the Availability Digest

www.availabilitydigest.com

The Availability Matrix

January 2011

Companies need to minimize the number of technologies in which their data-center personnel must be knowledgeable. On the other hand, the myriad applications that data-center system administrators must support depend upon a wide range of technologies to meet the differing availability SLAs (service level agreements).

Depending upon the application, SLA requirements can range from seconds to days for recovery times and for the allowable risk of lost data. The faster the recovery-time requirement and the greater the degree of data protection, the more costly is the support infrastructure required to meet the SLA.

Recovery time objectives (RTOs) and data-loss *recovery point objectives (RPOs)* do not come in handy pairs. One application may require a recovery time of four hours but cannot lose more than two minutes of data. Another application may require a recovery time of two minutes but cannot lose more than four hours of data. Numerous combinations can occur, each being satisfied by a different system configuration. Must all of the configurations be supported by the data center?

Fortunately, the technologies required to meet various RTOs and RPOs are independent. Therefore, the choice of technologies required to satisfy a given SLA can be reduced to a simple matrix – the *Availability Matrix* – in which one axis represents RTO solutions and the other axis represents RPO solutions. The intersections represent availability technologies supported by the data center.

Before we look at an example Availability Matrix, we review general RTO and RPO technologies that represent various compromises between performance and cost.

RTO Technologies

A variety of architectures satisfy RTOs ranging from seconds to days.

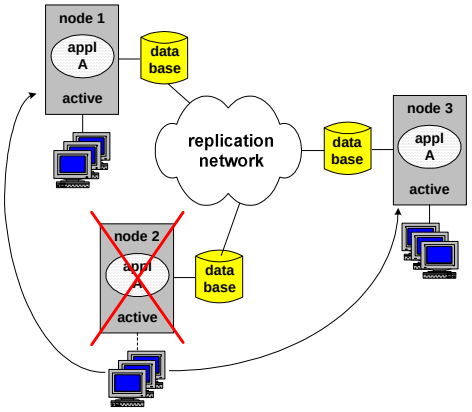
Active/Active Systems

An active/active system[1] contains two or more processing nodes, each with its own copy of the application database and each cooperating with the other nodes in a common application. The nodes can be geographically separated so that the application network can survive any common disaster.

[1] What is Active/Active?, *Availability Digest*; October 2006.
http://www.availabilitydigest.com/public_articles/0101/what_is_active-active.pdf

Any transaction can be sent to any node in the application network, and it will be processed in exactly the same way. To accomplish this, each node must have a current copy of the application database.

This is accomplished via data replication. Whenever a node makes a change to its local database, that change is replicated in real time to all of the other nodal databases in the application network. Thus, each processing node has the same view of the application state as do the other nodes.

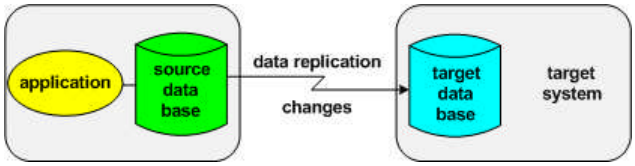


Should a node fail, all that needs to be done is to reroute transactions to one or more surviving nodes. Thus, RTOs measured in seconds can be achieved.

Active/Standby Systems

An active/standby system is a redundant pair of processing nodes, each with its own copy of the application database. One node is actively processing transactions; and the other node is standing by, ready to take over should the active node fail. The nodes can be geographically separated to achieve the desired degree of disaster tolerance.

The database of the standby node is kept in synchronism with the active node via data replication. Therefore, if it should have to take over processing, it can do so rapidly.



