

Oracle Automatic Storage Management (ASM) is used for storing the database data files. Refer to the Oracle documentation website at http://docs.oracle.com/database/121/CWAIX/toc.htm for the Oracle Grid Infrastructure installation guide.

Technology skills prerequisites include familiarity with:

- IBM AIX operating system
- Oracle Database software installation and Oracle Database administration
- Storage systems terminology

Architecture

Software

- AIX 7100-04-02 or later
- Single instance Oracle Database and Oracle Client software

Hardware

- IBM Power Systems™ server with POWER8™ processors
- SAN Fabric
- IBM Storwize V7000
- IBM FlashSystem® 900

Network

- 8Gb FC cards and switches
- 10Gb Ethernet cards and switches

Caching storage data

This section provides an overview of the storage caching concept, various cache components, and cache I/O operations.

Storage data caching concept

AIX 7.1 and later versions include server-side caching extension that allows workloads to take advantage of a read-only cache. This extension is available to all generic block devices making the solution available to multiple storage platforms. In addition, there is no need to stop and start the workload to start using the cache. The cache can be created, enabled, disabled, or removed dynamically while the workload is running. Depending on the requirement, a variable number of storage devices (logical unit numbers or LUNs) can be configured with server-side flash cache.

After caching is enabled for the specified storage LUNs, all read requests are redirected to the cache. Figure 1 provides an overview on how an application I/O request is handled for cached and non-cached devices.

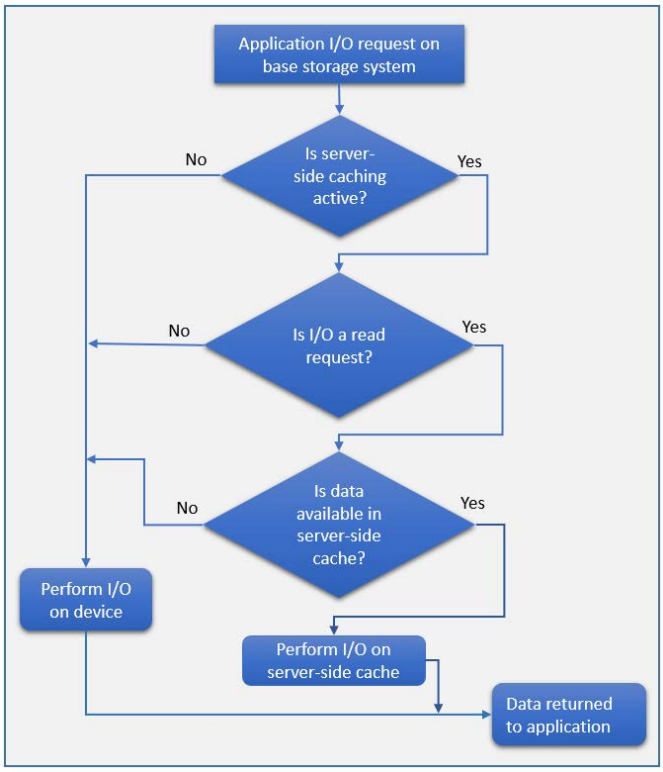


Figure 1: Overview of I/O request using server-side cache

It is important to know that the caching is completely transparent to the application or the workload. No application-level changes are required.

Storage data caching components

The cache devices can be of different types such as built-in solid-state drives (SSDs) in the server, flash devices that are directly attached using serial-attached SCSI (SAS) controllers, or flash devices in the storage area network (SAN).

Component	Usage
Cache management	The `cache_mgt` command is available on AIX and on Virtual I/O Server (VIOS) to start and stop caching, create cache pools, and assign cache partitions to target device.
Cache engine	The cache engine is the most essential part of the caching software suite that decides which blocks to cache and whether to retrieve the data from cache or primary storage.
Cache device	A cache device is an SSD or a flash disk that is used for caching.
Cache pool	A cache pool is a group of cache devices that is used only for storage caching,
Cache partition	A cache partition is a logical cache device that is created from the cache pool.
Target device	A target device is a storage device that is being cached.

Table 1: Server-side flash cache components

Configuration modes for storage data caching

Server-side caching of flash devices is supported in several distinct configurations. These configurations differ in how the cache device is provisioned to the AIX LPAR. Table 2 describes various modes supported by AIX for server-side caching.

