the Availability Digest

www.availabilitydigest.com

FileSync and CSR Synchronize NonStop Systems
Part 2 – Command Stream Replicator

November 2011

Failover to a backup system often fails because the backup system's software configuration is different from that being run by the production system. We call this *configuration drift*.

For HP NonStop systems, NonStop RDF (Remote Data Facility) and third-party data replication engines can synchronize database contents. FileSync from TANDsoft can synchronize files. However, what is left is configuration changes entered via a variety of utilities.

Command Stream Replicator (CSR) from TANDsoft fills in the last piece of the configuration synchronization puzzle. CSR replicates specified operator commands entered on the production system to the backup system or to other target systems in order to keep the configurations synchronized.

Fighting Configuration Drift

Configuration drift can cause failover problems, for instance, if versions of the applications are different, if scripts are missing, or if the database structure has changed. The result is that it might not be possible to bring up the backup system, or that the applications will not run properly. As an example, an up-to-date version of an application may not be able to process transactions against an earlier version of a database whose structure has been changed.

In order to ensure that failover will work properly, it must be tested regularly. Failover testing is often a complex, lengthy, and risky task that requires taking down application services until the backup system is up and running. The testing task is only further complicated and extended if version conflicts must be detected and resolved. For these reasons, full failover testing is often not performed periodically or not performed at all. The organization would rather rely on faith and hope that the backup will come up.

Therefore, it is imperative to maintain the software configuration of the production system and its backup in synchronism to avoid failover faults. Tools are available to periodically compare software on the two systems and to report discrepancies so that they can be corrected.

However, a better solution is to have a facility that not only will detect version errors but will automatically correct them. Such a facility for HP NonStop servers is FileSync from TANDsoft (www.tandsoft.com). FileSync, described in Part 1 of this series,[1] compares file versions of files on a production system to those on its backup system and automatically replicates updated file versions to the backup system. Validated files can include database files, source programs, executables, configuration files, scripts, and others.

[1] FileSync and CSR Synchronize NonStop Systems: Part1 – FileSync, *Availability Digest*; October 2011. http://www.availabilitydigest.com/public_articles/0610/filesync.pdf.

© 2011 Sombers Associates, Inc., and W. H. Highleyman

Maintaining file versions is only part of the synchronization problem. The other part is the impact on a system by operator commands entered via various utilities. With these, an operator can modify the configuration of a system. He can change the structure of an Enscribe, SQL/MP, or SQL/MX database, he can change configuration parameters for Pathway, and he can change the configuration of communication lines, to mention just a few. These configuration parameters are often not included in configuration files that are replicated by FileSync. Rather, if nothing else is done, the same operator commands must be entered at both systems. This is an error-prone manual procedure that leads to many failover faults.

Command Stream Replicator (CSR) from TANDsoft corrects this deficiency. CSR replicates specified operator commands entered on the production system to the backup system to keep the configurations synchronized.

Synchronizing the Backup System

As pointed out in the first part of this series, there are three classes of objects involved in system synchronization to ensure proper failover:

- *Audited Databases*: HP's RDF and several third-party products from vendors such as Gravic, Network Technologies, Attunity, and GoldenGate replicate changes made to a NonStop audited database. These products are typically used to keep SQL or audited Enscribe databases in synchronism.

- *Unaudited Files*: FileSync ensures synchronization of NonStop unaudited files by replicating the new version of an entire file or by replicating only changes made to a file.

- *Configuration Changes*: Various NonStop utilities such as FUP, SQLCI, MXCI, and PATHCOM are used to change the configuration of a system. It is the role of CSR to replicate these changes from a production system to its backup.

Coupled with a database replicator, TANDsoft's FileSync and CSR provide all the facilities needed to ensure that the software configurations of two systems are synchronized. In addition to active/backup configurations, these capabilities are useful for active/active networks, for system migration, and for other multisystem tasks.

Command Stream Replicator

Database and file replication utilities do not replicate system configuration changes. Consequently, configuration changes must be made to each system individually. This often requires taking both systems offline until the configurations of both systems have been synchronized.

The Command Stream Replicator fills this void. CSR is the final piece in the puzzle to keep NonStop systems synchronized. It replicates everything that database and file replicators do not, such as changes to database structures, system configuration changes, and other NonStop utility operations. CSR replication requires no system downtime or other operator intervention. It therefore enables reliable failover to a backup system as well as continuous processing for active/active systems. It is compatible with all NonStop data-replication products including HP's RDF and other third-party replicators.

Database Structures

Database structures are defined by the Data Definition Language (DDL). Modifications to a database's DDL are often made to increase the functionality of the database.

