IBM® BladeCenter®

Flexible Database Clusters on IBM BladeCenter H with AMD Opteron LS20 Blades and Oracle Database 10*g* Real Application Clusters (RAC)

A Performance Solution by IBM and PolyServe

Phil Horwitz and Martha Centeno
IBM xSeries® Performance Development & Analysis
Research Triangle Park, NC USA

Kevin Closson
PolyServe, Inc.
Beaverton, OR USA

Abstract

The audience for this paper is customers who are interested in implementing a flexible database infrastructure with the IBM BladeCenter® H and Oracle Database®10g Real Application Cluster (RAC) and the Linux® operating system.

The new IBM BladeCenter H delivers increased performance and many new capabilities to the BladeCenter family while maintaining compatibility with the BladeCenter family of products. With a design that's both intelligent and simple, IBM BladeCenter H integrates storage, networking, servers, management and applications.

Oracle Database 10*g* RAC is designed to help build flexible, high-performance, highly available, clustered database solutions on Linux. Connecting such clusters to a fault-resilient Fibre Channel storage area network (SAN) lays the foundation for the computing infrastructure known as Flexible Database Clusters (FDC).

The FDC cluster was the target of a series of tests that took an in-depth look at running and managing not just a single application, but three separate applications. The results of the testing are presented in this white paper.

Introduction

IBM and PolyServe joined forces to build a database cluster using a BladeCenter H running Red Hat Linux and attached it to a SAN configured with more than 84 physical disk drives (see Figure 1). Oracle10g RAC was installed with PolyServe Matrix Server to simplify the deployment and management of RAC and to enable its key features. The entire system was configured and deployed in one standard 42U rack.

The FDC cluster was the target of a series of tests that took an in-depth look at running and managing not just a single application, but three separate applications. The results of the testing confirmed that Flexible Database Clusters provide:

- A means to consolidate and deploy multiple databases and associated applications in a single, easily managed cluster environment

- Simplified management of large database clusters, made possible by PolyServe's Matrix Server clustered file system
- Dynamic "scalability on demand" architecture, enabling near-linear speedup to running applications—with little or no interruption
- Dynamic repurposing of server resources on demand to quickly and easily move processing capacity to where it is needed most.
- An autonomic, always-on operating environment with fast or even immediate self-healing capabilities with little or no performance degradation (and therefore increased utilization rates)
- Lower total cost of ownership (TCO) benefits from improved manageability, scalability, expandability, availability and asset utilization.

Flexible Database Cluster Concepts

Economics of consolidation

The cost-efficiency factors associated with building clusters running Linux with Oracle10g RAC are both proven and substantial. The common method for deploying Oracle on Linux is to create one small cluster for each database application. However, just as consolidating applications on a large SMP system is a proven cost-saving action, consolidating applications on a large cluster can provide great benefits as well, particularly for scale-out applications and middleware such as RAC. Using RAC to create a Flexible Database Cluster can yield the benefits of consolidation, which include reduced administrative overhead and increased flexibility.

Flexible Database Cluster Components

The Flexible Database Cluster requires flexible architectures such as those of the IBM BladeCenter, PolyServe Matrix Server software, and Oracle Database 10*g* RAC. These components complement each other to create a powerful platform for supporting multiple applications.

IBM BladeCenter H Architecture

To support the basic computing infrastructure needed by the Flexible Database Cluster proof-of-concept, it was important to choose a hardware platform that would showcase flexibility and manageability. The IBM BladeCenter H provides both of these attributes.

The BladeCenter H chassis accommodates up to 14 hot-swap 2-socket AMD Opteron™ processor-based blade servers in its innovative 9U form factor. Key infrastructure components such as Layer 2-7 Gigabit Ethernet switching, SAN switching, and centralized management tools are also integrated in the chassis.

Also, the chassis is designed to provide resources, such as power, switch, management and blower modules, which are shared among all the blades. The chassis provides high-speed I/O capabilities for all of the modules, enabling aggregated I/O throughput and reducing the amount of cabling required in the data center. The BladeCenter Management Module facilitates remote access to control the components in the enclosure.

The IBM BladeCenter H enables sites to reduce "server sprawl" and greatly reduce the complexity of their distributed IT infrastructure. Blades also deliver better management software and provide more expansion possibilities with smaller footprint requirements than comparable rack-optimized solutions.

The Flexible Database Cluster proof-of-concept required that large amounts of highly available disk storage be connected to the BladeCenter. The storage subsystem was designed around the IBM TotalStorage® DS4500 storage server. The DS4500 is a RAID storage subsystem that contains Fibre Channel interfaces to connect both the host systems and the disk drive enclosures. With its 2Gbps controllers and high-availability design, the DS4500 delivers the necessary throughput to support the FDC proof-of-concept.

Single points of failure were greatly reduced in the FDC test system through the combination of a fully redundant switched Fibre Channel SAN and the multi-path I/O feature provided by PolyServe Matrix Server.

For more information about the IBM BladeCenter H, visit IBM's Web site:

http://www-1.ibm.com/servers/eserver/bladecenter/

PolyServe Matrix Server

PolyServe Matrix Server enables multiple low-cost Linux- or Windows OS-based servers to function as a single, easy-to-use, highly available system. Matrix Server includes a fully symmetric cluster file system that enables scalable data sharing, high-availability services that increase system uptime and utilization, and cluster and storage management capabilities for managing servers and storage as one. With Matrix Server, customers gain an unparalleled level of scalability, availability, and manageability for scale-out application and middleware deployments such as RAC.

The Matrix Server cluster file system component is both general-purpose and optimized for Oracle. It provided the following advantages in the FDC analysis:

- Improved management of applications and a shared Oracle Home.
- Simple, contained database movement between applications (for example, transportable Tablespace from OLTP to DSS without accessing the network).
- Large database loads with External Tables and Parallel Query.
- Dynamic addition or repurposing of servers to improve throughput and response time for specific workloads.

The "shared Oracle Home" functionality is one of the keys to the Flexible Database Cluster architecture. Matrix Server's cluster file system component supports setting up a single directory for Oracle Home. Oracle needs to be installed only once—on the Matrix Server cluster file system as a single-node install. The single directory Oracle Home is then "converted" to a shared Oracle Home through methodology documented on Oracle's MetaLink Web site, allowing all executables to be stored in one place and all nodes in the cluster to use the same executables. Also, configuration files are located in the Matrix Server cluster file system and can be edited from any node in the cluster.

In addition, of course, Matrix Server can provide "shared home" for applications and middleware, other than Oracle10*g* RAC, running alongside Oracle (or in separate clusters), and can also provide high availability for all of those applications and middleware services.

With shared installation of Oracle10*g* RAC and applications, adding replacement nodes is greatly simplified. The only software that needs to be installed on the added or replaced node is the operating system and the easily installed Matrix Server RPM. Once a node has access to the SAN, it can join the cluster within five minutes. If it were necessary to install Oracle on the private drives of the replacement or added node, the time to join the Oracle10*g* RAC cluster would be in the range of 45 to 60 minutes.

The shared Oracle Home and Rapid Patch Methodology capabilities provided by Matrix Server greatly simplify deployment and management of RAC clusters for single or multiple Oracle workloads, and reduce storage requirements as well.

Matrix Server also provides these additional benefits in an Oracle10*g* RAC environment:

- With Matrix Server, all Oracle files can be stored in the file system. This includes, but is not limited to, the Oracle Cluster Management quorum disk, srvconfig file, control files, online and archived redo logs, data files, imp/exp files, SQL*Loader source files and External Tables.
- Matrix Server provides cluster-wide uniform device naming, which reduces “device slippage” and related problems. Device slippage complicates cluster administration and, if not handled carefully, can threaten to corrupt data.
- Matrix Server enables the Oracle Managed Files feature in an Oracle10g RAC environment. With Oracle Managed Files, the database itself creates and extends tablespaces dynamically, as needed, simplifying database administration.
- Matrix Server enables database tablespaces to be stored in standard file system files, and supports access by standard backup tools and utilities. This permits standard third-party backup tools to be used for Oracle database tables.
- Matrix Server enables Oracle’s External Table feature to allow all cluster nodes to access data stored in flat files.
- Matrix Server permits Extract/Transform/Load (ETL) processes to run in parallel across all nodes in the cluster.
- Matrix Server extends Oracle’s availability capabilities by providing system-wide wellness and failover for applications, middleware, servers, networking and file systems.
- Matrix Server also improves availability by supporting integrated multi-path I/O for multiple Fibre Channel connections from servers to the SAN and multiple switches within the SAN. In such a configuration, all cluster nodes can continue to operate even in the presence of multiple cable, host bus adapter (HBA), or switch failures.
- Matrix Server integrates with fabric access control mechanisms to ensure that only correctly functioning cluster members can access shared data.

For more information on the PolyServe Matrix Server value proposition for Oracle10*g* RAC, visit:
http://www.polyserve.com/ibm/ibm_oracle.html or
http://www.polyserve.com/products_literature.html

Matrix Server Oracle Disk Manager

PolyServe Matrix Server also provides an implementation of the Oracle Disk Manager (ODM) interface. The Matrix Server ODM implementation offers improved datafile integrity through cluster-wide file keys for access and enables Oracle10*g* with asynchronous I/O on the direct I/O mounted file systems where it stores datafiles and other database files such as redo logs. The monitoring capability of Matrix Server ODM is a major benefit in an FDC architecture deployment.

The I/O statistics package of Matrix Server ODM provides I/O performance information at a cluster-wide level (all databases in aggregate), database global level, instance, or node level. Because Matrix Server ODM understands Oracle file, process, and I/O types, it offers specialized reporting that focuses on key Oracle “subsystems” such as the Parallel Query Option (PQO), Log Writer, and Database Writer.

Oracle Database 10*g* RAC

Oracle10*g* RAC can be used in a large, flexible cluster or in the consolidation of multiple Oracle workloads (or clusters) to a single cluster. Oracle10*g* RAC capabilities include:

- Availability— Oracle10*g* RAC is fault-resilient and allows nodes to join an application in the event of a down server.
- Scalability—Applications scale well due in part to Oracle's Cache Fusion technology.
- Flexibility—Multiple Oracle database applications can share a SAN from within a single cluster, reducing administrative overhead, and nodes can be reprovisioned from one application to another.

Proof of Concept

Figure 1 shows the components used for the Flexible Database Cluster test system. The entire environment is configured and deployed in one standard 42U rack.

