the Availability Digest

Calculating Availability – Heterogeneous Systems Part 4

August 2008

In the first three parts of this series,[1] we showed how to calculate the availability of complex systems comprising serial and parallel combinations of subsystems with varying availabilities. We considered not only system downtime due to multiple system failures but also system downtime due to failover times and failover faults.

In this final Part 4, we demonstrate the use of these results to calculate the availability of an active/active system backed up by a standby system that takes over only in the event of the failure of the entire active/active system.

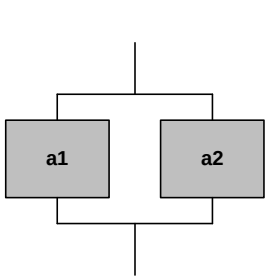
We will begin by reviewing where we left off.

A Review of Parts 1, 2, and 3

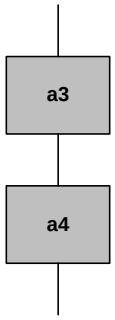
When considering system availability, it is convenient to think of a system being in one of several states. For instance, the system may be up, it may be down, or it may be failing over to a backup system.

From an availability viewpoint, there are two basic structures in a complex system – serial subsystems and parallel subsystems. A *parallel subsystem* contains two or more components arranged so that the subsystem survives the failure of one or more of its components. If there are *s* spare components in the subsystem, it takes the failure of *s*+1 components to bring the subsystem down.

A *serial subsystem* contains two or more components arranged so that all components must be operational for the subsystem to be operational. Should any one of the components fail, the subsystem fails.



A Parallel Subsystem



A Serial Subsystem

Parallel Subsystems

Let us consider a singly-spared dual-node parallel subsystem in which the nodes are the components of the subsystem. The subsystem is down under three conditions:

a) Both nodes have failed.
b) One node has failed, and the users are being failed over to a backup node.

[1] Calculating Availability – Heterogeneous Systems Part 1, *Availability Digest*; March 2008.
Calculating Availability – Heterogeneous Systems Part 2, *Availability Digest*; May 2008.
Calculating Availability – Heterogeneous Systems Part 3, *Availability Digest*; June 2008.

c) One node has failed, and the failover has failed (a failover fault).

Let

mtbf	be the mean time before failure for a node.
mtr	be the mean time to repair a node.
a	be the availability of a node = mtbf/(mtbf+mtr).
mtfo	be the mean time for a failed node to failover to a backup node.
d	probability that a failover will fail (the probability of a failover fault).
A	subsystem availability (probability that the subsystem is up).
F	probability that the subsystem is down = 1-*A*.

In Part 2, we showed that[2]

$$F = (1-a)^2 + (1-a)\frac{mtfo}{mtr} + (1-a)d \qquad (1)$$

If the two nodes in a parallel subsystem have different availabilities, say a_1 for Node 1 and a_2 for Node 2, then Equation (1) becomes

$$F = (1-a_1)(1-a_2) + \left(1-\frac{a_1+a_2}{2}\right)\frac{mtfo}{mtr} + \left(1-\frac{a_1+a_2}{2}\right)d \qquad (2)$$

In Equations (1) and (2), the availability, *A*, of the subsystem is (1-*F*).

In these equations, the first term is the probability that the system will be unavailable due to a dual-node failure. The second term is the probability that the system will be unavailable because it is in the process of failing over. The third term is the probability that the system will be unavailable due to a failover fault.

