Oracle Database 11*g* Architecture on Windows

An Oracle Technical White Paper
July 2007

Oracle Database 11*g* Architecture on Windows

Oracle Database 11*g* Architecture on Windows

EXECUTIVE OVERVIEW

Oracle Database 11*g* for Windows provides an optimized database solution for deployments that require enterprise scalability, reliability, and high performance. This paper describes the architecture of the Oracle database on Windows and how it differs from its counterparts on UNIX and Linux.

By using a native, thread-based Windows service model, Oracle Database 11*g* ensures high performance and scalability. The Oracle database tightly integrates with the advanced features of the Windows operating system and the underlying hardware, such as Large Page and NUMA support. Oracle provides enterprise-class performance through support for large memory, large and raw files, and grid computing.

The Oracle database is certified for 32-bit and 64-bit Windows operating systems. 32-bit Oracle database is supported on 32-bit Windows with standard x86 hardware, including Windows Vista. On 64-bit Windows, 64-bit Oracle runs on the Windows x64 (AMD64/EM64T) and Windows Itanium operating systems. 64-bit hardware provides greater scalability and performance over 32-bit systems.

INTRODUCTION

Oracle has always maintained a strong commitment to the Windows operating system. In 1993, Oracle was the first to release a relational database for Windows NT. Oracle has made specific changes to its database to improve its performance and usability on Windows.

The Oracle database has become one of the leading database solutions for the Windows platform. From the outset, Oracle's goal has been to provide the highest performing and most tightly integrated database on Windows and, as a result, Oracle invested early on to move its market-leading UNIX database technology to the Windows platform. In 1993, Oracle was the first company to provide a relational database for Windows NT.

Initially, Oracle's development efforts were concentrated on improving the performance and optimizing the architecture of the database on Windows. Oracle7 on Windows NT was redesigned to take advantage of several features unique to the Windows platform, including native thread support and integration with some of the Windows administrative tools such as Performance Monitor and the Event Viewer.

The Oracle database on Windows has evolved from a basic level of operating system integration to using more advanced services in the Windows platform including Itanium and AMD64/EM64T systems. As always, Oracle is continuing to innovate and leverage new Windows technologies. This white paper discusses the architecture of Oracle Database 11*g* on Windows in detail. It covers the innovations that improve the database for Windows, but does not cover features that apply to all hardware platforms.

ORACLE DATABASE ARCHITECTURE ON WINDOWS

Oracle Database 11*g* has the same features and functionality on Windows as on Linux and UNIX. However, underneath the covers, significant work has been done to take advantage of Windows-specific operating system features to improve performance, reliability, and stability.

When running on Windows, Oracle Database 11*g* contains the same features and functionality as it does on the various Linux and UNIX platforms that Oracle supports. However, the interface between the database and the operating system has been substantially modified to take advantage of the unique services provided by Windows. As a result, Oracle Database 11*g* on Windows is not a straightforward port of the UNIX code base. Significant engineering work has been done to make sure that the database exploits Windows' capabilities to the fullest extent and to guarantee that the Oracle database is a stable, reliable, and high performing system upon which to build applications.

Thread Model

The Oracle database architecture on Windows is based on threads, rather than processes. Threads provide faster context switches; a much simpler SGA allocation routine that does not require the use of shared memory; faster spawning of new connections; and decreased overall memory usage.

Compared to the Oracle database on UNIX, the most significant architectural change in Oracle Database 11*g* on Windows is the conversion from a process-based server to a thread-based server. On UNIX, Oracle uses processes to implement background tasks, such as database writer (DBW0), log writer (LGWR), dispatchers, shared servers, and the like. In addition, each dedicated connection made to the database causes another operating system process to be spawned on behalf of that session. On Windows, however, all of these processes are implemented as threads inside a single, large process. What this means is that for each Oracle database instance, there is only one process running on Windows for the Oracle database server itself. (Note: Other Oracle processes exist on Windows for other database services, such as the Enterprise Manager Database Console) Inside this process are many running threads, with each thread corresponding directly to a process in the UNIX architecture. So, if there were 100 Oracle processes running on UNIX for a particular instance, that same workload would be handled by 100 threads in one process on Windows.

Operationally, client applications connecting to the database are unaffected by this change in database architecture. Every effort has been made to ensure that the database operates in the same way on Windows as it does on other platforms, even though the internal process architecture has been converted to a thread-based approach.

The original motivation to move to a thread-based architecture resulted from performance issues with the first release of Windows NT when dealing with files shared among processes. Simply converting to a thread-based architecture and modifying no other code dramatically increased performance as this particular Windows NT bottleneck was avoided. No doubt that the original motivation for the change is no longer present; however, the thread architecture for Oracle remains since it has been proven to be a very stable, maintainable one.

