



“Breaking the Four 9s Barrier” An Informational Series on Enterprise Computing

**As Seen in *The Connection*, An ITUG Publication
September 2002 – December 2003**

About the Authors:

Dr. Bill Highleyman, Paul J. Holenstein, and Dr. Bruce Holenstein, have a combined experience of over 90 years in the implementation of fault-tolerant, highly available computing systems. This experience ranges from the early days of custom redundant systems to today’s fault-tolerant offerings from HP (NonStop) and Stratus.

Series Topics:

[Breaking the Four 9s Barrier, Part 6 - RPO and RTO \(12/03\)](#)
[Breaking the Four 9s Barrier, Part 5 - The Ultimate Architecture \(9/03\)](#)
[Breaking the Four 9s Barrier, Part 4 - Facts of Life \(6/03\)](#)
[Breaking the Four 9s Barrier, Part 3 - Sync Replication \(4/03\)](#)
[Breaking the Four 9s Barrier, Part 2 - System Splitting \(2/03\)](#)
[Breaking the Four 9s Barrier, Part 1 - The 9s Game \(11/02\)](#)
[Breaking the Four 9s Barrier, Part 0 - Intro/About the Authors \(9/02\)](#)

Gravic, Inc.

Shadowbase Products Group

17 General Warren Blvd.

Malvern, PA 19355

610-647-6250

www.ShadowbaseSoftware.com

Availability (Part 6) - RPO and RTO

Paul J. Holenstein
Dr. Bill Highleyman
December 22, 2003

In the previous parts of this series on availability¹, we have discussed how system availability can be significantly enhanced by replicating a system or by breaking it up into several smaller independent cooperating nodes. We have discussed to some extent the various failure modes of such distributed systems, the immediate recovery from these failures, and the ultimate restoration of the full system to service following the repair and recovery of a failed node.

This Part 6 explores two important questions relative to failures in a distributed system that must be considered when deciding how to replicate or split a system: an organization's tolerance to lost data and its tolerance to downtime.

RTO and RPO

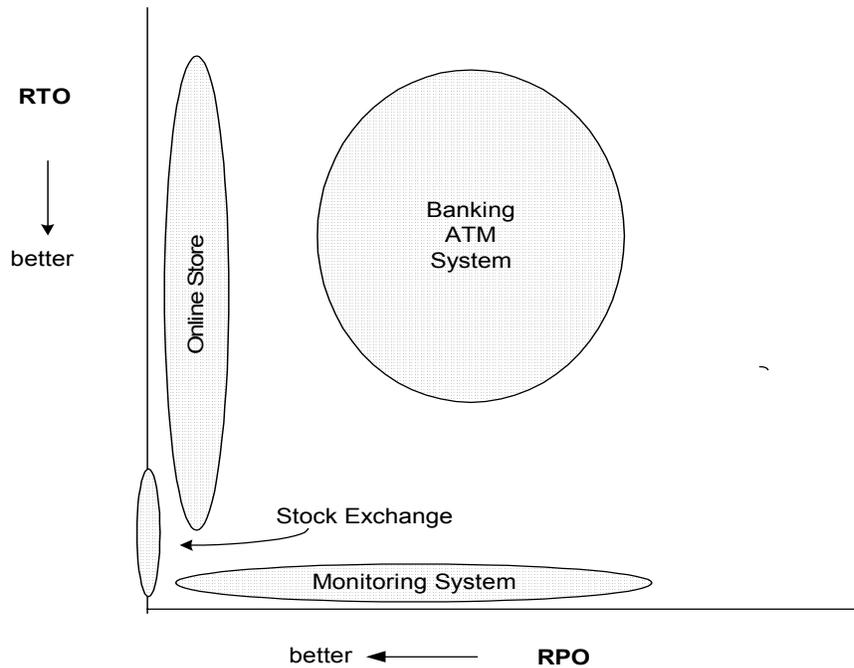
When considering the true and total costs of a node failure in a distributed system, two important factors are the time that the users of the system are denied service and the amount of data, if any, that may have been lost due to the failure. When deciding how nodes in a distributed system will back each other up, the amount of tolerance that a business has to recovery time and lost data should be established as a pair of objectives that the system must achieve in the event of a primary system loss. This tolerance can be established as Recovery Point and Recovery Time Objectives:²

- The Recovery Point Objective, or *RPO*, is a measure of how much data loss due to a node failure is acceptable to the business. A large RPO means that the business can tolerate a great deal of lost data.
- The Recovery Time Objective, or *RTO*, is a measure of the users' tolerance to down time. A large RTO means that users can tolerate extensive down time.