There are several availability levels of active/standby systems:

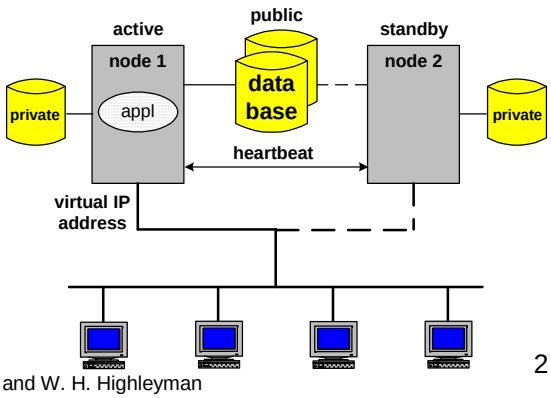
- Warm standby – In order to take over processing, the applications must be loaded; and they must mount the database. The network must be switched and the system tested. Failover is typically measured in hours.

- Hot Standby – The applications are already loaded. In order to take over processing, the applications must mount the database, the network must be switched, and the system tested. Failover can be accomplished in minutes.

- Sizzling-Hot Standby – The applications are loaded and have mounted the database. Since the standby is ready to process transactions, it can be continuously tested by sending it test transactions. All that must be done to take over operations is to reroute transactions to it. Failover can be accomplished in seconds.

Clusters

A cluster[2] comprises a set of processing nodes, all with a connection to a common application database. Generally, only one node can have the database mounted. Therefore, the



[2] Active/Active Versus Clusters, *Availability Digest*; May 2007. http://www.availabilitydigest.com/private/0205/clusters.pdf

application can only run on one of the nodes. All users access that node via a common virtual IP address. To the users, the cluster presents a single system image. The users have no indication that they are being served by a redundant architecture.

Should a node that is processing an application fail, the application is started in another node. The new application instance mounts the database and assumes the virtual IP address. Thereafter, traffic from the users is routed to the new node.

Typical failover times for clusters are measured in minutes. All cluster nodes must be collocated because they must all connect to a common data store. The cluster provides operational recovery, not disaster recovery. In order to be able to recover from a disaster, a standby cluster at a remote location must be provided. The database at the standby site is kept synchronized with the active database via data replication.

Cold Standby Systems

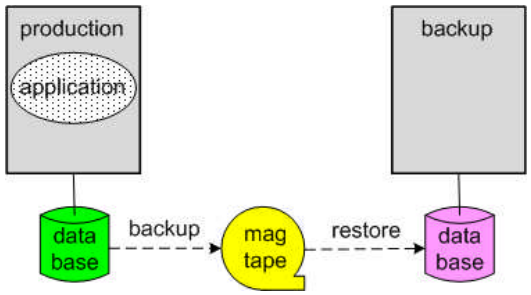
A cold standby system comprises the hardware required to run an application, but it is not otherwise configured to run the application. It does not contain the application database, nor is the application running on the system.

In order to put a cold standby into service, several things must happen. First, the database must be loaded onto the node. Then the application has to be brought up and the network switched so that it receives user traffic. The applications have to mount the database, and the system must be tested before putting it into service.

The major task in bringing up a cold standby is the loading of the database. This assumes, of course, that a backup copy of the production database is available. There are two primary methods for providing a backup copy – magnetic tape and virtual tape.

Magnetic Tape Backup

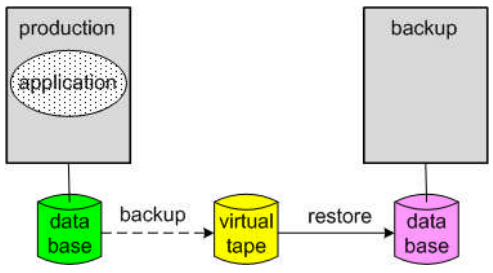
Magnetic tape is the classic backup method, going back decades to the early days of computing. Periodically, the entire production database is written to magnetic tape. Following that, according to some schedule, incremental backups of only the changes made to the database since the last full or incremental backup are written to tape. At some point, a full backup is once again taken; and the cycle with incremental backups is repeated.



Full or incremental backups are typically made daily. Should the standby system need to be brought up, the last full backup tape must be loaded, followed by each of the ensuing incremental backup tapes. For large databases, recovery times can be measured in days.

