HP NonStop Remote Database Facility configurations

Technical brief

Contents

invent

HP NonStop Remote Database Facility (NonStop RDF) software provides asynchronous, high-speed, low-latency database replication between two or more NonStop systems at any distance. It is not designed for cross-platform replication or data transformation, but is designed to move transactional database information as quickly as possible from one NonStop server to another for uninterrupted service. NonStop RDF extends the legendary fault tolerance of NonStop servers to disaster tolerance.

Unlike hardware solutions, such as enterprise storage systems, NonStop RDF software understands transactions rather than simply replicating bits. This means that the backup database can be used for browse access during replication and is transactionally consistent immediately after a takeover.

NonStop RDF software's exceptional flexibility can meet the needs of widely varying business requirements. This paper describes the various NonStop RDF configurations that can be used to protect customer applications and data.

Overview

NonStop RDF replicates changes from a source server holding the primary database to a target server holding the backup database in real time. The NonStop RDF family consists of NonStop RDF/IMP, NonStop RDF/IMPX, and NonStop RDF/Zero Lost Transactions (NonStop RDF/ZLT), a new add-on product to NonStop RDF/IMPX. The product(s) that the customer chooses depends on which features are needed for particular situations. For example, does the customer use NonStop Transaction Management Facility (NonStop TMF) auxiliary audit trails? Does the customer need to replicate network NonStop TMF transactions? Or does the customer need to ensure that no transaction committed to the primary database be lost in the event of a catastrophic failure?

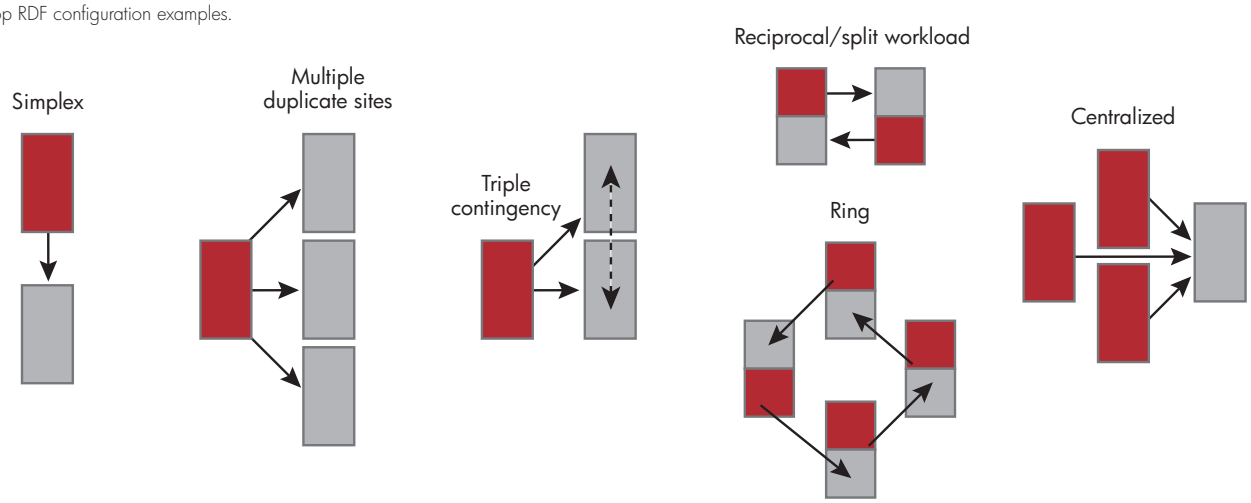
No matter which product a customer chooses or how it is configured, NonStop RDF must be licensed on all systems that it is running on, whether it is a source or target system.

The NonStop RDF data sheet includes a table comparing NonStop RDF/IMP and NonStop RDF/IMPX features. You can view this document at **www.hp.com/go/nonstopcontinuity**.

In contrast with all-or-nothing hardware solutions, NonStop RDF software does not need to replicate entire disk volumes, including noncritical information. The customer first determines what information needs to be replicated and then configures NonStop RDF replication at a file or file-set level. This also enables the customer to specify where the files should be replicated (on one or more target systems), and to adjust various system settings for the best performance. This allows maximum flexibility and minimum communications overhead. And because NonStop RDF offers asynchronous replication over ordinary communications lines, long-distance links will not cost a bundle or slow down your transaction rate. For some performance recommendations, see the *NonStop RDF/IMP and IMPX System Management Manual*; go to **www.hp.com/go/nonstopcontinuity** and select the Manual link.

NonStop RDF replication is asynchronous. Although every effort is made to move database changes from the source system to the target system as fast as possible, there are no latency guarantees. Applications must not assume that changes to the primary database will be made to the backup database within a specific time frame. Even if a benchmark shows a consistent delay time, it can be affected by load changes, degradation, or failure anywhere in the infrastructure.

Figure 1. NonStop RDF configuration examples.



