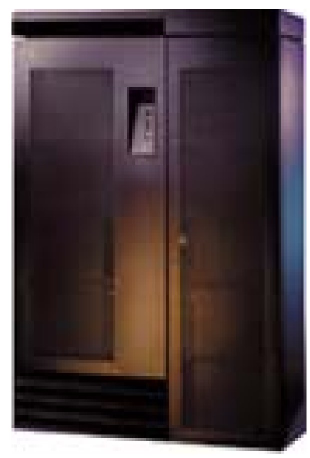
Storage Management for IBM DB2 UDB:
Snapshot Backup and Recovery
With the
IBM TotalStorage™ Enterprise Storage Server



DISCLAIMERS

The information in this publication is provided "AS IS" WITHOUT WARRANTY OF ANY KIND, implied or otherwise, including but not limited to any warranties of merchantability or fitness for a particular purpose. The responsibility for use of this information or implementation of any of these techniques lies with the customer and depends on the customer's ability to evaluate and integrate these techniques into their operating environment.

Accordingly, the recipient assumes all risks arising from the use of this information. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. While IBM may have reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.

The performance data contained in this document were obtained under a controlled, isolated environment. The results obtained in your operating environment may vary significantly. Accordingly, this performance data does not constitute a performance guarantee or warranty.

Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Copyrights

The following terms are either registered trademarks or trademarks of the IBM Corporation in the United States or other countries or both:
IBM, Enterprise Storage Server, ESS, ESCON, FlashCopy, Peer-to-Peer Remote Copy, PPRC, Seascape, StorWatch, TotalStorage, Netfinity, Tivoli, Tivoli Management Environment, TME, DB2, DB2 Universal Database, DB2 UDB

The following terms are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries:
Microsoft, Windows, Windows NT, BackOffice, Windows 2000

Pentium, Pentium III Xeon, and Intel are registered trademarks of the Intel Corporation.

McDATA is a registered trademark of McDATA Corporation.

QLogic is a registered trademark of QLogic Corporation.

Other product names mentioned herein may be trademarks of their respective companies.

Table of Contents

Executive Summary

This white paper describes how a customer can use the IBM TotalStorage™ Enterprise Storage Server's Advanced Copy Services to perform snapshot backup and recovery in a IBM DB2 UDB environment. One of these Copy Services, FlashCopy, is capable of performing backups of multi-terabyte databases in minutes. It can also help improve uptime by providing the ability to recover in minutes, instead of hours or days. Another Copy Service, Peer-to-Peer Remote Copy(**PPRC**), provides a synchronous remote mirror to protect from disasters. When used in combination, these two Copy Services have the ability to provide multiple instant copies of the database and still protect against disaster.

IBM DB2 UDB is a universal database supporting a wide variety of platforms and applications. With every new release, DB2 UDB continues to grow in the enterprise-class server segment. When a database is used for mission-critical applications, it requires an enterprise-class storage subsystem. For DB2 UDB customers, the IBM TotalStorage™ Enterprise Storage Server (**ESS**) can provide the reliability, scalability, and features required for a mission-critical application.

1 Introduction

The need for continuous availability requires innovative techniques for speedy backup and restore of large databases. These techniques can take advantage of features of ESS working together with DB2 UDB to allow the near-instantaneous creation of copies of data, without compromising performance of online operations. This paper describes the use of the IBM TotalStorage™ Enterprise Storage Server (ESS) and it's Advanced Copy Services to perform Snapshot Backup and Recovery with DB2 UDB 7.1.

2 Customer Requirements

Backup Requirements:
As customers rapidly move to a 24x7 global computing environment, the backup window is shrinking just as rapidly. Customers can no longer afford to impact their production server for several hours to perform backups every night. This becomes especially true as databases become larger and larger. Online backups improve the availability of the database, but costs valuable host CPU resources, disk resources, and network resources.

Recovery Requirements:
Should an error occur or disaster strike and recovery is needed, a restore often takes more time than backup. During this time, the database will often be unavailable.

Despite planned and unplanned outages, business needs often require that databases must be available within minutes. While disasters and hardware failures are rare, logical errors and software errors are not uncommon. These errors require a time-consuming restore as well.

As databases become larger and contain more mission-critical data, they increasingly require higher availability. Conversely, the time required to back up and restore becomes longer.

3 Enterprise Storage Server (codenamed “Shark”) Overview

To help meet the challenges inherent in an enterprise-class database solution, IBM introduced the Enterprise Storage Server (ESS), codenamed “Shark.” Its reliability, high performance, and Advanced Copy Services make it well equipped to meet the challenges.

The Enterprise Storage Server is designed to provide enterprise-class storage for the applications across an entire corporation. Based on a third-generation Seascape architecture, the ESS is designed to deliver scalability, availability, performance, and connectivity to most leading operating systems that are hosting enterprise class applications and databases. Availability and performance can be further improved by using its Advanced Copy Services features, including FlashCopy and Peer-to-Peer Remote Copy (PPRC).

3A Enterprise Storage Server Architecture

IBM designed the Enterprise Storage Server specifically for enterprises that require high reliability and continuous availability. This focus on reliability and availability clearly shows in the advanced architecture and features of the ESS.

Advanced Internal Architecture:
Each ESS consists of two complete storage subsystems attached to fault-tolerant disks in a clustered configuration. Each of the two subsystems, or clusters, contains two buses to connect to the hosts, and four internal storage controllers to attach to the disks. The ESS also contains vast amounts of cache, dedicated NonVolatile Storage (NVS), and intelligent load-balancing and parallelism. The result is a high performance, highly available storage server that is designed to avoid single points of failure. Figure 1 shows the multiple paths to data available within the internal architecture of the ESS:

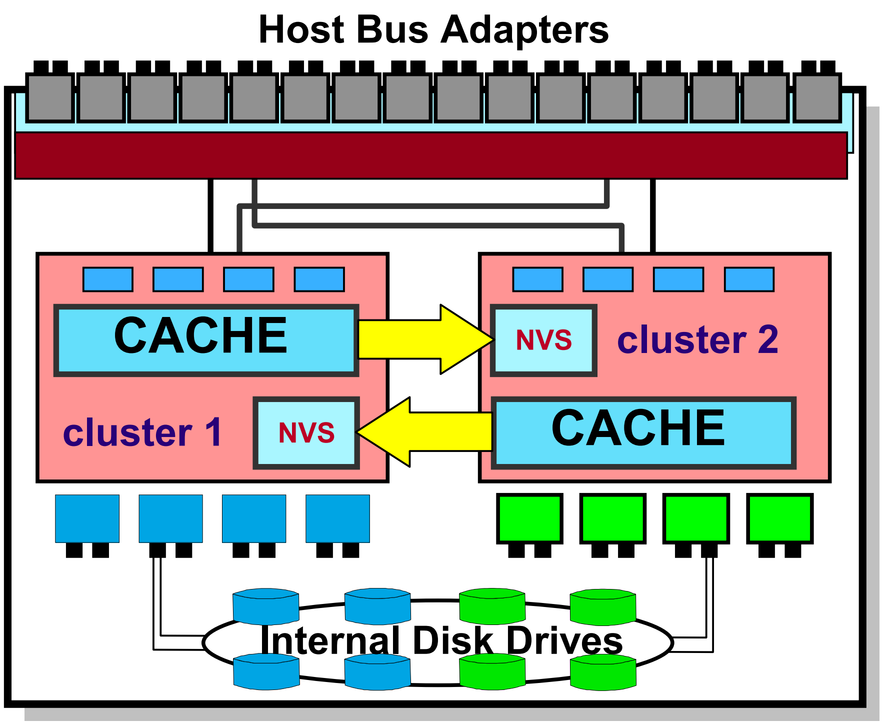


Figure 1: The Enterprise Storage Server's Internal Architecture

Advanced External Connectivity:
The ESS can connect to multiple host systems using multiple Fibre Channel / SCSI connections. These connections can be multipathed for greater redundancy and performance using IBM's Subsystem Device Driver (SDD). The SDD software, installed on the host, is designed to automatically balance the I/O load across all available connections from the host to the ESS. In case of connection loss, such as during a host bus adapter failure, SDD is designed to automatically failover to another path. These features support the host servers' need to have fast, continuous access to data.

Other ESS Features:
These features, along with support for non-disruptive upgrades and repairs, help provide the availability required in an enterprise-class solution. For more information about the ESS and other features of the ESS (such as SAN-capability, ease of management, and universal data consolidation), please visit the IBM Storage web site at http://www.storage.ibm.com.

3B Enterprise Storage Server Advanced Copy Services

To further improve availability and performance, the IBM ESS features Advanced Copy Services. These serverless copy functions are implemented entirely within the storage server, with little or no impact on the host server.

FlashCopy:
FlashCopy is designed to provides an instant T_0 copy of data. A T_0 copy is a point in time copy of the storage volumes when the FlashCopy was invoked. The copy is available almost immediately after invoking the command. The copy process is transparent to the host server, as the ESS storage subsystems manage the copy internally. The point in time copy can now be accessed by a secondary server and be backed up to tape, typically with no significant impact on the host server.

With FlashCopy, multi-*terabyte* databases often can be backed up and restored in minutes, instead of hours or days. Even for a tape subsystem capable of 100GB/hr, a 1 TB database might take ten hours to backup. A restore from tape will take at least the same amount of time. Using the FlashCopy, the same database can be backed up almost instantaneously. Furthermore, the database often can be restored in minutes. Aside from backup and restore, FlashCopy can also provide a copy of "near-live" data clone for business intelligence or data mining, application testing and development.

When FlashCopy is invoked, the ESS creates another copy of the data by building a bitmap that records changed data. When the bitmap is complete (typically in seconds), the copy is logically complete. Both copies can immediately be used (read and write) separately without affecting the other copy. The ESS then begins physically copying the data to the target set of disks (at a rate much faster than a host server could copy). Whenever the host server writes to a block that has not yet been copied, the data is copied first, and then the write continues. In this way, an identical T_0 copy can be made almost instantly with almost no impact to the host server.

Peer to Peer Remote Copy:
PPRC is a remote mirroring technology (similar to RAID-1) that is designed to provide a synchronous copy of production data at inside one ESS subsystem or between two ESS subsystems separated by distance. Because the mirroring is done at the storage subsystem level, there is almost no impact on the host. The remote site can be up to 103 km away, or more with channel extenders. In case of a disaster at the production site, such as fire or earthquake, an up-to-the-second copy can be available at the remote site. The remote copy can be brought online using a standby server, often with minimal business interruption.

4 DB2 UDB Features and Requirements

IBM DB2 UDB v7.1 with fixpak 2 contains many new features that are utilize the new advanced copy service function in ESS, such as, *db2inidb, db2rfpen, db2 set write suspend/resume*. These new feature are discussed below.

4A DB2 UDB Backup and Recovery Overview

Backup Types:
DB2 UDB allows for backing up at the database and table space levels. Snapshot backups are used to create full backups of the database. For very large databases, it is possible to divide the database into tablespaces, you should backup the set of tablespaces together using FlashCopy. This will help eliminate some complexity for future use. It is also possible to back up individual database files and sets of files. But since tables are created on tablespaces, file and set backups should only be used in special circumstances.

Recovery Types:
DB2 UDB provides two methods to recover a damaged database: Version recovery and Rollforward recovery.

Version recovery is the restoration of a previous version of the database, using an image that was created during a backup operation. This allows you to restore a database to a state identical to the one at the time that the backup was made (point-in-time recovery). Every unit of work from the time of the backup to the time of the failure is lost. This recovery model employs circular logging to minimize the amount of space used for transaction logs. This is the model implemented in Recovery Scenario 2.

Using Rollforward recovery, transactions recorded in database logs are applied following the database restore operation. The database logs record all changes made to the database. This method completes the recovery of the database to its state at a particular point in time, or to its state immediately before the failure (that is, to the end of the active logs.) To use the rollforward recovery method, you must have taken a backup of the database, and archived the logs (by enabling either the *logretain* or the *userexit* database configuration parameters, or both.)

During the rollforward process, all committed transactions are reapplied to the database. At the end of the rollforward, a list of open transactions exists. These transactions are still awaiting a commit. For data consistency reasons, open transactions will now be rolled back, to ensure that changes that were not committed are not applied to the database. As long as the transaction logs are not damaged, no data is lost during the restore. This model is implemented in Recovery Scenario 3.

For more information on designing a backup and restore strategy, please see the IBM Redbook *Backing Up DB2 Using Tivoli Storage Manager* available at http://www.redbooks.ibm.com

4B Ensuring Database Consistency in DB2 UDB

To obtain a consistent backup, all write operations must be suspended on the storage volumes before invoking the snapshot backup. DB2 UDB 7.1 (Fixpack 2) has added support for *snapshot* backups through the use of the *set write suspend* and *set write resume* commands for online local/remote backup. This new functionality allows the customer to suspend all writes to the database and log file, and create a consistent backup using FlashCopy.

> DB2 SET WRITE SUSPEND FOR DATABASE: This command suspends the I/O writes and puts the tablespaces into a new *SUSPEND_WRITE* state. Writes to the logs are also suspended by this command. However, read-only transactions are allowed to continue. Any changes to tablespace information needs to wait till the writes are subsequently resumed. Once the *SUSPEND_WRITE* state is in effect for the database, all files and directories within the database are protected from updates.
>
> DB2 SET WRITE RESUME FOR DATABASE: This command resumes I/O writing and removes the *SUSPEND_WRITE* state and makes the tablespaces available for updates.

The use of *set write suspend* and *set write resume* commands, exclusively developed for the use with hardware mirroring facilities, allows a consistent point-in-time snapshot replica of the live database. The suspension of writes to the database and log prevents any partial page writes from occurring until writes are resumed. Read-only transactions are able to continue, provided they do not request any resource held by the suspended I/O process.