¹ Highleyman, W., Holenstein, P., Holenstein, B., *Availability (Parts 1 – 5)*, The Connection, November, 2002 through December, 2003.

² LaPedis, R.; "Will Enterprise Storage Replace NonStop RDF," The Connection, Volume 23, Issue 6; November/December, 2002.

These two objectives are not closely related – they may both be almost zero, they both may be large, or one may be small but the other large.³ Various examples of the needs of different applications are shown in Figure 1. For instance, a stock exchange trading system must be brought back to life very quickly and can lose no data. Since the price of the next trade depends upon the previous trade, the loss of a trade will make all subsequent transactions wrong. In this case, the RTO may be measured as a few minutes or less, but the RPO must be zero.



**Example Objectives
Figure 1**

On the other hand, a critical monitoring system such as those used by power grids, nuclear facilities, or hospitals for monitoring patients must have a very small RTO, but the RPO may be large. In these systems, monitoring must be as continuous as possible; but the data collected becomes stale very quickly. Thus, if data is lost during an outage (large RPO), this perhaps impacts historical trends; but no critical functions are lost. However, an outage must end as quickly as possible so that critical monitoring can continue. Therefore, a very small RTO is required.

A Web-based store must have an RPO close to zero (the company does not wish to lose any sales or, even worse, acknowledge a sale to a customer and then not deliver the

³ LaPedis, R.; "RTO and RPO Not Tightly Coupled," Disaster Recovery Journal; Summer, 2002.

product). However, if shipping and billing are delayed by even a day, there is often no serious consequence, thus relaxing the RTO for this part of the application.

A bank's ATM system is even less critical. If an ATM is down, the customer, although aggravated, will find another one. If an ATM transaction is lost, a customer's account may be inaccurate until the next day when the ATM logs are used to verify and adjust customer accounts. Thus, neither RPO nor RTO need to be small.

Once a company decides what RPO and RTO are applicable to an application, the method for backup and recovery of that application becomes much more evident. The remainder of this paper relates various methods for backup and recovery to their RPO and RTO characteristics.

