

Technical Comparison of Oracle9i Database vs. IBM DB2 UDB: Focus on High Availability

An Oracle White Paper

February 2002

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

Executive Overview.....	3
Introduction	3
Detailed Comparison Of Capabilities That Address Unplanned Downtime	4
System Failures	4
Predict Average Recovery Time.....	4
Shorten Worst Case Recovery Time	5
Minimize Time to Resume Full Capacity	6
Increase Fault Tolerance with Real World Clustering	6
Data Failures and Recovery	8
Comprehensive Backup and Recovery Capabilities.....	8
Efficient VLDB Backup and Recovery.....	9
Maintenance Reduction and Failure Isolation With Partitioning	10
End to End Data Integrity.....	11
Data Center Disasters.....	11
Oracle Data Guard Ensures Data Protection Automatically.....	11
Human Error Protection.....	13
Data Recovery.....	0
Transaction Recovery.....	14
Point in Time Recovery.....	14
Detailed Comparison Of Capabilities That Address Planned Downtime.....	15
System Maintenance.....	15
Adding a Cluster Node.....	15
Adding or Removing Memory.....	16
Database Maintenance	16
Scalable Maintenance.....	16
What is 'Online'.....	17
Online Redefinition.....	0
Conclusion	0

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

EXECUTIVE OVERVIEW

Increased reliance on technology introduces new challenges for businesses. Should applications become unavailable, entire businesses may halt. Revenue and customers may be lost. Penalties incurred. Bad press can have a lasting effect on both customers and stock prices. Providing continuous data availability is essential for today's businesses.

Over recent years there have been various efforts to quantify the revenue 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 99.9 percent data availability can cost a company over \$5 million dollars a year.

Oracle9i 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 server failures, data failures, disasters, human errors, system and database maintenance operations. IBM DB2 UDB provides only very basic functionality for both high availability and data protection.

In an INPUT cost of ownership study¹, Oracle is found to be 28% cheaper than DB2 UDB when the cost of downtime is taken into consideration. Oracle is also the database that powers such businesses as Amazon.com and eBay, which depend on Oracle database's reliability to provide continuous service to their large customer base.

INTRODUCTION

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

¹ Buyers' Guide to Database Servers based on Cost of Ownership and Effectiveness, INPUT, 2000

Unplanned Downtime

Unplanned downtime due to server failure, data failure, site failure and human error are unfortunately an all too common challenge for businesses today.

According to *Disaster Recovery Journal*, disasters due to flood and fire do happen, however, 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 error and system hardware related failures are more likely causes of failure, accounting for 36% and 49% of unplanned downtime respectively.

Planned Downtime

While unplanned downtime is a bit of an unknown quantity, planned downtime is much more within the control of IT departments and service providers. Planned downtime for upgrades, data and index re-organizations, for example, occur within every business. Some businesses plan ahead and perform regular scheduled maintenance operations and others tend to perform maintenance on an ‘as needed and when required’ basis. Planned downtime is still downtime and incurs a cost. Your challenge lie in ensuring planned downtime is kept to a bare minimum.

Oracle9i Database is designed to address the causes of unplanned and planned downtime. In the following pages we take each high availability challenge and compare how Oracle9i Database and IBM DB2 UDB² address the challenge.

In the rest of this document, Oracle9i Database will be referred to as Oracle and IBM DB2 UDB will be referred to as DB2.

DETAILED COMPARISON OF CAPABILITIES THAT ADDRESS UNPLANNED DOWNTIME

Server Failures

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

Predict Average Recovery Time

To control the time to recover from system failures, Oracle allows the *Mean Time To Recover* (MTTR) to be directly specified via a dynamic parameter, FAST_START_MTTR_TARGET. Oracle continuously estimates the recovery

² 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.

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 log files filled between checkpoints. The DBA has to then guess at how this translates into actual recovery time.

Because there is additional overhead to frequent checkpointing, Oracle provides real-time advice on the cost of the target MTTR through the `v$instance_recovery` dynamic view. With DB2, it is impossible to determine the runtime cost of adjusting the SOFTMAX parameter.

Oracle also provides an advisory that simulates to 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 IO overhead. This takes the guesswork and risk out of configuring for fast 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.”³

Shorten Worst Case Recovery Time

Oracle’s crash recovery time is immune to long transactions. 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. Oracle does not wait until all transactions have been rolled back like DB2 does. Instead, transactions are rolled back in background while new user transactions update the database. 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. DB2’s crash recovery is severely impacted by long transactions. DB2 users cannot access the database until *all* active transactions are rolled back. Therefore, crash recovery time is dependent on the longest running transaction. In addition, since DB2 cannot switch into a log that contains uncommitted transactions, recovery time is further lengthened by long transactions.