Serial Subsystems

A serial subsystem requires that all nodes (which are its components) in the subsystem are operational in order for the subsystem to be operational. Should any one node fail, the subsystem will fail. For instance, in a two-node serial subsystem in which the nodes have availabilities of a_3 and a_4 respectively, the availability of the serial subsystem is

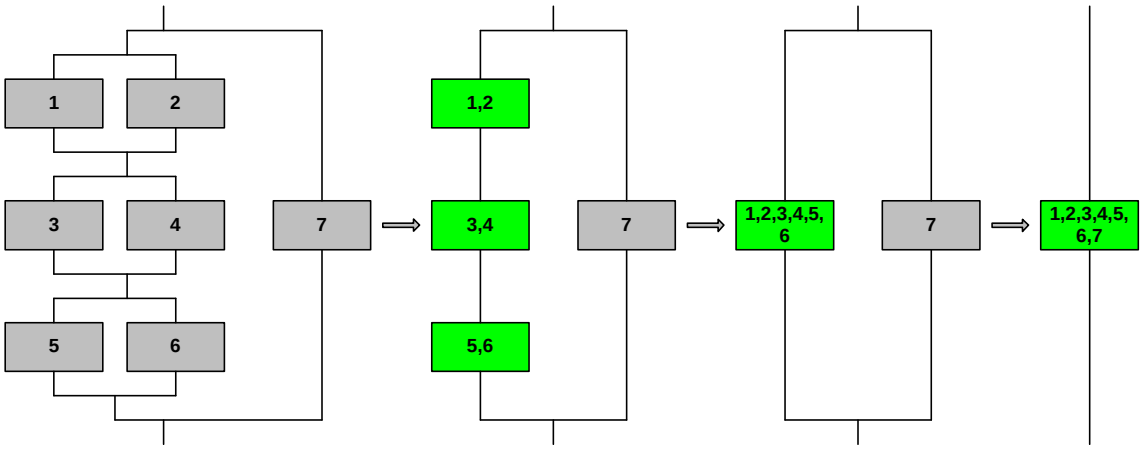
$$A = a_3 \times a_4 \qquad (3)$$

For *n* nodes in a serial subsystem, the subsystem availability *A* is the product of the availability of each of the nodes.

Complex Systems

In general, systems are constructed from parallel and serial subsystems. We call these *complex systems*. The availability of a complex system is determined first by reducing each parallel subsystem to a single node with an equivalent availability. This may leave one or more serial subsystems. Each serial subsystem is reduced to a single node with an equivalent availability. This may result in more parallel subsystems. This process continues until the system has been reduced to a single node with an availability that is the availability of the system.

[2] In Part 3, we extended this to an *n*-node subsystem with *s* spares. The example that we consider here uses only dual-node singly-spared subsystems.

