# Technical Comparison of Oracle9*i* Database vs. IBM DB2 UDB: Focus on Manageability

*An Oracle White Paper*
*May 2002*

ORACLE

# Technical Comparison of Oracle9*i* Database vs. IBM DB2 UDB: Focus on Manageability

# Technical Comparison of Oracle9*i* Database vs. IBM DB2 UDB: Focus on Manageability

## EXECUTIVE OVERVIEW

The Oracle database has always been known as the industry leader in terms of performance and reliability, while at the same time providing extremely rich functionality. Most successful businesses today use the Oracle database to power their mission critical applications. However, as the business environment turns increasingly competitive, companies are looking for solutions that can minimize their operational cost without compromising on the Quality of Service goals. Rising to this challenge, the Oracle9*i* database not only allows enterprises to build, deploy and manage critical business applications but also helps them maintain the highest level of service without increasing the information technology (IT) budget. In spite of misleading claims made by IBM, DB2 continues to remain far behind Oracle in terms of key functionality and ease of use. It lacks essential tools, does not have any significant self-tuning capability and, is far more expensive to own despite its lower licensing cost. This fact is conclusively proved by the database management cost of ownership study conducted by Rauch Associates, which found the cost of managing an Oracle9i Database 49% lower than that of DB2[1]. Oracle, therefore, continues to remain years ahead of all its competitors, including IBM, in delivering the most comprehensive and cost effective enterprise data management solution.

## INTRODUCTION

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 for Oracle9*i* and beyond. The Oracle9*i* Database, therefore, automates a number of routine administrative tasks, reduces

[1] http://otn.oracle.com/products/manageability/content.html

the complexity of administration and, includes a number of self-tuning features to deliver out-of-the box optimal performance.

The manageability enhancements introduced in Oracle9*i* are not just a bunch of odd features; instead they are the results of a structured methodology adopted by Oracle to deliver a comprehensive and holistic management solution. Consequently, Oracle9*i* has first sought to ***eliminate*** complexity, whenever possible, thereby completely removing some of the well-known concepts from a database administrators handbook. As an example, with automatic undo management, an administrator no longer needs to be aware of how the undo or rollback data is managed since the database performs this transparently. The administrative functions that could not be completely eliminated in the Oracle9*i* Release time frame have been ***automated*** in order to reduce the degree of manual intervention required to keep the database operational. As a result, Oracle9*i* relieves administrators of complex tasks such as tuning SQL execution memory and managing standby databases. But even with the highest degree of automation, there will always be certain tasks that must be performed manually such as adding or removing system resources. Such tasks have been significantly ***simplified*** with the objective to minimize the time an administrator spends in performing them. So, for adding additional disk space to the database, administrators just need to specify how much space should be added while the Oracle Databases automatically creates and manages the underlying operating system files.

All such automation and simplification is not possible unless the product has some built-in intelligence to learn about the workload and database usage pattern and, then use this learning to either automatically perform a given activity or help DBAs perform those tasks in a simple manner. The numerous advisories introduced in Oracle9*i*, discussed in detail later in the document, are examples of the features that make the Oracle Database more intelligent than any other competing product. Finally, the comprehensive enterprise management solution offered by Oracle Enterprise Manager (EM) framework makes it possible to manage all components of the IT Stack without having to buy and integrate a number of third party software which were never designed to work together in the first place!

The Oracle9*i* Database, therefore, is the only product that can satisfy the stringent scalability, availability and manageability requirements of modern eBusinesses. However, belying all these facts, IBM has been trying very hard lately to propagate the myth of DB2 being a more manageable solution vis-à-vis Oracle. The rest of this document presents a detailed technical comparison of the two products to clearly demonstrate Oracle's superiority in the manageability arena. Even though, IBM promotes DB2 as a single product across different platforms, there are actually 3 separate codes bases (OS/390, AS/400 and Windows/OS2/Unix) with distinctly different capabilities. The scope of the comparison in this document is limited to the version of DB2 that comes closest to being called "open" i.e. DB2 UDB for Windows, OS2 and Unix.

## MANAGEMENT TOOLS

Graphic management tools are a standard part of any enterprise software today. DB2's GUI administrative tool is called "Control Center" while Oracle Enterprise Manager is Oracle's single, integrated solution for administering and monitoring global e-Business enterprises. It is important to emphasize here that Oracle Enterprise Manager is much more than just another GUI database administrative tool; it is a complete enterprise IT infrastructure management framework. Unlike DB2 control center, which lets administrators perform certain database administrative task interactively from a GUI tool, EM product suite enables the centralized management of all components of an eBusiness infrastructure right from application through to the middle-tier and database down to OS and network levels. In essence, EM lets administrators manage the end-user experience, not just the individual system components. This itself is a big differentiator for Oracle since customers otherwise need to buy a plethora of third party tools and integrate them to create a solution that EM provides right out of the box.

Looking from a database-centric point of view, Oracle Enterprise Manager significantly simplifies the day-to-day database administration by combining simplicity in use with a rich feature set to offer a complete solution for managing any size environments regardless of platform. In addition, it automates crucial management tasks and extends the reach of administrator beyond his/her desktop by providing capabilities such as:

- Production-ready monitoring and proactive alerts to anticipate problems and fix them automatically using "fix-it" jobs
- Rich tuning advisories and guided diagnostics to optimize the performance and lower the DBA learning curve
- Comprehensive, automated reports that can be published to the web to measure and manage service level agreements
- Lightweight browser interface for access to management tools from anywhere

**"Enterprise Manager provides a centralized management framework for our distributed open systems. Oracle9i and the applications Oracle has developed to meet BT's requirements will put us a long way toward our goal of having a single management tool for all DBA's."** - **British Telecom**

A large number of Oracle customers such as British Telecom, Boeing, Telstra, and Shell have standardized on EM 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.