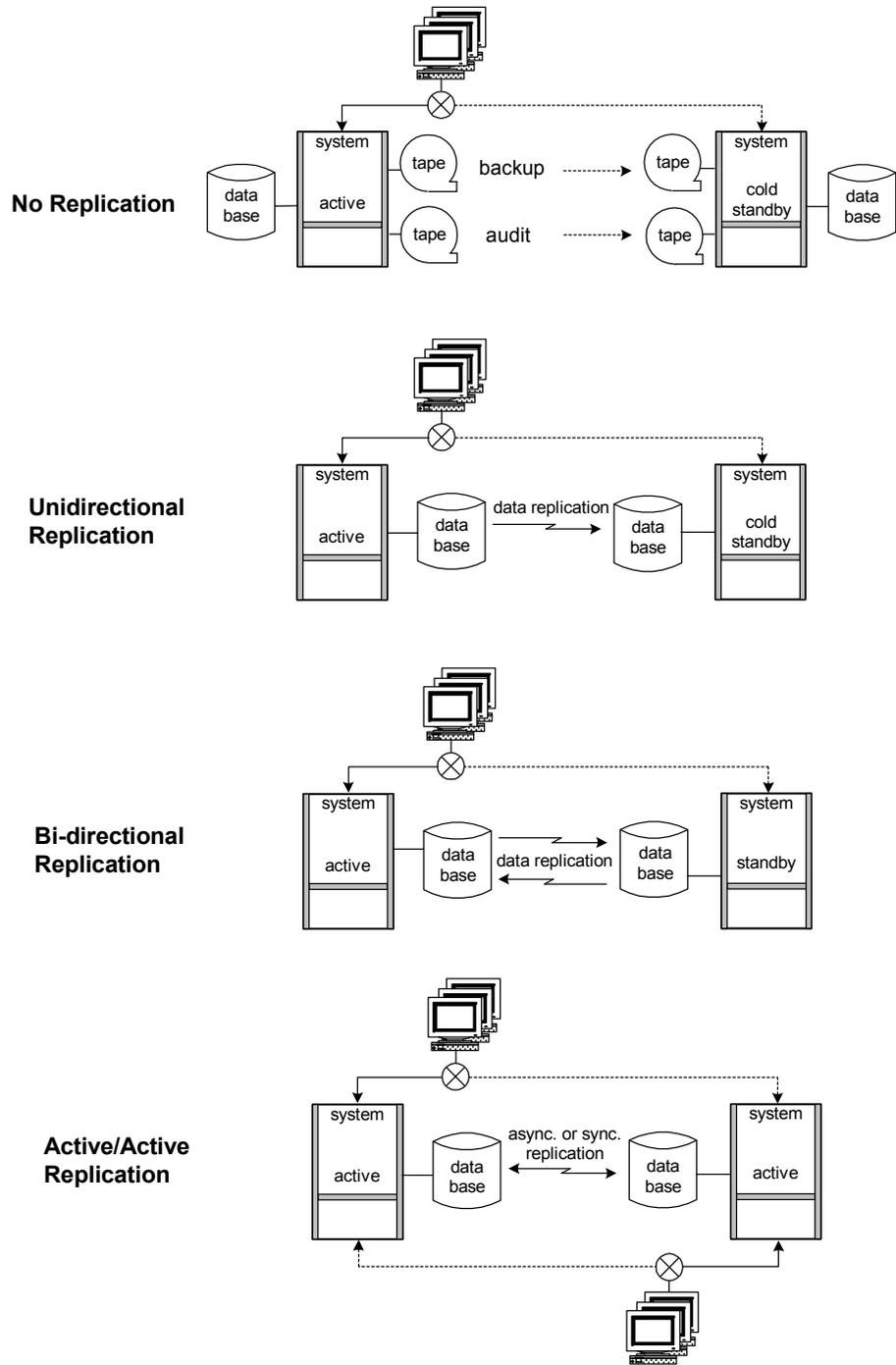
Replicating the Application Data

Recovering services to users on a failed node in a distributed system requires that a current copy of the application database be available on at least one other node accessible to all nodes in the system, and that users be switched to a surviving node. In order for the copy of the database to be current, changes made to the primary database by the applications must be passed, or *replicated*, to the backup database. Several products are available today that perform near-real time data replication.

Various data replication techniques are shown in Figure 2 and are described next. To simplify the discussion, we will first consider replication between two systems. Multi-node systems are considered later.

No Replication

The use of tape to back up a system requires no data replication facility. Such backup procedures tend to have long recovery times and may have a probability of significant data loss.



**Replication Methods
Figure 2**

Basic tape backup procedures involve periodically taking a snapshot of the database and writing it to tape. In the event of a failure of the primary system, the latest snapshot must be loaded onto the backup system before the backup site can take over operations. All database updates between the last backup and the time of the failure are lost. Thus, not only does this backup method have a potentially long recovery time (large RTO), but it is also susceptible to a large amount of data loss (large RPO).

In some systems, a continuing audit trail of database changes made since the last snapshot is also written to tape. Then, should a primary outage occur, the audit trail tapes written since the last snapshot are applied to provide a reasonably up-to-date database for the backup system.

Playing back the audit trail tapes may extend the recovery time (RTO) if snapshot tapes have to be loaded first, but doing so can dramatically reduce the amount of lost data (RPO).

Unidirectional Replication

With unidirectional replication, the application is only running on the active system. The backup system may use the replicated data for read-only purposes but cannot be actively updating that data.

The simplest form of unidirectional data replication feeds a backup database in near real time with updates that are being made to the active database. Except for data replication activity, the backup system has no participation in the application. There are no application processes running on the backup system, though the system may be used for other work. Thus, it is called a cold standby.

Alternatively, applications may be running with files opened for read-only (a warm standby) or with files fully opened (a hot standby).

Bi-Directional Replication

A primary system with a hot backup system may also be configured for bi-directional replication. It will behave exactly as if it were a unidirectionally replicated hot standby system as described above, except that now it is rather straightforward to switch active/standby roles by simply switching users to the standby system, with virtually no impact on the users. This can be particularly advantageous for testing system recovery and keeping operations personnel current in switchover procedures via periodic training.