Virtual Tape Backup

Virtual-tape backup is the modern form of magnetic-tape backup. Rather than writing to magnetic tape, tape images are written to disk on a system that is usually remotely located. Should a cold standby have to be brought up, its database is loaded from the tape images on disk rather than from magnetic tape.



Loading from disk is much faster and more reliable than using magnetic tape. Recovery times can be reduced to hours in many cases.

RPO Technologies

The recovery solutions described above can be used to satisfy various RTOs. Equally important is satisfying the SLA's RPO, which specifies the maximum amount of data that can be lost. Allowable data loss can be expressed either as time or as a number of data objects. For instance, if a system is processing 100 transactions per second, an RPO specification of 100 milliseconds is equivalent to an RPO specification of 10 transactions. We will use RPOs expressed in terms of time in the following discussions.

There are several data-protection methods that can satisfy a broad range of RPOs.

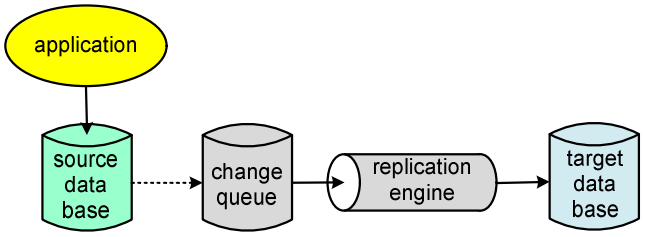
Logical Data Replication

The technique that results in minimal data loss is logical data replication. With this method, a change to the active database is immediately replicated to the target database.

There are two types of logical data replication – asynchronous replication and synchronous replication.

Asynchronous Replication

With asynchronous replication,[3] changes are sent from the active database to the standby database "under the covers." The application is unaware that replication is taking place.

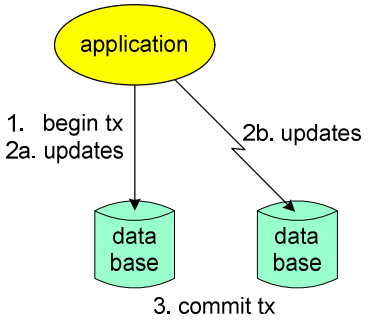


Asynchronous replication depends upon the existence of a change queue into which each database change is inserted as it is made to the active, or source, database. The asynchronous replication engine follows the change queue and sends each change to the standby, or target, database. There is a delay from when the change is made to the source database and when it is applied to the target database. This delay is called *replication latency*.

Should the source node fail, any transactions in the replication pipeline will be lost. As a general statement, the amount of data that will be lost is roughly equal to the replication latency of the replication engine. Real-time asynchronous replication engines can limit data loss to seconds.

Synchronous Replication

Synchronous replication[4] provides zero data loss. It can satisfy SLAs with an RPO specification of zero.



With synchronous replication, all database copies are included within the scope of a transaction. As an update is made, locks must first be acquired on all copies of the data

[3] Asynchronous Replication Engines, *Availability Digest*; November 2006. http://www.availabilitydigest.com/private/0102/asynchronous_replication.pdf
[4] Synchronous Replication, *Availability Digest*; December 2006. http://www.availabilitydigest.com/private/0103/synchronous_replication.pdf

object to be modified across the application network. Only then can the data object be modified. If locks on all copies of the data object cannot be obtained, the update cannot be made.