© 2011 Sombers Associates, Inc., and W. H. Highleyman

Many DDL operations, such as SQL CREATE and ALTER, are not inserted into the audit trail by NonStop's transaction monitor, TMF (Transaction Monitoring Facility). Therefore, these operations are not replicated by database replication engines that are keeping the databases synchronized.

CSR captures SQL/MP, SQL/MX, and Enscribe DDL operations and replicates them to the backup database automatically without stopping or suspending applications, TMF, or the database replicator products. It does so by linking into the utilities that update the DDL definition of databases, including FUP for Enscribe, SQLCI for SQL/MP, and MXCI for SQL/MX.

Before executing a DDL command on the target system, CSR will stop any database updaters that are modifying the target database. Once the target database has quiesced, the DDL command is executed against the database; and the updaters are then restarted.

.

Other Utilities

In addition to the DDL utilities, CSR supports TACL, PATHCOM, SAFECOM, SCF, and other FUP commands. TANDsoft is in the continual process of updating this list and will instrument other utilities upon request. Other utilities can even include custom utilities written by an organization.

Operational Modes

CSR has several modes of operation that have different characteristics:

- *Non-Sequenced Asynchronous Mode* intercepts selected utility commands and stores them in a local Command File. Periodically, the Command file is replicated as a batch to the target systems that must be kept in synchronization; and the commands are executed locally on those systems.

- *Sequenced Asynchronous Mode* intercepts selected utility commands and stores them in a local audited Command File. The Command File is accessed directly over Expand or is replicated in near-real time by a data replication engine to the target systems and the commands are executed locally on each system.

- *Interactive Synchronous Mode* intercepts selected utility commands and sends each immediately to one or more target systems for execution. Command responses are returned to the source system.

- *Serialized Mode* intercepts selected DDL commands and writes them to the TMF Audit File with the DML commands that are currently being executed. At the target, replicated DDL commands are executed as they are received, serialized properly with the DML commands.

CSR Architecture

Configuration

The commands that CSR must replicate are defined in a Configuration file. The Configuration file defines the CSR environment such as the licensing information, the Command File, and, most importantly, the specification of the commands to replicate.

The command filter specifies the utilities whose commands are to be replicated. For each utility, the specific commands to be replicated are listed; and the users whose actions for this utility will be replicated are specified. Wild cards may be used. Multiple filters can be specified for CSR. For instance, certain commands in a utility may be replicated if entered by certain users, and other commands may be replicated for other users.

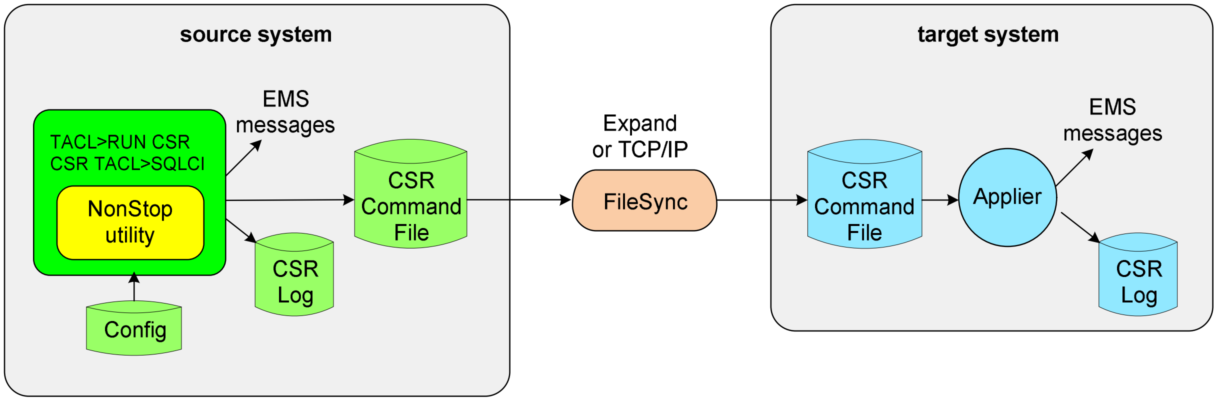
Non-Sequenced Asynchronous Mode

When using the asynchronous modes, CSR captures and stores commands in the Command File, from where they are transmitted to the target system for execution.

Capture

The Command Stream Replicator is started via a TACL (the NonStop scripting language) either manually or via a batch script. Once TACL is running, a utility such as FUP or SQLCI is invoked from the TACL.

As commands are entered into the utility, CSR captures them and compares them to the command filter. It the command is not to be replicated, it is passed to the source system for execution.



CSR Architecture
Non-Sequenced Asynchronous Mode

If the command is to be replicated, it is executed on the source system. When the command completes, it is written to the Command File along with its response, the user ID, and the user's environment (volume/subvolume).

