Data Migration Guide:
NetApp to IBM Storwize V7000 Unified
Abstract

Data migration is a critical business continuance need not only to move data from older technologies to newer technologies, but it needs to be accomplished with minimum downtime to the applications. Given the complexity of data, metadata, and configuration movement, it is an error-prone and a complex exercise. Because there exist many data migration techniques and tools in the market and there are more than one way to achieve it, customers face the dilemma of finding the best procedure to achieve a reliable data migration for their environment. A reliable migration would migrate the data and its properties from the source to the target and enable customers to move their workloads to the target systematically with minimum disruption. This document is a guide for migrating data from a NetApp system to an IBM Storwize V7000 Unified NAS platform.
# Contents

Abstract .......................................................... iii

**Getting started** .................................................. 1
  How to use this guide ........................................... 1
  Introduction ....................................................... 1
  Migration tools ................................................... 3
  Prerequisites for data migration ................................. 3

**NFSv3 data migration** .......................................... 5
  Migrating NFSv3 data using remote caching ................... 5

**CIFS data migration** ............................................. 11
  Prerequisites for using Robocopy ................................ 11
  Migrating CIFS data using remote caching and Robocopy .... 11

**Limitations of data migration using remote caching** .......... 19

References ............................................................ 21

Appendix A. Comparison of CIFS migration approaches .......... 23

Appendix B. Migrating CIFS data using Robocopy ................ 25

Index ................................................................. 27
Getting started

How to use this guide

If you are migrating...

Do this...

<table>
<thead>
<tr>
<th>NFsv3 data</th>
<th>Get familiar with migration method and tools:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• “Introduction”</td>
</tr>
<tr>
<td></td>
<td>• “Migration tools” on page 3</td>
</tr>
<tr>
<td></td>
<td>• Ensure that migration prerequisites are met:</td>
</tr>
<tr>
<td></td>
<td>• “Prerequisites for data migration” on page 3</td>
</tr>
<tr>
<td></td>
<td>• Review limitations:</td>
</tr>
<tr>
<td></td>
<td>• “Limitations of data migration using remote caching” on page 19</td>
</tr>
<tr>
<td></td>
<td>• Migrate your NFsv3 data:</td>
</tr>
<tr>
<td></td>
<td>• “Migrating NFsv3 data using remote caching” on page 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CIFS data</th>
<th>Get familiar with migration method and tools:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• “Introduction”</td>
</tr>
<tr>
<td></td>
<td>• “Migration tools” on page 3</td>
</tr>
<tr>
<td></td>
<td>• Ensure that migration prerequisites are met:</td>
</tr>
<tr>
<td></td>
<td>• “Prerequisites for data migration” on page 3</td>
</tr>
<tr>
<td></td>
<td>• Review limitations:</td>
</tr>
<tr>
<td></td>
<td>• “Limitations of data migration using remote caching” on page 19</td>
</tr>
<tr>
<td></td>
<td>• Migrate your CIFS data:</td>
</tr>
<tr>
<td></td>
<td>• “Migrating CIFS data using remote caching and Robocopy” on page 11</td>
</tr>
<tr>
<td></td>
<td>• [Optional] If you wish to explore another approach for migrating your CIFS data, use the following information:</td>
</tr>
<tr>
<td></td>
<td>• Appendix A, “Comparison of CIFS migration approaches,” on page 23</td>
</tr>
<tr>
<td></td>
<td>• Appendix B, “Migrating CIFS data using Robocopy,” on page 25</td>
</tr>
</tbody>
</table>

Introduction

Data migration in general can be done using one of the following methods:

- **Block level migration:** The data is migrated at the disk block level where file system data is moved in the following manner:
  1. Flush the file system.
  2. Quiesce the file system.
  3. Take a snapshot of the file system.
  4. Copy the data from the source to the target using disk block level migration.
- **File system migration:** The data is moved as files themselves from the source to the target.
- **Backup and restore:** The data is backed up to tape and then restored to the target.
File system level migration

Each method has its advantages and disadvantages. Choosing the right method depends on the specific requirements of the customer. This guide covers file system level data migration method that has the following advantages:

- File system metadata is transferred to the destination. By moving the data at the file level you do not need to know how the file system is configured at the target.
- File system level migration using remote caching moves data on demand as the application accesses it. Because of that files that are needed first are automatically migrated.
- Applications can be immediately moved after the file metadata is created on the destination allowing data to be moved in the background. This reduces application downtime significantly.
- By moving the applications early (note migration of data can take weeks) the application generates all the new data at the destination which solves the problem of changing data during migration.
- NFSv3 data can be moved entirely without any external applications.
- For CIFS, the data can be moved using remote caching. However, ACLs need to be moved using a third-party application called Robocopy.

The drawbacks of using file system level data migration include:

- For CIFS migration, a Windows machine is required.
- Block level migration could be faster but that does not migrate underlying file system constructs such NSDs (network shared disks) in the case of the GPFS or similar constructs underneath the file system.

IBM Storwize V7000 Unified serves as a file server supporting file serving protocols such as NFS, CIFS, HTTP, and FTP. For file system based migration, the following file system-related information needs to be considered for the migration:

- Authentication
- Migration of files and directories
- Migration of access permissions (ACLs)
- Migration of CIFS shares, NFS exports, etc.

The file system migration method must migrate data and metadata correctly to ensure complete migration.

