Technical Comparison of Oracle Real Application Clusters 11*g* vs. IBM DB2 v9.5 for Linux, Unix, and Windows

An Oracle White Paper
August 2008

ORACLE

Technical Comparison of Oracle Real Application Cluster 11*g* vs. IBM DB2 v9.5 for Linux, Unix, and Windows

Technical Comparison of Oracle Real Application Cluster 11*g* vs. IBM DB2 v9.5 for Linux, Unix, and Windows

INTRODUCTION

As more enterprises pursue grid computing to gain economic efficiencies and flexibility to adapt to changing business conditions, they will be faced with various technology decisions. As grid computing gains momentum and as the underlying database technology is evaluated for its ability to achieve the benefits of grid computing, careful consideration of vendors' offerings is essential.

This paper provides a technical comparison of Oracle Database Real Application Clusters 11*g* with IBM DB2 v9.5 for Linux, Unix, and Windows. This paper presents why Oracle's Real Application Clusters is the preferred solution over IBM's DB2 v9.5 in terms of performance, flexible scalability, resource utilization, manageability, and availability. Note IBM offers a distinct database product called DB2 v9.x for z/OS. This paper only applies to DB2 v9.5 for Linux, Unix and Windows and does not apply to DB2 v9.x for z/OS.

Enterprise Grid Computing

Many definitions of Grid Computing exist in the marketplace. At a high level, Enterprise Grid Computing provides the capability to pool various IT resources to act as a single, integrated, connected entity which can be automatically provisioned as needed.

At the core, a Grid is an interconnected structure of components providing application services and integrating a variety of resources like databases, application and database servers, storage systems and networks. The objective is to implement the integration in such a way that that all resources can be managed as a single system image and provisioned on demand based on the service requirements of each application.

Two core tenets uniquely distinguish grid computing from other types of computing:

- Virtualization: individual resources (e.g. computers, disks, application components and information sources) are pooled together by type; then made available to requesting applications through an abstraction. This means that you should not be concerned with where your data resides, or which server processes your request or which physical disk stores the data.
- Provisioning: When resources are requested through a virtualization layer, behind the scenes a specific resource is identified to fulfill the request and then it is allocated to the requesting element. Provisioning as part of grid computing means that the system determines how to meet the specific requests while optimizing operation of the system as a whole. Provisioning in the grid is about resource allocation, information sharing, and high availability. Resource allocation ensures that all those that need or request resources are getting what they need, that resources are not standing idle while requests are going without being serviced. Information sharing makes sure that the information that users and applications need is available where and when it is needed. High availability features guarantee all the data and computation is always available.

If virtualization and provisioning are important goals for improving your computing infrastructure, Oracle Real Application Clusters 11*g* offers a more integrated, functional, performance and cost-effective solution than the IBM DB2 offering.

Enabling Enterprise Grids

Oracle RAC 11*g* provides the foundation for Enterprise Grid Computing. Enterprise Grids are built from standard, commodity servers, shared storage, and network components. RAC is a cluster database with a shared cache architecture that runs on multiple servers, attached through a cluster interconnect, and a shared storage subsystem. Oracle RAC 11*g* enables transparent deployment of a single database across a cluster of servers and provides the highest levels of availability and scalability. Nodes, storage, CPUs, and memory can all be dynamically provisioned while the system remains online. This allows service levels to be easily and efficiently maintained while lowering Total Cost of Ownership (TCO). RAC provides dynamic distribution of workload and transparent protection against system failures.

Oracle RAC is a cluster database with a shared cache architecture that overcomes the limitations of traditional shared-nothing and shared-disk approaches to provide a highly scalable and available database solution for all your business applications.

Oracle RAC 11*g* includes Oracle Clusterware, a complete, integrated Clusterware management solution available on all Oracle Database 11*g* platforms. Oracle Clusterware functionality includes mechanisms for cluster messaging, locking, failure detection, and recovery—no third party Clusterware management software need be purchased. With RAC, Oracle Database continues its leadership in innovative database technology and widens the gap from its competitors.