Though DB2 Control Center is a database administration tool, it lacks support for some of the very basic DBA functions. Contrary to claims made by IBM, Control Center does not have any proactive monitoring and notification capability to alert administrators of problems any time they occur. Customers must buy a management framework to get 24 hour monitoring with paging and email. Also, Control Center gives the DB2 DBA no guidance whatsoever on how to solve a problem. EM's database health overview screen, on the other hand, instantly directs

administrators to the root cause of the problem and guides them to the solution. It is no wonder then that the eWEEK magazine, during their recent comparative analysis of database products, concluded, "Oracle9*i* had the best run time performance monitoring tool of all databases"[2]. EM also provides instant analysis of database functioning at the push of a button. An Oracle DBA can instantly see what the "Top" resource hogging sessions or SQL statements are in the database, quickly find out the optimal memory configuration and, get instant advice on indexes to create to increase query performance. DB2 Control Center does not provide any of this productivity-boosting functionality. A DB2 DBA has to spend hours handcrafting scripts, using tedious trial-and-error methods to optimize performance, and poring over trace files to get to the root of a problem.

## SCHEMA MANAGEMENT

### Table and Index Reorganization

**"In previous years, due to 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**

In an ideal world, a DBA should be able to perform all administrative actions online, i.e., all business transactions can continue while the operation takes place. Oracle not only performs more operations online than DB2[3] but also makes their execution significantly more manageable. Virtually every reorganization operation can be performed online in Oracle, be it creating a new index, coalescing or moving an existing index or, making any kind of changes to a table including adding and dropping columns. Of these, DB2 can only coalesce an index online. All other operations in DB2 require transactions stopped on the database and all locks released[4]. For more information on online maintenance operations allowed in Oracle9*i*, refer to Oracle white paper "Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on High Availability" at http://otn.oracle.com/deploy/availability/pdf/CWP_9IVSDB2_HA.PDF.

Reorganizing tables or indexes in Oracle is also far less complex than in IBM DB2. Oracle administrators can analyze the database to identify objects that need to be reorganized and complete the reorganization from a single tool. DB2 requires that the use at least 2 utilities, REORCHK and REORG. Since reorganizations can be resource-intensive, they are often scheduled during non-peak hours. Using EM tools, reorganizations can easily be scheduled to run unattended during the maintenance window. Furthermore, EM automatically alerts the administrator when the reorganization starts, fails or succeeds. In DB2, any reorganization job is completely manual process requiring development of custom scripts and constant administrator supervision.

Oracle's reorganization process also has sophisticated error-handling capabilities. If an error occurs during reorganization, the administrator can either undo the

---

[2] eWEEK, February 25 2002 issue, Article "Server Databases Clash"
[3] For details on Oracle and DB2 online operations, refer to Technical Comparison of Oracle vs. IBM DB2 UDB: Focus on High Availability, http://otn.oracle.com/products/oracle9i/pdf/CWP_9IVSDB2_HA.PDF
[4] DB2 Command Reference, REORG command, page 470.

operation or fix the problem and resume. When an error occurs during reorganization in DB2, changes will usually have to be rolled back or the database recovered.

### 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 Oracle9*i* Database offers several partitioning options designed to fulfill different requirements[5]:

- 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.
- 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.

DB2 offers very limited partitioning options. The table below summarizes the differences between Oracle and DB2 with regard to the partitioning options that each product supports:

---

[5] For more information about Oracle*9i*'s partitioning options, see Oracle*9i* Partitioning, Hermann Baer, Technical White paper, Oracle Open World 2001 Berlin.

| Feature | Oracle | DB2 |
|---|---|---|
| Range partitioning | Yes | – |
| List partitioning | Yes | – |
| Hash partitioning | Yes | Yes |
| Composite partitioning | Yes | – |
| Local index | Yes | Yes |
| Global partitioned index | Yes | – |
| Global non-partitioned index | Yes | – |

*Table 2: Partitioning options.*

DB2 only supports the hash partitioning method, which has considerable limitations and weaknesses when compared to Oracle's partitioning capabilities. By only supporting hash partitioning, DB2 does not allow for 'rolling window' support – a great drawback in manageability. A 'Rolling window' allows the data for a defined period, lets say a week or a month, to be kept online by continuously replacing the oldest data with the most recent one. DB2's hash partitioning scheme requires data in all partitions to be redistributed whenever new data is loaded, therefore increasing the load time and decreasing data availability as the table is locked during the data redistribution process. Similarly, when the old data is archived or deleted, all partitions need to be touched. This may interfere with regular insert operations and cause space fragmentation.

Another weakness of DB2 is that it requires equi-partitioning between tables and indexes. This means that global indexes, partitioned or non-partitioned, cannot be created. This is a major problem in OLTP environments where global indexes are commonly used to offer efficient access to any individual record. With DB2, application designers have no flexibility when defining their indexing strategy in partitioned configurations.

All these limitations in DB2 partitioning options make the task of managing large volume of data quite complicated. Oracle offers far more options in this area that not only enhance manageability but also improve performance.

### Schema changes

Administrators frequently need to react quickly to changing business needs. Applications and their databases may need to be altered to improve performance or reliability. At the same time, businesses frequently have close to zero tolerance for downtime or application unavailability. In this situation, DBA's must be able to implement complex schema changes efficiently and without error to support the evolution of application schemas.

When performing a schema change such as synchronizing the production and Quality Assurance (QA) or test schemas, a common task is to check for dependencies on objects that are being modified. In Oracle, the dependencies on the table are automatically checked and accounted for when a change is implemented. After the change, all the dependencies, data and privileges are restored to the table, so that any applications using that schema will continue to function without error. In addition, Oracle has the pre-change impact report that allows the administrator to foresee and correct any potential problems ahead of time[6]. This significantly reduces time spent in trial-and-error efforts making sure changes are implemented properly.

In DB2 the task is considerably more complex. The administrator must first perform painstaking research to ensure that all dependencies on the object to be changed are found. Then the administrator must use scripts or command line utilities to recreate the table with the desired changes, reload the data, recreate the dependent items, and then, test to make sure that the changes were made correctly. As with all hand scripting, the administrator must also invest additional time in testing his/her scripts. This makes the task very time-consuming and highly dependent on the skill and experience of the DBA.