There are other benefits that arise out of the thread architecture. These include faster operating system context switches among threads, as opposed to processes; a much simpler System Global Area (SGA) allocation routine which does not require the use of shared memory; faster spawning of new connections since threads are more quickly created than processes; decreased memory usage since threads share more data structures than processes do; and finally, a perception that a thread-based model is somehow more "Windows-like" than a process-based one.

Internally, the code to implement the thread model is compact and very isolated from the main body of Oracle code. Fewer than 20 modules provide the entire infrastructure needed to implement the thread model. In addition, robustness has been added to the architecture through the use of exception handlers and also through routines used to track and de-allocate resources. Both of these additions help Oracle database on Windows applications meet requirements for 24x7 operations with no downtime due to resource leaks or an ill-behaved program.

Services

The Oracle database runs as a Windows service, which is a background process that can be started by Windows when booting up.

In addition to being thread-based, Oracle Database 11*g* is not a typical Windows process. It is a Windows *service*, which is basically a background process that's registered with the operating system, started by Windows at boot time, and which runs under a particular security context. The conversion of Oracle into a service was necessary to allow the database to come up automatically upon system reboot, since services require no user interaction to start. When the Oracle database service starts, there are none of the typical Oracle threads running in the process. Instead, the process basically waits for an initial connection and startup request from SQL*Plus, which will cause a foreground thread to start and then eventually cause the creation of the background threads and of the SGA. When the database is shutdown, all the threads that were created will terminate, but the process itself will continue to run and will wait for the next connection request and startup command. In addition to the Oracle database service, further support was added to automatically spawn SQL*Plus to start up and open the database for use by clients.

The Oracle Net Listener is a service since it too needs to be running before users can connect to the database. Again, all of this is implementation detail that does not affect how clients connect to or otherwise use the database, although it is very relevant for Windows database administrators.

Scalability Enhancements

Over the years, Oracle has consistently built its database to serve large user populations. Oracle Real Application Clusters increases capacity for user connections and throughput by clustering multiple machines as one database.

One of the key goals of Oracle Database 11*g* on Windows is to fully exploit any operating system and hardware technologies that can help increase scalability, throughput, and database capacity.

Much activity has been undertaken to support large numbers of connected database users on Windows. As far back as Oracle7 version 7.2, there have been customers in production environments with over 1000 concurrent connections to a single database instance on Windows NT. As time has progressed, that number has increased to a point where well over 2000 users can connect concurrently to a single database instance on a single node in production environments. When using the Oracle shared server architecture, which limits the number of threads running in the Oracle database process, over 10,000 simultaneous connections have been accomplished to a single database instance. In addition, network multiplexing and connection pooling features can also allow a large configuration to achieve more connected users to a single database instance.

In recent years, Windows database administrators have been able to further increase their user counts by employing new 64-bit hardware, either Itanium or AMD64/EM64T, and Oracle Real Application Clusters (RAC). 64-bit improvements are discussed later in this paper. Oracle RAC allows multiple server machines access to the same database files, thereby increasing capacity for user connections and at the same time increasing throughput as well. Because commodity hardware can be added as additional nodes to a RAC cluster, RAC has been a popular solution for cost-effective scaling and high availability. On Windows, customers have scaled to a 23-node RAC cluster without any issues.

4GB RAM Tuning (4GT)

The Oracle database on Windows supports accessing large amounts of memory through a variety of means, including 4GB RAM Tuning, Very Large Memory, and Address Windowing Extensions. Because Oracle can use the maximum possible memory, 64GB, on 32-bit Windows, users experience better scalability and throughput.

When clustering and 64-bit Windows are not available options, it is necessary to maximize the resources available on 32-bit Windows systems. 32-bit Windows 2000 Server (Advanced and Datacenter editions) and 32-bit Windows Server 2003 (Enterprise and Datacenter editions) include a feature called 4GB RAM Tuning (4GT). This feature allows memory-intensive Windows applications to directly access up to 3GB of memory as opposed to the standard 2GB that is allowed by default. The obvious benefit to the Oracle database is that 50% more memory becomes available for database use, which can be used to increase SGA sizes or connection counts. All Oracle database server releases since version 7.3.4 have supported this feature with no modifications necessary to the standard Oracle installation. The only configuration change required is to ensure that the /3GB flag is used in the Windows boot.ini file.