Following such a planned switchover, the old primary system now acting as the standby system will continue to have its database kept in synchronism with the new

active system's database because of the bi-directional replication. A return to the original configuration can be accomplished at any time by simply switching users.

Active/Active Replication

A major advantage of bi-directional replication is that the full capacity of both systems in the network is available for application processing. Neither system is dedicated as a dormant backup. Thus, the reliability and disaster tolerance afforded by replicating systems can be achieved with a much smaller complement of equipment if all systems cooperate actively in the application. Each system can be supporting its own community of users, be making its own updates to its local database, and be replicating these updates to the remote database. Such distributed applications are called *active/active* applications.

However, there are some serious problems with active/active replication. One is ping-ponging, or the replication by a system of an update back to that system from which it was received. Provisions must be made to avoid ping-ponging.⁴

Another problem is data collisions. A data collision occurs when two users update the same data item on different copies of a database at nearly the same time. Since the same data item is now being replicated with different values, typically only data content or application-knowledgeable reconciliation procedures are capable of returning the database copies to a consistent state. Solutions that resolve data collisions or that avoid them have been discussed in earlier parts of this series.

Partitioned Active/Active Replication

Data collisions can be avoided by logically *partitioning* the database and the processing activity so that each system will still have a full copy of the database, but the users at each system will update only the database partition that their system owns. For instance, it may be that only the sales office that services a customer can make updates to that customer's records. These updates can then be replicated to the other system with no fear of data collisions.

Asynchronous Active/Active Replication

If data collisions are not deemed to be a serious problem, then both systems can be actively processing all transactions; and no partitioning is required. This may be the case, for instance, if users are geographically segregated and if the risk of data collisions is small. Alternatively, if the cost of resolving data collisions is small, or if data collisions

⁴ Strickler, G.; et al.; "*Bi-directional Database Replication Scheme for Controlling Ping-Ponging*," United States Patent 6,122,630; Sept. 19, 2000.

can be resolved automatically, asynchronous active/active replication may also be appropriate.

Synchronous Active/Active Replication

In many applications, partitioning will not work or may not be feasible. Any transaction may possibly affect any data item in the database, and resolving data collisions may simply be too difficult or require too much manual intervention to use asynchronous active/active replication. Both systems must cooperate to ensure that a transaction they own will not cause a data collision with a transaction owned by the other system.

This can be accomplished by ensuring that a transaction can lock throughout the network all of the instances of a data item which are affected by that transaction. It can then update all data item instances before releasing the locks. We call this *synchronous replication*.

One way to implement synchronous replication is to use distributed transactions to start a single transaction that spans both systems. Then updates to all data item instances are guaranteed, or else none will be made (the transaction aborts).

This technique is best applicable to closely located systems with small transactions because of communication delays.⁵ However, geographically dispersed systems are a requirement for disaster recovery. For these systems, data replication with coordinated commits offers a satisfactory alternative.⁶ With this technique, the data replicator will start independent transactions on each system. Updates are replicated asynchronously without slowing down the application, but the commits of these independent transactions are coordinated by the data replicator. Neither transaction is committed unless both are guaranteed to commit.