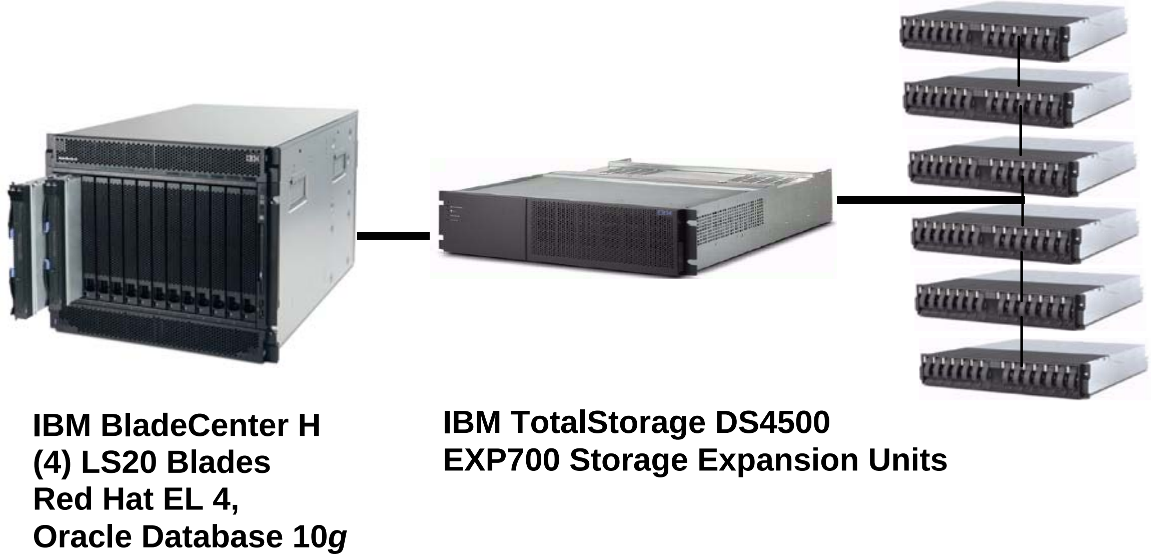


Figure 1. Components of the Flexible Database Cluster System

System Overview

The IBM BladeCenter H chassis was configured as follows:

- **Server nodes:** Four AMD Opteron LS20 for IBM BladeCenter blade servers. These servers are two-way SMP-capable Opteron™ processor-based and are highly scalable. Each LS20 supports single-core processors up to 2.80GHz as well as dual-core processors up to 2.40GHz. AMD Opteron processors offer an Integrated Memory Controller that allow for increased bandwidth and reduced latency. The LS20 supports both 64- and 32-bit applications which allows for a smooth transition to 64-bit enabled applications while leveraging the price and performance of existing applications.
- The architecture of the servers contributed to the high availability of the FDC environment, and, when combined with the multi-path I/O feature provided by Matrix Server, made it possible to build a SAN subsystem with minimal single points of failure.
- **BladeCenter I/O modules:** The 4-port Gigabit Ethernet Switch Module and 2-port Fibre Channel Switch Module provided standard Gigabit Ethernet connectivity and connection to a fault-resilient Fibre Channel SAN.
- **Storage:** IBM TotalStorage DS4500 storage server with eight EXP700 expansion enclosures held a total of (84) 36.4GB HDDs. The IBM Storage Manager software was used to configure the arrays and drives.

The BladeCenter H includes a built-in Web-based GUI that can be used for configuration and management tasks and for viewing system status. It also allows remote access to the BladeCenter to remotely power on and power off blades and to manage I/O modules.

Database Overview

To test the Flexible Database Cluster architecture, a database based on an ERP schema was created in the Matrix Server cluster filesystem using Oracle10*g* Release 2*.* The database contained tables and indexes supporting an Order Entry and Order Fulfillment application as well as customer credit activity tables. The goal was to have a realistic mix of processing running on the system while assessing the manageability, performance and availability of the FDC architecture.

On-Line Transaction Processing (OLTP) Schema

The OLTP schema was based on an order entry system and contained Customers, Orders, Line Items, Product, Warehouse, and Credit Card application tables. The total database size was approximately 1.1 TB.

The application workload accessing the OLTP database connected 200 sessions per node. The nodes under test were evenly loaded. Each user cycled through a set of transactions. At the end of each transaction, the client process slept for a short period of time randomly determined to simulate human interaction.

Decision Support System (DSS) Schema

The DSS workload consisted of analytical queries about customer credit. The fact table used for this decision support was the Credit Card activity table from the OLTP schema which had 2 Billion rows occupying roughly 75GB in the Card Tablespace.

Analysis

The goal of the FDC proof-of-concept was to validate the FDC architecture and to ascertain value-add in key areas such as high availability and “on-demand” scalability. These are the key points learned from the testing:

- The IBM BladeCenter H architecture and technology provide a high-availability platform for the Flexible Database Cluster.
- Capacity can be increased dynamically and transparently, without user interruption, to reduce workload completion time.
- Scalability is directly related to I/O throughput. With the IBM TotalStorage SAN architecture, adding disk drives to the array may resolve performance bottlenecks.

OLTP Workload Logging Analysis

During the proof of concept, the effect of the DS4500 storage array write cache on OLTP throughput was measured.

With Oracle10g, it is possible to start the database instance in a mode where Log Writer I/O is disabled. While this mode would never be used for production purposes[1], it is a convenient way to test the effectiveness of the write-caching functionality of a storage array[2]. The Oracle initialization parameter is called *_disable_logging*. When set to TRUE, the Log Writer process is still posted by server processes to flush the log during a log file sync event, but instead of actually issuing an I/O request, the Log Writer simply marks the flush as complete. Executing the server in this manner establishes a baseline of best theoretical throughput for a given transaction mix[3]. Any

[1] If the database goes down in any manner other than a normal shutdown, the database will be completely unrecoverable.
[2] TotalStorage management software allows enabling and disabling write cache.
[3] Given all other factors remain constant.

wide variance in throughput between this baseline and normal mode (e.g., _disable_logging=TRUE) proves there is a logging I/O bottleneck.

Of course only a workload that exhibits a significant amount of transaction logging would demonstrate any performance difference in such a test, so it is important to point out that the workload selected for this test exhibits a read to write ratio of roughly 65:35 and generated an average of roughly 500 Log Writer I/Os per second at four nodes.

Figure 2 shows the results of the testing. The write-caching technology in the TotalStorage array was so effective that throughput was as good with normal Redo Writer logging as it was with logging disabled. Essentially, the effect of the TotalStorage array cache is no-cost transaction logging I/O for this OLTP workload.

Another very important point this test established is that the PolyServe Cluster Filesystem (CFS) does not impose any performance impact on transaction logging. If the PolyServe CFS, with direct I/O, imposed any overhead at all, the perfect way to expose such overhead is to compare it to not doing the I/O at all.

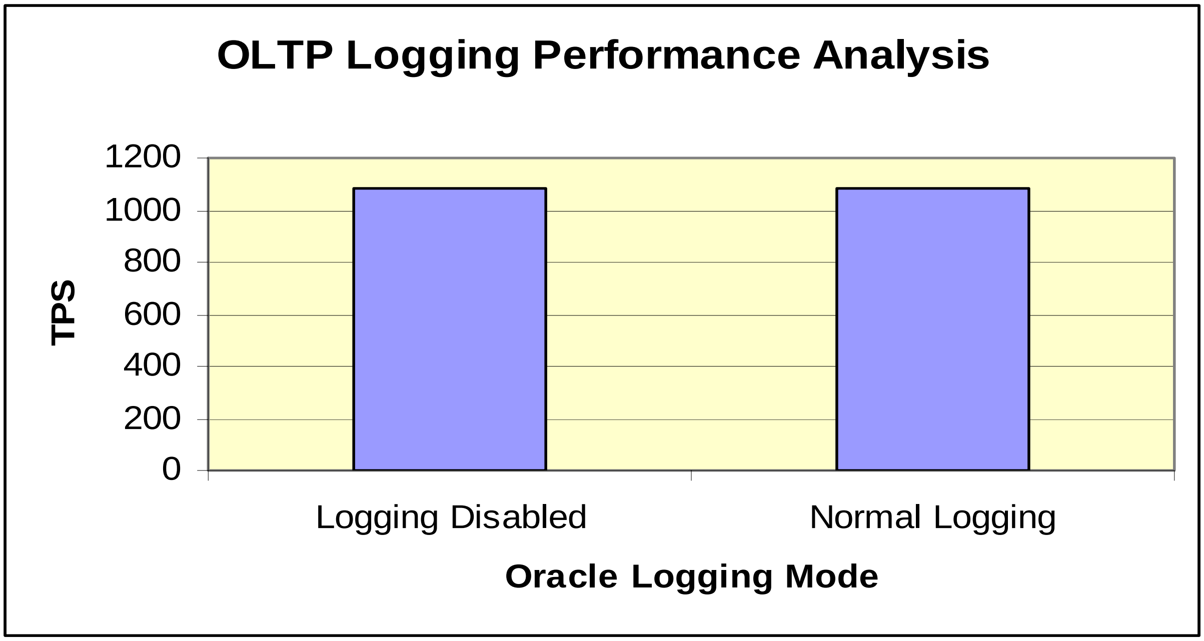


*Figure 2: OLTP throughput comparison. TotalStorage array versus disabled redo logging***.**

