Better backups through replication

HP OpenView Storage Mirroring

invent

When people consider replication software, their assumption is "high availability" or "disaster recovery." While traditional business continuance is an important focus, the routine and automated protection of data will always include a tape aspect—for long-term archival of data. The industry is learning that the two technologies are not mutually exclusive and, in fact, are complementary.

Several questions regarding the combination of tape and replication technologies are answered in this paper.

- What is a typical backup/restore solution?
- What are potential issues having tape alone?
- What does replication add to an existing tape solution?
- What about key operating system information such as the system state?
- What about application-specific data, configurations, and large "scrubs"?
- What about snapshots such as Volume Shadow Copy Service (VSS) from Microsoft® Windows® 2003?
- How does restore work?

The purpose of this document is to educate a "Technical Decision Maker" on how replication can add value and reliability to one's existing data protection strategy.

Strategic planning assumption
By 2003, 75% of large enterprises will combine data replication and tape technology for rapid application recovery (0.7 probability).

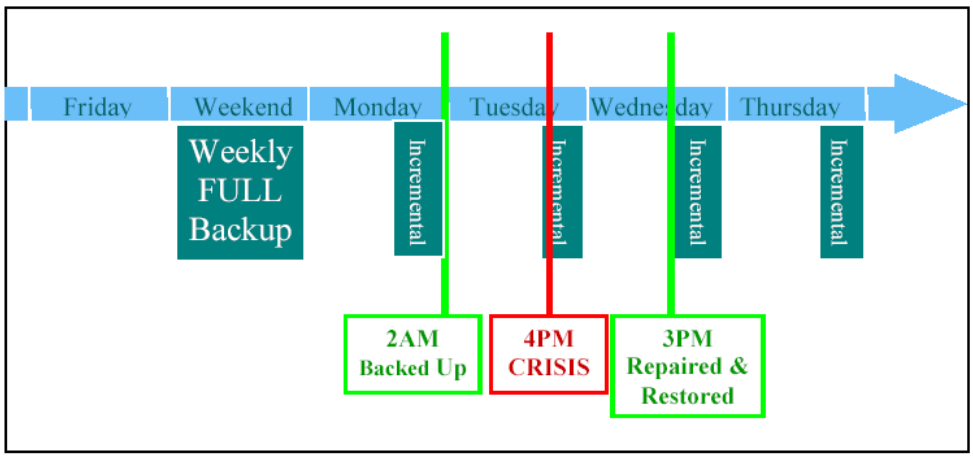
Bottom line: Organizations that are running 24 x 7 operations and are confronting shrinking backup windows, or that foresee an inability to meet service-level agreements, should plan and budget for deploying data replication technologies.[1]

[1] Gartner Designing to Restore From Disk: Backup Futures

What is a typical tape backup/restore solution?

A primary flaw with tape backup is that it usually occurs once per day. This means that one must always measure data loss windows and recovery times in "days."

Figure 1. Typical data loss and productivity loss with tape

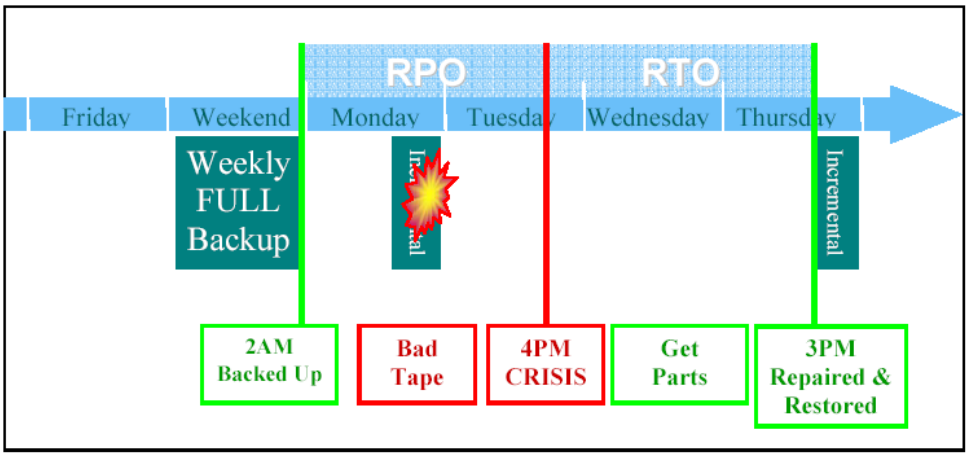


As an example, consider a primary server failed at 4:00 PM on Tuesday afternoon. If all parts were on hand, the server could be rebuilt on Tuesday evening and the data restored. When users return on Wednesday morning, their data will be as it was on Monday night's backup. Tuesday's data was lost.