ENTERPRISE GRID DEPLOYMENT

The architecture used to deploy your grid environment has a major impact on your Total Cost of Ownership (TCO). One of the main differences between Oracle RAC 11*g* and IBM DB2 is the implementation of shared disk vs. shared-nothing architecture explained below:

Shared-Nothing Architecture

In a pure shared-nothing implementation, the database is partitioned among the nodes of a cluster system. Each node has ownership of a distinct subset of the data and exclusively this "owning" node performs all access to this data. A pure shared-nothing system uses a partitioned or restricted access scheme to divide the work among multiple processing nodes.

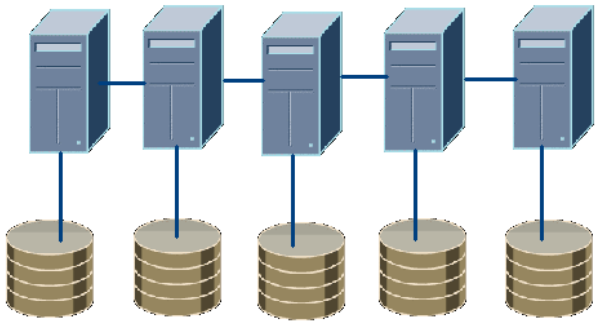

Figure 1 - Shared Nothing Architecture

In a shared-nothing architecture, parallel execution is related to the data-partitioning scheme. Performance is dependent on data being accurately partitioned.

Shared-nothing database systems may use a dual-ported disk subsystem so that each set of disks has physical connectivity to two nodes in the cluster. While this protects against system unavailability due to a node failure, a single node failure may still cause significant performance degradation.

DB2 Shared-Nothing

IBM DB2 is considered a shared-nothing architecture. However, in order to provide availability to the data, the database must be created on shared-disks. Shared-nothing refers to ownership of the data during runtime, not the physical connectivity. In IBM DB2, it is possible for the disks to be connected only to a subset of nodes that serve as secondary owners of the data in the partition. If only a

subset is used then some nodes will execute heavier workloads than others and reduce the overall system throughput and performance.

Unlike IBM DB2, Oracle RAC 11*g* requires full connectivity from the disks to all nodes and hence avoids this problem.

Shared Disk Architecture

In a shared disk database, database files are logically shared among the nodes of a loosely coupled system with each instance having access to all data. The shared disk access is accomplished either through direct hardware connectivity or by using an operating system abstraction layer that provides a single view of all the devices on all the nodes. In the shared-disk approach, transactions running on any instance can directly read or modify data by using inter-node communication to synchronize update activities performed from multiple nodes. When two or more nodes contend for the same data block, traditional shared disk database systems use disk I/O for synchronizing data access across multiple nodes i.e., the node that has a lock on the data writes the block to disk before the other nodes can access the

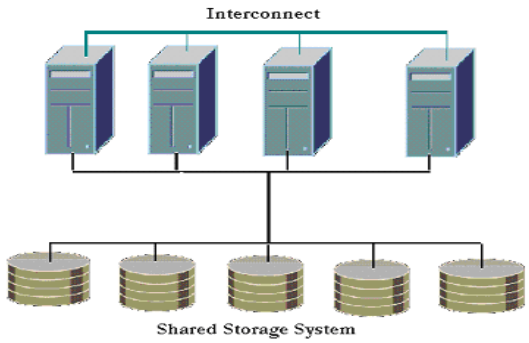

Figure 2 – Shared Disk Architecture

Shared-disk clusters provide true fault tolerance. If a node in the shared-disk cluster fails, the system dynamically redistributes the workload among all the surviving cluster nodes. This ensures uninterrupted service and balanced cluster-wide resource utilization. All data remains accessible even if there is only one surviving node.

same data block. Using disk I/O for synchronization causes significant challenges for scaling non-partitioned or high contention workloads on traditional shared-disk database systems.

Shared-disk cluster systems provide multi-node scalability and are good for applications where you can partition the data for updates. Performance will suffer significantly if the partitioning is not done correctly since the cost of maintaining the coherent caches will increase.Oracle RAC 11*g* Shared-Cache Architecture

Oracle RAC 11*g* uses Cache Fusion, a shared cache coherency technology that utilizes the high-speed interconnects to maintain cache coherency. Cache Fusion utilizes the caches of all nodes in the cluster for serving database transactions. Update operations in an Oracle RAC 11*g* often do not require disk I/O for

synchronization since the local node can obtain the needed data block directly from any of the cluster database node caches. This is a significant improvement over slower disk I/O solution and overcomes a fundamental weakness attributed to traditional shared-disk clusters.

