Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on High Availability

An Oracle White Paper
February 2003

Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on High Availability

Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on High Availability

EXECUTIVE OVERVIEW

Today's businesses depend heavily on their databases. Should applications and data become unavailable, the entire business may halt. Revenue and customers may be lost and penalties may be incurred. Bad press can have a lasting effect on both customers and stock prices. Certainly, providing continuous data availability is essential for today's businesses.

Over recent years there have been various efforts to quantify the cost of downtime, planned or unplanned. According to Standish Group's studies, one minute of system downtime can cost an organization anywhere from $2,500 to $10,000 per minute. Using that metric, even with 99.9 percent data availability, a company may face annual downtime costs of nearly $5 million dollars.

The Oracle database provides a complete and simple cost-effective high availability solution. It takes care of most scenarios that might lead to data unavailability, such as system failures, data failures, disasters, human errors, system maintenance operations and data maintenance operations. IBM DB2 Universal Database (DB2 UDB) provides only very basic functionality for both high availability and data protection.

In an AMR Consulting study[1], Oracle is consistently determined to be superior to DB2 in terms of supporting continuity of business operations, protecting business assets, and promoting business efficiency. Oracle is also the database that runs mission critical, highly available enterprise applications for well-known global companies[2] such as Merrill Lynch, Citigroup, Southwest Airlines, British Telecom, General Motors, Best Buy, Lufthansa, Priceline, eBay and Amazon.com. Worldwide, companies such as these depend on the Oracle database's reliability to provide continuous service to their customer bases.

---

[1] *"The Operational and Strategic Dimensions of Database Selection", AMR Consulting, Dec 2001*
[2] *http://www.oracle.com/customers*

INTRODUCTION

One of the true challenges in designing a highly available solution is examining and addressing all possible causes of downtime – planned or unplanned. Every business, regardless of industry and size, faces similar challenges in endeavoring to provide higher availability.

Unplanned Downtime

Unplanned downtime due to system failures, data failures, disasters, and human errors are unfortunately common challenges for businesses. According to Disaster Recovery Journal, however, disasters due to flood and fire typically only contribute to some 3% of unplanned downtime – although the potential for extended periods of downtime is far greater when faced with a disaster situation. Human errors and system/hardware failures are more likely causes of failure, accounting for 36% and 49% of unplanned downtime, respectively.

Planned Downtime

In contrast to unplanned downtime, planned downtime is much more within the control of IT departments and service providers. Planned downtime for upgrades, data and index reorganizations occur within every e-business. Some businesses plan ahead and perform scheduled maintenance tasks while others tend to perform maintenance on an as needed basis. However, planned downtime still interrupts system and data availability and incurs a cost. The challenges lie in ensuring that planned downtime is kept to a minimum.

Oracle's High Availability Solutions

As shown in Fig 1, Oracle Database has a comprehensive High Availability solution stack that addresses every aspect of unplanned and planned downtimes, ensuring 24x7 business continuity for an enterprise.

While Fig 1 shows the key Oracle solutions that address a critical area of downtime, Oracle also has other complementary solutions (e.g. LogMiner, Recovery Manager, etc.) in some of these categories. In terms of their breadth and depth of functionality, this makes Oracle's High Availability solutions simply unparalleled in the industry. Please visit http://otn.oracle.com/deploy/availability/content.html for further details on Oracle's High Availability solutions.

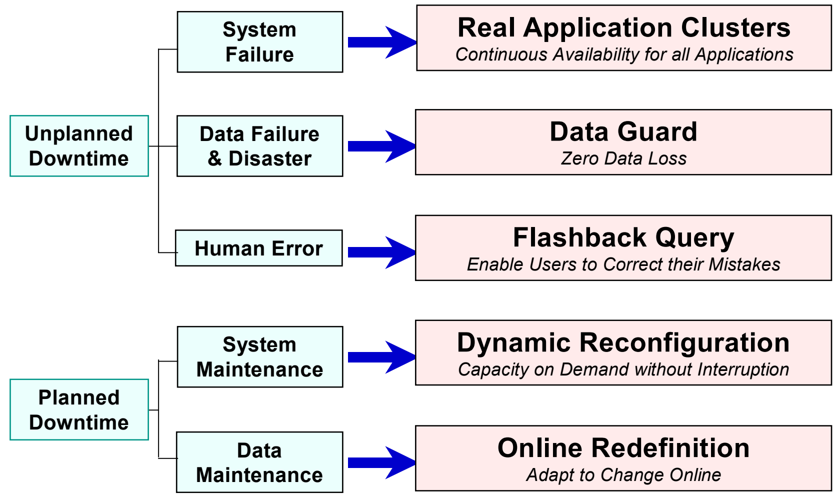

Fig. 1: Oracle's High Availability Solutions

For additional insights into Oracle's leadership in the OLTP space, refer to the Gartner Inc. report titled "*Online Transaction Processing RDBMS Magic Quadrant*" dated May 2002. Similarly, refer to the META Group report "*RDBMS for Unix/Windows METAspectrum: Results Overview – Performance Criteria*" dated Dec 2002 for an analysis of the leading databases using a *presence and performance* criterion and Oracle's leadership position in that regard.

IBM DB2, even with its recent v8.1 release, cannot match the depth and breadth of Oracle's high availability offerings. This paper takes each high availability challenge and compares how Oracle Database and IBM DB2 UDB[3] address the challenge. Following this analysis, it will be clear that the recent DB2 v8.1 release is only an attempt on IBM's part to fill some of the holes in its technology and architecture, and IBM continues to lag Oracle significantly in the area of High Availability.

In the rest of this document, *Oracle* refers to Oracle9*i* Database Release 2 Enterprise Edition, and unless otherwise stated, *DB2* refers to IBM DB2 Universal Database Enterprise Server Edition (DB2 ESE), version 8.1.

---

[3] *Although IBM promotes DB2 UDB as a single product, there are actually 3 code bases with distinctly different capabilities. For the purposes of this paper, all references are to the Unix/Windows version of DB2 UDB, unless otherwise stated.*

For easy reference, the following table provides a list of the major differentiators between Oracle and DB2 for every category of downtime. These differentiators are described in further details in the rest of the document.