Because DB2 requires that the administrator create a script to implement the changes, complex capabilities such as error handling or undoing changes must be coded into the script or performed manually. This increases the complexity of the task and the time required, since an administrator will likely have to monitor the process carefully to detect and minimize errors. To make matters worse, creation of the schema change script would probably not be a one-time activity since the script only implements the particular change being made to the schema – any future changes would require creation of another script.

Hence, for schema changes requiring the administrator to go beyond the standard ALTER statement, Oracle provides a significantly simpler, more flexible and less error prone implementation technique than DB2.

Rebuilding the program units stored in the database, such as stored procedures and triggers, is yet another common schema management task. Such changes often cannot be made until no users are accessing the object that is being rebuilt. Administrators, therefore, need to force users off the database to ensure that the rebuild operation is successfully completed. In DB2, the only way to do so is to shut the system down leading to application unavailability. Oracle allows administrators a more convenient option of quiescing the database, where users operations are temporarily stopped to allow for maintenance activities to be performed. As opposed to visible disruption caused by database shutdown, quiescing a database is completely transparent to the end users who can continue working as soon as the database is unquiesced.

---

[6] These schema change management functionalities are available in Change Management Pack, a part of Oracle Enterprise Manager product suite

## SPACE MANAGEMENT

### Row size limit

DB2 imposes a limit on the row size depending on the page size of the table space. A row cannot span multiple pages and so its size must be a little smaller – taking page overhead into account – than the page size. The table below summarizes the maximum row sizes for the different page sizes in DB2.

| Page Size | Maximum Row Size |
|---|---|
| 4K | 4,005 bytes |
| 8K | 8,101 bytes |
| 16K | 16,239 bytes |
| 32K | 32,677 bytes |

*Table 1: Row size limits for different page sizes in DB2*[7]

Since the maximum page size in DB2 is 32k, this means that the row size under no circumstances can be longer than 32,677 bytes. In order to overcome this limitation, a DBA would have to vertically partition the table into multiple smaller tables. Furthermore, the row size restriction could lead to a sub-optimal database design that can impact overall system performance adversely. For example, let us take the case a table with an average row size of 500 bytes that is mostly accessed randomly using an index, but has a few large rows of size 30k. In this case, the DBA would be forced to create a table space with a page size and buffer pool of 32k just to accommodate large rows in spite of the fact a smaller page size would have produced a better performance. Thus a sub-optimal design was forced upon the DBA just because of yet another artificial limitation imposed by DB2.

Oracle allows a row to span across multiple pages or, in Oracle's terminology, blocks thereby completely avoiding all the complications outlined above.

### Monitoring space and usage

All DBA's monitor space utilization of their databases to ensure that the system does not come to a halt due to lack of space. Using the EM framework, Oracle can proactively monitor space utilization and alert DBAs of any impending problems. In addition to notifying administrators about a tablespace full condition, Oracle also helps them plan for the future by predicting when a tablespace will run out of space based on the past growth trend[8]. Such proactive monitoring with intelligence to predict future problems allows the DBA to easily avoid space problems in Oracle. In DB2, administrators have to write manual scripts to analyze data dictionary

[7] http://webdocs.caspur.it/ibm/web/udb-6.1/db2q0/perform.htm

[8] This functionality is available in Oracle Capacity Planner, a part of the OEM product suite.

information in order to identify space-related issues. They would then have to schedule the scripts to run periodically and use a third-party tool to send notifications via email or page. Monitoring something as important and basic as space in DB2, therefore, can be a complex exercise requiring considerable investment of time and money.

### 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 the server 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 the administrator is notified of the problem and given a chance to fix it. 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 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, they can all be run in the "Resumable" mode.

This capability is unique to Oracle with no parallel in DB2. It saves Oracle DBAs enormous time that their DB2 counterparts spend in monitoring and re-executing failed long running operations In DB2, if an operation runs into an out-of-space situation, the entire operation may have to be repeated after the space issue has been addressed. This not only results in wastage of time but it can also hamper normal database performance if the out-of-space situation forces the DBA to re-run the batch job during normal workload hours.

Oracle's Resumable space allocation feature avoids all the pitfalls that a DB2 user would encounter by offering the user a graceful way to address out-of-space situations.

## DATA LOADING AND ARCHIVING

### Data Movement Utilities

Oracle provides the most complete and powerful set of tools and utilities to facilitate bulk movement and archiving of data. Oracle's EXPORT/IMPORT utilities can be used to offload data to operating system files, backup an Oracle Database and move data across databases. DB2 also offers similar tools but they come with some significant restrictions that severely limit their usefulness in a real world environment.

First of all, DB2 export allows for exporting data from only one table at a time. So, if a schema has, lets say, 20 tables, one needs to run 20 export jobs, one for each table. Besides being more time and labor intensive, this can also lead to inconsistencies in the exported data since tables with dependencies may have been

exported at different times. So, if the schema contains two table ORDERS and ORDER_ITEMS, the exported data for the ORDER_ITEMS may contain entries for orders not present in the export file for ORDERS table since the ORDER_ITEMS table may have been exported after ORDERS table. Such inconsistencies may render the exported data completely useless since it cannot be used to recreate a consistent snapshot of data. Oracle, on the other hand, not only allows administrators to export data at a schema, tablespace and database level but can also ensure that the exported data for various tables is consistent to the point in time when the export job was started, irrespective of the time when a given table was exported. Therefore, in the context of the above example, the administrator needs to execute only one job to export the complete schema and does not have to worry about any inconsistencies in the exported data.

Secondly, since the DB2 export file does not extract certain table and schema attributes, it can never be used to logically recreate a schema or a database. According to "DB2 Data Movement Guide and Reference"[9], DB2 EXPORT does not support tables with typed columns and does not retain attributes such as referential integrity constraints, check constraints, physical space allocation settings, column default values, foreign key definitions and triggers. It also does not support any non-table database objects such as views, stored procedures, etc. Therefore, unlike Oracle Export/Import utility that is frequently used to perform a logical backup and restore of a schema or a database, the DB2 tools are designed only for moving data in and out of a table. But even this may not be possible at times since DB2 import cannot be used to re-create tables with LOB columns defined to be 1 GB or larger[10].