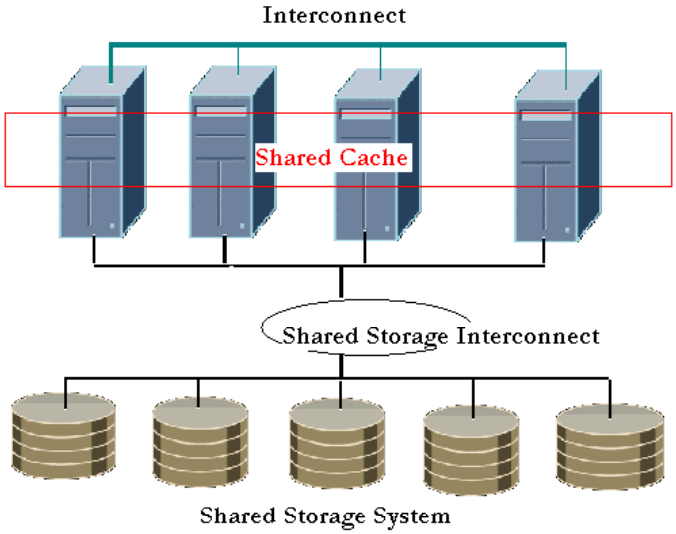

Figure 3 - Oracle RAC Shared Cache Architecture

Best Choice For Deployment

Oracle RAC 11*g* is the first and remains the only viable option for deploying your enterprise grid environment. It is easy to deploy and manage in contrast to IBM DB2. An Oracle RAC database not only appears like a single standard Oracle Database 11*g* to users, but the same maintenance tools and practices used for a single Oracle Database 11*g* can be used on the entire cluster. All of the standard backup and recovery operations work transparently with RAC. All SQL operations, including data definition language and integrity constraints, are also identical for both single instance and RAC configurations.

Migration to a RAC database from a single instance Oracle Database is easily completed through Enterprise Manager 11*g* once your cluster has been created and Real Application Clusters installed. The database files will continue to be accessed with no change. Most importantly, managing a balanced workload - adding or removing a node from the cluster does not require any data modification or partitioning.

In contrast, IBM DB2's deployment of a cluster environment from a single node environment can be a complex task. It requires migration of the database to take advantage of the additional nodes in the cluster. This means that data has to be unloaded and reloaded to disks owned by each logical node (partition).

Furthermore, partitioning is necessary whenever a node is added to the cluster. This is time-consuming, labor-intensive, and error-prone.

IBM DB2 Partitioning

The term partition is used differently in Oracle database than in IBM DB2. In DB2 terminology, the term partition refers to a logical node and the data that is owned by that logical node. Each partition has its own buffer pool, package cache etc. A DB2 partition can only access a subset of the data directly.

Optimal partitioning in an environment where an application has thousands of tables can be a complex task. While some applications change infrequently and it is possible to minimize this analysis, there are many other applications that change their application profiles frequently. Even if not all tables need to be partitioned, partitioning of the few heavily accessed tables will require extensive understanding of the application and its data access methods. Determining an optimal partitioning is non-trivial. To keep the workload balanced across all the nodes in an IBM DB2 cluster environment, re-partitioning is often required.

The Database Partitioning Feature is no longer available as an add-on Feature Pack for DB2 Enterprise, it is now only available with DB2 Warehouse Edition also known as IBM Infosphere Warehouse v9.5.1.

Nodegroups

In IBM DB2, nodes/partitions are assigned to *nodegroups*. Each tablespace is assigned to a *nodegroup*. When a table is created, it is assigned a partition key composed of one or more columns. As rows are inserted into the table, the row's partitioning value is hashed into a hash table where it is assigned to a specific partition based on the current partitioning map. The default partitioning map equally distributes data across partitions in the *nodegroup*.

