the Availability Digest

www.availabilitydigest.com
@availabilitydig

TANDsoft FileSync Adds Deduplication

February 2014

FileSync[1] is a file-replication utility from TANDsoft, Inc. (www.tandsoft.com). FileSync ensures that the configurations of two HP NonStop data-processing systems are kept synchronized by replicating Enscribe and OSS file changes between them. FileSync is useful to keep a backup system synchronized with its production system, to propagate upgrades between systems, and to migrate applications from one system to another.

In this article, we look at how data deduplication enhances FileSync's role in keeping a backup system synchronized with its production system. With deduplication, only changes to a file need to be replicated to the backup system rather than the entire file, thus dramatically reducing time and bandwidth requirements.

Active/Backup Systems for High Availability

A common technique for guaranteeing system availability is to configure a backup system that can take over processing should a production system fail.

Configuration Drift Causes Failover Faults

A major hurdle to achieving high availability with active/backup architectures is *configuration drift*. If the software versions of programs, scripts, and configuration files resident on the backup system are not up-to-date, version conflicts may prevent the backup system from operating properly. Failing over to the backup system following a production system failure may be unsuccessful, resulting in a *failover fault*. Equally important, testing failover is made much more complex if version errors must be tracked down and corrected in order to successfully pass a test.

To ensure that a failover will be successful, it is important that all versions of software running on the backup system match that of the properly operating production system. Tools are available to compare the production system software modules to those on the backup system to detect version errors. If such errors are found, operations staff must take steps to correct them.

A more advanced solution is to have a facility that not only will detect version errors on the backup system but that also will automatically correct such errors. Such a facility is FileSync from TANDsoft, Inc. Coupled with TANDsoft's Command Stream Replicator, FileSync relieves the operations staff from having to continually monitor and correct backup software versions.

[1] FileSync and CSR Synchronizes NonStop Systems: Part 1 – FileSync, *Availability Digest*; October 2011. http://www.availabilitydigest.com/public_articles/0610/filesync.pdf

Synchronizing the Backup System

Three classes of objects are involved in system synchronization to ensure proper failover:

- *Audited Databases*: Several products are available for replicating changes in real time made to a NonStop audited database, whether it be Enscribe or SQL. They include RDF from HP, Shadowbase from Gravic, Inc., DRNet from Network Technologies, Replicate from Attunity, and GoldenGate from Oracle. These products read changes from the TMF source audit trail and replicate them to the target database.

- *Unaudited Files*: The primary function of FileSync is to synchronize unaudited files. An unaudited file may contain several types of data, such as program source code and executables, scripts, configuration files, or application data. Prior to version 3.1, FileSync ensured synchronization of unaudited files by replicating in its entirety each file that had changed. As of version 3.1, FileSync uses data deduplication to replicate only the changes to these files. This dramatically reduces the time and the bandwidth required for unaudited file synchronization.

- *Configuration Changes*: Various NonStop utilities, such as FUP and SQLCI, are provided to change the configuration of the processing environment. These configuration changes must be made to the backup system as well. This is the job of TANDsoft's Command Stream Replicator.[2]

FileSync

FileSync synchronizes application environments and unaudited files across one or more NonStop servers. The target servers may be disaster-recovery systems or other production systems (for instance, servers in an active/active network). Bidirectional replication in an active/active environment is possible by configuring a FileSync subsystem for each direction.

FileSync can replicate any Guardian or OSS files on a NonStop system. FileSync can replicate files across either Expand or TCP/IP links, though the use of Expand is the more efficient option. However, TCP/IP is required if the node names of the source and target servers are the same.

FileSync replication is batched. The following schedule alternatives are supported:

- *Periodic*: Files may be replicated at fixed intervals in increments of minutes. The shortest interval is one minute.

- *Specified Times*: Replication can occur at specified times throughout the day.

- *Interactive*: An operator can initiate FileSync replication at any time by invoking a TACL script.

- *Event*: Replication can be invoked by a trigger generated upon the completion of some external event.

Files to be replicated are specified in *qualified file lists*. A qualified file list specifies a set of objects and files to synchronize. They optionally include clauses that restrict the objects and files based on their attributes. As an example, to specify all source-code files (filecode 101) on $DATA, the qualified file list would be "$DATA*.* where filecode=101."

