the Availability Digest
www.availabilitydigest.com

Improving Availability via Staggered Systems

February 2017
Dr. Bruce D. Holenstein
Dr. Bill Highleyman
Paul J. Holenstein

In our article in the January 2017 issue of the Availability Digest, "The Fallacy of Existing Availability Theory," we introduced the concept of staggering system start times to improve availability. We also introduced the term 'Mean Time To Failure,' MTTF, to replace the commonly used measure MTBF. MTBF is the Mean Time Between Failures and is constant. It is memoryless. If MTBF is 1,000 hours, and 500 hours have passed, MTBF is still 1,000 hours. The probability that the system will fail in the next 1,000 hours remains the same.[1]

Mean Time To Failure (MTTF)

However, this is not the real world, where as time passes, the probability of failure grows nearer. This is what MTTF measures. As time goes on, the Mean Time To Failure grows shorter in most real-life scenarios. This is illustrated in Figure 1 and Figure 2. In these figures, a typical failure probability distribution for a system, $p_f(t)$, is shown. Its value at time t_i is p_i. The probability that the system will fail in some small interval, Δt, around time t_i is $p_i \Delta t$. The MTTF is the average of these probabilities starting from some particular time. As shown in Figure 1 starting at time zero, the MTTF is

$$\text{MTTF} = \sum_{i=0}^{\infty} t_i p_i \Delta t \tag{1}$$

However, if we wait a time T as shown in Figure 2, the MTTF is now

$$\text{MTTF} = \frac{\sum_{i=T}^{\infty} (t_i - T) p_i \Delta t}{\sum_{i=T}^{\infty} p_i \Delta t} = \frac{\sum_{i=T}^{\infty} t_i p_i \Delta t}{\sum_{i=T}^{\infty} p_i \Delta t} - T \tag{2}$$

Comparing Equations (1) and (2), it is clear that MTTF has grown smaller as time goes on.

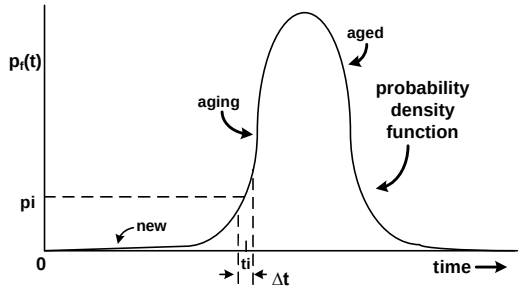

Figure 1: MTTF at Time 0

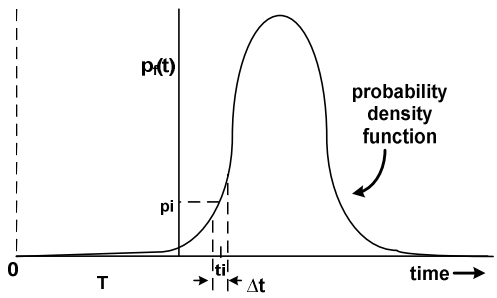

Figure 2: MTTF at Time T

[1] This paper was first published in the January/February 2017 issue of The Connection. It is republished here with the kind permission of The Connection.

© 2017 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com
For discussion, contact editor@availabilitydigest.com

System Staggering

By staggering the start times of the nodes in a redundant system, availability can be improved. If the systems are started at the same time, their peak failure probabilities are aligned; and the probability that they will fail at the same time is high. However, if they are started at different times (staggered starting times), the peak failure probability of one system likely will be aligned with a much smaller probability of failure of the other system. Therefore, the probability that both systems will fail at the same time is significantly less.

For instance, if the two servers in a redundant architecture are each expected to fail every six months, then they should be started three months apart.

In this article, we analyze the improvement in availability that can be obtained via staggered starts.

MTTF for Staggered Starts

The staggered redundant system we will analyze comprises System 1 and System 2. Let *s* be the amount of stagger time. That is, System 2 has been started after System 1 by a time of *s*.

The failure probability distribution for System 1 is $p_{f1}(t)$. The failure probability distribution for System 2 is $p_{f2}(t\text{-}s)$ since it was started at a time *s* after System 1.

