OpenVMS scalability with Oracle Rdb

Scalability achieved through performance tuning.

Table of contents

Abstract: By tuning and using new techniques, Oracle and HP engineers demonstrated a 10-fold improvement in Rdb performance with OpenVMS on HP AlphaServer systems. This technical achievement creates the business benefit of extending the scalability and expanding the capacity of an existing Rdb environment running on HP AlphaServer OpenVMS systems. No application changes are required to take advantage of these new features. This case study discusses how this performance enhancement is created incrementally and non-disruptively in the existing hardware, operating system, and database environment.

From technical achievement to business benefit

Executing 1.8 million transactions in one minute

The technical achievement arose when Oracle and HP engineers were testing an AlphaServer OpenVMS system optimized for maximum Rdb database transaction performance. The optimizations resulted in a sustained throughput of 1,811,478 Rdb database transactions per minute.

It is important to note that this test is not based on the TPC Benchmark™ C specifications nor were the results submitted to the Transaction Processing Performance Council (TPC) for auditing. Rather, it was a demonstration of the ability of this generally available technology combination to meet the needs of database transaction-intensive environments with significant longevity.

Oracle and HP tested the environment to demonstrate that it could successfully and productively cache one billion rows in one table plus the associated index nodes using the newly enhanced VLM (very large memory) support for Rdb row caches. It was a remarkable side benefit of this test that the environment executed 1.8 million transactions in one minute.

This result was achieved with:

- AlphaServer hardware using EV7 and mesh technology
- The OpenVMS operating system with performance improvements in version 7.3-2
- An Rdb database with extended row caching technology and other performance improvements in version 7.1.2

Extended scalability, expanded capacity

The business benefit arises from the fact that — using these technologies with the associated tools and techniques – you can extend the scalability and expand the capacity of an existing Rdb environment running on HP AlphaServer OpenVMS systems.

The benefit accrues most dramatically to any industry that requires high levels of database read/write transactions. Financial services, telecommunications, government, and manufacturing can all benefit from enhanced OpenVMS scalability with Oracle Rdb row cache technology.

- Financial services organizations would see the advantage of this capability in shortening end-of-period closes, realtime analysis, transaction frequency spikes, and the like.
- Telecommunications firms would benefit in billing systems, CDR, and at times of service demand spikes.
- This capability is advantageous in government computing environments, human services, defense, intelligence, and other areas that require intensive database transaction rates.
- In manufacturing environments, two examples of how this capability enhances operations are radio frequency identification (RFID) and anti-lock brake system (ABS) fabrication. In the radio frequency identification area enormous numbers of database transactions take place to ensure tracking, recording, and updating. In ABS manufacture it is essential to collect and use massive amounts of data in order to comply with safety and legal requirements.

Headroom – maximum potential database throughput
The HP OpenVMS and Oracle Rdb collaboration continues to deliver greater and greater maximum potential database throughput — "headroom." Current OpenVMS and Rdb administrators facing unpredictable, exponential database-transaction-rate growth rates can take immediate advantage of this headroom.

Scalability and investment longevity
Optimizations such as Rdb row caching or the AlphaServer GS1280 mesh architecture's impact on improved SMP scaling have boosted maximum potential headroom of currently available (off-the-shelf) solutions even faster.

For environments using Rdb databases on OpenVMS AlphaServer systems, maximum potential database throughput – and thus scalability – can be vastly increased with the result that investments in the environment provide more returns for a longer period of time.[1]

Customers who use a current AlphaServer OpenVMS system running Rdb (for example, a GS1280 with 8 CPUs and 32 GB of memory with the transaction throughput capacity they project they need for the next 24 months) have also effectively purchased a very low-cost insurance policy.

If they experience unpredicted, sudden jumps in transaction demand, they know they can bring additional capacity on line into their production environment at a predictable cost, with very low risk, and with very short lead times.

Adding CPUs and memory to their GS1280 as needed opens up the possibility of dramatically increasing Rdb database performance throughput and taking further advantage of the Rdb row caching performance boost.