Table 1: Key High Availability Differentiators – Oracle vs. DB2

Addressing System Failures	Oracle	DB2
Predictable recovery time	Yes	No
Recovery advisories	Yes	No
Data availability during rollback	Yes	No
Transparent application failover	Yes	No
Transparent application scalability with clustering	Yes	No
Expensive two phase commit overhead for writes	No	Yes
Expensive, unnecessary broadcast of queries to all nodes	No	Yes
Leverage data cached in any node to avoid disk access	Yes	No
Query performance unaffected by data skew in partitions	Yes	No
Load is balanced after node failure	Yes	No
Addressing Data Failures	**Oracle**	**DB2**
Comprehensive backup and recovery tool	Yes	No
Backup information in centralized repository	Yes	No
Split mirror without any impact	Yes	No
Querying during point-in-time recovery	Yes	No
Block-level media recovery	Yes	No
Read-only tablespace	Yes	No
Resumable backup and restore	Yes	No
Incremental backup of LOBs	Yes	No
Range/List Partitioning	Yes	No
Rolling window data management	Yes	No
Global partitioned indexes	Yes	No
Storage-level integration to prevent corruption	Yes	No
Addressing Disaster Recovery	**Oracle**	**DB2**
Ensure zero data loss	Yes	No
Automated failover and switchover operations	Yes	No
Physical and Logical standby databases	Yes	No
Flexible data protection modes	Yes	No
Delayed log application	Yes	No
Offload backups to standby database	Yes	No

Offload reporting to standby database	Yes	No
Auto-resynchronization after network is restored	Yes	No
GUI-based management interface	Yes	No
Addressing Human Errors	**Oracle**	**DB2**
Retrieve data from the past using SQL queries	Yes	No
Mine logs and audit changes using a SQL interface	Yes	No
Flexible tablespace point-in-time recovery	Yes	No
Addressing System Maintenance	**Oracle**	**DB2**
Add a node to a cluster online	Yes	No
Extensive support to adjust memory online	Yes	No
Helpful advisories on memory management	Yes	No
Most configuration parameters may be modified online	Yes	No
Addressing Data Maintenance	**Oracle**	**DB2**
Scalable data maintenance through rolling windows	Yes	No
Resumable space allocation	Yes	No
Consistent table export without locking	Yes	No
Online table load	Yes	No
Reorganize individual indexes	Yes	No
Add, rename, merge columns online	Yes	No
Create/rebuild/coalesce indexes online	Yes	No
Extensive set of online redefinition capabilities	Yes	No

ORACLE VS. DB2 – ADDRESSING UNPLANNED DOWNTIME

Addressing System Failures

System failures are the result of hardware failures, power failures, and operating system or server crashes. The amount of disruption these failures cause depends upon the number of affected users, and how quickly service is restored. The challenges with system failures lie in ensuring fast recovery, or better still, a higher level of fault tolerance.

As shown in the following table, Oracle provides an array of features that clearly differentiate Oracle from DB2 in terms of how effectively it addresses system failures.

Table 2: Addressing System Failures – Oracle vs. DB2

Addressing System Failures	Oracle	DB2
Predictable recovery time	Yes	No
Recovery advisories	Yes	No
Data availability during rollback	Yes	No
Transparent application failover	Yes	No
Transparent application scalability with clustering	Yes	No
Expensive two phase commit overhead for writes	No	Yes
Expensive, unnecessary broadcast of queries to all nodes	No	Yes
Leverage data cached in any node to avoid disk access	Yes	No
Query performance unaffected by data skew in partitions	Yes	No
Load is balanced after node failure	Yes	No

The following sections provide further details on these differentiators.

Fast Start™ Fault Recovery

The Oracle Fast-Start™ Fault Recovery scheme has been designed to minimize downtimes related to system failures. It has two components – *Fast Start Checkpointing* that optimizes roll forward recovery by continually and incrementally advancing the checkpoint position, and *Fast Start Rollback* that eliminates the delays associated with the rollback phase of recovery.

Fast Start Checkpointing – Predict Average Recovery Time

To control the time to recover from system failures, Oracle allows Mean Time

To Recover (MTTR) to be directly specified via a dynamic parameter, FAST_START_MTTR_TARGET. Oracle continuously estimates the recovery time and automatically adjusts the checkpointing rate to meet the target recovery time. DB2 provides no means to effectively predict or control recovery time. In DB2, the static *SOFTMAX* parameter controls the percentage of recovery log files filled between checkpoints. The DBA has to then guess how this translates into actual recovery time.

Because there is additional overhead with frequent checkpointing, Oracle provides real-time feedback on the cost of the target MTTR through the *v$instance_recovery* dynamic view, as well as a GUI-based advisory through Enterprise Manager. With DB2, it is impossible to determine the runtime cost of adjusting the SOFTMAX parameter.

Oracle also provides an advisory through the *v$mttr_target_advice* view that simulates the cost of a range of recovery scenarios. The simulation runs in real-time based on the current production workload. Based on the output of the advisory the administrator can choose the best tradeoff between very fast recovery time and extra I/O overhead. This takes the guesswork and risk out of configuring for fast recovery.

DB2 version 8.1 offers the ability to skip recovery log files that are not required during recovery of a tablespace. That is nowhere near the rich and comprehensive functionality provided by Fast-Start™ Fault Recovery.

Oracle internal testing has demonstrated a 17 second recovery time on a 400GB database with 2000 concurrent users running 300 transactions per second. According to DB2 documentation, "*If applications running on a partition are issuing frequent COMMITs, 10 minutes following failure on a database partition should be sufficient time to roll back uncommitted transactions and to reach a point of consistency in the database on that partition.*"[4]

Fast Start Rollback – Shorten Worst-Case Recovery Time

Oracle's crash recovery time is immune to long transactions, because Oracle allows users to access the database before instance recovery rollback operation is complete through a unique on-demand rollback technology. With Oracle, once the rollforward processing completes, the database opens for user access. Unlike DB2, Oracle does not wait until all transactions have been rolled back. Instead, transactions are rolled back in the background while new user transactions access the data. If one of these new user transactions encounters data that was locked by a dead transaction, the user transaction instantly rolls back the change to the data made by the dead transaction and continues executing.

---

[4] *DB2 Administration Guide: Planning: Version 7, Chapter 12 - HACMP ES for AIX*

In contrast, DB2's crash recovery is severely impacted by long transactions. Because DB2 cannot be accessed until all active transactions are rolled back, crash recovery time is dependent on the longest running transaction. In addition, recovery time is further lengthened by long transactions because DB2 cannot switch into a log that contains uncommitted transactions.

For additional details on the topic of crash recovery, refer to the article "*Debunking IBM's High Availability Deception*"[5].

Minimize Time to Resume Full Capacity

Oracle offers several advanced HA solutions that minimize the time to resume full capacity after a failure. Oracle's shared cache cluster model allows clients to pre-connect to an alternate instance, thus avoiding logon storms after system failures. This capability is known as Transparent Application Failover (TAF). TAF is implemented in Oracle Call Interface (OCI). However, ODBC drivers and precompilers are built on top of OCI, so OCI programming knowledge is not necessary to make use of TAF.

