

Enterprise Scalability on Clustered Servers

A Hewlett-Packard Company and Oracle Corporation
Joint White Paper

v1.1

July 2004

Raghunath K. Othayoth
Senior Engineer, ISS Performance
Hewlett-Packard Company

Vineet Buch
Director, Server Technologies
Oracle Corporation

Enterprise Scalability on Clustered Servers

INTRODUCTION

If you even dabble in databases, chances are you've heard of the TPC-C transaction processing benchmark (<http://www.tpc.org/tpcc/default.asp>). Mention "TPC-C world record", and you probably think of mainframe-class servers running proprietary operating systems. But Oracle, HP and Red Hat just proved that Oracle Real Application Clusters on Linux and Intel servers are the world's fastest platform for database applications – and at an astonishingly low price, too.

The TPC-C benchmark simulates a complete computing environment where a population of users executes transactions against a database. It has two metrics: performance, reported in transactions per minute (tpmC), and price-performance (\$/tpmC). The Oracle-HP system set a new TPC-C world record¹ of 1,184,893 tpmC (yes, that's over 1 million transactions per minute) at a price performance of \$5.52/tpmC – the lowest price per transaction in the TPC-C Top Ten (http://www.tpc.org/tpcc/results/tpcc_perf_results.asp). To put the transaction throughput in perspective – McDonald's serves 47 million customers a day, which translates to 32,638 customers per minute – still far short 1 million per minute.

This benchmark result demonstrates that a cluster of industry-standard HP servers running Oracle Real Application Clusters can handle the most demanding enterprise workload, and that Linux is a viable option for mission critical applications.

Besides providing obvious marketing benefit for Linux in the enterprise, this benchmark also resulted in several enhancements in Linux as a platform for database servers. HP, Oracle and Intel worked with Red Hat to optimize virtual memory management and I/O subsystems, and also made changes to the popular Intel compiler for Itanium 2. The operating system enhancements that arose out of this benchmark will be available as part of Red Hat Enterprise Linux 3.0.

This paper describes the benchmark configuration, how the benchmark tested different aspects of the system and the technical advances that enabled Linux to push the boundaries of scalability. It also gives tuning and system management suggestions for customers running large clusters in the enterprise.

BENCHMARK CONFIGURATION

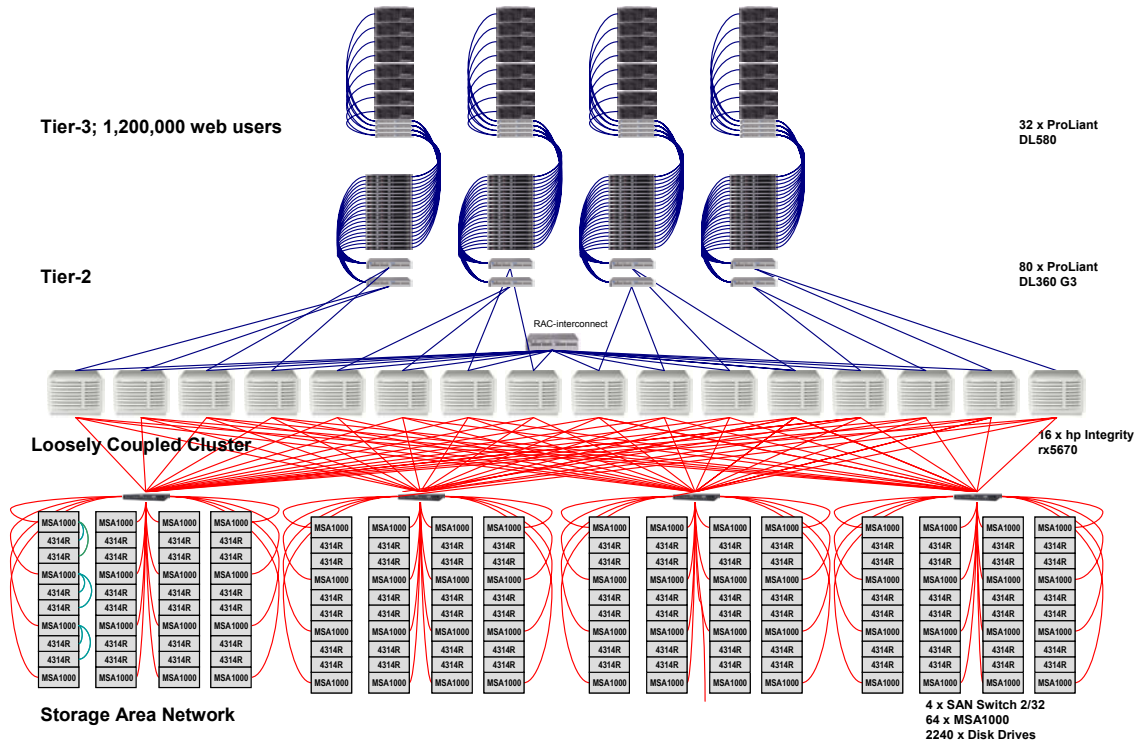
This benchmark ran on sixteen HP rx5670 servers, each with 4 Intel Itanium 2 1.5 GHz CPUs and 48 GB of main memory. A total of 93 TB of storage was configured using 64 HP StorageWorks Modular Storage Array 1000s (MSA1000), and each array was accessible to each