Very Large Memory (VLM)

Commonly used in high-memory 32-bit Windows applications is a key memory tuning feature, originally supported with Oracle8*i*, called Very Large Memory (VLM). VLM, available in Windows 2000 and higher, allows the Oracle database on Windows to break through the 3GB address space limit normally imposed by 32-bit Windows. Specifically, a single database instance can now have access up to 64GB of database buffers when running on a machine and an operating system that support that much physical memory. This support in Oracle Database 11*g* is very tightly integrated with the database buffer cache code inside the database kernel, thereby allowing very efficient use of the large RAM amounts available for database buffers. By configuring a database with a large number of buffers, more data is cached in memory. This reduces the amount of disk I/O, which is considerably slower than retrieving data from memory. Using this feature leads to a corresponding increase in database throughput and performance.

Under the covers, Oracle Database 11*g* on Windows takes advantage of the Address Windowing Extensions (AWE), which are built into Windows 2000 and higher operating systems. AWE are a set of API calls that allow applications to access more than the traditional 3GB of RAM normally available to 32-bit Windows applications. The AWE interface takes advantage of the Intel Xeon architecture and provides a fast map/unmap interface to all memory in a machine. As such, when accessing memory above 4GB, applications do not have direct memory access strictly speaking. If the requested database buffer is in an area of memory above 4GB, it must be mapped from this area to memory below 4GB to make it accessible to the 32-bit database. While this is slower than direct memory access, it is considerably faster than using disk.

The AWE calls allow a large increase in database buffer usage up to 64GB of buffers total. This support is purely an in-memory change with no changes or modifications made to the database files themselves.

Large Pages

Large Page support boosts performance for memory-intensive database applications, especially in cases when the buffer cache is several gigabytes in size.

Large Page support is a feature that provides a performance boost for memory-intensive database instances on both 32-bit and 64-bit Windows Server 2003. Oracle databases can make more efficient use of processor memory addressing resources using this feature. Specifically, when Large Page support is enabled, the CPUs in the system will be able to more quickly access the Oracle database buffers in memory. Oracle uses the Large Page support available on Windows. The large page size is 2MB if Physical Address Extension (PAE) is enabled or 4MB if PAE is disabled (on 32-bit Windows); 2MB (on Windows x64); or 16MB (on Windows Itanium) page sizes. The large pages are used for the SGA. All SGA components including buffer cache, shared pool, large pool, and others are allocated from these large pages.

This feature is particularly useful when the Oracle buffer cache is several gigabytes in size. Smaller-sized configurations will still see a gain when using Large Pages, but it will not be as great as when the database is accessing large amounts of memory. To enable this new feature, the registry variable ORA_LPENABLE should be set to 1 in the Oracle key of the Windows Registry.

Affinity and Priority Settings

Database administrators can assign CPU affinities and priorities to specific Oracle threads to improve their performance.

The Oracle database supports the modification of both priority and affinity settings for the database process and individual threads in that process when running on Windows.

By modifying the value of the ORACLE_PRIORITY registry setting, a database administrator can assign different Windows priorities to the individual background threads and also to the foreground threads as a whole. Likewise, the priority of the entire Oracle process can also be modified. In certain circumstances, this may improve performance slightly. For instance, if an application generates a great deal of log file activity, the priority of the LGWR thread can be increased to better handle the load put upon it. Likewise, if replication is heavily used, those threads that refresh data to and from remote databases can have their priority bumped up as well.

Much like the ORACLE_PRIORITY setting, the ORACLE_AFFINITY registry setting allows a database administrator to assign the entire Oracle process or individual threads in that process to particular CPUs or groups of CPUs in the system. Again, in certain cases, this can help performance. For instance, pinning DBW0 to a single CPU such that it does not migrate from one CPU to another can in some cases provide a slight performance improvement. Also, if there are other applications running on the system, using ORACLE_AFFINITY can be a way to keep Oracle confined to a subset of the available CPUs in order to give the other applications time to run.

Non-Uniform Memory Access (NUMA)

The Oracle database can automatically detect NUMA hardware and optimize itself by efficiently utilizing NUMA node affinities.

With the addition of Non-Uniform Memory Access (NUMA) support in Windows Server 2003, Oracle can now better exploit high-end NUMA hardware in which a single large physical server is comprised of several computing “nodes”. Since each node in a NUMA machine accesses different parts of physical RAM at different speeds, it is essential that the database can determine the topology of a NUMA machine and adjust its scheduling, memory allocations, and internal operations accordingly.

When running on a NUMA machine, the Oracle database automatically sets the ORACLE_AFFINITY setting to an appropriate default value at startup to maximize the machine’s resource utilization. In addition, the SGA and PGA memory allocations are made in a NUMA-aware fashion such that memory is accessed as efficiently as possible from all the various “nodes” on the server. Finally, the number of database writer threads is configured such that there is one per node, again as a performance-enhancing operation.