Replicate

In Non-Sequenced Asynchronous Mode, the Command File is sent periodically to the target system via FileSync over TCP/IP or Expand. Once it is resident on the target system, the Applier can read the Command File contents.

Apply

It is the responsibility of the Applier on the target system to execute the commands captured by CSR on the source system. The Applier will read commands from the Command File and execute them on the target system.

With FileSync, a trigger can be used to modify node-specific names and to inform the Applier that a new set of commands is ready for target processing.

Log

During operation, in addition to the EMS messages that CSR generates, all errors are logged to a local CSR Log file.

© 2011 Sombers Associates, Inc., and W. H. Highleyman

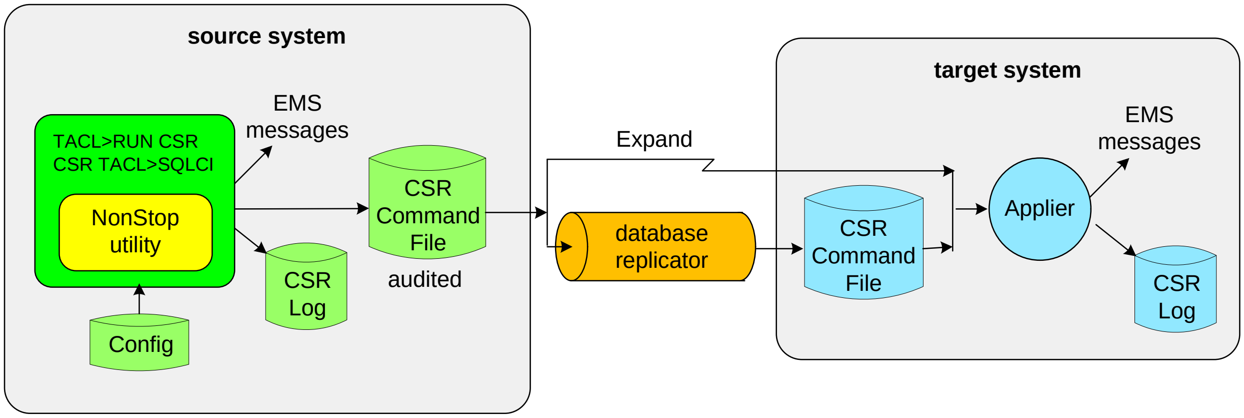
Sequenced Asynchronous Mode

CSR's Non-Sequenced Asynchronous Mode replicates commands periodically, typically every few minutes. Therefore, there can be a significant delay from when the command is executed on the source system to when it is executed on the target system.

The Sequenced Asynchronous Mode can significantly shrink this delay. For instance, it is important that a DDL operation be executed on the target system before a DML operation that depends upon the DDL change (for instance, a CREATE TABLE DDL command followed by an insert DML operation).

In this mode, the CSR Command File is TMF-audited and is sent to the target system via Expand or a data replication engine. If Expand is used, the CSR Applier accesses the audited Command File on the source system directly. If a data replication engine is used, it replicates data changes to a target-side Command File that is accessible by the Applier.

In this way, commands executed on the source system will be available to the target system in near-real time. The replication delay is reduced from minutes to typically seconds.



CSR Architecture
Sequenced Asynchronous Mode

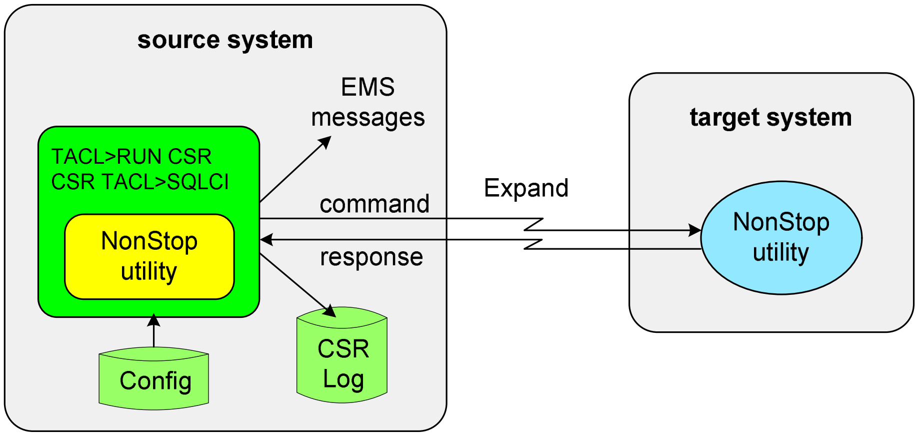
Interactive Synchronous Mode

In CSR's asynchronous modes, commands are executed at some time (seconds or minutes) after they are executed on the source system. The only way to determine the results of a command executed on the target system is to view the target system's Log file.

