# “Achieving Century Uptimes”
# An Informational Series on Enterprise Computing

**As Seen in *The Connection*, An ITUG Publication**
**December 2006 – Present**

## About the Authors:

Dr. Bill Highleyman, Paul J. Holenstein, and Dr. Bruce Holenstein, have a combined experience of over 90 years in the implementation of fault-tolerant, highly available computing systems. This experience ranges from the early days of custom redundant systems to today’s fault-tolerant offerings from HP (NonStop) and Stratus.

***Gravic, Inc*.**
Shadowbase Products Group
17 General Warren Blvd.
Malvern, PA 19355
610-647-6250
www.ShadowbaseSoftware.com

# Achieving Century Uptimes

## *Part 5: Modular Redundancy – To Need or Not To Need*

July/August 2007

Dr. Bill Highleyman
Dr. Bruce Holenstein
Paul J. Holenstein

Ever since the introduction of the K-series, HP NonStop systems have used lock-stepped processor pairs to ensure fast-fail operation. The argument for this architecture is compelling. NonStop systems are designed to provide exceptionally high levels of availability for OLTP applications. Equally as important as availability is data integrity. If a sick processor were to be allowed to continue processing, it could potentially corrupt computations and the database. Therefore, a processor problem has to be detected immediately before it can do any damage; and lock-stepping solves this problem. This philosophy has been carried through to the use of the new Itanium processors (even with their extensive internal checking).

However, with the introduction of the Neoview database appliance, HP seems to have reconsidered this position. Though Neoview is based to a large extent on NonStop technology, its architecture does not extend to multiple modular redundancy. Neoview is a massively parallel database engine that can be configured with hundreds of Itanium processors to attack issues in business intelligence. However, these processors are not lock-stepped. Neoview depends upon the internal error checking of the Itanium processors to shut down a sick processor.

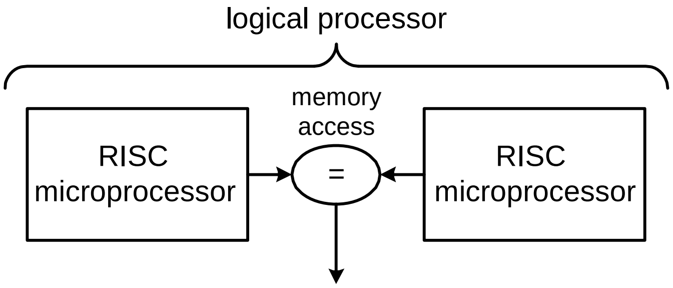
Herein lies an interesting dichotomy. On the one hand, many customers demand and the NonStop folks agree that lock-stepping is important to prevent data corruption. On the other hand, the HP storage people say that fast-fail via lock-stepping is not needed in a major database product. Is modular redundancy needed or not?

We explore this dichotomy below and rationalize the difference in choices made in these two very important areas – OLTP and business intelligence.

### A Review of the NonStop Integrity Modular Architecture

When Tandem introduced the K-series NonStop system, it moved to the use of RISC processors. These processors were very simple and included a minimum of internal error checking. Therefore, in order to achieve fast-fail, a logical processor was constructed using two physical RISC processors in a lock-step configuration.

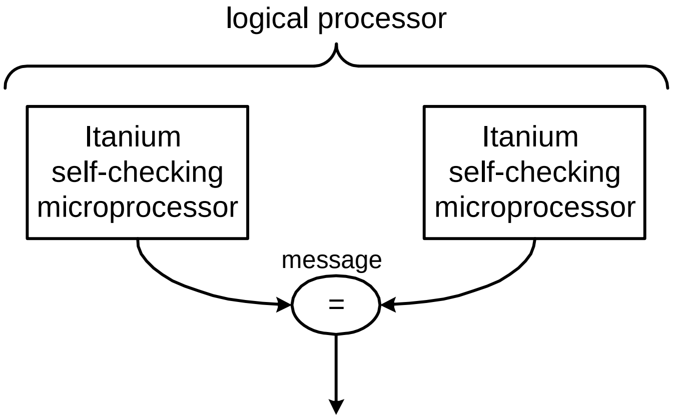
Both physical processors executed the same set of instructions at the same time. Lock-stepping was provided at the memory write level. Whenever a physical processor attempted to write to memory, a comparison was made with the other physical processor. If the data to be written by both was the same, the write was allowed. If the data was different, one physical processor was wrong; and the logical processor was taken out of service. The processes which it had been serving were switched to another logical processor, and operation continued with no downtime to the users.



**The NonStop RISC Logical Processor Architecture**

This logical processor architecture was carried over to the S-series systems. But when the NonStop Integrity systems came along with their Itanium processors, things changed. With their advanced pipe-lining and look-ahead architecture and their extensive internal error detecting and resolution capabilities, the Itanium processors could no longer be guaranteed to execute the same sequence of instructions at the same time. Lock-stepping at the memory level was no longer possible, yet lock stepping was still required.