¹ As of December 8, 2003: Oracle Database 10g Enterprise Edition, HP Integrity rx5670 Cluster 64p, 1,184,893 tpmC, \$5.52/tpmC, available 04/30/04.

database server through one of 4 HP StorageWorks 2/32 SAN switches. The cluster interconnect was an HP ProCurve Switch 4148gl running Gigabit Ethernet. The TPC-C mid-tier ran on 80 HP ProLiant DL 360 G3 servers, each with 2 Intel Xeon CPUs (2.4, 2.8 and 3.06 GHz). Tier-3 consisted of 32 ProLiant DL580 servers, which simulated 1,200,000 web users.

The diagram below shows the hardware configuration:

Figure 1: HP Integrity rx5670 Cluster 64P TPC C Configuration.



The database software used was Oracle Database 10g Enterprise Edition with Real Application Clusters. All hardware and software used for this benchmark is generally available to customers.

Storage Area network (SAN)

Shared-disk database cluster architecture is characterized by all nodes sharing all the data in the database: there is only one database - logically and physically. HP Storage Works Storage Area Network (SAN) provided the connectivity platforms to deploy a 93TB stered storage, significantly low I/O price/performance and Total Cost of Ownership (TCO).

The benchmarked SAN consisted of 64 HP StorageWorks Modular Storage Array 1000s (MSA1000), 96 StorageWorks Enclosure 4314Rs, 4 HP StorageWorks SAN Switch 2/32s. Each

HP StorageWorks SAN Switch 2/32 array was accessible to each server through one of 4 HP StorageWorks fca 2214 2GB PCI-X fibre channel HBAs.

The following section provides a brief description of the SAN components.

HP StorageWorks Modular Smart Array 1000 (MSA1000)

The HP StorageWorks Modular Smart Array 1000 (MSA1000) is a 2Gb/s Fibre Channel storage system for the entry-level to midrange storage area network (SAN). It provides the customer with a low-cost, scalable, high performance storage consolidation system with investment protection. It is designed to reduce the complexity and risk of SAN deployments.

HP StorageWorks SAN switch 2/32

The HP StorageWorks SAN Switch 2/32 provides the configuration flexibility, manageability, scalability, and performance needed by corporate storage network infrastructures. The SAN Switch 2/32 is the choice that provides all that configuration flexibility, manageability, scalability, and performance in one neat package.

HP StorageWorks FCA2214 FC HBA

FCA2214 is a 2 Gb/s, 64-bit fibre channel host bus adapter that creates a powerful connectivity solution.

Middle-tier

The Middle-tier consisted of 80 ProLiant DL360 Generation 3 systems running RedHat Linux 3.0, and BEA Tuxedo 8.0 as transaction monitor.

The ProLiant DL360 Generation 3 leads the scale out environment, combining concentrated, 1U compute power with unmatched system features such as embedded remote management and optional redundant power. Xeon processors with 533MHz front side bus, combined with DDR SDRAM and PCI-X technology means the DL360 can handle greater transaction workloads.

Database Layout

The benchmarked SAN had 64 HP StorageWorks MSA1000s.

48 MSA1000s which had 42 36GB 15K rpm disk drives attached were used for tables and indexes. Each MSA1000s had two RAID 0 volumes and 2 ADG volumes, spread over all 42 disk drives. RAID 0 volumes were used for the database under test and ADG volumes were used for database backup. Array accelerator caches on the MSA1000s were set to 100% write.

16 MSA1000s which had 14 146GB 10K rpm disk drives attached were used for redo logs. Each MSA1000 had two RAID 0 volumes. Array accelerator caches on the MSA1000s were set to 100% write.

Redo log files were protected from any single-point-of-failure (disk drive, controller cache, controllers etc) by placing the redo log file group members on separate MSA1000s (accessed through separate HBAs and SAN switches).

Detailed description of database layout and configuration can be found in the TPC-C full disclosure report at http://www.tpc.org/tpcc/results/tpcc_result_detail.asp?id=103120803.

HP StorageWorks ACU-XE, a web-based array configuration utility, was used to configure and maintain arrays.

PUSHING THE DATABASE

This benchmark simulated 1.28 million users and generated a huge load on the Oracle Database. The clustered Oracle Database for this test contained 15 TB of data, which is immense in the OLTP world. A total of 1464 datafiles were used to build the database. Oracle's ability to support different block sizes for different parts of the database (different tablespaces) enabled the default 4 KB data block size to be modified to 2 KB for tables accessed in single rows and 16 KB for tables accessed in large chunks.

This benchmark stressed the database redo logs to levels never seen in customer systems. It also generated significant inter-node traffic for cache fusion between database nodes, but HP ProCurve Switch 4148gl running Gigabit Ethernet was fast enough to allow for excellent scalability as nodes were added to the system.

