

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](#)

[CPA at Aqueduct, Belmont, Saratoga](#)

[Never Again](#)

[IRS Goof Costs US Taxpayers \\$300m](#)

[Best Practices](#)

[Availability Best Practices](#)

[Active/Active Topics](#)

[Hardware Replication](#)

[Recommended Reading](#)

[Continuous Availability Design Guide](#)

[Product Reviews](#)

[Beowulf Clustering on Linux](#)

[The Geek Corner](#)

[Hardware/Software Faults](#)

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

Dear Reader:

Our articles this month include a description of one of the earliest CPA systems, a race track totalizator put into service over 40 years ago. It used today's techniques, including triple modular redundancy and database synchronization by lock-stepping transactions.

Did you know that last year, the IRS decommissioned its fraud detection system before its replacement was even tested? The resulting fiasco cost US taxpayers over 300 million dollars.

Read also about best practices from those who have achieved nearly continuous availability as well as a variety of other timely topics.

Many of you are getting the Digest from our unsolicited mailings. However, to avoid spamming you, we will be ending this practice in the near future. To guarantee your continued receipt of the free Digest, be sure to sign up for it on our web site at <http://www.availabilitydigest.com/subscribe.htm>.

Dr. Bill Highleyman, Managing Editor

Case Studies

CPA at Aqueduct, Belmont, and Saratoga Race Tracks

Continuous processing architectures are nothing new. They have been around for decades and have used the same techniques that we use today. We take a look in this article at one of the early continuous processing

architectures implemented and put into production over four decades ago by the New York Racing Association to provide wagering functions at its Aqueduct, Belmont, and Saratoga race tracks.

This system, known as a totalizator system in the horse racing industry, accepted wagers from ticket issuing machines for a variety of pools, such as the Win, Place, Show, and Daily Double pools. Using the amount of money bet on each horse as an indicator of the horse's popularity, the system calculated the odds that each horse would place in the money in each pool and calculated the payoffs following the conclusion of each race. The odds and payoff amounts were posted on a large infield board and several displays around the race track.

The system used triple modular redundancy and kept the system database copies in synchronism by lock-stepped transaction replication.

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

Never Again

IRS Goof Costs U.S. Taxpayers \$300m +

For over a decade, the IRS (the Internal Revenue Service, which is responsible for the collection of taxes in the U.S.) has used computerized facilities to spot tax fraud. As tax laws changed and as perpetrators got smarter, it became obvious that the IRS's original Electronic Fraud Detection System had to be replaced with a significantly upgraded new system.

The new system was to be installed in late 2005 so that it could be used to process 2005 tax returns in 2006. Even though there were several warnings of problems with the new system, the IRS decided to shut down the old system in late 2005 before the new system was ready – before, in fact, it had ever been tested with live data.

Tests at the end of the first quarter of 2006 showed that the new system could not process a day's worth of data in a day. It would never work. By this time, it was too late to revert to the old system. As a result, the IRS lost an estimated \$300 million in fraudulent or improper refunds.

The IRS has since stopped work on the new system and is restoring the old system so that it can process 2006 tax returns in 2007.

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

Best Practices

Availability Best Practices

Availability best practices cover the entire gamut of a data processing operation. They start with the development and use of robust software and hardware and proceed with good operator training and documented procedures. Planned downtime is minimized or eliminated, and repair strategies that lead to minimal repair time are adopted to minimize unplanned downtime.

We can attempt to optimize availability by proper operator training, considered selection of hardware and software components, redundant networks, and so on. But failures will occur.

The secret to high availability is not to eliminate failures but to recover from them quickly. Only if recovery is so rapid that the user does not view it as a denial of service is availability maintained.

In addition to rapid recovery from failures, it is imperative to also eliminate planned downtime. The final step in achieving high availability is to continually monitor the availability characteristics of the system so that operational and recovery procedures can be continually improved.

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

Active/Active Topics

Hardware Replication

Databases can be synchronized at the hardware level. Hardware synchronization involves replicating entire disk data blocks. Whenever a data block is flushed from the cache of the source file system and written to the source system's disk, it is replicated to the target system and written to the target system's disk.

Hardware replication ensures that the source and target disks are copies of each other. Replication is very efficient since only full block writes are made to the target system's disks rather than individual data item updates, as is done with software replication.

However, this does not mean that the source and target databases are synchronized since the target system does not know of the data in the source system's cache which has yet to be flushed to disk. Consequently, the target database cannot be used for processing functions such as query processing or reporting. In addition, target database inconsistency precludes the use of hardware replication for synchronizing databases in an active/active system.

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

Recommended Reading

Continuous Availability Systems Design Guide

The Continuous Availability Systems Design Guide is an excellent coverage of issues that confront an organization interested in moving from a classical computing environment to a continuous availability environment.

It is not a "best practices" book for ensuring continuous availability, though it does touch on many of these topics. Rather, it deals with the decision-making process in determining how to approach the design and evaluation of a continuous availability solution.

This decision-making process starts with evaluating the benefits of a continuous processing environment. It then proceeds with determining the business requirements for availability and with mapping these requirements into data processing requirements. Also considered are the design and costing of the continuous processing solution plus product selection, implementation, and the ongoing maintenance and testing of the availability aspects of the new system.

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

Product Reviews

Penguin Computers Offers Beowulf Clustering on Linux

Clustering can provide high availability and supercomputer-scalable high performance computing at commodity prices. The original Linux clustering software was Beowulf, which was developed by NASA. Though available as open source, Beowulf clustering is offered as a supported product by Penguin Computing along with a line of servers supporting high-performance computing.

Penguin's Scyld ClusterWare provides all the support necessary for high-availability, high-performance computing. The result is a single-system image for system monitoring and management of scalable clusters comprising thousands of compute nodes. All cluster functions are managed from a single Master node.

Automated job scheduling is provided according to user-specified policies.

What makes Penguin unique is that the original developer of Beowulf, Donald Becker, is now Chief Technology Officer of Penguin Computing.

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

The Geek Corner

Calculating Availability – Hardware/Software Faults

In our previous articles on calculating availability, we have assumed that all node failures were caused by hardware faults and required repair. However, in today's systems, hardware is very reliable; and only a small portion of node failures are hardware-induced.

Therefore, it is important to distinguish between node failures caused by hardware faults, which require a repair time and a recovery time, and node failures caused by software faults or operator errors, which require only a recovery time.

As hardware becomes more reliable, there comes a point at which further increases in hardware reliability have no significant impact on system availability. A case in point is the triple modular redundancy (TMR) configuration of HP's new NonStop NSAA servers.

[--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