During replication, applications may access the backup database for browsing. In fact, some companies use the backup database to offload read-only access or for reporting functionality from the source system. Note that while NonStop RDF is running, business transactions in the backup database can be in a partially completed state. For example, when transferring $100 from checking to savings, there may be a window when the debit could be present in the backup database, but not the credit. The operator can make the backup database transactionally consistent by simply issuing the STOP UPDATE command with the TIMESTAMP parameter if the source system is still running, or by performing a NonStop RDF takeover if it is not.

When it has been determined that there is a failure of the source system, the operator issues a takeover command to ensure that the backup database is transactionally consistent and ready for use. The application on the target system can then open the backup database for read/write access and begin processing. Of course, there are many other activities that need to take place, such as client rerouting, network reconfiguration, relocation of personnel, and so on. Whereas automatic takeover might seem preferable, people make much better decisions than computers. What if the appearance of a primary site failure is only due to a communications glitch at the backup site? NonStop RDF is just part of a complete recovery plan. For more information on how to begin a recovery or continuity plan, including best practices, see the *Developing a business continuity plan* strategy brief at **www.hp.com/go/nonstopcontinuity**.

NonStop RDF configuration examples

Figure 1 shows some of the possible configurations for NonStop RDF software. Each of these configurations is discussed in the following sections.

Simplex

Description: The simplex configuration is the fastest and easiest to configure. A second NonStop server is connected via HP Expand networking software to the original server running the application, the primary database is copied to it, and NonStop RDF is started on both sides. The target system can be a system located at a different customer site, at a third-party hot site, or a development system.

Considerations: Is the system configuration sufficient to enable taking over application processing in the event of a takeover (if this is a requirement)? Is management willing to pay for a second system? If the target system is a development system, what happens to the programmers while it is acting as a production system?

Multiple duplicate sites

Description: This configuration is very similar to the simplex configuration except that the primary database is replicated to multiple targets. This architecture may be used if the customer has multiple physical sites that need low-latency browse access to the database or if the application is so read-heavy that the load needs to be split across more than 16 processors.

Considerations: The customer is gaining low-latency browse access or application scalability at the expense of the space needed by multiple copies of the same database. If replication to one or more of the duplicates is slowed or interrupted for any reason, the views could be inconsistent.

Triple contingency

Description: One key point that IT people often overlook is that the backup site could go down before or soon after the primary site. Triple contingency is meant to address application uptime for extremely critical business processes. During normal operation, database changes are sent from the source system to two target systems. If there is a site failure anywhere, a triple contingency takeover will route database changes from the new source system to the remaining target system. Additionally, any two sites could suffer an outage, and the third will be available to continue processing. When in this configuration, NonStop RDF has special commands built into it to enable switchover as transparently as possible.

Considerations: What other type of processing are the systems doing and will they be ready to take over the primary application? How much hardware is involved for the two target systems? NonStop TMF auxiliary audit trails cannot be used in this configuration, nor does it work with network NonStop TMF transactions.

Reciprocal/split workload and ring

Description: The reciprocal configuration is a more advanced and more productive version of the simplex configuration. With the simplex configuration, one system may sit idle until a failure occurs. With the reciprocal configuration, both systems are running their own applications, but sufficient capacity is available on each system to maintain the replicated database from the other system and to handle the workload that would be generated if the other system fails. For example, a customer has two different applications, such as customer service and inventory, that access different databases. Each application runs wholly on one system, and each application's database is replicated to the system that the other application runs on. Upon takeover, the surviving system runs both applications.

The ring is an extension of the reciprocal configuration in that each of the multiple systems running different applications is protecting another system. B protects A, C protects B, D protects C, and A protects D.