But for most of us, an extra server is not sitting on the shelf. Therefore, the components could be expedited on Wednesday morning and the restoration began in the afternoon; users would begin working by Thursday morning—with data as it existed on Monday night's backup. Tuesday's data would be lost, and Wednesday would have had limited productivity at best.

If off-site tape couriers are used, Monday's tape might have been off site, which adds an additional day before the restore could have begun. To imagine the story even worse, consider if the environment does a full backup only on weekends (and incrementals during the week). If something went wrong with that weekend's backup, the data loss and restore effort both would reach back through the entire previous week.

Figure 2. Realistic view of potential data loss and productivity loss with tape



The only way to reduce the window of exposure for data loss (RPO) and lost productivity (RTO) is to use a backup of data occurring more than once or multiple times per day—replication.

What are potential issues having tape alone?

When considering conventional backups, several caveats exist collectively forcing most companies into the dreaded "backup window." Backup windows are in place because of a few problems with most tape approaches:

- High CPU cycles and I/O during backups—While a server is being backed up, it tends to use excessively more CPU and I/O, which results in overall slower performance. This is a common reason why backups are completed during off hours.
- High network usage during backups—A majority of environments no longer have tape devices on every server and deploy a centralized or remote tape solution. Unfortunately, this requires that all files go across a network (from the production servers to the backup platform). In worst cases, these files traverse the production network segment, slowing down user traffic. In better cases, there is a "backbone segment" for the backup traffic, but the disk and network I/O on the production servers will still suffer. To reduce the impact, most backups are still completed after hours.
- Open files—By far, the most common reason for performing backups during nonproduction time is open files—those files held locked by applications (for example, SQL or Exchange), as well as by users applications (for example, Microsoft Office documents and spreadsheets). To combat this, one can deploy application-specific agents or open-file handlers. However, application agents are expensive, specific to a version of a particular application, and only available for a handful of common back-office applications. In addition, all agents and handlers still require the production CPU to do extra work to circumvent the file-locks, which inherently takes cycles away from production tasks.

So collectively, to eliminate open files, reduce CPU and I/O impact, and minimize network traffic many disaster tolerant solutions have been forced to back up data only between 2:00–6:00 AM (or other non-production hours).

Replication resolves these issues and eliminates the backup window by providing a copy of the data to be backed up:

- The copy of the data is not locked, so there are no open file issues.
- CPU and I/O usage issues are irrelevant on the redundant server because users are running from the production platform and are therefore unaffected.
- Because replication provides a real-time copy (by continually sending small updates), the backup platform already has access to the files without reaching across the network to the production servers.

What does replication add to an existing tape solution?

HP OpenView Storage Mirroring replicates data at byte level, meaning that if an application writes a string of 12 bytes within a file, then the actual 12 bytes plus header is transmitted across the existing IP infrastructure. At the target location, the 12-byte string is applied to the same location within the second copy of the file. This provides for multiple copies of one's data for fault tolerance.

Because the replicated files on the target servers are loosely coupled to the production files, the target data is not locked or in use, even when the production files are. This enables customers to utilize any backup software/hardware solution on the target data volumes, without open-file or application-oriented agents. In addition, backup jobs can run during the day, even when the primary files on production servers are in use. This removes consideration for open files so HP customers can back up the copied data 24 x 7, eliminating the backup window.

Disk to Disk to Tape (D2D2T)

To further enhance data protection strategies, a growing number of enterprises are bringing data from their remote sites to their corporate sites, before performing tape backups.