In addition, the alternate Oracle instance suffers no performance impact after failover, which greatly speeds up recovery time. It is not uncommon for large systems today to have gigabytes or tens of gigabytes of buffer cache. Warming up a buffer cache this large can take a very long time. With Oracle Real Application Clusters, failover occurs to an instance that has a warm cache. In DB2, failover always involves starting a new instance from scratch with a cold cache. This issue will become more important as all major platforms adopt 64-bit architectures, and the price of memory continues to decline.

Increase Fault Tolerance with Real World Clustering

"RAC achieved failover times of between 10 seconds and 1 minute. During this period, users were automatically transferred to the remaining node, and completely unaware of the failure."
– British Telecom

Oracle and DB2 both offer clustered database solutions. Oracle provides a shared cache clustered solution called Real Application Clusters (RAC). DB2 provides a shared-nothing clustered database available through the Database Partitioning Feature (DPF). DPF is available as an option on top of DB2 Universal Database Enterprise Server Edition (DB2 UDB ESE, or DB2 for short). The DB2 clustered model depends on this data-partitioning feature. Each node in a DB2 cluster houses one or more database partitions. In the event of an unexpected node failure, both RAC and DB2/DPF can transparently recover the database. However, recovery times will be faster with RAC due to the capabilities described above, plus the following:

- DB2 relies on the cluster manager to restart the partition on a surviving node. This requires the DB2 processes to be started, shared memory to be initialized, and database files to be opened.

---

[5] *http://www.oracle.com/features/insider/index.html?1212_oi_ibmha.html*

- After the database has been recovered, applications can be expected to obtain their original response times faster in RAC because the data and the packages needed by the application may have already been cached in the surviving nodes.

RAC allows the failed connections to be evenly distributed across the surviving nodes. With DB2/DPF, the underlying cluster manager determines which of the surviving nodes take over the disks belonging to the failed partition(s). To avoid load skew, DB2/DPF must be configured such that each surviving node takes ownership of the same amount of data. This is achieved by creating multiple partitions on each node. For example, if there are four nodes, three partitions are created on each node for a total of twelve partitions.[6] If there are n nodes, for even redistribution of partitions to the surviving nodes under all failure scenarios, the number of partitions equals the least common multiple of n, n-1,...,1, where n equals the number of nodes. For each partition, a preferred owner or takeover list is created using the cluster software (such as HACMP, MSCS) so that the partitions are evenly redistributed across the nodes.

Clearly, in a high-availability configuration, the number of partitions grows quickly with the number of nodes. This creates several problems:

- It takes more work to administer the cluster. Each partition has its own configuration parameters, database files, and redo logs that need to be administered.
- Each physical node's resources may be underutilized. Although multiple partitions are owned by the same physical node, the partitions cannot share memory for the buffer pool, package cache, etc. This causes under-utilization because it is possible to make better use of a single piece of memory rather than fragmented pieces of memory.
- The probability of load and/or data skew increases with the number of partitions.

Most importantly, in DB2/DPF, the cost of a query or update that does not specify the partitioning key in the 'WHERE' clause increases linearly with the number of partitions. For example, if the customer table is partitioned on customer number, then any query on customer name is twelve times more expensive on a twelve-way partitioned system than on a system with no partitions. That is, the query requires twelve times the CPU and twelve times the I/O. This surprising fact is inherent in the way DB2/DPF implements partitioning. Each table has exactly one partitioning key. SQL operations that specify the partitioning key can be routed to the correct partition for execution. SQL operations that do not specify the partitioning key must be broadcast to all partitions because DB2 has no way of knowing which partition holds the data.[7]

---

[6] *Nomani, Aslam and Pham, Lan IBM, DB2 Universal Database for Windows High Availability Supporting Using Microsoft Cluster Server - Overview, TR-74.176, May 2001*
[7] *Ref (a) DB2 Administration Guide: Planning: Version 8, Chapter 5 - Physical Database Design, (b) DB2*

In DB2/DPF, the inter-process communication for a given workload increases with the number of partitions. For example, an application that scales to four logical nodes may not scale to twelve logical nodes. However, for high availability, DB2/DPF will force the application to run on twelve partitions.

Figure 3 depicts this architectural drawback in DB2. In this case, customer number is the partitioning key, so any query that does not include it, will encounter these performance and scalability shortcomings.

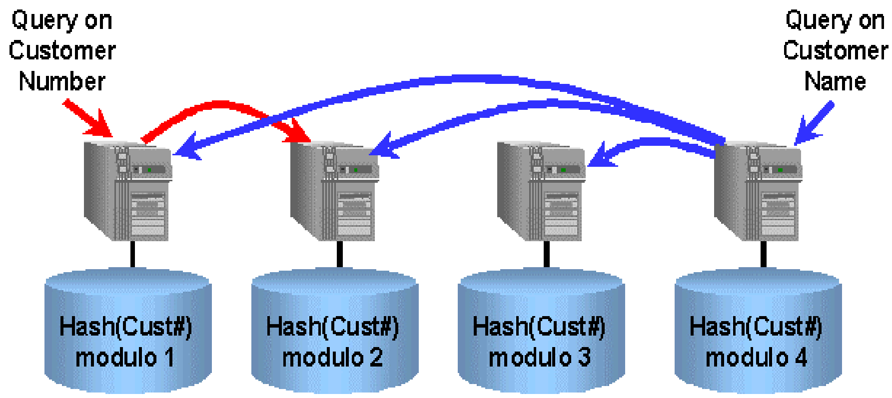

Fig. 2: Limitations of DB2's Partitioning Feature

IBM sometimes recommends replicating a table across all nodes to work around this fundamental scalability issue of DB2/DPF. For example, they might recommend replicating the customer table across all nodes. This is clearly not a scalable or efficient solution because the amount of disk space consumed by the customer table (and all its indexes) will quadruple in a four-node system and continue to increase as nodes are added. Also, if replication is done, then all inserts, updates, and deletes must be executed on all nodes. So queries are sped up only in return for a large slowdown in updates. Further, high availability is compromised because any node slowdown or crash will affect modifications across the entire system.

"The Oracle database has never caused any downtime after migrating to the clustered database architecture. We have been using clusters for 17 months with 100% uptime for the Oracle databases." – Simon Leung, Senior Manager, Database Architecture, VeriSign

Oracle provides additional cluster diagnostics capabilities that DB2 does not offer. For example, Oracle's Flash Freeze Diagnostic feature enables offline diagnosis of the failed instance independent of the cluster. This feature reduces the time the systems remain off-line for diagnostics. In addition, capturing all relevant data available upon the first failure can help reduce future failures and therefore increase availability.