Additional configuration considerations

Administrators need to consider the following additional configurations that this guide does not cover:

- Quotas
- Backup schedules
- Snapshots and snapshots schedules
- Data placement policies
- Monitoring or logging scripts
- Special legacy settings
  For example, delegated administrative rights and special create masks
Migration tools

The following two tools are used in file system based migration described in this guide:

IBM Storwize V7000 Unified remote caching (ACE - Active Cloud Engine)

Remote caching is a scalable, high-performance, file system caching layer integrated with the GPFS cluster file system that is a part of IBM Storwize V7000 Unified. Remote caching masks wide-area network latencies and outages using GPFS to cache massive data sets, allowing data access and modifications even when remote storage cluster or NAS platform is unavailable. In addition, remote caching performs updates to the remote cluster or NAS platform asynchronously, which allows applications to continue operating while not being constrained by limited outgoing network bandwidth. This caching technique is used here for data migration.

Note: Remote caching was earlier known as Active Cloud Engine (ACE).

The following remote caching features are used for the file system based migration described in this guide:

- **Local update mode:** In this Storwize V7000 Unified remote caching mode, reads at the cache file set bring the data into the cache file set if it is not already in cache. Writes on the cache file set, including the creation of new files, are not sent to the home file set. Files that are updated at the cache file set become local to the cache file set, and any further reads to these files are satisfied locally. In the local update mode, the files that are modified are not pushed back to, or pulled from, the home file set.

- **Prepopulation:** Storwize V7000 Unified remote caching fetches files on demand from the home file set to the remote system cache in real time during normal operation. The prepopulation feature moves files into the cache in batch mode so that they are already present in the cache when they are accessed by an application.

For more information on remote caching, see the Managing remote caching section of the Storwize V7000 Unified Information Center.

Robocopy

Robocopy is a Microsoft software tool provided with the Microsoft Resource Kit for Windows 2003 or as a standard tool in Windows 2008 and later versions. It is used to migrate only CIFS ACLs. CIFS data is migrated using IBM Storwize V7000 Unified remote caching. Robocopy requires dedicated Windows Server attached to the network. Robocopy has the capability to incrementally compare the source and the target ACLs and recopy any ACLs that have changed since the last copy. For more information on Robocopy, see the Robocopy page on the Microsoft website.

Prerequisites for data migration

About this task

Before you begin the migration process, ensure that the following prerequisites are completed.
Procedure

1. **On the target Storwize V7000 Unified system**, check the environment health and that Storwize V7000 Unified is installed correctly using the `lshealth` command.

   ```
   [V7KU]# lshealth
   Host       Sensor                  Status   Value
   mgmt001st001 HOST_STATE          OK       OK
   SERVICE    OK                      All services are running OK
   CTDDB      OK                      CTDDBSTATE_STATE_ACTIVE
   GPFS       OK                      ACTIVE
   SCM        OK                      SCM system running as expected
   NETWORK    OK                      Network interfaces are online
   mgmt002st001 HOST_STATE          OK       OK
   SERVICE    OK                      All services are running OK
   CTDDB      OK                      CTDDBSTATE_STATE_ACTIVE
   GPFS       OK                      ACTIVE
   SCM        OK                      SCM system running as expected
   MGMTNODE_REPL_STATE  OK         OK
   NETWORK    OK                      Network interfaces are online
   V7000      CLUSTER                OK       The component cluster is running OK
   ENCLOSURE  OK                      The component enclosure is running OK
   IO_GRP     OK                      The component io_grp is running OK
   MDISK      OK                      The component mdisk is running OK
   MIGRATE    OK                      The component migrate is running OK
   NODE       OK                      The component node is running OK
   EFSSG1000I The command completed successfully
   ```

   **Note:** If the Storwize V7000 Unified system environment is healthy and it is installed correctly, the `lshealth` command output shows no errors.

2. **On the target Storwize V7000 Unified system**, ensure that a file system is created with total data storing capacity and inode capacity greater than or equal to the capacity on the source system.

   You can create a file system using the following command:

   ```
   [V7KU]# mkfs gpfs0 --createdisks 100gb,0,N=10 -i 10000:1000
   ```

   **Note:** Storage space allocation on the target system must be at least equal to or greater than the allocated space for the source data that you plan to migrate. Migration might take some time, therefore a reasonable estimation of data growth experienced must be considered during storage allocation on the target.

3. **On the target Storwize V7000 Unified system**, ensure that the same authentication directory or mechanism is used as that of the source system.

   You can check the authentication directory or mechanism using the following command:

   ```
   [V7KU]# 1sauth
   AUTH_TYPE = ad
   idMapConfig = 10000000-29999999,1000000
   domain = POLLUX
   idMappingMethod = auto
   clusterName = 99k4927.ibm
   userName = administrator
   idMapRole = master
   adHost = WIN2K8.POLLUX.COM
   passwordServer = *
   krbMode = off
   realm = POLLUX.COM
   EFSSG1000I The command completed successfully.
   ```
NFSv3 data migration

Migrating NFSv3 data using remote caching

About this task

The following procedure describes the steps to migrate your NFSv3 data from a NetApp NAS setup to IBM Storwize V7000 Unified using remote caching. Migration of NFSv3 data to Storwize V7000 Unified using remote caching is a recommended approach because of the following advantages:

- This approach allows customers to move their workloads to the target system without waiting for migration to complete. Therefore, they can start using the target system almost immediately after triggering migration. This way of migrating data is also known as online migration.
- This approach significantly reduces application downtime because applications generate new data at the target that solves the problem of changing data during migration.

The source and target NAS cluster setups used in this procedure are as follows:

- **Source NAS cluster**: NetApp Clustered Data ONTAP 8.2.1
- **Target NAS cluster**: IBM Storwize V7000 Unified 1.6.0.0

**Important**: The steps that you need to execute from the Clustered Data ONTAP system are preceded with "On the source system...". The steps that you need to execute from the Storwize V7000 Unified system are preceded with "On the target system...".