4C db2inidb Tool

The *db2inidb* tool is used in conjunction with the *set write suspend* and *set write resume* commands to initialize the mirrored database as a "backup image" that can be used to copy the data on the mirrored disks back to the disks on the original system.. The mirror can be initialized to be used as a clone database, standby database or backup image. This command can only be issued against the split-off mirror, and the split-off mirror must first run db2inidb before it can be used. The db2inidb tool is also used to place the database in a roll forward pending state.

For further details regarding the *db2inidb* tool, please refer to the *DB2 Universal Database and DB2 Connect Online Support*.

4D db2rfpen Utility (DB2 APAR JR16566)

The *db2rfpen* utility *(DB2RFPEN ON <DATABASE_ALIAS>)* puts the offline FlashCopy database into roll forward pending state for roll forward recovery. This utility is not included in install image for V7 and APAR *JR16566* is required to be installed on the system.

5 Snapshot Backup and Recovery Setup and Configuration

To validate these methods of snapshot backup and recovery, we applied a set of test scenarios in the lab. We created an environment that closely resembles a typical customer environment. We used OLTP application to generate a load on the system while we were validating the various backup scenarios.

5A Storage Configuration

Though the ESS is designed to be fully fault-tolerant, this separation provides yet another layer of protection. Should the data files be lost, it would still be possible to recover up to the point of failure using the log files and a backup of the data files. Also, there can be performance benefit in placing these files on separate media. With this in mind, we set up the DB2 UDB database to spread the data and logs across all of the available disks in the ESS for best performance. We also spread the load across both clusters, all host bus adapters, and all internal logical subsystems (LSSs) available within the ESS. This help avoid "hot spots" within the storage subsystem that receive most of the load while other parts of the subsystem are idle.

Furthermore, all of the transaction logs were mirrored on a separate set of disks. The ESS is designed to avoid single points of failure (such as a disk, adapter, CPU, or memory module failure), and most double and triple failures, with its fully redundant, highly available architecture. The possibility exists, however remote, that a combination of hardware failures can occur that would cause data loss (usually during a disaster). Having a mirror of the log is extra protection against such a scenario. Additionally, in some scenarios, we have a mirror at a remote site to protect against disasters, such as earthquake or fire.

5B Hardware Configuration

The test environment contained two Intel-based IBM Netfinity Servers and two Enterprise Storage Servers all connected through an enterprise-class McData ED-5000 Fibre Channel Director. Using Fibre Channel, the Storage Area Network (SAN) provided a fast path to data without adding congestion to the IP network. Each of the Netfinity Servers connected to the SAN via two Fibre Channel host bus adapters (HBAs). Each of the Sharks connected to the SAN using four Fibre Channel adapters, and to each other over 8 ESCON links. In total, there were eight redundant logical paths from each host to each ESS. Each server was configured to use four of the eight available paths. The paths were logically combined through the use of IBM's multipathing software, SDD. They appeared to Windows as a single logical path (i.e., one instance of each volume). The servers were configured as shown in Appendix A
.

5C Network Topology

These servers were connected into a Storage Area Network (SAN). The SAN had the following topology, shown in Figure 2:

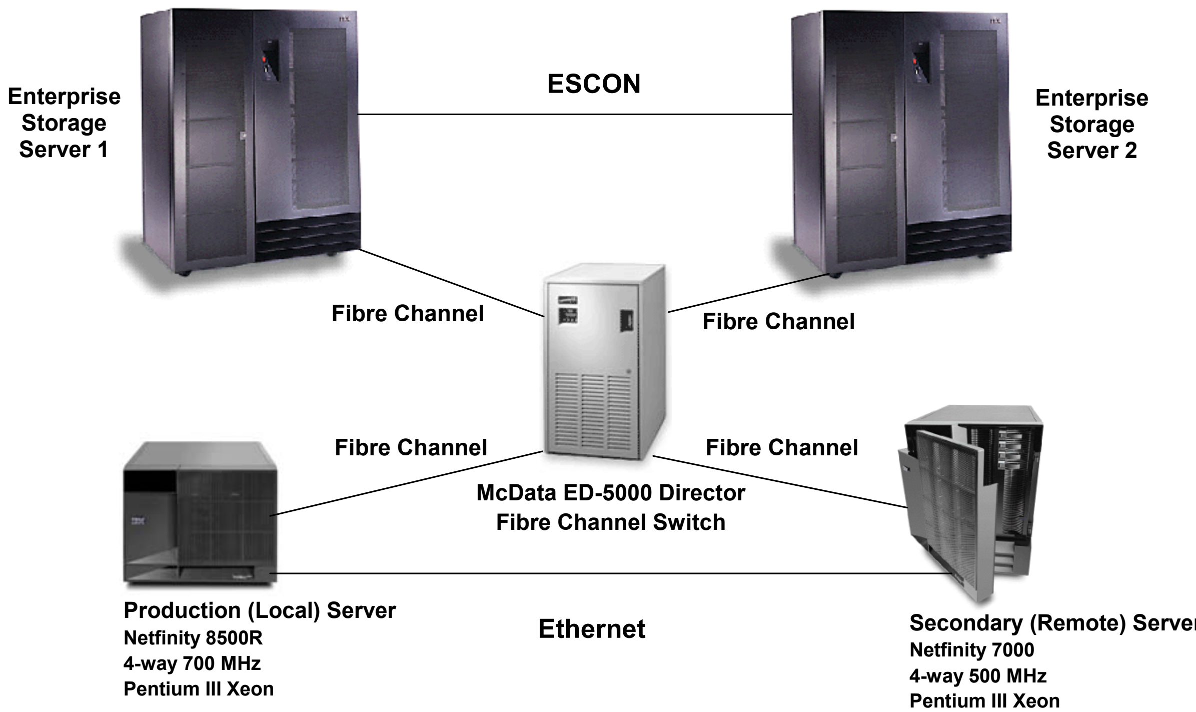


Figure 2: The Storage Area Network Physical Topology

In this configuration, with multiple Fibre connection to McData switch via multiple ports cards and Subsystem Device Driver (SDD) supports fault tolerance. Even though the switch is an enterprise-class, Director, the possibility of disaster and multiple failure still exists. The McData switch is meant to represent a SAN fabric, consisting of several redundant paths and switches from edge to edge.