Recovery Time

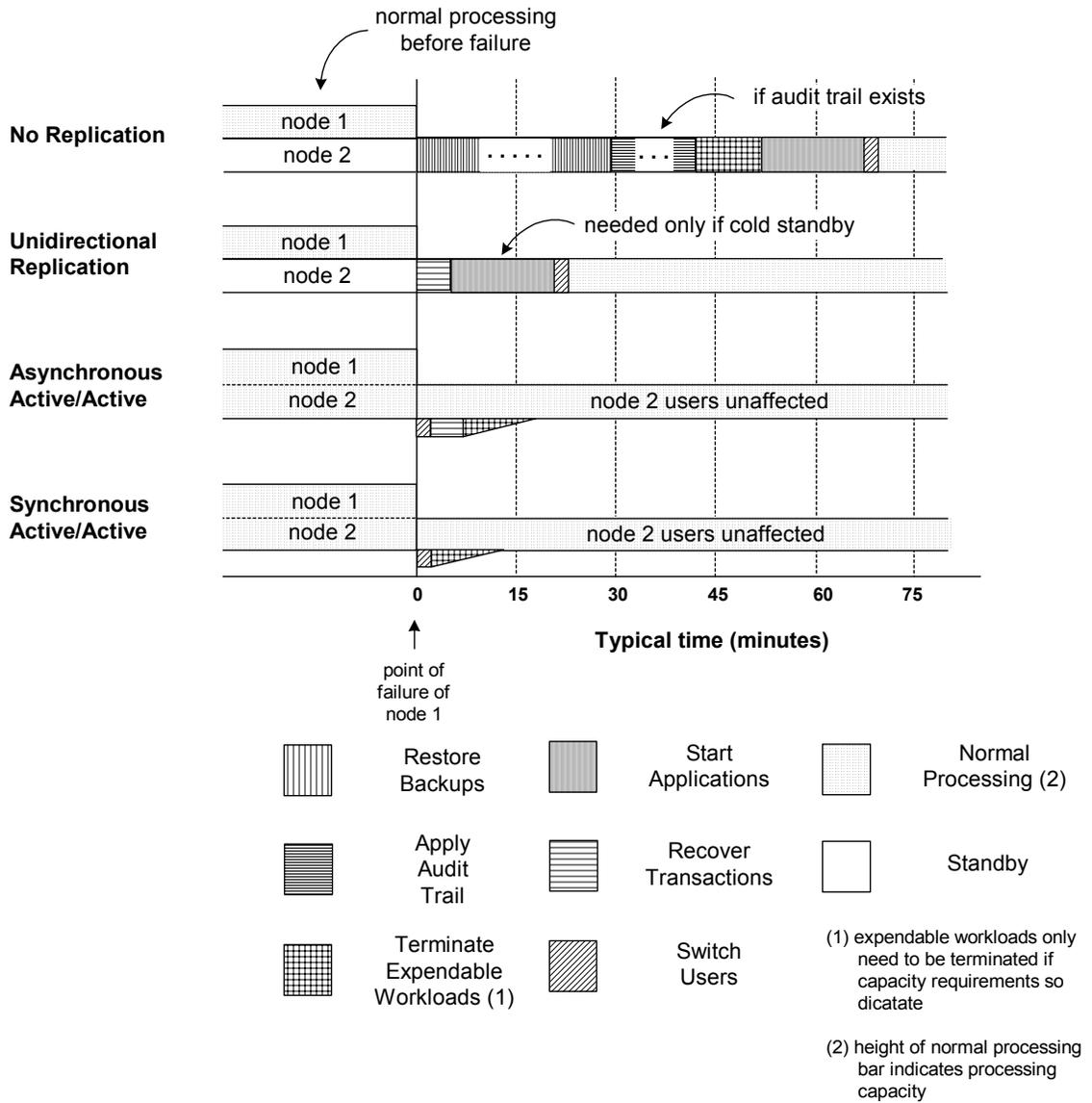
Recovery time, and hence RTO, can be vastly different for these recovery strategies, as shown in Figure 3.

No Replication

⁵ Highleyman, W., Holenstein, P., “*Availability Part 3 – Synchronous Replication*,” The Connection, Volume 24, No. 2; March/April, 2003.

⁶ Holenstein, B. D.; et al.; “*Collision Avoidance in Bi-directional Database Replication*,” United States Patent 6,662,196; Dec. 18, 2003.

If data replication is not used, the last snapshot of the database must be loaded onto the backup site. At this point, non-critical workload can be shed; and the applications to be recovered are started. Finally, users can be switched from the off-line system to the backup system; and normal activity can continue.



Typical Recovery Times
Figure 3

The reconstruction of the database can take several hours to several days, depending upon its size. Oftentimes, expendable workload shedding, application startup, and user transfer can be done during database recovery.

A major problem with the simple tape backup procedure discussed above is that there can be hours or days of data updates that are lost from the time of the last snapshot to the time of the system failure. This results in a very large RPO.

The RPO can be significantly reduced if provisions have been made for audit trail tapes which record all updates made to the database since the last backup or snapshot. However, the use of audit trail tapes has the effect of increasing the recovery time, or RTO. Not only must the database be restored, but now the audit trail tapes must be re-played to bring the database more up-to-date.

