Oracle9*i* Release 2 Database Architecture on Windows

An Oracle Technical White Paper
April 2002

ORACLE

Oracle9*i* Release 2 Database Architecture on Windows

Oracle9*i* Release 2 Database Architecture on Windows

EXECUTIVE OVERVIEW

With the introduction of the Oracle9*i* Release 2 Database for Windows, Oracle once again provides the enterprise scalability, reliability, and high performance that customers require. The Oracle9*i* Release 2 Database provides enterprise-class data solutions through tight integration with the advanced features of the Windows operating system and the underlying hardware. By using a native, thread-based Windows service model, Oracle9*i* Release 2 ensures high performance and scalability. Oracle can provide enterprise-class performance through the use of large and raw file support, large memory support, and parallel computing via clustering. Future performance and scalability enhancements will be available with the impending release of the 64-bit version of the Oracle database on Windows. This paper discusses how the Oracle database architecture has been designed to take full advantage of advanced Windows operating system features and the underlying hardware.

INTRODUCTION

As the introduction of Windows XP and Windows .NET take place, Oracle9*i* Release 2 has become the leading database for the Windows platform. From the outset, Oracle's goal has always been to provide the highest performing and most tightly integrated database on Windows and as a result, Oracle invested early to move its market-leading UNIX database technology to the Windows platform. In 1993, Oracle was the first company to provide a database for Windows NT.

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

However, Oracle9*i* Release 2 on Windows has evolved from the basic level of operating system integration to utilize some of the more advanced services in the Windows platform including the Itanium-based 64-bit version of Windows. As always, Oracle is continuing to innovate and leverage new Windows technologies. This white paper discusses the Oracle9*i* Release 2 database architecture on Windows in detail.

ORACLE9*i* RELEASE 2 ARCHITECTURE ON WINDOWS

The Oracle9*i* Release 2 database has the same features and functionality on Windows as on 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, Oracle9*i* Release 2 contains the same features and functionality as it does on the various UNIX platforms that Oracle supports. However, the interface between Oracle9*i* Release 2 and the operating system have been substantially modified to take advantage of the unique services provided by Windows. As a result, Oracle9*i* Release 2 on Windows is not a straightforward port of the UNIX code base. Significant engineering work has been done to make sure that Oracle9*i* Release 2 exploits Windows to the fullest and also to guarantee that Oracle9*i* Release 2 is a stable, reliable, and high performing system upon which to build applications.

Thread Model

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

Compared to Oracle9*i* Release 2 on UNIX, the most significant architectural change in Oracle9*i* Release 2 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), Multi-threaded Server (MTS) dispatchers, MTS 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 Oracle9*i* Release 2 server itself. Inside that process will be 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 had to do with 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 the particular operating system 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. In addition, 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 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 allow for 24x7 operation 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, the Oracle9*i* Release 2 database is also 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 which will 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 which allows the automatic spawning of SQL*Plus to start up and open the database for use by clients. Finally, 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 it basically an implementation detail that does not affect how clients connect to or otherwise use the database, although this is very relevant for administrators of the database on Windows.

Scalability Enhancements

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 on Windows 2000, 64GB, users experience better scalability and throughput.

One of the key goals of the Oracle9*i* Release 2 product on Windows is to fully exploit any technologies that can help increase scalability, throughput, and database capacity. The following section describes a few of these technologies, how they affect Oracle, and the benefits that can be derived from them.

4GB RAM Tuning (4GT) Support

Windows 2000 Server (Advanced and Datacenter editions) along with Windows .NET Server (Enterprise and Datacenter editions) include a feature called 4GB RAM Tuning (4GT). This feature allows memory-intensive applications running on Windows to access up to 3GB of memory as opposed to the standard 2GB that is allowed in other editions of Windows. The obvious benefit to Oracle9*i* Release 2 is that 50% more memory becomes available for database use, which can 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 a standard Oracle installation. The only configuration change required is to ensure that the /3GB flag is used in Windows's boot.ini file.

Large User Populations

Over the years, Oracle has consistently built its database to serve large user populations. Oracle9*i* Release 2 RAC increases capacity for user connections and at the same time increases throughput.

One area in which much activity has been undertaken is an effort to support large numbers of connected database users on Windows. As far back as Oracle7 version 7.2, there have been customers in production 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 concurrent to the database in production environments. When using the Oracle Multi-threaded 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. Finally, Oracle Real Application Clusters can be used to again increase connection counts dramatically by allowing multiple server machines access to the same database files, thereby increasing capacity for user connections and at the same time increasing throughput as well.