Procedure

1. **On the source system**, identify the NFSv3 shares whose data you plan to migrate using the `volume show` command.

   The `volume show` command lists the NAS volumes.

   ```bash
   ontap-cm-01> volume show
   Vserver Volume Aggregate State Type Size Available Used%
   --------- ----------- ---------- --------- ----- ---------- ------- -------- ------
   vs1   vol0   aggr0   online RW 851.5MB 336.6MB 60%
   vs1   vol1   aggr2   online RW 20MB  18.88MB 5%
   vs1   vol2   aggr3   online RW 20MB  18.88MB 5%
   vs1   vs1root aggr1   online RW 20MB  18.87MB 5%
   4 entries were displayed.
   ```

2. **On the source system**, identify the junction path of the corresponding volume using the `volume show` command.

   The following command output is trimmed for clarity.

   ```bash
   ontap-cm-01> volume show -vserver vs1 -volume vol1
   Vserver Name: vs1
   Volume Name: vol1
   Aggregate Name: aggr2
   Volume Size: 20MB
   Volume Data Set ID: 1026
   Volume Master Data Set ID: 2147484674
   Volume State: online
   Volume Type: RW
   Volume Style: flex
   Is Cluster-Mode Volume: true
   ```
3. Ensure that the NFS shares are exported with root access allowed from the target system (Storwize V7000 Unified) so that the target system can access the source shares that are restricted to root.

   a. **On the target system**, identify the subnet mask to which access is to be allowed over NFS using the `lsnw` command.

   ```
   [V7KU]# lsnw
   Network VLAN ID Network Groups IP-Addresses Routes
   9.118.46.0/23 DEFAULT 9.118.46.159,9.118.46.160 0.0.0.0/0:9.118.46.1
   EFSSG10001 The command completed successfully.
   ```

   b. **On the source system**, allow access from the target system using the `export-policy rule create/modify` command.

   ```
   ontap-cm::> export-policy rule modify -policyname default -ruleindex 1 -vserver vs1 -protocol nfs3 -clientmatch 9.118.46.0/23 -rorule any -rwrule none
   (vserver export-policy rule modify)
   ontap-cm::> export-policy rule show
   (vserver export-policy rule show)
   Vserver Policy Rule Access Client RO
   vs1 default 1 nfs3 9.118.46.0/23 any
   ```

   **Note:** The `export-policy rule create/modify` command allows read-only access to the source system only from the specified IP address.

4. **On the target system**, define a gateway node using the `mkwcachenode` command.

   ```
   [V7KU]# mkwcachenode --nodelist mgmt001st001,mgmt002st001
   EFSSG10001 The command completed successfully.
   ```

5. **On the target system**, create a cache file set in the local update mode for each NFS share from the source system using the `mkwcache` command.

   ```
   [V7KU]# mkwcache gpfs0 mi_fset1 /ibm/gpfs0/mi_fset1 --cachemode local-updates --remotepath 9.118.55.72:/data --noverify -i 10000 -h 100G
   ```

   **Note:** Ensure that you assign enough inodes, using the `-i` option, to accommodate all files from the source system.

   **Note:** Ensure that you allocate enough quota, using the `-h` option, to accommodate all data from the source system.

   - `mi_fset1` is the file set to which the data from source export is to be migrated.
   - `/ibm/gpfs0/mi_fset1` is the junction path for the file set where the data resides after migration.
   - `remotepath` is the export path from source system.

   Use the source system’s public IP in this parameter. The part following “:” is the export junction path which is `/data` in the preceding command.

6. Disable the eviction mode on the cache file set using the `chwcache` command.

   ```
   $ chwcache gpfs0 mi_fset1 --disable eviction
   ```
Note: You need to disable the eviction mode to avoid inadvertently evicting files from the cache that you plan to migrate.

7. [Optional] For application performance considerations, you can bring in the application critical files from the source to the target system. Doing this ensures that there is no delay in the application startup on the target system when the workloads are moved to the target system.

   a. **On the target system**, create a temporary export on the cache file set using the **mkexport** command.

   ```
   [V7KU]# mkexport export1 /ibm/gpfs0/mi_fset1 --nfs "*(rw,no_root_squash,insecure)"
   EFSSG0019I The export export1 has been successfully created.
   EFSSG1000I The command completed successfully.
   ```

   b. **On an NFS client machine**, mount the export using the following command.

   ```
   $ sudo mount 9.118.46.159:/ibm/gpfs0/mi_fset1 /mnt/
   ```

   c. **On an NFS client machine**, generate a list of application critical files using the **find** tool and copy the list in a file.

   ```
   $ find .
   ./dt/ws/.cfg
   ./dt/ws/.cfg/osgi
   ./dt/ws/.cfg/osgi/.bundledata.6
   ```

   **Note:** This list must contain only the files, not paths to the directories.

   d. Prefix the entries in the list with cache file set link path that is used on the target system such that the list entries are changed as follows.

   ```
   /ibm/gpfs0/mi_fset1/dt/ws/.cfg
   /ibm/gpfs0/mi_fset1/dt/ws/.cfg/osgi
   /ibm/gpfs0/mi_fset1/dt/ws/.cfg/osgi/.bundledata.6
   ```

   e. Save the file and copy it to the target system using the SCP upload. The file is copied for the /ftdc directory.

   f. **On the target system**, trigger the prepopulation of the application critical files using the **runprepop** command.