Instead of relying on weekly/daily rotations of tape at remote offices where human error can affect the reliability of tape handling, tape backups can occur at the data centers. This is enabled by continuously replicating the data changes from the remote sites to the data center with HP OpenView Storage Mirroring.

For remote sites without a local admin staff, file data can be replicated to an upstream hub site using Storage Mirroring and backed up as part of the normal hub site backup process. By centralizing file backup, HP clients can increase the reliability of tapes, reduce manpower costs, and eliminate hardware and backup software in the remote sites.

This architecture can also be complemented by the use of snapshot technologies like those by PSM (by Columbia Data Systems) or Microsoft VSS.

The broader view on business continuity

To achieve larger business continuity goals, HP OpenView Storage Mirroring is used to replicate data between facilities.

- Data is replicated between hub data centers to allow for disaster recovery between geographies.
- Data is replicated from remote branches to the nearest data center (and optionally replicated again to the alternate data center) for data protection and resilience of the branches.

Storage Mirroring also provides failover capability, whereby a target platform can be configured to stand in or "fail over" for a production server, offering the name, IP, and file shares (as needed). This allows remote users to utilize the data center copy of the data, if the branch server were to fail.

What must change for me to implement?

Implementing HP OpenView Storage Mirroring does not require changes to the existing Active Directory user environment. Permissions to production files will also be applied to the replicated copies. To protect the replication environment, Storage Mirroring creates a local machine group on each machine that runs the software. By adding an AD-domain group to this local machine group, authentication for management will be automatically bestowed.

Storage Mirroring does use the existing infrastructure, requiring merely native IP connectivity between sources and targets. Specifically, two defined TCP/UDP ports are used for all Storage Mirroring traffic, thus allowing network management and monitoring, as well as "quality of service" or packet-prioritization to be optionally used.

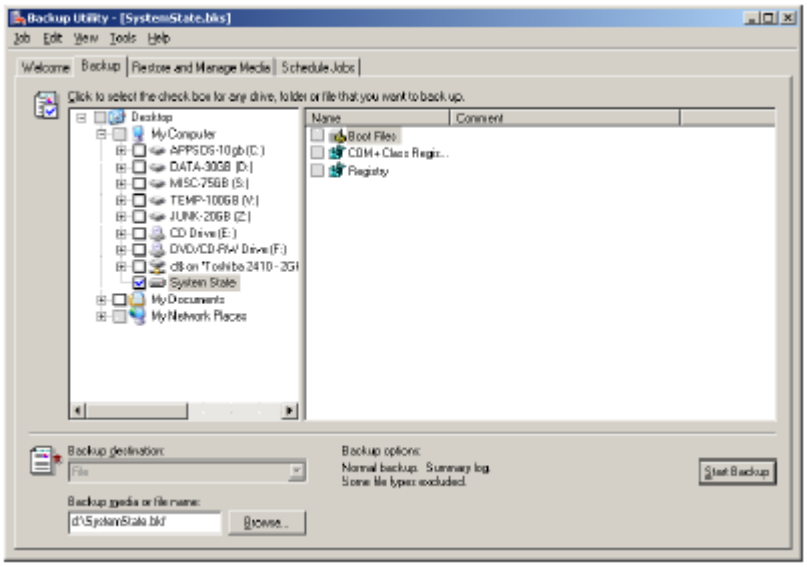
In addition, Storage Mirroring does not rely on any particular backup technology, although using one that is supported by Microsoft is suggested. Without changing one's tape backup software or hardware, the backup process can be enhanced by pointing the backup solution at a Storage Mirroring target server, instead of a production server.

What about the operating system–specific information?

If the primary goal is a predictable and reliable restoration of the production environment, one simple step is to automatically back up the system state of each production server. According to Microsoft, the system state "comprises the registry, COM+ Class Registration database, files under Windows File Protection, and system boot files" and can also include information from Active Directory, DNS, IIS, and the Cluster Service. In short, one can completely restore a failed server by doing a clean operating system installation followed by restoring the system state.