To accommodate Itanium “lock-stepping”, the logical processor architecture was changed so that lock-stepping was accomplished at the message level rather than at the memory level. Whenever a physical processor was ready to send a message to the outside world (always over ServerNet), it would pause and wait until its companion processor was ready.  If the message to be sent by each physical processor was the same, the message was released to ServerNet. Otherwise, the logical processor comprising these two physical processors was taken out of service. Thus was born HP’s new NonStop Advanced Architecture (NSAA).[1]


logical processor
Itanium self-checking microprocessor
Itanium self-checking microprocessor
message
=

**The NonStop Itanium Logical Processor Architecture**

[1] R. Buckle, W. Highleyman, “*The New NonStop Advanced Architecture: A Massive Jump in Processor Reliability*,” The Connection; September/October, 2003.

This configuration is referred to as dual modular redundancy (DMR). A sick physical processor is immediately detected and taken out of service. As in the RISC architecture described above, this means that the logical processor is taken out of service and the processes which it was serving are failed over to another processor.

A major enhancement was also made by offering as an option triple modular redundancy (TMR). In this configuration, not two but three Itanium processors comprise a logical processor. Now if one fails, it is known which physical processor is trying to send an erroneous message since the two good physical processors will agree on the message to be sent. In this case, the sick physical processor is removed from service, but the logical processor continues in operation in dual modular redundancy mode with the surviving two processors. This architecture saves a failover, maintains the full capacity of the system in the event of a single physical processor failure, and greatly improves the availability of the logical processors (to the point that processor availability is no longer significant in the overall availability equation).

Of course, the NSAA architecture described above will also support singular modular redundancy (SMR), in which a logical processor contains only a single physical processor. After all, this is how a logical DMR processor functions should it lose one of its physical processors. The NonStop folks were reluctant to offer this as a supported product because of its lack of full NonStop protection features. It would not be as reliable as a DMR or TMR configuration since the failure of a single physical processor would take down its logical processor (though its processes would fail over to a surviving processor), and it could not guarantee that a sick physical processor would not corrupt data in the database. However, an SMR configuration is now offered in the smallest NonStop line, the NS1000 line of NonStop servers. HP recommends that this configuration be used only for development and not for operational service.

## An Introduction to Neoview

Neoview, in the simplest of terms, is very similar to a database appliance. It provides a massively parallel SQL database with exceptionally high query performance in a black box that is completely managed remotely by HP's GMCSC support organization.

Applications access Neoview via an ODBC or JDBC interface. Neoview also provides bulk data loading facilities.

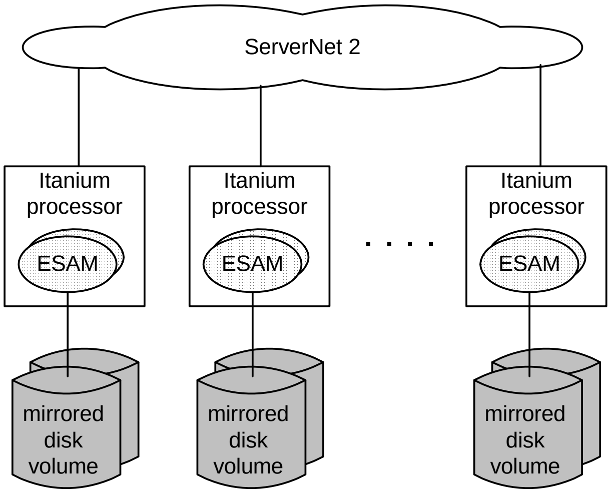
### *Neoview Hardware Architecture*

From what HP has made public, the basic Neoview hardware element is an Itanium processor that manages either a single physical or a pair of mirrored disk volume(s). Each Itanium processor connects to its mirrored pair(s) via dual channels and dual disk controllers. Since the logical Neoview processor is a single physical Itanium processor, there is no lock-stepping available for fast-fail. Neoview depends upon the extensive error checking facilities of the Itanium processor to rapidly take a sick processor out of service.

Each mirrored pair is managed by an Encapsulated SQL Access Manager (ESAM), which is similar to the DP2 disk process in a NonStop server except that it is endowed with a great deal more intelligence, as we will describe later.

The ESAMs are configured as fault-tolerant process pairs running in different processors so that the failure of a processor does not prevent access to the mirrored volume.

Sixteen processors are organized into a *segment*, which is the unit of scalability for Neoview. Initially, Neoview can be configured with up to sixteen segments, or 256 processors with 256 (or 512) mirrored volumes. Each segment can have 3 or 6 terabytes of disk capacity and gives a total Neoview capacity of up to 96 terabytes.



The processors are all interconnected via a redundant ServerNet 2 backbone fabric, which has eight times the capacity of ServerNet. The ServerNet 2 fabric used by Neoview can support up to 1,024 processors. This offers a future 4:1 expansion of capability to Neoview.