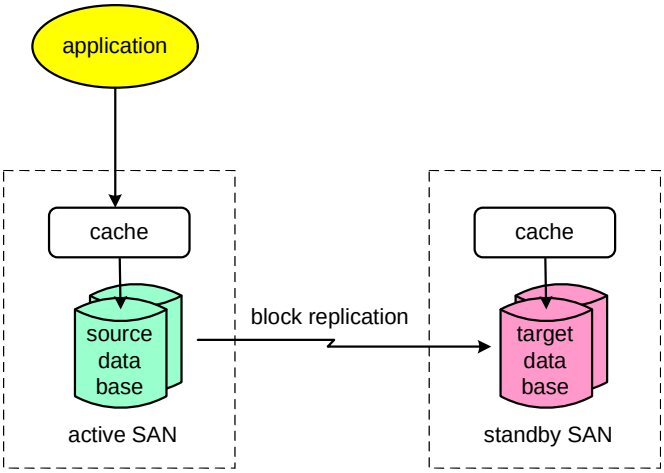
However, synchronous replication comes with its own problem. Because the application must wait while locks are acquired and data objects are modified across the network, its performance is slowed. Synchronous replication typically limits the distance that nodes can be separated, thus compromising the degree of disaster tolerance that can be achieved. Synchronous replication is typically used in campus or metro environments.

Synchronous replication guarantees that either all copies of a data object are updated or that none are. Therefore, there is no data lost should the active node fail.

Block Data Replication

Most storage area networks (SANs) provide data replication at the hardware level. As a disk block is written from the SAN cache to physical disk, the block is replicated to the standby disk.

A characteristic of block data replication is that the standby disk is essentially in a corrupted state and cannot be used as is. This is because the consistent image of the database includes the recent updates made to it that are still resident in the cache of the active SAN. Since the standby SAN has only a replicate of the active physical disk and not the entire database, it does not reflect a consistent copy of the database. Therefore, it cannot be opened by any application for query or reporting purposes as can database replicates created with logical data replication.



As with logical data replication, SAN replication can be asynchronous or synchronous. Asynchronous replication is typically scheduled every several minutes or more. Thus, RPOs measured in minutes can be satisfied.

With synchronous replication, no change can be made to the source disk unless it can also be made to the standby disk. Synchronous SAN replication significantly minimizes data loss but does not reduce RPO to zero since any data still in the active SAN's cache will be lost should the active SAN fail.

Some SAN implementations provide an option to synchronize at the cache level rather than at the disk level. In these cases, the standby database is consistent. If synchronous cache replication is used, there is no data loss following a production-node failure.

The Availability Matrix

Is there some relationship between these RTO and RPO solutions of which we ought to be aware? The answer is no. The solution to RTO is completely independent of the solution to RPO. RTO has its own set of solutions to meet different requirements, and RPO has its own set of solutions. This leads to the Availability Matrix, an example of which is shown in Figure 1. It can be used to visualize the solution to an RTO/RPO specification.

The vertical axis shows solutions for increasing RPO. The horizontal axis shows solutions for increasing RTO. By picking an RPO/RTO point that meets an SLA, the pertinent solution is determined.

Some points on this matrix may not exist because they do not make sense or because they are not supported. For instance, one would not use block replication in an active/active solution.

However, given a set of supported solutions, the solutions can be easily mapped with the Availability Matrix into the RTOs and RPOs that they do support.