5D Snapshot Infrastructure Setup

To attain higher availability, we devised a two-ESS and two-server configuration. The production server and ESS 1 are located at the production site, of course. The tertiary server and ESS 2 are located at a remote site. Figure 3 is a logical view of the environment configuration:

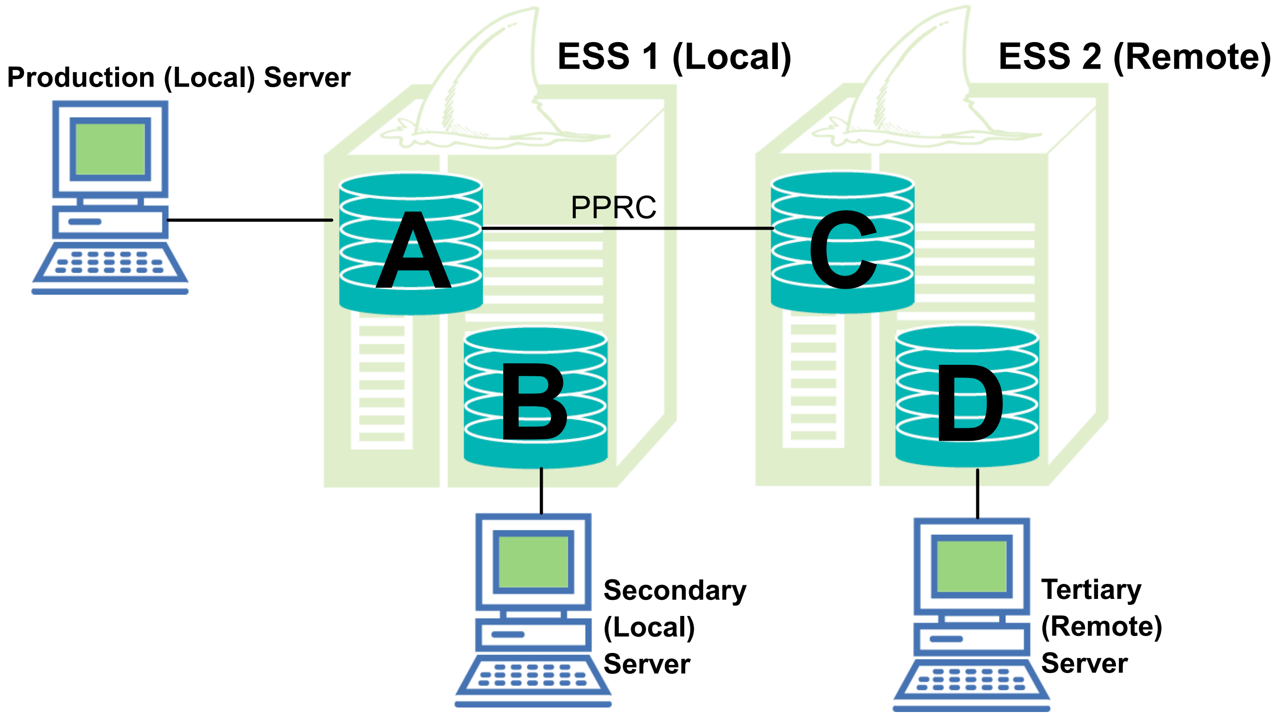


Figure 3: The Storage Area Network Logical Topology

At the production site, the ESS contains two copies of the production database. The first copy (A) is used for production. Periodically, possibly every night, the database is quiesced and a FlashCopy (B) is taken. The FlashCopy (B) is then sent to tape using the resources of a secondary server (not the production server). After that, the FlashCopy (B) is kept in case a rapid restore is required. At the remote site, the second ESS keeps a synchronous copy (C) of the production data. Should the primary server fail, the backup server at the remote site can take over, typically with little or minimal business interruption. It is also possible to take a FlashCopy of that remote copy to produce a fourth copy (D) that can be used for additional protection, data mining/business intelligence, application development, etc.

Ideally, we would recommend having a secondary server at the local site to send the backup to tape, and a tertiary server at the remote site able to take over in case of disaster. In our test environment, our second server served as both the secondary and tertiary.

6 Snapshot Backup Process

This section discusses various situations in which customers might use FlashCopy and PPRC to perform backups. Restore scenarios are discussed in Section 7, "Recovery". Using FlashCopy and PPRC, it is possible to make dozens of copies in different configurations. In the lab, we devised four likely scenarios a customer might use to perform a database backup.

Scenario 1 : Customer requires serverless fast backup and recovery
Solution : Local FlashCopy Backup
As with each of these scenarios, Scenario 1 provides an instant copy of the database to perform a backup. For customers who require the ability to recover as quickly as possible, this scenario allows for a fast FlashCopy restore.

Scenario 2 : Customer requires disaster recovery + Scenario 1 requirements
Solution : Offline Remote FlashCopy Backup
Using PPRC and FlashCopy, this scenario provides an instant copy on a second ESS. This scenario adds further protection and reduces or eliminates any impact on the primary storage subsystem. With the second ESS, the database can typically survive a total site failure.

Scenario 3 : Customer requires 24x7 availability + Scenario 2 requirements
Solution : Online Remote FlashCopy Backup
Customers that require 24x7 availability can use this solution and achieve the disaster protection benefits of Scenario 2 without having to shut down the database.

Scenario 4 : Customer requires 24x7 availability with extra copy + Scenario 3 requirements
Solution : Online Local FlashCopy with additional Remote FlashCopy Backup
As with Scenario 3, this scenario provides 24x7 availability and the protection of a remote copy. Additionally, this scenario provides an extra copy of the database that can be used for functions such as data mining/warehousing, application development/testing, etc.

While these backup scenarios performed as expected under significant load in the lab, we recommend that the backup be performed at times of minimal load. In each case, DB2 UDB had been installed at the secondary server and is standing by. Furthermore, each of our scenarios included error detection mechanisms. We recommend that customers use rigorous error checking in their implementation of these scenarios.

Warning: Do not start DB2 UDB on the secondary system if the secondary database will be used for backup and restore. Instead, use an application such as NTBackup or Tivoli Storage Manager to make a flat file-level backup. Restarting DB2 UDB can modify the logging chain of the database so that it can no longer be used for rollforward recovery. The steps for starting the database on the secondary should only be used for data mining, application development, etc.

Each of these scenarios are described in further detail in subsequent sections.

6A Backup Scenario 1 - Local FlashCopy Backup

This scenario requires one ESS at the production site. While this scenario provides very fast backups and restores, it does not offer any protection against disasters, such as fire or earthquake. Logically, the data is being copied from copy (A) to copy (B) in the following Figure 4:

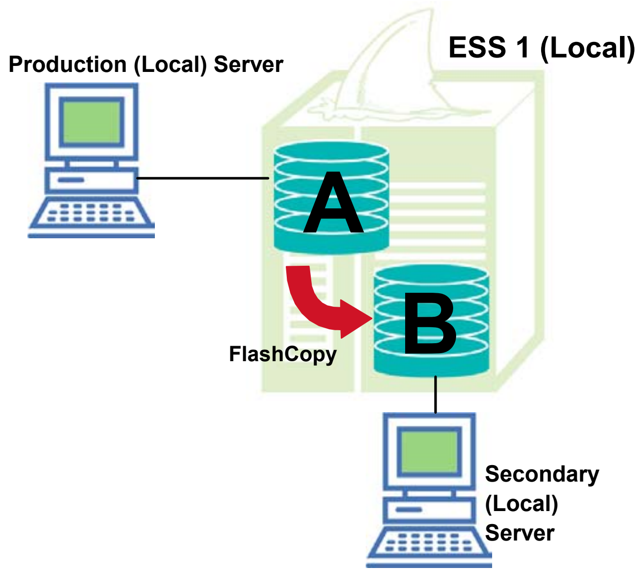


Figure 4: Backup Scenario 1 - Local FlashCopy Backup

Offline Variant:
Customers that require a known, consistent state, and do not require 24x7 availability should use an offline FlashCopy backup. The steps for performing the backup are as follows:

1. Stop DB2 UDB at production server
2. Flush file system buffers on production server
3. Perform FlashCopy Backup
4. Restart production DB2 UDB - system is back in production
5. Resync secondary filesystem (i.e. make secondary server aware of new disks)
6. Start DB2 UDB database at secondary server
7. Perform database integrity checks at secondary server

Online Variant:
DB2 UDB customers that require 24x7 availability can use a backup Scenario 1 variant to perform an online backup instead of an offline backup. While the backup is not necessarily in a known state, it is still consistent and can be used for recovery. In general, the online backup is done by freezing I/O on the production server, performing the FlashCopy, and thawing I/O. This is accomplished by using the *set write suspend* and *set write resume* commands described in Section 4B.

6B Backup Scenario 2 - Remote FlashCopy Backup

To add protection from disasters (such as fire, earthquake, etc.), the three remaining scenarios require one ESS at the production site and one ESS at the remote site for disaster recovery. For both Scenarios 2 & 3, the FlashCopy is done from the remote synchronous mirror (C) to the remote copy (D), as shown in Figure 5:

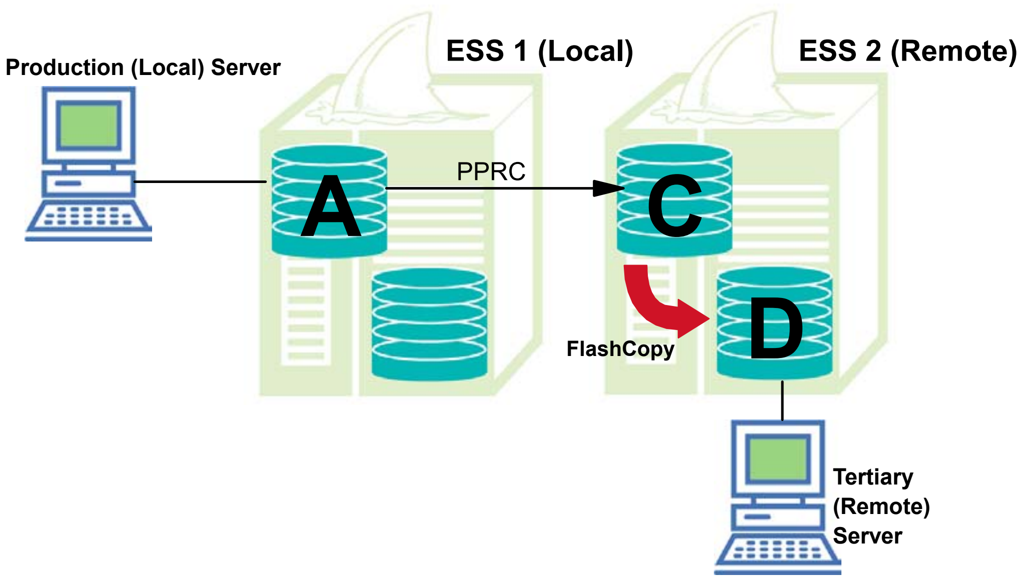


Figure 5: Backup Scenarios 2 & 3 - Remote FlashCopy Backup

This setup provides the rapid backups of Scenario 1 and adds disaster recovery protection. Furthermore, the FlashCopy (D) of the database at the remote ESS (ESS 2) can be used to send the backup to tape, eliminating any overhead on the production ESS and production server. At the beginning of the scenario, the database volumes should already be in a PPRC relationship. The following steps are involved in the offline remote FlashCopy backup:

1. Stop DB2 UDB at production server
2. Flush file system buffers on production server and remote server
3. Perform FlashCopy Backup at remote ESS 2
4. Restart production DB2 UDB - server is back in production
5. Resync remote filesystem (i.e. make remote server aware of new disks)
6. Start DB2 UDB database at remote server
7. Perform database integrity checks at remote server

6C Backup Scenario 3 - Online Remote FlashCopy Backup

This scenario is an online version of Scenario 2. The setup is as described in Figure 6 above. This scenario would be used by customers requiring 24x7 availability. Instead of shutting down DB2 UDB, we suspend writes on DB2 UDB using the *set write suspend* and *set write resume* commands introduced in DB2 UDB 7.1 (Fixpack 2). The writes are frozen only for the time it takes to invoke FlashCopy on the remote ESS. The steps involved in performing the online remote FlashCopy backup are as follows:

1. Flush file system buffers on production and remote server
2. Freeze I/O on production server
3. Perform FlashCopy Backup at remote ESS 2
4. Resume I/O on production server
5. Resync remote filesystem (i.e. make remote server aware of new disks)
6. Start DB2 UDB database at remote server
7. Perform database integrity checks at remote server

Faster Backup Variant for Very Large Database Customers:
For large, multi-terabyte databases, FlashCopy may take more than one minute to logically complete. Some applications may not tolerate writes being frozen for that length of time. Using the suspend feature of PPRC can be greatly reduces this time. In this situation, the I/O is only frozen for the time it takes to suspend the PPRC link (typically a few seconds). Then, the remote mirror is in a consistent state, and I/O resumes on the production server. Now, the FlashCopy backup can take minutes without impacting the production server. As soon as the FlashCopy is logically complete, the PPRC link is resynched (the data that occurred while PPRC was suspended is written to the remote mirror), and the mirrors are in sync again.

Note that, in this variant of Scenario 3, the mirrors are out of sync for the few minutes it takes to perform the FlashCopy. Should a disaster strike at the production site, data may be lost. Therefore, this variant is not appropriate for all customers.