Application Development Guide: Programming Client Applications: Version 8, Chapter 17 - Programming Considerations for Partitioned Database Environments

For further details, please refer to the Oracle white paper titled "*Technical Comparison of Oracle9i Real Application Clusters vs. IBM DB2 UDB ESE v8.1*".

Addressing Data Failures

It is extremely important to design a solution to protect against, and recover from, data and media failure. A system or network fault may prevent users from accessing data, but media failures without proper backups can lead to lost data that cannot be recovered.

As shown in the following table, Oracle provides an array of features that clearly differentiate Oracle from DB2 in terms of how effectively it addresses data failures.

Table 3: Addressing Data Failures – Oracle vs. DB2

Addressing Data Failures	Oracle	DB2
Comprehensive backup and recovery tool	Yes	No
Backup information in centralized repository	Yes	No
Split mirror without any impact	Yes	No
Querying during point-in-time recovery	Yes	No
Block-level media recovery	Yes	No
Read-only tablespace	Yes	No
Resumable backup and restore	Yes	No
Incremental backup of LOBs	Yes	No
Range/List Partitioning	Yes	No
Rolling window data management	Yes	No
Global partitioned indexes	Yes	No
Storage-level integration to prevent corruption	Yes	No

The following sections provide further details on these differentiators.

Comprehensive Backup and Recovery Capabilities

Both Oracle and DB2 can perform basic online and offline backup and recovery. However, Oracle's extensive backup and recovery capabilities provide much more than the basic functionality provided by DB2. These capabilities are integrated through Oracle Recovery Manager (RMAN), which is a comprehensive tool to automate and manage the backup and recovery of the database – eliminating operational complexity while providing superior performance and availability for the database. As a result, Oracle can handle almost any backup and recovery requirement.

RMAN keeps backup metadata in the controlfile of the backed up database or optionally in a recovery catalog. This increases the resilience of the backup information and allows easy querying of backup information. It also acts as a central repository for backup information across the enterprise, providing a single point of management. DB2 does not allow backup information to be placed in a central repository.

Split mirror backups are useful because they produce instant backups. Both Oracle and DB2 provide facilities for split mirror backups. However, Oracle can split a mirror while the database is running and writing to the disks. DB2 has to suspend database I/O to perform a mirror split, thus making the database unavailable for write I/O-s during this operation.

Archived log files can become damaged. Oracle allows damaged archivelog files to be scavenged using Oracle LogMiner™, thus recovering some of the transactions recorded in the log files. With DB2, a corrupt archived log file means loss of all transactions in that particular log file plus any archived log files created after the damaged log file.

When performing a point in time recovery, Oracle allows querying the database without terminating recovery. This is useful to determine whether errors affect critical data or non-critical structures (such as indexes). Oracle also allows trial recovery in which recovery continues but can be backed out if an error occurs. It can also be used to "undo" recovery if point-in-time recovery has gone on for too long. DB2 does not have this capability.

Efficient VLDB Backup and Recovery

Very Large Databases (VLDBs) such as data warehouses require efficient backup, restore, and recovery methods. Towards this purpose, Oracle offers several innovative technologies such as block-level media recovery, read-only tablespaces, etc. that are integrated through RMAN.

With Oracle's block-level media recovery feature, if only a single block is damaged then only that block needs to be recovered. The rest of the file, and thus the table containing the block, remain online and accessible, increasing data availability. DB2 cannot recover data in single-block units, thus requiring the entire file to be taken offline, restored, and recovered.

Oracle can minimize backup time through the use of read-only tablespaces. Backing up read-only tablespaces needs to occur only once. DB2 does not support read-only tablespaces, thus tablespaces must be backed up often because they cannot be placed into read-only mode.

Another time saving feature Oracle provides through RMAN is resumable backup and restore operations. With Oracle, these operations can be restarted from the point of failure. Because DB2 has no such capability, problems during backup or restore means time lost while the entire operation starts from the beginning. To further compound the problem, in DB2 "*a table space backup operation and a table space restore operation cannot be run at the same time, even if different table spaces are involved.*"[8]

LOBs are very large and often store images, sound files, etc., that never change. Incremental backup is critical for these. While Oracle can perform incremental backups of LOBs, DB2 is unable to do so: "*DB2® now supports incremental backup and recovery (but not of long field or large object data).*"[9].

Maintenance Reduction and Failure Isolation with Partitioning

As databases grow larger, they may become extremely cumbersome to manage. Partitioning of data allows administrators to divide large tables into smaller and more manageable chunks without having to change any underlying application code. This allows maintenance tasks to be performed at the smaller partition level, allowing the bulk of the data to remain unaffected during maintenance. Another benefit of partitions is fault containment. A failure, such as a media failure or corruption, is contained to partitions resident on the failed disk. Only the affected partition needs to be recovered – reducing recovery time and leaving other unaffected partitions online during partition-recovery process. This increases overall data availability.

Though both Oracle and DB2 support data partitioning, Oracle offers a wider array of partitioning options. While Oracle supports hash, list, range, and composite (both range/hash, and range/list) partitioning schemes, DB2 only supports hash partitioning. This difference is significant because though IBM claims that DB2 offers the maintenance reduction and failure isolation benefits of partitioning, the fact that DB2 is limited to hash partitioning prevents the administrator from being able to determine exactly what data, and hence operations, will be affected. With hash partitioning, the database server determines the data placement. With range and list partitioning, the user controls data placement. For example, if the data were range-partitioned by geographic region, only a single geographic region would be affected by a failure. Furthermore, if the data is partitioned by region and time, a failure affects only a small time window in a single region.

Often, not all data in a large table has the same access characteristics. Pending orders may be accessed more frequently than closed orders, or analysis of last quarter's sales may be more common than analysis of sales from a quarter three

---

[8] *DB2 Data Recovery and High Availability Guide and Reference: Version 8, Chapter 2 - Database Backup*
[9] *DB2 Data Recovery and High Availability Guide and Reference: Version 8, Chapter 1 - Developing a Good Backup and Recovery Strategy*

years ago. Partitioning by range and/or list allows for intelligent storage management of data, whereby frequently accessed data can be stored in a separate partition that is kept on the fastest or most reliable disk subsystem. This technique, known as *rolling window*, is used frequently in data-warehousing environments. Using rolling windows, frequently-accessed data can be backed up more often than infrequently-accessed data. Restore operations can also be sped up. During a restore, the administrator can quickly restore just the last three months of data and bring the system online while restoring older data in the background.