Although DB2 will automatically partition tables based on primary key or the first non-long column for those tables that do not have a partition key specification, determining the partition keys for optimal partitioning of heavily accessed tables is contingent on the following competing factors:

- The partitioning key should evenly distribute data to all the logical nodes (partitions). This requires understanding of the table definitions and column usage. In some cases, even with this understanding, it is possible that the data cannot be evenly distributed.
- In joining columns you must be careful about choosing your partitioning keys to allow for co-located joins (i.e. a join that can be performed on the same node without inter-partition communication). Co-location is key for performance on DB2. In some schemas, it is not possible to co-locate all the joins. While high-speed interconnects make this less crucial, reducing message traffic is still important for cluster database performance.

- Data access patterns, (i.e. table sizes and frequency of queries used) need to be considered for determining the partitioning keys. If a table does not have the appropriate co-location columns, you can create replicated copies on each node. However, updating a replicated table impacts the performance negatively.
- Finally, any unique index or primary key must be a superset of the partitioning key. This means that application changes are required if there is a need to enforce unique constraints on columns other than the partitioning columns.

IBM DB2 Design Advisor

IBM DB2 v8 introduced a new feature called "Design Advisor" that provides recommendations and advice on how to partition the data either when creating a database or when migrating from a non-partitioned DB2 database to a partitioned DB2 database. Despite the advisory, this complex task will still need to be carried out manually by the DBA. As described above, the recommendations made by the Design Advisor may also need adjusting by the DBA if the workload changes to prevent data skew or hot spots, which affect performance.

SCALABILITY AND PERFORMANCE

Oracle Real Application Clusters with its shared cache architecture provides flexibility for scaling applications. RAC offers flexible and effortless scalability for all types of applications. Application users can log onto a single virtual high performance cluster server. The Cache Fusion technology implemented in Real Application Clusters enables capacity to be scaled near linearly without making any changes to your application. The more complex the application or the more dynamic the workload, the more appealing it is to use Oracle Real Application Clusters 11*g*.

Unlike IBM DB2, Oracle RAC 11*g* with its shared cache architecture does not require a data partitioning scheme to scale. Oracle RAC 11*g* offers cluster scalability for all applications out-of-the box — without modification.

Grids can be built from standard, commodity servers, storage, and network components. When you need more processing power, you can simply add another server without taking users offline. When you reach the limit of the capacity of your current hardware, Oracle RAC allows horizontal scalability of your grid environment by simply adding similar servers to the cluster. This makes scalability cost effective compared to replacing existing systems with new and larger nodes. In contrast to IBM DB2, adding nodes to your grid is easy and manual intervention is not required to partition data when processor nodes are added or when business requirements change.

Adding nodes

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

- Add hardware
- Configure new partition (set partition-specific parameters, etc.)
- Redistribute the data to spread it across a larger number of partitions. Redistributing the data in a DB2 ESE system involves DBA work and downtime for the affected tables.

In contrast, to add a node to a RAC cluster, the DBA or Sysadmin will only have to do the following:

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

There is no data re-partitioning. No off-line maintenance is required– just a seamless scale-up. RAC allows nodes to be added without interrupting database access.

Performance: OLTP Applications

Scalability and performance are also dependent on type of the application and the workload they run.

In an OLTP application, performance is measured by throughput (i.e., number of transactions processed in a unit of time) or response time (i.e., amount of time a given transaction will take).

In RAC, committing any transaction that modifies data is dependent on a log write on the node that is running the transaction. If the transaction needs to access data modified by other nodes in the cluster, these blocks are transferred using the high-speed interconnect without incurring any disk I/O. Through cache fusion, RAC transparently optimizes the message traffic for resolving inter-node cache conflicts.

Any transaction that modifies data in more than one partition on a DB2 ESE system must use the two-phase commit protocol to insure the integrity of transactions across multiple machines. That means you will have to wait for at least two log writes and at least one round-trip message as part of the two-phase commit process before committing a transaction. Hence, committing a transaction in DB2 takes much longer and will increase the response time. Furthermore, DB2 ESE forces the index to be equi-partitioned with the table, queries on indexes other than the partitioning key (or a prefix of the partitioning key) will result in a broadcast to all the partitions. This causes multiple partition probes for queries that do not result in partition pruning. In DB2 systems, inter-node communication is used to access data from another partition even if it has not been modified since the last access.