Unidirectional Replication

If data is being replicated to a cold standby, and if the standby must take over processing, then the applications must be started and the users switched over before service can be restored. Application startup generally requires many minutes before users can reinitiate their sessions and continue their activities. In addition, incomplete transactions must be purged. This may take a few more minutes.

However, if data is being replicated to a warm or hot standby, then all that is required is for incomplete transactions to be recovered or aborted and for the users to be switched over. Also, if the backup is a warm standby, files must be reopened for full access.

Bi-Directional Replication

So far as recovery time is concerned, a hot standby bi-directional replication system recovers in the same recovery time as the unidirectional hot standby system described above.

Active/Active Replication

The recovery from a system failure when using active/active replication simply requires that downed users be switched over to the surviving system and transactions be recovered. It may be desirable to shed some load by stopping non-critical applications in order to provide enough capacity for all users.

Furthermore, if synchronous active/active replication is being used, no transactions need to be recovered since any transactions being processed by the failed system at the time of failure will be aborted.

Data Loss

RPO is the goal for the maximum amount of data that may be lost due to a failure. The amount of data that may be lost varies dramatically with these techniques.

No Replication

If no audit trail tapes are used, all data since the last backup is lost. This can be hours or even days worth of data.

The use of audit trail tapes can significantly reduce data loss.

Asynchronous Replication

With asynchronous data replication, the source application does not wait for the data to be safely stored and/or applied at the target system. The interval between the time that an update is applied to the source database and the time that it is applied to the target database is known as *replication latency*. Replication latency is the time that replicated data is in the replication pipeline.

In the event of an outage, data in the replication pipeline may be lost. This typically represents several seconds of database updates, providing that updates are sent to the target system as soon as they have been applied at the source system.

Synchronous Replication

Synchronous replication not only avoids data collisions, but it also guarantees that no data is lost. The source application does not commit a transaction until it is assured that the transaction will complete on the other system. If it is unable to commit, then the transaction is aborted at the other system. Therefore, RPO is zero.

Recovery Strategies

If either the primary or backup system should fail, or if a network fault should isolate a system, then the failed system's database will become rapidly out-of-date as processing activity continues at the surviving system. When the failed system is returned to service, its database must be brought to the current state.

No Replication

If magnetic tape is being used to back up a system, then the database recovery of the failed system is straightforward. The latest tape copy of the database and any change tapes are simply loaded onto the recovered system, and operation proceeds as usual.

Unidirectional and Bi-Directional Replication

Asynchronous Replication

If data is being replicated to a cold, warm, or hot standby, and if data replication is asynchronous, then no special action need be taken. During the period of failure, the queue of transactions to be replicated simply grows at the source system.

When the failed system (or the interconnecting network, whichever caused the failure) is returned to service, asynchronous replication continues; and the queue to the downed system is drained. When the queue size falls below a level considered to be normal, the failed system can be considered to be recovered.

Synchronous Replication

If synchronous replication is being used, then the active system (which now may be the original backup system if the primary system failed) must switch to asynchronous replication and queue new transactions for later replication.

When the failed system is returned to service, the queue must be drained. The saved transactions are replicated to the system being recovered, at which time synchronous replication may be restored.

Active/Active Replication

The recovery of an active/active system node outage depends upon the nature of the failure.

System Failure

If both systems are processing transactions in an active/active configuration, and one system fails, then the affected users may be switched to the surviving system and full functionality continued (within the capacity, of course, of the surviving system). Operation and recovery are as described above with respect to unidirectional replication, with the surviving system queuing transactions that must be posted to the failed system.

Network Failure

If the network connecting the two systems should fail during bi-directional replication, then both systems are capable of continuing independent processing. However, in the general case, this can lead to extensive data collisions if the network

outage is lengthy. This is often referred to as the “split brain” problem. There are several options for continuing operation in this case:

- a) If the database is partitioned so that any data item can be modified only by users at one system, then each system can continue to function independently. Each will queue its transactions to the other system for later replication when the network becomes operational.
- b) If the application is not partitioned, but data collisions can be tolerated, then each system can continue to function and to queue its transactions for later delivery, with recovery proceeding as described previously. Data collisions occurring during the outage must be detected and resolved during recovery.
- c) If the application is not partitioned, and if data collisions are not acceptable, then all users can be switched to one of the systems. The isolated system is, in effect, down and is recovered as described previously when the network becomes available.