DB2, due to its limited partitioning capability, cannot support *rolling window* data management. DB2's hash partitioning scheme requires data in all partitions to be redistributed, therefore increasing the time required to load new data and also decreasing data availability as the table is locked during the data redistribution process.

DB2 offers a way to simulate rolling windows through a UNION ALL view over its partitions[10], but this is not a scalable and easily manageable approach.

Oracle also implements global partitioned indexes to isolate index failures while maintaining query efficiency. DB2 only supports local indexes, causing queries to be broadcast across all partitions unless the partitioning key is specified in the query predicate.

End-to-End Data Integrity

Oracle's Hardware Assisted Resilient Data (H.A.R.D.) initiative[11] allows storage vendors to validate Oracle blocks before writing them to disk. Several storage vendors have already validated their products by participating in this initiative. DB2 has no way to prevent corrupt blocks from being written to disk.

---

[10] *Refer to "Partitioning in DB2 Using the UNION ALL View", 2002, http://www7b.software.ibm.com/dmdd/library/techarticle/0202zuzarte/0202zuzarte.pdf*

[11] *Refer to http://otn.oracle.com/deploy/availability/htdocs/HARD.html for further details*

Addressing Disaster Recovery

Oracle Data Guard Ensures Bulletproof Data Protection

Oracle Data Guard offers the most complete and robust disaster recovery solution in the industry. It provides the following benefits:

> *"Oracle's Data Guard scored highest in our rating as a HA solution what was cost effective, reliable & had an architecture that can scale to our needs." – Kiran Dattani, Data Technology Manager, ProQuest Information and Learning (ref. http://www.aaoug.org/pres/ioug_whitepaper.doc)*

- Disaster recovery and high availability through the use of transactionally consistent standby databases
- Complete data protection, including zero data loss protection
- Efficient utilization of standby system resources
- Flexibility in data protection to balance availability against performance requirements
- Automatic resynchronization after network communication problems are repaired and connectivity is restored
- Centralized and simple management through a GUI
- Complete integration with Oracle database

Oracle Data Guard empowers customers to survive disasters of many forms. Data Guard automates the complex tasks of disaster recovery and provides the monitoring, alerting and control mechanisms to maintain a standby operation. In addition, Data Guard reduces planned downtime by utilizing the standby server for maintenance and routine operations in addition to reporting.

IBM does not offer a solution comparable to Oracle Data Guard. With DB2, every standby database is a custom job; tasks as basic as shipping redo logs to the standby site depend on user-written log-transfer callouts. Besides, a critical shortcoming[12] of DB2's log-shipping feature is that the standby database is in a continuous roll-forward phase. This means, unlike Data Guard, the standby database in DB2 is completely offline and not available for use, wasting valuable system resources. For DB2, if the primary database fails, any remaining log files have to be copied over to the standby machine, and subsequently applied (the *rollforward* phase). For this reason, and because the granularity of shipment is logs, following a disaster a DB2 standby database environment stands to lose a much larger volume of data (i.e. poor *Recovery Point Objective*: how much data a business can afford to lose in the event of a disaster) and stay down for a longer amount of time (i.e. poor *Recovery Time Objective*: how much time a business can afford to be down in the event of a disaster).

The following table summarizes the key differentiators between Oracle and DB2 in terms of how effectively Oracle addresses disaster recovery.

---

[12] *DB2 Data Recovery and High Availability Guide and Reference: Version 8, Chapter 5 - Introducing High Availability and Failover Support*

Table 4: Addressing Disaster Recovery – Oracle vs. DB2

Addressing Disaster Recovery	Oracle	DB2
Ensure zero data loss	Yes	No
Automated failover and switchover operations	Yes	No
Physical and Logical standby databases	Yes	No
Flexible data protection modes	Yes	No
Delayed log application	Yes	No
Offload backups to standby database	Yes	No
Offload reporting to standby database	Yes	No
Auto-resynchronization after network is restored	Yes	No
GUI-based management interface	Yes	No

The following sections provide further details on these differentiators.

Automated, Flexible Standby Environment

Oracle Data Guard supports two kinds of standby databases – physical standby databases that use Redo Apply technology, and logical standby databases that use SQL Apply technology. A single Data Guard configuration can contain multiple physical and logical standby databases, offering customers the flexibility to choose the standby databases that best meet their high availability needs. Data Guard completely automates the process of keeping these standby databases transactionally consistent with the primary.

Compared to Data Guard, DB2 offers a much coarser granularity of data transmission from the primary database to the standby databases. DB2 ships logs, whereas Data Guard ships redo data as transactions occur on the primary. Thus, in the face of disasters, DB2 may lose a large amount of business-critical data. Besides, with Data Guard, and unlike DB2, customers have the flexibility to choose either a configuration that ensures zero data loss, or a configuration that, in case of disasters, may result in only a bounded amount of data loss.

Data Guard supports post-failure automated recovery. It automatically fetches missing logs caused by network outages and fills archive gaps, as required, thereby automatically resynchronizing the standby databases with the primary database. Data Guard also automatically detects damaged logs and retrieves replacements from the primary or other standby databases. DB2 has no built-in method to catch up following a network outage or to handle damaged redo logs. DB2 customers have to manually re-ship logs following a network disconnection or a problem with the standby database.

Oracle Data Guard provides automated planned switchover and unplanned failover from the affected primary database to a chosen standby database. The DBA initiates it with a single command or through the push-button GUI. By facilitating failover and switchover activities, the possibility of administrative errors is dramatically reduced. In contrast, DB2 has no tools for creating or monitoring the standby configuration. DB2 administrators have to code switchover and switchback scripts themselves.

Data Guard provides a GUI (Data Guard Manager) and a command-line based management framework to administer a Data Guard configuration. Data Guard Manager provides a consolidated view of the primary database and all of its standby databases. Data Guard Manager also includes an instantiation wizard that automates the process of creating a standby database, or adding an existing standby database to the Data Guard configuration. Oracle Enterprise Manager is leveraged for notification and monitoring services. DB2 is completely missing such a critical management framework.

Zero Data Loss Protection

The same automated log transport services are used by both physical and logical standby databases. Redo data is transmitted from the primary to the standby as soon as they are available. Additionally, Oracle has the ability to synchronously write redo log updates directly from the primary to the standby database. This provides for a comprehensive "zero-data loss" disaster recovery solution. Oracle provides built-in zero data loss protection modes; no additional third party product is required.