CSR's Interactive Synchronous Mode executes a command on the target system simultaneously with executing it on the source system. The results of the target execution are returned to the source system. The target and source systems must be connected via an Expand network.

In this configuration, commands are intercepted as they are in CSR's asynchronous modes. However, rather than being buffered in a Command File, they are sent directly over the Expand link to the NonStop utility on the target system. The NonStop utility on the target system will execute the command immediately and return the response to the source system. There the target command and its response will be displayed along with the local response.

© 2011 Sombers Associates, Inc., and W. H. Highleyman



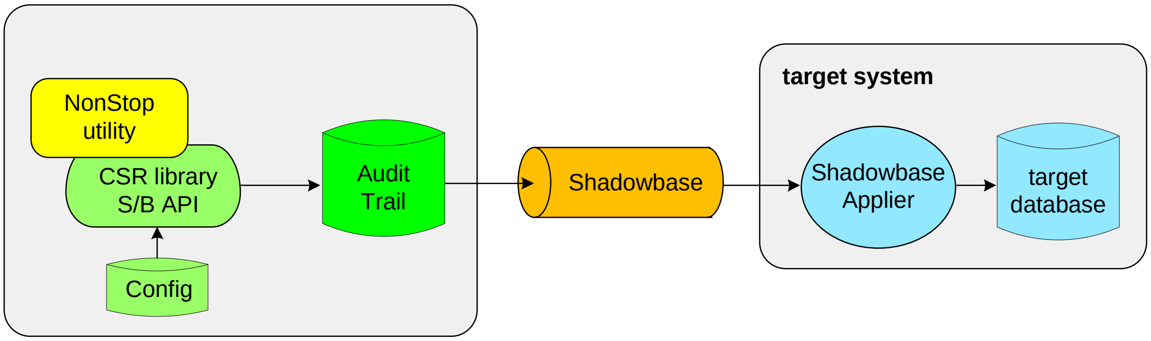
CSR Architecture
Interactive Synchronous Mode

Serialized Mode

Using the asynchronous architectures, CSR will not replicate commands in sequence with other database operations that are being simultaneously replicated in real time by a database replication engine. This is because CSR buffers commands in the Command File and either sends them to the target only periodically via FileSync or replicates them via a data replication engine which has some latency. In some cases, synchronizing the DDL commands being replicated by CSR with the DML commands (data manipulation language – inserts, updates, and deletes) being replicated by a database replicator is imperative.

For instance, if a DDL command is issued by SQLCI to CREATE a table, and an insert into that table is subsequently issued, it is important that the CREATE Table command be executed at the target database before the insert command. Otherwise, the target database will reject the insert command, and the source and target databases will be out of synchronization. Though a window of a few seconds may serve most needs, it timing is really critical, stricter serialization may be required.

TANDsoft has partnered with Gravic, Inc, to solve this problem. Gravic provides an API in its Shadowbase replication engine that is used by CSR to pass intercepted commands. Shadowbase will insert these commands into the TMF audit trail so that they are replicated by Shadowbase properly sequenced with other DML commands that it is currently replicating.



CSR Architecture
Serialized Mode

© 2011 Sombers Associates, Inc., and W. H. Highleyman

In this case, command interception is different than in the other architectures previously described. The utility must first be *prepared* by CSR. CSR's PREPARE utility processes the utility's object file and inserts an intercept software layer – the TANDsoft CSR runtime library – between the utility and the NonStop Kernel operating system. Since PREPARE impacts only the utility's object code, CSR does not require access to the utility's source code.

The runtime library intercepts and filters commands and feeds commands to be replicated to an API provided by Gravic. The Gravic API inserts the command into the TMF Audit Trail in proper sequence with other DML commands currently being executed.

Gravic's Shadowbase replication engine replicates the Audit Trail contents to the target system, where the Shadowbase Applier applies the DDL and DML commands in proper sequence to the target database. As with the CSR Applier, the Shadowbase Applier will stop the database updaters while it is applying a DDL command.

Summary

The Command Stream Replicator replicates everything that other replicators don't. It provides the final link required to keep NonStop system configurations synchronized. No operator action is required at the target system to execute a command that has been entered at the source system. CSR requires no application modifications, nor does it require access to the utility source code.

CSR improves failover reliability to a backup system by ensuring that the production and backup systems are uniformly configured. It supports replicating configuration changes to all systems in an active/active configuration. The result is reliable failovers and a significant simplification of NonStop system administration procedures in a multisystem environment.

© 2011 Sombers Associates, Inc., and W. H. Highleyman