³ *DB2 UDB v7 Administration Guide: Planning*, page 219

Minimize Time to Resume Full Capacity

Oracle brings advanced capabilities that minimize the time to resume full capacity after a failure. Oracle's shared cache clustering 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 this large a buffer cache can take a very long time. With Oracle Real Application Clusters, failover happens 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

Oracle and IBM both offer a clustered database. Oracle provides a shared cache clustering solution called Real Application Clusters (RAC). IBM provides a shared nothing clustering database called DB2 Enterprise-Extended Edition (DB2 EEE). DB2 EEE clustering model depends on data partitioning. Each node in a DB2 EEE cluster houses one or more database partitions. In the event of an unexpected node failure, both RAC and DB2 EEE can transparently recover the database. However, we can expect the recovery times to be faster with RAC due to the capabilities described above plus the following:

- DB2 EEE 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.
- 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 EEE, the underlying cluster manager determines which of the surviving nodes takeover the disks belonging to the failed partition(s). To avoid load skew, DB2 EEE must be configured such that each surviving node takes over 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 partitions⁴. If there are

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

⁴ Nomani, Aslam and Pham, Lan IBM, DB2 Universal Database for Windows High

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 EEE, 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 IO. This surprising fact is inherent in the way DB2 EEE 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 since DB2 has no way of knowing which partition holds the data.⁵

In DB2 EEE, the interprocess 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 EEE will force the application to run on twelve partitions.

IBM will sometimes recommend replicating a table across all nodes to work around this fundamental scalability issue of DB2 EEE. For example, they might recommend replicating the customer table across all nodes. This is clearly not a scalable or efficient solution since the amount of disk space consumed by the

Availability Supporting Using Microsoft Cluster Server - Overview, TR-74.176 May 2001

⁵ See DB2 Performance Administration Guide V7.2, Planning: Chapter 18 See also DB2 V7.2 Application Development Guide: Chapter 18

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 since any node slowdown or crash will impact modifications across the entire system.

Furthermore, Oracle supports automated failover on all clustering platforms. DB2 failover support for HP-UX, Linux, and the Dynix/ptx operating systems requires a manual restart⁶ of the failing node on another node that has access to the failing node's disks.

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 overall.

Data Failures and Recovery

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.

Comprehensive Backup and Recovery Capabilities

Both Oracle and DB2 can perform basic online and offline backup and recovery. Though it is feasible to determine and implement a backup plan in advance, it is difficult to anticipate all recovery scenarios. Oracle's comprehensive backup and recovery capabilities extend beyond the basic functionality provided by DB2. As a result, Oracle can handle almost any backup and recovery requirement.

Oracle allows backup information to be stored in an independent repository. 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.

⁶ "At this time, DB2 failover support for HP-UX, Linux and the PTX operating system is a manual process requiring you to restart the failing node manually on another node that has access to the failing node's disk." [DB2 EEE v7 Quick Beginnings for Unix Guide](#)

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 writers during this operation.

Archived log files can become damaged. Oracle allows damaged log files to be scavenged using the LogMiner utility, 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.

Efficient VLDB Backup and Recovery

Very Large Databases (VLDBs) such as data warehouses require efficient backup, restore, and recovery methods. Oracle offers innovative technologies such as block-level media recovery, read-only tablespaces and transportable tablespaces, which satisfy this requirement.

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, remains 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 backups through the use of read-only tablespaces. Backup of read-only tablespaces only needs to occur once. DB2 does not support read-only tablespaces, thus tablespaces must be backed up often since they cannot be placed into read-only mode.

Another time saving feature Oracle provides through the Oracle Recovery Manager (RMAN) utility is resumable backup and restore operations. With Oracle, these operations can be restarted from the point of failure. Since DB2 has no such capability, problems during backup or restore means time lost as the entire operation needs to start 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.”⁷

⁷ *IBM DB2 High Availability and Recovery Guide and Reference, version 7, page 103*

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 so⁸.

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 up into smaller 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 partition affected needs to be recovered, reducing recovery time, leaving other unaffected partitions online during partition recovery process. This increases overall data availability.

Though both Oracle and DB2 EEE 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 impacted. With hash partitioning, the database server determines the data placement while with range and list partitioning, there is user control over data placement. For example, if the data is 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 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', also finds use frequently in a data-warehousing environment. 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

⁸ "DB2 now supports incremental backup and recovery (but not of long field or large object data)." *IBM DB2 High Availability and Recovery Guide and Reference, version 7*, page 26

restoring older data in background. . . Again, due to its limited partitioning capability, DB2 cannot support ‘rolling window’ data management.

