the Availability Digest

www.availabilitydigest.com
@availabilitydig

# Digest Managing Editor Speaks About Staggered Systems

December 2016

Dr. Bill Highleyman, the Managing Editor of the *Availability Digest*, presented two papers at the 2016 NonStop Technical Boot Camp, which was held at the Fairmont Hotel in San Jose, California, U.S.A. from November 13th through November 17th. The Boot Camp was managed by Connect Worldwide, the HPE Technology User Community, and was sponsored by several NonStop vendors. The conference was a resounding success with over 450 attendees. The program featured almost 100 breakout sessions given by HPE, customers, and vendors.

One paper presented by Dr. Bill was entitled "Staggered Systems for Improving Mission-Critical System Availability." This paper was coauthored with Dr. Bruce Holenstein, President of Gravic, Inc. The other paper was "Why Does My Toaster Require High Security?"

"Staggered Systems" explored the improvements that staggering system start times in redundant systems can have on overall system availability. The paper began with an exposure of a fallacy in standard availability theory. It is usual to talk about the MTBF (Mean Time Between Failure) and MTR (Mean Time to Repair) to describe the availability of a system. A system's availability is (MTBF – MTR)/ MTBF, and its probability of failure is MTR/MTBF. However, these parameters are 'random' variables.' This means they have no memory. If MTBF is 1,000 hours, and the system has been in service for 500 hours, its average time to the next failure is still 1,000 hours. If MTR is four hours, and the system has been under repair for two hours, the average time for the repair to complete is still four hours.

A better measure for availability is Mean Time to Failure (MTTF). The "Staggered Systems" paper points out that MTTF is a function of the probability distribution of failure as of the current time. As time goes by, the peak of the failure probability distribution becomes closer; and MTTF becomes smaller – that is, the probability that the system will fail at some point in the future becomes greater, as would be expected.

Highly available and continuously available systems often use active/backup or active/active systems in a redundant pair so that should one system fail, operations can continue with the other system. The databases of the two systems are kept synchronized via data replication.

In an active/backup system, only the active system is processing transactions. Active/backup systems use unidirectional data replication to keep the passive backup system's database synchronized with the active system's database. As database changes are made to the database of the active system, those changes are replicated via unidirectional database replication to the passive backup system.

In an active/active system, both systems are processing transactions. Active/active systems use bidirectional data replication. Whenever a change is made to one database, that change is replicated to the other database. Provision must be made to prevent ping-ponging (sending a change back to the system that initiated it) and data collisions. A data collision occurs if applications in both systems change a common data object at the same time. Those changes are then replicated to the opposite system, where they overwrite the original changes. Now both databases are different, and both are wrong.

It is standard practice in a redundant active/backup or active/active pair to start both systems at the same time. However, this means that the probability of failure distributions of the two systems are aligned. When the probability of failure is high for one system, it is equally high for the other system. This enhances the chances of a dual system failure in which both systems of the redundant pair fail simultaneously.

By staggering the starting times of the two systems, their probability distributions can be misaligned so that there is little correlation between the failure of one system and the failure of the other system. Thus, when one system is likely to fail, the other system in unlikely to fail, thus improving system availability significantly.

The theory of system staggering was presented in the "Staggered Systems" paper. Though it contains a lot of mathematics, it can be read and easily understood by the mathematically challenged, since the concepts are clearly explained without mathematics. This paper is being published in The Connection as a two-part series. The Connection is a journal for the HPE business technology community.

The other paper presented by Dr. Bill, "Why Does My Toaster Require High Security, or The Day the Internet Died," deals with the security issues of the Internet of Things (IoT). 'Things' are typically simple sensors with a small amount of compute capability. The computer supporting a 'thing' manages the 'thing's' sensing capability and transmits its data to a data collection device of some sort. There is little capacity left over for other functions such as security. Therefore, a 'thing' can be easily infected with malware.

This was brought to light on Friday, October 21, 2016, when the Internet suddenly died over a large portion of the United States. The internet was down for two hours in the morning, and then died for a couple of hours at noontime and again in the afternoon. Major web sites went offline, and many users had no Internet access at all.

The problem was a DDoS (Distributed Denial of Service) attack against a major DNS (Domain Name System) server, Dyn. A DNS server is like a telephone book. It converts the URLs that users use to specify web sites to the IP addresses needed to actually communicate with the web sites. The attack prevented Dyn from providing DNS services to any web site which used it and to any users which depended upon it. Since Dyn serviced many major web sites such as NetFlix and CNN, a major part of the Internet was down.

Where did the DDoS attack come from? It turned out that the assault came from a botnet made up of IoT devices infected by the Mirai malware. Mirai is a sophisticated piece of malware that searches the Internet for 'things' that still use default passwords (most do, it appears). It infects these 'things' and creates a botnet of 'things.' Evidently, a botnet comprising hundreds of thousands of things was created in this fashion. The botnet director then commanded the 'things' botnet to attack Dyn, thus taking down a major part of the internet.

An article on this attack will be published in a later issue of The Connection.

Dr. Bill has spoken at every NonStop Boot Camp and its predecessors. Look for his sessions at Boot Camps to come.