the Availability Digest

www.availabilitydigest.com

Help! My Data Center is Down!

Part 7: Lessons Learned

April 2012

In the first six parts of this series, we described several spectacular data-center failures that were caused by a variety of factors – power outages, storage crashes, Internet and intranet failures, upgrades gone wrong, and the actions of IT staff. These stories were taken from the Never Again series published in the *Availability Digest* (www.availabilitydigest.com).

Interestingly, most of these failures could not have been prevented by a better hardware/software infrastructure. In none of the outages was a server failure the root cause. We seem to have learned how to manage redundancy in our server farms quite well. A few outages were caused by dual storage-system crashes. The recovery from intranet failures could have been mitigated with better internal network monitoring.

The predominant cause of the failures was the direct action – or lack of action – by IT staff. In some cases, the failure was directly caused by the actions of an IT staff member. In others, a failure due to another cause was aggravated by human actions. In still others, the failure could have been averted if earlier actions had taken place, such as better testing or documentation. In fact, studies have shown that about 70% of all data-center outages involved at least to some extent the actions of human beings.

When a data center fails, two questions must be answered – how long will it take to recover IT services, and what can be done in the future to prevent a repeat of the failure? In the final part of this series, we review some of the lessons that address these two questions based on what we have learned from the data-center failures that we have discussed.[1]

People Need Redundancy, Too

We spend a lot of money on making our data centers redundant so that they will survive one or more failures. However, we seem to never provide redundancy for the most fault-prone component – the human being.

It is probably obvious to most that there should be multiple staff members skilled in each critical task. Someone must always be available when others are unavailable due to vacation or sickness. We reviewed only one case (the Google Apps Engine outage) in which the outage was extended because of the difficulty in reaching knowledgeable staff.

[1] Our thanks to The Connection for giving us permission to reprint this series of articles.

However, we seem to pay no attention to an equally important human component that needs redundancy – the fat finger. There were an abundance of failures that we described that were caused by a staff member's inappropriate action:

- A State of Virginia technician pulled the wrong controller and crashed a redundant SAN that already had suffered a controller failure.
- A technician with DBS Bank made an unauthorized repair on a redundant SAN and took down both sides.
- A system operator mistakenly deleted the $38 billion Alaska Permanent Fund database and then deleted its backup.
- A maintenance contractor's mistake shut down the Oakland Air Traffic Control Center.
- Thirteen million German web sites went dark when an operator mistakenly uploaded an empty zone file.
- A test technician failed to disable a fire alarm actuator prior to testing the fire suppression system. The resulting siren noise damaged several disks, including the virtual backup disks.
- A system administrator closed all applications on one server in an active/active pair to upgrade it and then shut down the operating server.

All of these outages could have been prevented if a second pair of eyes were checking the actions of the staff member. The lesson is that any critical operation that could have a negative impact on operations should be checked by a second person prior to initiation. Let one person create the command or point to the board to be pulled, and have a second person confirm the action prior to executing it. This is especially important following a component failure that leaves the system with a single point of failure.

Test, Test, and Test

Failover

Testing failover procedures is a time-consuming and risky task that is often avoided by companies. They would rather rely on faith and hope that a failover will be successful following a production-system outage. Too often, faith and hope don't come through. Failover faults are all too common, as experienced by Google, Amazon, and BlackBerry. American Eagle found that not only could it not fail over to its backup system, it could not even fail over to its backup data center.

Backup/Restore

Do you know how long it will take to rebuild your database from backup tapes? You probably don't know if you have never tried it. After multiple failover faults, American Eagle tried to restore its database from magnetic tape and found that it would take an unacceptable amount of time. Clearly, it had never tested tape recovery. The result – its online store was offline for eight days.

Fallback Planning for Failed Upgrades

An upgrade is notorious for causing problems. An upgrade problem is survivable if fallback can be made to the original system so that the upgrade can be corrected. However, much too frequently, there was no fallback plan. When an upgrade caused problems, the systems were down, as experienced by PayPal, Google, and BlackBerry.

Perhaps the most striking example of a failed upgrade with no fallback plan was the experience of the IRS, the U.S. Internal Revenue Service. It confidently decommissioned its old fraud-detection system and disposed of the hardware before it tested a new, upgraded system. The new system failed to work. With no system to which to fall back, fraudulent tax returns went undetected for the next year, costing U.S. taxpayers an estimated $300 million.