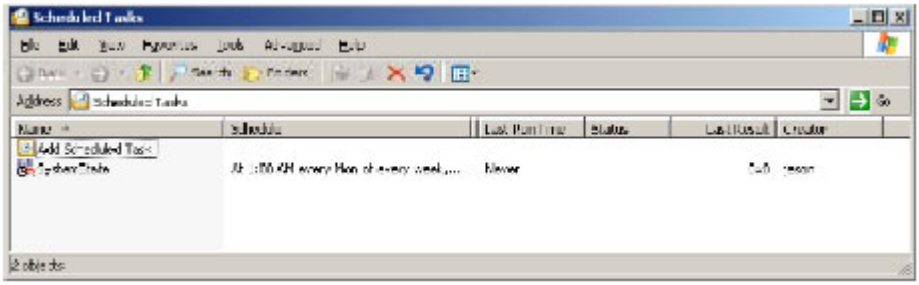
Thankfully, Microsoft server operating systems provide a backup utility that can be run while users are active on the machine. The default location for this utility is c:\windows\system32\ntbackup.exe and it can be initiated from selecting Start>Programs>Accessories>SystemTools>Backup. It can also be executed from the command line or scheduler.

Figure 3.



First-time users of the backup utility should consider using the GUI to configure a backup of the system state (plus additional key files, such as INIs within program directories). During configuration of the backup job, users can schedule the job to run routinely with a best practice being at least weekly.

Figure 4.



The results will be an individual file (*.BKF) instead of using actual tape or other media. By selecting the directory (where the backup file will reside) as part of the Storage Mirroring replication set, this BKF file will also be replicated to the target server. During any recovery, the BKF can be used to restore the system state including registry and other in-use files.

As part of the recovery process, one should configure the new production server with the same operating system. Then, if the various system drive directories (for example, Windows and Program Files) have been replicated, those can be copied to the new server. If you are using a third-party backup package, one might consider backing up the remote source server's operating system volume (including system state) monthly. This will cause some network congestion, but once per month is typically tolerable. To restore, one would restore from tape, and then still replay the latest system state backup that was replicated by Storage Mirroring to the target server.

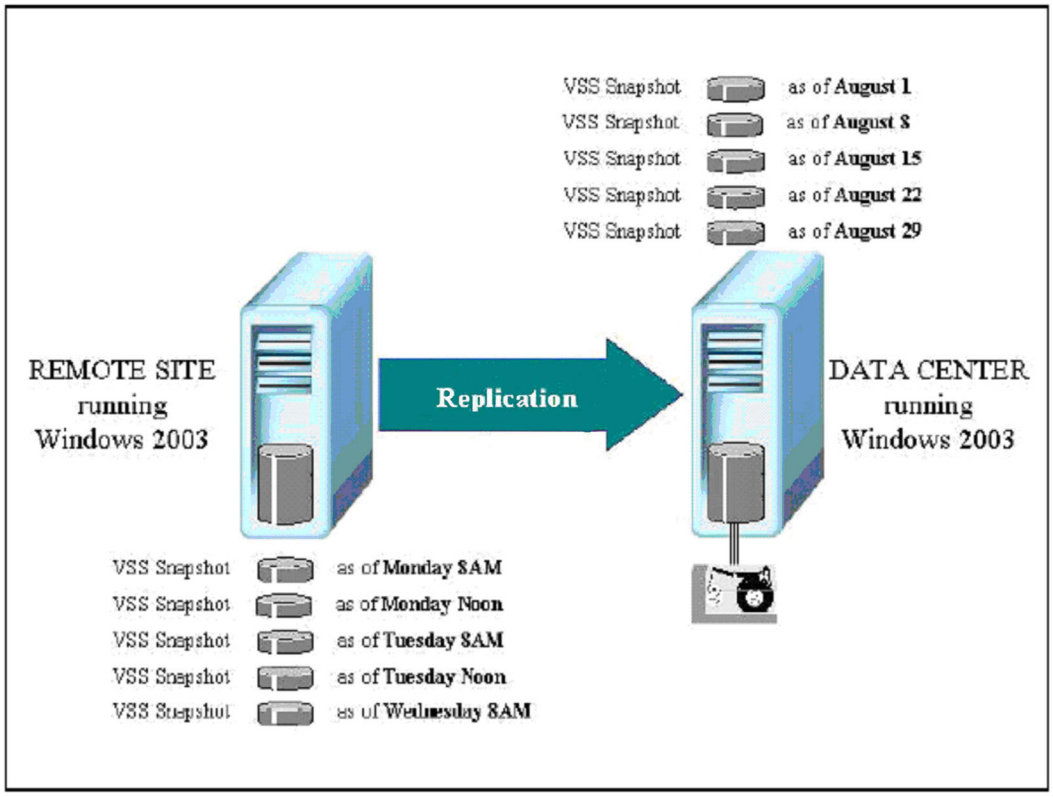
In either model, after the "new" server has a functioning operating system and application directory, the only restoration is the data set (from Storage Mirroring), which will be seconds old. This results in near zero loss of data, including the precious registry information.