   ```
   $ runprepop gpfs0 mi_fset1 -listfile /ftdc/application_files
   EFSSG1011I This is an asynchronous operation.
   Please refer to man page for details on checking its status.
   EFSSG1000I The command completed successfully.
   ```

   g. **On the target system**, check the status of the prepopulation task using the **lsprepop** command.

   ```
   $ lsprepop
   EFSSG1000I The command completed successfully.
   When the **lsprepop** command output shows status as Finished, prepopulation can be considered complete. This step must be performed you wish to get the application critical data prior to migrating the applications to target system.

8. Do the following steps.
a. Shut down the applications.

b. **On the source system**, export the source file system as read-only for access from all clients using the `export-policy rule modify` command to prevent any modifications at the source and ensure that the local permissions are also read-only.

```
ontap-cm::> export-policy rule modify -policyname default -ruleindex 1 -vserver vs1 -protocol nfs3 -clientmatch 0.0.0.0/0 -rorule any -rwrule none
```

9. **On an NFS client machine**, bring in the directory structure from the source system to the target system by doing a recursive listing on `/mnt` that is mounted in the previous step.

```
$ cd /mnt/

$ ls -R
New Text Document - Copy.txt  rktools - Copy.exe  robocopy commands - Copy.txt
New Text Document.txt  rktools.exe  robocopy commands.txt
```

This step activates the cache file set and brings in the files and directory structure from the source system to the target system.

10. Identify and create list of files to be migrated from the source system.

a. Go to the mounted directory and list the files to be migrated using the `find` tool.

```
# find .
./data
./data/f
./data/lun0
./test
./sdfsdf
./f
./data1
./anotherf
./.snapshot
```

b. Copy this list in a file after excluding the files and directories that belong to other volumes. Also, exclude any LUN files.

**Note:** This list must contain only the files, not paths to the directories.

c. Prefix the entries in the list with cache file set link path that is used on the target system such that the list entries are changed as follows.

```
/ibm/gpfs0/mi_fset1/test
/ibm/gpfs0/mi_fset1/sdfsdf
/ibm/gpfs0/mi_fset1/f
/ibm/gpfs0/mi_fset1/anotherf
```

d. Save the file and copy it to the target system using the SCP upload. The file is copied for the `/ftdc` directory.

**Note:** You must unmount the temporary NFS mount created in the preceding steps now to prevent any accidental writes.

11. **On the target system**, create an equivalent NFS export for each source export using the `mkexport` command.

```
$ mkexport export1 /ibm/gpfs0/mi_fset1/exp1 --nfs "*(rw,no_root_squash)"
```

EFSSG0019I The export export1 has been successfully created.

EFSSG1000I The command completed successfully.

You can obtain the list of source exports using the `volume show` command as shown in step 1 on page 5.

12. **From an NFS client machine**, unmount the source exports from the client and mount exports from the target system. This effectively moves NFS clients to the target system.
Note: It is recommended to maintain the connectivity between two systems till migration is complete. If the cache gets disconnected from the home and if the data is present in the cache, it is served. Otherwise, it results in an I/O failure. In case of a disconnection between the source and the target system during migration, the migration process resumes when the cache comes back to the disconnected mode.

Note: At this point, you are ready to move your applications to the target system even if migration is not complete.

13. On the target system, trigger remote caching to copy files listed in this earlier step from the source to the target system using the `runprepop` command.

```bash
$ runprepop gpfs0 mi_fset1 -listfile /ftdc/mi_fset1_files
EFSSG1011I This is an asynchronous operation.
Please refer to man page for details on checking its status.
EFSSG1000I The command completed successfully.
```

This command triggers the data prefetch and starts migration. Only the files that are in the list are migrated. Other files are migrated when the application demands them.

14. On the target system, check the prepopulation status using the `lsprepop` command.

```bash
$ lsprepop
Cluster ID filesystem FilesetName Status Last update Timestamp Files Failed Message
792079394835903749 gpfs0 mi_fset1 RUNNING 6/27/14 2:31 PM 0 RUNNING
EFSSG1000I The command completed successfully.
```

The prepopulation is complete when the `lspreop` command output shows the status as `FINISHED`.

```bash
$ lsprepop
Cluster ID filesystem FilesetName Status Last update Timestamp Files Failed Message
792079394835903749 gpfs0 mi_fset1 FINISHED 6/27/14 3:26 PM 0
EFSSG1761 Cache pre-population completed successfully
```

15. [Optional] Convert the migrated file set into a regular file set anytime after the migration using the `chwcach`e command.

```bash
$ chwcache gpfs0 mi_fset1 --disablewcache
EFSSG0071I Fileset mi_fset1 changed successfully.
EFSSG1000I The command completed successfully.
```

Note: After using the `--disablewcache` option, any exports on the file set need to be deleted and recreated again. This task requires application to be shut down. Therefore, it must be planned accordingly. Use the `rmexport` command to delete the exports and then use the `mkexport` command to recreate the exports.

You can confirm that the file set is now a regular file set using the `lsfset` command.