Oracle SQL Loader is another powerful tool to load data from external files into Oracle database tables. Unlike DB2 Load, SQL Loader allows for loading data into multiple tables during the same load session, makes it possible to manipulate data using SQL functions and, leaves table data accessible after the load process completes. This is not true for DB2 unless the administrator has backed up the tablespaces and enabled the integrity constraints[11].

But going beyond simple data loading, Oracle9*i* provides a comprehensive set of server functionality to address the typical requirements of ETL (Extraction, Transformation, Loading) processes. Using Oracle9*i*'s external table feature, data in external sources such as flat files, can be exposed within the database just like a regular database table. These "External Tables" can be accessed via SQL, so that external data can be queried directly and loaded into the database in parallel using the full power of SQL, PL/SQL, and Java. External tables significantly simplify the

---

[9] Chapter 2, Page 32,33,67

[10] DB2 Data Movement Guide and Reference, Chapter 1, Page 4

[11] Fast Path to DB2 for Experienced Relational DBAs CBT, Load Considerations, Slide 9

ETL process by combining data-transformations and loading into a single step; some thing that will require several manual steps with DB2. With DB2, the external data first needs to be loaded into a "staging' table before any SQL based transformation and processing can be applied. The net result, more work for the DBA and a longer ETL cycle!

### Resumable/Restartable Operations

Since any operation involving bulk movement of data could be highly time consuming, the ability to restart a failed operation is a very important usability yardstick. SQL Loader, therefore, like DB2 Load, can restart an aborted operation by skipping processing of records already loaded prior to the failure. Though this capability prevents processing of data already loaded, this still requires a manual re-execution. Moreover, sometimes scanning the complete input file to identify the point from where the load is to be restarted could take an unacceptably long time due to extremely large input file size. Under such circumstances, it is desirable that the operation should not be aborted at all even when a failure condition is encountered so that the overhead of re-executing the operation is avoided. Thanks to Oracle9*i*'s Resumable Space Allocation feature, any transaction in Oracle can be prevented from failing due to an "out-of-space" situation. It is important to note here that the Resumable Space Allocation is a much more powerful restartable mechanism since it can be used to make any operation restartable, not just Import or Loader sessions.

### Transportable Tablespaces

Moving or copying data across databases is another very commonly performed database administrative task. In DB2, this usually involves first offloading data from the source database into external files and then, using either import or load utility to insert it into the destination database. Since this process incurs all the usual overhead associates with SQL processing and running the load utility, it could take a long time for this task to complete when moving a large volume of data. With Oracle, on the other hand, transportable tablespace capability could be used which is orders of magnitude faster since it only involves copying datafiles and integrating the tablespace metadata. Also, when a tablespaces is transported, the index data can also be moved thereby eliminating the need to rebuild the indexes after importing or loading the table data.

## BACKUP & RECOVERY

### Bounded Recovery Time

Oracle has worked hard over the last several releases to develop a unique architecture for recovery that is both very powerful and easy to use. Oracle is the only database product that can put a bound on the total crash recovery time regardless of the system workload. The administrator merely needs to specify the

desired crash recovery time target[12] and Oracle dynamically adjusts the checkpoint rate to ensure that recovery completes within the specified target. The closest DB2 comes to meeting this requirement is to allow administrators to manually adjust the soft checkpoint interval using the SOFTMAX database configuration parameter[13]. However, since the effect of any changes in the value of this parameter on the actual recovery time cannot be ascertained with any precision, limiting recovery time in DB2 is a trial-and-error game. This is totally unacceptable in modern, highly demanding IT environments that provide little room for down time and are governed by stringent service level agreements. Oracle not only lets administrators limit the recovery time in a deterministic manner but also enables them to make an informed trade off between desired recovery time and acceptable performance. Oracle9*i*'s Recovery Cost Estimator further helps administrators optimally set the target recovery time by predicting the number of checkpoint IO for various target settings ranging from 10% to 200% of the current value.

### Automatic Log Archiving

In order to recover from media failures, both Oracle and DB2 require the redo logs to be archived. In Oracle, automated archiving is enabled by default and all that an administrator needs to do is put the database in the archivelog mode. In contrast in DB2, one not only needs to set the LOGRETAIN parameter but also write an additional piece of external code called "user exits" which can be invoked by DB2[14]. In other words, DB2 by itself, has no capability to archive log files and relies on the administrator to extend its code using the "user exits" to fulfill this extremely basic functionality.

### Backup Throttling

Extending the discussion on "Online" operations, a truly online backup should not only ensure that the database stay up during the backups but also limit its performance impact so that users can continue to use the database. This is extremely important in today's around-the-clock economy since users do not care to differentiate between an unavailable and a poorly performing system. Oracle Recovery Manager (RMAN), therefore, 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 DB2 currently. In fact, this is one of the features planned for the DB2 version 8.0[15] whereas Oracle introduced this capability several years ago.

---

[12] Using FAST_START_MTTR_TARGET parameter
[13] DB2 Data Recovery and High Availability Guide and Reference, Chapter 8, Page 286
[14] Fast Path to DB2 UDB for Experienced Relational DBAs, Logging Concepts, Slide 15
[15] DB2 Version 8 Preview Presentation by Jim Lawrie, Slide 47

### Centralized Recovery Catalog

Both Oracle and DB2 store backup & recovery historical information in the database. While Oracle uses control files for this purpose, DB2 creates a separate file called "Recovery History File" to keep a track of backup & recovery related operations. However, from an enterprise wide database management perspective, it may be desirable to consolidate this information centrally in order to create a single point of management. Oracle, therefore, allows the backup & recovery catalog information to be stored in a central, independent repository. Besides making it easy to query the backup information for all databases in the enterprise, this also increases the resilience of catalog information by separating it from the individual databases. Such a facility does not exist in DB2.

