the Availability Digest
Texto

Innocuous Fault Leads to Weeks of Recovery

December 2008

The bank's ultimate horror started with a single disk failure on one node of a three-node, geographically-distributed system. Through a sequence of unimaginable events, this presumably innocuous fault spread through all three processing nodes run by the bank, taking them all down. The international bank suddenly found that its POS and ATM services had come to a halt.

It would take weeks to recover, and full recovery was impossible. Significant amounts of data were lost forever, though some of it was recoverable from other incompatible systems. Manual reconciliation of disputes carried on for months.

The Bank

Established in 1835, the bank that suffered this disaster is an international bank operating in several countries. It is the largest bank in its home country and is a recognized leader in credit- and debit-card transactions. On peak days, it handles over 10,000,000 card transactions.

The System

The bank's system that handles ATM and POS transactions comprises three nodes. They include a production node (PRD), a disaster-recovery node (DR), and a development node (DEV). Because the system is in an active earthquake and volcanic zone, the nodes are geographically separated between two sites.

The production node, PRD, is located at one site. The DR and DEV nodes are resident at a second site 1,000 kilometers away. All nodes are large multi-CPU systems. The PRD and DR nodes each have thirty mirrored pairs of disks to hold the production files and tables.

A Simple Disk Fault

On February 17th, the unimaginable happened. A manageable disk problem grew into a triple system failure situation that took months from which to completely recover.

February 17

20:39: One of sixty data disks on the PRD system fails with a hard disk error. It is the mirror of one of the mirrored pairs. There is no impact on system operation. The system vendor is notified to obtain a disk replacement.

23:55: The vendor's customer engineer (CE) arrives on site with a replacement disk.

February 18

01:01: The faulty disk is replaced. However, one of the CPUs halts with an unrecoverable error. Again, there is no impact on system operation as the other CPUs assume the transaction load.

01:45: A bank technician finds a reference to this problem in the vendor's documentation and calls the vendor's Customer Support.

02:00: Customer Support verifies that the workaround described in the documentation is applicable to this system and recommends that the referenced workaround be applied to the PRD system.

02:30-03:15: The workaround is applied to the PRD system and, after a brief test, to the DR and DEV systems. MISTAKE!

03:30: Sector checksum errors occur on the DR system, which otherwise continues to operate.

03:45: Business applications on the PRD system begin to show faults.

04:08: The PRD system freezes.

04:25-05:20: The PRD system is cold-loaded. However, checksum errors dominate the logs.

05:40: The decision is made to fail over to the DR system.

06:40: The communications network is switched to the DR system and is tested and ready.

09:00-10:40: The production applications are started and are running on the DR system.

11:00-19:00: The DR system experiences sector and block checksum errors.

19:05: Customer Support provides a facility to back out the workaround. The workaround is backed out on the DR machine.

20:00-23:00: The ATM batch run is successful on the DR system, but the POS batch run inexplicably fails.

00:00: The exhausted staff is ordered home to rest until 09:00 the next day.

February 19

05:15: Customer Support verifies that the corruption issues are a workaround problem and pages the bank's staff to apply the backout to the other systems immediately.

05:55: The workaround is backed out on the PRD system.

09:00: The bank's support staff arrives, and the enormity of the issues becomes apparent.

12:20: The primary partitions of the POS log files are moved to a spare disk for safekeeping.
The PRD and DEV systems are unusable and are quarantined.

The production applications continue to run on the DR machine.

The Crash Analysis

As the crisis developed, it began to become clear what was happening. The workaround to cure the original processor fault was an undocumented utility that had been used successfully in the past. Although it had been blessed by Customer Service for use with the bank's version of the operating system, it turned out that this was erroneous. Rather, the workaround was the cause of the sector checksum errors. The likelihood of corruption turned out to be relative to the length of time that the workaround was in place.

Unfortunately, the workaround had been applied to all three of the nodes in the system without first verifying that it worked properly. This caused all of their disks to become corrupted. Even worse, as database updates were being made to the active system, the corrupted data was being replicated to its backup, resulting in no way to recover the data.

Furthermore, when checksum errors reached a certain threshold, the disks automatically went into write-verify mode. In this mode, every block write was read back into memory and verified. This process slowed down disk activity tremendously, causing applications to time out.

Even worse, as system processes were allocated disk extents with corrupted segments, their failure caused CPU halts.

It was later determined that there was no way to recover the corrupted data. The result of this disaster was that much of the transaction data for the POS and ATM systems was unrecoverable. It would take four months to repair the damage.