Should System 1 fail, it will take an average time of MTR (mean time to repair) to return it to operation. The probability that system 1 will still be failed at time *t* is

$$\text{probability that System 1 will still be failed at time t} = \int_{t-MTR}^{t} p_{f1}(t')dt' \tag{3}$$

The probability that System 2 will fail by a later time *T* while System 1 is still down is

$$\text{probability that System 1 and System 2 will fail by time T} = \int_0^T \left[p_{f2}(t-s) \int_{t-MTR}^{t} p_{f1}(t')dt' \right] dt \tag{4}$$

The MTTF of the staggered system is

$$MTTF(s) = \int_0^\infty t \left[p_{f2}(t-s) \int_{t-MTR}^{t} p_{f1}(t')dt' \right] dt \tag{5}$$

MTR is typically measured in hours, whereas the probabilities of failure are measured in months. For small values of MTR, Equation (5) can be simplified since $p_{f1}(t)$ will be almost constant over the time MTR. Thus,

Probability of a dual-system failure as MTR → 0 =

$$\lim_{MTR \to 0} \int_0^T \left[p_2(t-s) \int_{t-MTR}^{t} p_1(t')dt' \right] dt = MTR \int_0^T p_2(t-s)p_1(t)dt \tag{6}$$

MTTF is a function of the stagger time, *s*. To determine the optimum value for *s* (that is, the value that will yield the maximum MTTF value), differentiate MTTF(s) with respect to *s*; and set the result to zero:

$$\frac{d}{ds} MTTF(s) = 0 \tag{7}$$

© 2017 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com
For discussion, contact editor@availabilitydigest.com

Solving for *s* will enable one to pick the optimal value of the stagger time to reduce the mean time to failure.

Software Failures

Software has different failure modes than hardware. Software bugs lurk in areas that seldom get executed and typically go undetected until the erroneous code is executed. When this happens, the system typically fails.

As an example, consider a software counter overflow. The program includes a transaction counter that is incremented on every transaction. However, if the transaction counter should reach its limit, the next transaction will cause a counter overflow that in our example will cause the system to crash.

The probability of failure of the system, $p_f(t)$, is shown in Figure 3. $p_f(t)$ is normally zero (the system is operational). However, when the counter overflows, $p_f(t)$ jumps to one (the system is down). The system will remain down until it is rebooted, at which time the system will again be operational; and $p_f(t)$ will return to zero. The availability of the system is shown in Figure 4.

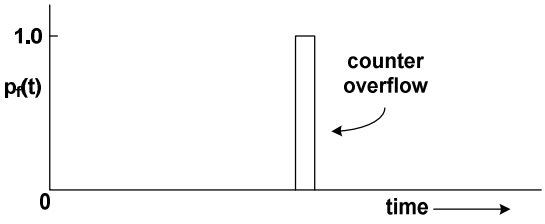

Figure 3: Software Counter Overflow Failure

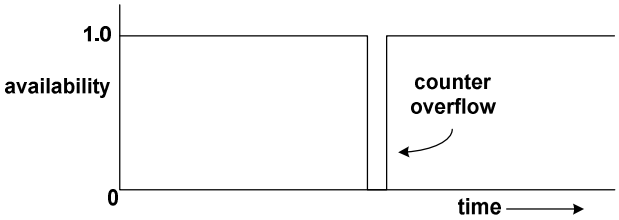

Figure 4: Availability Due to a Software Counter Overflow Failure

In a redundant system, if the two systems are started simultaneously or are not sufficiently staggered, the software fault will occur at the same time in both systems, as shown in Figure 5 and Figure 6. When both systems fail, the capacity of the redundant systems falls to zero.