High Availability: Fault Injection Testing

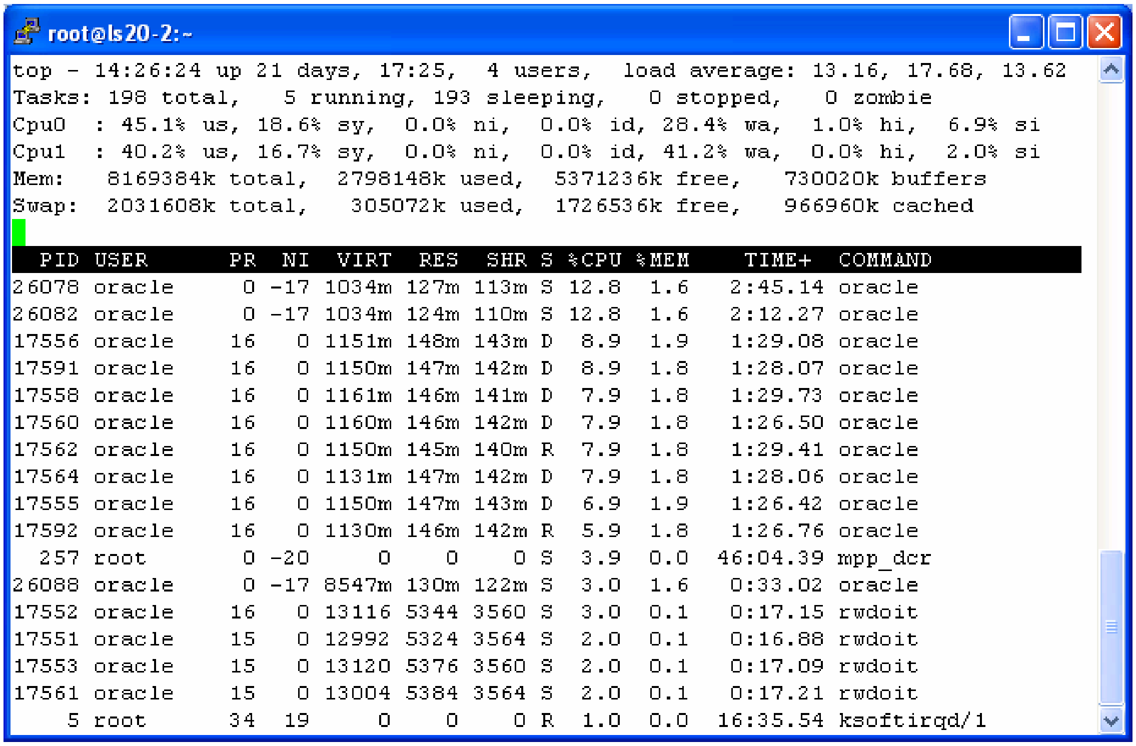
The test proved the ability of Oracle10*g* RAC to handle a crash of one of the 4 nodes, as well as its ability to reprovision a DSS node to take the place of the crashed node. These are highlights of the test:

- Application reconfiguration was unnecessary.
- The crash and replacement of a node were completed transparent to users.
- All operations were completely dynamic.

The true value in any clustered architecture is the ability to respond to system failures in a manner that supports the availability characteristics of Real Application Clusters. To establish the suitability of the PolyServe Database Utility for RAC, a fault injection test with recovery timings was conducted.

Nearly any cluster architecture can recover from a failure suffered at low system utilization levels. The true test of a good cluster architecture is that all the platform components (e.g., operating system, cluster filesystem) are able to respond to a node failure in a timely manner—even under extreme load. When a failure is suffered by one of the servers in a PolyServe Matrix Server cluster, the first thing that must take place is recovery of the cluster filesystem. While the cluster filesystem is fully journalled, there is recovery that must be performed by one of the surviving nodes. If that recovery code is not efficient, long delays will be suffered before Oracle can begin its internal recovery.

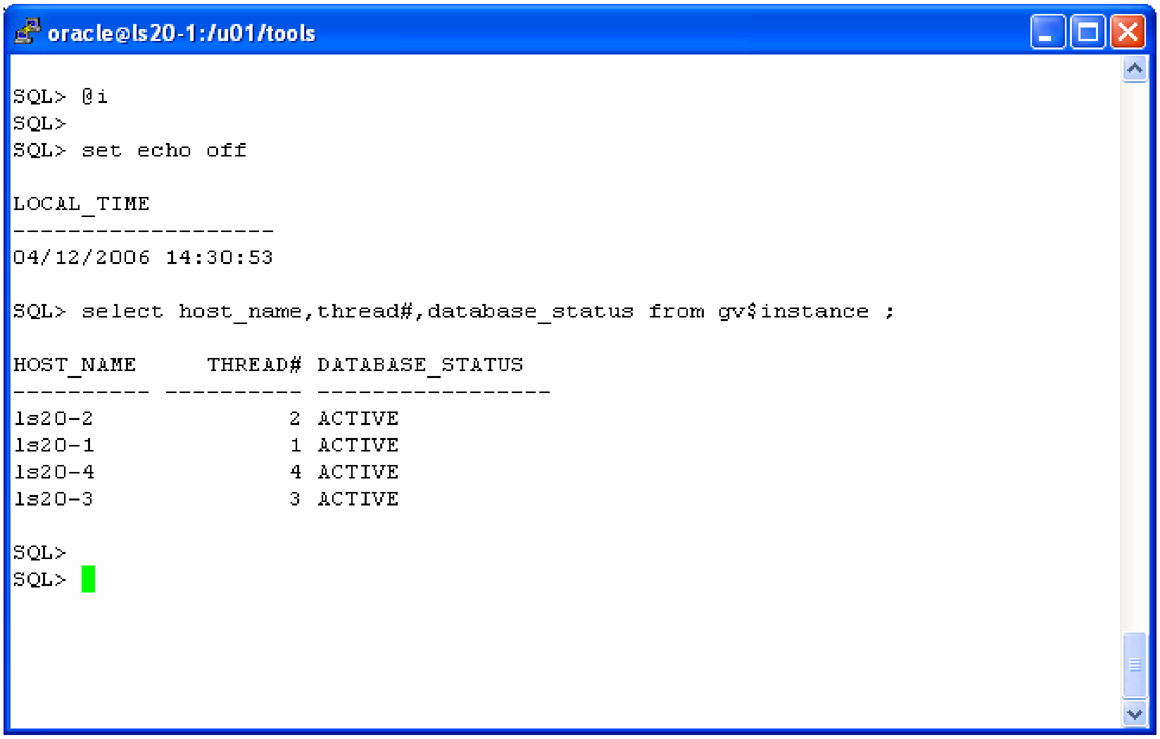
To that end, a test was set up with four RAC nodes executing OLTP at high levels of server utilization. Figure 3 shows a screen shot of the top (1) utility running on one of the servers executing the OLTP workload once it achieved steady state. This is the level of server utilization all nodes were brought to in preparation for the fault injection test. As Figure 3 shows, the load average was over 13—a clearly overloaded server. There's no better situation under which to analyze the ability of a cluster architecture to respond to a failure than under extreme load.



```
top - 14:26:24 up 21 days, 17:25,  4 users,  load average: 13.16, 17.68, 13.62
Tasks: 198 total,   5 running, 193 sleeping,   0 stopped,   0 zombie
Cpu0  : 45.1% us, 18.6% sy,  0.0% ni,  0.0% id, 28.4% wa,  1.0% hi,  6.9% si
Cpu1  : 40.2% us, 16.7% sy,  0.0% ni,  0.0% id, 41.2% wa,  0.0% hi,  2.0% si
Mem:   8169384k total,  2798148k used,  5371236k free,   730020k buffers
Swap:  2031608k total,   305072k used,  1726536k free,   966960k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
26078 oracle     0 -17 1034m 127m 113m S 12.8  1.6   2:45.14 oracle
26082 oracle     0 -17 1034m 124m 110m S 12.8  1.6   2:12.27 oracle
17556 oracle    16   0 1151m 148m 143m D  8.9  1.9   1:29.08 oracle
17591 oracle    16   0 1150m 147m 142m D  8.9  1.8   1:28.07 oracle
17558 oracle    16   0 1161m 146m 141m D  7.9  1.8   1:29.73 oracle
17560 oracle    16   0 1160m 146m 142m D  7.9  1.8   1:26.50 oracle
17562 oracle    16   0 1150m 145m 140m R  7.9  1.8   1:29.41 oracle
17564 oracle    16   0 1131m 147m 142m D  7.9  1.8   1:28.06 oracle
17555 oracle    16   0 1150m 147m 143m D  6.9  1.9   1:26.42 oracle
17592 oracle    16   0 1130m 146m 142m R  5.9  1.8   1:26.76 oracle
  257 root       0 -20     0    0    0 S  3.9  0.0  46:04.39 mpp_dcr
26088 oracle     0 -17 8547m 130m 122m S  3.0  1.6   0:33.02 oracle
17552 oracle    16   0 13116 5344 3560 S  3.0  0.1   0:17.15 rwdoit
17551 oracle    15   0 12992 5324 3564 S  2.0  0.1   0:16.88 rwdoit
17553 oracle    15   0 13120 5376 3560 S  2.0  0.1   0:17.09 rwdoit
17561 oracle    15   0 13004 5384 3564 S  2.0  0.1   0:17.21 rwdoit
    5 root      34  19     0    0    0 R  1.0  0.0  16:35.54 ksoftirqd/1
```

Figure 3: System load prior to fault-injection test.

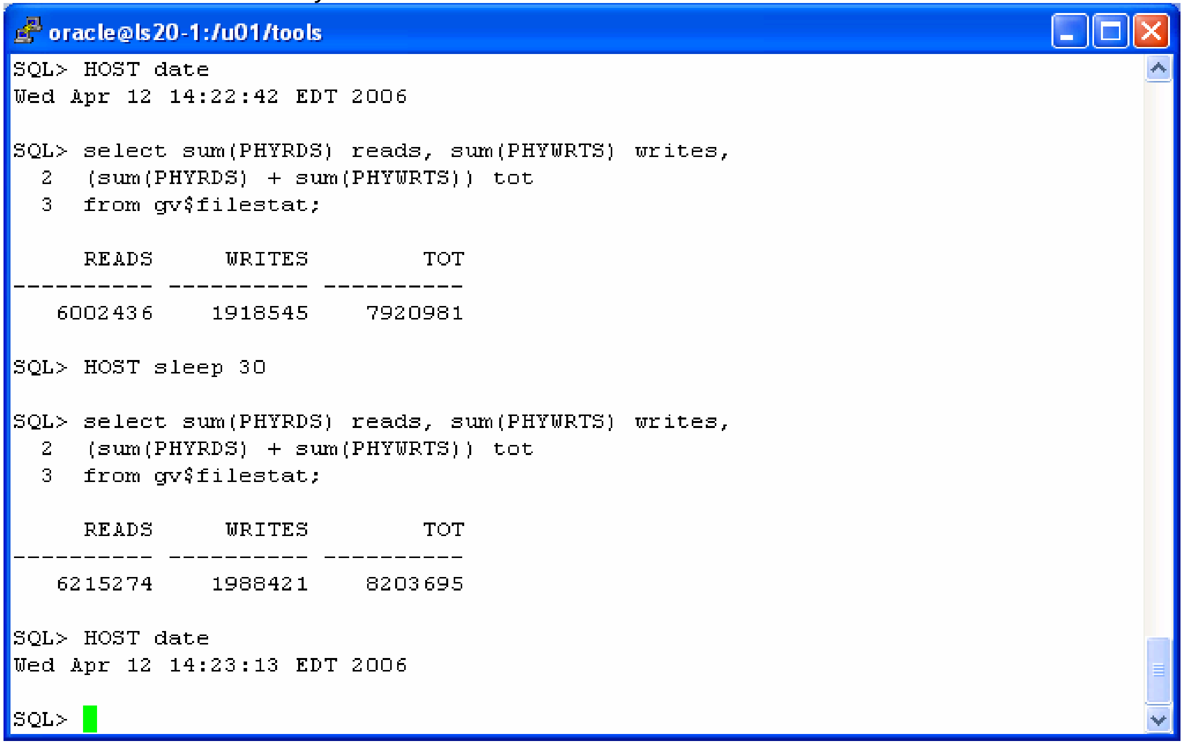
Figure 4 shows the output from the **i.sql** script. There were four RAC instances executing at 14:30:43.



```
SQL> @i
SQL>
SQL> set echo off

LOCAL_TIME
-------------------
04/12/2006 14:30:53

SQL> select host_name,thread#,database_status from gv$instance ;

HOST_NAME     THREAD# DATABASE_STATUS
---------- ---------- -----------------
ls20-2              2 ACTIVE
ls20-1              1 ACTIVE
ls20-4              4 ACTIVE
ls20-3              3 ACTIVE

SQL>
SQL>
```