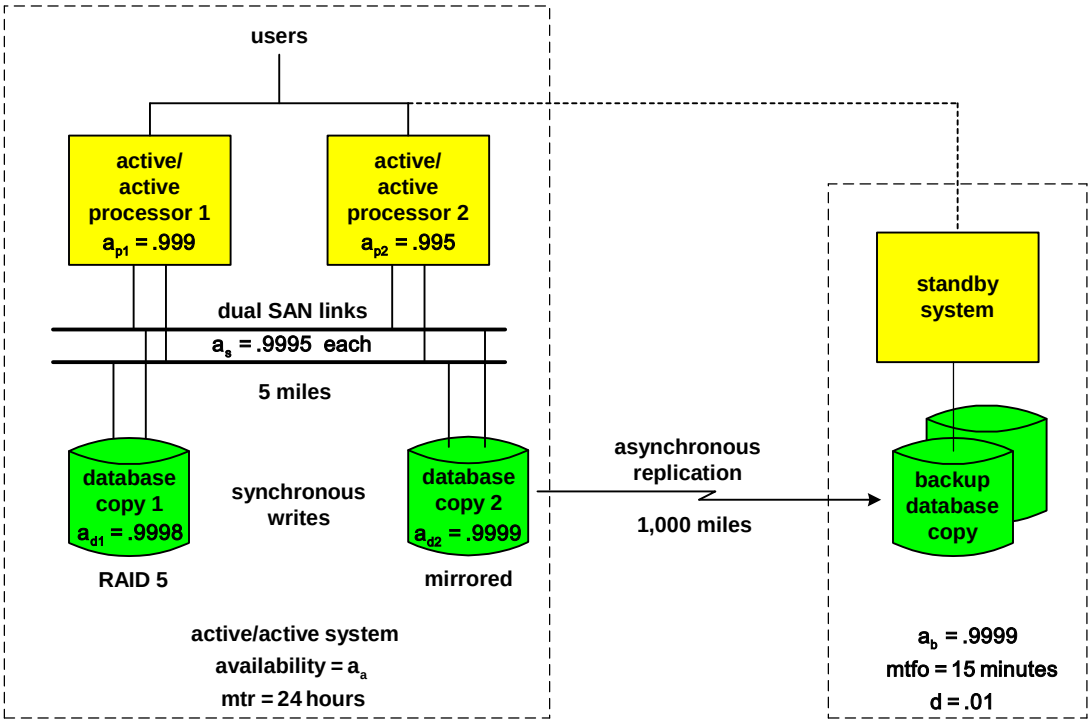


Resolving a Complex System

An Example

To illustrate the application of these concepts, consider a configuration that is an active/active system backed up by a hot standby system. The active/active system comprises two nodes that are five miles apart. The active/active databases at the two sites are synchronized via synchronous replication.

The hot standby system is located 1,000 miles from the active/active system. The standby system is idle so long as the active/active system is operational. However, should the active/active system fail, all users will be switched to the standby system; and operations will continue.[3]



A Mixed Active/Active/Standby System

[3] This is similar to a configuration that can be used to provide disaster tolerance for OpenVMS clusters. See OpenVMS Active/Active Split-Site Clusters, *Availability Digest*; June 2008.

The active/active system in this example is a two-node system split across two sites. Each processor at each site has access to two identical databases connected via a dual fibre-channel SAN. One database is resident on a RAID disk subsystem, and the other is resident on a mirrored disk subsystem. The active/active system is up if at least one processor is up as well as one SAN and one disk subsystem. Alternatively, the active/active system is down if the processor pair is down or if the dual SAN is down or if the disk subsystem pair is down. We assume that failover time from one node to another is zero and that the probability of a failover fault is also zero (reasonable approximations for an active/active system).

Should the active/active system fail, the standby system will take over operations for all the users within 15 minutes and with a 99% success rate. It will take an average of 24 hours to return the active/active system to service.

The Active/Active System

We first calculate the availability of the active/active system.

Processor Availability

The processors are both industry standard servers, but they are different models. Processor 1 has an availability a_{p1} of .999 (three 9s) and processor 2 has an availability a_{p2} of .995. Therefore, the probability of failure of the processor pair is

$$p(\text{processor pair is down}) = (1-a_{p1})(1-a_{p2}) = (1-.999)(1-.995) = (10^{-3})(5\text{x}10^{-3}) = 5\text{x}10^{-6}$$

and its availability a_p is

a_p = processor pair availability = .999995

or somewhat in excess of five 9s.

SAN Availability

Each SAN interconnect has an availability a_s of .9995. Therefore, the probability that the SAN network will be down is

$$p(\text{SAN is down}) = (1-a_s)^2 = (1-.9995)^2 = (5\text{x}10^{-4})^2 = 25\text{x}10^{-8}$$

and the availability of the SAN network a_n is

a_n = SAN availability = .99999975

or almost seven 9s.

Both processors connect to both SANs as do both disk subsystems. Therefore, either processor can access either disk subsystem over either SAN.

Database Availability

The two disk subsystems are kept in synchronism by dual writes. That is, each write is written synchronously to both disk subsystems. Therefore, either processor may use either disk subsystem over either SAN. We ignore here the problems of distributed lock management and distributed cache (see Footnote 3 above).

There are two disk subsystems accessible via the SAN. One is a RAID 5 array with an availability a_{d1} of .9998. The other is a mirrored (RAID 1) array with an availability a_{d2} of .9999 (twice as reliable as the RAID array). Therefore, the probability that the disk subsystem will be down is

$$p(\text{disk pair is down}) = (1-a_{d1})(1-a_{d2}) = (1-.9998)(1-.9999) = (2\text{x}10^{-4})(10^{-4}) = 2\text{x}10^{-8}$$

and the availability a_d of the disk subsystem is

a_d = disk subsystem availability = .99999998

or almost eight 9s.