File I/O Enhancements

The Oracle database on Windows supports a cluster file system, easing manageability. 64-bit file I/O support permits file sizes beyond 4GB. Raw files, or unformatted disk partitions, are supported to provide some performance gain over using a conventional file system.

Another area in which much work has been done in the Oracle database code concerns support for cluster files, large files, and raw files. The Oracle cluster file system is an integral part of Oracle Database 11*g* that makes administration and installation of Oracle clusters easier. In an effort to guarantee that all Windows features are fully exploited, the database supports 64-bit file I/O to permit the use of file sizes larger than 4GB. In addition, physical and logical raw files are supported for data, log, and control files to enable improved performance using Oracle RAC and single instance databases on Windows.

Cluster File System

Oracle RAC manageability has been greatly improved through the Oracle cluster file system (CFS). The Oracle CFS was created for use with RAC specifically. Oracle RAC executables are installed on either the CFS or on raw files. In the latter case, at least one database instance runs on each node of the cluster. In a single Oracle home install with CFS, the database will exist on the shared storage, generally a storage array. CFS allows the Oracle software to be accessible by all nodes in the cluster, but controlled by none. All CFS machines have equal access to all the data and can process any transaction. In this way, RAC with CFS ensures full database software redundancy for Windows clusters while simplifying installation and administration.

64-Bit File I/O

Internally, all Oracle database file I/O routines support 64-bit file offsets, meaning that there are no 2GB or 4GB file size limitations when it comes to data, log, or control files as is the case on some other platforms. In fact, the limitations that are in place are generic Oracle limitations across all ports. These limits include 4 million database blocks per file, 16KB maximum block size, and 64K files per database. If these values are multiplied, the maximum file size for a database file on Windows is calculated to be 64GB, while the maximum total database size supported (with 16KB database blocks) is 4 petabytes.

Raw File Support

Like UNIX, Windows supports the concept of raw files, which are basically unformatted disk partitions that can be used as one large file. Raw files have the benefit of no file system overhead, since they are unformatted partitions. As a result, using raw files for database or log files can produce a slight performance gain. However, the downside to using raw files is manageability since standard Windows commands do not support manipulating or backing up raw files. Therefore, raw files are generally used only by very high-end installations and by Oracle Real Application Clusters, requiring optimized performance.

To use a raw file, all Oracle requires is the filename specifying which drive letter or partition to use for the file. For instance, the filename \\.\PhysicalDrive3 tells Oracle to use the 3rd physical drive as a physical raw file as part of the database. In addition, a file such as \\.\log_file_1 is an example of a raw file that has been assigned an alias for ease of understanding. Aliases can be assigned with the Oracle Object Link Manager (OLM). OLM provides an easy to use graphical interface and maintains the links across the cluster and through reboots. When specifying raw filenames to Oracle, care must be taken to choose the right partition number or drive letter, as Oracle will simply overwrite anything on the drive specified when it adds the file to the database, even if it's already an NTFS or FAT formatted drive.

To Oracle, raw files are really no different from other Oracle database files. They are treated in the same way by Oracle and can be backed up and restored via Recovery Manager as any other file can be.

Direct Network File System Client – New for 11g

Oracle Database 11*g* can be configured to access Network File System (NFS) Version 3 servers directly using an internal Oracle Direct Network File System client.

This feature is implemented as part of the Oracle database kernel in the Oracle Disk Manager library. Network Attached Storage (NAS) based systems use NFS to access data. In previous Oracle releases, the operating system provided the kernel network file system driver to access NAS storage devices. This setup required specific configuration settings to ensure efficient and correct usage with Oracle. If the configuration parameters were not correctly specified, the following problems arose:

- NFS clients were very inconsistent across platforms and vary across operating system releases.
- The configuration parameters were difficult to tune. There are more than 20 NFS parameters with subtle differences among them across platforms.
- The NFS client stack was designed for general-purpose use. As such, it contains features, such as file attribute management that are not required by Oracle.

Oracle Direct Network File System implements NFS Version 3 protocol within the database kernel, leading to easier manageability with better and more predictable performance characteristics. The following are the main advantages for using this new implementation:

- It enables complete control over input-output paths to NFS servers, resulting in predictable performance, simplified configuration management, and superior diagnostics.
- Its operations avoid the kernel network file system layer bottlenecks and resource limitations. However, the kernel is still used for network communication modules.