Here are the steps in performing this type of backup:
1. Flush file system buffers on production and remote server
2. Freeze I/O on production server
3. Suspend PPRC to remote ESS 2
4. Resume I/O on production server
5. Perform FlashCopy at remote ESS 2
6. Resync PPRC to remote ESS 2
7. Resync remote filesystem (i.e. make remote server aware of new disks)
8. Start DB2 UDB database at remote server
9. Perform database integrity checks at remote server

6D Backup Scenario 4 - Online Local FlashCopy with Extra Remote Copy

In this comprehensive solution, which uses the technologies from the three previous scenarios, there are four copies of the database at any time. Copy (A) is the production copy; (C) is its synchronous mirror for disaster recovery. Copy (B) is a local FlashCopy that can be used for extremely rapid restores, while (D) is a copy that can be used to backup to tape, data mining, and other functions. This scenario would be used by customers requiring 24x7 availability, rapid restores, and disaster recovery capability. As with the previous two scenarios, Scenario 4 requires one ESS at the production site and one ESS at the remote site for disaster recovery. The logical view of this scenario is described in figure 6:

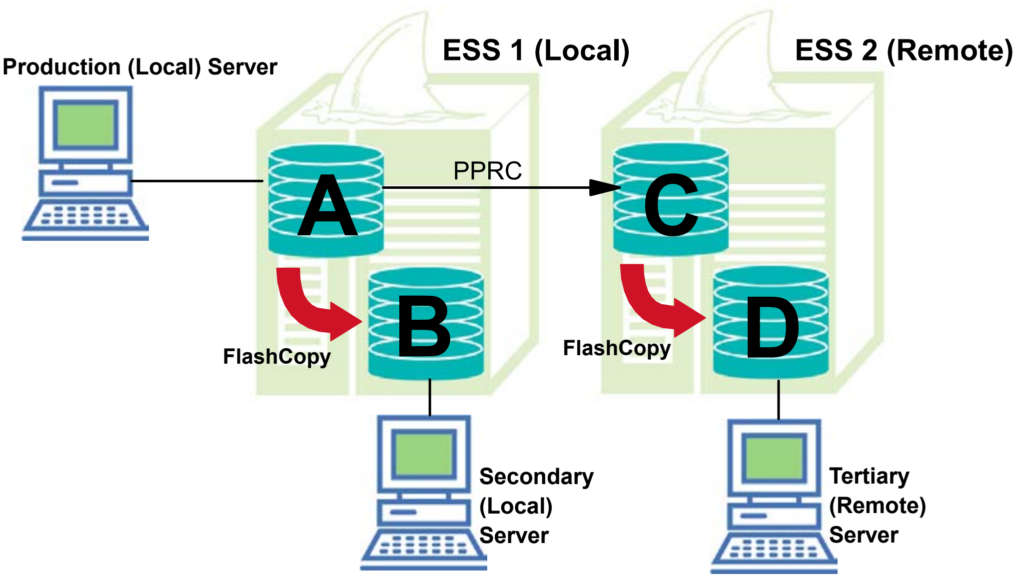


Figure 6: Backup Scenario 4 - Online Local FlashCopy with Extra Remote Copy

As in Scenario 3, we suspend the I/O on DB2 UDB using the *set write suspend* and *set write resume* commands. The writes are suspended only for the duration of the FlashCopy tasks. The FlashCopy tasks on the production and remote ESSs are usually executed at the same time; these copies can also be individually scheduled at different intervals to allow the customer to recover to different points in time. This allows for the maximum level of flexibility for the customer.

1. Flush file system buffers on production and remote server
2. Freeze I/O on production server
3. Perform FlashCopy Backup at production ESS 1
4. Perform FlashCopy Backup at remote ESS 2
5. Resume I/O on production server
6. Resync remote filesystem (i.e. make remote server aware of new disks)
7. Start DB2 UDB database at remote server
8. Perform database integrity checks at remote server

7 Recovery

In addition to the four backup scenarios, we devised three recovery scenarios that can be used to recover from most errors/disasters.

Scenario 1 : Customer requires disaster recovery
Solution : Reverse PPRC Recovery

In the situation where the production database is corrupted or the production ESS is lost (due to natural disasters, etc.) we can use the backup taken in Backup Scenario 3 to restore from the remote copy on ESS 2. While the synchronous nature of PPRC protects against disasters, it does not protect against logical errors; the errors are propagated to the PPRC Target on the remote ESS as well. This scenario would protect against logical errors. It can also help protect against rolling disasters. The FlashCopy taken at the remote site is a known consistent backup, and can be used to restore the production database much faster than possible from tape. Optionally, if the log files are not damaged and contain no logical errors, the user can use them to roll forward to the point of failure.

Scenario 2 : Customer requires Point-In-Time recovery
Solution : FlashCopy Simple Recovery

In the scenario where the production database is corrupted or contains logical errors, the customer would need to restore the database to a previous point-in-time from the FlashCopy backup. The FlashCopy backup taken in Scenario 1 contains a known, consistent state, and is used to restore the database.

Scenario 3 : Customer requires recovery to Point-Of-Failure
Solution : FlashCopy Full Recovery

This scenario would be used by customers who can not afford to lose any data in case of a system failure. As long as the transaction logs are not damaged, this scenario can typically be used to recover the database up to the point of failure. This scenario, combined with frequent log backups, offers a very short restore window usually with no data loss.

In order to recover the database from a FlashCopy backup that was created while writes were suspended, we need to use the db2inidb command to initialize the mirrored disks during the restore process.

In each recovery scenario, it is assumed that DB2 UDB is already installed and all pertinent patches applied. Each of these scenarios are described in further detail in subsequent sections.

7A Recovery Scenario 1 - PPRC Recovery

This Scenario requires two ESSs and assumes the database setup of Backup Scenario 3. Backup Scenario 3 makes periodic (perhaps daily) FlashCopy backups of the PPRC target on the remote ESS, and this backup is used to restore the production database on the primary ESS. Figure 7 shows the logical view of the scenario:

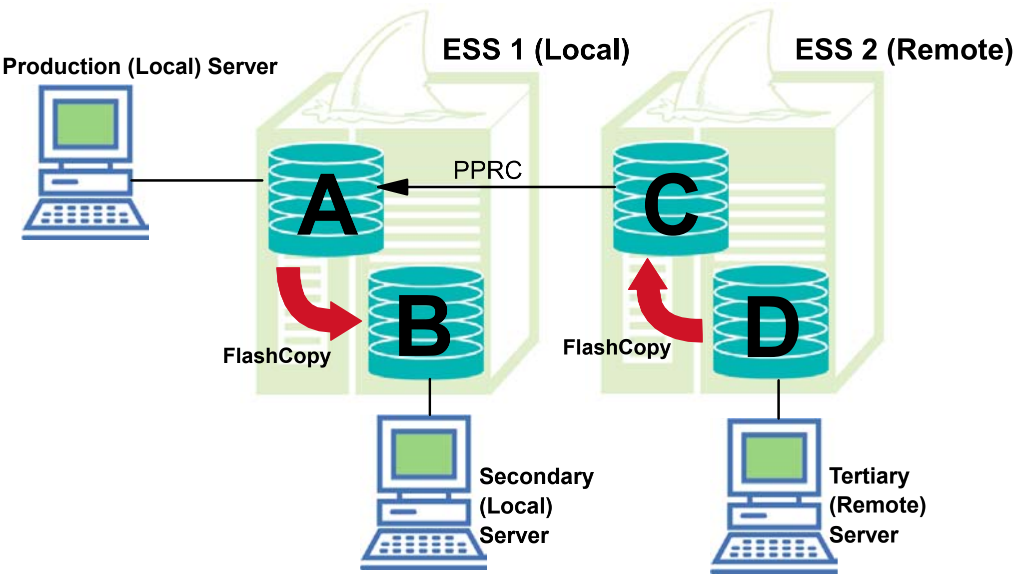


Figure 7: Recovery Scenario 1 - PPRC Recovery

This scenario assumes that there is no local copy available. The steps involved are:

1. Terminate PPRC relationship from production ESS 1 to remote ESS 2
2. Stop DB2 UDB at production server
3. Flush file system buffers on production system and remote system
4. Perform reverse FlashCopy at remote ESS 2 from copy (D) to copy (C)
5. Establish PPRC relationship from remote ESS 2 copy (C) to production ESS 1 copy (A)
6. Wait for PPRC to complete (check status with rsQueryComplete)
7. Terminate PPRC relationship from remote ESS 2 copy (C) to production ESS 2 copy (A)
8. Resync production filesystem (i.e. make production system aware of new disks)
9. Start DB2 UDB at production server
11. Perform a safety FlashCopy at production ESS 1 from copy (A) to copy (B) with suspend write
12. Re-establish PPRC relationship from production ESS 1 copy (A) to remote ESS 2 copy (C)

Once the database is restored successfully (after step 11), we take a safety FlashCopy (B) to be used for rapid restores in case the production database becomes corrupt again. Then, at the conclusion of this scenario, we restore the ESS configurations to the production state, with the production database (A) being mirrored on the remote site (C) through PPRC.

7B Recovery Scenario 2 - FlashCopy Simple Recovery

While Recovery Scenario 1 provides protection against disasters, its method of restore can take several hours for very large databases. Though PPRC is usually faster than multiple tape drives, FlashCopy can provide much faster restores. With this scenario, the customer is able to restore to the time of the last backup within a very short period of time. The logical setup for this scenario as well as Recovery Scenario 3 (Full Roll Forward recovery) is as shown in Figure 8:

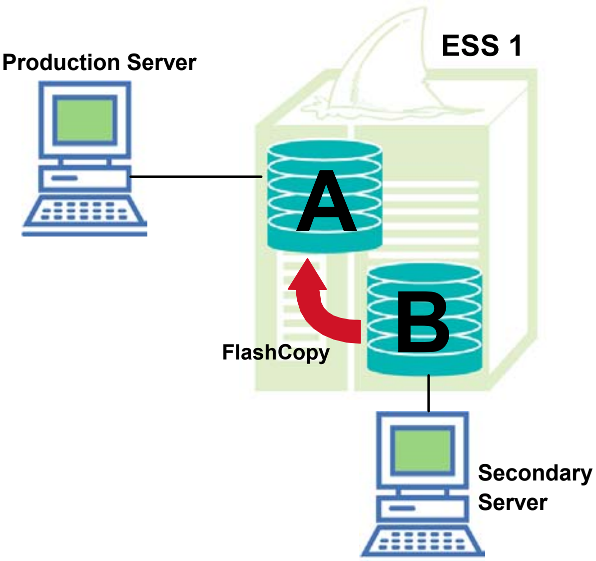


Figure 8: Recovery Scenarios 2 & 3 - Local FlashCopy Restore

This scenario is useful for recovering from logical errors. If a user accidentally drops a table, for example, a FlashCopy can return the database to a consistent, known prior state (the state at the time of the last backup). Any changes made after that last backup need to be redone (except the ones that caused logical errors, of course).

1. Stop DB2 UDB at production server
2. Flush file system buffers on production system and remote system
3. Perform a FlashCopy restore (of data files and logs)
4. Resync production filesystem (i.e. make production system aware of new disks)
5. Start DB2 UDB at production server
6. Initialize the snapshot and remove the suspend write state
   db2inidb <database_alias> as snapshot

Offline Backup Variant:

If the backup was created offline (as in Scenario 1 or 2), the steps for the restore vary slightly as the db2inidb command is not used to initialize the snapshot.

1. Stop production DB2 UDB
2. Flush file system buffers on production system and remote system

3. Perform a FlashCopy restore (of data files and logs)
4. Resync production filesystem (i.e. make production system aware of new disks)
5. Start Production DB2 UDB

7C Recovery Scenario 3 - FlashCopy Full (Roll Forward) Recovery

While Recovery Scenario 2's ability to recover to the time of the last backup is useful, many customers require the ability to recover to the moment of failure (or any arbitrary point in time). Recovery Scenario 3 provides exactly this capability, while maintaining the extremely fast restores. As long as no transaction logs are damaged, no data should be lost.

It is possible to replace the data files with the files from the backup. Then, DB2 UDB will recover using the old data files and the new log files using the ROLLFORWARD DATABASE command. This command allows the user the flexibility of rolling-forward to the end of the log file, to a certain time, or upto a certain transaction. Logically, the FlashCopy is from copy (B) to copy (A), shown in Figure 8 above.

The steps in detail are:

1. Stop production DB2 UDB
2. Flush file system buffers on production system and remote system
3. Perform a FlashCopy restore (of only the data files; not the logs)
4. Resync production filesystem (i.e. make production system aware of new disks)
5. Start DB2 UDB at production server
6. Place database in rollforward pending state and remove the suspend write state
   db2inidb <database_alias> as mirror
7. Rollforward the database logs

Offline Backup Variant:

If the backup was created offline (as in Backup Scenario 1 or 2), the steps for the restore vary slightly. In such a scenario, the database has to be explicitly placed in rollforward pending state using the db2rfpen utility, since the db2inidb command can only be used for initializing on-line backups.

The steps for the restore process are:

1. Stop production DB2 UDB
2. Flush file system buffers on production system and remote system
3. Perform a FlashCopy restore (of only the data files; not the logs)
4. Resync production filesystem (i.e. make production system aware of new disks)
5. Start DB2 UDB at production server
6. Place database in rollforward pending state
   db2rfpen on <database_alias> [1]
7. Rollforward the database logs

For more information about restoring transaction logs, please consult DB2 UDB Online Support.

[1] DB2 UDB APAR number of this utility is JR16566.

8 Process Automation and Solution Integration

In the lab, we developed various applications and utilities to fully integrate DB2 UDB and ESS features to create a complete backup and recovery solution. We also created scripts to fully automate the snapshot backup and recovery scenarios described in previous sections. The scripts execute a customized set of snapshot routines set up for a particular customer environment. The customer can use these scripts, applications and utilities as a one-stop backup and restore solution for even the most complex storage configurations and customer requirements. The solutions developed are focused on providing a lightning quick, seamless backup with little or no impact on the production environment, while also protecting against disasters using the remote mirroring technology of the ESS.

These solutions are designed to work in many different environments, and are compatible with various customer-preferred enterprise solutions management consoles like TME, OpenView or BMC. In establishing this snapshot backup and recovery solution for DB2 UDB and ESS, IBM has created end-to-end procedures that are intended to enable this solution to seamlessly integrate into customer environments.