Mode	Description
Dedicated	In the dedicated mode, the cache device is directly provisioned to the AIX LPAR.
Virtual	In the virtual mode, the cache device is assigned to VIOS.
N_Port ID Virtualization (NPIV)	In the NPIV mode, the cache device is available as a virtual Fibre Channel (N_Port ID Virtualization) device on the AIX LPAR.

Table 2: Possible configurations for server-side caching

For more information about configuring various storage data caching modes, refer to the IBM Knowledge Center for AIX at: **ibm.com**/support/knowledgecenter/ssw_aix_71/com.ibm.aix.osdevice/caching_configuring.htm

Note: For detailed information about dedicated mode configuration, refer to IBM Knowledge Center for AIX at: **ibm.com**/support/knowledgecenter/ssw_aix_71/com.ibm.aix.osdevice/caching_dedicated_mode.htm

Managing storage data cache

The server-side cache is configured as per user workload requirements. This means that any change in the existing workload might require reconfiguration of the assigned cache. The cache management command (`cache_mgt`) described in Table 1 is used to provision the cache initially or eventually for making any changes to the preconfigured cache.

For our tests we used a SAN attached IBM FlashSystem 900. Figure 2 shows how to create a volume on the FlashSystem 900 storage system.

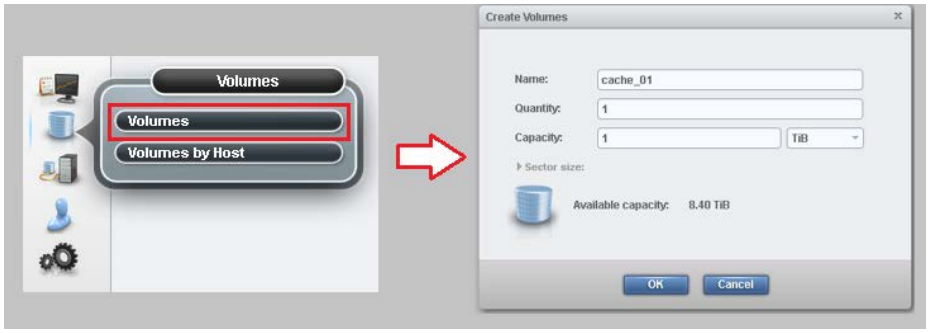


Figure 2: Creating cache volume on FlashSystem 900

After the volume is created, it can then be mapped to the required AIX database host. Figure 3 shows the host-mapping procedure.

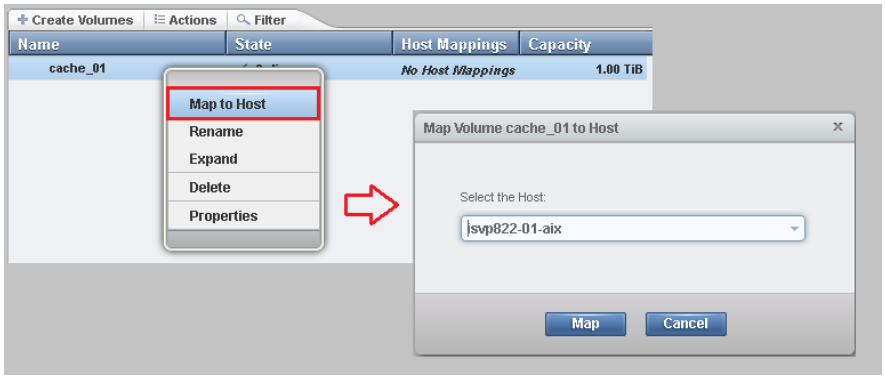


Figure 3: Volume mapping to host on FlashSystem 900

Example 1 shows the sequence of commands used in the lab environment from volume discovery to storage devices assignment to cache along with the output for each command. Application related data files were stored on hdisk6, hdisk7 and the database was hosted on hdisk4.