### Incremental Backup

Oracle provides a whole gamut of unique technologies to meet the exacting requirement of efficient backup and recovery methods for Very Large Databases (VLDBs). Incremental backups, where only the data blocks that have changed since the last backup are copied, provide an excellent mechanism to limit the amount of back up data generated and have been supported in Oracle since version 8.0. DB2 has just begun offering incremental backups in release 7.2 but still does not allow LOB data to be backed up incrementally[16]. This, ironically, is a very severe limitation since incremental backups make most sense for LOB columns, which often store large multimedia data that rarely changes. Similarly, Oracle's block-level media recovery feature allows only damaged blocks to be recovered in case of a failure, while the rest of the file, and thus the table containing the damaged blocks, remains online and accessible. DB2 cannot recover at the block level, thus requiring the entire file to be taken offline, restored and recovered.

### Resumable Backup & Restore

Another Oracle feature that saves administrators enormous time is restartable backup and restore. In Oracle9*i*, Recovery Manager (RMAN) backs up only those datafiles that have not already been backed up. Similarly during a restore operation, RMAN scans the datafile headers and determine if it is necessary to restore the file to complete a database recovery. This feature improves the performance of backup/restore operations by eliminating unnecessary activity. Not having any such capability in DB2 means that any failed backup and restore operations must be re-executed from the very beginning, resulting in lost time and, possibly, prolonged system outage. In fact, DB2 does not even allow simultaneous backup and restore operations to be performed on different tablespaces in the database[17]. So, while any tablespace in a database is being restored, no part of the database can be backed up. Needless to say, there is no such restriction in Oracle.

---

[16] DB2 High Availability and Recovery Guide and Reference, Chapter 1, Page 26
[17] DB2 High Availability and Recovery Guide and Reference, Chapter 2, Page 103

### Human Error Recovery

Numerous availability studies have highlighted the human error as one of the most prominent causes of application outage. While it is possible to develop 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. Oracle's Flashback Query enables administrators and users alike to view data at a point-in-time in the past and use it to reconstruct the lost data. This is a very significant benefit. To illustrate this further, let us take the example of a banking application where a set of bank accounts was accidentally deleted from the ACCOUNTS table at 11 AM. The inconsistencies caused by this mistake were later discovered at 2 PM. Using Flashback Query, the database administrator could execute a simple SQL command to recover from such an error. The following query retrieves the deleted accounts by comparing the current data in the ACCOUNTS table as against what it was just prior to 11 AM and, inserts them back into the table.

```
INSERT INTO ACCOUNTS
   (SELECT * FROM ACCOUNTS AS OF TIMESTAMP
    TO_TIMESTAMP('13-MAR-02 10:59:58','DD-MON-YY
    HH24:MI:SS')
    MINUS
    SELECT * FROM ACCOUNTS);
```

If the bank were using DB2, they would have to rollback the database just prior to 11 AM, when the error occurred, in order to "recover" the deleted data thereby loosing all subsequently committed transactions. Since rolling back the database requires restoring a backup copy and rolling it forward till the desired point in time, it will also lead to the system being unavailable for the duration of recovery. Clearly, this is not the solution the bank is looking for since it forces the bank to compromise system availability and data integrity. Flashback Query, on the other hand, made it possible for the bank to correct this mistake without sacrificing any data and with absolutely no affect on the normal operations. This enabled the bank to avoid any revenue loss due to "lost" data and application outage.

Oracle provides another extremely powerful tool called "Log Miner" to find and correct unwanted changes. Using Log Miner, administrators can scan the redo log files to precisely determine when and how the incorrect change was made and, who was responsible for this change in case it was done with malicious intent. The tool can display the change history for a given set of data, if desired, and provides the SQL statements needed to "undo" the erroneous transaction. Once again, this is much easier than performing a point in time recovery and is completely transparent to the normal functioning of the database. DB2 does not have any log-mining tool and hence, leaves administrators only with the dreadful option of recovering the database any time a logical corruption takes place.

IBM often touts its "dropped table recovery" capability, which is nothing more than creating another database, rolling it forward to the point in time before the

table was dropped and, then exporting the table and importing it into the original database. Oracle, undoubtedly, supports the same operation but unlike DB2, where the dropped table recovery facility cannot be used for table containing LOB and LONG columns[18], there are not any restrictions in Oracle on the kind of tables that can be recovered using point-in-time tablespace recovery facility.

It is therefore clearly evident that Oracle's comprehensive backup & recovery capabilities extend beyond the basic functionality offered by DB2. Only Oracle makes it possible to recover from any kind of failures easily and with minimal downtime.

## PERFORMANCE MANAGEMENT

### Managing Locking Conflicts

One of the biggest strengths of Oracle from the performance management point of view comes from its fundamental architecture. Deviating from the industry norm of lock-based concurrency control, 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 DB2 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 as well as to wrap around the log files without waiting for transactions to commit. IBM admits this in its own documentation. According to "Oracle to DB2 Migration - Comparison White Paper" published by IBM, *"As a possible result of different currency controls between Oracle and DB2 UDB, an application ported directly from Oracle to DB2 UDB may experience deadlocks that it did not have previously. As DB2 UDB requires shared locks for readers, updates may be blocked where that was not the case in Oracle"*[19]. Consequently, Oracle DBAs need not even think about some of the most time consuming tasks that their DB2 counterparts have to perform on a daily basis i.e. resolving deadlocks and monitoring uncommitted transactions.

### Application/SQL Tuning

#### Identifying and Tuning SQL statements

Deficiencies in application design are the most common cause of performance problems. Database administrators, therefore, spend a significant amount of their time identifying and tuning resource intensive SQL statements. Here again, Oracle makes the life simpler for administrators by automatically compiling the information regarding resources consumed by each active operation. All that the

[18] DB2 High Availability and Recovery Guide and Reference, Chapter 4, Page 137
[19] Oracle to DB2 Migration Comparison White Paper, Page 58

administrator needs do in order to identify problem SQL statements is to either launch the SQL Analyze GUI tool, a part of the EM product suite, or execute a simple query against the database[20]. To do the same in DB2, one must go through a series of time consuming steps of first setting up an event monitor to enable tracing and then analyzing the contents of trace files. This is a very significant manageability issue for DB2 administrators for a couple of reasons. First of all, monitoring "rogue" SQL statement is a common database administrative task and hence, spending anything more than a few minutes to get this information is completely unacceptable. Secondly, for a busy system, the generated trace file can easily grow to a very large size making it almost impossible to extract meaningful information from it.

