the *Availability Digest*

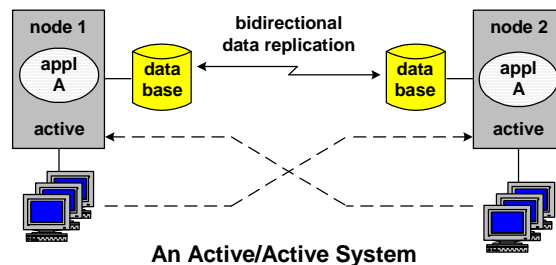# Why Are Active/Active Systems So Reliable?
September 2008

Active/active systems[1] can achieve availabilities of six 9s and beyond. Six 9s is an average of just 30 seconds of downtime per year. These systems achieve such high availabilities by providing very rapid recovery from faults – recovery times measured in seconds or subseconds. In fact, if the recovery time is fast enough, users will not realize that there has been a fault. In effect, no fault has occurred.

In redundant systems, recovery from the effects of a failed component is accomplished by removing the failed component from the system and by continuing its functions with another equivalent component that is operational. The result of this *failover* process is a reconfigured system that can continue its processing functions.

The classical method for failover is to bring up a backup system that has been idle or that has been doing other work and to put it into operation to replace the failed system. This failover process typically takes minutes to hours to restore system operation, depending upon how it is done.

In contrast, an active/active system comprises two or more processing nodes that are all actively participating in a common application. Each processing node has access to a copy of the application database. The database copies are kept synchronized via data replication. When a change is made to one database copy, that change is immediately replicated to the other database copies.

An Active/Active System

Should a processing node fail, all that is required is to reroute the users that had been using that node to one or more surviving nodes. This can be accomplished in seconds or less, resulting in failover times that may not even be noticeable to the users.

Failover time becomes an important factor in system availability. Furthermore, there is always the possibility that a failover will not work properly; and the entire system will be down until it can be recovered. We have discussed failover time in several earlier articles as part of other subjects.[2] In this article, we focus on failover time and demonstrate that it is the key to active/active system reliability.

---

[1] What is Active/Active?, *Availability Digest*; October 2006.
[2] Calculating Availability – Failover, *Availability Digest*; February 2007.
  Calculating Availability – Heterogeneous Systems Part 2, *Availability Digest*; May 2008.

Availability

In review, we consider a redundant system to be one that comprises two or more processing nodes. The availability of a node in the system is the probability that it will be operational. Let

mtbf be the mean time between failures of a node in the system.
mtr be the mean time to repair a node.
a be the availability of a node, which is the probability that the node will be operational.
f be the probability that a node has failed.

Then the nodal availability, *a*, is

$$a = \frac{mtbf}{mtbf + mtr} = \frac{1}{1 + mtr/mtbf} \approx 1 - \frac{mtr}{mtbf} \qquad (1)$$

where the approximation is valid if mtr << mtbf, which is true for our case.

Also,

$$f = (1 - a) \approx \frac{mtr}{mtbf} \qquad (2)$$

Failure Modes

We look at failure from the user's viewpoint. If a user cannot perform his duties because of a data-processing system malfunction, so far as he is concerned, the system is down. The malfunction could be a system failure, or it could be degraded performance that makes it impossible for the user to effectively function. Other users may be unaffected. But to this user, the system is down.

For simplicity purposes, we will analyze only singly-spared, dual-node systems. That is, the system comprises two processing nodes and can survive the failure of either node. The relationships derived below are extended to *n*-node systems with *s* spares in the previously referenced articles. However, we do not need that complexity to demonstrate our point.

In the classic case, one of the nodes is the primary node. The other node serves as its backup. Should the primary node fail, the applications are failed over to the backup node. An exception to this configuration is an active/active system. In this case, both nodes are active. Should one node fail, its activities are failed over to the other active node. We will look at both of these cases.

We consider three reasons why a user may be denied service.

- There has been a failure of both processing nodes (a *dual-node failure*).
- One of the processing nodes has failed, and the system is in the process of failing over (*failover*).
- One of the processing nodes has failed, and the failover has failed (a *failover fault*).