```bash
$ lsfset gpfs0 -r
EFSSG0015I Refreshing data.
ID Name Status Path Is independent Creation time Comment Timestamp
0 root Linked /ibm/gpfs0 yes 3/7/14 7:26 PM root fileset 6/27/14 3:36 PM
5 mi_fset1 Linked /ibm/gpfs0/mi_fset1 yes 6/26/14 11:44 AM 6/27/14 3:36 PM
EFSSG1000I The command completed successfully.
```

Note: These steps must be performed for all volumes on the source system for which data is to be migrated. It is recommended to map one volume on the source system to one cache file set on the target Storwize V7000 Unified system.
CIFS data migration

Prerequisites for using Robocopy

Ensure that the following prerequisites are met before you use Robocopy for migrating your CIFS data.

- The Windows Time service must be in sync with the domain source and the target.
- The migration server must be a part of the domain.
- The migration user:
  - must have administrator rights, preferably domain administrator rights.
  - must be logged on with that account.
  - must have permission to read all files and ACLs.

There are several options in Robocopy which are used to copy file information. For more information, see the documentation from Microsoft Windows 2003 Resource Kit.

Migrating CIFS data using remote caching and Robocopy

About this task

The following procedure describes the steps to migrate your CIFS data from a NetApp NAS setup to IBM Storwize V7000 Unified using remote caching and Robocopy. When migrating CIFS data, you can use remote caching to migrate data and metadata, but you cannot migrate CIFS ACLs using this approach. You can use Robocopy to migrate CIFS ACLs because it provides you the capability to compare and copy security related information between source and target files.

Note: All CIFS shares that you plan to migrate, must also be exported as NFS shares for using remote caching to migrate data and metadata.

Before using Robocopy ensure that you are familiar with the prerequisites for using Robocopy for migrating your CIFS data.

The source and target NAS cluster setups used in this procedure are as follows:

- **Source NAS cluster**: NetApp Clustered Data ONTAP 8.2.1
- **Target NAS cluster**: IBM Storwize V7000 Unified 1.6.0.0

Important: The steps that you need to execute from the Clustered Data ONTAP system are preceded with "On the source system...". The steps that you need to execute from the Storwize V7000 Unified system are preceded with "On the target system...".

Procedure

1. **On the source system**, identify the CIFS shares exports that you plan to migrate using the `cifs share show` command.

   ```
   ontpcm::> cifs share show
   Vserver  Share  Path  Properties  Comment  ACL
   ---------  -------  ------  ---------  -------  -------
   vs1        admin$  /  browsable  -       -
   ```

   © Copyright IBM Corp. 2015
2. **On the target system**, create NFS exports for the CIFS shares that you plan to migrate using the `mkexport` command.

   ```
   [V7KU]$ mkexport export1 /ibm/gpfs0/mi_fset1 --nfs 
   "*(rw,no_root_squash,insecure)"
   EFSSG0019I The export export1 has been successfully created.
   EFSSG1000I The command completed successfully.
   ```

3. Ensure that CIFS shares exports are exported with root access allowed from the target system (Storwize V7000 Unified) so that the target system can access the source shares that are restricted to root.

   **a. On the target system**, identify the subnet mask to which access is to be allowed over NFS using the `lsnw` command.

   ```
   [V7KU]$ lsnw
   Network VLAN ID Network Groups IP-Addresses Routes
   9.118.46.0/23 DEFAULT 9.118.46.159,9.118.46.160 0.0.0.0/0:9.118.46.1
   EFSSG1000I The command completed successfully.
   ```

   **b. On the source system**, allow access from the target system using the `export-policy rule create/modify` command.

   ```
   ontap-cm::> export-policy rule modify -policyname default -ruleindex 1 
   -vserver vs1 -protocol cifs -clientmatch 9.118.46.0/23 -rorule any -rwrule none
   (vserver export-policy rule modify)
   ontap-cm::> export-policy rule show
   (vserver export-policy rule show)
   Vserver Name Index Protocol Match RO
   -------------- ------- --------- ---------
   vs1 default 1 cifs 9.118.46.0/23 any
   ```

   **Note:** The `export-policy rule create/modify` command allows read-only access to the source system only from the specified IP address.

4. **On the target system**, define a gateway node using the `mkwcachenode` command.

   ```
   [V7KU]$ mkwcachenode --nodelist mgmt001st001,mgmt002st001
   EFSSG1000I The command completed successfully.
   ```

5. **On the target system**, create a cache file set in the local update mode for each NFS share from the source system using the `mkwcache` command.

   ```
   [V7KU]$ mkwcache gpfs0 mi_fset1 /ibm/gpfs0/mi_fset1 
   --cachemode local-updates --remotepath 9.118.55.72:/data --noverify -i 10000 -h 100G
   ```

   **Note:** Ensure that you assign enough inodes, using the `-i` option, to accommodate all files from the source system.

   **Note:** Ensure that you allocate enough quota, using the `-h` option, to accommodate all data from the source system.

   * mi_fset1 is the file set to which data from the source export is to be migrated.
• /ibm/gpfs0/mi_fset1 is the junction path for the file set where the data resides after migration.
• remotepath is the export path from source system.

Use the source system's public IP in this parameter. The part following “:” is the export junction path is /data in the preceding command.

6. Disable the eviction mode on the cache file set using the chwcache command.

   $ chwcache gpfs0 mi_fset1 --disableeviction

   **Note:** You need to disable the eviction mode to avoid inadvertently evicting files from the cache that you plan to migrate.

7. **On the target system**, create a temporary export on the cache file set.

   [V7KU]# mkexport export1 /ibm/gpfs0/mi_fset1 --nfs "*(rw,no_root_squash,insecure)"

   EFSSG0019I The export export1 has been successfully created.
   EFSSG1000I The command completed successfully.

8. **On a client machine**, mount the export using the following command.

   $ sudo mount 9.118.46.159:/ibm/gpfs0/mi_fset1 /mnt/

9. **On a client system**, bring in the directory structure from the source system to the target system by doing a recursive listing on /mnt that is mounted in the previous step.

   $ cd /mnt/

   $ ls -R

   New Text Document - Copy.txt  rktools - Copy.exe  robocopy commands - Copy.txt
   New Text Document.txt  rktools.exe  robocopy commands.txt

   This step activates the cache file set and brings in the files and directory structure from the source system to the target system.

10. **On the target system**, change the export created in this earlier step such that it can also export CIFS using the chexport command.

    [V7KU]# chexport export1 --cifs 'admin users=[domain]\Administrator'

    EFSSG0022I Protocol CIFS is configured for share export1.
    EFSSG1000I The command completed successfully.

11. For all folders which have inherited permissions set, make a note of the ACLs on the source system and use them on the target system as follows.

    a. **On the source system**, right-click the export (\\9.118.55.74\migrate in this example) and then click Properties > Security > Advanced. The Advanced Security Settings window opens.
    b. **On the source system**, under the Permissions tab, make a note of all the permissions.
    c. **On the target system**, right-click the export (\\9.118.46.159\export1 in this example) and then click Properties > Security > Advanced. The Advanced Security Settings window opens.
    d. **On the target system**, under the Permissions tab, click Change Permissions to change the permissions according to those on the source system.

12. **On the migration server**, migrate ACLs using Robocopy.

    robocopy \9.118.55.74\migrate \9.118.46.159\export1 \COPY:ATSO /seccfix /Z /E /MT:32 /R:5 /W:3 /sl /log:logfile

    The Robocopy options are as follows:

    • /E: Copy all subdirectories including empty ones.
    • /MT: Create multi-threaded copies with N threads. N must be an integer between 1 and 128. The default value for N is 8.
• /R: Specifies the number of retries on failed copies. The default value is 1,000,000 (one million retries).
• /W: Specifies the wait time between retries, in seconds. The default value is 30 (wait time 30 seconds).
• /s1: Copies the symbolic link instead of the target.
• /COPY:ATSO: Copy file information.
  – A: Attributes
  – T: Timestamps
  – S: Security - NTFS ACLs
  – O: Owner information

**Note:** Ensure that the administrator or the user that is running Robocopy has read access on all files and folders on the source system that you are migrating. If this user does not have explicit access, the user must be given backup right and Robocopy must be run with the /B flag.

After this command finishes executing, you can view the log file to review any errors. For more information about using Robocopy, see “Prerequisites for using Robocopy” on page 11.

13. Do the following steps.
   a. Shut down the applications.
   b. **On the source system**, export the source file system as read-only for access from all clients using the `export-policy rule modify` command to prevent any modifications at the source and ensure that the local permissions are also read-only.

