# Technical Comparison of Oracle Database 10*g* and SQL Server 2000: Focus on Manageability

*An Oracle White Paper*
*Aug. 2004*

ORACLE

# Technical Comparison of Oracle Database 10*g* and SQL Server 2000: Focus on Manageability

# Technical Comparison of Oracle Database 10*g* and SQL Server 2000: Focus on Manageability

## EXECUTIVE OVERVIEW

Today acquisition costs represent only a small portion of the total cost of ownership of any system or software. The ongoing costs of managing and maintaining a system far outweigh the initial acquisition costs. Many factors contribute to this cost. Software that is not easy to manage requires highly trained administrators, making maintenance operations inherently expensive. Direct costs to businesses resulting from undue complexity translate into the increased expense of hiring, training, and retaining experienced personnel. There are also hidden costs due to poor availability and performance, since systems that are difficult to diagnose or tune tend to have poor availability. Furthermore, complex management procedures increase the likelihood of administrator error, further damaging availability. All these factors make it essential that database systems not only excel in performance, availability, scalability, but that they also be easy to use and manage. Manageability, therefore, is a key consideration today when evaluating any enterprise software or application.

The superiority of the Oracle Database in the areas of performance, scalability, and availability to its competitors, including SQL Server 2000, has never been in question. With Oracle Database 10g, the first self-managing database for enterprise systems, Oracle has established it dominance in the area of manageability as well. Oracle Database 10g automates administrative functions in all the key areas of database management such as installation, space & object management, performance diagnostics & tuning, and backup & recovery. It relieves the DBA from reactive and mundane management tasks, enhances their productivity, and allows them to focus more on strategic issues. In contrast, SQL Server 2000 lacks some basic capabilities that make its management challenging. It does not have most of the self-managing capabilities of Oracle, its administrators have practically no control over how a database or an application built on top of it performs, and to make matters worse, their DBAs are burdened with unnecessary management tasks such as managing locking conflicts. SQL Server 2000 does not provide any non-intrusive way of recovering from human errors either, which forces administrators to perform complicated database recovery operations and exposes valuable enterprise data

to grave risk. The list of such basic capabilities missing from SQL Server 2000 goes on and on. The Oracle Database 10g is distinctly superior to SQL Server 2000 and provides the most reliable, scalable and high performing data management solution available today that is also the easiest to use and manage.

## INTRODUCTION

"Our database administrators spend more than half of their time doing mundane management functions, and we expect more than half of this idle time to be cut using Oracle 10g."

Arthur Fleiss
Senior Manager
Colgate-Palmolive

A well-managed IT infrastructure is critical to the success of any modern business. As IT vendors deliver increasingly sophisticated solutions to meet the exacting demands of the "networked society", the task of systems management has never been more complex. Hiring highly skilled administrative staff to manage such complicated environments is an expensive proposition. This coupled with frequent shortage of experienced administrative personnel often results in spiraling management costs.

In order to meet these challenges, Oracle has made the enhanced manageability of its products one of its primary goals. The Oracle Database 10g introduces a sophisticated self-managing database that automatically monitors, adapts, and fixes itself. These self-managing solutions found in Oracle allow DBAs to become more productive, and helps their organizations reduce management costs without compromising service level objectives.

There are two aspects to Oracle's manageability strategy. Firstly, it seeks to make each of Oracle's products, particularly the database, as self-managing as possible so that they require minimal manual administration. The Oracle Database 10g represents a giant step towards achieving this vision. Its objective is to enhance administrator productivity and reduce the cost to businesses of managing enterprise database systems. The Second part of Oracle's manageability strategy focuses on making the management of completely data center easier, scalable and more effective. Oracle Enterprise Manager 10g provides this solution. Enterprise Manager 10g is the next generation HTML based tool that is the single, integrated solution for administering and monitoring applications and systems that are based on the Oracle technology stack. This means that businesses do not have to buy and integrate disparate set of third party software products for end-to-end management of their enterprise. These two pieces put together provide a complete, sophisticated and effective solution to address all management requirements for data centers.

SQL Server is generally accepted to be lagging Oracle in the areas of scalability, availability, and performance. This deficiency has manageability implications as well. It forces SQL Server DBAs to frequently resort to non-optimal and inefficient methods to overcome its shortcomings, which in turn enhance the complexity of their applications and makes them more difficult and expensive to manage. This fact was substantiated in a recent study conducted by an independent analyst firm, Edison Group, which showed that Oracle Database 10g is in fact significantly easier to manager that SQL Server 2000. The study

established that DBAs could complete typical administrative tasks in 30% less than with Oracle than with SQL Server and that performing these tasks was markedly more complex in SQL Server[1]. This fact was further corroborated by a research report published by INPUT, an independent market research firm, which found that the cost of running packaged applications such as SAP, Siebel, and PeopleSoft, was significantly higher in a SQL Server environment compared to Oracle[2].

In this paper we will take an in depth look at the various capabilities of the two systems and compare them in the area of manageability.

## MANAGEMENT TOOLS

"Oracle Enterprise Manager is the only way to manage a large number of databases effectively from a single location. It is an essential tool for us."
— Swiss Rail

Both Oracle and SQL Server offer GUI tools for the management and maintenance of their systems. Coincidentally, both the tools are called Enterprise Manager. Enterprise Manager 10g is Oracle's single, integrated solution for administering and monitoring applications and systems that are based on the Oracle technology stack. It provides the ability to seamlessly manage hundreds and thousands of systems located across organizational and geographical boundaries from a single point of control. Enterprise Manager 10g has a web-based architecture, which is robust, reliable, globally scalable, and is easy to deploy and operate within today's internet-enabled environments. It has an HTML-based console interface, which allows an administrator to manage from anywhere — all that is needed is a Web browser. Its robust grouping and task automation functionality provide core features that enable reliable, rapid, and secure automation of traditionally time-consuming, error prone tasks, such as application performance management, policy-based standardization and system provisioning.

Looking from a database-centric point of view, Enterprise Manager 10g significantly simplifies day-to-day database administration by combining simplicity with a rich feature set to offer a complete solution for managing an environment of any size regardless of platform. It automates many crucial management tasks and enhances administrator productivity by providing:

- Immediate out-of-box value by configuring and starting all the necessary framework component processes during database creation. Enterprise Manager is completely configured and ready for use, with the console automatically popping up in a browser window when a new database is created.

---

[1] Detailed study findings can be found at http://www.theedison.com/theedison/Research/gems/040401RDBMSCMCS.pdf

[2] Detailed study findings can be found at http://www.oracle.com/ip/deploy/database/oracle9i/db_sqlex.html

- Consolidated overview of the enterprise's health on the Home page. It shows availability, alert and job statuses for the database, configuration information, software maintenance advisories for critical patches, and the ability to drill down into more details in any of these areas.
- System monitoring and alert notification to anticipate problems and possibly fixing them automatically using "fix-it" jobs.
- EM2Go, the mobile component of Enterprise Manager, which enables on-the-road access and management of critical database management operations from hand-held devices such as PDAs.

"Hutchison-Priceline takes the added advantage of using Oracle Enterprise Manager to provide pro-active monitoring of all databases via a Personal Digital Assistant (PDA). This gives our database administrators the ability to comprehensively manage the Oracle systems from any location."
— Hutchison-Priceline

A large number of Oracle customers such as British Telecom, Boeing, Telstra, and Shell have standardized on Enterprise Manager to manage their production systems. Their experience is live testimony of how effective Oracle management tools are in boosting administrator productivity and increasing the number of databases each DBA can manage.