*Figure 4: RAC instance status at the time of the fault injection test***.**

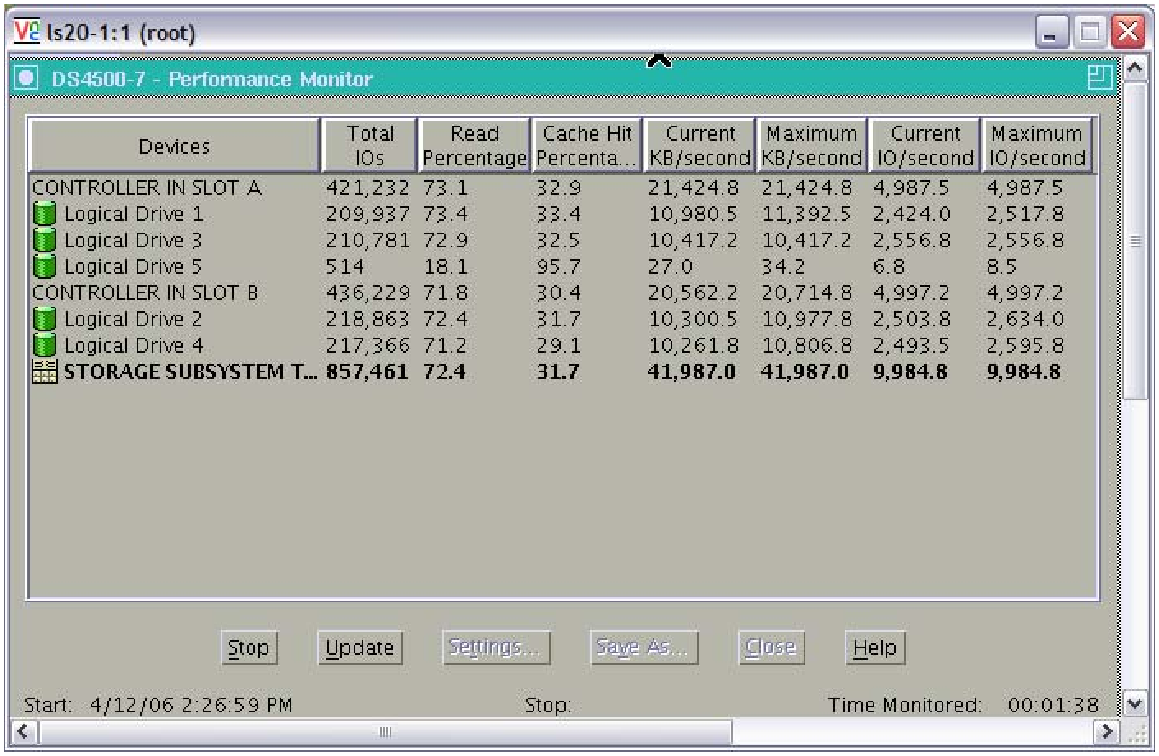
In Figure 5, the **io.sql** script is executed to show that Oracle was executing roughly 9,424 physical I/Os per second cluster-wide at 14:23:13. This establishes the fact that the cluster had reached a saturated steady state.



```
SQL> HOST date
Wed Apr 12 14:22:42 EDT 2006

SQL> select sum(PHYRDS) reads, sum(PHYWRTS) writes,
  2  (sum(PHYRDS) + sum(PHYWRTS)) tot
  3  from gv$filestat;

     READS     WRITES        TOT
---------- ---------- ----------
   6002436    1918545    7920981

SQL> HOST sleep 30

SQL> select sum(PHYRDS) reads, sum(PHYWRTS) writes,
  2  (sum(PHYRDS) + sum(PHYWRTS)) tot
  3  from gv$filestat;

     READS     WRITES        TOT
---------- ---------- ----------
   6215274    1988421    8203695

SQL> HOST date
Wed Apr 12 14:23:13 EDT 2006

SQL>
```

Figure 5: Cluster-wide physical I/O reported by gv$ performance views.

Using the TotalStorage Performance Monitor, it is simple to get a birds-eye view of cluster-wide I/O activity as well. As Figure 6 shows, the I/O rate was 9,985 transfers per second at 14:28:37[4].



Devices	Total IOs	Read Percentage	Cache Hit Percenta...	Current KB/second	Maximum KB/second	Current IO/second	Maximum IO/second
CONTROLLER IN SLOT A	421,232	73.1	32.9	21,424.8	21,424.8	4,987.5	4,987.5
Logical Drive 1	209,937	73.4	33.4	10,980.5	11,392.5	2,424.0	2,517.8
Logical Drive 3	210,781	72.9	32.5	10,417.2	10,417.2	2,556.8	2,556.8
Logical Drive 5	514	18.1	95.7	27.0	34.2	6.8	8.5
CONTROLLER IN SLOT B	436,229	71.8	30.4	20,562.2	20,714.8	4,997.2	4,997.2
Logical Drive 2	218,863	72.4	31.7	10,300.5	10,977.8	2,503.8	2,634.0
Logical Drive 4	217,366	71.2	29.1	10,261.8	10,806.8	2,493.5	2,595.8
STORAGE SUBSYSTEM T...	**857,461**	**72.4**	**31.7**	**41,987.0**	**41,987.0**	**9,984.8**	**9,984.8**

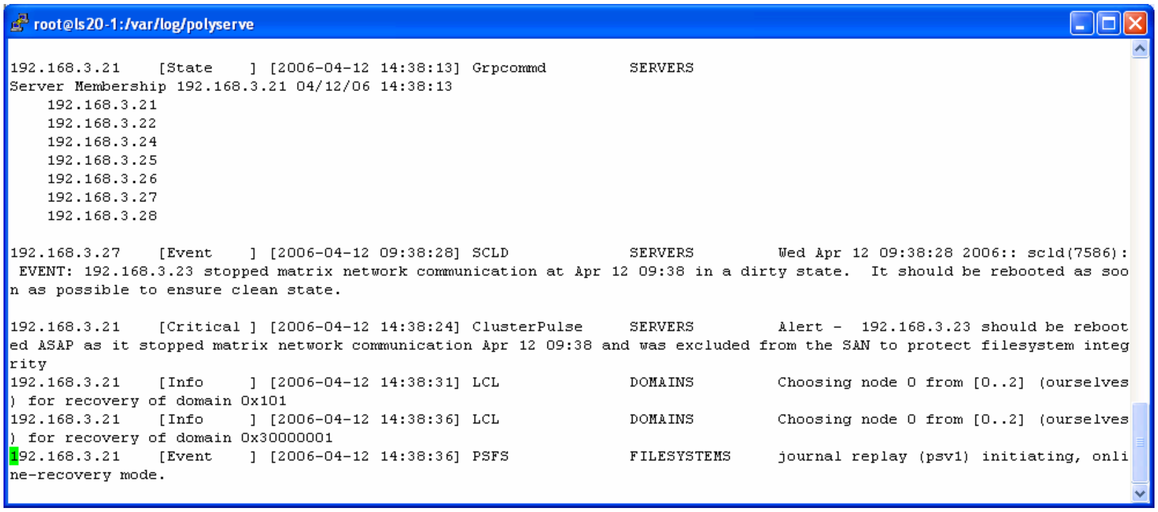
Start: 4/12/06 2:26:59 PM Stop: Time Monitored: 00:01:38

Figure 6: TotalStorage Performance Monitor just before node 3 failed.

Once steady state was archived with the workload, a server failure was simulated by abruptly powering off node 3 of the cluster. Figure 7 shows a clip of the PolyServe Matrix Server matrix.log file from one of the RAC nodes indicating that at 14:38:24, node 3 (192.168.3.23) was fenced out of the PolyServe Matrix cluster.

The clock now begins for how long it takes to resume OLTP processing.

[4] The TotalStorage Performance Monitor shows when the monitoring session started (14:26:59) in the lower left-hand corner and how long the monitoring session has been running (1 minute, 38 seconds) on the lower right-hand corner. This screen shot was taken at 14:28:37.



```
192.168.3.21    [State    ] [2006-04-12 14:38:13] Grpcommd         SERVERS
Server Membership 192.168.3.21 04/12/06 14:38:13
    192.168.3.21
    192.168.3.22
    192.168.3.24
    192.168.3.25
    192.168.3.26
    192.168.3.27
    192.168.3.28

192.168.3.27    [Event    ] [2006-04-12 09:38:28] SCLD             SERVERS         Wed Apr 12 09:38:28 2006:: scld(7586):
 EVENT: 192.168.3.23 stopped matrix network communication at Apr 12 09:38 in a dirty state.  It should be rebooted as soo
n as possible to ensure clean state.

192.168.3.21    [Critical ] [2006-04-12 14:38:24] ClusterPulse     SERVERS         Alert -  192.168.3.23 should be reboot
ed ASAP as it stopped matrix network communication Apr 12 09:38 and was excluded from the SAN to protect filesystem integ
rity
192.168.3.21    [Info     ] [2006-04-12 14:38:31] LCL              DOMAINS         Choosing node 0 from [0..2] (ourselves
) for recovery of domain 0x101
192.168.3.21    [Info     ] [2006-04-12 14:38:36] LCL              DOMAINS         Choosing node 0 from [0..2] (ourselves
) for recovery of domain 0x30000001
192.168.3.21    [Event    ] [2006-04-12 14:38:36] PSFS             FILESYSTEMS     journal replay (psv1) initiating, onli
ne-recovery mode.
```

Figure 7: Node 3 of the cluster went down at 14:38:24 as reported in the PolyServe matrix.log file.