A file list contains all files that have the same replication options specified. A file will be replicated if its source time stamp is greater than the time stamp of the target file (that is, the source file has been modified subsequent to its last replication to the target). Files that do not exist on the target system or that are marked as corrupted on the target system are also replicated.

[2] FileSync and CSR Synchronize NonStop Systems: Part 2 – Command Stream Replicator, *Availability Digest*; November 2011. http://www.availabilitydigest.com/public_articles/0611/command_stream_replicator.pdf

Should a system or network failure interrupt a FileSync replication job, the current FileSync job is terminated. Files that were in the process of being replicated are marked as being corrupted on the target system. The remaining files and any corrupted files will be replicated on the next invocation of the FileSync job.

In addition to replicating files, FileSync can list all files that are in sync, out of sync, or designated for purging.

Deduplication Increases FileSync Efficiency

Prior to FileSync Version 3.1, FileSync replicated an entire file when something – even one byte – was changed. This is appropriate for small files that seldom change, such as configuration files. However, some files can be very large and very active. Replication of these files in their entirety can be prohibitive.

Data Deduplication

FileSync Version 3.1 adds the option of data deduplication to file transfers. With FileSync Dedup, only blocks that have changed since the last time that the Guardian or OSS file was replicated are sent to the target system. Consequently, the time and the bandwidth required to replicate files is substantially reduced because a multi-megabyte file can now be updated by sending just the few blocks that have changed rather than having to send the entire file over the communication channel.

Deduplication proceeds as follows. If a file does not exist on the target system, it is replicated in its entirety as in the earlier versions of FileSync. In addition, each 4K block in the file is hashed, and the hash values are stored in a Hash File. Thus, the Hash File represents the file as a sequence of hash values rather than data blocks. The MD5 128-bit hashing algorithm is used by FileSync Dedup.

Furthermore, an entry for the new file name is made in an Index File. The Index File is a registry of all files that are being replicated. For each entry, the characteristics of the file are maintained as a file label (similar to a magnetic tape label). For instance, the current time stamp for the file is recorded in its Index-File label entry.

Subsequently, if FileSync determines that a file has changed based on the file's timestamps in its source and target versions, FileSync repeats hashing the entire file and compares the new hash values to those stored in the Hash File. It stores any changes in a Data File by block number. For instance, if a block is added, the new block data is stored along with the block number that it now follows. If a block is updated, the new data is stored along with the block number of the block that is being changed. If a block is deleted, its block number is stored.

When the parsing of the file changes has been completed, FileSync inserts the file's Index File entry into an Update File. The Data File with all of the block changes and the Update File with the file's new label information is sent to the target system, where the changes are applied to the target file.

Deduplication is specified as an option for each file list. Therefore, files are deduplicated only if it is desired to do so.

FileSync Deduplication Architecture

The software architecture for FileSync Dedup is shown in Figure 1. FileSync can be invoked from a TACL command or according to a schedule, as described earlier. FileSync is given a file list that contains the names of the files to be replicated along with replication options.

If "dedup" is not specified for the file list, the file list is sent to the original FileSync utility. The architecture of this utility is described in Reference 1 above. To transfer Guardian files (including SQL/MP), the

NonStop backup/restore and the utilities PAK and UNPAK are used. To transfer OSS files and SQL/MX files, NonStop BR2 (backup/restore) and utilities PAK2 and UNPAK2 are used.

However, if “dedup” is specified as an option, the file list is sent to the Dedupe process in FileSync Dedup instead. If FileSync does not have a file in its registry, it will add it to its Index File.

The Dedupe process will check each file in the file list to see if it needs to be replicated. If a file does not exist on the target database or is marked as corrupted, it is replicated in its entirety to the target database. The blocks comprising the file are hashed and added to the Hash File, and the file’s label information is inserted or updated in the Index File.

If a file is to be replicated because it has changed, the Dedupe process parses the file and calculates the hash values for each block. It compares these hash values to the current hash values for the file as stored in the Hash File to determine what has changed. The block changes are added to the Data File, and the Hash File is updated with the new file contents. The registry information for the file is read from the Index File and stored in the Update File. If the replication channel is TCP/IP, the Data File and the Update File are sent to the target system where they are stored on the target-side. If replication is being done over Expand, the target system reads the Data File and Update File on the source system.