SQL Server 2000's Enterprise Manager tool lacks the depth of management capabilities provided by Oracle Enterprise Manager 10g. It does not have an HTML-based console interface and hence, requires installation of its client tool on all systems from which an administrator may want to manage the SQL Server database. This seriously restricts the ability of the administrators to manage their database from anyplace at any time. Another deficiency in SQL Server's management tool is its inability to provide end-to-end management and diagnostics of the entire system from a single console. Metrics relating to the O/S, middle-tier application servers, etc., cannot be monitored by the SQL Server's management tool.

A further difference between Oracle and SQL Server is in their management model. Oracle Enterprise Manager 10g follows the management-by-exception model, where DBA intervention is only needed when an exception condition occurs or is about to occur. Oracle highlights problem areas on its Home page such as space pressures, performance issues, etc., so that any potential problems or exceptions can be identified and dealt with immediately. This approach is very different from SQL Server's where DBAs themselves have to monitor the system for potential problems. For example, if a SQL statement is taking an inordinately long time, it will proactively be shown under the performance findings section of the Oracle Enterprise Manager 10g Home page. In comparison, for SQL Server, a DBA would have to first turn on tracing on the database (this has an obvious negative impact on performance), rerun the workload (this may not always be possible and is never convenient), and visually parse the trace files to determine most resource consuming SQL. Only then can the DBA know if there any problem SQL statements that are affecting system performance. Thus, SQL Server's management tool, unlike Oracle's, requires its users to invest considerably more time for common administrative

functions, is fundamentally reactive as opposed to proactive in its management approach, and lacks many of the management capabilities provided by Oracle.

## STORAGE MANAGEMENT

Database administrators face many challenges in the area of storage management. As the rapid growth in database size and the demand for uptime continue to increase, taking the database offline to perform maintenance operations such as changing disk configurations, is becoming increasingly difficult to schedule. In addition, the process of working with system administrators to locate and remove I/O hot spots or fragmentation can be grueling. At the same time the pressure of cost reduction no longer allows the growth in the DBAs staff to keep pace with the database growth rate. This results in what some refer to as the "management gap". DBAs need better tools that can increase their productivity and help automate many of their manual tasks.

Automatic Storage Management (ASM) is a feature of Oracle Database 10g that provides the solution for these storage management challenges by automating this entire function. All an administrator has to do is to allocate a set of storage devices to a database instance and the Oracle's ASM solution takes care of the rest. It automates the placement of the database files within those disk groups and spreads data evenly across all available storage resources to optimize performance and utilization. This even distribution of database files makes the manual I/O performance tuning obsolete. ASM also provides three mirroring options for protection against disk failure: none, two-way, and three-way mirroring. Furthermore, ASM enables the DBA to change the storage configuration without having to take the database offline. ASM automatically rebalances files across the disk group after disks have been added or dropped.

ASM is a feature specifically built to simplify the job of storage management. It is a capability that saves the administrators time and provides flexibility to manage a dynamic database environment with increased efficiency.

SQL Server does not have any storage management capabilities like those offered by Oracle. Storage management has to be done manually or alternatively, a separate tool has to be purchased to facilitate this function.

## SPACE & OBJECT MANAGEMENT

### Proactive Space Management

Oracle provides proactive space management capabilities with its space monitoring, notification, and space trending features. Rather than using external tools to poll the database to monitor space usage — something that is

inherently resource intensive — Oracle database server has in-built intelligence that provides a non-intrusive and timely check of space usage in the database. If the space usage of a tablespace crosses a user-defined threshold, Oracle server will issue an alert to forewarn the appropriate person — DBA — that the database is running out of space so that corrective measures can be taken. By default, Oracle will issue a warning alert when a tablespace is 85% full and a critical alert when it is 97% full. A DBA can override these defaults for a given tablespace or set a new default for the entire database. Once corrective measures have been taken, the alerts are cleared automatically by the server.

Oracle also provides a sophisticated alert notification system. DBAs can receive alerts via email, pagers, Enterprise Manager, or directly by querying views inside the database server.

The key aspect of space management in Oracle is its efficiency. The database server tracks space utilization while performing regular space management operations, e.g., allocating new extent. A background process compares the state of the tablespace against the thresholds specified to determine if an alert should be raised or cleared. If a state transition has occurred, appropriate action is taken and an alert is either triggered or cleared.

Finally, Oracles' proactive space management is enabled by default and causes no measurable performance impact on the system.

SQL Server also has the capability to monitor space usage, but it differs from Oracle in several aspects. First, space usage monitoring is done using the SQL Server agent, whose primary function is job scheduling and alert handling. Thus, the monitoring and alerting mechanism is based on polling of database meta data and is intrinsically inefficient and expensive. Second, the space monitoring and alerting infrastructure in SQL Server is not as flexible as Oracle's. SQL Server lets you set space usage thresholds for alerting at the database level only. This means that if you have two different filegroups (tablespace in Oracle terminology), one for tables and the other for indexes and you want to set two different alert thresholds for them — a very basic requirement — you cannot do it. Oracle, on the other hand, allows you to set alert thresholds at the database or the tablespace level. This added granularity in monitoring gives Oracle DBAs more flexibility and precision is setting space usage guidelines for honoring service level agreements.

"In previous years, due in part to the extremely large volume of data maintained at Amazon, we could spend hours with our systems offline while we performed indexing operations. Online indexing operations have eliminated this downtime, and helped us optimize performance and availability throughout the site."

Matt Swann
Director, DB Services
Amazon.com

### Online Reorganization

Oracle Database 10g enables database administrators to perform almost all segment maintenance operations online, i.e., data is fully available for queries, updates, and deletes. Schema evolution allows table definitions to be modified while the data table is in service. Tables can be relocated, defragmented, reorganized or have their storage parameters changed. Indexes can be added, rebuilt or defragmented online.

SQL Server 2000 does not support the extensive set of online operations as Oracle Database 10g. It does not have online schema evolution, table reorganizations, table redefinitions and secondary index creations on index-organized tables. Since many of these maintenance operations are done frequently and can take hours to complete, SQL Server 2000 applications can suffer significant data unavailability. As a result, SQL Server DBAs have to skillfully devise elaborate schedules on when to perform these operations in order to minimize the impact on system unavailability. This significantly adds to the management workload of the DBA, and is something that Oracle DBAs do not have to consider at all in their day-to-day operations.

### Space Defragmentation

Handling space fragmentation is an area where DBAs have historically faced a lot of challenges. Oracle Database 10g provides a comprehensive solution for eliminating space fragmentation, external and internal, online. Oracle's locally managed tablespace feature when used in conjunction with online segment shrink functionality completely eliminates space fragmentation. Locally managed tablespaces prevent fragmentation within a tablespace (called external fragmentation) that results when segments have different space attributes and grow at different rates. The online segment shrink capability removes fragmentation within a data block of a segment (called internal fragmentation) that occurs when tables and indexes are heavily updated, leaving behind unusable free space in the data block. Additionally, Oracle Database 10g comes with a Segment Advisor that identifies segments that have significant internal fragmentation to warrant the online segment shrink operation. Thus, Oracle Database 10g has a complete solution that enables efficient space utilization and enhances query performance by eliminating all forms of space fragmentation.