Comparison Summary

Recovery times (RTO), data loss (RPO), and the possibility of data collisions for these various techniques are summarized in Figure 4. As can be seen, if replication is not used, both RTO and RPO can be very high – it may take hours to days to recover; and hours to days of data may potentially be lost.

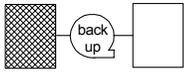
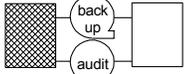
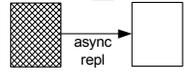
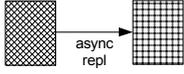
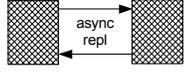
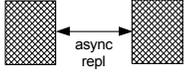
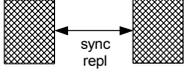
If replication of any kind is used, RTO is reduced dramatically to minutes, seconds, or even to near-zero if active/active replication is used (depending upon the time that it takes to switch users on the failed system to the active system). Likewise, the amount of data that may be lost, as measured by RPO, may be as little as those changes which occurred in the few seconds before failure or, in the case of synchronous replication, may even be reduced to zero.

Even when replication is used, the RPO and RTO that can be achieved is very dependent upon the replication technology that is used. This is illustrated in a general way in Figure 5.

As shown in Figure 5, synchronous replication achieves a zero RPO. That is, no data is lost when a system fails. Any transaction in process at the time of the failure is aborted. Furthermore, if the application is configured as an active/active application, RTO is also zero for all users but those on the failed system. Even the users on the failed system can be quickly restored to service by switching them to a surviving system.

The amount of data which may be lost with asynchronous replication is a direct function of the replication latency of the replication channel. If replication is substantially

process-to-process (i.e., changes are extracted directly from the source database and applied directly to the target database with no intermediate queuing), replication latency can be quite small (sub-second) and RPO can be driven fairly close to zero. To the extent that there are queuing points in the replicator or that there are intermediate disk storage points, replication latency will be longer and RPO higher. Typical queuing points occur at the communication channel as well as at processes that are reading source data or are updating target data.

Type	Recovery Time (RTO)	Data Loss (RPO)	Data Collisions
 No Replication, Periodic Backup	hours to days	hours to days	no
 No Replication, with Audit Trail	hours to days	seconds to hours	no
 Unidirectional Cold Standby	minutes	< 1 second	no
 Uni/Bi-directional Warm/Hot Standby	seconds	< 1 second	no
 Partitioned Active/Active	seconds	< 1 second	no
 Asynchronous Active/Active	none	< 1 second	yes
 Synchronous Active/Active	none	none	no