DB2 has no built-in zero data loss failover. That is why IBM admits that for DB2, "*With a disaster, however, it is generally not possible to recover all of the transactions up to the time of the disaster*"[13]. Zero data loss with DB2 is only possible with third-party software and disk-mirroring products[14]. This means a much higher cost due to additional hardware/software investment and much higher complexity due to integration of disparate technologies. Besides, choosing a third-party remote mirroring solution may force DB2 customers to be tied to the one storage vendor that is providing the remote mirroring solution, for both the primary and standby databases. Data Guard imposes no such restriction on its customers.

Minimal Data Loss Combined with Best Performance

Oracle Data Guard also provides a minimal data loss mode that ships transaction changes to the standby database as they are generated on the production database. This mode does not wait for an acknowledgement from the standbys and therefore does not guarantee zero data loss in the event of a

[13] *DB2 Data Recovery and High Availability Guide and Reference: Version 8, Chapter 1 - Developing a Good Backup and Recovery Strategy*

[14] *Creating No Dataloss Standby Databases with IBM DB2 Universal Database V7.2, EMC TimeFinder, and EMC SRDF, http://www7b.boulder.ibm.com/dmdd/library/techarticle/0208cialini/0208cialini.pdf*

disaster. However, the potential for transaction loss is generally minimal, and this mode incurs no extra commit latency on the primary. DB2 has no equivalent mode. In DB2, only full log files can be shipped to the standby, leading to large data loss windows.

Delayed Log Application

Oracle Data Guard also allows the specification of a delay for application of the redo log data once it arrives at the standby site. When specified, this delay creates a window during which user errors can be discovered and prevented from propagating to the standby database. Immediate application means faster recovery and a transactionally consistent standby database. However, delaying the application of logs allows administrators to discover user errors and failover to the standby (which still has consistent data), and resume services on it.

DB2, by contrast, does not support such configurable log application delays and applies all user errors or data corruptions contained in primary database log files to the standby database as soon as they are received.

Offload Work from Primary

Oracle's standby database can be used to offload work from the primary database. While Oracle's physical standby database can be used for reporting once it is open for read-only access, DB2 standby database does not allow end user access because DB2 does not provide a read-only database capability.

In addition, because Oracle's logical standby databases are open for both read and write during recovery, it is possible to query the standby database while the changes in the redo logs are being applied. For example, the logical standby can be used for decision support and can be optimized using different indexes and materialized views than the primary. DB2 offers change data capture mode through its DataPropagator component. It is largely a data replication facility and does not provide the additional data protection mechanisms available through Oracle's logical standby database.

Oracle RMAN allows backups to run on the physical standby database, at the same time redo data is being applied on the standby database (i.e. while it is in the recovery mode). This offloads the backup operation from the production database, reduces resource contention, boosts performance, and enables no-downtime backup windows. On the other hand, DB2 standby databases cannot be backed up while logs are being applied.

With Data Guard, it is also possible to restore a primary database with a backup taken on the standby database. This is not possible with DB2: "*You cannot back up a cloned database, restore the backup image on the original system, and roll forward* *through log files produced on the original system.*"[15]

Automated Standby for Clustered Primary

"From our perspective, going with Oracle9i Real Application Clusters (Oracle9i RAC) and Oracle9i Data Guard was a no-brainer. Nothing else came close to offering us the peace of mind of knowing that the data and systems will always be available to our customers and employees." – David MacDonald, President, Softchoice Corporation

With Oracle Data Guard, either of both of the primary and standby database can be a RAC cluster. All protection modes are supported in these configurations. Automated shipping of redo data and recovery are available for all configurations. With a well-integrated RAC and Data Guard offering, Oracle offers an end-to-end High Availability solution that is simply unparalleled in the industry.

IBM never discusses standby database configurations for DB2. What happens if one node fails, or a node is added to the configuration? What happens if transactions are shipped from some nodes but not from all of them?

DB2's custom-created standby database *solution* is less robust, costs much more to implement, and certainly does not meet the disaster recovery and high availability needs of a 24x7 business.

Addressing Human Errors

Many studies on availability have concluded that human error is the greatest threat to application availability. A survey by the Disaster Recovery Journal estimated that some 36% of unplanned downtime is due to human errors. Human errors include accidents (e.g. deleting important data), unintended outcomes (e.g. an action that monopolizes system resources), and even sabotage. The real challenge with human errors lies in identifying the impact of the errors, and taking the fastest route to recovery.

As shown in the following table, Oracle provides clearly differentiating features compared to DB2 in terms of how effectively it addresses human error situations.

Table 5: Addressing Human Errors – Oracle vs. DB2

Addressing Human Errors	Oracle	DB2
Retrieve data from the past using SQL queries	Yes	No
Mine logs and audit changes using a SQL interface	Yes	No
Flexible tablespace point-in-time recovery	Yes	No

The following sections provide further details on these differentiators.

[15] *DB2 Data Recovery and High Availability Guide and Reference: Version 8, Chapter 5 - Introducing High Availability and Failover Support*

"Before Oracle9i's Flashback Query, a restore was required to recover lost data. Now, using the Flashback option, human error can be easily undone."
– Tim Donar, Acxiom

Data Recovery

Oracle's Flashback Query allows an administrator or user to view the state of the data at a point in time in the past without requiring any structural changes to the database. This powerful feature can be used to view and reconstruct lost data that may have been deleted or changed by accident. Developers can use this feature to build self-service error correction into their applications, empowering end-users to undo and correct their errors without delay, rather than burdening administrators to perform this task. Flashback Query is extremely simple to manage, as the Oracle server automatically keeps the necessary information to reconstruct data for a configurable time in the past. This feature is unique to Oracle – DB2 has no ability to query data at a point in time.

Transaction Recovery

Oracle LogMiner is a powerful audit tool that enables a DBA to find and correct unwanted changes. Its simple SQL interface allows searching by user, table, time, type of update, value in update, or any combination of these. It supports a multi-versioned dictionary that gives it the ability to track a database object even as the object goes through DDL-related structural changes. LogMiner provides SQL statements needed to undo the erroneous operation. Additionally, the GUI interface graphically shows the change history. It is much easier and quicker than restoring a backup to perform a recovery. DB2 does not have the ability to mine logs to recover erroneous or malicious transactions.