For more information about how to protect and restore the registry and other system state components, visit the Microsoft website at www.microsoft.com/.

What about snapshots such as VSS in Windows Server 2003?

One of the most exciting enhancements to data protection beyond tape backup is the built-in feature of Windows Server 2003 called Volume Shadow Copy Service (VSS), which allows administrators to create a point-in-time snapshot of a file server volume. A snapshot can be taken at any time even if files are still open and can be configured automatically at intervals up to every two hours. Each time a snapshot is taken the current contents of a file are frozen and any future changes are tracked and saved to a different part of the disk. This process is transparent to users but provides them with the ability to restore a file to a previous version on their own without the need to restore from tape, reducing the number of support calls.

Figure 5. Data protection using snapshots, replication, and backup



Real data protection = snapshots + replication + backup

HP OpenView Storage Mirroring operates in complement to the Windows Server 2003 VSS. VSS can be used in conjunction with the replication technology between servers.

If one creates snapshots from the production (source) server, then users can have historical access to older copies of their local data. Transparent to this, the current data will continue to replicate from the local server to the remote data center.

If one snapshots the redundant (target) server, then the IT team at the corporate data center will have the same historical access to the data. This is preferable if storage space is limited at the branch, but multiple copies are desired.

Used together, one might provide 14 daily snapshots within a local branch, which allows the users to do self-directed restores of data for two complete weeks. In addition, one might do weekly snapshots of the data center copy, which might allow for upwards of 60–90 days of online restorability from the data center (all without ever mounting a tape).

How does restore work?

By this point, it should be clear that there are several advantages to using replicated data for "better backups." Nevertheless, backing up is simply preparing to restore. Therefore, it is important to understand how restorations work in this solution.

- Whole Dataset Recovery—For the scenario where a data volume (or disk set) has been damaged and must be restored, the Storage Mirroring replication and mirroring processes can be put "in reverse"—pushing the data from the target to the source. One repairs or replaces the storage on the production source server. The Storage Mirroring database is aware of where the various target data files came from. Therefore, one can use the Storage Mirroring Restoration Manager to select a set of files and then use Storage Mirroring engine to put the files back where they came from.

  The difference between using the replicated files for restoration and last night's tape is the currency of the restore. The Storage Mirroring copy of the files will be seconds away from what the production source had at the moment of failure. Last night's tape would have lost all the files that had been changed during the entire business day.

- Individual File Recovery—For the scenario where the source server has lost a few files, there are a few options:
  - Storage Mirroring can be configured to "burst" the changes instead of real-time replication. The result is a copy of the files on the target, which can be minutes to hours behind the source server. This allows a redundant copy from which to quickly restore.
  - Tape or disk snapshots can be configured to protect the files on the target server—even while the production source files are in use. With this approach, one can go to a snapshot from this morning or a differential tape from two hours ago—and recover the file (all without impacting the production users). Restoring the errant file directly to the source server (by way of snapshot UI or backup console) will provide the recovered file to the users—and it will be immediately replicated back to the target to provide consistency for all copies.

In all cases, the inherent tools of Storage Mirroring provide for easy restoration. As an example, the Storage Mirroring patented "partial difference mirror" allows large files to be restored by only restoring the partial sections of the file that have actually been changed. The unchanged sections are untouched, which significantly reduces bandwidth requirements and restoration times.

What about application-specific data and configurations?

For those applications that store parts of their configuration information outside of the data set, replication can still be used to provide a better backup.