At this point, three forms of recovery need to take place before OLTP processing can resume:

1. **PolyServe Matrix Server Recovery**. PolyServe will replay whatever journal entries there might be in question from the freshly departed node.

2. **Oracle Clusterware Recovery**. Oracle clusterware needs to work out what the state of the cluster is since a node has "died."

   It is important to point out that Oracle10*g* Release 2 Real Application Clusters for Linux and Windows no longer integrates in any way with any host clusterware. Instead, Oracle clusterware and host clusterware run in parallel. Since PolyServe Matrix Server is the lower-level host clusterware[5] executing its cluster-state code in Kernel mode, it will always determine that a node has died and fence it long before Oracle Clusterware realizes the node has died. Since Oracle Clusterware does not integrate with host clusterware, it has to figure out on its own that a node has died.

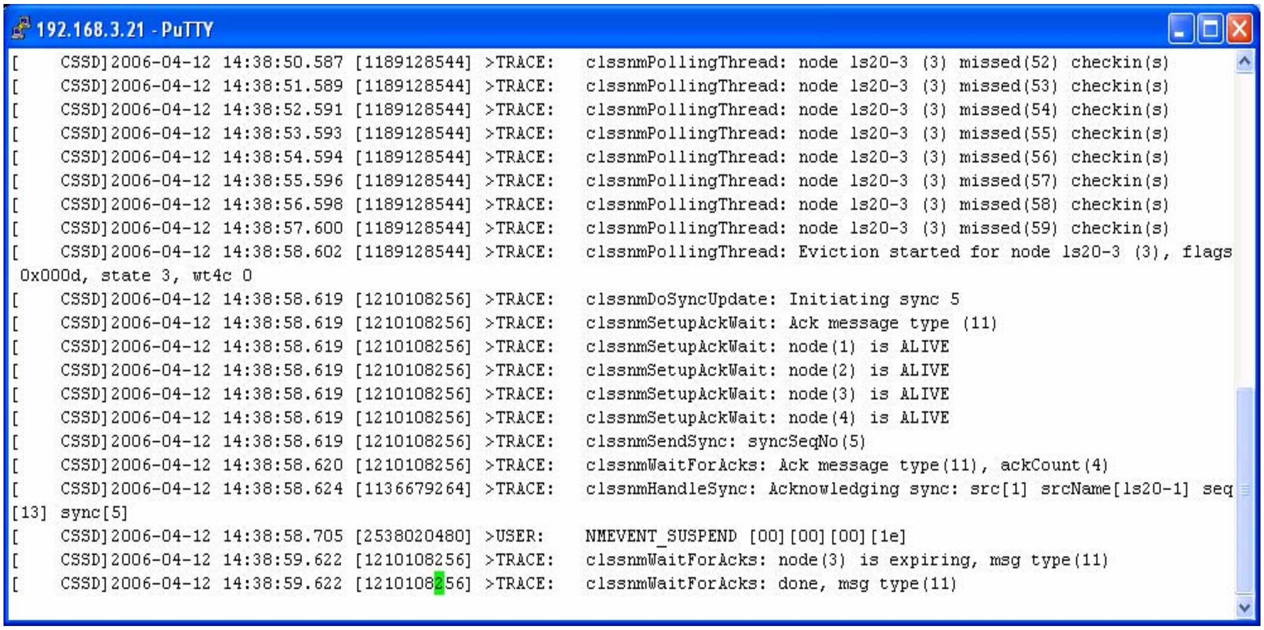
   The method it uses to determine a node has died depends on the manner in which it died, but for simplicity sake, the method is based upon a scheme of nodes "checking in" with the Oracle Cluster Synchronization Services (CSS) master on a regular basis. If a RAC node has not been "checking in" for a tunable amount of time, Oracle clusterware will evict it from the RAC cluster[6]. The tolerance for missed check-ins is tunable and the default is 60 seconds. Tuning this parameter too low can result in false ejections, under extreme system load, so a balance needs to be struck. This is how Oracle Clusterware is architected and has nothing to do with the PolyServe Database Utility for Oracle.

3. **Oracle Instance Recovery**. An Oracle instance died so one of the other instances needs to perform transaction rollback/rollforward for that thread of redo.

Since PolyServe Matrix Server is a fully symmetric and distributed approach to lock management and performs some of the recovery in online-mode, its portion of the overall recovery is very brief. Figure 7 shows that within 12 seconds (14:38:36) PolyServe had entered its recovery in online mode. Oracle processing is able to commence once online recovery begins.

[5] Oracle10*g*R2 on all Unix clusterware (e.g., HACMP) follows this same paradigm.
[6] Oracle evicts a server from the RAC cluster by telling it to reboot itself.

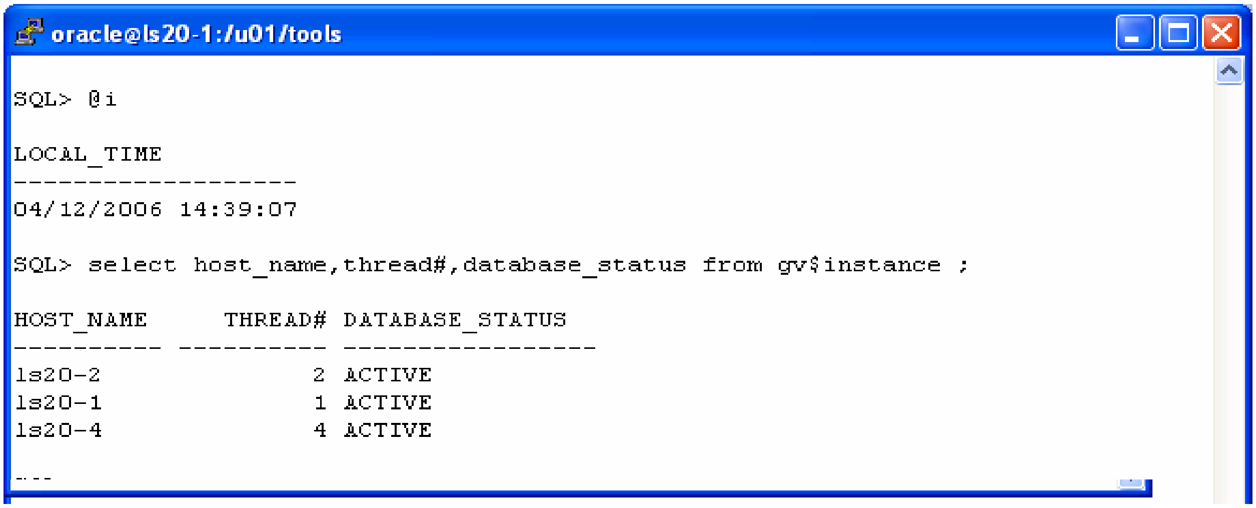
Figure 8 shows that at 14:38:58, Oracle Clusterware determined that node 3 was no longer "alive"[7]. At that point Oracle Clusterware recovery was initiated.



```
[    CSSD]2006-04-12 14:38:50.587 [1189128544] >TRACE:   clssnmPollingThread: node ls20-3 (3) missed(52) checkin(s)
[    CSSD]2006-04-12 14:38:51.589 [1189128544] >TRACE:   clssnmPollingThread: node ls20-3 (3) missed(53) checkin(s)
[    CSSD]2006-04-12 14:38:52.591 [1189128544] >TRACE:   clssnmPollingThread: node ls20-3 (3) missed(54) checkin(s)
[    CSSD]2006-04-12 14:38:53.593 [1189128544] >TRACE:   clssnmPollingThread: node ls20-3 (3) missed(55) checkin(s)
[    CSSD]2006-04-12 14:38:54.594 [1189128544] >TRACE:   clssnmPollingThread: node ls20-3 (3) missed(56) checkin(s)
[    CSSD]2006-04-12 14:38:55.596 [1189128544] >TRACE:   clssnmPollingThread: node ls20-3 (3) missed(57) checkin(s)
[    CSSD]2006-04-12 14:38:56.598 [1189128544] >TRACE:   clssnmPollingThread: node ls20-3 (3) missed(58) checkin(s)
[    CSSD]2006-04-12 14:38:57.600 [1189128544] >TRACE:   clssnmPollingThread: node ls20-3 (3) missed(59) checkin(s)
[    CSSD]2006-04-12 14:38:58.602 [1189128544] >TRACE:   clssnmPollingThread: Eviction started for node ls20-3 (3), flags
 0x000d, state 3, wt4c 0
[    CSSD]2006-04-12 14:38:58.619 [1210108256] >TRACE:   clssnmDoSyncUpdate: Initiating sync 5
[    CSSD]2006-04-12 14:38:58.619 [1210108256] >TRACE:   clssnmSetupAckWait: Ack message type (11)
[    CSSD]2006-04-12 14:38:58.619 [1210108256] >TRACE:   clssnmSetupAckWait: node(1) is ALIVE
[    CSSD]2006-04-12 14:38:58.619 [1210108256] >TRACE:   clssnmSetupAckWait: node(2) is ALIVE
[    CSSD]2006-04-12 14:38:58.619 [1210108256] >TRACE:   clssnmSetupAckWait: node(3) is ALIVE
[    CSSD]2006-04-12 14:38:58.619 [1210108256] >TRACE:   clssnmSetupAckWait: node(4) is ALIVE
[    CSSD]2006-04-12 14:38:58.619 [1210108256] >TRACE:   clssnmSendSync: syncSeqNo(5)
[    CSSD]2006-04-12 14:38:58.620 [1210108256] >TRACE:   clssnmWaitForAcks: Ack message type(11), ackCount(4)
[    CSSD]2006-04-12 14:38:58.624 [1136679264] >TRACE:   clssnmHandleSync: Acknowledging sync: src[1] srcName[ls20-1] seq
[13] sync[5]
[    CSSD]2006-04-12 14:38:58.705 [2538020480] >USER:    NMEVENT_SUSPEND [00][00][00][1e]
[    CSSD]2006-04-12 14:38:59.622 [1210108256] >TRACE:   clssnmWaitForAcks: node(3) is expiring, msg type(11)
[    CSSD]2006-04-12 14:38:59.622 [1210108256] >TRACE:   clssnmWaitForAcks: done, msg type(11)
```

Figure 8: Oracle Clusterware evicts node 3 at 14:38:58 as per the ocssd.log file.

