Oracle9*i* Real Application Clusters

Cache Fusion Delivers Scalability

An Oracle White Paper
Feb. 2002

ORACLE

Oracle9*i* Real Application Clusters

THE BENEFITS OF REAL APPLICATION CLUSTERS

Oracle's Cache Fusion™ cluster database architecture is a new shared cache architecture that overcomes the limitations of traditional shared nothing and shared disk approaches to provide highly scalable and available database systems for e-business. This paper describes this new cluster database architecture and illustrates how full Cache Fusion in Oracle9*i* Real Application Clusters exploits modern hardware and software technologies to provide the most flexible cluster database platform. This platform provides scalability and high availability to meet the increasing needs of Oracle's e-business customers.

Real Application Clusters exploits modern hardware and software clustering technologies to provide the most flexible cluster database platform.

- The server must be highly available; end users should enjoy uninterrupted server access regardless of hardware and/or software failures.
- The server must be flexible when adding processing power as dictated by business growth; customers want to enlarge their systems incrementally. They cannot afford to invest in large servers that may be idle for long periods.
- The server must be adaptable to changing workloads; the database must provide seamless support for changing workload requirements as mandated by rapidly evolving business needs.
- The server must be easily consolidated; customers want to minimize the number of servers and also reduce maintenance. This approach is less expensive, whether managing the system in-house or by way of application service providers.

Real Application Clusters with its Cache Fusion cluster database architecture helps customers meet these demanding and evolving business needs.

WHAT IS A CLUSTER?

A cluster is a group of independent servers that cooperate as a single system. The primary cluster components are processor nodes, a cluster interconnect, and a shared disk subsystem. The clusters share disk access and resources that manage data, but the distinct hardware cluster nodes do not share memory.

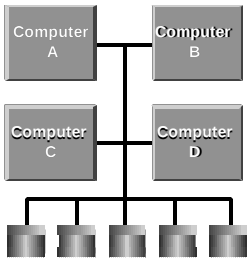

Figure 1: A Cluster Comprises Processor Nodes, The Cluster Interconnect, and A Disk Subsystem

A cluster can be made up of multiple single-processors, otherwise known as Symmetric Multi-Processor or SMP nodes. Each node has its own dedicated system memory as well as its own operating system, database instance, and application software.

Clusters provide improved fault resilience and modular incremental system growth over single symmetric multi-processors. In the event of system failures, clustering ensures high availability to users. Redundant software components, such as additional nodes, interconnects, and disks, provide high availability. Such redundant hardware architectures avoid single points-of-failure and provide exceptional fault resilience.

A cluster database lays on top of a hardware cluster. The hardware cluster architecture and the database cluster architecture may the same; or may differ.

While clustered database systems share some of these characteristics, they differ widely in architecture, implementation, and actual end-user experience. This is true for both the hardware cluster systems as well as the clustered database software that runs on these clustered systems.

A REVIEW OF CLUSTER ARCHITECTURES

In discussing different cluster architectures, it is important to distinguish between hardware clustering architectures and the cluster database architectures that run on cluster servers.

Cluster Hardware Systems

There are two main cluster hardware implementations:

- Shared disk clusters
- Shared nothing clusters: these cluster systems (IBM RS/6000 SP is the last popular shared nothing system) are less common and are diminishing in popularity, giving way to shared disk clustering.

The industry is moving toward shared disk hardware clusters. All popular hardware vendors—Sun, HP, Compaq, IBM and Intel are following this trend.

These cluster implementations differ mainly in the way they share disks. In shared disk clusters, all nodes have direct access to all disks. In shared nothing clusters, each node has access to a discrete set of disk devices. Some shared

nothing clusters resemble hybrid implementations by allowing logical access. This is enabled through an operating system abstraction layer that accesses all disk devices, for example, Pyramid RM/1000 and IBM RS/6000 SP.

So why did shared nothing clusters enjoy greater popularity in the mid-nineties and fall out of favor by the turn of the century? The answer lies in prior system limitations and rapidly emerging technology trends. Many hardware vendors are racing to establish high-performance cluster standards and provide superior cluster implementations to embrace these standards.

Advances in Cluster Hardware Technologies

Storage Area Networks solve the problem of having large disk farms directly attached to all nodes of a cluster.

Cluster hardware technologies are currently undergoing a product revolution. More application users and larger transaction volumes translate to greater disk requirements. Emerging Storage Area Networks (SAN) provide sophisticated schemes for disk connectivity. These robust storage networks circumvent the prior limitations of directly attached disks, allowing each cluster node to be connected to a large number of disk devices.