To reduce inter-node communication, you can create replicated copies of the table for each partition. However, replicated tables require additional space and updating them impacts the performance.

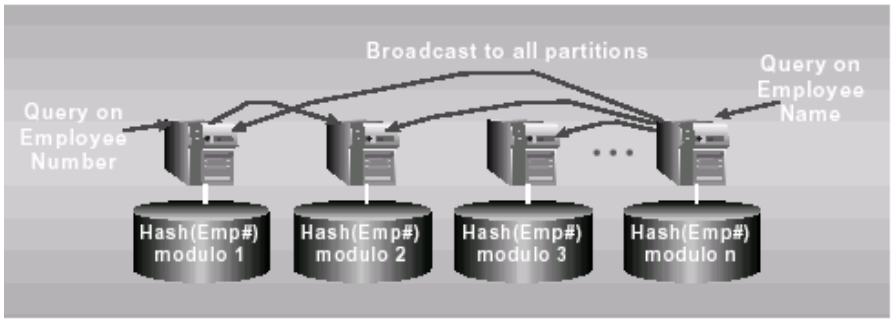

Figure 4 - Example - Partition Probes & DB2 Hash Partitioning

Performance: Data Warehousing & Business Intelligence Applications

Business intelligence operations are built on data repositories that handle geometrically increasing information requests from its global audience. It also provides customers instant information regardless of where the data resides. The workload usually consists of complex, long running queries, so the focus is that each query be executed with optimal parallelism across all the nodes in the cluster. This is to reduce overall elapsed query time.

Although DB2 the now supports multiple parallel slaves per node for a single query, with IBM's solution all nodes that own a partition of the table being accessed participate in every query. Oracle, meanwhile, gives administrators and developers more fine-grained control over the number of or database instances that participate in a query. Query parallelism within Oracle is intelligently determined based on the number of CPUs, the number of files to be accessed, the instances on which particular database services are running, and other variables. Oracle has always been able to parallelize any query, regardless of its partitioning scheme, using any degree of parallelism.

RAC's parallel query architecture is unique in its ability to dynamically determine the degree of parallelism for every query. It considers the current load on the data warehouse when choosing the degree of parallelism. RAC improves the throughput of large data warehouses executing concurrent parallel queries. It provides maximum utilization of all available resources.

Oracle's dynamic intra-partition parallel execution enables the server to spread the workload across multiple processors or nodes; this ability to localize a query will minimize inter-node communication during query execution, and will improve

query performance. Oracle is the only database to provide dynamic parallel query architecture with robust multi-node capabilities.

WORKLOAD MANAGEMENT

In DB2 ESE, transactions have to be routed based on the data partitioning; hence you have less flexibility in routing requests. Routing transactions by function in DB2 requires knowledge of the location of the data accessed by the transactions. It also results in a less flexible environment because executing the transactions on more or a fewer number of logical nodes (partitions) without data redistribution will impact performance. DB2's hash partitioning scheme requires data in all partitions to be redistributed whenever data distribution changes. This increases the maintenance time and decreases data availability as the table is locked during the data redistribution process. Similarly, when the old data is archived or deleted, all partitions need to be touched. This may interfere with regular insert operations and cause space fragmentation.

In contrast, RAC does not require the underlying data to be partitioned. Oracle RAC 11*g* offers a much more dynamic approach based on connection pooling mechanisms. Oracle Real Application Clusters includes a load balancing advisory that monitors the workload running in the database for a service and provides recommendations on where to send workload requests for the best service level. To provide the best possible throughput of application transactions, Oracle Database 11*g* JDBC Implicit Connection Cache, ODP.NET connection pools provide intelligent load balancing for applications. This feature is called Runtime Connection Load Balancing that integrates the connection pools with the load balancing advisory. When an application requests a connection from the connection pool, instead of receiving a random free connection, it is given the free connection that will provide the best possible response based on current processing activity in the cluster.

Oracle Database 11*g* Services provide a simple solution to the challenges of managing different workloads. With Oracle Database 11*g,* Services allow flexibility in the management of application workloads in a grid environment. Services allow workloads to be individually managed and controlled. DBAs control which processing resources are allocated to each Service during both normal operations and in response to failures. Users connecting to a Service are automatically load balanced across the cluster. For example, you could have multiple batch workloads with different business priorities. The highest priority workloads will be allocated the largest amount of processing resources. As priorities change, resource allocations are shifted in turn to meet business needs. If failures occur, the high priority workloads can preempt the resources of the lower priority workloads as needed.