SQL Server has a partial solution in this area. It has the capability for removing fragmentation from indexes and tables with clustered indexes, but does not have a similar solution for regular tables. Furthermore, the process for identifying which segments have fragmentation is entirely manual in SQL Server. DBAs have to manually run the SHOWCONTIG utility and analyze its output to determine whether indexes and tables are fragmented. Once this determination is made, they can then only defragment indexes, whereas tables have to be manually recreated for defragmentation.

### Partitioning

Partitioning allows large database structures (tables, indexes, etc.) to be decomposed into smaller and more manageable pieces and, at the same time, improves query performance and resource utilization. The Oracle Database 10g offers several partitioning options designed to fulfill different requirements[3]:

- Range partitioning uses ranges of column values to map rows to partitions. Partitioning by range is particularly well suited for historical databases. Range partitioning is also the ideal partitioning method to support 'rolling window' operations in a data warehouse.
- Hash partitioning uses a hash function on the partitioning columns to place data into partitions. Hash partitioning is an effective means of evenly distributing data for optimal performance.
- List partitioning allows users to have explicit control over how rows map to partitions. This is done by specifying a list of discrete values for the partitioning column in the description for each partition.

In addition, Oracle supports range-hash and range-list composite partitioning.

Oracle also provides three types of partitioned indexes:

- A local index is an index on a partitioned table that is partitioned using the exact same partition strategy as the underlying partitioned table. Each partition of a local index corresponds to one and only one partition of the underlying table.
- A global partitioned index is an index on a partitioned or non-partitioned table that is partitioned using a different partitioning-key from the table.
- A global non-partitioned index is essentially identical to an index on a non-partitioned table. The index structure is not partitioned.

Oracle allows all possible combinations of partitioned and non-partitioned indexes and tables: a partitioned table can have partitioned and non-partitioned indexes, and a non-partitioned table can have partitioned and non-partitioned indexes.

SQL Server 2000 does not support table or index partitioning but has what are called partitioned views. A partitioned view joins horizontally partitioned data from a set of member tables across one or more servers, making the data appear as if from one table. The data is partitioned between the member tables using range partitioning concepts. Each member table has a CHECK constraint that defines the range on the partitioning column, ensuring that the right set of data goes to the right member table. A view is then created that uses UNION ALL to join all the members tables, thus creating a single partitioned view.

---

[3] For more information about Oracle9*i*'s partitioning options, see Oracle9*i* Partitioning, Hermann Baer, Technical White paper, Oracle Open World 2001, Berlin.

Partitioned views, which the Oracle Database has supported since version 7.3, solve a small part of the challenges involved in managing large data sets. While partitioned views provide a simple way for users to select data from a set of underlying tables, each of these tables must be managed separately. For example, data must be loaded in each table individually, indexes must be created on them separately, etc. Also, a DBA has to manually create and maintain triggers on each underlying table in the view. Therefore, from a DBA perspective, partitioned views do very little to simply the management of large volumes of data.

The table below summarizes some of the other differences between Oracle Database 10g and SQL Server 2000 with regard to the partitioning options:

*Table 1: Partitioning options.*

| **Feature** | **Oracle** | **SQL Server** |
|---|---|---|
| Range partitioning | Yes | Yes (partition views) |
| List partitioning | Yes | No |
| Hash partitioning | Yes | No |
| Composite partitioning | Yes | No |
| Local index | Yes | Yes |
| Global portioned index | Yes | No |
| Global non-partitioned index | Yes | No |

SQL Server 2000's partitioned views do not support global indexes. This means that queries that do not specify a search condition on the "partition" column will have to search all the tables in the partitioned view, since there are no global indexes to let the system know which "partition" contains the desired data. This is a major problem in OLTP environments where global indexes are a must for efficient data access. In partitioned view configurations, SQL Server application designers have no flexibility in defining their indexing strategies. This lack of support considerably restricts the ability of SQL Server 2000 to be used in real-world OLTP applications.

Another advantage that Oracle has over SQL Server is that it offers much easier 'rolling window' support. A 'rolling window' allows the data for a defined period, e.g., a week or a month, to be kept online by continuously replacing the oldest data with the most recent one. In SQL Server, all the steps associated with 'rolling window' support, such as creating new member tables, creating constraints, etc., have to be done manually. Oracle's Change Manager, on the

other hand, provides a user-friendly, intuitive tool that provides complete 'rolling window' support requiring no manual steps at all.

All these limitations in SQL Server make the task of managing large volume of data quite complicated. Even Microsoft has finally acknowledged SQL Server's weakness the area of partition management. Peter Spiro, leader of Microsoft's SQL Server engine team and an 8-year veteran of the database group, in the October issue of the SQL Server Magazine agreed that in SQL Server, "*partitioned views are difficult for customers to use*." He goes on to say that "*distributed partitioned views are a stopgap measure for people who are using multiple machines*."[4] Thus, Oracle's partition management solution is far superior to SQL Server, as it offers a comprehensive set of features that not only enhance manageability but also improve performance.

### Row Size Limit

SQL Server imposes a limit of 8060 bytes (8KB) on the size of a row[5]. This is because in SQL Server, just like in IBM DB2, a row cannot span multiple pages. Since the maximum size of a SQL Server page is 8KB, a row — after discounting for overhead — can be no larger than 8060 bytes. For tables with rows longer than 8 KB, a SQL Server DBA has to somehow overcome this limitation. In such instances, a common practice among SQL Server DBAs is to vertically divide the table into multiple tables. Whereas this solves one problem by overcoming the 8KB restriction, it creates another by compromising the optimality of the application design. This is yet another area that an Oracle DBA does not have to worry about. Oracle allows a row to span across multiple pages (or blocks in Oracle parlance) and imposes no limits on its row size.

### Handling Out-of-Space Situations

Operations such as data loads or batch updates can encounter errors when they run out of space. Sometimes these errors occur just when the operation is about to finish. Oracle's Resumable Space Allocation feature enables it to handle such errors in a very graceful manner. Whenever an operation encounters an out-of-space situation, it is held in a "suspended" state while an alert is generated to notify the administrator of the problem so that the problem can be fixed. The "suspended" operation automatically resumes as soon as the error condition is corrected. In case of a transient problem, such as a query running out of temporary space, no administrator intervention may be required since Oracle will resume the operation automatically as soon as the transient problem disappears. Virtually any kind of operation, be it a PL/SQL or Java stored procedure, a DML or DDL statement, or an export/import or loader session, can all be run in the "resumable" mode.

---

[4] *Revving the SQL Server Engine*, SQL Server Magazine, October 2002, ID #26435.
[5] http://www.devx.com/codemag/articles/2002/March/sqlconfig/sqlconfig-4.asp

This capability is unique to Oracle with no parallel in SQL Server. It saves Oracle DBAs enormous time that their SQL Server counterparts spend in monitoring and re-executing failed long running operations. In SQL Server, if an operation runs into an out-of-space situation, the entire operation has to be repeated after the space issue has been addressed. This not only results in wastage of time but can also hamper normal database performance, for instance, by forcing the DBA to re-run a failed batch job during normal workload hours. Thus, Oracle's Resumable space allocation feature avoids all the pitfalls that a SQL Server user would encounter by offering a graceful way to manage out-of-space situations.

## PERFORMANCE MANAGEMENT

### Performance Diagnosis

"I'm the only DBA in the department, … 10g's offering so many automation features. I'm excited about production with 10g because it allows me to automate day-to-day stuff: managing table spaces, and performance tuning, more than anything. It offers you a fix on the fly without having to go through the process of finding it out yourself."