```
ontap-cm::> export-policy rule modify -policyname default -ruleindex 1 \n  -vserver vs1 -protocol nfs3 -clientmatch 0.0.0.0/0 -rorule any -rwrule none
```

14. Some ACLs on source may change during the Robocopy operation. To ensure that the target system gets the latest ACLs, disable writes on the source system by making the exports read-only and execute the Robocopy command again as shown in this earlier step.

15. **[Optional]** For application performance considerations, you can bring in the application critical files from the source to the target system. Doing this ensures that there is no delay in the application startup on the target system when the workloads are moved to the target system.
   a. **On a client system**, generate a list of application critical files using the `find` tool and copy the list in a file.

```
$ find .
./dt/ws/.cfg
./dt/ws/.cfg/osgi
./dt/ws/.cfg/osgi/.bundledata.6
```

**Note:** This list must contain only the files, not paths to the directories.

b. Prefix the entries in the list with cache file set link path that is used on the target system such that the list entries are changed as follows.

```
/ibm/gpfs0/mi_fset1/dt/ws/.cfg
/ibm/gpfs0/mi_fset1/dt/ws/.cfg/osgi
```
Save the file and copy it to the target system using the SCP upload. The file is copied for the /ftdc directory.

d. **On the target system**, trigger the prepopulation of the application critical files using the `runprepop` command.

```
$ runprepop gpfs0 mi_fset1 -listfile /ftdc/application_files
```

EFSSG1011I This is an asynchronous operation.
Please refer to man page for details on checking its status.
EFSSG1000I The command completed successfully.

e. **On the target system**, check the status of the prepopulation task using the `lsprepop` command.

```
$ lsprepop
```

<table>
<thead>
<tr>
<th>Cluster ID</th>
<th>filesystem</th>
<th>FilesetName</th>
<th>Status</th>
<th>Last update Timestamp</th>
<th>Files</th>
<th>Failed</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>792079394835903749</td>
<td>gpfs0</td>
<td>mi_fset1</td>
<td>RUNNING</td>
<td>6/27/14 2:31 PM</td>
<td>0</td>
<td>0</td>
<td>RUNNING</td>
</tr>
</tbody>
</table>

EFSSG1000I The command completed successfully.

When the `lsprepop` command output shows status as Finished, prepopulation can be considered complete. This step must be performed if you wish to get the application critical data prior to migrating the applications to target system.

**Note:** It is recommended to maintain the connectivity between two systems till migration is complete. If the cache gets disconnected from the home and if the data is present in the cache, it is served. Otherwise, it results in an I/O failure. In case of a disconnection between the source and the target system during migration, the migration process resumes when the cache comes back to the disconnected mode.

16. **On the target system**, create CIFS exports for every export on the source system using the `mkexport` command.

```
$ mkexport cifsexport1 /ibm/gpfs0/mi_fset1/exp1 --cifs
```

EFSSG0019I The export cifsexport1 has been successfully created.
EFSSG1000I The command completed successfully.

You can obtain the list of source exports using the `cifs share show` command as shown in step [1 on page 11](#).

**Note:** At this point, you are ready to move your applications to the target system even if migration has not completed.

17. **From a client system**, unmount the source exports from the client and mount exports from the target system. This effectively moves clients to the target system.

18. Identify and create list of files to be migrated from the source system.

a. Go to the mounted directory and list the files to be migrated using the `find` tool.

