the Availability Digest

Repair Strategies

October 2014

Mission-critical applications are typically protected by providing one or more backup systems to take over processing should the production system fail. The application is considered down if its services are unavailable to its users. We consider in this article one of the factors that determines application downtime – the repair capabilities of the organization.[1]

A Production System with a Backup

Let us first consider a production system with a single backup system, the normal case. There are three primary reasons why the application may be down:

- The production system has failed and is in the process of failing over to the backup system.
- The production system has failed, followed by a later failure of the backup system. Both systems are now down and are awaiting the repair of at least one of the systems so that the application can be restarted.
- The production system has failed, and the failover to the backup system has failed. Now, both systems are down and are awaiting the repair of at least one of the systems so that the application can be restarted.

In the latter two cases, the recovery time for the application is dependent upon how long it takes to repair one system. This time, of course, depends upon the repair resources that can be brought to bear. If there is only one repair team, it will be working on the failed production system when the backup system fails. The repair of the backup system will have to await the repair of the production system. We call this *serial repair*.

However, in many cases, there may be two repair teams. This is especially true if the production and backup systems are located at different sites. In this case, the repair of the backup system will begin immediately following its failure. There is some chance that the backup system will be repaired before the production system and result in a shorter downtime than what would have occurred with serial repair. We call this *parallel repair.*

Let us let the parameter *a* represent the availability of a single system (a node in the total redundant system). *a* is the percent of time that the node is up – that is, it is the probability that

[1] This analysis is an extension of an analysis in a previous Digest article, but with a more accurate approach: Calculating Availability – Repair Strategies, *Availability Digest*; November 2006. http://www.availabilitydigest.com/private/0102/calculating_availability_repair_strategies.pdf

the node will be up. Therefore, the probability that a node will be down is (1-*a*), and the probability that the system will be down is the probability that both nodes will be down.

Sequential Repair

For sequential repair, there is only one repair team. There are two ways that the two nodes can fail. One way is that the production node fails, followed later by the backup node. The repair team will be working on the production node at the time of the failure of the backup node. The other way is for the backup node to fail first, followed by the production node. The repair team will be working on the backup node at the time of the dual failure.

In either case, the probability of system failure, F, is the probability that both systems are down, or $(1-a)^2$. Since there are two ways in which both nodes will be down, then

$$F = 2(1-a)^2 \qquad \text{Probability of system failure with sequential repair} \qquad (1)$$

Parallel Repair

With parallel repair, the repair of the second downed system will begin immediately upon its failure. The repair of both systems will occur simultaneously.

This brings us to an interesting observation about the probability distribution that we use throughout availability analysis. We assume that events such as failures and repairs are random. That is, within a certain time interval, the probability that an event will happen is fixed and is not affected by any history. The probability of any event happening during that time interval does not change whether a similar event has just happened or has not happened for a long time.

Thus, if the average (mean) repair time for a system is *mtr* (mean time to repair), then when the first system fails, its average repair time will be *mtr*. However, when the second system fails, if the first system is still down, its expected repair time from that point in time will still be *mtr*, as will the repair time of the second system.

Thus, in the interval *mtr* following the failure of the second system, on the average there will be two repairs, one for each system. Since repairs are random, the average time to repair the first system will be half the average repair time, or *mtr*/2. Consequently, the system downtime is reduced by half relative to that with sequential repair:

$$F = 2(1-a)^2/2 = (1-a)^2 \qquad \text{Probability of system failure with parallel repair} \qquad (2)$$

Multinode System with One Spare

We now consider a system with *n* nodes and one spare. This means that the system will continue to function with (*n*-1) nodes operational. It is down if two nodes have failed.

In this case, there are *n*(*n*-1) ways for two nodes to fail (note that this is equal to 2 for two nodes, as noted in the previous section). First, one of *n* nodes must fail, then one of the remaining (*n*-1) nodes must fail. Thus, for serial repair,

$$F = n(n-1)(1-a)^2 \qquad \text{Probability of system failure with sequential repair} \qquad (3)$$

As argued above, with parallel repair, the probability of system failure is cut in half:

$$F = \frac{n(n-1)}{2}(1-a)^2 \qquad \text{Probability of system failure with parallel repair} \qquad (4)$$

Multinode System with Multiple Spares

If there are *n* nodes with *s* spares, then (*s*+1) nodes must fail in order for the system to fail. The next question is how many ways are there for *s*+1 nodes to fail? This is the number of failure modes, *f*, for the network and is the number of ways that *s*+1 nodes out of *n* nodes can fail. The number of such combinations of (*s*+1) nodes out of *n* nodes failing is given by the expression

$$f = \frac{n!}{(n-s-1)!} \qquad (5)$$

The symbol "!" means "factorial." For instance, 3! is 3x2x1 = 6. 0! Is a special case and is equal to 1.

Note that for *n* = 2 and *s* = 1, f = 2 as would be expected (see Equation (1).

Thus, in this case,

$$F = f(1-a)^{s+1} \qquad \text{Probability of system failure with sequential repair} \qquad (6)$$

If full repair is available, then following the failure of (*s*+1) nodes, there will be (*s*+1) repairs in the time *mtr* following the failure of the node that took the system down. The system repair time will be reduced from *mtr* to *mtr*/(*s*+1), and

$$F = \frac{f}{s+1}(1-a)^{s+1} \qquad \text{Probability of system failure with parallel repair} \qquad (7)$$

Failure Modes

In general, in an *n*-node system with *s* spares, not every combination may cause the system to fail. In this case, the number of failure modes, *f*, may be less than the maximum given by Equation (5). It is important to analyze the distribution of critical processes and hardware across the nodes to determine the actual value of *f*.

In any event, the value of *f* for a dual node system with a single spare (the common case) will always be 2.

Repair Teams

In the general case, if there *n* nodes, each node may be in a different data center, and in that case there will be multiple repair teams to provide parallel repair. However, often there are multiple nodes in multiple data centers, with one repair team in each data center. In this case, there may be fewer than (*s*+1) repair teams. If there are *r* repair teams that can be effectively used, then Equation (7) becomes

$$F = \frac{f}{r}(1-a)^{s+1} \qquad \text{Probability of system failure with parallel repair} \qquad (8)$$

where

F = probability of system failure
a = availability of a node
s = number of spares
f = number of failure modes
r = number of repair teams

In some cases, a single repair team might be able to provide parallel repair. For instance, the repair team might include a hardware expert, a network expert, a database expert, an operating system expert, and an application expert. It is likely that multiple failures might be due to different factors. For instance, one system has a hardware failure and another has an application fault. In this case, the repair team can provide parallel repair.

There are some cases in which the calculation of the effective number of repair teams may be more complex. For instance, consider the case of a system with four nodes split evenly across two data centers. If two systems in one data center fail, there is effectively only one repair team. If one system fails in each data center, there are effectively two repair teams. It is important to analyze the repair capabilities of a distributed system to come up with an accurate value for *r*, the effective number of repair teams.

Summary

The repair strategy used by an enterprise can have a significant effect on application downtime. Especially in the case of a production/backup pair, it is important to have two repair teams so that the recovery of a dual-node failure can be executed in parallel. This simple technique will cut application downtime in half.