Step 1: Discover the cache device on the database host.

```
# cfgmgr
```

Step 2: List the devices available to system.

```
# lsdev -Cc disk
hdisk0  Available 03-00-00 SAS RAID 0 Disk Array
hdisk1  Available 03-00-00 SAS RAID 0 Disk Array
hdisk4  Available 00-00-01 MPIO IBM 2076 FC Disk
hdisk6  Available 00-00-01 MPIO IBM 2076 FC Disk
hdisk7  Available 00-00-01 MPIO IBM 2076 FC Disk
hdisk9  Available 00-00-01 MPIO IBM FlashSystem Disk

# cache_mgt device list
hdisk9
```

Step 3: Create a cache pool.

```
# cache_mgt pool create -d hdisk9 -p isvp822-01-pool
Pool isvp822-01-pool created with devices hdisk9.
```

Step 4: Create a partition.

```
# cache_mgt partition create -p isvp822-01-pool -s
120G -P part1
Partition part1 created in pool isvp822-01-pool.
```

Step 5: Assign the partition to a disk (target).

```
# cache_mgt partition assign -t hdisk6 -P part1
Partition part1 assigned to target hdisk6.

# cache_mgt partition assign -t hdisk7 -P part1
Partition part1 assigned to target hdisk7.
```

Example 1: Configuring cache device and storage disks for caching

You can view additional usage options for the `cache_mgt` command by running the `cache_mgt` command with the `help` option.

Lab topology

The server-side flash caching feature can be used for both online transaction processing (OLTP) or decision support systems. In order to showcase the benefits of server-side caching, various tests were performed in the lab environment against a single instance Oracle Database.

Figure 4 shows the lab topology view.

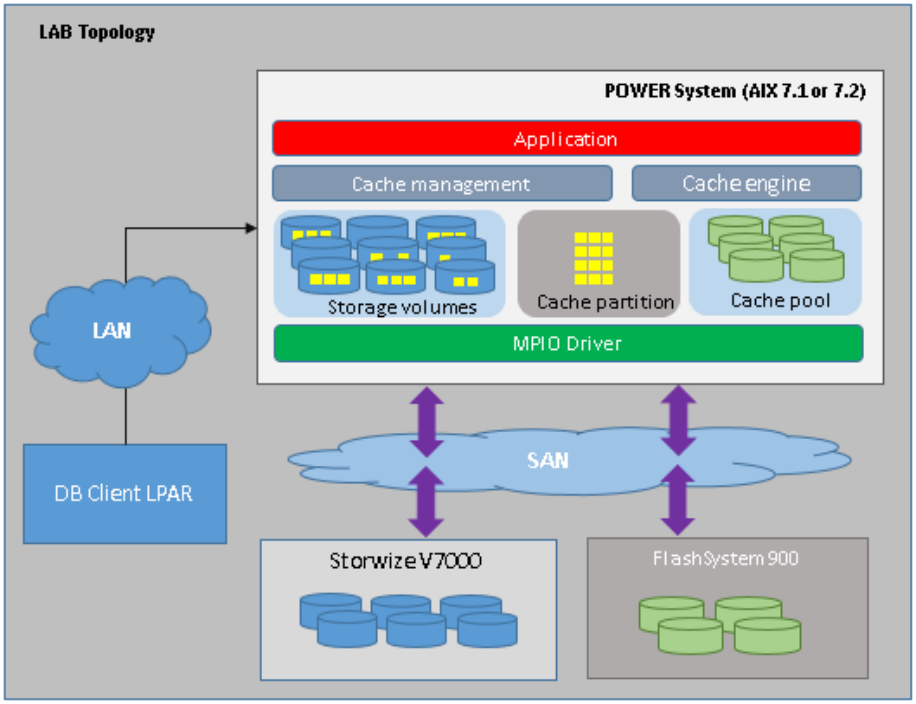


Figure 4: Lab topology view

Overview of lab environment

This section provides a detailed overview of the lab hardware and software used for the testing.

- **SAN fabric**

  The lab environment made use of two 16Gb SAN switches, thus enabling multiple paths to both storage system and database host.

- **Storage**

  A Storwize V7000 system was used for database storage that had two node canisters, each one with a four-port 8Gb Fibre Channel (FC) card. For this testing, 13 managed disks were spread across 95 enterprise-class 10,000 rpm disk drives with 278 GB capacity each in RAID-5 configuration. Three 1 TB LUNs were carved out of the managed disks in order to host the database.

  Additionally, three internal SSDs were used to keep the redo logs and archived redo logs.

- **Database host**

  The database host system was an IBM Power® Systems server with POWER8 processors using a single LPAR with 62 GB RAM. The

LPAR had a single processor chip with 10 cores. Each core was capable of a maximum of eight threads.