LINUX ENHANCEMENTS

Early versions of Linux had limitations that made it difficult to use for the enterprise. Intel, RedHat and Oracle have worked closely together to enhance Linux to enable companies to deploy large-scale configurations using the latest version of Oracle Database, Oracle Database 10g, with RedHat Linux 3.0. These enhancements fall in two areas: the I/O subsystem and the virtual memory architecture.

Linux I/O subsystem

Asynchronous I/O

Before the introduction of asynchronous I/O, processes submitted disk I/O requests sequentially and synchronously. With synchronous I/O, when a request is submitted to the operating system, the writing process blocks until the write is completed. It is only after the request completes that the calling process can continue processing.

Asynchronous I/O allows a process to submit an I/O request without waiting for it to complete. Because the system does not put the process to sleep while the I/O request is submitted and processed, the calling process is able to perform other tasks while the I/O proceeds.

Another advantage of this implementation is that it also allows Oracle processes to issue multiple I/O requests with a single system call, rather than a large number of distinct I/O requests. The system can optimize disk activity by reordering requests or combining individual requests that are adjacent on disk into fewer, larger, and thus less expensive, requests.

Relieve contention for global lock

Another area of improvement in the Linux I/O subsystem is the elimination of global lock usage for maintaining the integrity of kernel data structures that are accessed concurrently by multiple

processes. In earlier releases of Linux, I/O requests were queued one at a time while holding a global lock (io_request_lock), used for the entire device block subsystem. This forced all processes performing I/O to compete for this single resource, even when I/Os were performed to unrelated devices, creating an unnecessary bottleneck and reduced overall system I/O throughput.

With the optimization, I/O requests are now queued while holding a lock specific to the queue associated with the request. With this fine-grained locking scheme, a separate lock is held for each individual block device. The result is a more scalable concurrent I/O queuing scheme and significantly better I/O throughput for multi-processor systems with multiple I/O controllers under heavy database load.

Variable Block I/O Sizes

Early versions of Linux required I/Os to be broken up into several blocks with a maximum size of 4KB (RAWIO_BLOCKSIZE), and then merged again by the SCSI layer or driver.

Support for variable block I/O sizes, removes this constraint. Raw I/O requests can now be submitted as one single request, with variable block sizes.

This optimization improves the performance and the scalability of I/O operations.

Virtual Memory: Huge TLB Pages

The HP rx5670 directly addresses 96GB of main memory (48GB used per server in the benchmark). Red Hat Linux optimizes HP's large memory addressability using huge pages. Linux uses very sophisticated virtual memory management algorithms to make the most efficient use of physical memory resources. Linux allocates virtual memory to meet the total memory requirements of each process, and then manages the available physical memory to meet the actual memory requirements. With this environment, each Linux process executes as if it had the entire address space of the CPU available to itself.

The mapping between virtual addresses and physical addresses is done using a data structure called the page table. Each processor uses a small amount of associative memory, called a Translation Look-aside Buffer (TLB), to cache the page table entries of recently accessed virtual pages and uses the page table entry stored in the TLB to calculate the physical address, making it possible for a virtual memory access to be as fast as a direct physical memory access.

There are very few TLB entries on a processor (typically a few dozen), so applications like Oracle that access large amounts of memory can have a high TLB miss rate.

With these vast amounts of memory to access, there is a need to make each TLB mapping as large as possible to reduce the pressure on each processor's TLB. Large continuous regions in a process address space, such as contiguous data, may be mapped by using a small number of large pages rather than large number of small pages. Thus a requirement for having a separate large page size facility from the operating system becomes very important for both functionality and performance.

A new feature called Huge TLB pages allows applications to benefit from using large pages without affecting many other aspects of the OS. The Huge TLB page support brings several benefits to applications:

1. It increases the TLB coverage per CPU and reduces TLB miss rates

With large page support, the processor deals with more memory for each page table entry. Consequently, the TLB miss rate decreases significantly.

2. It reduces the memory usage of page tables

Page table size is greatly reduced which leads to improved memory utilization. This is because a single page table entry can address vast amounts of memory.

3. It eliminates the risk of swapping physical pages and reduces the page cache complexity

Better performance is also achieved because the big pages are not swapped, so the entire buffer cache is locked in physical memory. System performance increases as a result of less page swapping activities and less page cache complexity.

160 Huge TLB pages of 256MB (160x 256=40GB) were configured in each server.

CONCLUSION

Oracle and HP have collaborated to fully optimize Oracle10g for HP Integrity Servers. This partnership not only delivers record setting benchmarks but also ensures customers the highest performance and cost effective Oracle and HP database systems.

This benchmark result demonstrates that a cluster of industry-standard HP servers running Oracle Real Application Clusters can handle the most demanding enterprise workload. Enterprise customers can now deploy Oracle Databases on HP hardware with Red Hat Linux and obtain high availability and lower cost.