Services are a generic feature of Oracle Database 11*g* and can be used in both single instance and Real Application Cluster (RAC) deployments. With RAC deployments, however, the full power of Services is realized. Services enable you to control which

nodes in the cluster are allocated to different Services at different times. This provides a very effective mechanism to ensure that business priorities are met by controlling how processing resources are applied to your application workloads.

Using Services, the load from the applications can be evenly spread across all servers based on their actual usage rather than allocating for peak, as is the case with IBM DB2.

Performance Management & Tuning

Once an Enterprise Grid Computing infrastructure is deployed, end-to-end performance management and self-tuning is critical to ensure the infrastructure continues to support application service level objectives. This requires a high degree of automation.

With Oracle Database 11*g*, this automation can be achieved by using the tools such as Automatic Database Diagnostics Monitor (ADDM), Automatic Workload Repository (AWR), Oracle Enterprise Manager (EM) and others. Oracle Database 11*g*'s built-in server-generated alerts and Enterprise Manager's propagation framework along with its browser-based interface provide the foundation to manage systems performance problems and database maintenance tasks. The various self-managing initiatives provided by Oracle Database 11*g*, assist in enabling automation of Oracle systems reducing manual intervention, lowering costs and providing better quality of service.

With Oracle Database 11*g*, tuning can be based on Service and SQL. This has the advantage that each business process can be implemented as a database service and each of these workloads can in turn be measured independently and automatically. It also makes the resource consumption of business transactions measurable and additionally the workloads can be prioritized depending on how important they are.

Oracle Database 11*g* Advisories

In a Grid environment self-tuning capabilities are crucial. Integrating all the components of a Grid makes it necessary to have a high level of automation. This high level of automation is supported by the various advisories in Oracle Database 11*g*.

Advisors are server-centric modules that provide recommendations to improve resource utilization and performance for a particular database sub-component. Advisors provide the DBA with the necessary context sensitive information, related to the problems that are encountered. They complement the alert mechanism. The advisors may reference historical data in addition to current in-memory data in order to produce their recommendations. They are especially powerful because they provide vital tuning information that cannot be obtained any other way.

Performance Advisors such as the SQL Tuning, SQL Access, Memory, Undo, and Segment are essential in identifying system bottlenecks and getting tuning advice

for specific areas, on probable resolution paths. The SQL advisory helps in tuning SQL statements.

In contrast, IBM DB2 v9.5 does not have a self-diagnostic component to assess its own performance, identify the root cause of problems, and recommend solutions. All that DB2 v9.5 offers for performance monitoring is Health Monitor alerts, which notify the DBA any time a problem symptom is detected. These alerts are accompanied by a recommendation about how to fix a particular problem; however, there is no infrastructure, which investigates all those symptoms and identifies the root cause of problem.

IBM DB2 Design Advisor

The Design Advisor, a new feature of DB2 v8, falls short of providing root cause analysis and a self-tuning mechanism for performance issues. It provides recommendations and advice for how to partition the data either when creating a database or when migrating from a non-partitioned DB2 database to a partitioned DB2 database. Despite the advisory, this complex task will still need to be carried out manually by the DBA. The recommendations made by the Design Advisor may also need adjusting to prevent data skew or hot spots.

HIGH AVAILABILITY

One of the challenges in designing a highly available grid environment is addressing all possible causes of downtime. It is important to consider causes of both unplanned and planned downtime when designing a fault tolerant and resilient IT infrastructure.

Unplanned Downtime

Unplanned downtime is primarily attributed to either of computer (hardware) failures or data failures. Oracle Database 11*g* provides protection against data failures due to possible storage failures, human errors, corruption and site failures. Oracle RAC 11*g* enables the enterprise to build a grid infrastructure composed of database servers across multiple systems that are highly available and highly scalable. If one or more nodes in the cluster fail, Oracle RAC 11*g* ensures that processing continues on the surviving cluster nodes; hence providing transparent protection against system unavailability. In case of a system crash, Oracle RAC 11*g* continues to provide database services even when all except one node is down.