Ultimately, these individual levels of recovery don't matter much. The important thing is how long it takes for OLTP transactions to resume. Figure 9 shows a screen shot of a session that executed the **i.sql** script at 14:39:07 to find that there were three instances online. By this point, 63 seconds had passed since node 3 died. It so happens that 14:39:07 was when the **i.sql** script was executed but that doesn't accurately depict when transactions resumed. A look at the TotalStorage Performance Monitor is actually better for that task.

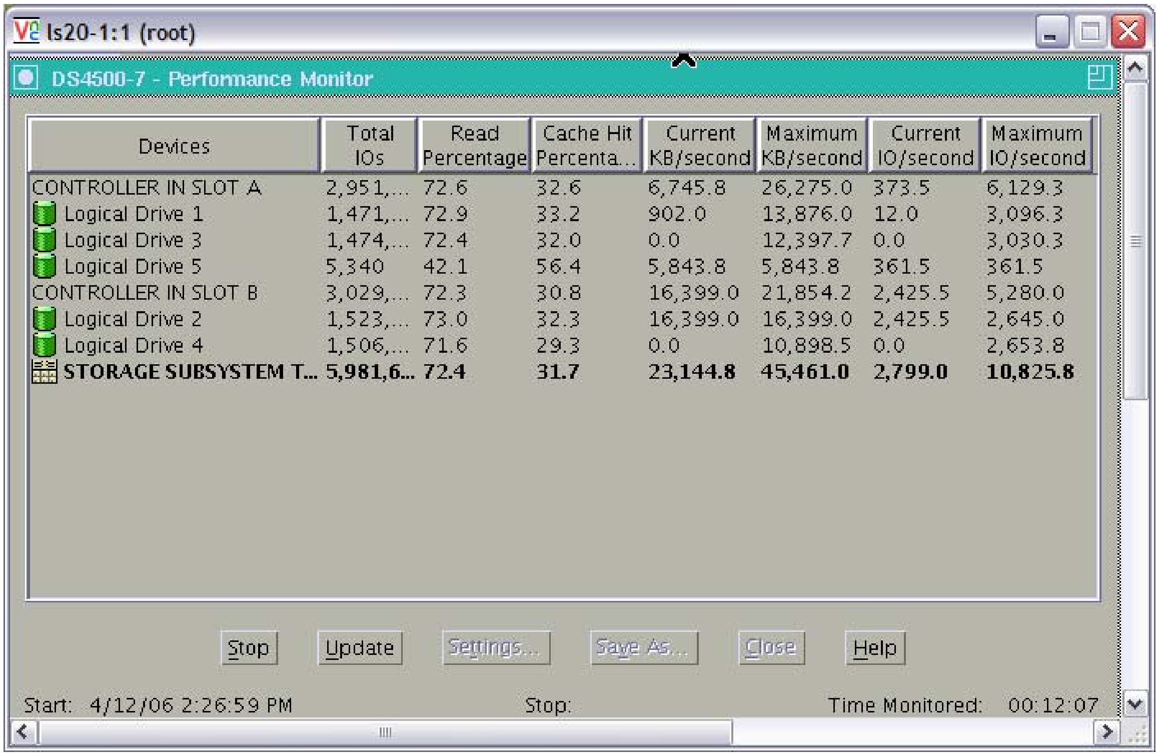


```
SQL> @i

LOCAL_TIME
-------------------
04/12/2006 14:39:07

SQL> select host_name,thread#,database_status from gv$instance ;

HOST_NAME     THREAD# DATABASE_STATUS
---------- ---------- -----------------
ls20-2              2 ACTIVE
ls20-1              1 ACTIVE
ls20-4              4 ACTIVE
```

Figure 9: After recovery, the gv$instance view reports three surviving nodes.

[7] Oracle Clusterware has tunable parameters to reduce the number of missed check-ins. Use the crsctl command to investigate the default, which is 60 with 10.2.0.1 on Linux.

Figure 10 shows a screen shot taken at 14:39:06[8] that shows cluster-wide RAC I/O requests had already ramped up to 2,799 per second. The database had been available for at least a few moments before this screen shot was taken in order to ramp back up to this I/O level.



ls20-1:1 (root)

DS4500-7 - Performance Monitor

Devices	Total IOs	Read Percentage	Cache Hit Percenta...	Current KB/second	Maximum KB/second	Current IO/second	Maximum IO/second
CONTROLLER IN SLOT A	2,951,...	72.6	32.6	6,745.8	26,275.0	373.5	6,129.3
Logical Drive 1	1,471,...	72.9	33.2	902.0	13,876.0	12.0	3,096.3
Logical Drive 3	1,474,...	72.4	32.0	0.0	12,397.7	0.0	3,030.3
Logical Drive 5	5,340	42.1	56.4	5,843.8	5,843.8	361.5	361.5
CONTROLLER IN SLOT B	3,029,...	72.3	30.8	16,399.0	21,854.2	2,425.5	5,280.0
Logical Drive 2	1,523,...	73.0	32.3	16,399.0	16,399.0	2,425.5	2,645.0
Logical Drive 4	1,506,...	71.6	29.3	0.0	10,898.5	0.0	2,653.8
STORAGE SUBSYSTEM T...	**5,981,6...**	**72.4**	**31.7**	**23,144.8**	**45,461.0**	**2,799.0**	**10,825.8**

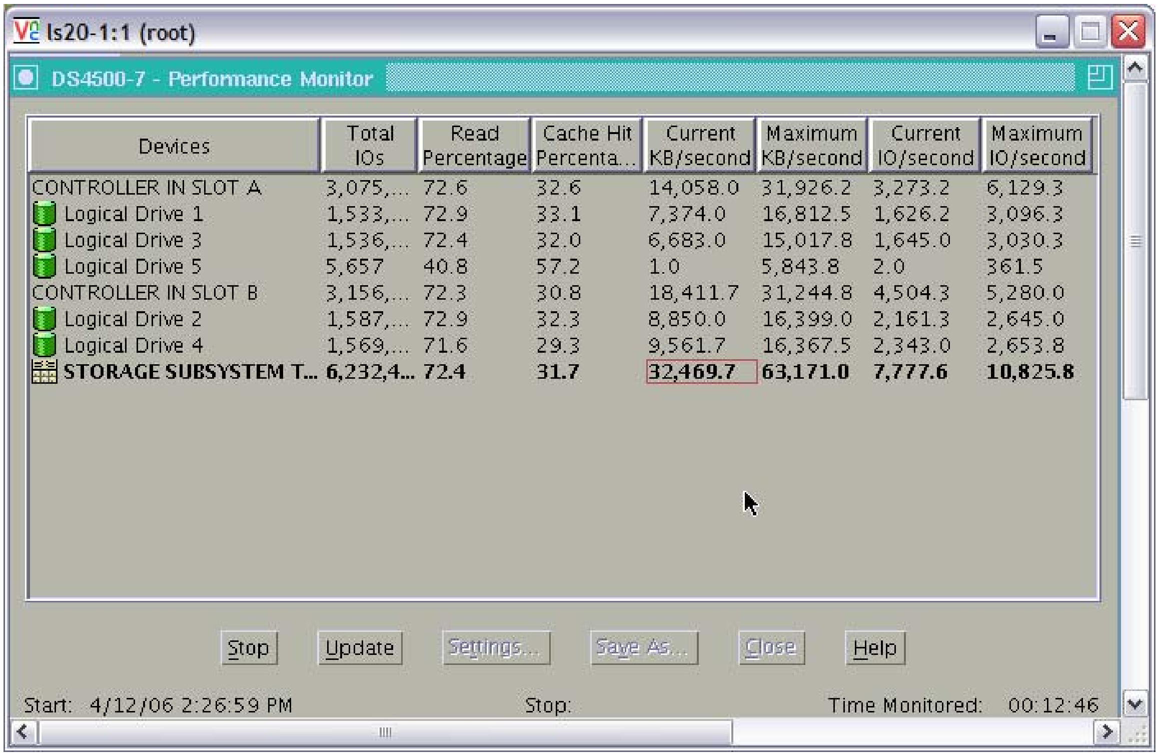
Start: 4/12/06 2:26:59 PM Stop: Time Monitored: 00:12:07

Figure 10: The TotalStorage Performance Monitor shows I/O ramped back up to 2,799 IOps within 42 seconds of the fault injection.

Figure 10 shows that the I/O was only ramping back up at 14:39:06. Figure 11, on the other hand, shows that within 69 seconds of node 3's dying (14:39:33[9]), the cluster-wide I/O rate had ramped back up to 7,777 I/Os per second, which represents 77% of the I/O rate attained by four nodes. So, PolyServe was able to recover and Oracle was able to work through its 60-second CSS strategy and OLTP I/O resumed to the expected 3-node level all within roughly 69 seconds. There is little doubt that tuning the Oracle Clusterware timeout tolerances would speed this up. The point is the cluster architecture recovered after the loss of a node under heavy load.

[8] The monitoring session commenced at 15:26:59 and this screen shot was taken 12 minutes and 6 seconds after the session started—14:39:06.

[9] The monitoring session commenced at 15:26:59 and this screen shot was taken 12 minutes and 46 seconds after the session started—14:39:33.



Devices	Total IOs	Read Percentage	Cache Hit Percenta...	Current KB/second	Maximum KB/second	Current IO/second	Maximum IO/second
CONTROLLER IN SLOT A	3,075,...	72.6	32.6	14,058.0	31,926.2	3,273.2	6,129.3
Logical Drive 1	1,533,...	72.9	33.1	7,374.0	16,812.5	1,626.2	3,096.3
Logical Drive 3	1,536,...	72.4	32.0	6,683.0	15,017.8	1,645.0	3,030.3
Logical Drive 5	5,657	40.8	57.2	1.0	5,843.8	2.0	361.5
CONTROLLER IN SLOT B	3,156,...	72.3	30.8	18,411.7	31,244.8	4,504.3	5,280.0
Logical Drive 2	1,587,...	72.9	32.3	8,850.0	16,399.0	2,161.3	2,645.0
Logical Drive 4	1,569,...	71.6	29.3	9,561.7	16,367.5	2,343.0	2,653.8
STORAGE SUBSYSTEM T...	**6,232,4...**	**72.4**	**31.7**	**32,469.7**	**63,171.0**	**7,777.6**	**10,825.8**

Start: 4/12/06 2:26:59 PM Stop: Time Monitored: 00:12:46

Figure 11: TotalStorage Performance Monitor shows I/O ramped back up to the expected rate for the three surviving nodes.