Jeremy Forman
New Mexico Department Of Transportation

Performance diagnosis is a very critical DBA function that traditionally has proven to be both complex and time consuming. Oracle Database 10g provides automatic performance diagnosis and monitoring technology, built inside the database server, that automates the entire performance diagnosis process and provides detailed information on the cause of performance problems and their remedies.

#### Automatic Workload Repository (AWR)

Oracle Database 10g maintains a repository of information on the operation of the system called the Automatic Workload Repository (AWR). This infrastructure is a key building block for Oracle's performance diagnostic solution. It runs automatically to collect data about the operation of the Oracle system and stores it in the database. Being part of the database kernel, AWR captures the most relevant data in the most optimal fashion. It is designed to be lightweight and self-managing. Data is captured every 60 minutes and is purged after 7 days automatically. Both the frequency and length of time for which data is kept can be configured. Additionally, AWR can also be run on-demand to capture information at specific times deemed to be interesting.

AWR captures all of the data required to perform a thorough system or user level performance analysis. By proactively capturing this data it obviates the need to re-run the workload for problem diagnosis purposes.

AWR provides the basis for improved performance diagnostic facilities in Oracle Database 10g, and it is on AWR data that the Automatic Database Diagnostic Monitor, Oracle's performance diagnostic engine, performs analysis to identify and remedy performance problems.

#### Automatic Database Diagnostic Monitor (ADDM)

The Automatic Database Diagnostic Monitor (ADDM) is the solution that proactively analyzes the wealth of diagnostics statistics available in the AWR and tells the DBA exactly what the problem is and how to fix it. Because ADDM is part of the database kernel and is not an external tool, the diagnosis is done at a much lower cost (both in terms of money spent and system resources utilized) compared to traditional monitoring systems. The code instrumentation put in place to support the capabilities also enhances the functionality for real-time reactive tuning method supported by Oracle Enterprise Manager 10g.

ADDM runs proactively every hour and reports on the health of the database. AWR, by default, has a retention period of 7 days for all performance related information, and this allows the DBA to diagnose any problem that occurred within the retention period time window. All a DBA has to do is to look at the ADDM report for the relevant time period and implement its advice.

The key benefits provided by ADDM can be summarized as follows:

1. Automatic performance diagnostic report every hour.
2. Comprehensive problem diagnosis based on decades of tuning expertise by Oracle experts.
3. Time-based quantification of problem impacts and recommendation benefits.
4. Identification of root cause, not just symptoms.
5. Greatly reduced need to replay workload for detailed analysis due to completeness of the data held in the AWR.

"The self-managing features are the best, because they tell you where the problem is and give you what to do to fix the code ... If you choose to deploy those fixes, it's also automated, and for that I have to really thank Oracle."

Arvind Gidwani
Staff Manager
Qualcomm

SQL Server has nothing remotely resembling Oracle Database 10g's self-diagnostic capabilities. All performance problems have to be manually diagnosed. SQL Server does provide tools such as SQL Profiler for capturing workload traces and provides GUI interfaces to look at the captured data for problem identification. Whereas this does facilitate problem diagnostic to some extent, the process still requires an expert performance engineer to read and interpret the data collected, and provides little help in advising a remedy for performance problems. The other limitation of SQL Server's approach is that it is intrinsically reactive. For example, if the SQL Server system is running slow, a DBA would first have enable tracing using SQL Profiler and then rerun the workload, before any analysis of performance issues such as poorly performing SQL, can take place. In contrast, if an Oracle system is performing below expectations, A DBA simply needs to look at the current ADDM report to identify the real cause and its remedy.

Oracle stands out in the area of performance diagnosis compared to SQL Server. It provides a solution that is comprehensive and powerful such that even novice users can fix performance problems with ease. SQL Server, on the

other hand, still employs manual tuning approach that is reactive, requires skill and expertise, and is time-consuming.

### Identifying and Tuning High-load SQL Statements

DBAs can spend a significant amount of their time identifying and tuning resource intensive SQL statements. In this area as well, Oracle automates the function completely. High-load SQL identification is performed proactively by ADDM. An Oracle DBA has to just look at the ADDM report, which is produced every hour automatically, to see if any high-load SQL statements need attention. In addition to ADDM, Oracle also captures high-load SQL statements hourly in the Automatic Workload Repository (AWR). These statements are displayed in the TopSQL page of Enterpriser Manager 10g.

Identification of high-load SQL statements in SQL Server is a more involved process. A DBA has to first setting up a trace event through the SQL Profiler, enable the relevant events and then re-run the workload so that run-time statistics are captured in the trace files. Once trace files have been generated, they then have to be analyzed manually for high-load SQL identification. This is a very significant manageability issue for SQL Server administrators for a couple of reasons. First, monitoring resource consuming SQL statements is a common database administrative task and hence, spending anything more than a few minutes to get this information is often unacceptable. Second, for a busy system, the generated trace file can easily grow to a very large size quite rapidly, making it extremely difficult to extract meaningful information from it. SQL Server very clearly lags behind Oracle in this basic but very important DBA task.

The next step is tuning the SQL. Oracle Database 10g has the SQL Tuning and Access Advisors that automate this process as well. They take one or more SQL statements as input and invoke the query optimizer in a special tuning mode to comprehensively tune the SQL statements. Four kinds of analysis are performed here:

- **Statistics Analysis**: The query optimizer needs up-to-date object statistics to generate good execution plans. In this analysis objects with stale or missing statistics are identified and missing statistics collected. It should be noted that Oracle Database 10g automates the collection of statistics. As a result, this problem will generally only be seen when the automatic statistics collection feature has been deliberately disabled.
- **SQL Profiling**: This is a new feature introduced in Oracle Database 10*g* (not to be confused by the SQL Profiler tool of SQL Server, which is basically a GUI interface for trace file management) that revolutionizes the approach to SQL tuning. It eliminates the need for using optimizer hints for manually tuning a SQL statement and tunes the statement without requiring any change to the SQL code. In this analysis, a profile of a SQL statement, called a SQL Profile, is built consisting of auxiliary statistics

*specific to that statement*. The query optimizer can sometimes generate sub-optimal execution plans due to the lack of information about the correlations between the different objects in a query. Traditionally, both Oracle and SQL Server handled this problem by manually adding query hints to the code. SQL Profile eliminates the need for this manual process in Oracle by collecting additional information using sampling and partial execution techniques.

- **Access Path Analysis**: Indexes and materialized views can tremendously enhance performance of a SQL statement by reducing the need for full table scans. In this analysis, new indexes and materialized views that can significantly enhance query performance are identified and recommended.
- **SQL Structure Analysis**: Problems with the structure of SQL statements can lead to poor performance. These could be syntactic, semantic, or design problems with the statement. In this analysis relevant suggestions are made to restructure the SQL statements for improved performance.

By performing the afore-mentioned analysis and recommending appropriate actions, Oracle Database 10g with its SQL Tuning and Access Advisors offers a comprehensive SQL tuning solution that obviates the need for the slow, tedious and expensive manual tuning process.

SQL Server has only a partial solution in this area. It offers no way of tuning a SQL other than by way of manually adding query hints, nor does it provide any help in rewriting poorly structured SQL queries. All these have to be done by hand and require deep understanding of SQL Server optimization techniques. The only area where it does assist is with the creation of indexes thorough its Index Tuning Wizard, while leaving the more challenging and time consuming aspects of SQL tuning to the DBAs and application developers.

