the Availability Digest

Major Bank Uses Active/Active to Avoid Hurricanes

October 2007

Headquartered in the Midwest, a major U.S. bank serves much of the eastern United States plus Michigan, Ohio, Kentucky, Indiana, Illinois, and West Virginia. It engages in five main businesses:

- branch banking
- consumer lending
- commercial banking
- investment advising
- processing solutions

The bank's roots go back well over a century. It first opened in 1863 and then grew by acquisitions and mergers to become a major force today in the banking industry.

Card Authorization Services

As part of the bank's processing solutions business, it provides credit and debit card processing for its merchant customers. The bank feels that these services must be highly reliable and available. The services must survive any system failure, no matter the cause, with rapid failover time. This is because should these services fail, users would be denied the use of their credit or debit cards for the duration of the outage.

Therefore, the bank decided to go with highly reliable HP NonStop servers in a two-node active/active configuration to provide these services. One node is located in St. Petersburg, Florida, and the other is located in Grand Rapids, Michigan. This geographical separation ensures that no single environmental disaster, manmade disaster, or system failure will take down both nodes.

Though each node normally handles only one-half of the total processing load, both nodes are configured to handle the entire load so that full transaction processing can continue unimpeded in the event of a node failure.

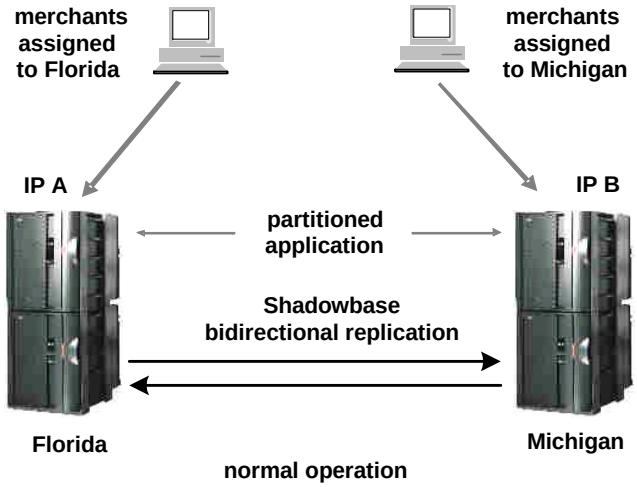
Active/Active with Customer Partitioning

In the bank's active/active configuration, both nodes are always active so long as they are properly functioning and the network connecting them is operational. Both are providing the same set of debit and credit card transaction services to their merchant customers.

In active/active systems, data collisions can occur if two users at two different nodes attempt to update the same row or record at substantially the same time (within the replication latency

period). To prevent data collisions, the bank splits its merchants between the two nodes. Roughly half of the merchants are assigned to the Florida node and half to the Michigan node. Each node has its own set of IP addresses, and each merchant is assigned the IP address of its primary node to use for transaction processing. However, the network is also configured so that each merchant can switch its IP address to the alternate node. In this way, that node can be used as a backup in the event that the merchant's primary node fails.

As transaction processing occurs, database changes at each node are replicated to the other node via the Shadowbase asynchronous bidirectional replication engine (www.gravic.com). Thus, each node contains the entire database for the application network.



Since Shadowbase replicates on a transaction basis, the results of each transaction are committed at the backup database by Shadowbase as soon as that transaction commits on the source database. This keeps the nodal databases in transaction synchronization.

Because the work of each customer merchant is being done on only one node, no data collisions can occur as a result of the asynchronous data replication. That is, in no case will one node be making a change to a row that is also being changed by the other node during the replication latency interval.

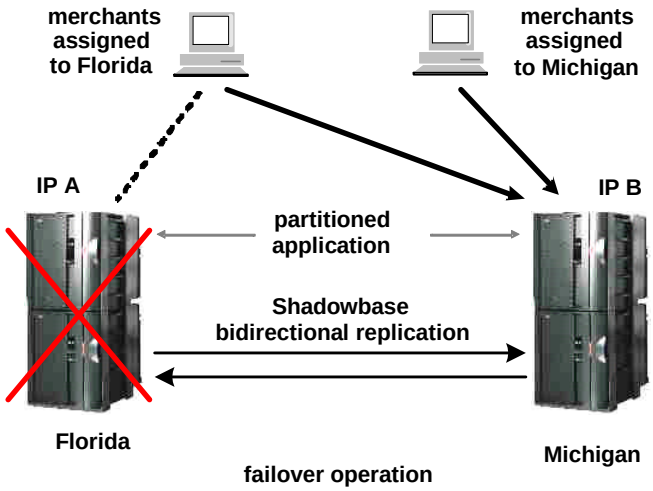
Failover

Should a node experience a failure, the merchants which were assigned to that node as their primary node simply switch their network routing IP address to that of the alternate node. Processing for the failed-over merchants continues on their backup node, uninterrupted from the last completed transactions. Processing continues as usual for those merchants who are primaried on that node.

The bank uses the ease of node failover to its advantage to avoid potential disasters. For instance, whenever Florida is threatened by a hurricane, the bank will instruct all of its customer merchants assigned to the Florida node to switch over to the Michigan node until the hurricane threat has passed.

In 2005, during a particularly intense hurricane season, the bank did this five times. As a result, it avoided any potential system downtime due to the devastating hurricanes Dennis, Katrina, Rita,

and Wilma, each of which pounded Florida with sustained winds of over 130 miles per hour and which caused a combined total of over $120 billion in damage.



Summary

This banking system demonstrates in practice how active/active systems can provide extraordinary availabilities, even in the face of frequent natural disasters.

Furthermore, it proves the ease of switching users to an alternate (backup) node in a very short period of time. This is a very important aspect of active/active systems since it can be used to test failover procedures in a risk-free manner with no impact on the users. Failover testing is one of the most important practices that ensures high availability; yet because of the interruptive and risky nature of failover testing, it is often avoided. Active/active architectures, with their ease of failover, solve this dilemma. Not only is failover fast, but it is reliable since it is known that the other node is up and running. After all, the other node is currently actively processing transactions.

By the same token, the system can be easily load-balanced by simply directing one or more merchant customers to switch over to their alternate system.

The bank's implementation of active/active illustrates several advantages of this architecture:

- Failover is fast. Node outages are transparent to the users.
- Failover testing is simple and reliable, with no impact on the users.
- The system can be easily load-balanced by moving users from one node to another, again with no impact on the users.

The bank has avoided data collisions by partitioning its users. The only downside of this implementation is the potential loss of transactions still in the replication pipeline should a node fail (a common problem with backup systems of any kind when asynchronous data replication is used).[1] However, no transactions are lost as a result of controlled failovers performed for preventive measures, failover testing, or load-balancing.

[1] See our November, 2006, article, Asynchronous Replication Engines, and our December, 2006, article, Synchronous Replication.

It is for these reasons that active/active systems are being used more and more for credit and debit card authorization services, as described in our previous case studies Bank-Verlag – The Active/Active Pioneer and BankServ Goes Active/Active.