- **Database client**

  A single LPAR was configured to act as the client to start the workload over the network.

- **Oracle database**

  For the testing Oracle Database 12*c* version 12.1.0.2.0 was used. Oracle Grid Infrastructure release 12.1.0.2.0 was installed to use ASM for managing database raw devices. No special tuning parameters were used to tune the instance.

- **Workload**

  A *brokerage house* workload simulator was used to perform the OLTP runs. A 1.2 TB sized single instance database was used with three redo logs sizing to 3 GB each. A single LPAR database client was used to generate the load across the network.

  The workload was run with approximately 58 virtual users. Before each run, the pristine copy of the database was restored to keep a consistent starting point.

Workload details

Server-side flash cache is read only cache. Data is populated in the cache based on the access pattern for a particular block. Blocks on which read operation is more frequent are known as *hot blocks*.

Write requests will always be on the primary storage devices.

The cache engine uses an internal mechanism to populate or evict the data.

An OLTP business application simulating *brokerage house* workload was used during the test. The runtime duration of the workload is configurable and various runs were made in order to adjust parameters, such as `queue_depth` for Fibre channel (FC) adapters, for storage LUNs, and the database instance's System Global Area (SGA).

For each run, system statistics such as I/O statistics, processor load, and memory statistics were captured using the `nmon` system performance capture tool. Runs were performed with various server-side cache sizes.

After the workload is started the application eventually reaches its maximum throughput and there is little variation in the TPS. At this point, the system can benefit from the server-side flash cache.

Once the caching engine is enabled it starts caching the data on the flash device. With the data now available on the flash cache device with sub-millisecond latency, the query-response time is greatly reduced and a gradual improvement is seen on the TPS. When the cache is completely filled up to the configured size the system once again attains a maximum throughput level and little variation is seen in the TPS.

The cache engine keeps a constant check of any block modifications occurring on storage devices that are already cached. As cache is read-only in nature, any write requests to a specific block (that might be cached) will go

directly to the primary storage system invalidating the cached block. Now that the primary storage and cached block differ in contents, an entire block from primary storage will be copied back to cache during subsequent read requests.

Benefits of server-side cache

During the lab testing, various cache sizes were configured on the database host. The workload was run with a constant number of virtual users to see the effect of different cache sizes. It was observed that the cache takes approximately 30 to 45* minutes to warm up. During this warm-up period heavy write and read activity was seen since the data not found in cache must be written to cache (write operation to cache). At the same time the data found in cache is read (read operations to cache) and returned to the application.

Figure 5 shows the TPS data captured during a five hour run. The run was started without starting the server-side cache. In both graphs that follow we have presented the measured data (I/O operations per second) as a normalized value. We see that the application quickly attains a saturation level and the TPS at a baseline value. One hour into the run, the cache was started, and there is a gradual improvement seen in the TPS. After the configured cache is completely filled, there is no more room for new data to cache. At this point, the flash cache is serving the application read requests. Data not found in cache will be written to cache based on the caching algorithm. Blocks heavily accessed are seen as *hot blocks*. Blocks transition from cold to warm to hot state and then from hot to warm to cold state depending on their access pattern.

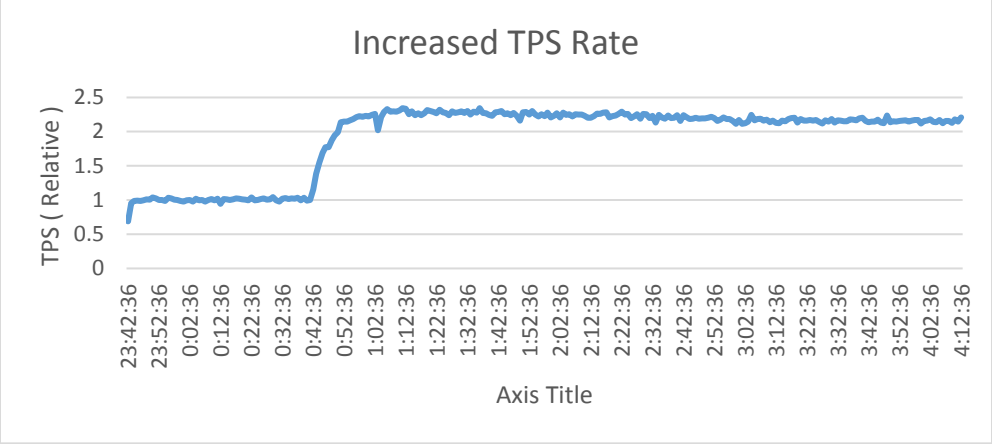


Figure 5: TPS improvement with server-side cache

For the run data showed in Figure 5, a cache size of 128 GB (that is, 10% of the database size) was configured. Figure 6 shows the I/O activity recorded on