- It provides a common NFS interface for Oracle for potential use on all host platforms and supported NFS servers.
- It enables improved performance through load balancing across multiple connections to NFS servers and deep pipelines of asynchronous input-output operations with improved concurrency.

64-BIT WINDOWS OPERATING SYSTEMS

The next major scalability step for the Oracle database architecture has been achieved with the move to 64-bit AMD64/EM64T and Itanium platforms. Because the Oracle database has already been ported to other 64-bit platforms, the move to 64-bit Windows results in a stable, high performing database.

64-bit Windows and hardware starts a new leap in Oracle database performance and scalability. Two 64-bit Windows platforms are available: the AMD64 and Intel EM64T platform and the Intel Itanium platform. The former platform uses the Windows x64 operating system. Both platforms provide greater scalability and higher performance than their 32-bit counterpart.

Oracle has been strongly committed to these 64-bit platforms. It was the first to make a database developer's release publicly available for 64-bit Windows on both Itanium and AMD64/EM64T. Oracle has continued to lead the way in 64-bit Windows computing by releasing a production version of the Oracle database on the same day that 64-bit Windows Server 2003 for Itanium was launched. Oracle's development teams have been working closely with Microsoft, Intel, and AMD to guarantee that the database works optimally on both sets of 64-bit hardware and operating systems.

As with Oracle's 64-bit databases on the UNIX platforms, the 64-bit Oracle database on Windows is able to handle more connections, allocate much more memory, and provide much better throughput than the 32-bit database. Oracle's performance and scalability greatly benefit from the larger caches and memory available on 64-bit systems. There is no longer a 4GB memory limitation as on 32-bit systems, making 64-bit Oracle perfect for large transaction processing or business intelligence applications. Moreover, Oracle benefits from the improved parallelism, scheduling, and throughput available on 64-bit architectures. All these performance enhancements are transparently available in the Oracle database; thus, they require no code changes for existing database deployments to use.

In addition to the inherent performance gain achieved by moving to 64-bit, one of the major performance improvements employed by Oracle is profile-guided optimization (PGO). With Intel's 64-bit Windows compiler, Oracle has designed its database to perform optimally for typical customer workloads on both Itanium and AMD64/EM64T. By using simulated customer workloads during compilation, a feedback loop is provided to the compiler, which then can analyze the most heavily and lightly used code paths. Based on that information, the compiler can arrange the code paths to be more efficient when run on 64-bit hardware. Just by using PGO with no other changes, Oracle has seen approximately a 15%-25% improvement in performance. The PGO improvements are transparent for existing applications, requiring no code changes.

The migration path from 32-bit to 64-bit Oracle is very straightforward. There is no need to recreate databases, nor is a full export and import required. All that is needed is to copy the current data files to the new system, install the 64-bit version of Oracle, start the database as normal, and run a few SQL scripts to update the data dictionary.

From an architectural perspective, the current, proven thread-based architecture is used for the 64-bit port. As a result, creating the new 64-bit Oracle software basically entailed re-compiling, re-linking, re-testing and re-releasing the new version. Very little new code was written during the move to 64-bit since the underlying operating system APIs are substantially the same. In addition, since the Oracle database has already been ported to other 64-bit operating systems, moving to 64-bit is a straightforward process that will produce a quality, stable product in a very short period of time.

One of the benefits of using AMD64/EM64T is the ability to easily migrate applications from 32-bit to 64-bit on the same system. With this hardware, customers can run the 32-bit Oracle database server and client on 32-bit Windows. Or they can run the operating system in 64-bit mode, while the Oracle client remains in 32-bit mode, while other applications are converted to 64-bit. Or they can fully migrate to a 64-bit Oracle stack on top of Windows x64. These options provide an easier 32-bit to 64-bit migration path if there are multiple applications running on the same machine. Customers can migrate their applications to 64-bit in a staggered format.

CONCLUSION

Oracle Database 11*g* for Windows has evolved from a port of its UNIX database server to a well-integrated native application that takes full advantage of the services and features of the Windows operating system and underlying hardware. Oracle continues to improve the performance, scalability, and capability of its Windows database server, while at the same time producing a stable, highly functional platform on which to build applications. Oracle is fully committed to providing the highest performing database for both 32-bit and 64-bit Windows platforms.

For additional information about Oracle database on Windows, visit:

Technical - http://otn.oracle.com/windows

Business - http://www.oracle.com/windows

Oracle Database 11*g* Architecture on Windows
July 2007
Author: David Colello
Contributing Authors: Alex Keh, Ravi Thammaiah

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

This document is provided for information purposes only and the contents hereof are subject to change without notice.
This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.