

## Calculating Availability – Nodes, Subsystems, and Systems

June 2007

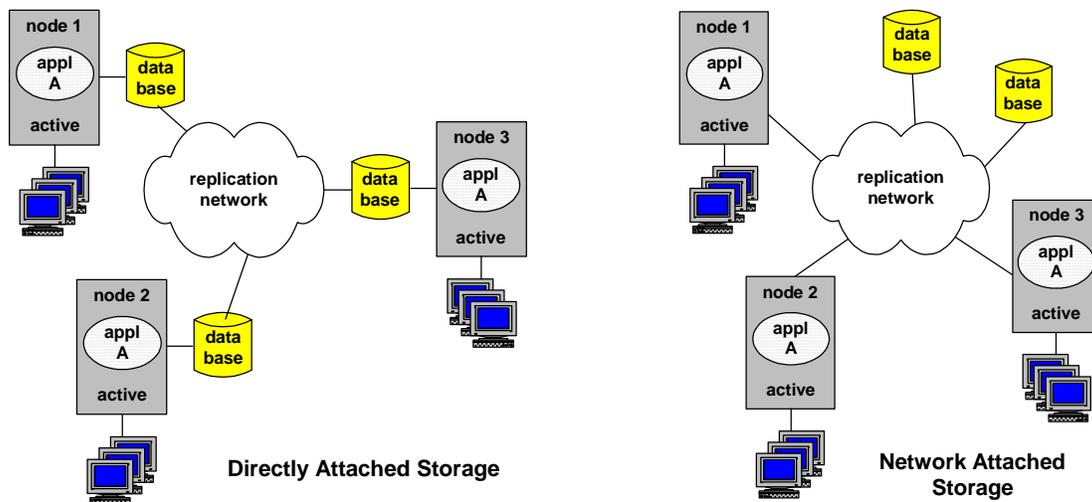
In our previous articles, we have described multinode architectures that can provide very high availabilities. Predominant among these architectures are active/active<sup>1</sup> and clustered systems.<sup>2</sup> These *systems* are made up of *nodes* that are themselves computing systems.

In our analyses of system availability in our Geek Corner articles, we have talked throughout of *systems* and the *subsystems* that make up these systems. It is now time to bring the terms *system*, *subsystem*, and *node* together into a consistent whole.

### A Review of Multinode Architectures

#### Active/Active

In an active/active system, two or more computing systems cooperate in a common application. We call each such computing system a *node* in the active/active system. Each node has access to a copy of the common application database, which is either directly attached to the node or is accessible through the network. The database copies are synchronized via data replication. Whenever a change is made to the database at one node, that change is immediately replicated to the other database copies in the application network. Thus, each node has local or network access to a consistent and up-to-date (at least within the replication latency) copy of the application database.



<sup>1</sup> What is Active/Active?, *Availability Digest*, October, 2006.

<sup>2</sup> Active/Active vs. Clusters, *Availability Digest*, May, 2007.

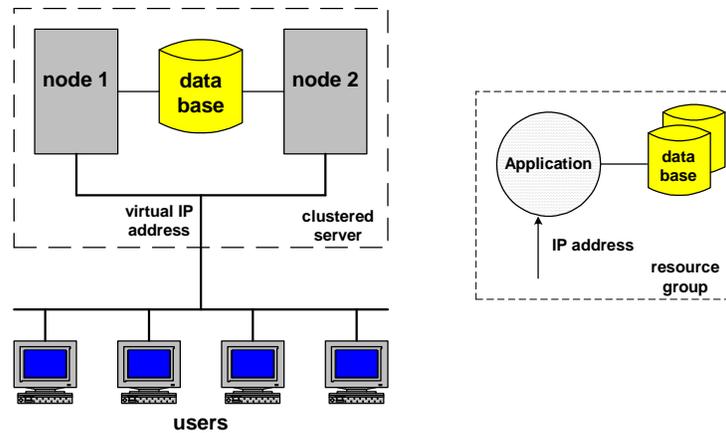
Should a node fail, all that is necessary to restore service to its users is to switch those users to a surviving node. This can typically be done in seconds.

The nodes in an active/active system can be geographically distributed to provide disaster tolerance. Any event which takes down one node is unlikely to affect the other nodes in the network.

## Clusters

A cluster is a grouping of two or more computing systems, or nodes, that can back each other up. They each have access to the various application databases and to the network that interconnects the nodes with the users.

Each node is capable of running each application, but only one node can run a specific application at any one time. Otherwise, two nodes might try to update the same data item in the database, thus causing database corruption.



Should a node fail, the *resource group* for each application running on that node is failed over to its backup node. A resource group includes the application code, the application database, and the IP address that is used by users to access the application.

Clustered nodes are generally collocated. However, if performance issues permit, the nodes may be distributed across a campus environment. Disaster tolerance can be achieved by having two or more remotely located clusters that back up each other. Even in this architecture, there can be only one instance of any given application running.

## Nodes and Systems

Each of these active/active or clustered *systems* is made up of two or more *subsystems*, or *nodes*.

Furthermore, a node might itself be a system made up of its own nodes. For instance, in an active/active system implemented with NonStop servers, each *node* in the active/active system is a NonStop server. However, a NonStop server is itself a system made up of *nodes*. In this case, the nodes are the processors comprising the NonStop server.

## ***A Review of Availability***

In our previously published availability analyses, we generally talked about *systems* and *subsystems*. When applying these analyses to multinode systems such as those described above, a *subsystem* is a *node*.