Oracle Database 11*g* Automatic Storage Management

Through Oracle Database 11*g* Automatic Storage Management (ASM), a native mirroring mechanism to protect against storage failure is provided. By default, mirroring is enabled and triple mirroring is also available. With ASM mirroring, an additional level of data protection can be provided with the use of failure groups. A failure group is a set of disks sharing a common resource such as a disk controller or an entire disk array. ASM places redundant copies of the data in separate failure

groups to ensure that the data will be available and transparently protected against the failure of any component in the storage subsystem. In addition, ASM supports the Hardware Assisted Resilient Data (HARD) capability to protect data against corruption.

IBM DB2 relies on the underlying operating system or additional software to provide this type of protection.

Fast-Start Fault Recovery

With Fast-Start Fault Recovery, Oracle Database 11*g* automatically will self-tune checkpoint processing to safeguard the desired recovery time objective. Oracle's Fast-Start Fault Recovery can reduce recovery time on a heavily loaded database from tens of minutes to less than 10 seconds. This makes recovery time fast and predictable, and improves the ability to meet service level objectives. Furthermore, Oracle Database 11*g* provides added protection against human error through its Flashback suite of features.

Fast Connection Fail-Over

Oracle RAC 11*g* includes a highly available (HA) application framework that enables fast and coordinated recovery. Oracle RAC 11*g* provides built-in integration with the Oracle Application Server 11*g,* and other integrated clients such as Oracle JDBC, Oracle Data Provider for .NET and Oracle Call Interface (OCI).

In a RAC environment, Fast Application Notification (FAN), a highly adaptable notification system, sends UP and DOWN signals to the application mid-tier immediately so that appropriate self-correcting recovery procedures can be applied. This is much more efficient than detecting failures of networking calls; and will significantly reduce recovery time. Rather than waiting many minutes for a TCP timeout to occur, the application can immediately take the appropriate recovery action and the load will be distributed to other available servers. When additional resources are added to the cluster (I.E. an instance starts), the UP signal allows new connections to be made so the application can transparently take advantage of the added resource.

In contrast, in an IBM DB2 cluster environment, the cluster manager (e.g. HACMP) determines which of the surviving nodes will takeover the disks belonging to the failed partition(s). To avoid load skew, DB2 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.

In DB2, partitions do not share memory. As the number of partitions grows, so does memory fragmentation. This adds to the administrative tasks for managing the cluster environment. Furthermore, inter-process communication for a given workload increases with the number of partitions.

In DB2 when a node fails, the cluster manager restarts 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; hence the much longer time for database recovery.

After the database recovery, applications obtain their original response times much faster in RAC than in DB2. This is because the data and the packages needed by the application may have already been cached in the surviving nodes.

Planned Downtime

Planned downtime is primarily due to data changes or system changes that must be applied to the production system. This is done through periodic maintenance and new deployments. In a global enterprise, planned downtime can be as disruptive to operations as unplanned downtime. It is important to design a system to minimize planned interruptions.

Zero Downtime Patching and Upgrades

Oracle Clusterware and Oracle Automatic Storage Management support rolling upgrades for patches and upgrades. For these releases, the upgrade or application of a patch to the nodes in the cluster can be done in a rolling fashion with no downtime. New software is applied one node at a time while the other nodes in the RAC environment are up and operational. When all nodes in the cluster have been patched or upgraded the rolling update is complete and all nodes are running the same version of the Oracle Clusterware or ASM software. In addition, a patch or upgrade can be uninstalled, or rolled back without forcing a cluster-wide outage. Certain database patches can also be applied in a rolling fashion. Oracle will mark applicable database patches as rolling upgradeable.

Oracle Clusterware, ASM and Oracle Real Application Clusters support rolling upgrades of the operating system when the version of the Oracle Database is certified on both releases of the operating system.

CLUSTERING OUTSIDE THE DATABASE

Recently IBM has been announcing solutions with partners that enable creating a cluster of DB2 servers via middleware. The middleware will replicate data to various databases in the cluster. Queries are then balanced between the copies, and in the event of a failure, work is redirected to one of the copies. IBM Partner solutions implementing clustering outside the database include Avokia apLive and xkoto GRIDSCALE.