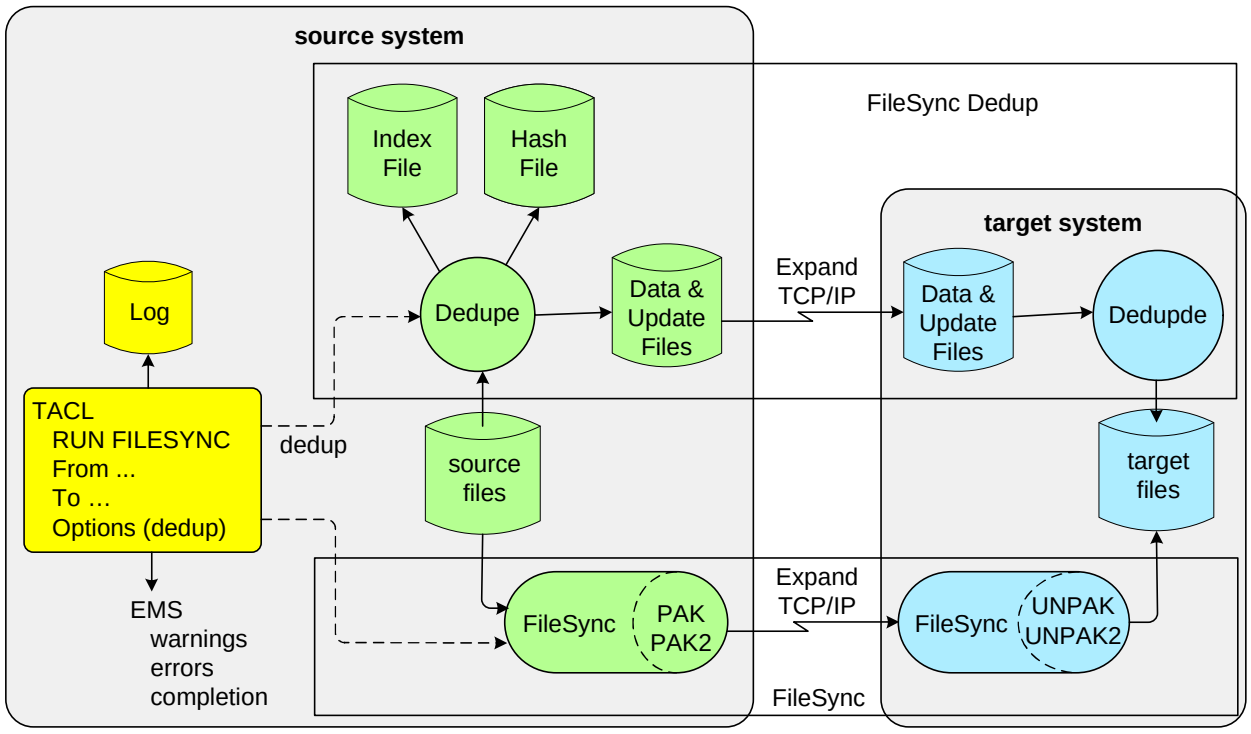

Figure 1: FileSync Data Deduplication Architecture

At the target system, the Dedupde process reads the changes from the Data and Update Files and updates the target database with the changed blocks. Therefore, the only data that has to be sent from the source system to the target system to update the target files are the changed blocks. Any block that had not changed is not replicated.

Summary

Configuration drift is a major problem leading to failover faults when a production system fails and its backup won't come into service. FileSync is a major utility to help prevent configuration drift. It automatically keeps Guardian and OSS files on the target system synchronized with the source system by replicating files that have changed, that have become corrupted, or that do not exist on the target system.

With Version 3.1, FileSync adds data deduplication. This greatly increases the efficiency of FileSync because only changed data must be sent to the target system rather than entire files. If a file does not exist on the target system, it is first replicated to the target system in its entirety. Thereafter, only changes to the source file need to be sent to keep the target file synchronized with the source file.