Prior to SAN technologies, customers with high capacity disk requirements used a shared nothing cluster system to provide disk connectivity. SANs remove this artificial requirement and make shared disk clusters the preferred choice for reasons of higher system availability and flexible data access. Major UNIX vendors such as Sun, Hewlett-Packard, Compaq, and IBM (both on UNIX and OS/390) have all chosen this technology. Furthermore, major NT cluster vendors such as Compaq and IBM are also adopting this strategy.

SAN architectures lay the foundation for open systems to provide mainframe-like capabilities in having CPUs for I/O. Hardware vendors provide modern disk sharing technologies in a variety of forms, such as Fiber Channel, Switch or Hub topology, and Shared SCSI.

Infiniband is an industry movement that's bringing about a common standard for system area networks that will be capable of transporting both I/O messages and cluster interconnect messages, with low latency.

InfiniBand[SM], an emerging standard for a common high-performance I/O specification, is backed by nearly all the major system vendors such as Compaq, Dell, Hewlett-Packard, IBM, Intel, Microsoft and Sun Microsystems. These standards address the connectivity of large disk farms from any server. Furthermore, InfiniBand also uses a common protocol for both I/O and interconnect messaging. This enables system area networks that carry both data and cluster interconnect messages.

While SMPs continue to grow in CPU and memory capacity, the cluster interconnect is a key technology for providing cluster scalability. Modern cluster interconnects provide extremely high bandwidth with low latencies for message traffic. These hardware interconnects provide fast communications between cluster nodes and efficient message passing as well as application load balancing. High performance interconnects come in a variety of architectures, including switch fabrics, fiber channel, and memory channel.

Commodity interconnect standards, such as Virtual Interface Architecture (VIA), are rapidly emerging among Intel-based platform and interconnect vendors. Standards such as VIA allow vendors to build high-performance clusters from standard high-volume (SHV) commodity components with superior price/performance. Major UNIX server vendors such as Sun and Hewlett-Packard are investing in high-performance interconnect technologies.

The previous discussion in this paper pertains to the cluster hardware system level. Let's investigate how the different cluster database architectures exploit these cluster hardware systems to address the business requirements of demanding e-business applications.

Traditional Approaches to Clustered Database Systems

Traditionally, clustered database systems are either shared disk or shared nothing databases. However, the evolution of some these technologies has caused the lines to blur significantly. In this section, we analyze the following database architectures:

- Shared disk approach
- Shared nothing approach
- Federated database approach

Traditional Shared Disk Database Approach

Shared disk databases map directly on top of popular shared disk hardware clusters. However, using the disk subsystem for cache coherency poses scalability challenges.

This approach is based on the assumption that every processing node has equal access to all disks (data). This assumption is valid for almost all the major hardware vendors in the industry today. In a pure shared disk database architecture, 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 any part of the database. Such systems require the use of inter-node communication to synchronize update activities performed from multiple nodes. The multi-node scalability characteristics of shared disk databases largely depend on the efficiency of these inter-node synchronization activities. Using the shared disk subsystem for cache synchronization is slow, and hence, non-optimal when applications are not partitioned for parallel access.

"Pure shared disk" is an effective strategy for clustered systems in which equal and direct access to all disks is available from every node. A single node variant of the shared disk scheme also ideally maps to SMP systems. Shared disk offers excellent resource utilization because there is no concept of data ownership and every processing node can participate in accessing all data.

Further, this approach provides unmatched levels of fault-tolerance with all data remaining accessible even if there is only one surviving node. If a node in the shared disk cluster fails, the system dynamically re-distributes the workload among the surviving cluster nodes. This ensures uninterrupted service and balanced cluster-wide resource utilization.

Traditional Shared Nothing Database Approach

In pure shared nothing architectures, database files are partitioned among the instances running on the nodes of a multi-computer system. Each instance or node has affinity with a distinct subset of the data and all access to this data is performed exclusively by this "owning" instance. In other words, a pure shared nothing system uses a partitioned or restricted access scheme to divide the work among multiple processing nodes. Data ownership by nodes changes relatively infrequently. The typical reasons for changes in ownership are database reorganizations and node failures.

Parallel execution in a shared nothing system is directly based on the data partitioning scheme. When the data is accurately partitioned, the system scales in near linear fashion. Multiple partitions are accessed concurrently, each by a single process or thread, with no provision for intra-partition parallelism.