In addition, Oracle also implements global partitioned indexes to isolate index failures while maintaining query efficiency. DB2 EEE only supports local indexes, causing queries to be broadcast across all partitions unless the partitioning key is specified in the query predicate. For more details on performance issues with DB2 EEE, see the Oracle white paper titled “Technical Comparison of Oracle Real Application Clusters vs. IBM DB2 UDB EEE”.

End to End Data Integrity

Oracle’s Hardware Assisted Resilient Data (H.A.R.D.) initiative allows storage vendors to validate Oracle blocks before writing them to disk. EMC already has a product that does this in the Symmetrix RAID devices. DB2 has no way to prevent corrupt blocks from being written to disk.

Data Center Disasters

Oracle Data Guard Ensures Data Protection Automatically

Oracle has invested 50 people years in development of Oracle Data Guard, which offers the most complete and robust disaster recovery solution in the industry. Oracle Data Guard provides:

- Protection from Human Error Corruptions and Disasters
- Zero Data Loss Protection
- Integrated GUI-Based Management Framework

Oracle Data Guard empowers customers to survive disasters of many forms. Data Guard automates the complex tasks and provides the monitoring alerting and control mechanisms to maintain a standby operation. Plus, 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 depends on user written log transfer callouts. DB2’s user created standby database “solution” is less robust and costs more to implement.

Automated Standby Environment

Oracle has GUI and command line driven monitoring of log shipping and standby status. The Oracle Data Guard Manager provides a consolidated view of the primary database and all its standby databases. The instantiation wizard generates the standby database. The Oracle Enterprise Manager is leveraged for alerting and discovery services.

Oracle supports post failure automated recovery. Oracle will fetch missing logs caused by network outage and fill gaps as required. Oracle will also automatically detect damaged logs and will retrieve replacements from primary database. DB2 has no built-in method to catch up following a network outage or handle damaged logs. DB2 customers have to manually handle reshipping of logs following a disconnection of network or a problem with the standby database.

Oracle also provides automated planned switchover and unplanned failover with a single command. By facilitating failover and switchover activities, the possibility of administrative errors is dramatically reduced. In contrast, DB2 has no tools for creating or monitoring standby configuration; along with log shipping and monitoring, DB2 administrators have to code switchover and switchback scripts themselves.

Zero Data Loss Protection

The same automated log transport services are used by both physical and logical standby database components in Oracle Data Guard. Traditionally, archive logs are shipped from the primary to the standby as soon as they are created. 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. With DB2, zero data loss is only possible with third party software and disk mirroring products. This means higher cost due to additional hardware/software investment, and higher complexity due to integration of disparate technologies.

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 to for an acknowledgement from the standby, and therefore does not guarantee zero data loss. However, 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

Before human errors such as logical data corruption or accidentally dropped tables are propagated to the standby, Oracle Data Guard also allows the specification of a delay for application of the redo log data once it arrives at the standby site. Immediate application means faster recovery and a more up-to-date standby database. However, delaying the application of logs allows administrators to failover to the standby and resume services on the standby.

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 since DB2 does not provide a read-only database capability.

In addition, because Oracle's logical standby databases are open for 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 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.

Oracle RMAN allows backups to run on the standby database. This offloads the backup operation from the production database, reduces resource contention, and boosts performance. On the other hand, DB2 standby databases cannot be backed up while logs are being applied.

Double Failure Protection

Oracle Data Guard can use two standby databases with synchronous log data shipping to ensure that no transaction data will be lost, even if there are two simultaneous or correlated failures. DB2 has no such protection mechanism.

Automated Standby for Clustered Primary

With Oracle, the primary database can be a RAC cluster and/or the standby database can be a RAC cluster, all protection modes work. Automated log shipping and recovery are available for all configurations.

IBM never discusses standby database configuration 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 all of them?

Human Error Protection

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

“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 data at a point-in-time in the past and compare with current. 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 will automatically keep 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’s LogMiner 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. LogMiner provides SQL statements needed to undo the erroneous operation. Additionally, the GUI interface shows graphically 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 time⁹, which is the earliest point in time for which point in time recovery can be done for a tablespace. Some things that cause the DB2 minimum recovery time for a tablespace to be updated include:

- When the system catalogs are changed as a result of altering a table in the tablespace
- When constraints are turned off for a table in the tablespace
- When adding or altering a column of a table in the tablespace
- When creating/dropping a table/index in a tablespace
- When altering tablespace attributes such as bufferpool and prefetch size

⁹ 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

DB2 has a special ‘dropped table recovery’ mode to tablespace point in time recovery. However, it requires the dropped table to be exported and reloaded and is very inefficient for very large tables.