Oracle, on the other hand, makes this task as simple as it can possibly get. Once the high impact SQL statements have been identified, Oracle offers a number of tools such SQL Analyze and Index Tuning Wizard to aid administrators in tuning the statement. The SQL Analyze tool can automatically check the SQL statements for basic violations such as, inadvertent disabling of an index by the use of certain operations, or using unnecessary filter and sort operations. If any violation is detected, it can correct the problem by rewriting the SQL and also provide an estimate of the projected performance improvement based on the reduction in optimizer cost of the rewritten SQL. Similarly, the Index Tuning Wizard guides administrators step-by-step and helps them identify the optimal index requirements for a given workload. In the absence of any such tools in DB2, administrators are left on their own, forcing them to spend countless hours manually tuning the problem queries.

#### Query Optimizer

All databases use a query optimizer to determine the best execution plan for SQL statements. The query optimizer takes crucial decisions which have a tremendous impact on the performance of a query such as, whether to use an index or not, which join techniques to use if the query involves joining multiple table, etc.. Oracle's Cost Based Optimizer has evolved over the years into the industry's most sophisticated, mature, and thoroughly tested optimizer. IBM continually claims - without proof of any kind - to have the most advanced optimizer, but a look under the covers shows something entirely different. The DB2 optimizer lacks many important optimization techniques that Oracle has had for years such as count optimizations for bitmap indexes, join-back elimination, etc. It is therefore no surprise that DB2 DBAs rate query optimization (or tuning) as their top-most task in terms of time, frequency and difficulty, ahead of day-to-day administrative tasks

[20] The dynamic performance view V$SQLAREA provides the resource consumption statistics for SQL statements in memory.

such as database performance tuning, data loading, problem diagnosis, backup, application support, etc[21].

The Oracle optimizer, in addition to supporting more functionality and optimizations as already indicated, is thoroughly tried and tested. One major test bed for the Oracle optimizer is Oracle Applications, which consists of 279,000 queries. This application generates very complex queries with very heavy use of views and many multiple-table joins, all of which are successfully optimized by the Oracle optimizer. PeopleSoft ERP application has over 100,000 SQL statements and does not use hints, relying solely on the cost-based optimizer. There is not even a remotely equivalent testing ground for the DB2 optimizer. Oracle's cost-based optimizer is also used by other packaged applications such as Siebel, and SAP. Since the Oracle database is the platform for the vast majority of the customers using these applications, this demonstrates the success of the Oracle optimizer in a huge number of real application settings. For more information on Oracle's leadership in query optimization, refer to the white paper "Query Optimization in Oracle9i" (http://otn.oracle.com/products/bi/pdf/o9i_optimization_twp.pdf).

#### Hints

Query optimization is a continuously evolving technology. No optimizer can claim to be perfect and DB2 optimizer is certainly not[22]. As such, there will always be rare cases where manual query tuning can lead to more efficient plans than those selected by the optimizer. This is particularly true in large DSS environments where execution of extremely complex queries, such as those involving joins across many tables or the ones using complex views with multiple layers of nested queries, is fairly common. Oracle, therefore, provides hints to manually influence the query execution plan in such rare circumstances. Hints provide a very useful mechanism to adjust the execution plan of a particular query without impacting the functioning of other queries accessing the database. There is no way to do the same in DB2. A DB2 DBA can only change the "optimization level" but that affects the execution plans of all queries (i.e. queries other than those that the DBA is trying to tune) – and not necessarily for the better. DB2 users and administrators feel this pain. Scott Hayes, a long time DB2 user, writes in an article in the DB2 magazine, *"One bad SQL statement can ruin your whole day. Time and time again I've seen a single, relatively simple SQL statement bring a finely tuned database and machine to it's knees. For many of these statements, there isn't a DB2 UDB configuration parameter under the sun (or in the doc) that can make right the high cost of an errant SQL statement"* [23]. DB2 users have no recourse

---

[21] "A DB2 That Manages Itself", An International DB2 User's Group Presentation by Guy M Lohman and Daniel Zillo

[22] "Why Does DB2 Optimizer Generate Wong Plans?", A Presentation by IBM Almaden Research Center at International DB2 Users Group Conference, 1999

[23] DB2 Magazine, Spring 2001 issue, Article "Top 10 Performance Tips" (http://www.db2mag.com/db_area/archives/2001/q1/hayes.shtml)

when the DB2 optimizer does not do the right thing. They must simply live with it and according to the DB2 DBA survey mentioned above, they live with it a lot.

#### Statistics Gathering

Cost-based optimizers rely on accurate data distribution statistics to generate optimal execution plans. Since these database characteristics tend to change over time as the data changes, corresponding statistics need to be updated periodically. The way Oracle manages the optimizer statistics is another good indication of how much more sophisticated Oracle technology is compared to that of DB2. In Oracle, gathering statistics is a single-step operation. The DBA just issues a command and Oracle automatically determines which tables have "stale" statistics and need to be refreshed, what sample size to use, which columns to build histograms on, etc., and then gathers the appropriate statistics. In DB2, it is the DBA's responsibility to decide which, how and when the statistics are to be refreshed, without any help from the database whatsoever. In other words, A DB2 DBA has to manually track the tables that have changed since last statistics generation, be intimately familiar with data distribution to determine the columns needing histogram and, go through trail-and-error process to determine the appropriate sample size for the data to be analyzed. That is not all. They must also rebind all the application packages in order to take advantage of new statistics. According to DB2 Administration Guide: Performance, "*Ideally, you should rebind application program after running statistics because the query optimizer may choose a different access plan given the new statistics*"[24]. When gathering statistics is complex as it is with DB2, it may deter DBAs not to do it as often as needed, resulting in poor execution plans. This perhaps is another reason why DB2 administrators rate query optimization as their most time-consuming task.