The managers of large GS160 or GS140 database servers using earlier versions of Rdb now know that they have headroom limits for transaction throughput. These limits are at least three times greater (and more likely five to ten times greater) than their current usage levels.

The maximum configurability of the GS1280 to 32 CPUs[2] and 512 GB ensures that applications will have all the performance possibly required for use today and all that is needed for the foreseeable future.

Upgrades and expansions are non-disruptive
GS1280 CPU and memory upgrades and expansions are specifically non-disruptive. And they don't have any expense other than a known and predictable cost for configuring a system of the right size. Customers can pay for performance in proportion to what they need, when they need it. Even upgrades from ES45 to GS1280 systems have this non-disruptive feature.

With this approach, one avoids transitions that have large and especially unpredictable disruptions and costs that can be magnified many times beyond the hardware cost. This can be a result of delay, error, unreliability, and difficulty in predicting and sizing for performance in advance. This class of transition would include any case of changing the operating system, changing the database, or moving to a new hardware platform other than Alpha including moving to OpenVMS and Rdb on Itanium.

[1] Because, as of the publication of this case study, OpenVMS has not yet been qualified on the full line of HP Integrity servers, we have not conducted these tests in the Integrity server environment. However, Oracle fully intends to support Row Cache on OpenVMS for HP Integrity servers. Therefore, we should be able to use the same techniques to achieve high performance on Itanium systems.

[2] Currently as 2 x 32p, 256 GB hard partitions.

Technology overview

Achieving the level of scalability described above requires three technology elements: hardware, operating system, and database management software. Moreover, proven tools and techniques must exist to capitalize on the technological advantages that these three elements afford.

Hardware

The hardware used in this test was an AlphaServer GS1280 with 32 processors and 256 GB of main memory functioning as an SMP environment. Alpha EV7 chip technology and the mesh architecture deliver vastly improved performance and reduced SMP latency. (Please see http://h18002.www1.hp.com/alphaserver/announce/oct_03_perf.html for more information.)

Operating system

The operating system employed was OpenVMS V7.3-2. This latest version incorporates enhancements that take best advantage of the AlphaServer EV7 technology. OpenVMS with EV7 technology scales well up to 32 CPUs on this Rdb workload and in environments that can make effective use of row caching.

Database

The database software used is Oracle Rdb release 7.1.2. The Rdb database is specifically built and optimized for OpenVMS. This version includes enhanced row cache technologies that vastly increase the number of table rows and index nodes that can be cached. This dramatically improves database transaction performance for databases of typical enterprise scale.

What is an Rdb row cache?

An Oracle Rdb row cache is a section of globally accessible memory that contains copies of database rows or index nodes. This cache provides the ability to store, fetch, and modify frequently accessed rows in memory, avoiding disk I/O and page locking. No application changes are necessary when using the Oracle Rdb row cache feature. Row caching provides the following advantages:

- Reduced database read and write I/O operations
- Reduced database page-locking operations
- Reduced CPU overhead for accessing a database row in cache
- Improved response time
- Efficient use of system memory resources for shared data

Multiple row caches can be active for a database. A row cache is available to, and shared by, all processes attached to the database. When using OpenVMS Galaxy cluster configurations, processes on different nodes within the galaxy environment share row caches that are stored in "galactic" shared memory.

Row cache feature enhancements

Oracle Rdb for OpenVMS Alpha Release 7.1.2 introduced two significant enhancements to the Row Cache feature:

Snapshots in row cache: Using the snapshots in row cache feature allows many applications to approach zero disk I/O operations per transaction by reading and updating database rows entirely within memory while using the after-image journal to provide persistent storage and data protection.

Native 64-bit addressing support for row cache: Native 64-bit addressing allows vastly larger row caches to be created and avoids performance penalties related to the previously available row cache use of VLM capabilities within Oracle Rdb.

Test tools

The engineering team used classic OpenVMS and Rdb performance monitoring tools for this test, including VMS Monitor and RMUSTAT. T4 collection was also employed to observe how system operation changed over time as the load was increased or the number of active CPUs was changed for scaling tests. (See http://h71000.www7.hp.com/openvms/products/t4/index.html and http://h71000.www7.hp.com/openvms/journal/v3/t4.pdf for more information about T4 and related performance tools.)