A related problem is upgrades uploaded by online services. McAfee sent out an anti-virus update that had not been completely tested and took down millions of PCs worldwide. The lesson – burn in upgrades on a handful of test systems before distributing them widely throughout the organization.

Document Failover and Recovery Procedures

When things go wrong, people get stupider. You cannot rely on memory or good judgment when your staff is under great pressure to correct an outage. They must have good documentation to guide them. The documentation should be simple and brief – there is no time to read a fifty-page document. Most importantly, the documentation must be tested to ensure that it is complete and accurate; and your people must be trained in its use.

Google lost its App Engine services when it could not fail over to its backup data center following damage due to a power outage. It turned out that the failover procedures had changed, and the documentation was not yet up-to-date.

Provide Independent Checking of Installations

There were a few incidents in which an installation was done improperly or was put off and forgotten. It is a good idea to have a separate team thoroughly check an installation following its completion. Keep a log of installation changes that are pending, and have someone responsible for monitoring the log.

Review Security Logs

Speaking of logs, a recent report by Verizon and the U.S. Secret Service reported that the majority of system intrusions by hackers were easily identifiable by information that had been logged, but no one noticed. Many intrusions went on for weeks until some third party notified the company about the fraudulent activity.

Store all security-related information in a SIEM (Security Information and Event Management) system, and use a log-analysis tool to continually monitor the logged data for indications of security breaches.

The Internet Is a Best-Efforts Network

No one says that the Internet is reliable. The Internet goes down all the time, sometimes taking out massive areas for weeks. If Internet communication is the life-blood of your company's services, you must have a backup plan. One viable option is to subscribe to one of the many satellite Internet backup services.

Intranets Are Not Much Better

Many of the outages that we reported were caused by failures in a company's intranet that tied its systems together. Some of these were equipment failures, and many were routing problems.

Intranets can be very complex and expensive. Companies typically do not put the time and effort into making their intranets as redundant as the Internet. Even worse, when an intranet problem does occur, the network is so complex that it can take an unacceptable amount of time to determine where the problem is so that it can be corrected.

This argues for a significant investment in network monitoring so that problems can be detected before they become catastrophic, and outages can be rapidly identified and corrected.

Don’t Trust the Cloud

If you are using the cloud for critical applications, you must concern yourself with the potential loss of your data. You can perhaps survive a few hours of application downtime, but you may not be able to survive the permanent loss of your databases.

It is therefore prudent to maintain a local snapshot of your critical databases on systems under your control. Amazon, WestHost, and T-Mobile’s Sidekick service have all experienced permanent database losses.

A Strange One

When WestHost ran its yearly test of its Inergen fire-suppression system, a technician failed to properly disable the actuators. The result was that the fire suppression system was triggered. Unexpectedly, several disks in the data center were damaged or destroyed. It took six days for WestHost to get its data center back into operation, and much data was lost.

The initial suspicion was that the explosive force of the gas release was the culprit. However, subsequent studies by the manufacturers of the fire-suppression system and of the Inergen gas showed that the disk damage was caused by the ear-splitting sound level from the warning alarms.

Resulting recommendations are to ensure that sirens are not directed at cabinet enclosures and that disk enclosures be sound-proofed. It’s no help if the fire is put out but the data center is destroyed by noise.

And Finally …

Many of the failures could not have been anticipated or avoided. Who would expect that their battery room would explode or that the entire Northeast United States and large areas of Canada would be blacked out? No matter how much redundancy you have built into your system, no matter how well you have your failover recovery procedures documented and your people trained, something is going to happen to take down your data center.

Consequently, your last line of defense must be a good Business Continuity Plan that guides your company along a path of survival when it has lost critical IT services, perhaps for days. And remember, a good BCP must be well-documented and well-tested; and your people must be trained in its use.

Based on the failures that we have seen, the IT portion of the BCP should not focus on the cause of an outage but rather on its impact. It doesn’t make much difference whether your systems have been destroyed by flood or fire. In either case, your systems are down; and what are you going to do about it? After all, would you ever worry about an old lady severing your Internet service because she thought a fiber cable was valuable copper that she could sell? Yes, that happened.