### Instance Resource Tuning

#### CPU

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. 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. For a detailed discussion on the Database Resource Manager's capability, refer to Oracle white paper titled "Oracle9i Database Resource Manager" at

[24] DB2 Administration Guide: Performance, Chapter 5, Page 117

http://otn.oracle.com/products/manageability/database/pdf/9i_Resource_Mgr_TWP.pdf,

DB2 also provides a tool called "Governor" which can be used to monitor and change behavior of applications that run against a database. DB2 Governor claims to have many of the same capabilities as the Database Resource Manager including, enforcement of resource limits, assignment of relative priorities and, application of different behavioral rules (or resource management policies) during different time window. However there are some significant fundamental differences in how the two tools work and what they actually do. Unlike the Database Resource Manager, which is completely integrated with the Database kernel, DB2 Governor runs as an external daemon that needs to be separately managed[25] meaning more work for the database administrator. Also, while the Database Resource Manager allows administrators to implement deterministic resource management policies by allocating CPU in terms of percentages among different applications, DB2 governor can only prioritize processing of one application over another. The resource allocation mechanism provided by the Database Resource Manager is significantly superior to the conventional priority based scheme that DB2 Governor offers. Using percentages to allocate CPU ensures that all resource consumer groups receive a certain minimum resource and hence cannot be starved by a high demand from other groups. Finally, since the Database Resource Manager runs continuously within the database, it is in a better position to respond to workload fluctuations as against the DB2 governor whose daemon wakes up only periodically. Overall, Oracle provides a much more comprehensive and effective resource management solution than DB2 does.

#### Memory

Since memory access is several times faster than accessing data from disk, effective utilization of memory is key to optimal system performance. Administrators therefore, continuously strive to tune memory related database parameters to maximize performance on one hand and ensure the most efficient use of system memory on the other. The Oracle Database makes this job simpler by automating the major memory management tasks and allowing the database to dynamically alter its memory usage.

Changing the memory configuration of an active DB2 instance requires shutting the instance down, potentially resulting in lost business opportunities. As a matter of fact, changes to any DB2 database manager parameter take affect only after the instance is restarted[26]. Oracle answers this challenge by allowing memory to be added to or removed from a database instance while it is still running. The dynamic memory management capability of Oracle allows administrators to respond to the changes in the available system memory without compromising database, and therefore application, availability. These capabilities also enable

[25] DB2 Administration Guide: Performance, Chapter 9, Page 281
[26] DB2 Administration Guide: Performance, Chapter 13, Page 332

enterprises to integrate the Oracle Database with other workload management products, such as operating system resource managers, allowing distribution of system resources among different applications according to business needs.

Besides eliminating the downtime required for reconfiguration, Oracle also helps administrators in ensuring optimal use of available memory. It includes a number of advisories to help administrators precisely determine the amount of memory required to maximize database performance. It must be stressed here that these advisories are not a part any external tool. Instead, they are built right into the server and use patent pending simulation algorithm to generate the accurate advisory data out of the box with absolutely minimal overhead.

The Buffer Cache Advisory, for example, can predict the cache "miss" rate for various sizes of the buffer cache ranging from 10% to 200% of the current size. Similarly, the Shared Pool Advisory estimates the changes in total time that the database spends in parsing queries for different sizes of the shared pool. Since excessive parsing activities can cause a CPU bound system, the Shared Pool Advisory helps administrators in optimizing the use of both CPU and memory resources. Thanks to these advisories, Oracle DBAs no longer need to indulge in time-consuming bouts of trial-and-error to determine memory allocations, nor do they waste system memory due to over-allocation. The absence of any similar functionality in DB2 requires administrators to either rely on their experience or use empirical approaches to tune the database performance.

**"Oracle9i also automates many of the processes we've always done manually..... Now our staff doesn't have to spend time tuning, which is a tedious, time-consuming process--it can work on the big challenges that must be dealt with by experts, not machines." - Acxiom**

In addition, Oracle completely self-tunes the memory used by active queries for operations such as sort, hash join, bitmap merge, etc.. Administrators just need to specify the total amount of private memory available for an instance and Oracle automatically distributes this memory among various active sessions for maximum throughput. With DB2, DBAs need to manually adjust as many as 12 parameters to control the memory used by an active operation[27]. Failure to set these parameters correctly may lead to disastrous performance. Oracle's automatic management of SQL execution memory not only relieves administrators of this manual task but also helps improve response time for queries, especially in a Decision Support System environment. Furthermore, Oracle also helps administrators decide how much of private memory an instance requires to support a given workload. Using the SQL execution memory advisory, administrators can predict the effect of increases or decreases in instance private memory on the performance of long running queries. DB2 provides absolutely no advice on how the 12 parameter governing the use of private memory should be adjusted making an already difficult task of performance tuning almost impossible.

## CROSS-PLATFORM PORTING

One of the great strengths of the Oracle database architecture is its portability. Oracle is recognized as the most portable database in the world. This means that

[27] DB2 Administration Guide: Performance, Chapter 13, Page 341

an Oracle database on Unix has the same code base as the one running on an IBM Mainframe ensuring that the management of the Oracle database on one platform in no different than on another.

The same is not true for IBM's DB2 database. As stated earlier, it has separate code bases across different platforms. In reality, DB2 on UNIX/Windows, DB2 on AS/400 and DB2 on OS/390 (i.e. IBM Mainframe) are three completely different products, even though they are both called by the same name. The "three DB2s" have different features and functionality, which at times are odds with each other. For example, DB2 on OS/390 uses shared disk approach for clustering while DB2 on AS/400 and Unix/Windows uses the shared nothing architecture. Similarly, DB2 on OS/390 supports only range partitioning while the other two code bases support only hash partitioning. Consequently, it may not possible to move a DB2 application from one platform to another without first making significant design modifications resulting in additional development costs. This imposes an artificial restriction on application development and might even lead to unnecessary design compromises, as business applications have to be designed for a specific set of machine and database software. With Oracle, an application can be designed independent of the platform with the sole objective of addressing the business requirements in the best possible manner.