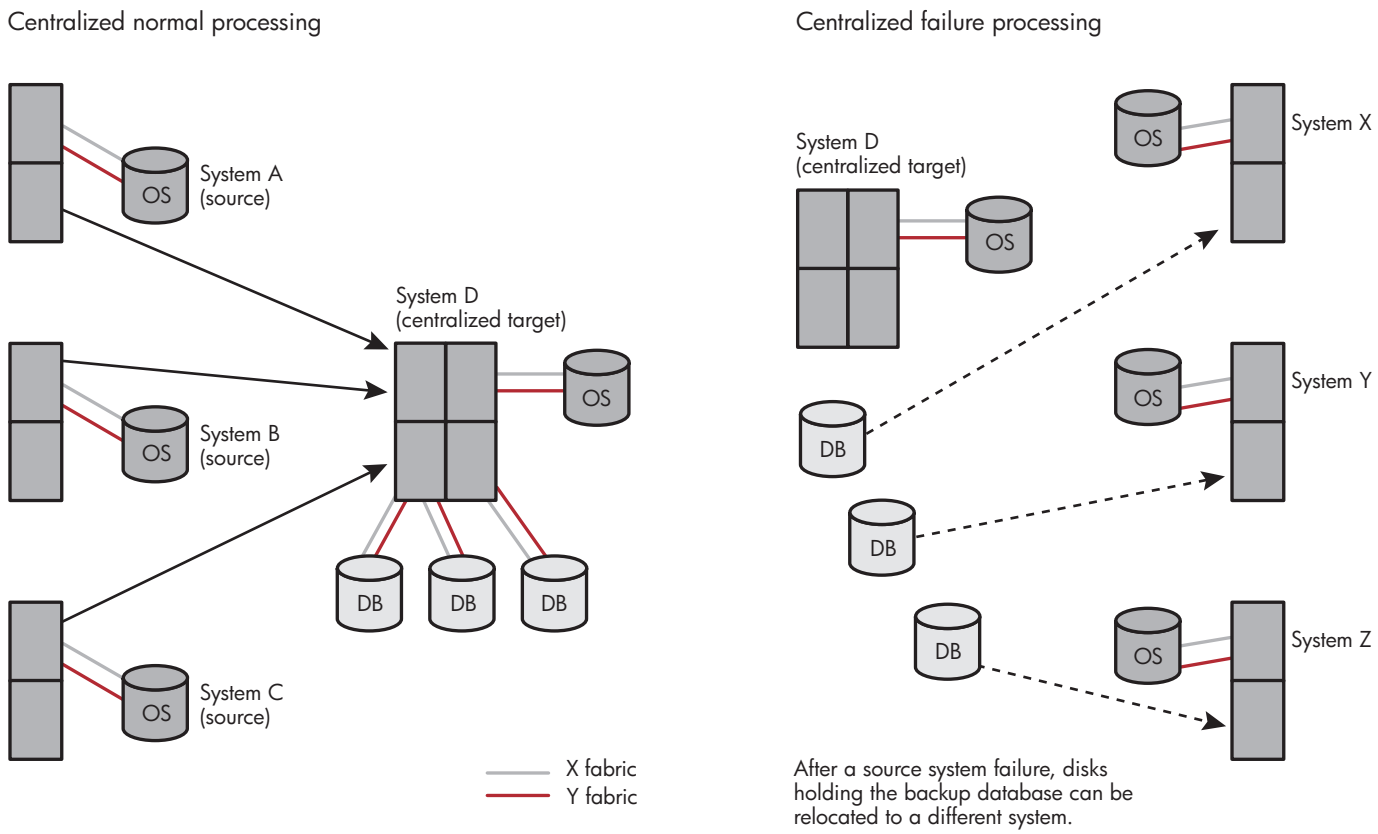
In addition to supporting different applications, both ring and reciprocal configurations can be used to support a split workload. In this configuration, the application database is logically split, and both it and the application are spread among the two reciprocal or multiple ring systems. Transactions to any primary database are replicated to the associated backup database on a different system.

During normal or takeover processing, transactions are always routed to the system that is maintaining the part of the database that the transaction modifies. When that system fails, an instance of the application on the system where the backup database resides opens the database for processing. Transactions must then be routed to that system.

Using the ACI Base24 application running on two systems as an example, each system might have one network up and one network down. Each system performs settlement only for the up network. Transactions made to the up network on one system are replicated to the down network on the other system. Upon failure of one system, the down network is brought up on the remaining system and that system performs settlement for both networks.

Considerations: A ring configuration is very much like multiple simplex configurations except that all systems are doing actual production work. For a customer with many applications, this may be one of the best ways to maintain continuous application availability at minimal total cost.

Figure 2. Centralized configuration for a managed services environment.



Centralized

Description: This configuration can take over processing for one or more failed sites or can be used for aggregating data for data mining. For example, if a customer wants to protect 50 remote nodes, rather than running NonStop RDF on 50 target nodes, one node is the centralized recipient for multiple NonStop RDF environments. Upon failure of a remote node, one or more application environments are bought up using the replicated databases. Remember that each primary database needs to be replicated into a different backup database.

A managed services organization or hot-site provider might use a special case of this configuration to act as a data vault for a large number of remote nodes belonging to different customers. The central system is tasked solely with safekeeping database information without being able to process it. Unlike a normal centralized configuration, each backup database is configured across exclusive disks. HP Safeguard security software can be used to ensure that one organization cannot access disks protecting another organization's information.

Upon failure of any of the primary systems vaulting to the centralized target system, the physical disks holding the database could be moved to another system preconfigured to accept the disks so that they can be brought into production. (See figure 2.)

Considerations: The customer may be left without any backup for multiple systems if this is the first system to fail.

Network transactions

The NonStop RDF/IMPX product supports applications where NonStop TMF transactions span multiple source systems (network NonStop TMF). During normal operation, changes made on each source system are replicated independently to the associated target systems. This means that the systems can be out of synchronization with each other while NonStop RDF is running. Using the STOP UPDATE command with the TIMESTAMP parameter is meaningless in a network environment because the system times will not be exactly in synchronization, and this can leave the databases transactionally inconsistent with each other. A rolling failure (individual communications lines or systems failing serially instead of all at once) can also cause the target systems to be out of synchronization with each other.

When one or more source systems fail, all processing must cut over to the target systems. That is, if a network transaction spans databases on more than one system on the source network, all of those databases must exist on systems on the target network. If the operator chooses to execute a takeover command for any system on the source network that supports network transactions, then it must be done for all the systems, or the databases will not be transactionally consistent. When a network takeover command is issued, NonStop RDF will first resolve all indeterminate local transactions followed by all indeterminate network transactions. NonStop RDF will finish by backing out any completed transactions that were built upon network transactions that were backed out. This will ensure that the database is transactionally consistent from the earliest failure of any of the networked systems.