If an application stores its configuration information as flat files (for example, INIs), then one configures the replication set to protect those directories, with optional filters to include the configuration files, exclude large binaries, or both.

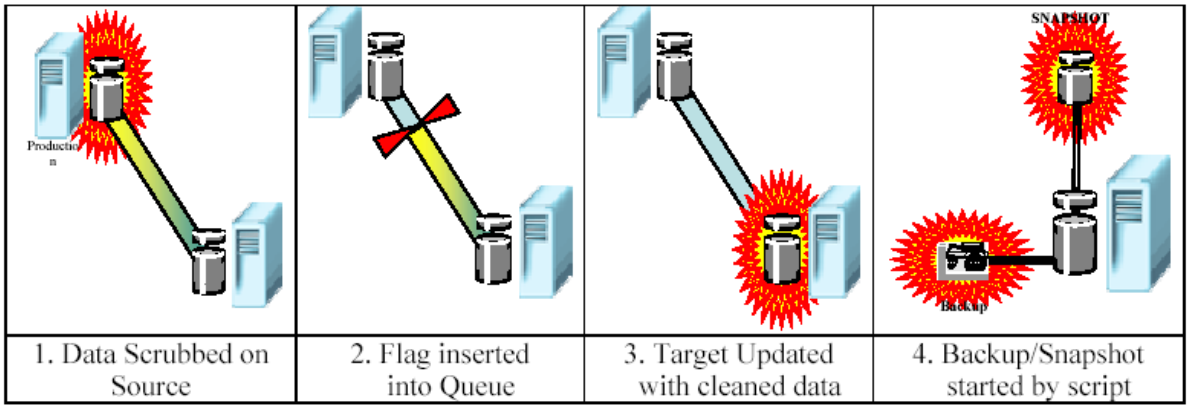
If an application stores its configuration information in the registry, one can use REGDMP to routinely secure those particular registry hives to a flat file (which would be replicated to the target server, similar to the system state information discussed earlier). REGINI would be used to restore those registry hives during a restoration activity.

Some applications do a month-end "scrub" (wholesale changes, compaction, and so on). For those environments, three additional benefits of Storage Mirroring come into play:

- Extended queuing—Storage Mirroring provides for a queuing model to cache up to 4 TB of byte-level changes, so that even the most dramatic data changes can be propagated to the target server. Since most environments do these types of operations on weekends, a properly configured queue and infrastructure will ensure that both copies are maintained by Monday morning.
- Scheduled verification and scripting—Some users choose to temporarily disable replication, when the same large data areas will be repeatedly scrubbed within a short window. Instead, using the Storage Mirroring application Command Language (CL), the real-time replication of Storage Mirroring is turned off while the data is modified. Upon completion of the data compaction, CL can be used to re-enable the replication and initiate a "scheduled verify." The verification will compare strings within the source and target files, and only send those sections that are determined to be different. This can reduce the bandwidth impact during large repetitive scrub operations.
- In-band Command Processing—Many HP customers want to do other activities to the target copy of the data, after they are assured that all of the scrubs are complete.

  Common examples include a fresh backup or invoking a snapshot. To accomplish this, HP provides the ability to insert a flag immediately behind the replication traffic of some operation. So, an application can perform its scrub and then use CL to insert the flag. Upon the target server receiving the flag, users can be assured that the target has also received all of the data from the scrub operation. At that point, a script is invoked—for a backup or snapshot.

Figure 6. How In-Band Command Processing works



These benefits are based on using replication to enhance a user's existing or planned backup solution because the target's data is a better copy to back up.

All of these solutions are based around the fundamental philosophy that all business continuity efforts start with protecting the data. From there, it is a matter of deciding what to do with it. And while HP continues to be the leader in Windows business continuity, high availability, and disaster recovery solutions, the same software can be leveraged for the more daily protection of Windows file systems by complementing an existing backup solution.

For more information

HP Storage Products:

http://welcome.hp.com/country/us/en/prodserv/storage.html

HP Storage Software:

http://h18006.www1.hp.com/storage/software.html

HP OpenView Storage Mirroring:

http://h18006.www1.hp.com/products/storage/software/sm/index.html

5982-6824EN, 06/2004

invent