DETAILED COMPARISON OF CAPABILITIES THAT ADDRESS PLANNED DOWNTIME

System Maintenance

As business needs change, system changes may also be required to meet the business needs. 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* hardware resource dynamically.

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 Sysadmin has to do to add a node to a DB2 EEE database:

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

DB2 EEE data redistribution

Redistributing the data in a DB2 EEE system 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*
- Create new *nodegroup*
- Piecewise redistribution

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. Since this is an in-place redistribution the operation is logged and prone to running out of log space.

Amazon migrated a very large database from a 14 processor non HP machine to a 2 node HP cluster with 32-CPU's each.

This was accomplished with less than five minutes of customer visible downtime

- Matt Swan
- Director of DB Services
- Amazon.com

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 a time. This spreads the redistribution 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 off-line for a non-trivial amount of time.

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.)

And 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.

Adding or Removing Memory

Oracle allows dynamic resizing of all memory structures without shutting down and restarting the database. In addition, Oracle provides buffer cache sizing advisory that shows how much reduction in IO can be achieved for various caches sizes. DB2 cannot dynamically add or remove shared memory – cannot resize buffer cache or package cache.

Oracle dynamically tunes sorts and hash joins based on concurrency and amount of available memory. Administrators must statically specify DB2 sort and hash memory then restart DB2 or the application for the change to take effect.

Almost all of Oracle's parameters are dynamically modifiable. DB2 has only 2 dynamic parameters – *dft_monswiches* (default database system monitor switches) and *mincommit* (number of commits to group). Most changes to DB2 parameters require either application or database restart.

Database Maintenance

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 “DB2 EEE data redistribution” 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’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. Oracle also supports restartable backup and restore/recovery.

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’.

What is ‘Online’

Oracle says an operation is online if all business transactions continue while the operation takes place – this is the conventional and common sense definition. IBM has made up its own definition of *online* so that it can claim it has online operations. IBM says an operation is online if *some* business transactions continue to work – *even if most stop*.

IBM says:

- DB2 can create index online because reads can continue, even though updates are locked out.
- DB2 can add a node online because reads can continue, even though updates are locked out.
- DB2 can change the buffer pool online because the configuration file can be changed, even though the buffer pool size change doesn’t happen until the database is restarted.

Online Redefinition

Oracle 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. Schema changes are a regular occurrence and reorganizations are essential to ensure optimum data access performance is maintained.

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.

DB2 cannot even perform simple changes to tables online. 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.

Oracle can reorganize table data, add, rebuild, or defragment indexes, all without exclusive access, therefore user operations can continue as normal. The only reorganization operation DB2 can perform online is defragmentation of an existing index.

As maintenance windows start to evaporate, Oracle ensures that all maintenance operations – planned or otherwise – occur while users remain online.

It is fashionable to talk about the number of ‘nines’ of availability that a system provides. For example, a ‘five nines’ system is available 99.999% of the time. To achieve five nines availability, a system can only be unavailable for five minutes in a year. To achieve four nines, a system can only be unavailable for 50 minutes in a year. A single index creation operation on a large table can take several hours to execute. If index creation cannot be done online, then that single operation will drop the availability of the system from five nines to three for the year. This example shows that online indexing operations and online table redefinition are crucial for any 24 by 7 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, VeriSign

VeriSign offers validation of credit card numbers for over 50,000 merchants

CONCLUSION

DB2 offers the very basic set of back and recovery capabilities but lacks the completeness and depth of data protection required by most businesses today. In fact, DB2 is behind even Microsoft SQL Server when it comes to high availability.

In recognizing the high availability challenges every business faces, Oracle provides comprehensive, unique, powerful and simple to use capabilities to protect against most forms of unplanned downtime, including system faults, data corruption, human error and disaster. And, achieves this in an environment where the need for planned downtime is marginalized.



Technical Comparison of Oracle Database vs. IBM DB2 UDB: Focus on High Availability
February 2002

Author: Jenny Tsai

Contributing Authors: Juan Loaiza, Ron Weiss

Contact: Ashish Ray

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

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

www.oracle.com

Oracle Corporation provides the software
that powers the internet.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2002 Oracle Corporation
All rights reserved.