Thus, the system is down if

two nodes fail (dual-node failure) OR
one node fails, and the system is in the process of failing over (failover) OR
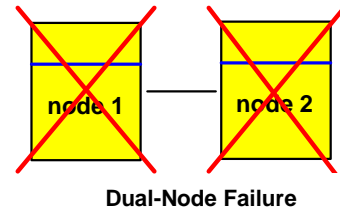one node fails, and the failover fails (failover fault).

Therefore, the probability that the system will be down is

probability (system down) =
    probability (dual-node failure) + probability (failover) + probability (failover fault)   (3)

Dual-Node Failure

A redundant system is configured with a certain number of spare nodes. If enough nodes fail so that the spare nodes are exhausted, and then one more node fails, the system is down.

In a dual-node, singly-spared system, if one node fails, the system is still capable of providing its functions. It takes the failure of both nodes to take down the system.

Dual-Node Failure

Let

    a = the availability of a node.

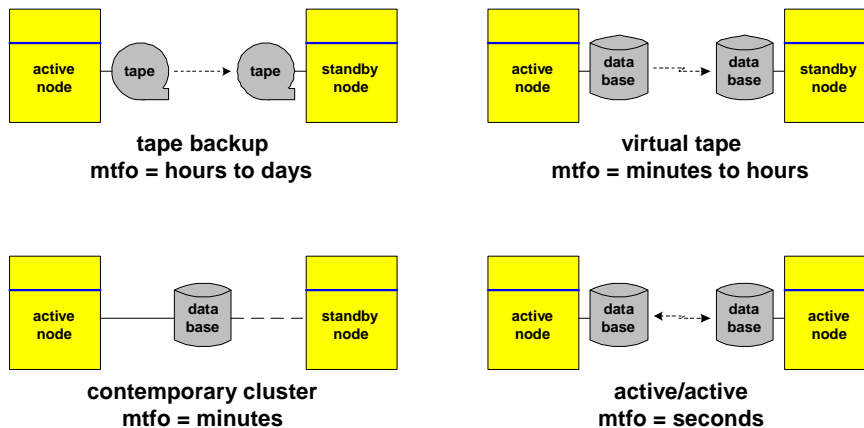The probability that either node will fail is (1-a). Therefore, the probability that both nodes will fail is

    probability (dual-node failure) = $(1-a)^2$ (4)

The probability of a dual-node failure depends only upon the availability of the nodes. We typically can't control node availability (short of buying nodes that have higher availability). Therefore, the probability of a dual-node failure is what it is. We call this the *inherent availability* of the system.

The system's inherent availability, however, is reduced by failover factors over which we do have some control. We consider these factors next.

Failover

Should the primary node in an active/backup system fail, or should either node in an active/active system fail, some or all users must be switched over to the surviving node. This process is called *failover*. We define *mtfo* as the mean time to failover – that is, the average time that it takes for the backup system to take over operations.

tape backup
mtfo = hours to days

virtual tape
mtfo = minutes to hours

contemporary cluster
mtfo = minutes

active/active
mtfo = seconds

Failover Scenarios

The amount of time that it takes to fail users over to a surviving node, depends upon the way in which failover is implemented. In the early days of computing and still to a great extent today,

magnetic tape is used to periodically record the current application database. During failover, the last recorded version of the database is read from tape to the backup node, applications are started, the node is tested, and the backup node can then be put into operation. This failover process could take hours or even days.

More recently, magnetic tape has been replaced with virtual tape, which is periodic disk-to-disk replication of the database. With virtual tape, failover can typically be accomplished in minutes to hours.

Cluster technology provides an active node and a backup node, each of which has physical connections to a common database subsystem.[3] Should the active node fail, the backup node mounts the database disks and continues processing. Recovery is typically measured in minutes.

In an active/active system, all that needs to be done is to switch users from the failed node to the surviving node. This can be done in subseconds to seconds.

Thus, depending upon the redundant technology used, failover times can range from seconds to hours or even to days.

The analysis of the probability that a system will be down while it is failing over is slightly different for active/backup systems than it is for active/active systems, but the results are the same. We look at this next.