Spinlock tracing capability built into SDA allowed observation in more detail about how these very heavy workloads impacted SMP scalability. During the tests multiple interactive monitors were running, often at high resolution, while at the same time Monitor and RMUSTAT data was collected and saved for the whole run in their respective recording files for later review and analysis.

Continuous improvement

In addition to the performance indicators, the ability to run these test workloads on a large multi-processor system also provided the opportunity for Oracle Rdb and HP OpenVMS engineers to measure and evaluate various performance indicators. The results of such analysis helped provide direction for further areas of investigation for performance enhancements in future releases of both OpenVMS and Oracle Rdb.

Ported to the HP Integrity

OpenVMS and Rdb are both being ported to the HP Integrity server line of Intel® 2-based Itanium® servers. The first production release of OpenVMS on Integrity servers, version 8.2, is scheduled for the second half of 2004. Oracle has announced an Advanced Developer's Kit for Rdb for OpenVMS on the Integrity server line, version 7.2 that will allow compilation and testing of user

A closer look at the test environment

Each release of Oracle Rdb version 7.1 is currently shipping as two variants: one compiled for all Alpha processors and one compiled for Alpha EV56 and later processors. The EV56 variant includes code compiled to use the Alpha byte-word instructions and quad-issue instruction scheduling. For this performance experiment, the Rdb code was compiled explicitly for the EV67 and later Alpha processors. This configuration allowed the language compilers to produce an Alpha instruction sequence that was optimal for the EV67, EV68, and EV7 processors in both use of available instructions and scheduling of instructions.

A 32-processor GS1280 with 128 GB was configured and made available for additional testing. OpenVMS V7.3-2 was installed along with the experimental compilation of Oracle Rdb.

A number of workload experiments were run. Areas of interest included measuring performance with varied numbers of active CPUs to determine how effectively the system scaled from 8 to 32 processors, the effect of the Rdb AIJ Log Server (RDMALS) process, impact of the Record Cache Server (RDMRCS) process during checkpoint operations, and so on.

1,811,478 transactions were completed in sustained, measured database activity for an interval greater than one minute. This represents a rate of 30,191 transactions per second.

applications on Itanium processors. It will be available at the same time as the beta release of OpenVMS version 8.2 in July, 2004. Beta test for Rdb running natively on Itanium processors will begin before the end of 2004. Production release is scheduled for the first half of 2005. The performance capabilities of Rdb on OpenVMS for HP Integrity servers will be shared as those results become available.

Conclusion

By tuning and using new techniques, Oracle and HP engineers demonstrated a 10-fold improvement in Rdb performance with OpenVMS on HP AlphaServer systems. What's more, this performance enhancement was achieved at a lower cost because it was built incrementally and non-disruptively on the existing hardware, operating system, and database environment. This technical achievement gives industries that require high rates of database read/write transactions the benefits of extended scalability and expanded capacity in their existing AlphaServer OpenVMS Rdb environments.

Appendix

For more information about Oracle Rdb performance and configuration options, you may wish to refer to the following documents available from Oracle Corporation:

- Oracle Rdb Release 7.1.2 Release Notes
- Oracle Rdb7 Guide to Database Performance and Tuning
- Oracle Rdb7 Guide to Database Design and Definition

For detailed information about the Alpha processor or OpenVMS memory management and programming, you may wish to refer to the following documents available from Hewlett-Packard:

- OpenVMS Programming Concepts Manual
- HP OpenVMS System Services Reference Manual
- OpenVMS Calling Standard
- HP OpenVMS System Manager's Manual
- HP OpenVMS Alpha Partitioning and Galaxy Guide
- Alpha Architecture Handbook
- Alpha Microprocessor Hardware Reference Manuals

To learn more about HP's offering, visit **www.hp.com**.

© 2003 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

5982-6837EN, 06/2004

invent