

The digest of current topics on Continuous Processing Architectures. More than Business Continuity Planning.

BCP tells you how to *recover* from the effects of downtime.
CPA tells you how to *avoid* the effects of downtime.

In this issue:

Case Studies

[Bank-Verlag – the Active/Active Pioneer](#)

Never Again

[Console Command Takes Down A/A System](#)

Best Practices

[Can 10,000 Chickens Replace Your Tractor?](#)

Active/Active Topics

[Synchronous Replication](#)

Recommended Reading

[The Economics of Linux and Open Source](#)

Product Reviews

[MySQL Clusters Go Active/Active](#)

The Geek Corner

[Calculating Availability – The 3 Rs](#)

Complete articles may be found at
www.availabilitydigest.com.

Dear Reader:

Bank-Verlag of Germany may well be the pioneer in active/active systems. Read about its system in our Case Study.

A fat finger can take down an active/active system, as related in our Never Again story.

Open source and commodity hardware can be much cheaper than mainframes. But do you really want to make that transition? Read about one such experience in our Best Practices article.

Speaking of open source, Martin Fink has explained in great clarity for those managers who are new to open source the intricacies of this new business model. Review his book in Recommended Reading.

Also learn about synchronous replication, MySQL Clusters and their active/active approach, and the impact of system recovery on availability.

Dr. Bill Highleyman, Managing Editor

Case Studies

Bank-Verlag – The Active/Active Pioneer

Wolfgang Breidbach and his colleagues may well be the fathers of active/active systems. They implemented their configuration twenty years ago.

Interestingly, their driving motivation was not initially availability. It was zero downtime migration.

Bank-Verlag is responsible for the production of debit cards for the German banks. The technology of the mid-1980s was to simply keep debit card data on the magnetic stripe of the card, with later batch updates of the customer accounts. There was no online verification of a debit card transaction against the customer's account.

This system worked fine until a TV investigative report showed how easy it was to counterfeit these cards. As a consequence, Bank-Verlag implemented an online debit card processing system on an IBM System 370 so that a debit card transaction could be checked against the corresponding customer account before authorizing the transaction. Later, for uptime reasons, Bank-Verlag switched to a Tandem system and had to migrate the IBM database and applications to the Tandem without denying debit card service to its customers. Thus was born active/active.

Today, Bank-Verlag performs this function on a pair of NonStop NS 16000s using transaction replication in an active/active configuration.

[--more for subscribers--](#)

Never Again

Console Command Takes Down Active/Active System

You have to work hard to take down an active/active system. However, one way to do this is for an operator to erroneously enter a series of commands that adversely affect all systems in the network.

Just such an incident happened to an active/active system that had run for years without an outage. In fact, the system had undergone many rolling upgrades without a planned outage of any sort.

However, during one fateful upgrade, the procedure had been followed flawlessly through the switching of all users to one node and next shutting down the applications on the other node which was to be upgraded. Then came the next step – shutting down the node to be upgraded. Oops! The system manager shut down the wrong node.

[--more for subscribers--](#)

Best Practices

Can 10,000 Chickens Replace Your Tractor?

Fault-tolerant systems are expensive. Commodity hardware and open source are cheap. So why not replace these expensive systems with the latest technology?. After all, with clustered technology, very high system availabilities can be achieved at a much lower cost.

Maybe so. Maybe not. Several organizations have tried this. Some have succeeded. Others have failed after spending a significant amount of money and time on the trial.

Certainly, such a step requires a lot of analysis and planning. The experience of one financial institution which tried this is telling. They replaced four fault-tolerant active/backup pairs with over one hundred industry-standard servers, RAID arrays, routers, and other components.

The result – twice the total cost of ownership and fifty times the failure rate. Not to mention a real administrative headache.

[--more for subscribers--](#)

Active/Active Topics

Synchronous Replication

Synchronous replication solves many of the problems inherent with asynchronous replication. Asynchronous replication introduces a delay, known as *replication latency*, from the time that a source database is updated to the time that the update appears in the target database. Because of replication latency, there is the possibility of data collisions, of data loss following a node failure, and of a compromise in fairness (the simultaneous availability of all data to all users). None of these problems exist with synchronous replication.