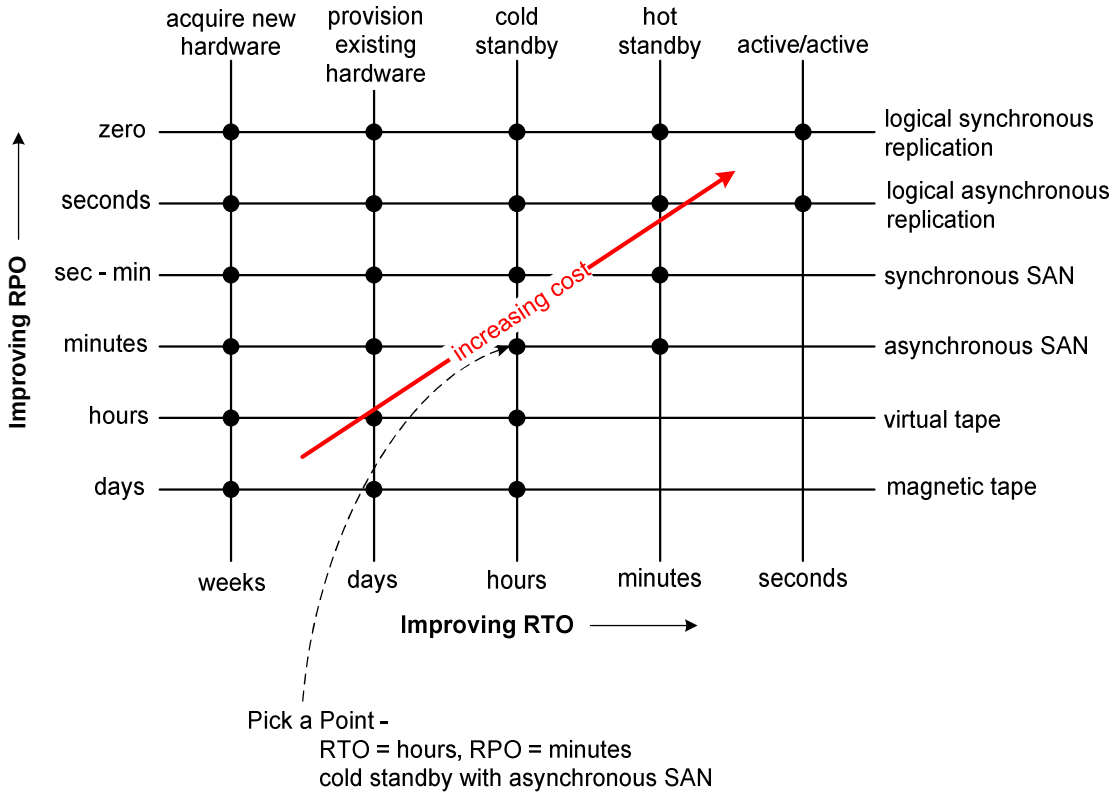


Figure 1: An Availability Matrix

For instance, using Figure 1 as an example, if the SLA calls for an RTO of eight hours and an RPO of five minutes, the technical staff might recommend a cold standby system with a SAN synchronized with the primary SAN via asynchronous replication. If this solution is too expensive for the application, the SLA will have to be relaxed to a point that is within budget.

Availability

Availability is a third parameter specified by the SLA. It is often specified as a number of *9s*. An availability of three 9s means that the system will be up 99.9% of the time. That is, it can be down about eight hours per year. A system with an availability of five 9s (99.999% uptime) will be down no more than about five minutes per year.

Availability is independent of the RTO and RPO specifications. A small RTO does not necessarily mean a high availability. Rather, the two specify the system's failure rate. If a system has a recovery time of one hour and an availability of three 9s, it can fail eight times per year and meet

the SLA requirements. Conversely, if a system has a recovery time of one hour and an availability requirement of five 9s, it can fail on the average only once every twelve years.

Availability is, in effect, a third dimension of the availability matrix but not a simple one. The availability of a system is dependent upon many factors, including the availability of its components (servers, storage devices, networks, power and air conditioning, and so forth) and the extent of redundancy used in the architecture. A straightforward way to analyze the availability of a complex solution is given in the *Availability Digest* article entitled Calculating Availability – Heterogeneous Systems Part 3.[5]

Summary

A typical data center is a mix of technologies, and there is understandable reluctance to expand the technology base. The staff will require additional training and may have more difficulty managing the expanded technology base.

With the Availability Matrix, the staff can easily present its existing capabilities to the user community to choose the appropriate availability solution. If an application requires an availability not yet supported by the data center, the Availability Matrix provides a useful method for determining the best additional technologies to incorporate into the data-center technology mix.

[5] Calculating Availability – Heterogeneous Systems Part 3, *Availability Digest*; June 2008. http://www.availabilitydigest.com/public_articles/0306/calculating_availability_heterogeneous_syst.pdf