Appendix A: Hardware and Software Configuration

Production Server:
IBM Netfinity 8500R 8RY
4-way 700 MHz Pentium III Xeon processors
4 GB memory
2 9.1 GB 7200 RPM internal SCSI disk drive
1 Qlogic 2200 Fibre Channel Adapter
1 IBM Netfinity 10/100 Ethernet adapter
Windows NT Enterprise Edition, Service Pack 6a
DB2 UDB 7.1 Enterprise Edition, Fix Pack 2
Windows 2000 Advanced Server, Service Pack 2

Secondary Server:
IBM Netfinity 7000 M10
4-way 500 MHz Pentium III processors
1 GB memory
2 9.1 GB 7200 RPM internal SCSI disk drive
1 Qlogic 2200 Fibre Channel Adapter
1 IBM Netfinity 10/100 Ethernet adapter
Windows NT Enterprise Edition, Service Pack 6a
DB2 UDB 7.1 Enterprise Edition, Fix Pack 2
Windows 2000 Advanced Server, Service Pack 2

Production Storage Server:
IBM Enterprise Storage Server F20
16 GB cache
64 36.4 GB 10000rpm HDDs
12 Fibre Channel Adapters
4 Dual Port ESCON Adapters

Backup Storage Server:
IBM Enterprise Storage Server F20
16 GB cache
64 36.4 GB 10000rpm HDDs
12 Fibre Channel Adapters
4 Dual Port ESCON Adapters

Fibre Channel Switch:
McData ED-5000 Fibre Channel Director
32 Ports

Appendix B: Operating System Considerations

Snapshot disk-copying technologies have only recently gained popularity in the marketplace. Most operating systems predate these technologies, and as such, these operating systems were not designed to accommodate them. Therefore, there are some aspects to consider when using FlashCopy and PPRC to support a smooth backup and restore. These aspects are related to volume management in Windows only, and not to DB2 UDB. For more information on using FlashCopy and PPRC in a Windows environment, please consult the IBM Redbook, *Implementing ESS Copy Services on UNIX and Windows NT/2000,* available free of charge at http://www.redbooks.ibm.com. A few of the operational considerations affecting Windows are briefly outlined below:

Windows NT considerations

Both PPRC and FlashCopy are supported when using simple disks and fault-tolerant disks (such as volume sets). These four tips should help in preparing Windows NT to use Copy Services:

- For fault-tolerant disks, essential configuration information is stored in the Windows Registry (not on the actual disk). Therefore, when initially defining volume sets, (since FlashCopy and PPRC does not copy the data stored in the Windows Registry) a utility from IBM called FCVolSet automatically performs this registry copy. Alternatively, a special procedure can be used, as outlined below:

    1. Define and format on primary system
    2. Document the Drive Signature, Partition Number, and Order Selected
    3. Perform FlashCopy
    4. Run FTEdit (available on the NT resource kit) on secondary and define volume set
    5. Repeat whenever Volume Set is changed

- After Service Pack 6, it is possible to have the FlashCopy source and target volumes accessible by the same server. In this case, use Disk Administrator to write a different disk signature on the target volume and assign a drive letter. Prior to Service Pack 6 the FlashCopy source and target volumes must be attached to different servers.

- To avoid rebooting after a FlashCopy, define an identical set of disks on the target machine. Then, use Disk Administrator to unassign the drive letter, perform the FlashCopy, and reassign the drive letter, and the drive is immediately available.

- To flush file system buffers in Windows NT and 2000, use Disk Administrator to unmount (unassign) the drive letter. In the lab, we developed a program to perform the file system flush without unassigning the drive letter.

Windows 2000 considerations

Windows 2000 supports two types of disks: basic disks and dynamic disks. Basic disks are the same as Windows NT disks with the same restrictions. For dynamic disks, Windows 2000 incorporates a volume manager called the Logical Disk Manager (LDM). The LDM can create five types of dynamic volumes: simple, spanned, mirrored, striped, and RAID-5.

On Windows NT the information relating to the disks is stored in the Windows NT registry. With Windows 2000, this information is stored on the disk drive itself in a partition called the LDM database, which is kept on the last few tracks of the disk. Each volume has its own 128 Bit Globally Unique

Identifier (GUID). As the LDM database is stored on the physical drive itself, with Windows 2000 it is possible to move disk drives between different computers.

Having the drive information stored on the disk itself imposes some limitations when using Copy Services functionality on a Windows 2000 system with dynamic disks:

- The source and target volumes must be of the same physical size. Normally the target volume can be bigger than the source volume. With Windows 2000 this is not the case, for two reasons:

    1. The LDM database holds information relating to the size of the volume. As this is copied from the source to the target, if the target volume is a different size from the source, then the database information will be incorrect, and the host system will return an exception.
    2. The LDM database is stored at the end of the volume. The copy process is a track-by-track copy, and unless the target is an identical size to the source the LDM database will not be at the end of the target volume.

- It is not possible to have the source and target FlashCopy Volume on the same Windows 2000 system. Each dynamic volume has its own 128 Bit Globally Unique Identifier (GUID). As its name implies, the GUID is unique to one system. When performing a FlashCopy, the GUID is copied as well, so this means that if you try to mount the source and target volume on the same host system, you would have two volumes with exactly the same GUID. This is not allowed, and you will not be able to mount the target volume.

- Each disk contains information about every other dynamic disk on the system. Therefore, after a FlashCopy or PPRC, the information on the other disks may be inaccurate. Windows only checks the information in the LDM database on bootup. While the disks can be used without rebooting, as in Windows NT, the drives will continue to work as expected only until the first reboot. The first reboot after a FlashCopy will require the use of Disk Management to “import” the volume. Therefore, if possible, reboot the system and perform an import before using the disk.

References

1. "Storage Management for SAP and DB2 UDB: Split Mirror Backup / Recovery with IBM's Enterprise Storage Server (ESS)" -- White Paper http://www.storage.ibm.com/hardsoft/products/sap/smdb2.pdf

2. *IBM ESS and IBM DB2 UDB Working Together* -- IBM Redbook, Document Number SG24-6262-00, September 2001, http://www.redbooks.ibm.com

3. *Implementing ESS Copy Services on UNIX and Windows NT/2000* -- IBM Redbook, Document Number SG24-5757-00, March 2001, http://www.redbooks.ibm.com

4. *IBM Enterprise Storage Server* -- IBM Redbook Document Number SG24-5465-00, July 1999, http://www.redbooks.ibm.com

5. *IBM Enterprise Storage Server Performance Monitoring and Tuning Guide* -- IBM Redbook, Document Number SG24-5656-00, March 2000, http://www.redbooks.ibm.com

6. *Backing Up DB2 Using Tivoli Storage Manager* -- IBM Redbook, Document Number SG24-6247-00, June 2001, http://www.redbooks.ibm.com

7. "Data Recovery and High Availability Guide and Reference" -- DB2 Universal Database and DB2 Connect Online Support
ftp://ftp.software.ibm.com/ps/products/db2/info/vr7/pdf/letter/db2hae71.pdf