Live-live versus split workload configurations

A live-live configuration has multiple systems running one or more applications, which are updating the same records in their local copy of the database and replicating those changes to the backup database. With unsynchronized live-live replication, the same record may be updated on both the primary and backup databases, and then replicated to the opposite database. Explicit attention must be paid to collision prevention or detection to prevent the databases from getting out of synchronization. The following example illustrates why collision prevention or detection is important.

- System A
  - Checking $100
  - Savings $100
- System B
  - Checking $100
  - Savings $100

On system A, $50 is transferred from checking to savings. On system B, $30 is transferred from savings to checking. Assume that this is done at exactly the same time or at different times while replication is suspended.

- System A
  - Checking $100 – $50 = $50
  - Savings $100 + $50 = $150
- System B
  - Checking $100 + $30 = $130
  - Savings $100 – $30 = $70

Although locally consistent, each system has different views of the customer's balance, and the final state of the databases on both systems depends on which order the data is replicated. If future or cross-user transactions build off of these transactions (for example, if these are stock buy and sell orders), both databases could wildly diverge from each other in a very short time period. If you do not fully understand or cannot implement an internode synchronization or a collision detection scheme, a split workload approach may be more appropriate.

In a split workload configuration, the database is logically divided across the multiple systems so that the same record does not appear on more than one system. Some ways of doing this are by account name, number, or geographical location. Transactions must be routed to the proper system for processing. An application router or front-end message switch can be used for this purpose. Transaction routers can be configured to look into the actual message, and based on one or more fields, route the transaction to the appropriate system.

If the record is in the database on a different system than the one that the request comes into, it is sent to the proper system to make a local update. Now there is only one database of record, although it can reside across multiple systems.

There could also be files that need to be shared by all of the systems running the application. If these are read-only files, perhaps updated in a batch mode, the entire file set can be replicated after changes are made to them.

Figure 3. NonStop RDF data path.

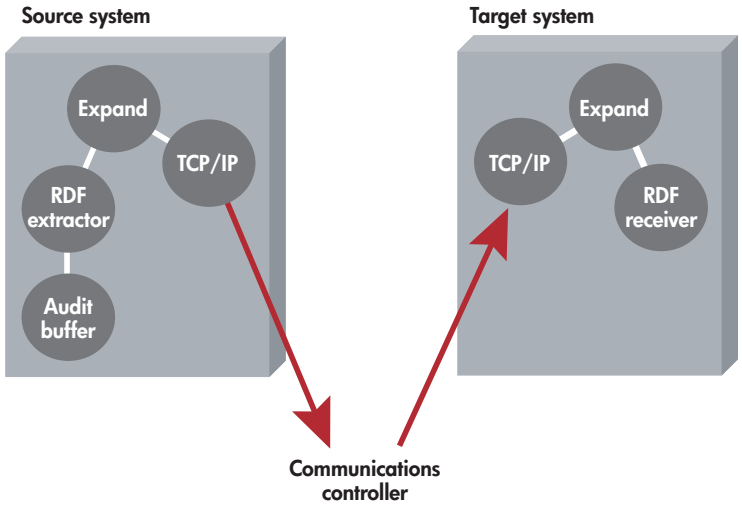
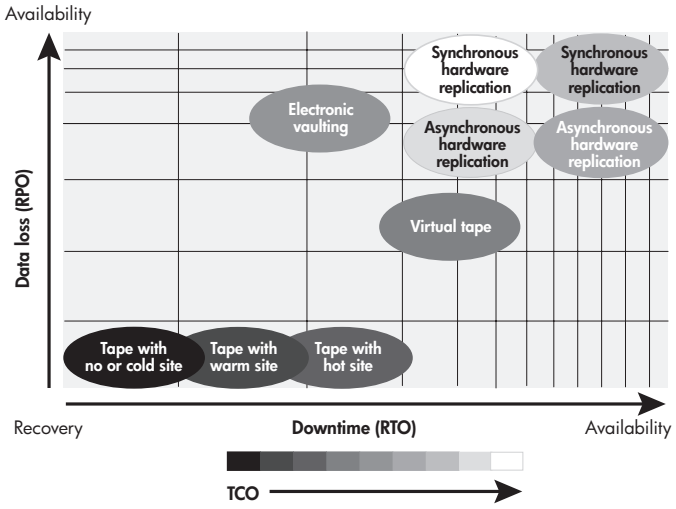


Figure 4. Separating RTO and RPO.