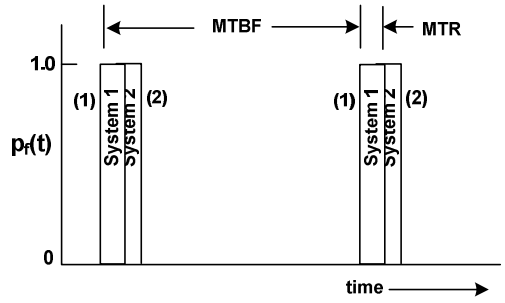

Figure 5: Insufficiently Staggered Software Systems

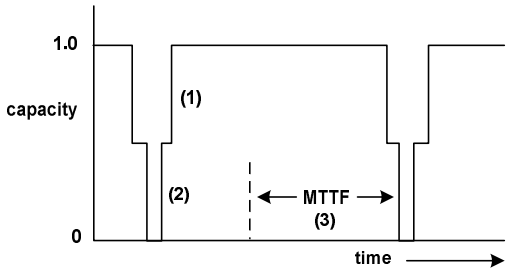

Figure 6: Availability of Insufficiently Staggered Software Systems

However, if the two systems are sufficiently staggered, as shown in Figure 7 and Figure 8, one system will always be operational; and the redundant system capacity will not fall below 0.5.

© 2017 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com
For discussion, contact editor@availabilitydigest.com

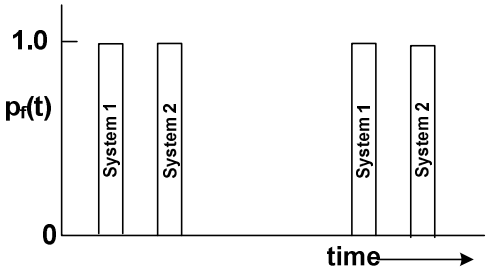

Figure 7: Staggered Software Systems

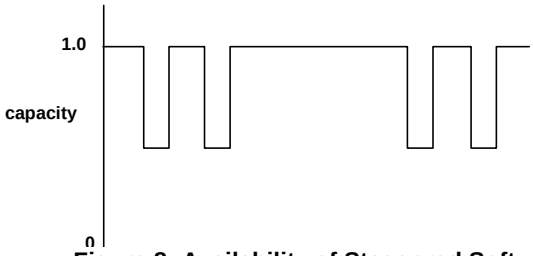

Figure 8: Availability of Staggered Software Systems

Correlation

A primary goal in choosing the proper stagger time *s* is to minimize the correlation between the failure probability distributions of System 1 and System 2. If the two distributions are highly correlated, the peak of one distribution likely will align with the peak of the other distribution. The probability of a dual system failure is higher than if the peaks of each distribution are not aligned.

The correlation between the two failure probability distributions can be measured by the correlation coefficient. Let there be two data sets, $\{x_1 \ldots x_n\}$ and $\{y_1 \ldots y_n\}$. The degree to which these two data sets are correlated is given by the correlation coefficient, *r*:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (8)$$

where $\bar{x}$ and $\bar{y}$ are the means of the data sets.

The meaning of the correlation coefficient can be seen more clearly graphically, as shown in Figure 9 and Figure 10. Figure 9 shows two highly correlated distribution functions, x(t) and y(t). Note that at each point in time, $(x_i - \bar{x})$ and $(y_i - \bar{y})$ are either both positive or negative. Therefore, the correlation coefficient, *r*, is positive since the summation in the numerator, $(x_i - \bar{x})(y_i - \bar{y})$, is always positive.

Similarly, Figure 10 shows two distribution functions that are poorly correlated. At many points in time when $(x_i - \bar{x})$ is positive, $(y_i - \bar{y})$ is negative and vice versa. Therefore, the numerator in *r* can approach zero.

Note that for a pair of perfectly correlated functions, $(x_i - \bar{x}) = k(y_i - \bar{y})$, and *r* is equal to 1. For a pair of functions that are exactly anti-correlated, $(x_i - \bar{x}) = -k(y_i - \bar{y})$, and *r* is equal to -1. Therefore, the correlation coefficient, *r*, ranges from +1 for perfectly correlated functions to -1 for perfectly anti-correlated functions. If r = 0, there is no correlation between the functions.