Besides the comprehensiveness and simplicity of Oracle's tuning solution, another key differentiator between Oracle and SQL Server solutions is that the tuning in Oracle is actually carried out by the query optimizer itself and not by an external tool trying to coerce or trick the optimizer through various approaches to generate a better execution plan. This is a significant advantage for Oracle compared to tuning tools provided by other vendor, including Microsoft, as the tuning actions performed by the query optimizer are superior in quality, reliability, and effectiveness than anything an external component can do.

### Tuning Packaged Applications

Tuning packaged applications like SAP, Siebel, or PeopleSoft, offer a different type of tuning challenge. Since customers do not have the privileges to modify application code, any time a serious performance issue is encountered in a packaged application, a customer has to log a bug with the application vendor, and then wait for several weeks, months or more, to receive a code fix for the

problem. Historically, this has been the only option available to customers of database management systems and with SQL Server 2000 this is true even today. Oracle, on the other hand, offers an alternative to its customers. Its SQL Profile feature, which is part of the SQL Tuning Advisor, tunes SQL statements without changing the application code. As described earlier, SQL Profiles consist of information pertaining to the data correlations between the different tables referenced in a SQL statement, and by providing this additional information to the query optimizer, tune the SQL. Because SQL Profiles are stored in the data dictionary and not in the application code, they make it possible for Oracle to tune packaged applications. Oracle is the only commercial database in the market today that has this functionality. Oracle customers do not have to file bugs nor do they have to wait to get their performance problems fixed. With SQL Profiling tuning is immediate.

### Execution Plan Stability

DBAs crave predictability. One of the areas where DBAs desire it the most is in query performance. The performance of a query depends on the optimality of its execution plan. The more efficient an execution plan, the better the query performance. However, cost based optimization — an optimization model used by both Oracle and SQL Server — among other factors also relies on the database environment in generating an execution plan for a query. As a result, changes in database environment such as optimizer statistics of the underlying objects, memory parameters, etc., can suddenly change the execution plan of a query, mostly improving it but sometimes making it drastically worse.

This is a serious problem in SQL Server, not only because the database has no way of providing execution plan stability, but also because the database server, by default, continually regenerates optimizer statistics, hence opening up the possibility of continually changing execution plans. The only thing a DBA can do in this situation is to turn off auto-creation and update of optimizer statistics so that execution plans do not change. But this opens up a new set of problems because now the table and index statistics can become stale and the optimizer will be forced to generate execution plans based on stale data, resulting in poor optimization. SQL Server has no real solution to the issue of execution plan stability, and in fact its default behavior of continually updating optimizer statistics actually exacerbates the problem.

Unlike SQL Server, Oracle provides DBAs the ability to freeze execution plans by creating *stored outlines*. This allows the DBA to ensure plan stability for those queries where they cannot risk changes in execution plans, as well as during critical periods, e.g., quarter-close. This gives the DBA the predictability in query behavior that they crave without compromising good administration practices like regularly refreshing optimizer statistics. In this way, Oracle presents an elegant solution to a serious performance management concern of DBAs, whereas SQL Server offers no practical way of dealing with the issue.

### Managing Locking Conflicts

One of the biggest strengths of Oracle from the performance management point of view comes from its fundamental architecture. Rejecting the old model based on lock-based concurrency control — used by SQL Server — Oracle chose to implement its own multi-version read consistency model ensuring that readers and writers never block each other. In Oracle, whenever a change is made by a transaction, the original data values are copied in database structures called undo segments. Consequently, unlike SQL Server, which uses locks to prevent records from being changed by others while being read, or to prevent queries from reading uncommitted changes, Oracle uses the undo information stored in the database to construct a read-consistent version of data. This ground-breaking technology allows the Oracle database to service queries without requiring any read locks.

"Locking at a smaller granularity, such as rows, increases concurrency, but has a higher overhead because more locks must be held if many rows are locked. Locking at a larger granularity, such as tables, are expensive in terms of concurrency because locking an entire table restricts access to any part of the table by other transactions, but has a lower overhead because fewer locks are being maintained".

Microsoft SQL Server documentation

SQL Server 2000 does not provide multi-version read consistency. Instead it requires applications to either use shared locks for read operations with various levels of isolation, or to accept dirty reads. Shared locks prevent data that is read from being changed by concurrent transactions. Clearly, this implementation restricts the ability of the system to properly service concurrent requests in environments involving a mix of reads and writes, as explained in Microsoft's documentation: *"SQL Server, in contrast, uses shared locks to ensure that data readers only see committed data…. A reader waits for a writer to commit the changes before reading a record. A reader holding shared locks also blocks a writer trying to update the same data."* As a consequence, releasing locks quickly for applications that support high numbers of users is far more important in SQL Server than in Oracle.[6]

The only alternative developers have to this problem is to build separate workload environments, where intensive read activities, such as reporting, cannot interfere with on-line transactional applications. Regardless of which approach is used, SQL Server 2000 developers usually have to find some compromise in their application design in order to get acceptable data concurrency and accuracy.

In Oracle, writers and readers never block each other. Oracle's powerful multi-version read consistency allows mixed workload environments to function properly without incurring any performance penalty for the users.

Another problem that occurs as a result of SQL Server's lack of multi-version read consistency and its locking model is lock escalation. SQL Server puts a limit on the maximum numbers of locks that can be supported in a database. This coupled with the fact that SQL Server uses many more locks (due to read locks), means that as the transaction volume increases, databases can easily reach a threshold value at which row level locks will escalate to page or table level locks to conserve memory. This in turn means that fewer users can access

[6] *Migrating Oracle Databases to SQL Server 2000*, SQL Server Resource kit, p. 57.

the data at the same time — users will have to wait — and the chances of getting false deadlocks is also greatly increased. According to SQL Server documentation, "*Locking at a smaller granularity, such as rows, increases concurrency, but has a higher overhead because more locks must be held if many rows are locked. Locking at a larger granularity, such as tables, are expensive in terms of concurrency because locking an entire table restricts access to any part of the table by other transactions, but has a lower overhead because fewer locks are being maintained.*[7]"

Oracle imposes no limit on locks, its locks never escalate and, as a consequence, Oracle users never experience false deadlock situations due to lock escalation. Oracle DBAs, therefore, need not even think about some of the most time consuming tasks that their SQL Server counterparts have to perform on a daily basis i.e., monitoring lock escalations and resolving deadlocks.

### Managing CPU Resources

The ability to easily and accurately perform system and resource management is critical to maintaining application and database performance, scalability and availability. The Oracle Database Resource Manager enables administrators to align the distribution of system resources with enterprise goals by allowing allocation of CPU resources among database users and applications according to business priorities. Its ability to automatically limit the resources consumed by batch jobs helps in ensuring that such operation do not adversely impact online users in a mixed workload environment. Furthermore, Database Resource Manager also provides the ability to limit the number of concurrent long operations and prevent execution of highly resource intensive queries during certain times of the day. The Database Resource Manager, therefore, makes it extremely easy to deliver predictable service level with minimal human intervention and facilitates almost unlimited system scalability without compromising performance[8].

SQL Server has no feature that compares to the Oracle Database Resource Manager. Its DBAs have no control over how CPU resources are allocated which means that important processes can be starved for CPU while those not so important can unduly monopolize it. This is because the database server has no way of distinguishing between higher and lower priority processes. To overcome this shortcoming, SQL Server DBAs have to constantly monitor resources consumed by various processes and when necessary take corrective action manually to ensure efficient use of CPU resources. An obvious problem with this approach is that in most cases a DBA may not have a good option for

---

[7] Microsoft SQL Server documentation: Understanding Locking in SQL Server http://msdn.microsoft.com/library/default.asp?url=/library/en-us/acdata/ac_8_con_7a_7xde.asp

[8] For more details please refer to paper titled *Oracle9i Database Resource Manager* at http://otn.oracle.com/products/manageability/database/pdf/9i_Resource_Mgr_TWP.pdf.

corrective action, and secondly, manually controlling CPU resources is a very time-consuming and difficult task with limited chances of success.