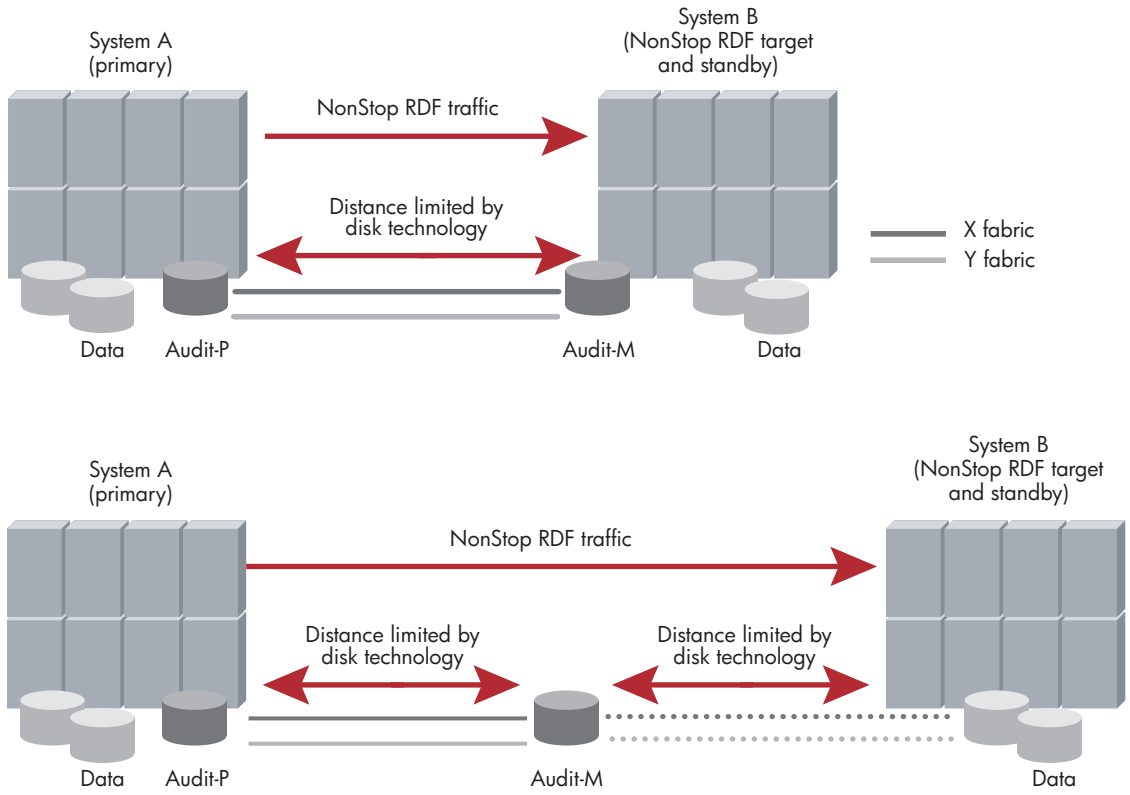
Minimizing your recovery point objective

NonStop RDF software's exceptional flexibility can meet the needs of widely varying business requirements. However, one of the tradeoffs in using asynchronous replication is that transactions committed on the source system can be lost if that system fails before the transactions are sent to the target system. Because of latency from the time that a transaction is committed on source system's primary database until it is transferred via Expand networking software to the NonStop RDF "image trail" on the target system (see figure 3), it is possible to lose a small number of in-flight transactions. The problem is exacerbated if NonStop RDF is not allocated sufficient processing or communications resources.

As shown in figure 4, recovery to availability is a continuum. Not all business processes need to have 100 percent availability. There are two goals to keep in mind for each business process: the recovery time objective (RTO) and the recovery point objective (RPO). RTO defines the tolerable maximum length of time that a business process can be unavailable, whereas RPO defines how much work in progress can be lost.

For NonStop system customers that need zero RPO, HP offers NonStop RDF/ZLT software, an add-on to the NonStop RDF/IMPX product. NonStop RDF/ZLT software is supported by the appropriate remote disk infrastructure, which locates half of the mirrored NonStop TMF audit trail volumes remotely from the source system.

Figure 5. NonStop RDF/ZLT target system as standby configuration.