### *Neoview Data Architecture*

To provide for massive parallel processing, each SQL table is distributed among all of the disk volumes in the system (except for small tables, which can be located on a single volume). A table is distributed on a row basis via a hashing algorithm on the key for each row. Provision is made to collocate rows and their primary indices on the same disk volume for faster access.

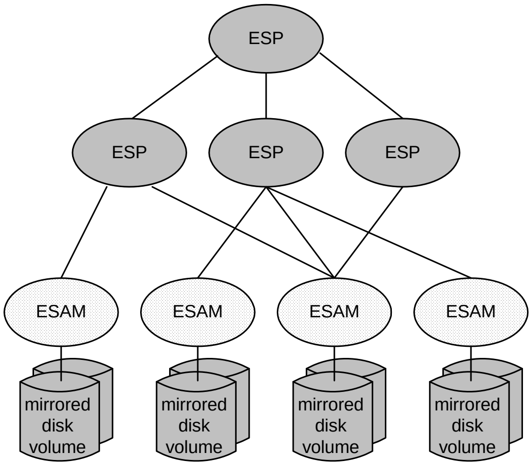
Therefore, for a large Neoview system, a table will be distributed across up to 256 processors and mirrored disk volumes served by up to 256 ESAMs. The processing of a query utilizes the parallel services of all ESAMs. This is the massive parallel processing capability provided by Neoview.

### *Neoview Software Architecture*

In Neoview, a query is handled by an Executive Server Process (ESP). An ESP can invoke the services of other ESPs to help break down a large query into tasks that can be executed in parallel.

The ESPs are responsible for the balanced distribution of the query workload. They handle the parsing of SQL statements, their compilation and statement caching, the consolidation of results, the assignment of priorities, and the logging of query metrics.

The ESPs pass off the real work to the ESAMs. For their particular volumes, the ESAMs provide loading, index maintenance, read/insert/update operations, projections, joins, row locking, aggregation, and sorting. They also provide the more general functions of cache management, RAID 1 (mirroring) management, priority management, and transaction management.



As we have mentioned earlier, the ESAMs are fault-tolerant process pairs. However, the ESPs are persistent processes. Should the processor in which an ESP is running fail, the ESP is lost. It will be automatically restarted in a surviving processor, but any queries in which it was involved will fail and must be restarted.

## The Rational for Neoview SMR

We now return to the architectural dichotomy. NonStop servers use DMR and TMR modular redundancy to guarantee data integrity. Neoview, which is all about data, does not. It uses an SMR configuration. What is missing in this picture?

The answers are several fold:

- With the extensive error checking provided by the Itanium processor, data corruption will almost never happen. Most systems may never see a case of data corruption in their lifetime.

- However, data corruption can happen. In a high-transaction value OLTP system, which is the marketplace for NonStop servers, a corrupt transaction or database can bring the enterprise to a halt. The costs are significant if not fatal. Should this unlikely event happen in a business intelligence data warehouse, opportunities may be lost; but the enterprise will generally not be brought to its knees. The cost of a corrupt business intelligence transaction, while not acceptable, is nevertheless not fatal. Therefore, data corruption must be avoided at all costs for a NonStop server but must be avoided at reasonable cost for a Neoview.

- The primary market for Neoview is a marketplace in which systems are primarily implemented via industry standard servers, not NonStop servers. This marketplace is accustomed to the chance of data corruption and is willing to accept that risk. Enterprises live with this possibility every day with their systems.

- Finally, the cost of providing fast-fail, corruption-free systems for real-time data warehousing is simply not justified at this time.

## Is Neoview NonStop Inside?

One question that is often asked is why isn't Neoview marketed as "NonStop Inside?" After all, isn't Neoview simply SQL/MX running on a NonStop system?

The answer is "no" and "no." As we have described above, though Neoview has drawn a lot from the NonStop Guardian operating system, it does not use standard NonStop hardware. Specifically, modularly redundant lock-stepped processors are not used for fast-fail. Each processor stands on its own and depends upon the error detection capabilities of the Itanium chip to shut it down if there is a processor fault.

Secondly, though the launch pad for Neoview's SQL was SQL/MX, significant changes and BI-specific enhancements have been made. For instance, in order to allow small queries to coexist with large queries, large scans do not flush cache. Priorities are managed differently so that high priority queries will not starve lower priority queries. There is no version checking required since all processes are running the same version of the software.

Can these enhancements be retrofitted to SQL/MX? According to HP, generally no. Many of the enhancements simply do not fit into the NonStop architecture or OLTP focus of a classic NonStop system.

## Summary

Neoview is a massively parallel database appliance derived from NonStop technology. However, its hardware architecture is different; and its SQL engine, though derived from SQL/MX, has been significantly enhanced to support BI-specific features such as very large queries.

Neoview does not embrace the philosophy of the NonStop Advanced Architecture since it does not use multiple modular redundancy (DMR or TMR) for its processors. In its marketplace, single modular redundancy (SMR) is adequate to meet the demands of its users. When coupled with the software fault-tolerance of the other components (e.g. the

ESAM’s), the availability is generally greater than other competing BI products that do not have these capabilities.