* The warm up time of cache can be different depending on the size of the configured device.

the base storage system, and at 0:38:36 shows the increased read activity enabled by the sever-side cache system which leads to the TPS improvement.

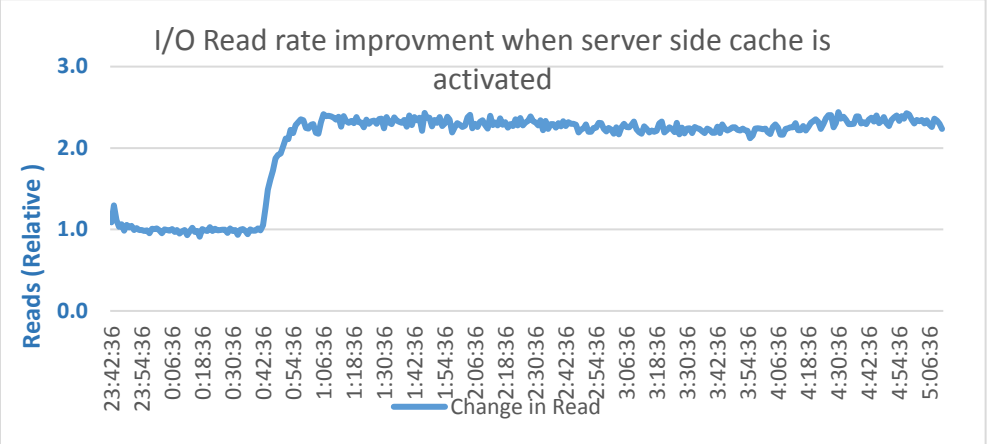


Figure 6: Total storage I/O read rate improvement

In the tests documented above, a flash cache that is 10% of the database size was used. Table 3 shows relative TPS measurements made with larger cache partition sizes used.

Cache size in GB	Cache size percentage relative to DB size	Relative TPS improvement
128	~ 10%	2.4
256	~ 20%	3.2
500	~ 40%	4.6
700	~ 60%	5.2

Table 3: TPS throughput improvement with different server-side cache sizes

Note: The TPS throughput listed in Table 3 is observed in the controlled lab environment. The actual values for TPS might change depending on the workload and overall network traffic.

Summary

There are multiple ways of adding server-side caching to an existing environment. The lowest latency and most cost-effective are internal flash-based SSDs. SSDs might be suitable for a dedicated environment where the local cache is needed for the host server alone.

For a larger and shared environment, IBM FlashSystem 900 might be more suitable, which provides much higher manageability and scaling across multiple servers and workloads. The choice depends on the customer environment and their preferences.

Oracle has confirmed server-side caching as a generic storage technology. Thus, the use of server-side caching in the production environment does not have any certification requirements.

Acknowledgement

Special thanks to the team members directly or indirectly involved in this performance testing. The author wishes to thank Bhargavaram Akula for his valuable contribution in the testing. The author also thanks Majidkhan Remtoula for bringing his expertise in AIX, Oracle Database, and storage during this testing, and Vamshi Thatikonda for his guidance on caching mechanism internals.

Get more information

The following websites provide useful references to supplement the information contained in this paper:

- IBM AIX Knowledge Center
  ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.osdevice/caching_storage_data.htm

- Oracle documentation
  docs.oracle.com/en/database/

- IBM Storwize Knowledge Center
  ibm.com/support/knowledgecenter/ST3FR7

About the author

Shashank Shingornikar is an ISV solutions engineer in IBM Systems. Shashank has experience in Oracle technology products, Oracle data migration, and disaster recovery solutions. Currently, he is working on enabling Oracle solutions for IBM storage.

You can reach Shashank Shingornikar at sshingor@in.ibm.com

IBM Systems
3039 Cornwallis Road
RTP, NC 27709

Produced in the United States of America