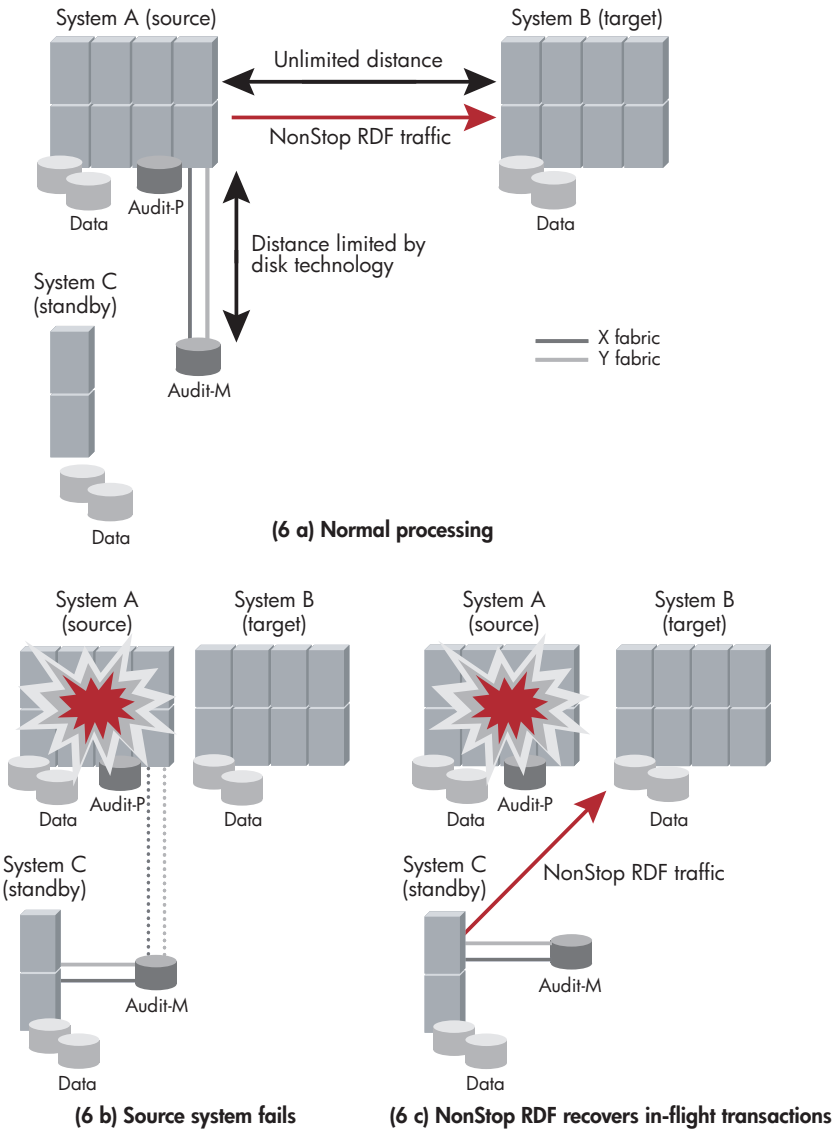
Zero lost transactions solution

NonStop RDF/ZLT software is part of a complete solution that also includes hardware, infrastructure, and procedures, which are selected, configured, and installed by HP Services. The hardware and infrastructure consist of one or more remote disk mirrors and the communications infrastructure to support them. The NonStop RDF/ZLT *standby system* refers to the system that the remote disk mirrors will be connected to after a failure of the source system. The standby system can be the target system, a third system, or even the source system for testing purposes. The distance from the disks to the source and standby systems is dictated by the disk technology used in the configuration, but the source and target systems can be any distance from each other. In any configuration, the tradeoff is slightly longer recovery time versus no loss of committed transactions. Other than the physical connections to the additional disks, no changes are required to the application, source, standby, or target systems. Transaction processing already running on any of the systems can continue with little to no impact before a system failure, and only applications on the source system are affected when it fails.

The target system as the NonStop RDF/ZLT standby system

The disk mirrors containing the NonStop TMF audit trail are located remotely from the source system and are connected to both the source and target systems but controlled by the source system (figure 5). The distance between the source and target/standby systems can be twice as long as the distance between the mirrors and each system because the remote mirrors can be located in between the systems. After failure of the source system, control of the disks is switched to the target/standby system. With a command to NonStop RDF/ZLT on the target system, the records not already transmitted to the target system are read from the audit mirrors and applied to the database. Note that if the remote disk mirror fails before the complete failure of the primary system, NonStop RDF/ZLT falls back to asynchronous mode. For most customers this is not an issue, but will be addressed in the future.