Point-in-Time Recovery

Oracle allows full tablespace point-in-time recovery with no limit on the operations that can be backed out. DB2 does not allow point-in-time recovery of a tablespace if there has been a DDL operation performed on or in the tablespace. Additionally, DB2 imposes a *minimum recovery time*[16], which is the earliest point in time for which point in time recovery can be done for a tablespace. Events that cause the DB2 minimum recovery time for a tablespace to be updated include:

- System catalogs are changed as a result of altering a table in the tablespace
- Constraints are turned off for a table in the tablespace
- A column of a table in the tablespace is added/altered
- A table/index in a tablespace is added/dropped
- Tablespace attributes such as bufferpool and prefetch size are altered

[16] Ref (a) DB2 Technote 1006525, http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?last_page=list.d2w&fn=1006525, (b) DB2 Data Recovery and High Availability Guide and Reference: Version 8, Chapter 4 - Rollforward Recovery

For a dropped table to be recoverable in DB2, the table space in which the table resides must have a DROPPED TABLE RECOVERY option turned on. Besides, it requires the dropped table to be exported and reloaded – a process that is very inefficient for very large tables.

ORACLE VS. DB2 – ADDRESSING PLANNED DOWNTIME

Addressing System Maintenance

As business needs change, system changes may also be required. For example, business growth often entails growth in data processing volume. This may translate into a requirement for additional processing power through hardware upgrades of disks, memory, CPUs, nodes in a cluster, or entire systems. Though DB2 allows the addition of disk space dynamically, Oracle is unique in the ability to change any system resource dynamically, as shown in the following table.

Table 6: Addressing System Maintenance – Oracle vs. DB2

Addressing System Maintenance	Oracle	DB2
Add a node to a cluster online	Yes	No
Extensive support to adjust memory online	Yes	No
Helpful advisories on memory management	Yes	No
Most configuration parameters may be modified online	Yes	No

The following sections provide further details on these capabilities provided by Oracle.

Adding a Cluster Node

Data partitioning in a shared-nothing environment makes adding new servers to a cluster time consuming and costly, because redistribution of partitioned data according to the new partitioning map is required. Here's what a DBA or System Administrator has to do to add a node to a DB2 database with the Partitioning option:

- Add hardware
- Configure a new partition (set partition-specific parameters, etc.)
- Restart the database (i.e. shut down and restart all nodes)
- Re-distribute the data to spread it across a larger number of partitions

Re-distributing the data in a DB2 system with the Partitioning option involves DBA work and downtime for the database. There are three ways to redistribute this data, but all of them interrupt database operations:

- *Redistribute existing nodegroup* – The data in the nodegroup is inaccessible until the command completes. The time taken for the command to complete grows with the amount of data to be redistributed. Beacuse this is an in-place redistribution, the operation is logged and prone to running out of log space.
- *Create new nodegroup* – Replicas of the old table are created in the new nodegroup. This requires sufficient space to store the data stored in the old nodegroup. Even with this space, the data in the old nodegroup will not be available for modification while it is being copied to the new nodegroup. Further, all dependencies such as indexes, triggers, constraints and privileges will need to be recreated.
- *Piecewise redistribution* – This is similar to the first option except that data in the hash buckets can be redistributed one at time. This spreads the re-distribution over a longer period of time controlled by the user, thereby limiting the window of unavailability at any given time. This is an immense management burden, and will require that the database be taken off-line for a non-trivial amount of time.

"Amazon migrated a very large database from a 14 processor non HP machine to a 2 node HP cluster with 32-CPUs each. This was accomplished with less than five minutes of customer visible downtime."
– Matt Swan, Director of DB Services, Amazon.com

Consider, on the other hand, the management tasks needed when you add a node to RAC:

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

That's it! No data re-partitioning, no offline maintenance – just a seamless scale-up. RAC allows nodes to be added without interrupting database access.

Configuring Memory

Oracle allows dynamic resizing of all memory structures without shutting down and restarting the database. IBM, on the other hand, has begun to offer the ability to add a new buffer pool, alter the size of an existing buffer pool and drop a buffer pool without requirement to restart the database, only with version 8.1 of DB2.

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, using patent-pending simulation algorithms to generate the accurate advisory data out-of-the-box with absolutely minimal overhead. 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. DB2 does not offer such advisories, which implies DB2 administrators either rely on their experience or use empirical approaches to tune the database performance.

For further details, please refer to the Oracle white paper titled "*Technical Comparison of Oracle9i Database vs. IBM DB2 UDB: Focus on Manageability*".

Configuring Parameters

Almost all of Oracle's parameters can be dynamically modified. Prior to version 8.1, DB2 had only 2 dynamic parameters – dft_monswitches (default database system monitor switches) and mincommit (number of commits to group) and most changes to DB2 parameters required either application or database restart. Version 8.1 has made some improvements in this regard. However, considering that almost 60% of the total number of its Database Manager Configuration parameters and Database Configuration parameters still cannot be configured online[17], DB2 continues to lag behind Oracle in this aspect of High Availability.

Addressing Data Maintenance

As business requirements and processes change, the underlying data has to be maintained and transformed to suit the new environment, and done in such a way that there is minimal or no disruptions to the business. Maintaining, re-defining and transforming the data that supports a business is a critical activity for any DBA – this may be required unexpectedly with new business conditions, or this may even be a regularly scheduled activity. As shown in the following table, Oracle, in contrast to DB2, provides a suite of capabilities in this regard.

Table 7: Addressing Data Maintenance – Oracle vs. DB2

Addressing Data Maintenance	Oracle	DB2
Scalable data maintenance through rolling windows	Yes	No
Resumable space allocation	Yes	No
Consistent table export without locking	Yes	No
Online table load	Yes	No
Reorganize individual indexes	Yes	No
Add, rename, merge columns online	Yes	No
Create/rebuild/coalesce indexes online	Yes	No
Extensive set of online redefinition capabilities	Yes	No

[17] *DB2 Information Center, "Configuration Parameters", http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8document.d2w/report?fn=r0005181.htm*

The following sections provide further details on these differentiators.

Scalable Maintenance

DB2's lack of range and list partitioning makes it difficult to perform scalable maintenance. Oracle's *rolling window* time-based partitioning makes it easy to perform maintenance on old partitions while keeping new ones online. In addition, DB2 cannot modify the partitioning for a table such as splitting a partition, or merging two partitions, or exchanging a table and a partition. Instead, to change data partitioning, DB2 relies on time-consuming methods that lock down the data as described in the "Adding a Cluster Node" section.

DB2 cannot resume an operation that runs out of space while executing. Oracle's resumable space allocation feature allows space issues to be fixed and the operation to continue transparently.

While both Oracle and DB2 support restartable data loading, only Oracle can perform a consistent export without locking the entire table. DB2 cannot take a consistent export of a table without locking the entire table. Oracle can use multiversioning to make a consistent export of a table or set of tables. DB2 has to use an isolation level of repeatable read to get a consistent view of a table. The side effect of this is locking of every block the table scan touches. IBM calls them *innocent bystander locks*.