The Recovery

There were four steps in the recovery process:

- Identify the disk corruption
- Identify the data corruption
- Recover the business data
- Recover the platforms

Disk Corruption

Fortunately, there was a system utility that could verify the sanity of a disk. This utility discovered the following problems:

- PRD – Of the 60 disks on the system (30 mirrored pairs), 40 were corrupted.
- DR – One physical disk was corrupted.
- DEV – One physical disk was corrupted.
- Three mirrored volumes were unrecoverable because their defects tables were filled.

Data Corruption

The multiple disk failures caused the loss of much of the business-transaction data. The unrecoverable files and tables were identified.

Data Recovery

The recovery of lost data was the biggest problem facing the bank. The vendor created a utility that could ignore checksum errors. Bad files and tables were read with this utility, but all data was found to be meaningless.

Much of the POS transactional data was recoverable from the saved POS files. This process, however, took weeks. Settlement and payment functions using this data were successful. Merchants were settled from totals accumulated in surviving files, though the transactional detail was largely lost.

Some lost data could be partially retrieved from other incompatible systems operated by the bank. Still other data was able to be retrieved from the bank's interchange partners.

Remaining disputes were settled manually over an extended period of time.

Resolving data discrepancies was a resource-intensive process. Reflecting the severe load on its personnel, the bank imposed a maximum fifteen-hours per work day for its people.

Platform Recovery

It was found that the corrupted disks could not be used without scrubbing them with a data-clearing utility. It took until March 6th to cleanse all of the PRD disks. Even with disk cleansing, the system vendor had to supply two new mirrored volumes to be installed on PRD. A system disk was created and installed on PRD on March 8th. At this point, the PRD system could finally be returned to service.

During the time that the PRD system was down, a backup system was rented to provide redundancy for the DR system.

Lessons Learned

As a result of this multi-month incident, the bank learned several things. Some of these were that the bank did many things right. It did one thing massively wrong.

Good Disaster Recovery Procedures

Once the decision was made to fail over to the DR system, the failover went smoothly, even in the face of continuing data-corruption problems. The staff realized that a real failover is a lot different than a failover exercise. For one thing, there is no preparation time during which the failover is planned and all pertinent staff is available.

For another, there is no fallback capability should the failover fail. The failover had to work, or the entire system would be down. Up to this point, a disaster-recovery system was a good idea though a bit of a nuisance. It now has proven to be a life saver.

The bank's excellent failover documentation and checklists proved to be the backbone of the successful failover. Frequent testing ensured not only that these were up-to-date but that the DR system was an exact replicate of the PRD system that it was backing up.

Efficient Service Management

A single point of contact had been set up with a service manager to resolve incidents and complaints raised by the users of the system. This facilitated speedy assistance from other bank areas and allowed the technical staff to do its job.

Effective Communication Procedures

Two open conference calls with the technical staff, bank management, and vendor support staff ran for the duration of the recovery.

A management conference line was established to update key stakeholders on the resolution status.

Media exposure was limited to a small, nondescript article regarding application timeouts.

Vendor Support

Once the problem was escalated to the vendor on February 17th, the vendor Customer Support staff maintained an open conference bridge around the clock to support problem resolution. The resident vendor engineers worked tirelessly with the bank's technical staff to resolve problems.

The vendor even supplied additional on-site manpower for several days to allow bank technical staff some time off.

The Hard Lesson – Test, Test, Test

The primary error lay on the shoulders of both the vendor's Customer Support staff and the bank's technical staff. This was the rapid replication of the "workaround" to all systems in the application network. The workaround was supposed to cure a processor halt that occurred on the PRD system shortly after the first defective disk was replaced. What caused this problem was not determined, but in hindsight it probably was a localized problem with the failed CPU.

Firstly, the workaround was undocumented. Though it had been used many times in the past, it appears that it was not thoroughly tested with the operating system version being run by the bank. As was painfully seen, the workaround did not work on the bank's system.

The onerous effects of the workaround took a while to manifest themselves. Disk sector errors did not seem to appear until the workaround had run for a while, or least the operations staff did not see them.

There seems to be no reason that the workaround had to be made to the DR and DEV systems since their CPUs were not exhibiting this problem. In fact, this problem had never been seen up to this point, so why bother to install it on those systems?

But given that decision, the workaround was not thoroughly tested. It was given a cursory test, and the conclusion was that it worked – ship it. Unfortunately, the problems created by the workaround did not surface until it had been running for a while on all systems.

In hindsight, the main lesson to be learned from this incident is rather obvious. When rolling an upgrade through a redundant system, unless it is an emergency, take your time; and thoroughly test it in production on one node. Testing might take days or weeks to achieve sufficient confidence to roll it out to the other nodes. In this way, if the upgrade exhibits problems, a backup node can take over operations; and the faulty upgrade can be rolled back and corrected.

Of course, in this case, the faulty workaround was also causing corrupt data to be replicated to the backup DR system. Maybe the bank was doomed from the start.