VLM Support

One of the key Windows 2000-specific additions originally introduced in Oracle8*i* was support for Very Large Memory (VLM) configurations. Oracle9*i* Release 2 enhances this support and allows the RDBMS on Windows to break through the 3GB address space limit normally imposed by Windows 2000 and Windows .NET Server. Specifically, a single database instance can now access up to 64GB of database buffers when running on a machine and an O/S that support that much physical memory. In addition, this support in Oracle9*i* Release 2 is very tightly integrated with the database buffer cache code inside the RDBMS kernel, thereby allowing very efficient use of the large amounts of RAM available for database buffers. By configuring a database with a large amount of buffers, disk I/O can be diminished since more data is cached in memory. This leads to a corresponding increase in throughput and performance.

Under the covers, Oracle9*i* Release 2 on Windows takes advantage of the Address Windowing Extensions (AWE), which are built into all Windows 2000 and Windows .NET Servers. The AWE are a set of API calls which allow applications to access more than the traditional 3GB of RAM normally accessible to Windows 2000 and Windows .NET applications. The AWE interface takes advantage of the Intel Xeon architecture and provides a fast map/unmap interface to all memory in a machine.

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.

Affinity and Priority Settings

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

The Oracle9*i* Release 2 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 some applications. 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.

Both ORACLE_PRIORITY and ORACLE_AFFINITY are described in more detail in the Windows-specific documentation that accompanies Oracle9*i* Release 2 on Windows.

64-bit Support

The next major step for the Oracle database architecture on Windows is the move to 64-bit Itanium, which will greatly improve scalability. Because the Oracle database has already been ported to other 64-bit platforms, the move to 64-bit Windows will result in a stable, high performing database from Oracle.

The next leap in Oracle database performance and scalability on Windows will happen when a 64-bit version of the Oracle database is released on Intel Itanium processor-based machines and the 64-bit server version of Windows. The development teams at Oracle have been working closely with the vendors of these technologies to guarantee that the Oracle database on Windows will be released in production form soon after the hardware and operating system are generally available. As with other Oracle 64-bit ports to different UNIX variants, a 64-bit port of the Oracle database to Windows will be able to handle more connections, allocate much more memory, and provide much better throughput than the current version of the database on Windows.

The migration path from 32-bit to 64-bit Oracle will be very straightforward. There will be no need to recreate databases, nor will a full export and import be

required. All that will be needed will be to copy the current datafiles 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 will be used for the 64-bit port. As a result, creating the new 64-bit Oracle software basically entails re-compiling, re-linking, re-testing and re-releasing the new version. Very little new code will need to be written during the move to 64-bit since the underlying operating system APIs are expected to remain substantially the same. In addition, since the Oracle database has already been ported to other 64-bit ports, moving to 64-bit is a straightforward process that will produce a quality, stable product in a very short period of time

File I/O Enhancements

The Oracle database on Windows supports a new 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 file system.

One other 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 being introduced in Oracle9*i* Release 2 to make administration and installation of Oracle clusters easier on Windows. In an effort to guarantee that all features of Windows are fully exploited by Oracle, the database supports 64-bit file I/O to allow the use of files larger than 4GB in size. In addition, physical and logical raw files are supported as data, log, and control files in order to enable Oracle Real Application Clusters on Windows and also for those cases where performance needs to be maximized.

Cluster File System

In Oracle9*i* Release 2, Real Application Cluster (RAC) manageability has been greatly improved through the introduction of an 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 the local disks of each node. 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. The Oracle software will be accessible by all nodes in the cluster, but controlled by none. All cluster machines have equal access to all the data and can process any transaction. In this way, RAC ensures full database software redundancy for Windows clusters while simplifying installation and administration.

64-bit File I/O

Internally, all Oracle9*i* Release 2 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, and 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. As a result, raw files are generally used only by very high-end installations and by Oracle Real Application Clusters unless the CFS is used.

To use a raw file, all Oracle needs to be told 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 GUI interface and maintains the links across the cluster and 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.

CONCLUSION

In summary, Oracle's database on 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 database server on Windows, 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, most well integrated database on the Windows platform on both the 32-bit and the 64-bit versions of Windows. For further information on Oracle's Windows products, please visit:

http://otn.oracle.com/tech/windows/

http://www.oracle.com/ip/deploy/database/oracle9i/index.html?windows.html

Oracle9*i* Release 2 Database Architecture on Windows
April 2002
Author: David Colello
Contributing Authors: Alex Keh

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

Oracle Corporation provides the software
that powers the internet.