File Based Backup Technology

File Based Backup Technology

Backup applications can read data from the disk in one of two ways using files or blocks.

The easiest way is by using the standard file API (Application Program Interface) like fopen() and fread(). These C/C++ programming standard API functions exist in one form or another in all programming languages and work on all platforms and they simply open and read data from files, nothing more.

On Unix and Linux the stat() function is used to read file attributes like modification date and file permissions. On Windows operating systems backup applications use the equivalent API specialized for backup applications and call BackupRead(). BackupRead() does nothing more than read all of the data belonging to a file and also packs in file attributes like access control lists.

The backup application simply starts at the top of the drive and traverses down examining all of the files and directories using one of the defined Delta Computation Methods (Computing Deltas - File Attributes, Computing Deltas - Check Sums, [Computing Deltas - VSS For Shared Folders]) to determine if a file has changed and should be included in the backup set.

The advantage of this approach is that it usually results in a highly portable application written in C or C++ can be made to compile and execute on almost any operating system. If you see a backup application that seems to support all operating systems as some do you can probably be sure it is a file based backup application. Symantec BackupExec, CA ArcServeIT, NT Backup, and IBM Tivoli are all examples of file based backup applications. Some of them claim to support a giant list of operating systems and this is how they do it.

There are serious disadvantages to backing up data at the file level.

1. Bare-metal restore is not possible. Its impossible to perform a true bare-metal restore with file level backups.

2. File level backups often consume problematic amounts of system resources on servers with moderate to large numbers of files. This is because these applications must index the path to every file on the server. This can consume a very large amount of memory. It also takes a very long time to traverse the directory tree to simply list all of the files eligible for backup.

Windows

For example how long does it take to do this:
Open Windows Explorer on your server. Search for files. Search your largest volume and start searching at the top. Leave the search criteria blank. Doing this will generate a list of all files on your volume. How long does it take? That is what file based backup software must do every backup operation even if no data has changed. And all that time is just getting a list.

Linux
How long does it take to run this command on your Linux Server?

```
# find /
```

If you haven't tried it give these examples a try. Then add in time to compute deltas and its no wonder file based backup is so slow. Many minutes can go by some times even hours and you have done nothing but index all of the files. Now add in the time it takes to actually read data, and its no wonder you are backing up weekly or daily at best.