Active/Active System Availability

The active/active system is up if the processor pair is up and if the SAN is up and if the disk subsystem is up. These subsystems are in series. Therefore, the availability a_a of the active/active system is, from Equation (3),

$$a_a = a_p a_n a_d = .999995 \text{ x } .99999975 \text{ x } .99999998 = .99999473$$

or over five 9s. This represents a probability of failure of $5.27\text{x}10^{-6}$.

In this example, it is clearly the processors that govern availability of the active/active system. Adding a third three-9s processor as an additional spare would decrease the failure probability of the processor group to $5\text{x}10^{-9}$ (over eight 9s), substantially removing it from the availability equation. The SAN network, providing almost seven 9s of availability, would now be the predominant factor.

System Availability

Should the active/active system fail, the standby system will take over. The standby system is a NonStop system with an availability a_b of four 9s:

$a_b = .9999$

The standby system will be brought into operation; and all users will be switched over to it, a process that requires 15 minutes on average. This is the mean time to failover, or mtfo:

mtfo = 0.25 hours

Since the standby system is a hot standby, all applications are running and have the local database mounted, sharing it with the asynchronous data replication facility. During the failover time, users are switched, incomplete transactions are rolled back, and test transactions are executed to ensure that the standby is correctly functioning. At this point, the standby system can begin to provide service to the users.

There is a 1% chance that this failover process will fail (the failover fault probability, d):

$d = 0.01$

Using Equation (2) as a guide, the failure modes for recovery to the backup system are as follows:

Dual-System Failure

The probability that both the active/active system and the backup system will be down is

p(dual-system failure) = $(1-a_a)(1-a_b)$
= $(1-.99999473)(1-.9999) = (5.27\text{x}10^{-6})(10^{-4}) = 5.27\text{x}10^{-10}$

Failover

The probability that there will be a failover is the probability that the active/active system will fail. Users are then failed over to the backup system. We assume that it takes 24 hours on the average to return the active/active system to service. This is its mtr:

mtr = 24 hours

From Equation (1), the probability that the system will be down because of a failover is

p(failover) = $(1-a_a)$mtfo/mtr = $(5.27\text{x}10^{-6})\text{x}0.25/24 = 5.49\text{x}10^{-8}$

Failover Fault

Also from Equation (1), the probability that the system will be down due to a failover fault is

p(failover fault) = $(1-a_a)$d = $(5.27\text{x}10^{-6})\text{x}0.01 = 5.27\text{x}10^{-8}$

System Availability

The probability of failure for the system is therefore

p(system failure) = p(dual-system failure) + p(failover) + p(failover fault)
= $5.27\text{x}10^{-10} + 5.49\text{x}10^{-8} + 5.27\text{x}10^{-8} = 10.81\text{x}10^{-8} \approx 10^{-7}$

Thus, this active/active system with a hot standby can achieve seven 9s of availability. Seven 9s represents an average downtime of 3 seconds per year. Assuming that it will take 24 hours to restore service to the users if all systems failed, this represents a system mean time before failure of $24/10^{-7}$ hours, or almost 300 centuries.

Summary

By breaking down a complex system into an iterative series of parallel subsystems, the availability of the overall system can be determined. A parallel subsystem comprises a set of nodes that can withstand the failure of one or more nodes. A serial subsystem will fail if any node in the subsystem fails.

The first step in the calculation of the system's availability is to resolve the availability of parallel subsystems in which one or more nodes may fail and the system remains operational. If there is a failover time required, the downtime during failover and the probability of a failover fault must be considered.

The next step is to resolve the availability of any serial subsystems in the system. This may leave more parallel subsystems to resolve and so on until the system has been reduced to a single node whose availability is the availability of the system.

The example demonstrates that system failure intervals measured in centuries can be achieved with today's technology used in reasonable system configurations.