## BACKUP & RECOVERY

### Automatic Backups

Oracle automatically creates and schedules a predefined backup job during database creation that implements the Oracle recommended strategy for backup management. The Oracle recommended strategy calls for backups to be made as follows:

- Perform full database backup once
- Subsequently perform incremental backups
- Update full database backup image copy with incremental backups.

This strategy is a sophisticated approach that utilizes several new features of Oracle Database 10g such as the Flash Recovery Area, Incremental Backups, and Incrementally Updated Backups and keeps the time required for media recovery of the database to a minimum.

In SQL Server, backup management is the burden of the DBA. A DBA has to first understand SQL Server architecture, learn about all the backup and recovery features that it provides, and then devise a strategy that provides the best possible recovery for the business needs. Once all this is done, the DBA then has to create the backup job that implements the strategy. This is another area where a DBA has to worry about one less thing in Oracle than in SQL Server.

### Backup Files Management

A database backup consists of multiple files of different types such as data files, transaction log files, redo log files, control files, etc. As regular backups are taken over time, these files can grow in number and managing them manually can quickly become a non-trivial matter. For DBAs to manually manage backup files would require that they ensure that all the different types of files necessary for recovery are indeed backed up properly; they must have a reliable mechanism of tracking the backups in order to determine whether they have become redundant and hence, can be deleted; they must diligently monitor backup location to guarantee that sufficient space is available for new backups; etc. Even for a small system, to do this manually can be burdensome. Oracle Database 10g automates the management of all backup files by introducing the Flash Recovery Area feature. The Flash Recovery Area is a unified storage location for all recovery related files in an Oracle database. By defining one initialization parameter, DB_RECOVERY_FILE_DEST, all RMAN (Recovery Manager) backups, archive logs, control files, and data file copies are automatically written to a specified disk location that is auto-managed by

Oracle. When there is space pressure, the Flash Recovery Area automatically deletes obsolete backups and archive logs that are no longer required based on retention policy specified by the DBA. If you set the retention policy to a recovery window of 7 days, then Oracle will retain all backups required to recover the database to 7 days in the past. With the Flash Recovery Area feature, Oracle Database 10g offloads yet another mundane task off the DBA workload, and makes media recovery operations faster, simpler, and more reliable.

SQL Server also provides the ability to manage backup files but it does not have the same level of sophistication as that offered by Oracle. It has a backup wizard that backs up the relevant files but, unlike Oracle, it does not have an intelligent mechanism to clean up the backup location automatically. In SQL Server a DBA can specify a time duration beyond which all files in the backup location expire and consequently can be deleted. The approach is too simplistic to be useful. In an environment where backups consist of a mix of full and incremental backups — a backup strategy that is fairly common nowadays — SQL Server does not have the intelligence to recognize the difference between the two types of backups and will maintain the redundant incremental backups even if the system is under space pressure as long as they not older than the predefined expiration time. A DBA will actually have to manually separate the files that are needed from those that are not in case the system is running out of space and unnecessary files need to be removed. Another fact to keep in mind is that in SQL Server terminology, an instance typically consists of several databases (a SQL Server database can be considered equivalent to an Oracle tablespace.) This makes matters even more complicated because a DBA has to not only track files needed files for a single database, but also has to correlate and maintain them across multiple (SQL Server) databases in order to ensure that all requisite files are available if a recovery operation becomes necessary. This adds to the burden of the DBA and further opens up the possibility for human errors leading to failed recovery situations.

### Automatic Log Archiving

In order to recover from media failures, both Oracle and SQL Server require the redo logs (or transaction logs in SQL Server terminology) to be backed up. Oracle automatically backs up the redo logs by a process called *archiving,* which is enabled by simply putting the database in archivelog mode. In contrast, the transaction logs in SQL Server, once they fill up, have to be backed up by the DBA manually. To avoid this manual task, DBAs will often set up a batch job that runs periodically to back up the transaction log. This is not really an acceptable solution because batch jobs cannot adapt to changes in workload. The batch job frequency is based on expected workloads, and if for some reason the database undergoes a higher than expected transaction volume, then the transaction log can easily grow and use up all the free space, resulting in the database coming to a halt. On the other end of the spectrum, if a database has

very little or no transaction volume, the batch job will still back up the transaction log even though it might be empty. In short, SQL Server's lack of automatic archiving necessitates that DBAs carefully monitor the transaction logs and must back them up as required. An Oracle DBA is completely free of worries in this area.

### Backup Throttling

Both Oracle and SQL Server offer online backup capabilities. However, a truly online backup should not only ensure that the database stays up during the backups but must also limit its performance impact so that users can continue to use the database. This is extremely important in today's round-the-clock economy since users do not care to differentiate between an unavailable and a poorly performing system. Oracle Recovery Manager (RMAN) allows administrators to throttle the backup read rate in order to contain the performance impact within an acceptable limit. There is no way to achieve this in SQL Server. As a result, when a SQL Server database is being backed up, users may experience performance degradation, since there is nothing a DBA can do about it other than to optimize the system for maximum I/O bandwidth so as to speed up the backup process[9]. This limits the flexibility of a SQL Server DBA significantly as far as scheduling a backup is concerned since they have to be extra careful not to disrupt normal system functioning.

### Self-Contained Backups

In Oracle, the backups are complete and fully self-contained. A DBA can recover an Oracle database from any situation as long as there is a good backup of the database. This is not the case in SQL Server. If the SQL Server system database, *msdb*, is lost, a DBA cannot recover the system without undergoing a frantic search for the original install CD even though he/she may have performed regular backups. This is because SQL Server backups are not self-contained. A DBA must first manually recreate the system database, *msdb,* from the original install CD using the command line utility, *rebuildm.exe*[10]. The recovery of application databases can be started only after the master database has been recovered. This makes the complete recovery of a SQL Server instance quite complex and highly intuitive unlike Oracle where the same operation can be performed simply using the recovery wizard. Even in this most critical DBA function of recovering a down database instance, SQL Server poses a serious manageability impediment to its DBAs.

### Recovery from Human Errors

Numerous availability studies have highlighted human error as one of the most prominent causes of application outage. While it is possible to develop

---

[9] Whalen, Edward, et al., Microsoft SQL Server 2000 Performance Tuning, p. 281.
[10] SQL Server Books Online: How to rebuild the master database.

safeguards against hardware and software failures, it is nearly impossible to insulate a system from human mistakes such as accidentally deleting critical data. Once again, only Oracle provides easy and completely non-intrusive options to recover from such failures leveraging its Flashback technology features. The Flashback technology provides a set of new features to view and rewind data back and forth in time. The Flashback features offer the capability to query past versions of schema objects, query historical data, perform change analysis or perform self-service repair to recover from logical corruptions while the database is online. With the Oracle Database 10g's Flashback technology, you can indeed go back in time to undo your mistakes!