```
# find .
./data
./data/f
./data/lun0
./test
./sdfsdf
./f
./data1
./anotherf
./.snapshot
```

CIFS data migration 15
b. Copy this list in a file after excluding the files and directories that belong to other volumes. Also exclude any LUN files.

**Note:** This list must contain only the files, not paths to the directories.

c. Prefix the entries in the list with cache file set link path that is used on the target system such that the list entries are changed as follows.

```
/ibm/gpfs0/mi_fset1/test
/ibm/gpfs0/mi_fset1/sdfsdf
/ibm/gpfs0/mi_fset1/f
/ibm/gpfs0/mi_fset1/anotherf
```

d. Save the file and copy it to the target system using the SCP upload. The file is copied for the /ftdc directory.

**Note:** You must unmount the temporary NFS mount created in the preceding steps now to prevent any accidental writes.

19. **On the target system,** trigger remote caching to copy files listed in this earlier step from the source to the target system using the `runprepop` command.

```
$ runprepop gpfs0 mi_fset1 -listfile /ftdc/mi_fset1_files
```

EFSSG1011I This is an asynchronous operation.

Please refer to man page for details on checking its status.

EFSSG1000I The command completed successfully.

This command triggers the data prefetch and starts migration. Only the files that are in the list are migrated. Other files are migrated when the application demands them.

20. **On the target system,** check the prepopulation status using the `1sprepop` command.

```
$ 1sprepop
```

Cluster ID filesystem FilesetName Status Last update Timestamp Files Failed Message
7920739435930749 gpfs0 mi_fset1 RUNNING 6/27/14 2:31 PM 0 RUNNING
EFSSG1000I The command completed successfully.

The prepopulation is complete when the `1sprepop` command output shows the status as FINISHED.

```
$ 1sprepop
```

Cluster ID filesystem FilesetName Status Last update Timestamp Files Failed Message
7920739435930749 gpfs0 mi_fset1 FINISHED 6/27/14 3:26 PM 0
EFSSG4176I Cache pre-population completed successfully

21. **[Optional]** Convert the migrated file set into a regular file set anytime after the migration using the `chwcache` command.

```
$ chwcache gpfs0 mi_fset1 --disablewcachecache
```

EFSSG0071I File set mi_fset1 changed successfully.

EFSSG1000I The command completed successfully.

**Note:** After using the `--disablewcachecache` option, any exports on the file set need to be deleted and recreated again. This task requires application to be shut down. Therefore, it must be planned accordingly. Use the `rmexport` command to delete the exports and then use the `mkexport` command to recreate the exports.

You can confirm that the file set is now a regular file set using the `1sfset` command.

```
$ 1sfset gpfs0 -r
```

EFSSG0015I Refreshing data.
ID Name Status Path Is independent Creation time Comment
0 root Linked /ibm/gpfs0 yes 3/7/14 7:26 PM root fileset 6/27/14 3:36 PM
5 mi_fset1 Linked /ibm/gpfs0/mi_fset1 yes 6/26/14 11:44 AM 6/27/14 3:36 PM

EFSSG1000I The command completed successfully.
Note: These steps must be performed for all volumes on the source system for which data is to be migrated. It is recommended to map one volume on the source system to one cache file set on the target Storwize V7000 Unified system.

What to do next

You can also use only Robocopy to migrate your CIFS data and ACLs. For more information, see Appendix B, “Migrating CIFS data using Robocopy,” on page 25.

For a comparison of the two approaches for migrating CIFS data, see Appendix A, “Comparison of CIFS migration approaches,” on page 27.
Limitations of data migration using remote caching

The following limitations apply when you use remote caching for migrating your NFS or CIFS data from a NetApp NAS setup to IBM Storwize V7000 Unified as described in this guide.

- If you do not move your critical application data before moving applications, it can have significant impact on performance while the Storwize V7000 Unified system fetches the files requested from the source.
- You cannot link any file set, whether dependent or independent, into a migrated file set because it is a cache file set unless you disable remote caching.
- NFSv4 migration is not covered in the procedures described in this guide.
- When using Active Directory with NIS, Storwize V7000 Unified does not support CIFS users that are not configured in NIS.
- SAMBA PDC or NT4 authentication is not supported.
- Any configuration information such as snapshots or backup policies is not migrated. You need to reconfigure such information on the target.
- The hard links on the source system are not copied as hard links to Storwize V7000 Unified.
- Sparse files are not maintained after migration. Full blocks are allocated for a file after migration depending on its size.
- You cannot migrate special files such as device files and sockets.
- Converting a cache file set into a regular file set after migration involves downtime because you need to temporarily delete the exports.
- The Clustered Data ONTAP information and the Clustered Data ONTAP command examples used in this document are based on the Clustered Data ONTAP Simulator software.
References

- IBM Scale Out Network Attached Storage (SONAS) Documentation
- IBM Storwize V7000 Unified Documentation
- IBM Spectrum Scale Documentation (Formerly known as GPFS)
- SONAS File Migration Charts (Technical Enablement Workshop, Unit 18)
- Performance Guide (System Storage Service Offering 1.17)
Appendix A. Comparison of CIFS migration approaches

<table>
<thead>
<tr>
<th></th>
<th>Approach 1 (CIFS migration using remote caching and Robocopy)</th>
<th>Approach 2 (CIFS migration using Robocopy only)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>• Applications can move to the target system even while the data migration is in progress. Therefore, you can start using the target system much early resulting in a minimal application downtime.</td>
<td>• Complete data migration is faster than that in approach 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Single command is used for data and ACL migration.</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td>• Complete data migration time is longer than that in approach 2.</td>
<td>• Applications cannot be moved to the target system until the entire data migration is complete. This could take a very long time depending on the data, resulting in a longer application downtime.</td>
</tr>
<tr>
<td></td>
<td>• Converting a cache file set into a regular file set after migration involves downtime because the exports need to be temporarily deleted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Separate administrative steps are required for data and ACL migration.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** IBM recommends approach 1 for migrating your CIFS data.
Appendix B. Migrating CIFS data using Robocopy

About this task

The following procedure describes the steps to migrate your CIFS data from a NetApp NAS setup to IBM Storwize V7000 Unified using Robocopy.

Before using Robocopy ensure that you are familiar with the prerequisites for using Robocopy for migrating your CIFS data.

The source and target NAS cluster setups used in this procedure are as follows:

- **Source NAS cluster**: NetApp Clustered Data ONTAP 8.2.1
- **Target NAS cluster**: IBM Storwize V7000 Unified 1.6.0.0

Important: The steps that you need to execute from the Clustered Data ONTAP system are preceded with "On the source system...". The steps that you need to execute from the Storwize V7000 Unified system are preceded with "On the target system...".

Procedure

1. **On the source system**, identify the CIFS shares whose data you plan to migrate using the `cifs share show` command.

   ```
   ontpcm::> cifs share show
   