Active/Backup Systems

In an active/backup system, there is no failover if the backup node fails. There is a failover only if the active node fails.

The active node will fail on the average of once every mtbf hours. During that time, all users will be down for a time of mtfo, where

mtfo = the mean time to failover.

Thus, the system will be down with a probability of mtfo/mtbf due to failover:

$$\text{probability (failover)} = \frac{\text{mtfo}}{\text{mtbf}}$$

From Equation (2), we can write

$$\text{mtbf} = \frac{\text{mtr}}{(1-a)}$$

Therefore,

$$\text{probability (failover)} = (1-a)\frac{\text{mtfo}}{\text{mtr}} \tag{5}$$

---

[3] <u>Active/Active Versus Clusters</u>, *Availability Digest*; May 2007.

Active/Active Systems

In an active/active system, there is a failover should either node fail. Some node in a dual-node active/active system will fail every mtbf/2 hours. Therefore, the probability of users being down due to a failover is (2 mtfo)/mtbf. However, only half of the users are affected. Therefore, the probability that a user will be down due to a failover is

$$\text{probability (failover)} = \frac{1}{2}\frac{(2\ \text{mtfo})}{\text{mtbf}} = \frac{\text{mtfo}}{\text{mtbf}}$$

and Equation (5) holds.

Failover Faults

Failover is a complex process. As system upgrades are made, will the current failover process still work? Are all components at the required revision level? Are the scripts still current?

Failover should be frequently tested to make sure that it still works. However, failing over a major system is not only expensive, but it is dangerous. For one thing, users will be down during the failover test. They may have to brought down again if the system must be failed back to the original configuration.

Even worse, there may be problems in the failover with equivalent problems when a failback is attempted. Users may be down for an extended period of time as these problems are worked out.

Consequently, failover testing is often not done frequently or thoroughly. The result is that a real failover attempt may fail. This is called a failover fault.
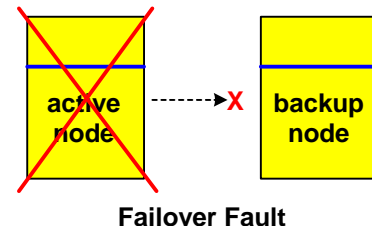
Again, the analysis of system downtime due to failover faults is slightly different for active/backup systems and for active/active systems; but the results are the same.

Active/ Backup Systems

Let

       d = probability of a failover fault.

If the backup node fails, there will be no failover and therefore no failover fault. However, if the active node fails, there will be a failover; and therefore there will be the possibility of a failover fault.

Failover Fault

The probability that the active node will fail is (1-*a*). The probability that the resulting failover attempt will fail is *d*. Therefore, the probability that the system will be down due to a failover fault is

       probability (failover fault) = (1-*a*)d   (6)

Active/Active Systems

In an active/active system, a failover will occur should either node in a dual-node system fail. Therefore, a failover will occur twice as often as it will in an active/backup system. However, only half the users will be affected. Therefore, the probability of the system being down due to a failover fault is the same as for an active/backup system; and Equation (6) holds.

It may be tempting to say that failover faults are not a factor in system availability. After all, if the failover fault probability is, say, 1%, and if a node is expected to fail on the average of once per year, the system will experience a failover fault only once every one hundred years. We will see in the following examples that this is false reasoning.

System Failure Probability

Substituting Equations (4), (5), and (6) into Equation (3), we have the probability that a dual-node, singly-spared system will fail:

probability (system down) =
   probability (dual-node failure) + probability (failover) + probability (failover fault) =

$$(1-a)^2 + (1-a)\frac{mtfo}{mtr} + (1-a)d \hspace{2cm} (7)$$

We will use Equation (7) to evaluate some interesting cases. However, first note that Equation (7) can be written as

$$\text{probability (system down)} = (1-a)\left[(1-a)+\frac{mtfo}{mtr}+d\right] = (1-a)\left[1-(a-\frac{mtfo}{mtr}-d)\right] = (1-a)(1-a') \hspace{0.5cm} (8)$$

Equation (8) states that the dual-node system acts like one node with an availability of *a* and one node with a reduced availability of *a'*, where

$$a' = a - \frac{mtfo}{mtr} - d$$

That is, the effects of failover and failover faults directly affect the availability of the system once one node has failed.