#### Recovery of Dropped Table

Oracle's Flashback Drop feature allows instant recovery of a table dropped by mistake. When a user drops a table, Oracle automatically places it into the "Recycle Bin". The Recycle Bin is a virtual container where all dropped objects reside. Objects remain in the Recycle Bin until either Oracle needs to reclaim the space in order to accommodate new data or the owner of the dropped objects decides to permanently remove them using the new PURGE command. As long as a dropped object remains in the recycle bin, it can be recovered using a simple SQL statement:

```
FLASHBACK TABLE <table_name > TO BEFORE DROP;
```

#### Recovery from Logical Data Corruptions

Due to user errors, e.g., a batch job being run twice, logical corruptions can occur in the database. In such instances, the most common remedy is to restore the database to a time prior to the action that caused the corruption took place. Traditionally, this meant first restoring the database from a backup and then performing point-in-time media recovery on it. This form of recovery can be tedious and usually takes a long time. This is the only option available in SQL Server. Oracle, on the other hand, offers the traditional media recovery option as well as a distinctly faster and simpler solution based on its Flashback technology. Flashback supports recovery at any level of granularity including database, table, transaction, and row level.

- *Flashback Database*: For widespread data corruptions, this offers the best option. Flashback Database quickly rewinds an Oracle database to a previous time specified by the user using a special type of file called Flashback Log. This file captures previous images of the data blocks as they change over time. The bigger this file, the more the database can be rewound back. DBAs can size this file based on the recovery requirements of their business. Recovering a database using the FLASHBACK DATABASE command has no down time associated with restoring a backup, and makes recovery from logical data corruptions caused by human errors very easy and fast.

- *Flashback Table*: For data corruptions that affected only a small set of tables, this offers a more targeted recovery option. Flashback Table provides the DBA the ability to recover a table or a set of tables to a specified point in time in the past quickly and online. It restores the tables while automatically maintaining its associated attributes such as the current indexes, triggers and constraints, and does not require the DBA to find and restore application specific properties. Flashback Table alleviates the need for performing more complicated recovery operations on the entire database.
- *Flashback Transaction*: This feature allows more granular recovery by reversing the specific transactions that caused the corruption. It provides a mechanism to view changes made to the database at the transaction level and reversing them.
- *Flashback Versions Query*: This allows the recovery at row level. It provides a mechanism to view changes made to the database over time at the row level and gives the user to ability to selectively undo any undesired changes caused by human errors.

The Flashback technology is unique to Oracle and SQL Server has nothing comparable to it. In SQL Server the remedy for all the afore-mentioned human errors is a full database point-in-recovery. This form of recovery is an offline operation, is slow, and results in data loss. In contrast, Oracle's Flashback technology provides the user with a wide range of options for recovering from human errors without sacrificing any data or compromising system availability. This technology is very precise in its ability to repair damage, is very efficient, and most notably is fast and easy to use.

### Block Media Recovery

Oracle provides the option to perform recovery to the granularity level of data blocks. Using the block media recovery feature of Oracle, if only a single block is damaged then only that block needs to be recovered, while the rest of the file and the table containing the block remains online and accessible. This not only speeds up the recovery process but also increasing data availability. Here too, SQL Server is not able to compete with Oracle. It cannot recover data in single block units, thus requiring the entire file to be taken offline, restored, and recovered.

## INSTALLATION AND CONFIGURATION

### Instance Creation

Instance creation is among the most basic and important DBA functions. Oracle has a simple, easy-to-use GUI tool, Database Configuration Assistant (DBCA), for creating database instances. DBCA performs all the necessary steps for the user with minimal input and eliminates the need to plan in detail

the parameters and structure of the database. Furthermore, DBCA also makes recommendations about certain database settings based on user input and thus helps database administrators make correct decisions for optimal database configuration depending on whether it will be used for Data Warehousing, OLTP or mixed workload applications.

In SQL Server, on the other hand, every instance creation requires a complete new installation with all the binaries and scripts. This is in contrast to Oracle where multiple instances on the same machine share binaries and other relevant files. Consequently, any time multiple instances are needed on a single machine, a SQL Server DBA has to contend with considerable amount of wasted space, as each instance needs its own private binaries and scripts. Furthermore, during installation and instance creation SQL Server also offers no help in setting database parameters, thus adding to the workload of its DBAs.

### Database Duplication

Creating a standard definition for enterprise wide databases is a fairly common practice. Having all databases conform to a standard greatly simplifies their management by allowing the use of uniform administrative and monitoring procedures. DBA's also often want to duplicate databases for the purpose of setting up development and test environments. Oracle database templates, which allow for storing database definition in XML format, make both these tasks extremely simple. The template definition includes all characteristics of the database, such as initialization parameters, redo log settings, etc., and can be used to create identical databases on local or remote machines. There are two kinds of templates, those that contain only the structure of a database, and those that contain both the structure and the data. This functionality allows administrators to either create a new structurally identical database or clone a database along with its data. A template can be created by reverse engineering an existing database. This ability is extremely useful since it saves administrators a significant amount of time and effort in creating and testing scripts for duplicating a database.

SQL Server has no comparable feature to this. The only way to create a new SQL Server database is to use the Copy Database Wizard. There are many drawbacks to this method. First, in this method the database structure alone cannot be duplicated; instead both the data and the database have to be copied. Second, the database being copied must not have any active sessions on it, that is, the database must be offline to users[11]. If the SQL Server DBA wants to duplicate just the structure of the database, which is the more common case, the database and all its objects have to be recreated manually using scripts or the GUI tool. This is a cumbersome and error-prone method and, is certainly not comparable to the functionality provided by Oracle.

[11] SQL Server 2000 Books Online, Administering SQL Server.

### Installation Cloning

The Oracle Enterprise Manager 10g provides users with a convenient and flexible way to intelligently duplicate Oracle software installations (a.k.a. Oracle homes) across hosts. Guided by an intuitive wizard, users can designate an Oracle software home on a source system and select one or more destination hosts to which they want to clone that home. Via "multicasting", multiple new installations can be created in a single operation. The only requirement is that Enterprise Manager 10g agents are present on all systems involved.

The cloning of an Oracle home is performed in an intelligent manner, i.e. environment-dependent home properties such as the host name, IP address or listener settings are automatically adjusted during the cloning process. Furthermore, the Oracle Universal Installer (OUI) inventory that keeps track of all Oracle installations on a system is automatically updated as part of the cloning process. Cloning operations can be scheduled as Enterprise Manager jobs to be run during off-hours in order to minimize network load.

SQL Server does not support software cloning. Users have to manually install SQL Server software on all systems and then have to separately copy the database in the new location using the Database Copy wizard feature. The inability to clone software installations is yet another deficiency in SQL Server and it only serves to exacerbate the manageability challenge of its users.

## SOFTWARE MAINTENANCE

Administrators spend a significant portion of their time in software maintenance operations such as applying software patches. Being able to quickly identify and apply relevant patches on the system is key to maintaining system reliability and performance. Oracle's Enterprise Manager 10g provides administrators with powerful new patch management tools. The Critical Patch Advisor is a new tool that alerts users of critical patches issued by Oracle and immediately identifies those systems across the enterprise that may require a new patch. The administrator can then decide whether and when to apply the patch.

This functionality is possible because Oracle collects detailed configuration information about all databases being managed across the enterprise. Data collected includes information on:

Host hardware specs including number and clock speed of the CPUs, memory, hard disk and network information

- Operating system parameter settings, file system information and installed packages and patches
- Oracle software installed on the host including version and component information, patch sets and interim patches, as well as software configuration settings

This comprehensive system inventory is stored in the Enterprise Manager Repository and is the foundation for many of Oracles' configuration management solutions including software maintenance. By default, the configuration data is refreshed daily. In addition, users may refresh this data at any time with the click of a button.