**"Through 2003, IBM's main DB2 offerings on S/390, AS/400 and its portable Unix and NT products will continue to evolve as architecturally disparate products without transparent application portability or significant DBA skills leverage (0.8 probability)."**

- **Gartner Group, "IBM DB2: Family Unity Through Marketing" Research Note, January 2000**

DB2's lack of cross-platform portability also demands that a DBA be minutely aware of the differences between "different DB2s" in order to avoid taking wrong design decisions. This just makes the already tough life of DBAs and programmers more difficult. Also, since administrators may need to learn new commands/tools to perform common administrative tasks and familiarize themselves with the new concepts, businesses must either retrain their existing DBAs or hire new ones while moving their application across platforms. As a result, training and hiring costs are also needlessly increased due to this complexity.

## INSTALLATION AND CONFIGURATION

### Instance Creation

Instance creation is among the most basic, and at the same time, a very vital DBA function. Oracle has a simple, easy-to-use GUI tool, Database Configuration Assistant (DBCA), for creating database instances. In DB2, on the other hand, numerous manual steps are needed that require the DBA to switch between command line and the GUI tool (Control Center) to create a new instance[28].

DBCA also automatically tailors the initialization parameter settings depending on the workload that the new database is intended to support. In other words, a newly created Oracle database using DBCA will have the appropriate parameter settings depending on whether it will be used for Data Warehousing, OLTP or Mixed workload applications In this area as well, DB2 is not very accommodating since

[28] IBM DB2 Universal Database, Administration Guide: Implementation, pp. 62-66.

the newly created instance has to be manually tuned subsequently which means more work for the DBA.

In general, creating an instance in Oracle is significantly easier and far more flexible than DB2. Instance creation is a discontinuous and unintuitive process in DB2 requiring an in-depth knowledge of the system. Oracle is the clear winner in this area.

### 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.

DB2 has no comparable feature to this. The only way to create a new DB2 database that is identical to an existing one is restore a backup of the source database to the new location. The obvious drawback of this method is that the database structure alone cannot be duplicated; instead both the data and the database have to be copied. The second drawback of this method is that it requires DBA handling and hence cannot be done unsupervised. In DB2, if the 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 re-created manually using scripts or the GUI tool. This is an extremely cumbersome and error-prone method and, is certainly not as easy as the functionality provided by Oracle.

## CONCLUSION

According to an Aberdeen Group study, deployment and ongoing management costs contribute as much as 65% to the total database cost of ownership[29]. By providing a rich set of self-management capabilities and administrative tools to simply database management, coupled with its traditional strengths in scalability, performance and high availability, Oracle makes it possible for businesses to meet

[29] Server Cost of Ownership Study, The Aberdeen Group, 1998

the difficult challenge of managing their IT infrastructure at the lowest possible cost without compromising service level objectives. On the other hand, DB2 provides a very rudimentary set of functionality, is more expensive to own, and is years behind Oracle in terms of technology innovation.

## APPENDIX: SUMMARIZED COMPARISION TABLE

| Feature/Task | Oracle9*i* | DB2 UDB 7.2 |
|---|---|---|
| **Management Tools** | | |
| - Enterprise IT Management Framework | Yes | No |
| - Proactive Monitoring and Alerts | Yes | No |
| - Quality of Service Reporting | Yes | No |
| **Schema Management** | | |
| - Online Table Redefinition | Yes | No |
| - Online Index Creation | Yes | No |
| - Online Index Rebuild | Yes | Yes |
| - Hash Partitioning | Yes | Yes |
| - Range Partitioning | Yes | No |
| - List Partitioning | Yes | No |
| - Composite Partitioning | Yes | No |
| - Local Index | Yes | Yes |
| - Global Partitioned Indexes | Yes | No |
| - Global non-partitioned Indexes | Yes | No |
| - Schema Change Management Tools | Yes | No |
| - Quiesce Database (to rebuild packages, procedures) | Yes | No |
| **Space Management** | | |
| - Maximum Row Size Limit | No | Yes |
| - Tool Assisted Capacity Planning | Yes | No |
| - Resumable Transactions | Yes | No |
| **Data Loading and Archiving** | | |
| - Logical Backup/Restore using EXPORT/IMPORT Utilities | Yes | No |
| - Restartable Load Utility | Yes | Yes |
| - Loading Data into Multiple Tables During the Same Session | Yes | No |
| - Data Accessible After Load (without any manual intervention) | Yes | No |
| - SQL Access to Data Stored in External Files | Yes | No |
| - Transportable Tablespaces | Yes | No |
| **Backup & Recovery** | | |
| - Bounded Crash Recovery Time | Yes | No |
| - Automatic Log Archiving | Yes | No |
| - Backup Throttling | Yes | No |
| - Centralized Recovery Catalog | Yes | No |
| - Resumable Backup & Restore | Yes | No |
| - Incremental Backup of LOB Data | Yes | No |
| **Performance Management** | | |
| - Manual Lock Conflict Monitoring & Resolution | No | Yes |

| | | |
|---|---|---|
| - Automatic Identification of Resource Intensive Queries | Yes | No |
| - Granular Adjustment of Query Execution Plans | Yes (using hints) | No |
| - Automatic Identification of Tables Needing Optimizer Statistics Refresh | Yes | No |
| - Automatic Sample Size Determination for Optimizer Statistics Refresh | Yes | No |
| - Automatic Creation of Histograms | Yes | No |
| - Manual Application Rebind after Optimizer Statistics Refresh | No | Yes |
| - Deterministic Distribution of Resources Among Users and Applications | Yes | No |
| - Prioritizing Users/Applications | Yes | Yes |
| - Dynamic Memory Management | Yes | No |
| - Memory Sizing Advisories | Yes | No |
| - Self Tuning SQL Execution Memory | Yes | No |
| **Seamless Cross Platform Porting** | Yes | No |
| **Installation & Configuration** | | |
| - Tools Assisted Database/Instance Creation | Yes | No |
| - Tool Assisted Database/Instance Duplication | Yes | No |

**Technical Comparison of Oracle9*i* Database vs. IBM DB2 UDB: Focus on Manageability**
**May 2002**
**Author: Sushil Kumar, Mughees Minhas**
**Contributing Authors:**

**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**