© 2017 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com
For discussion, contact editor@availabilitydigest.com

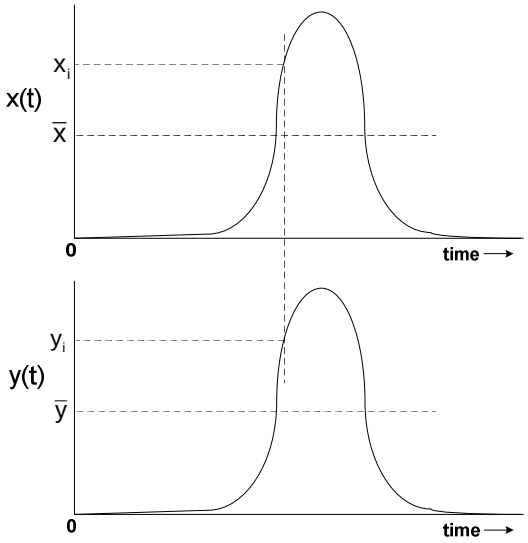

Figure 9: Highly Correlated Distributions

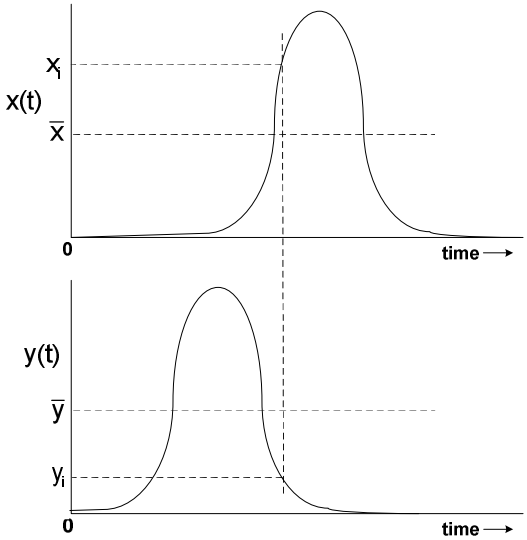

Figure 10: Poorly Correlated Distributions

As previously discussed, Equation 7, $\frac{d}{ds}MTTF(s) = 0$, yields the optimal value for the stagger time *s* as it takes into consideration MTR. However, a close approximation that is useful in practice is to simply use the stagger value that minimizes the correlation between the failure probability distribution for System 1 and System 2. As stagger time is varied, the correlation coefficient between the two probability distributions will look something like that shown in Figure

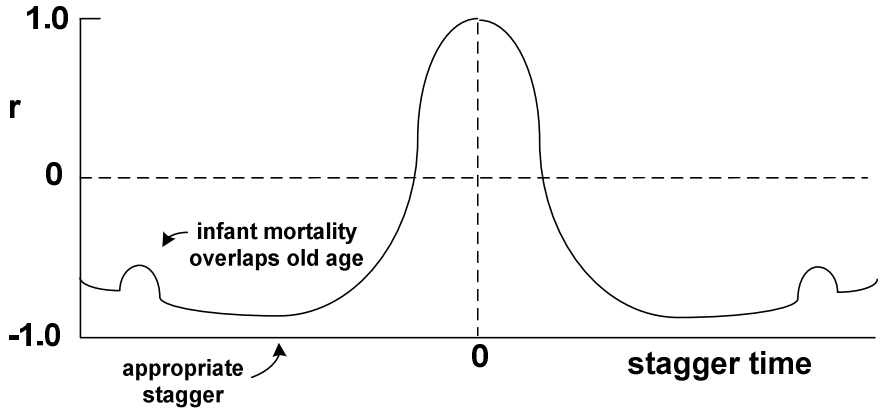

Figure 11: Correlation Coefficient *r* as a Function of Stagger Time *s*

11.

Assuming that the failure probability distributions for System 1 and System 2 are nearly identical, the correlation coefficient, *r*, will be +1.0 when they are started simultaneously (*s* = 0). As their starting times are staggered, the correlation coefficient will decrease. Ultimately, it will increase as the starting times become once again coordinated. The stagger time at which the correlation coefficient reaches its minimum value is the stagger time that will lead to the maximum redundant system reliability.