Fault Resilience Summary

By powering off one of the nodes in the cluster at peak OLTP throughput, this test established the resiliency of Real Application Clusters in the PolyServe Database Utility for Oracle. Without any tuning, the only effect a user would observe would be a 12-second pause of the PolyServe cluster filesystem access. From that point, only 30 seconds had passed until physical I/O resumed to roughly 28% of the pre-failure level of 9,984 I/Os per second. All told, 69 seconds after the loss of node 3, OLTP throughput had resumed to pre-failure level on all three remaining nodes of the cluster. This was a very respectable result obtained without any specialized tuning of Oracle clusterware as well. Most importantly, the sessions attached to the Oracle instances on the three remaining nodes were not disconnected when node 3 failed. Instead, they remained connected and their transactions resumed as soon as Oracle recovery had completed.

Scale on Demand, Dynamically and Transparently

Decision Support System (DSS): Lightweight Scan

To test the performance characteristics of the TotalStorage DS4500 SAN, as configured, an Oracle Parallel Query Option (PQO) lightweight scan test was conducted.

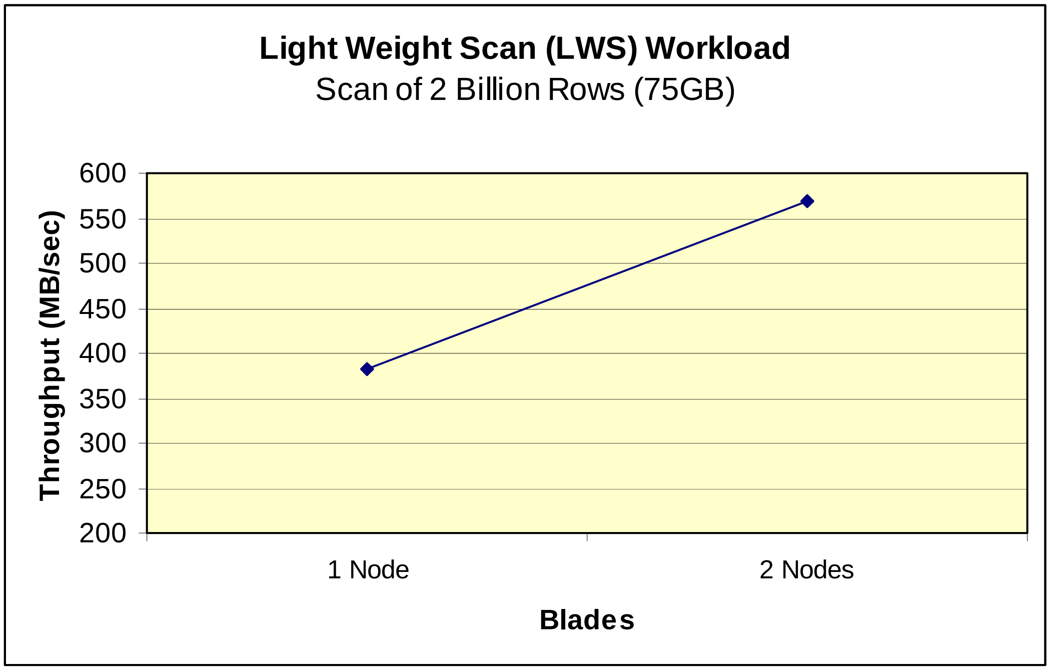
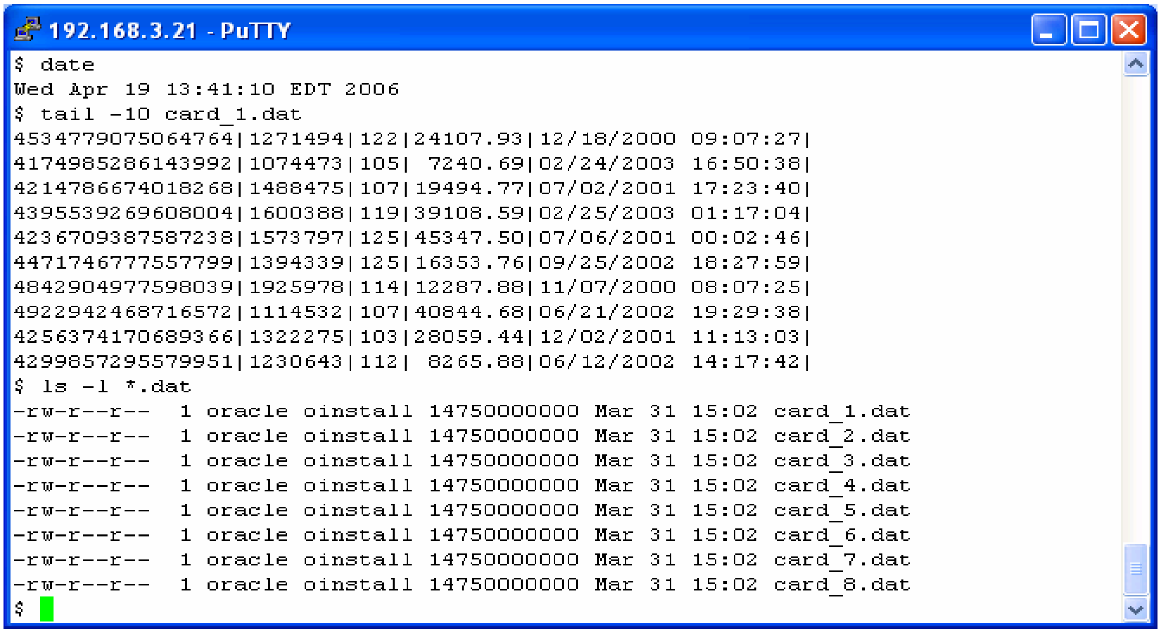


Figure 12. Decision Support System (DSS): Light Weight Scan (LWS) Workload

The lightweight scan consisted of a *select count(*)* from the 2-billion-row credit card transaction table. With Parallel Query Option the I/Os are direct-path reads. Direct-path reads are asynchronous 1MB sequential I/O operations, buffered in the PGA, with four or eight requests in-flight at a time per parallel query slave. Parallel Query lightweight scans are the simplest way to test the base scan throughput the Oracle server can achieve on a given hardware configuration.

Decision Support System (DSS): Data Loading Throughput

The loading of the Card table was also measured. The test consisted of a timed loading of the 2 billion pipe-delimited records of data from flat files into the Card table. Both the flat files and the target Card tablespace resided in the PolyServe Matrix Serve cluster filesystem. The single node test consisted of eight streams of SQL*Loader each loading an equal amount of flat file data At four nodes there were two concurrent SQL*Loader processes per node and on four nodes, two loader streams executed concurrently on each node. The records were roughly 54 bytes in length. The flat files required roughly 110GB filesystem space in aggregate. Once inserted, the table required approximately 75GB space. Figure 13 shows a cluster filesystem listing of the 8 flat files as well as a display of the last few records in one of the files.



```
$ date
Wed Apr 19 13:41:10 EDT 2006
$ tail -10 card_1.dat
4534779075064764|1271494|122|24107.93|12/18/2000 09:07:27|
4174985286143992|1074473|105| 7240.69|02/24/2003 16:50:38|
4214786674018268|1488475|107|19494.77|07/02/2001 17:23:40|
4395539269608004|1600388|119|39108.59|02/25/2003 01:17:04|
4236709387587238|1573797|125|45347.50|07/06/2001 00:02:46|
4471746777557799|1394339|125|16353.76|09/25/2002 18:27:59|
4842904977598039|1925978|114|12287.88|11/07/2000 08:07:25|
4922942468716572|1114532|107|40844.68|06/21/2002 19:29:38|
4256374170689366|1322275|103|28059.44|12/02/2001 11:13:03|
4299857295579951|1230643|112| 8265.88|06/12/2002 14:17:42|
$ ls -l *.dat
-rw-r--r--  1 oracle oinstall 14750000000 Mar 31 15:02 card_1.dat
-rw-r--r--  1 oracle oinstall 14750000000 Mar 31 15:02 card_2.dat
-rw-r--r--  1 oracle oinstall 14750000000 Mar 31 15:02 card_3.dat
-rw-r--r--  1 oracle oinstall 14750000000 Mar 31 15:02 card_4.dat
-rw-r--r--  1 oracle oinstall 14750000000 Mar 31 15:02 card_5.dat
-rw-r--r--  1 oracle oinstall 14750000000 Mar 31 15:02 card_6.dat
-rw-r--r--  1 oracle oinstall 14750000000 Mar 31 15:02 card_7.dat
-rw-r--r--  1 oracle oinstall 14750000000 Mar 31 15:02 card_8.dat
$ 
```

Figure 13. Cluster filesystem listing.

The LS20 blades data loading times exhibited good scalability. Figure 14 shows that bulk data loading with the LS20 blades scales at 82% from one node to four nodes. The insertion rate at one node was 564,972 rows per second with a complete time of 59 minutes.. Adding a second node and executing the test again resulted in 1,075,269 rows inserted per second with a completion time of 31 minutes—95% scalability. Finally, the test was executed on four nodes where 1,851,852 rows per second were inserted for a completion time of 18 minutes. .

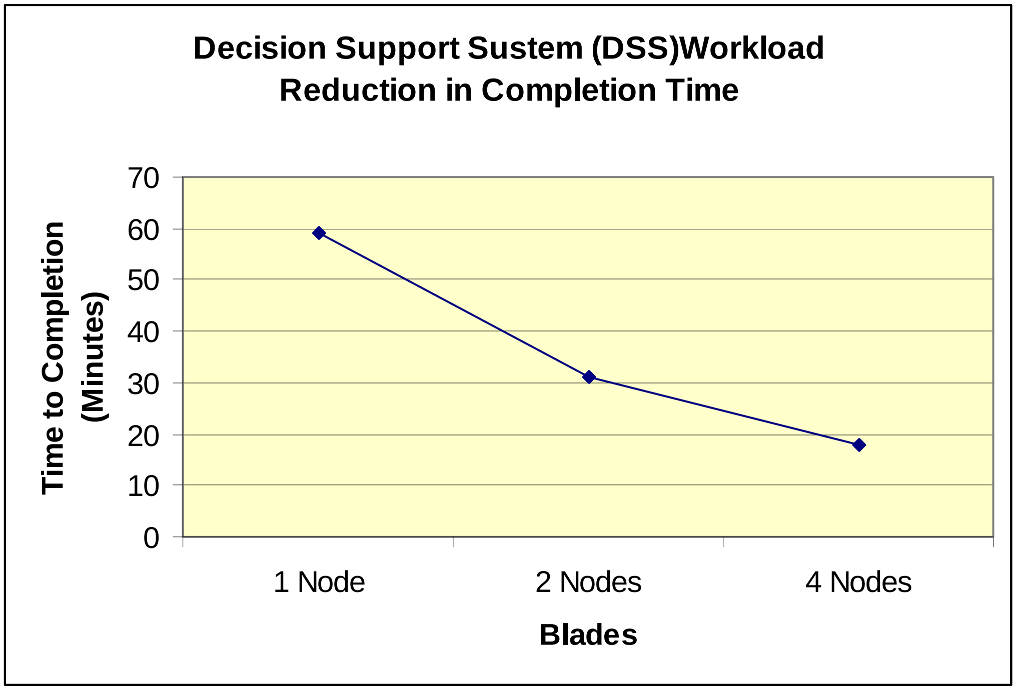


Figure 14. Reduction in completion times as nodes are added.