Some Examples

Let us use the result given by Equation (7) to analyze a cluster configuration and to compare it to an active/active system. We will use industry-standard servers (ISS) with three 9s availability (*a* = .999) to build a two-node cluster and a two-node active/active system. We will also study a two-node active/active system using fault-tolerant nodes with four 9s availability (such as HP's NonStop servers).

ISS Cluster

Assume the following parameters for a cluster built with ISS nodes:

a	.999
mtr	4 hours
mtfo	5 minutes
d	.01

Then

probability (dual-node failure) = $(1-.999)^2$	1.0×10^{-6}
probability (failover) = $(1-.999)(5/60)/4$	20.8×10^{-6}
probability (failover fault) = $(1-.999)(.01)$	$\underline{10.0\times10^{-6}}$
	31.8×10^{-6}

The cluster availability is 0.9999682, or a little less than five 9s. This is reasonable for a cluster.

Note that the bulk of the failure probability is due to failover and failover faults. It reduces the inherent availability of the system (the probability that it will fail due to a dual-node failure) by a factor of over 30:1.

ISS Active/Active System

In an active/active system, we can assume that failover faults will not happen. This is because users at the failed node are failing over to a node that is known to be operational since this node is, in fact, providing application functions to its users.[4] In addition, failover is very fast – subseconds to seconds – because all that must be done is to switch users to the surviving node.

Assume the following parameters for an active/active system built with ISS nodes:

a	.999
mtr	4 hours
mtfo	3 seconds
d	0

Then

probability (dual-node failure) = $(1 - .999)^2$	1.0×10^{-6}
probability (failover) = $(1 - .999)(3/3600)/4$	0.2×10^{-6}
probability (failover fault) = $(1 - .999)(.01)$	$\underline{0.0 \times 10^{-6}}$
	1.2×10^{-6}

The active/active system availability is 0.9999988, or a little less than six 9s, an order of magnitude better than the equivalent cluster. The inherent availability of the two-node system has been reduced by a factor of 1.2 rather than by a factor of over 30 in the cluster configuration. Failover is a minor consideration.

NonStop Active/Active System

Assume the following parameters for an active/active system built with NonStop fault-tolerant nodes:

a	.9999
mtr	4 hours
mtfo	3 seconds
d	0

Then

probability (dual-node failure) = $(1 - .9999)^2$	1.0×10^{-8}
probability (failover) = $(1 - .9999)(3/3600)/4$	2.0×10^{-8}
probability (failover fault) = $(1 - .9999)(.01)$	$\underline{0.0 \times 10^{-8}}$
	3.0×10^{-8}

The NonStop active/active system availability is 0.99999997, or a little less than eight 9s. Because of the reduced probability of losing the system due to a dual-node failure, failover is now a factor, decreasing the inherent availability by a factor of three.

---

[4] Actually, a failover fault could occur if users were not properly switched over to the surviving system. However, this is an easy and relatively risk-free test that can be frequently made to ensure that switching over can be done smoothly and reliably if needed.

Summary

The results of this analysis are summarized in the following table:

	ISS Cluster	ISS Active/Active	NonStop Active/Active
Dual-Node Failure	1×10^{-6}	1×10^{-6}	1×10^{-8}
Failover	21×10^{-6}	0.2×10^{-6}	2×10^{-8}
Failover Fault	10×10^{-6}	0	0
P(System Down) Availability	32×10^{-6} five 9s⁻	1.2×10^{-6} six 9s⁻	3×10^{-8} eight 9s⁻

From this comparison, it is clear that active/active systems achieve their high availabilities because of their failover characteristics. Not only do they fail over very quickly compared with active/backup and cluster configurations, but they are relatively immune from failover faults.

In fact, active/active systems do not really fail over. They simply resubmit the failed work to a known operating node. Resubmission may be done externally by the client or internally by the system. The bottom line is that active/active systems achieve their high availabilities via the philosophy of

Let it fail, but fix it fast.

"Fix it fast" is achieved by the technique of

Resubmit rather than fail over.