# Block Based Backup Technology

## Block Based Backup Technology

Block based backups bypass files and file systems almost completely. All operating systems have a specialized component of the O/S called the File System. Examples of popular server file systems are NTFS on Windows and ext3 for Linux. The file system divides the hard disk, volume or RAID array (software and hardware RAID) into chunks or groups of bytes called blocks (fixed size) or extents (variable size). Typically these are ordered 0 - N. The size of a block depends on the file system used and potentially settings used when the file system was created. NTFS and ext3 use fixed block sizes. Some file systems support a concept of variable length block sizes typically called extents.

The file system is responsible for keeping track of the tree or hierarchy of files. It also stores a file in neat little fixed size blocks on the disk and keeps track of where these blocks are which can be scattered across the disk. Backup applications that read files use the file system to get at data and are inherently very slow and time consuming no matter what file system is used.

Block based backups bypass the file system and read data directly form the Disk or Volume. Block based backup applications can either read data in the same sized blocks as the file system or in different sized blocks. It is not important.

A big advantage of bypassing the file system is that there is no penalty on backup performance for having a large number of files. The backup application never looks at files and does not care how many there are! It also reads blocks in the order that they are on the disk, not the order that they appear in files which tends to be heavily fragmented and causes the disk(s) to spend more time seeking than actually reading data.

Block based backups always have built-in support for point-in-time snapshots. So using block based backups always gives you open file backup and snapshot features by their nature. They always start a backup operating by first taking a snapshot of the live running volume. They may also perform special functions to put running applications and their in-memory data on the server into a consistent state like databases. They then read block level data from the snapshot not the actual disk. The method for taking a snapshot can vary from application to application and by O/S.

Almost always the snapshot is maintained using some kind of a copy-on-write mechanism. This works by maintaining the snapshot by pausing writes that overwrite data in the snapshot and making a backup copy to another location before allowing the write to proceed. This backup copy in combination with parts of the disk that have not been overwritten maintain the snapshot.

Typical Block Based Backup Process

1. Take point-in-time snapshot of live volume

2. Compute Check Sum Deltas By Reading All Disk Blocks (R1Soft CDP skips this step when CDP is in sync!)

3. Read Deltas

4. Delete point-in-time snapshot

> All backup applications that backup at the block level support some kind of snapshot. Beware that not all snapshot methods produce consistent snapshots and it can be difficult to determine if snapshots are file system consistent or not. Ask technical contact at your vendor directly if their block based backups take a file system consistent snapshot and under what conditions.

R1Soft uses Volume Shadow Copy on Windows and a proprietary snapshot device driver on Linux. MS Volume Shadow Copy Service is a giant facility that can be used for many different purposes. One feature of VSS is the ability to take a block level point-in-time snapshot of a disk Volume. VSS guarantees that the file system is consistent when a snapshot is performed.

The R1Soft Linux snapshot driver uses special proprietary techniques only available to Linux device driver authors that place the file system into a guaranteed consistent state on 2.6 Linux kernels. On 2.4 kernels R1Soft uses a quiesce method of making a snapshot.

Typical Implementation of Consistent File System Snapshots (e.g. R1Soft Linux Snapshot Driver and MS Volume Shadow Copy Service)

1. Notify supported applications of backup operation.
2. Wait for applications to write in-memory changes to disk and become consistent.
3. Initiate File System Freeze.

4. File System Locks and Pauses All Requests for New Transactions (e.g. append, truncate, create, write, etc.).

5. File System allows in-process transactions to finish all their block level changes to disk.

6. Dirty file system pages in O/S Page Cache is flushed from memory to physical media.

7. File System is Unfrozen and normal activity proceeds.
8. Supported applications are notified snapshot has completed and to resume normal activity.

9. Backup application copies data from snapshot as needed. Could be all data or deltas. Deltas can be computed using several methods.
10. Backup application destroys the snapshot.