© 2017 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com
For discussion, contact editor@availabilitydigest.com

Methods for Improving the Availability of Redundant Systems

The availability of a redundant system can be improved by incorporating a Failure Analysis Engine and a Failure Prevention Engine. The Failure Analysis Engine continually monitors the current MTTF of the system. The Failure Prevention Engine performs one or more appropriate actions to increase the availability of the system should its MTTF fall below an acceptable value, as determined by the Failure Analysis Engine.[2]

Failure Analysis Engine

Typical functions for a Failure Analysis Engine include:

1. Determine the probability distribution of failures for each subsystem in the redundant system.

2. Determine the stagger time that leads to the lowest MTTF.

3. With the distributions of the two systems appropriately staggered, determine the probability distribution of failures for the redundant system by multiplying the probability of failure at each instant of time for the two independent systems.

4. Determine the MTTF of the redundant system via the summation methods described in the section entitled “Mean Time to Failure.”

5. Periodically recalculate the MTTF from the current time.

6. If the MTTF falls below a critical threshold, issue a warning message; and inform the Failure Prevention Engine.

7. When the Failure Prevention Engine completes whatever actions it is going to take, return to Step 1.

Failure Prevention Engine

When the Failure Prevention Engine is notified by the Failure Analysis Engine that the system MTTF has fallen below an acceptable threshold, it will take actions to improve the availability of the system by lengthening its MTTF. Typical actions include:

1. Restart a node if it is about to fail due to a software problem.

2. Replace a critical hardware component if hardware failures happen after some period of time (such as a solid-state disk drive).

3. Dispatch a repairman if manual intervention is required.

4. Replace a system in its entirety if it is nearing end-of-life.

5. Restart the nodes according to a staggering schedule if both nodes are apt to fail simultaneously from a software or hardware problem.

6. Reroute users to the node with the lowest probability of failure (i.e., the longest MTTF).

[2] Note that the functions of the described Engines may be accomplished by operator inspection and intervention until a product becomes available that automates them and makes them transparent.

© 2017 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com
For discussion, contact editor@availabilitydigest.com

7. Add another node to the cluster or network of nodes such that the overall probability of failure of the cluster is reduced.

8. Move to an active/active configuration (in which both nodes are processing transactions) or to a sizzling-hot-takeover configuration (an active/active system in which only one node is processing transactions) to lower the recovery time (Recovery Time Objective, or RTO) and the amount of data loss (the Recovery Point Objective, or RPO).

Summary

The prior art for calculating estimated availability from any point in time is flawed because it is based on memoryless random variables. The calculation of the average time to the next failure is always the same regardless of how long a system has been in service. Rather, an alternative term, Mean Time To Failure (MTTF), provides an estimate of the expected time to failure that decreases as time goes on.

The reliability of a redundant system is optimized by minimizing the probability that both systems will fail simultaneously. If they both have the same failure probability distribution, when one system is most likely to fail, so will be the other system.

By staggering the system starting times so that their probability distributions are not aligned, the time that the two systems are most likely to fail will be different. When one system is most likely to fail, the probability that the other system will fail is reduced significantly. Therefore, the probability of a dual system failure is reduced. Redundant system reliability can be greatly enhanced by staggering the starting times of the two systems. This strategy applies both to hardware failures and to software failures.

A method for estimating the remaining availability of a system is to use a Failure Analysis Engine to calculate the system's MTTF from the current time based on the known failure probability distribution of the system. This can be accomplished by sampling the failure probability distribution and summing the probabilities of failure at each of a series of small time intervals. If the MTTF falls below a critical threshold, actions can be taken via a Failure Prevention Engine to mitigate the potential failure based on the expected cause of the failure.

Acknowledgement

The concept of improving availability by staggering system start times was originally conceived by Dr Bruce Holenstein, CEO of Gravic, Inc. Dr. Holenstein is the lead author on this article.

© 2017 Sombers Associates, Inc., and W. H. Highleyman
www.availabilitydigest.com
For discussion, contact editor@availabilitydigest.com