The next step after being alerted to a patch is its application. This function is facilitated by another tool, the Patch Wizard. This tool searches, downloads and applies patches. Using the Patch Wizard, all patches applicable to a specific target database can be searched or, if desired, the administrator can query all the databases in the enterprise that are candidates for a specific patch. Once the necessary patch is located, Enterprise Manager 10g can be used to download and deploy it. Optionally, Enterprise Manager 10g can execute an end-user provided script to install the patch. Each of these steps allows for quicker application of patches across the customer's enterprise with minimal effort on behalf of the administrator.

SQL Server does not offer any solution in this area. It puts the burden on the administrator to determine what patches are needed and offers no help in their application. For proactive administrators this means that they either scan the Microsoft web site constantly for critical patches or call their support organization on a regular basis to find out whether there are some patches out there that they should apply. Others administrators would probably just wait for a problem to occur before they act to counter the damage and patch the problem. This not only occupies administrator time unnecessarily but potentially also impacts availability, reliability, and performance of a system.

## CROSS-PLATFORM PORTABILITY

One of the major and unique strengths of Oracle's architecture is its portability to different platforms. Oracle is available on practically all platforms and it is by far the most portable database than any of its competitors. This means that an Oracle database on Unix has the same code base as the one running on a Windows OS, ensuring that the management of the Oracle database on one platform is no different than on another. This makes it possible for an Oracle DBA to manage databases on any and all platforms without additional training. Another benefit of extensive portability is that an application can be designed independent of the platform with the sole objective of addressing the business requirements in the best possible manner.

The same cannot be said about SQL Server as it runs only on the Windows platform. As a result, any future growth of a business is limited to the Windows platform. Outgrowing Windows means a huge investment of time and money to upgrade all the hardware and software systems to Unix, migrate the data to a more scalable database, rewrite all the applications to run against the new database, and hire or retrain DBAs. With Oracle, you never need to worry about outgrowing your hardware, since Oracle data and applications are fully

compatible and portable across all major hardware and operating systems platforms.

## SCALABILITY

One of Oracle's greatest assets is its scalability. Oracle databases can scale seamlessly from hundreds to thousands of users. In addition, Oracle Real Applications Cluster (RAC) offers the most transparent cluster database solution from a deployment and management perspective. Nodes can be added to a RAC database on the fly with no impact on existing nodes. Also, since all nodes in a RAC environment access a single database, addition or removal of nodes does not generate any additional tasks for a DBA. The most unique aspect of RAC, however, is that unlike other database products (including SQL Server) where an application has to be rewritten for a clustered environment, moving from a single node to a RAC environment requires absolutely no application change thanks to its patented Cache Fusion technology that was introduced in Oracle9*i*.[12]

Microsoft's recommended implementation of a scalable model for SQL Server is either in a distributed database environment, or in a federated database architecture, where many independent databases are connected together with no common data dictionary while a single application accesses data across all these databases. Imagine a simple task such as adding a node in SQL Server's federated architecture environment. A DBA or system administrator will have to do all of the following:

- Add hardware
- Configure new instance (set instance-specific parameters, etc.)
- Create new database
- Disconnect all users
- Unload data from existing tables
- Redefine partitioned tables and indexes
- Redefine triggers on partitioned or replicated tables
- Redefine distributed partitioned views
- Reload the data to spread it across a larger number of partitions
- Reconnect all users.

[12] For more details on RAC and Cache Fusion, refer to the white paper titled "Oracle Real Application Clusters: Cache Fusion Delivers Scalability", which can be found at http://otn.oracle.com/products/oracle9i/pdf/cache_fusion_rel2.pdf.

This represents a highly complex and elaborate set of management tasks that involve unloading and reloading of data, redefining of views, modification of triggers, etc. Furthermore, when these tasks are being performed all the databases have to be unavailable for the entire duration of the operation. Compare this to Oracle Real Application Clusters. The administrator needs to do the following two tasks only when adding a node:

- Add hardware
- Configure new instance (set instance-specific parameters, etc.)

The DBA did not have to unload and reload data, change any schema definitions, nor was the system taken offline at all. Thus, when compared to Oracle, SQL Server's scalability solution is a manageability ordeal even for basic tasks.

"Microsoft sells ease of use and hesitates to point out that a SQL Server environment is often just as complex — and expensive — to manage as its UNIX counterparts".

Brian Moran
Editor, SQL Server Magazine

Another point to note about SQL Server is that the complexity of management goes up dramatically when there are tens of databases to manage, administer, maintain, backup and upgrade. As an example, to achieve a high TPC-C benchmark Microsoft had to use 32 different databases. Thus, Microsoft's approach is to distribute rather than contain complexity. Managing so many databases is quite an unappealing scenario for DBAs when compared to implementing and administering a single, more scalable database like Oracle. Brian Moran, SQL Server Magazine editor says it all: "Microsoft sells ease of use and hesitates to point out that a SQL Server environment is often just as complex — and expensive — to manage as its UNIX counterparts."[13]

Oracle's scalability advantage is not restricted to the multi-node RAC model only. Being the most portable database in the market, it runs on all major OS platforms including large SMP boxes. SQL Server, on the other hand, is restricted to the Windows operating system. This means that SQL Server's scalability within a box is restricted to the smaller Windows machines, while Oracle has no such limitation as it runs on all major platforms. It can easily scale on large 64-cpu SMP boxes without any problem. From a manageability perspective this is a great advantage because as the applications grow over time, an Oracle DBA can address the growth by just adding more CPUs to the machine, whereas in the case of SQL Server an application can quickly outgrow the smaller Windows machines. This leaves the DBA with no option but to add more machines and then hope that the application can be redesigned for a federated architecture. Even when possible, this is an extremely painful and time-consuming task.

To overcome the lack of scalability of SQL Server 2000, a DBA has to administer an environment that is complex and virtually unmanageable. Simple tasks like adding a new node can involve very elaborate redistribution of application data requiring deep understanding of the application logic and the

[13] SQL Server Magazine UPDATE News Editor, Jan. 2001.

database layout. These are just some of the manageability issues that a SQL Server DBA has to contend with on a regular basis, whereas Oracle's scalable architecture makes all these non-issues for its DBA.

## CONCLUSION

"The top manageability challenges that we face today are being more proactive in managing systems and working with development teams. With Oracle 10g, DBAs spend less time on time consuming tasks. Getting that extra time saving is the most important value we get from Oracle 10g."

Rajesh Sheth
Manager DB Engineering & Operations
Amazon.com

Oracle Database 10g is not only rich in functionality but, thanks to its bold initiative to enhance manageability, it is also easier to manage than SQL Server 2000. With management costs of software quickly outpacing acquisitions costs, ease-of-use or manageability has become a most crucial factor affecting purchase decisions. Oracle is the only solution in the market that offers the best of both worlds. SQL Server 2000 lacks many of the basic management functionality critical for effective database management. Consequently, SQL Server DBAs are often left with no choice but to undertake cumbersome workarounds that are unintuitive, complex, and, quite frequently, a management nightmare. Oracle's comprehensive and structured approach to manageability will only further its lead over its competitors in the future releases.

**Technical Comparison of Oracle Database 10*g* and SQL Server 2000: Focus on Manageability**
**August 2004**
**Author: Mughees A. Minhas, Sushil Kumar**

**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**