However, synchronous replication comes with its own set of problems. The most obvious is the introduction of *application latency*, or the delay in completing a transaction until it has committed across the network. Application latency negatively affects the response times of applications.

In addition, synchronous replication can induce network deadlocks if these are not considered in the application design. Provision must be made to exclude failed nodes from the scope of a transaction and to recover those nodes following their return to service. Certain synchronous replication approaches may require significant application changes.

There are several techniques for synchronous replication, each with its own characteristics. These include network transactions, coordinated commits, and distributed lock management.

[--more for subscribers--](#)

Recommended Reading

The Business and Economics of Linux and Open Source

Free software is gaining ground in the marketplace. By “free,” we don’t mean free of cost. We mean freedom, as in freedom to use, freedom to modify, freedom to distribute. What was originally known as free software is now known as open source since the very nature of “free” means generally available access to the source code.

The free software movement was put into full motion with the introduction of the Linux operating system kernel by Linus Torvalds in 1991. There are currently an estimated 40,000,000 copies of Linux in use, with about half of them being purchased from distributors. Linux is the most ported operating system of all time.

There is much concern in the corporate world about how to deal with this phenomenon. As an end-user, can I trust its use? As a software vendor, how am I going to compete with it?

Martin Fink in his book, *The Business and Economics of Linux and Open Source*, clarifies the mysteries of open source for those in management who want to try it, who want to learn more to see if it is applicable, or who want to fight it. It is a clear and complete discussion on open source, its development and licensing models, its business advantages, and its business disadvantages.

[--more for free--](#)

Product Reviews

MySQL Clusters Go Active/Active

MySQL is the most popular open source database available today, with over 4,000,000 installations. MySQL AB, the developers of the MySQL database, recently announced the availability of MySQL Clusters to provide a highly reliable and fast database.

MySQL Clusters use an active/active architecture to create storage engines that provide five 9s reliability. A storage engine comprises a set of storage node groups, each of which holds a set of tables or table partitions. Each node group can contain up to four storage nodes, all kept in synchronism by synchronous replication.

All databases are memory-resident for very fast access and throughput. Disk checkpoints ensure recoverability of the database in the unlikely event of a total system failure.

Multiple geographically-dispersed MySQL Clusters can be kept in synchronism via MySQL asynchronous replication for disaster tolerance. However, because this replication engine does not support data collision detection and resolution, multiple Clusters are generally configured as a master feeding a set of slaves. The slaves can serve as hot backups or as query nodes in an active/active application network.

[--more for subscribers--](#)

The Geek Corner

Calculating Availability – The Three Rs

In our two previous articles concerning the calculation of active/active system availability, we assumed that once the first node was repaired after a system outage, it was returned to service; and the system was up and running.

However, things are not that simple. There are, in fact, three “r”s to consider – repair, recovery, and restore. First, the node must be *repaired*. Then it must be *recovered*, which can take hours as software is loaded, its database is synchronized, and the node is reintroduced into the network.

At this point, the active/active system is *restored* to service. However, the return of service to the users may be further delayed by other necessary activities. For instance, transactions that had been manually executed during the outage may have to be reentered prior to allowing further online transactions.

Each of these “r”s has its own impact on availability.

[--more for subscribers--](#)

Would you like to Sign Up for the free Digest by Fax?

Simply print out the following form, fill it in, and fax it to:

Availability Digest

+1 908 459 5543

The free Digest, published monthly, provides abbreviated articles for your review.

Access to full article content is by subscription only at

www.availabilitydigest.com/subscribe.

Name: _____

Email Address: _____

Company: _____

Title: _____

Telephone No. _____

Address: _____

The Availability Digest may be distributed freely. Please pass it on to an associate.
Access to most detailed article content requires a subscription,
To sign up for the free Availability Digest or to subscribe, visit www.availabilitydigest.com/subscribe.
To be a reporter (free subscription, visit www.availabilitydigest.com/reporter.
Managing Editor - Dr. Bill Highleyman editor@availabilitydigest.com.
© 2006 Sombers Associates, Inc., and W. H. Highleyman