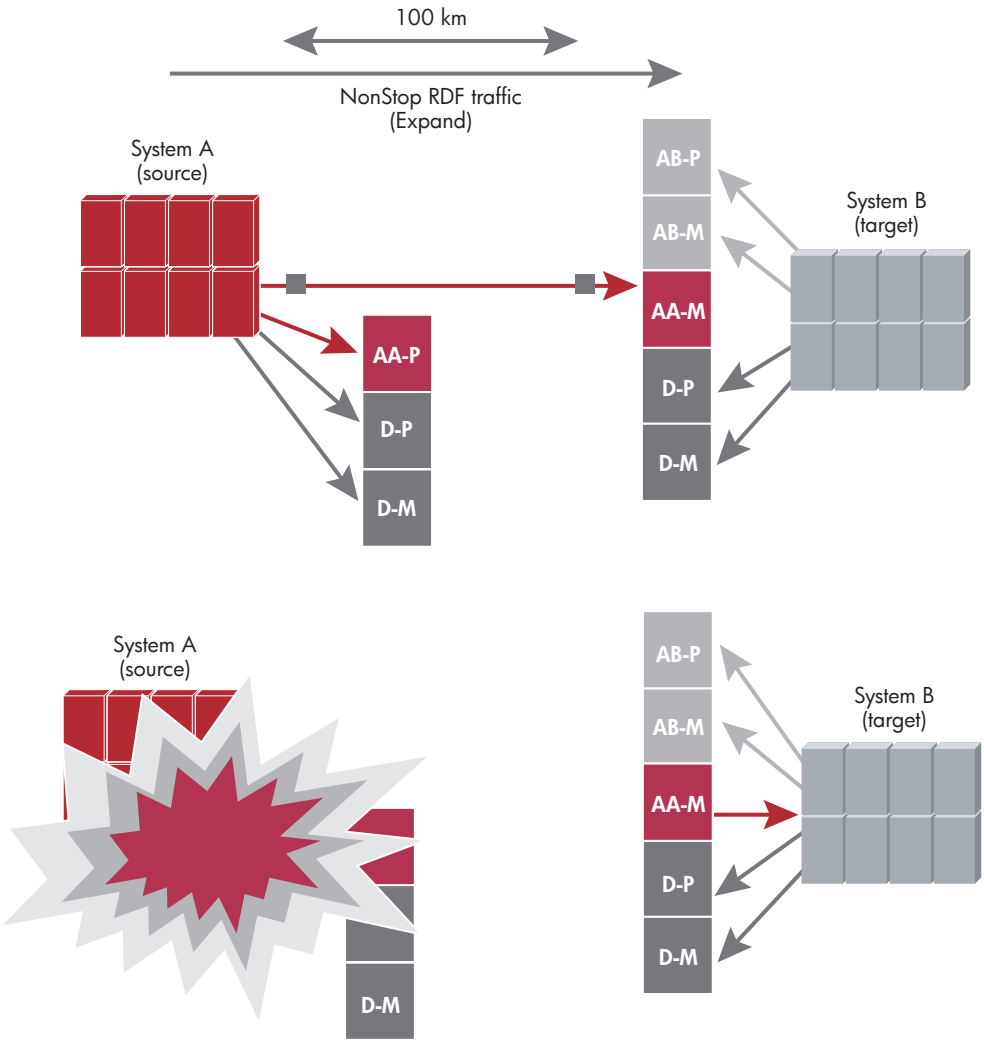
Figure 6. NonStop RDF/ZLT third system as standby configuration.



A third system as the NonStop RDF/ZLT standby system

The disk mirrors containing the NonStop TMF audit trail are located remotely from the source system and are connected to both the source and standby systems but controlled by the source system (figure 6 a). After a failure of the source system, control of the remote disk mirrors containing the NonStop TMF audit trails is switched to a third system (figure 6 b). With one command, NonStop RDF/ZLT on the target system makes a connection to its components on the standby system, locates the proper location in the NonStop TMF audit trails, and applies any missed transactions to the database (figure 6 c). As before, no changes are required to the source system, target system, or application, and the source and target systems can be any distance from each other.

Figure 7. NonStop RDF/ZLT configuration with a single StorageWorks XP system at each site.