Following are highlights from the DSS tests:

- Adding nodes to speed up the workload was a non-intrusive effort. An instance of the DSS database was started on another blade, the task was executed again, and completion times improved.
- Oracle10*g* RAC and PolyServe Matrix Server take full advantage of all available disk subsystem bandwidth and do not exhibit scalability limits at the software level.
- The scalability attributes were the same although the CPU and memory requirements of each test were dramatically different. If there had been a server bottleneck, substantial performance variance would have been apparent.

Scalability in an OLTP Environment

To test the scalability of the Flexible Database Cluster Architecture in an OLTP environment, the Order Entry workload described above was executed on nodes 1 through 4 in the RAC cluster. This workload is very contentious and exhibits an approximate read to write ratio of 65:35. There was no form of partitioning (e.g., workload, data-dependent request routing) used.

Figure 15 shows that the scalability measured was 87% from one node (310 TPS) to four nodes (1081 TPS).

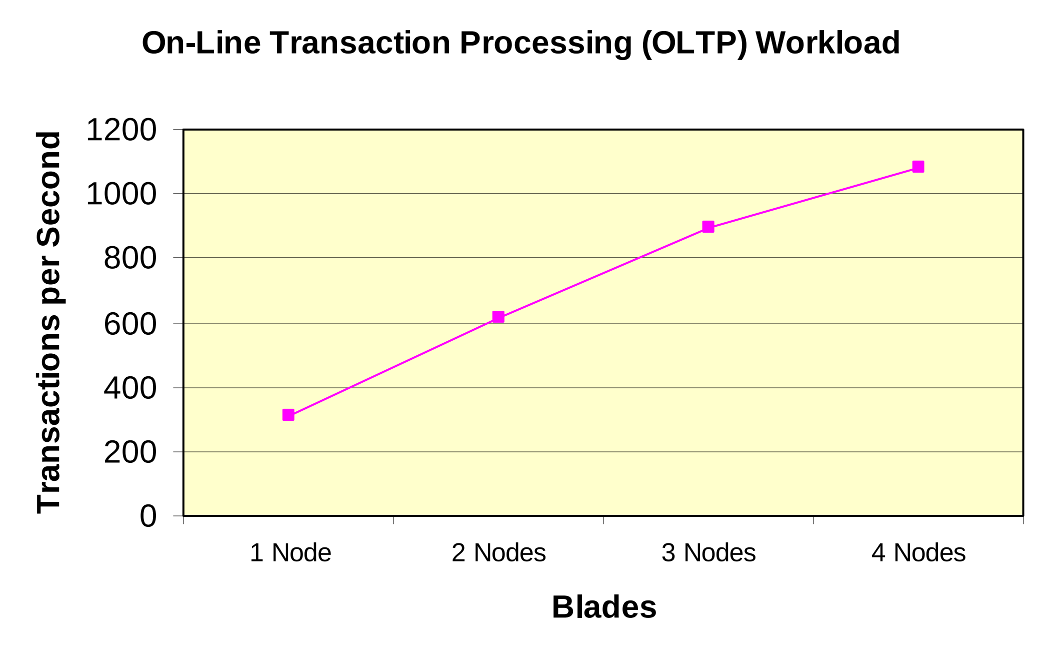


Figure 15. Scalability in an OLTP environment.

This workload saturated CPU resources, but it is also important to investigate the I/O load generated as well. As Figure 16 shows, the I/O curve closely mimics the throughput curve, which establishes the fact that the workload scales consistently. Peaking at over 6,500 physical I/O operations per second at four nodes, it's clear the LS20 blades are capable of handling high OLTP I/O loads.

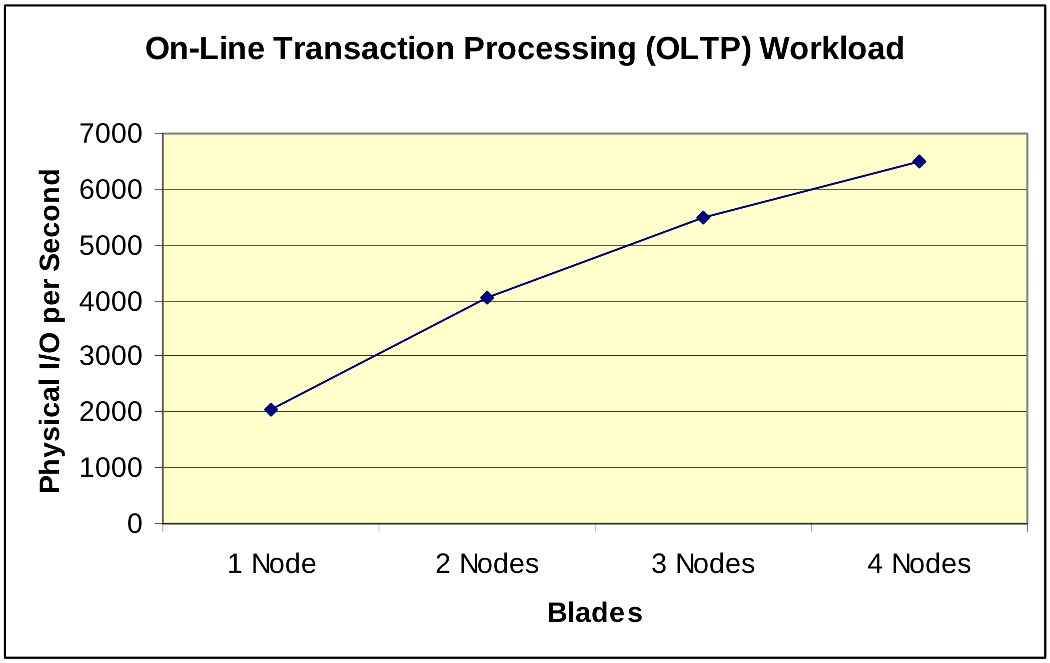


Figure 16. I/O Scalability running an OLTP Workload.

The LS20 blades are not constrained to the 2,185 physical I/Os measured at one node as depicted in Figure 16. This was an Oracle OLTP workload, which carries a significant amount of CPU-intensive work in the transaction layers dealing with the data read in from disk. It is possible to measure the maximum theoretical Oracle OLTP I/O a server is capable of without running the full, processor-intensive, transaction workload that generates the I/O request. The Orion test kit can be used for this type of measurement.

Oracle I/O Workload (Orion)

Orion stands for Oracle IO Numbers. Orion is available[10] from Oracle's Web site. In short, the Orion kit consists of the actual server code that performs I/O when Oracle is running in production. It comes with a layer that drives these I/O routines with varying workload characteristics such as large sequential scan and random single-block transfers. For more in-depth understanding of the Orion test kit, visit the Orion Web page at Oracle's Web site.

To test the single-node maximum theoretical Oracle server OLTP read throughput on a single LS20 node, a 256GB Orion file was created in the PolyServe Matrix Server cluster filesystem and accessed via Direct I/O.

The Orion test revealed that a single LS20 blade is capable of servicing 9,496 random 4KB transfers from the 256GB file per second. Since the target file was substantially larger than the TotalStorage array cache, the I/O latency measured by Orion at 9,496 operations per second was 10.12ms. This test establishes the fact that the I/O subsystem in the LS20 is quite adequate for servicing OLTP I/O requests—to a much higher degree, in fact, than the full Oracle Server would ever be likely to drive.

[10] http://www.oracle.com/technology/software/tech/orion/index.html

Summary

The synergy of IBM BladeCenter H, PolyServe Matrix Server, and Oracle10*g* RAC makes the Flexible Database Cluster a powerful platform for supporting multiple applications. The FDC analysis presented in this paper validates the FDC architecture and technology and confirms that:

- PolyServe Matrix Server and Oracle10*g* RAC are fully supported on IBM BladeCenter H.
- The BladeCenter architecture and technology provide an unparalleled high-availability platform for implementing Flexible Database Clusters.
- IBM, PolyServe and Oracle are leaders in the development of technology for scale-out computing.
- The architecture and technology of the Flexible Database Cluster enables on-demand computing. Cluster nodes provide a pool of flexible resources for use among applications, and the availability of Oracle10*g* RAC is enhanced, because nodes can be dynamically reprovisioned using Matrix Server to cover the loss of another node.
- The Flexible Database Cluster provides strong management tools such as Matrix Server for performance and availability. A single large cluster is now easier to manage than many small clusters.
- A general-purpose cluster filesystem such as the one included with Matrix Server provides a single-system feel and greatly enhances manageability. A shared Oracle home used by all nodes also simplifies management. Support is available for all database operations that require a file system.
- Improved manageability, scalability, expandability, availability and asset utilization in an FDC cluster can also dramatically improve TCO.

IBM Systems and Technology Group

Department 23U

Research Triangle Park, NC 27709

Produced in the USA.

4-06

Visit *www.ibm.com/pc/safecomputing* periodically for the latest information on safe and effective computing. Warranty Information: For a copy of applicable product warranties, write to: Warranty Information, P.O. Box 12195, RTP, NC 27709, Attn: Dept. JDJA/B203. IBM makes no representation or warranty regarding third-party products or services including those designated as ServerProven or ClusterProven.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM reserves the right to change specifications or other product information without notice. References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. IBM PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This publication may contain links to third party sites that are not under the control of or maintained by IBM. Access to any such third party site is at the user's own risk and IBM is not responsible for the accuracy or reliability of any information, data, opinions, advice or statements made on these sites. IBM provides these links merely as a convenience and the inclusion of such links does not imply an endorsement.

This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any software, software feature, or service offered or to be offered by PolyServe, Inc. PolyServe, Inc., reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact PolyServe corporate headquarters for information on feature and product availability.