Before applying these analyses to multinode architectures, let us first review them. For simplicity's sake, we do this for the case of a single-spared system. That is, the system will survive the failure of a single node. If more than one node fails, a multinode system may continue to hobble along but with an unacceptable performance; and it is therefore considered to be down.

### Mean Time to Return a Node to Service

In order to return a failed node to service, it must first be repaired if necessary; and then it must be recovered by loading its applications, by recovering its database, and by opening the database.

Let

- mtr be the mean time to return a node to service.
- $h$  be the probability that a node failure requires a hardware repair.
- $r_h$  be the average time required to repair the hardware.
- $r'$  be the average time to recover the node.

Then the average time required to return a node to service following a failure, mtr, is<sup>3</sup>

$$\text{mtr} = hr_h + r'$$

### Node Probability of Failure

Let

- $a$  be the availability of a node.
- $f$  be the probability of failure of a node.
- mtbf be the mean time before failure for a node.

The availability,  $a$ , of a node is the probability that it will be operational.

$$a = \frac{\text{mtbf}}{\text{mtbf} + \text{mtr}} \approx 1 - \frac{\text{mtr}}{\text{mtbf}}$$

The probability,  $f$ , that a node will fail is  $(1 - a)$ :

$$f = \frac{\text{mtr}}{\text{mtbf}}$$

### Mean Time to Return a System to Service

Should a system fail because two of its nodes have failed, it can be returned to service as soon as one of its nodes is returned to service and as soon as system recovery has been completed. System recovery might involve such tasks as synchronizing the recovered node's database and entering transactions that had been processed manually during the outage.

---

<sup>3</sup> Calculating Availability – Hardware/Software Faults, *Availability Digest*, January, 2007.

Note that mtr is the time to return a node to service. However, if two nodes are being repaired in parallel, the average time required to return the first node to service is mtr/2.<sup>4</sup>

Let

MTR be the mean time to return the system to service.

R be the system recovery time.

Then the average time to return the system to service is

$$MTR = mtr / 2 + R$$

### System Probability of Failure

With the above parameters, our analyses showed that the system probability of failure,  $F$ , is<sup>5</sup>

$$F = \frac{mtr / 2 + R}{mtr / 2} \frac{n(n-1)}{2} (1-a)^2$$

where

$F$  is the probability of failure of the system.

$n$  is the number of nodes in the system.

### System Availability

System availability,  $A$ , is

$$A = 1 - F$$

### System Mean Time Before Failure

Let

MTBF be the system mean time before failure.

Since

$$A = (1 - F) = \frac{MTBF}{MTBF + MTR} \approx 1 - \frac{MTR}{MTBF}$$

then

$$F = \frac{MTR}{MTBF}$$

and

$$MTBF = MTR / F$$

### **Nodes**

Let us now apply these analyses to the nodes of a multinode system. We take as an example of a node a NonStop system, which is in itself a multinode system made up of sixteen processors (its nodes). We assume that a hardware fault causes a processor failure 20% of the time and requires an average of 24 hours to repair. Once the processor is repaired, it takes two hours to

<sup>4</sup> Calculating Availability – Repair Strategies, *Availability Digest*, November, 2006.

<sup>5</sup> Calculating Availability – The Three Rs, *Availability Digest*, December, 2006.

recover the processor and then four hours to return the server to service. Furthermore, let processor mtbf be 10,000 hours.

Thus

$n$	= 16 processors
$h$	= 0.2
$r_h$	= 24 hours
$r'$	= 2 hours
$R$	= 4 hours
mtbf	= 10,000 hours

Applying the above analyses to these parameters, we have for the NonStop nodes:

Node Availability Attributes		
mtr	= processor mean time to return to service	= 6.8 hours
$(1 - a)$	= processor probability of failure	= $6.8 \times 10^{-4}$
MTR	= system mean time to return to service	= 7.4 hours
$F$	= system probability of failure	= $1.2 \times 10^{-4}$
$A$	= system availability	= .99988 $\approx$ four 9s
MTBF	= system mean time before failure	= 7.1 years

### Systems

Let us consider an active/active system in which the nodes are now the NonStop servers which we have analyzed above. The system's node mtr and mtbf are now the MTR and MTBF of the NonStop server calculated above, as is the node probability of failure. Let us further assume that as soon as a node is returned to service following its repair and recovery, the active/active system is returned to service. There is no hardware repair time ( $h = 0$ ) nor is there any further recovery time or system restore time ( $r' = R = 0$ ). Let us take the case of a four-node active/active system. In this case,

$n$	= 4 nodes
mtr	= node MTR = 7.4 hours
$h$	= 0
$r_h$	is not applicable
$r'$	= 0
$R$	= 0
mtbf	= node MTBF = 7.1 years

Thus, for this active/active system, we have:

System Availability Attributes		
mtr	= node mean time to return to service	= 7.4 hours
$(1 - a)$	= node probability of failure	= $1.2 \times 10^{-4}$
MTR	= system mean time to return to service	= 3.7 hours
$F$	= system probability of failure	= $9 \times 10^{-8}$
$A$	= system availability	= .99999991 $\approx$ seven 9s
MTBF	= system mean time before failure	= 47 centuries

## **Summary**

Active/active and clustered systems are made up of subsystems, or nodes. If a node itself comprises multiple subsystems, then its availability attributes must first be calculated based on the availability attributes of its subsystems. These nodal availability attributes are then carried over as subsystem parameters for the calculation of the availability of the full active/active or clustered system.