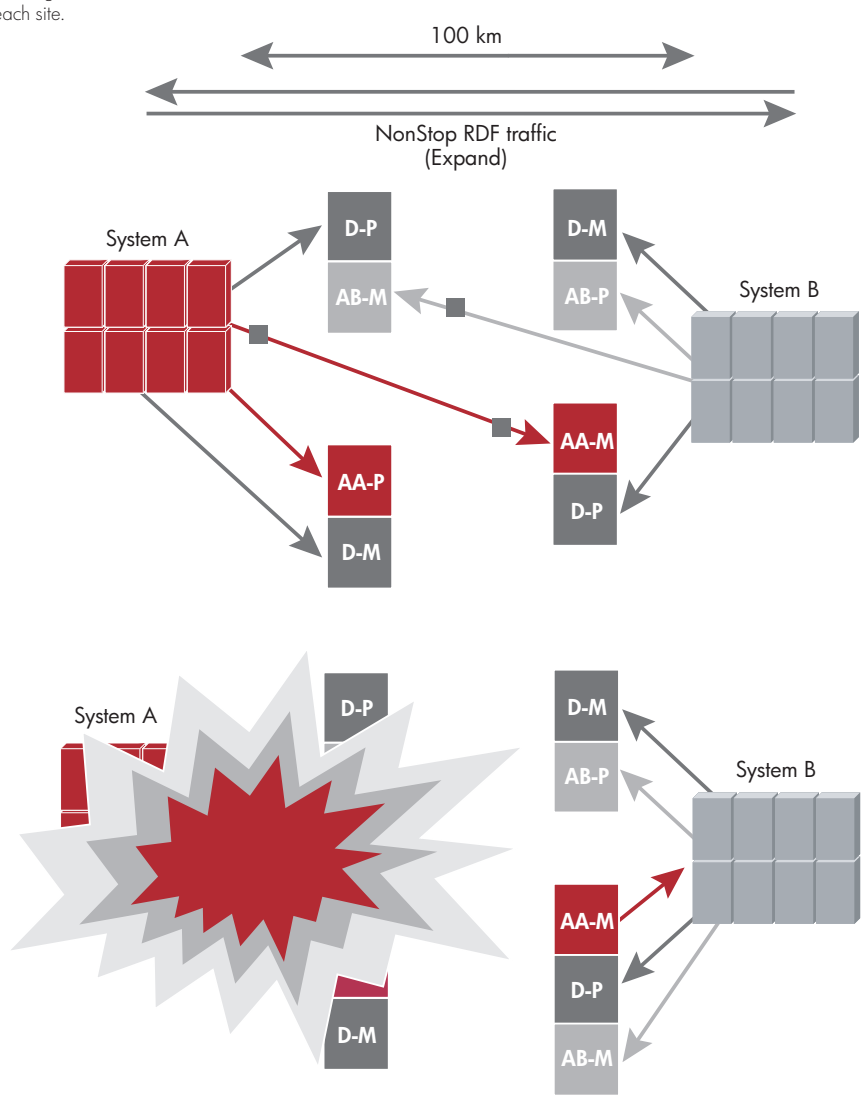


NonStop RDF/ZLT and HP StorageWorks XP enterprise storage system configurations

Any scenario mentioned earlier can be implemented with HP StorageWorks XP enterprise storage systems. Figure 7 shows an example of how one enterprise storage system might be configured at each site with NonStop RDF/ZLT in a simplex configuration from the source system to the target system. The data volumes are located locally to each system, and only one system's audit trail mirror is located remotely. Figure 8 shows how dual StorageWorks XP systems with reciprocal NonStop RDF/ZLT might be configured. In this example, each system is protecting the other, so each system's audit trail is located at the other system's location. When one site fails, the remaining system not only takes over processing for the failed system, but also needs to re-create its audit trail mirrors locally.

In addition to the StorageWorks XP enterprise storage system configurations shown in the figures, there are other configurations available to meet customers' needs; for example, simplex replication with two StorageWorks XP systems at each site or reciprocal replication with one StorageWorks XP system at each site. An HP Services representative can help you determine which configuration is suited to your business continuity needs.

Figure 8. Reciprocal NonStop RDF/ZLT configuration with dual StorageWorks XP systems at each site.



Online software migration using NonStop RDF

Hardware such as processors, I/O cabinets, and disk drives can be added to a NonStop system without taking your application down, but software upgrades, whether supplied by HP, ISVs (independent software vendors), or customers, may not be replaceable online for many reasons.

NonStop RDF software can be used to perform a "rolling upgrade" with little to no application downtime. Essentially, application processing is moved off of one of the systems in an application domain so that it can be upgraded to a new software release. Once upgraded, processing can be moved back to the upgraded system for testing. If application verification is successful, application processing is then migrated off of another system so that it can also be upgraded. If application verification is not successful, fallback to a system that has not yet been upgraded can be done with no loss of data.

Figure 9. Performing a rolling upgrade using NonStop RDF.

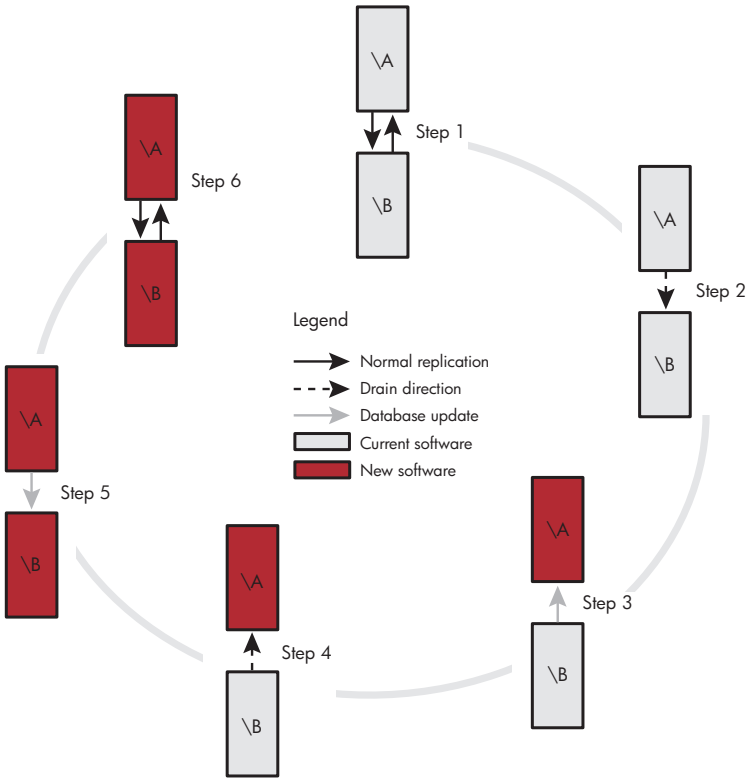


Figure 9 shows the steps to be followed to perform a rolling upgrade of two systems that are in a split workload configuration. Transaction processing must be stopped on one of the systems, and the replication queue must be "drained" to the opposite system before routing transactions to it. Once a system is upgraded, its database must be updated with all of the changes made to the opposite system while it was down.

In this example, transactions are stopped on system A, and any final database changes are allowed to drain to system B. If the operating system on system A has to be brought down for any reason, such as upgrading, replication from system B to system A needs to stop temporarily. If the operating system is not brought down, replication can continue in that direction. Once system A is upgraded and tested, its database needs to be updated with the changes that have been made on system B while it was down. Once system A is current, processing is moved back to it. The same steps are followed for system B.

For more information, go to www.hp.com/go/nonstopcontinuity.

5982-6616EN, 07/2004

invent