 cold standby

 hot standby

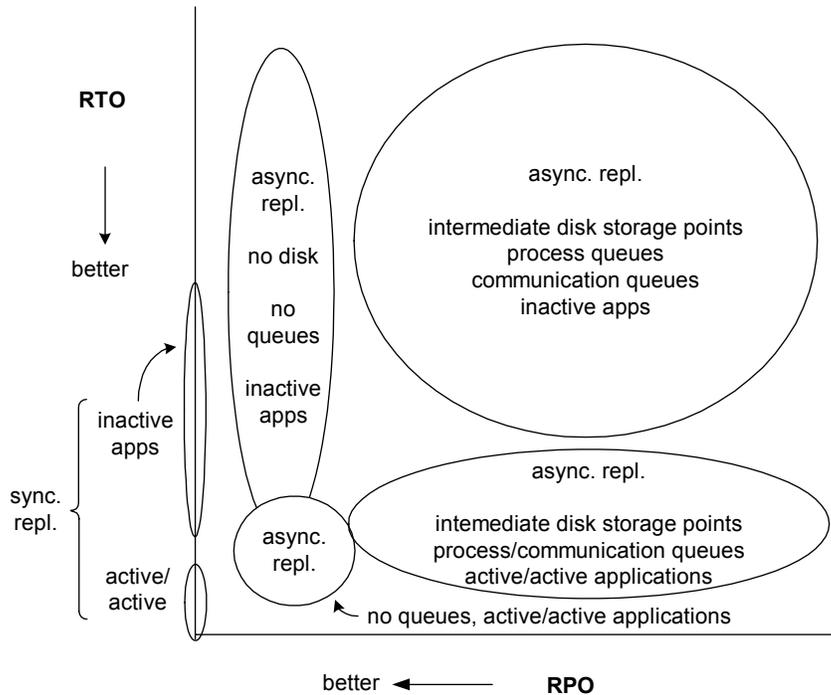
 active system

Replication Method Comparison
Figure 4

Recovery time, which directly affects RTO, is also a function of several factors. If the application processes in the backup system are inactive, they have to be activated; this can take several minutes. However, if the application is configured as an active/active application, then the only requirement is for the users on the failed system to be switched to the surviving system. The RTO is zero for all other users.

Thus, as summarized in Figure 5, the actual RPO and RTO that can be achieved are strongly dependent upon the specific technology used to implement the replication facility.

Rule 21: *RPO and RTO are both a function of the data replication technology used to maintain databases in synchronism.*



The Impact of Replication Technology on RPO and RTO
Figure 5

Multi-Node Applications

Any of these data replication techniques can be extended to multi-node applications to provide multiple recovery sites or to provide backup at one site for multiple primary sites.

Moreover, bi-directional replication can be used to spread an application load over several nodes, creating a highly available multi-node active/active system. One advantage of this is a significant increase in full capacity availability.⁷ Another advantage is that if a node fails, only a portion of system capacity is lost. Except for users at the failed node, substantially full service continues to be provided. Moreover, the downed users can be returned to service within seconds by switching them to surviving nodes.

⁷ Highleyman, W., Holenstein, B. "Availability Part 2 – System Splitting," *The Connection*, Volume 24, No. 1; January/February, 2003.

Recovery Decision Time

Should a disaster destroy a site, the site's inability to function is pretty obvious. There is not much need for a decision process to decide to activate the backup site.

However, replicated architectures also protect against normal system outages. Now when a failure occurs, a decision must be made whether to recover the failed system or to switch over to the backup system.

This can be a high-stress process. Remember Rule 19?

Rule 19: *When things go wrong, people get stupider.*

First, operational personnel must acknowledge that there is a problem. Then the problem must be diagnosed and consensus reached on the optimum recovery action. Often, management must be consulted to approve any drastic actions such as a switchover and the termination of other applications. This process can take an hour or two, especially if it happens during off-hours. This was expressed earlier as Rule 18:

Rule 18: *Rapid recovery of a system outage is not simply a matter of command line entries. It is an entire business process.*

Thus, for non-disastrous system outages, the time lines in Figure 3 should be preceded by an extended decision-making time. This directly impacts the non-replication and unidirectional replication methods.

Note, however, that decision time only affects the downed partition for active/active replication. In other words, when using active/active architectures, many users are unaffected by the outage and continue processing without incident. Recovering the rest typically consists of reconnecting them to a surviving node.

Summary

In the early days of computing, only tape was available for backups. In those days, the batch nature of computing made long RPOs and RTOs acceptable. However, as systems came online and as real-time became a reality and a necessity, understanding RPO and RTO became more important. This importance only increased as systems became more-mission critical to enterprises.

Replication technology came about to satisfy the needs of these real-time, mission-critical systems. Early replication facilities reduced RPOs and RTOs from days or hours to minutes. As pressure increased to further improve these objectives, replication technology improved to reduce replication latency and shorten recovery times. The use of

higher speed communication channels and multithreaded replicators and the elimination of intermediate disk storage points have all contributed to faster replication and less loss of data when failures occur.

Recent solutions to critical bi-directional replication problems such as the ping-ponging of updates and the detection and resolution of data collisions have allowed active/active systems to become a reality. Active/active applications allow all of the data processing power in a network to be utilized. Now, with the inherent increase in computing system power which compensates for the additional latency inherent with synchronous replication, efficient active/active systems can be built using synchronous replication. These systems will not only reduce RPO to zero, but they will also virtually eliminate RTO as a consideration.

As a result, the technology to build extremely highly available and disaster-tolerant systems at little additional cost is here today, with no data loss due to a network or node failure, and with no service loss to users except briefly for those serviced by a failed node.