Oracle Real Application Clusters is integrated into the kernel of the Oracle Database. Solutions which use clustering software outside the database cannot always control or efficiently monitor the internal operation of each database, this puts it at a disadvantage compared to similar functionality inside the database.

Clustering outside the database (sometimes referred to as active/active clustering) can provide benefits in some situations. However this technique is architecturally flawed. These solutions rely on replication to create multiple copies of the data for

redundancy and for scaling queries. These replicas require each update to be repeated multiple times, reducing overall scalability for all but the most read-intensive applications. With large databases, the redundant copies of data can be expensive to maintain just in terms of disk space. Transaction latency can also be an issue when data is asynchronously replicated, allowing some applications to potentially read stale data. Lastly, the middleware controllers orchestrating these clusters are single points of failure unless multiple copies are deployed. This creates additional complexity and consumes additional resources.

Since these solutions are 3rd-party solutions, they are not covered in depth in this paper. A companion white paper, "Oracle Real Application Clusters vs. Active/Active Clustering," more fully examines these 3rd-party clustering solutions and contrasts this approach to the clustered database solution provided by Oracle Real Application Clusters.

FUNCTIONALITY COMPARSION MATRIX

Category	**Oracle Real Application Clusters** 11*g*	**IBM DB2 v9.5**
Architecture	Shared Cache	Shared Nothing – data is partitioned – partitioning scheme needs to be well understood by DBAs, developers.
Migration from single node to cluster	Oracle Database 11*g* – easily completed through Oracle Enterprise Manager function.	Complex – data must be unloaded, reloaded and partitioning scheme established.
Maintaining performance, balanced workload	Automatically occurs through features such as Automatic Workload Manager, Runtime Connection Load Balancing.	Manual effort required to evaluate workload against partitioning scheme and then re-adjust based on changes.
Adding nodes – scaling out	RAC automatically incorporates additional capacity into the pool of database resources.	Manual re-partitioning exercise requires database downtime or interruption.
Performance: OLTP	Committed transactions result in log force on that node. Blocks containing data needed for transaction transferred from other nodes – no disk I/O required.	Requires two-phase commit across all nodes. Wait for at least two log force writes and at least one round-trip message.

Performance: Data Warehousing	Evaluates load on data warehouse and intelligently distributes parallel query work across the nodes to maximize utilization while minimizing inter-node communication.	Parallelism constrained by partitioning scheme – can result in over/under-utilized resources forced by partitioning scheme.
Workload Management	Runtime Connection Load Balancing, work is routed automatically to nodes that can provide the best response time based on current cluster activity.	Transactions are routed based on partitioning scheme – less flexible, can result in unbalanced environment.
High Availability	Shared Cache, Active-Active architecture ensures processing continues as long as one node of the cluster is up. Automatic Storage Management enables data mirroring.	If a cluster node goes down, database activity halts for data partitioned to that node until standby is started and active. Relies on operating system or additional software to provide mirroring.
Planned Downtime	Minimized due to ability to apply rolling patches and upgrades (clusterware and ASM) to cluster nodes – no cluster-wide outages are forced.	Any change to partitioning or patches requires planned outages.

CONCLUSION

Compared to Oracle RAC 11*g*, IBM DB2 v9.5 has numerous drawbacks in deployment, performance, and management of an Enterprise Grid environment.

IBM DB2 shifts a lot of the implementation burden of their shared-nothing solution to its users. In contrast, Oracle RAC 11*g* with its shared-cache architecture provides key advantages for deployment and management of an enterprise grid computing infrastructure. In contrast to IBM's DB2 v9, Oracle RAC 11*g* provides:

- Flexible scalability for all types of applications
- Higher availability solution
- A single automated management entity
- Lower Total Cost of Ownership

When compared with IBM DB2 v9.5, Oracle RAC 11*g* is a better integrated, more functionally complete offering, and is an easier to deploy and manage solution for achieving the benefits of grid computing.

White Paper Title: Technical Comparison of Oracle Real Application Clusters 11*g* vs. IBM DB2 v9.5
January 2008
Authors: Bob Thome, Barb Lundhild, Randy Hietter, Zahra Afrookhteh
Version 2.3
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