   Vserver | Share | Path | Properties | Comment | ACL
   --------|-------|------|------------|---------|------
   vs1     | admin$ | /browsable | -       | -       | -
   vs1     | c$     | /oplocks | -       | BUILTIN\Administrators | Full Control
   vs1     | ipc$   | /browsable | -       | -       | -
   vs1     | share1 | /oplocks | -       | Everyone | Full Control
   vs1     | testshare | /oplocks | -       | Everyone | Full Control
   
   ```

2. **On the target system**, create file sets to which the data from source you plan to migrate and then export them as CIFS shares.

   ```
   [V7KU]# mkfset gpfs0 mi_fset1 --link --junction /ibm/gpfs0/ mi_fset1
   (1/3) Creating file set
   EFSSG0070I File set mi_fset1 created successfully.
   (2/3) Linking file set
   EFSSG0078I File set mi_fset1 successfully linked.
   (3/3) Setting owner
   EFSSG1000I The command completed successfully.
   
   [V7KU]# mkexport cifsexport /ibm/gpfs0/mi_fset1 --cifs
   EFSSG0019I The export cifsexport has been successfully created.
   EFSSG1000I The command completed successfully.
   ```

3. **On the migration server**, migrate your CIFS data and ACLs using Robocopy.

   ```
   robocopy \\9.118.46.159\export1 \\9.118.55.74\migrate /copy:ATSO /secfix /Z /E /MT:32 /R:5 /N:3 /s1 /log:logfile
   
   The Robocopy options are as follows:
   ```
• /E: Copy all subdirectories including empty ones.
• /MT: Create multi-threaded copies with N threads. N must be an integer between 1 and 128. The default value for N is 8.
• /R: Specifies the number of retries on failed copies. The default value is 1,000,000 (one million retries).
• /W: Specifies the wait time between retries, in seconds. The default value is 30 (wait time 30 seconds).
• /s1: Copies the symbolic link instead of the target.
• /COPY:ADTSO: Copy file information.
  – A: Attributes
  – D: Data
  – T: Timestamps
  – S: Security - NTFS ACLs
  – 0: Owner information

**Note:** Ensure that the administrator or the user that is running Robocopy has read access on all files and folders on the source system that you are migrating. If this user does not have explicit access, the user must be given backup right and Robocopy must be run with the /B flag.

After this command finishes executing, you can view the log file to review any errors. For more information about using Robocopy, see “Prerequisites for using Robocopy” on page 11.

4. Some ACLs on source may change during the Robocopy operation. To ensure that the target system gets the latest ACLs, disable writes on the source system by making the exports read-only and execute robocopy command again as shown in this preceding step.

5. **On the target system,** create CIFS exports for every export on the source system using the `mkexport` command.

```
$ mkexport cifsexport1 /ibm/gpfs0/mi_fset1/exp1 --cifs
EFSSG0019I The export cifsexport1 has been successfully created.
EFSSG1000I The command completed successfully.
```

Migration can now be termed complete and you can move your applications to the target system.

6. **From a CIFS client machine,** unmount the source exports from the client and mount exports from the target system. This effectively moves NFS clients to the target system.

**What to do next**

Validate files, access, and permissions for a sample of target files and users to ensure accuracy.
# Index

## A
- abstract iii
- authors iii

## B
- before you begin 3

## C
- CIFS
  - migration 11, 25
- CIFS migration
  - approaches 23
  - comparison 23
  - remote caching and Robocopy 11
    - Robocopy only 25

## D
- data migration
  - limitations 19

## H
- high-level process 1

## I
- introduction 1

## M
- migration limitations 19
- migration tools 3
- more information 21

## N
- NFSv3
  - migration 5
- NFSv3 migration
  - remote caching 5

## O
- overview 1

## P
- prerequisites 3
  - CIFS migration 3
  - NFS migration 3

## R
- references 21
- remote caching
  - local update mode 3
  - prepopulation 3
  - remote caching migration
    - limitations 19
  - Robocopy 3