Note: Most Windows CDP applications (R1Softis an exception) use MS VSS for shared folders. These applications don't delete the snapshot in step 10 when done and instead must keep the snapshot and perform copy-on-write the entire time until the next backup operation. This is because the method of computing deltas (VSS for shared folders) works by comparing two Volume Shadow Copy Snapshots.

The process of making a file system snapshot usually takes less than a second and if done correctly is never noticed by users or running applications. R1Soft's Linux snapshot driver usually completes this function in a few milliseconds. MS VSS usually takes around a second but can be delayed by Writers (notified applications) taking too long to become consistent for the backup operation.

Some backup application vendors use methods for taking snapshots that do not guarantee file system consistency. These should be avoided when possible. In the case of 2.4 Linux it is unavoidable. Linux kernels greater than 2.6.8 have low level device driver facilities for making file systems consistent during a snapshot. Windows XP and greater support Volume Shadow COpy which gauruntees a consistent file system during a snapshot. Beware of applications that brag about their method for determing a snapshot using a "quiesce period". You can identify these by reading about their snapshot process. They will usually describe a process that waits for a quiesce period (e.g. 5 seconds) to be "observed" and once the disk is quiesced or observed to be quiet it is therefore "assumed" the file system is in a consistent state! For an example see the OTM technical paper at http://www.cdp.com/library/OTMSale.pdf

Block based backups are typically file system dependent. This is because they must at a minimum read low level file system data structures to determine which parts of the disk are not in use so that those blocks can be excluded form the backup.

This means to get all the performance advantages of block based backups the application must give up supporting a wide variety of platforms. For example R1Soft only currently supports Windows and Linux. In fact most block based backup applications only support one or two platforms. Most only support Windows. R1Soft is the only near-Continuous backup application that supports Linux as well as Windows. All other near-Continuous backup software only supports Windows. For example Symantec BakcupExec supports a variety of O/Ss. However their near-Continuous add-on module Panther aka 10d only supports Windows.

Block based backup applications are usually dependent on the type of file system used on your server. Check and make sure your file system is supported. If you have a Windows server you have little to check. You are using the NTFS file system. If the vendor supports block based backups of Windows servers then you are covered. If you are using Linux than check your file system to see if it is supported. Linux has support for over a dozen different file systems, while all popular Linux distributions use either reiserfs or ext3. R1Soft supports NTFS, ext2/3/4, and reiserfs.

Not all block based backup applications support individual file restores. For example R1Soft supports both bare-metal image style restores and file level restores. Some applications only support imaging and bare-metal type restore. This is because it takes a lot of development work to be able to read files out of low level block backups. And the vendors that do support restoring files from block level backups only do it if you read the block level backups on the same system you ran the backup on. For example if you install Acronis True Image on a Windows server. It creates block based backups and uses a checksum delta method. If you want to restore a set of files form the image Acronis True Image has a special device dirver that makes the image backup appear to Windows like a real hard disk. They then use Windows to read files out of it. The same thing is true on Linux.

R1Soft takes a different approach. R1Soft has made a comitment to being cross-platform on Windows and Linux. Truely cross platform. This means you can use a Linux CDP Server to perform Continuous Data Protection for both Windows and Linux and the opposite is also true. You can use a Windows CDP Server to perform Continuous Data Protection for both Linux and Windows. For example this means you you prefer to maintain Windows servers and have to provide backup services for Linux servers you can do it all with Windows.

R1Soft is the only vendor doing and we do it using special proprietary file system code that can read files and file attributes from raw disk blocks in a platform independent fashion. This means we don't depend on the O/S when we read files. We do it ourselves. And this has some big advantages. In addition to being cross platform in how you manage your server backups. It allows is to do cross-platform file restores. You can restore files out of block-based backup done on a Linux server to a Windows server and visa-versa. R1Soft can also transfer the files for you at restore time. The other applications require that you setup your own NFS or CIFS share for the backup application to store and read backup data from. R1Soft has its own network protocol for moving the files and block data so you don't have configure a network file system.