DB2's export utility can only handle one table at a time and does not export all the metadata associated with the table. In addition, the DB2 export utility is only restartable if the entire table is locked during export.

Online Data Redefinition and Data Reorganization

IBM, with its DB2 v8.1 release, has created a lot of press regarding its "new online utilities" (e.g. "*DB2 also provides customers with new online utilities, enabling them to perform tasks, such as table reorganization, index maintenance and database loads online. These capabilities provide customers with more flexibility in managing online environments to ensure the highest level of availability for critical business information.*[18]"). A closer look at these new capabilities will show how DB2 really stacks up with respect to Oracle in terms of online data redefinition capabilities.

Basically, most of the *online* enhancements in DB2 v8.1 are related to online reorganization of fragmented data. For example, the REORG INDEXES command "*reorganizes all indexes defined on a table by rebuilding the index data into unfragmented, physically contiguous pages*". Similarly, the REORG TABLE command

---

[18] *IBM press release, dated July 22, 2002, "IBM Simplifies Data Management With New Self-Managing DB2 Software"*

> *"Oracle's efforts to make its database easier to manage has helped us in enhancing administrator productivity and reducing administrative costs. By extensively using Locally Managed Tablespaces, our DBAs don't have to worry about space fragmentation issues any more."*
> *– Matt Swan, Director of DB Services, Amazon.com*

"*reorganizes a table by reconstructing the rows to eliminate fragmented data, and by compacting information*"[19]. Note that the availability of tools to avoid fragmentation may make one wonder why fragmentation is an issue for a database that is supposed to be self-managing. This contrasts with Oracle's locally managed tablespaces, which, along with automatic segment space management, eliminates fragmentation altogether, improves ease of space management and space utilization, and improves parallel data manipulation language (DML) performance.

IBM claims that some of the major *online* enhancements in DB2 v8.1 are:

- Online table load
- Online table reorganization
- Online index reorganization
- Dynamic configuration parameters
- Online buffer pool management
- Incremental maintenance of materialized query tables during load append

However, these DB2 *enhancements* have already been available with Oracle for a while. For example, the "*Incremental maintenance of materialized query tables*" concept has been available with Oracle through the Fast Refresh feature of materialized views since Oracle8*i*! Besides, some of these online features[20] are not really online in the way one would expect. For example, for online table load, while the enhancement is an improvement over the bad restriction that the whole tablespace is locked, it still implies that the table itself may not be available to users during the load operation, which begs the question – what is *online* in this process? Similarly, online table reorganization is allowed only on tables with type-2 indexes and without extended indexes.

While online index maintenance is a good thing (first available in Oracle8), it's not good if it is required, as by DB2, to remove pseudo-deleted type-2 index entries that couldn't actually be deleted during transaction processing workload. In addition, DB2 does not offer the capability to re-organize a single index at a time – IBM documentation indicates that all indexes on the same table must be re-organized together as a set, through the REORG INDEXES ALL FOR TABLE command, which again violates the commonly understood principles of *online* operations.

Oracle, in contrast, supports the highest degree of maintenance while data is available and accessible to users. Maintenance operations such as schema changes, data and index reorganizations, can all be done without impacting data availability.

[19] *DB2 Command Reference: Version 8, Chapter 3 – CLP Commands*
[20] *IBM DB2 Universal Database, What's New: Version 8, Chapter 4 – Availability enhancements*

The following list summarizes the key online data reorganization and online data redefinition capabilities of Oracle that improve data availability, database performance, response time and disk space utilization:

"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 Swan, Director of DB Services, Amazon.com

- Create/rebuild indexes online
- Coalesce indexes online, in-place, without additional storage
- Modify the physical attributes or storage parameters of a table
- Move a heap table or IOT to a different tablespace
- Add support for parallel queries
- Add or drop partitioning support
- Recreate a heap table or IOT
- Change a heap table to IOT and vice versa
- Analyze validate structure online
- Online table redefinition without a primary key
- Add, drop, or rename columns in a table
- Rename constraints
- Transform data in a table

DB2 cannot perform table redefinition online, including even simple changes to tables. For example, DB2 has no ability to add a constraint to a table online – a table lock is required to add a constraint. Oracle's multi-versioning technology avoids locking the table to add a constraint. Similarly, unlike Oracle, DB2 cannot convert a non-partitioned table to a partitioned table online, cannot change a table structure online and cannot transform data online.

DB2 does not address schema changes, the biggest cause of maintenance downtime, and in fact, still does not even support DROP or RENAME column operations – either online or offline. So this latest release still does not help DBAs who need to upgrade their data models.

Oracle can reorganize and redefine table data, and add, rebuild, or defragment indexes, all without exclusive access. Therefore user operations can continue as normal. As maintenance windows start to evaporate, Oracle ensures that all maintenance operations – planned or otherwise – occur while users remain online. Without a doubt, Oracle is still several years ahead of DB2 in its rich and comprehensive support of online data reorganization and redefinition.

CONCLUSION

Recognizing the high availability challenges every e-business faces, Oracle provides comprehensive, unique, powerful, and simple-to-use capabilities that protect businesses against all forms of unplanned downtime, including system faults, data corruption, disasters, and human errors. Oracle achieves this in an environment where the downtime that occurs during planned maintenance activities is also minimized.

Oracle's High Availability solutions are a compelling choice for any business because they are all well-integrated with each other – whether it's RAC, or Data Guard, or RMAN, etc. This saves customers time, money and system/people resources – factors that are extremely critical in today's economy. Oracle has gone one step further by publishing best practice guidelines for configuring a High Availability solution through its Maximum Availability Architecture framework, and making it available for its customers (ref. http://otn.oracle.com/deploy/availability/htdocs/maa.htm for details). The long list of Oracle customers (ref. http://www.oracle.com/customers) who have embraced its High Availability solutions is a testimonial to Oracle's unparalleled technical leadership in this area.

In contrast to Oracle, DB2 offers a very basic set of backup and recovery capabilities and lacks the completeness and depth of High Availability functionality required by most e-businesses today. In fact, DB2 is behind even Microsoft SQL Server when it comes to High Availability. The v8.1 release of DB2 is essentially an attempt by IBM to catch up with Oracle's technical superiority in this area. Readers are advised to refer to the Gartner Inc. report titled "*Most Enterprises Shouldn't Rush to IBM's DB2 Version 8*" dated July 2002, for an analyst perspective on DB2 v8.

Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on High Availability
February 2003
Author: Ashish Ray
Contributing Authors: Juan Loaiza, Laurence Clarke, Madhu Reddy, Robert McGuirk, Ron Weiss, Tammy Bednar, Vivian Schupmann

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