Business requirement changes that affect the usage of the data in

Shared nothing databases have the potential to scale. However, the demands on manual re-allocation of resources may pose significant challenges for customers.

Shared nothing databases usually require that you manually reorganize or re-partition the data when changing business requirements affect the data. The system will not perform optimally without the appropriate reallocation of resources due to unbalanced system effects or hot spots. Such re-partitioning efforts are also necessary when the clustered system is scaled up by adding nodes.

Shared nothing database systems may use a dual booted disk subsystem so that each set of disks has physical connectivity to two nodes in the cluster with a notion of "primary and secondary disk ownership." Such a configuration protects against system unavailability due to node failures. It requires two node failures to render the system unavailable. While this is unlikely to occur, a single node failure may cause significant performance degradation. This is due to the fact that the one node (in some n node cluster) is now processing roughly twice the work that any other node is processing and it is on the critical path for transaction completion.

On a superficial level, a pure shared nothing system is similar to a distributed database. A transaction executing on a given node must send messages to other nodes that own the data being accessed. It must also coordinate the work done on the other nodes perform the required read/write activities. Such messaging is commonly known as "function shipping." However, shared nothing databases

are fundamentally different from distributed databases in that they operate one physical database using one data dictionary.

The Federated Database Approach

A federated database or distributed database is not a cluster database. The reason it is presented here is merely to make the necessary distinctions. A federated database comprises multiple individual database server nodes where each node runs a separate database, each with its own database dictionary. Here lies the key difference.

Federated databases are not cluster databases. They are not suitable for running real world applications.

There are some fundamental deficiencies in a federated database approach that make it unsuitable for e-business applications. E-business customers have certain basic expectations of a database such as data integrity. Federated databases do not inherently provide guarantees for uniqueness of records and referential integrity across the database nodes. There are other serious limitations with federated databases that make them unsuitable for online e-business applications. One limitation is the lack of flexibility with regard to restricted query capabilities. Another is the manual consistency management of data dictionaries.

However, federated databases offer some advantages over single server databases. These advantages are almost exclusively are due to queries that can span multiple databases to obtain the necessary information. While commercial distributed database implementations have been available for almost a decade, this technology has been adopted in a very limited degree by database customers.

What Do E-Businesses Need?

Clearly, no single traditional architecture presented so far provides e-business customers with an 'ideal' cluster database platform for scalable and flexible e-business deployments. Online e-business applications need an architecture that provides both the highly fault-resilient and flexible implementation of shared disk database systems and that provides the near-linear scalability of shared nothing database systems.

ORACLE CACHE FUSION ARCHITECTURE

Cache Fusion utilizes a shared cache for performing cache coherency. This eliminates the need for disk IO for inter-node synchronization.

Oracle's Cache Fusion™ architecture is a new shared cache architecture that provides e-business applications the benefits of both shared disk and shared nothing databases without the drawbacks of either architecture. It does so by exploiting rapidly emerging disk storage and interconnect technologies. While Cache Fusion architecture is revolutionary in an industry sense, it is a natural evolution of the Oracle Parallel Server architecture.

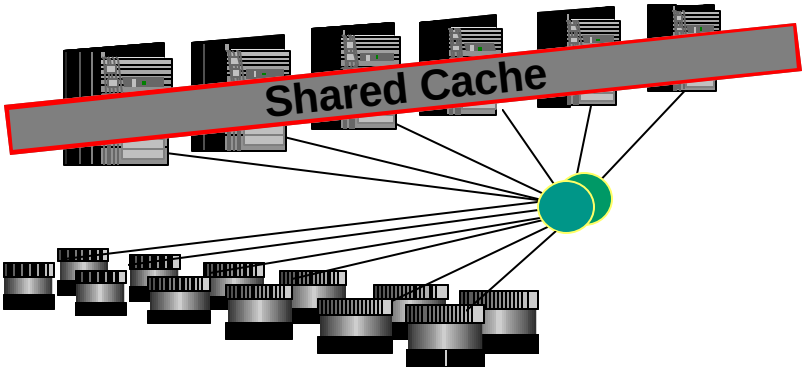

Figure 2: Cache Fusion utilizes a scalable shared cache for cache coherency; eliminates disk IO

Oracle Real Application Clusters with its Cache Fusion architecture provides the following key benefits to enterprise e-business application deployments:

Real Application Clusters scale application to multiple nodes without any modification to the application or changes to data placement.

- *Flexible* and *effortless* scalability for e-business applications; application users can log onto a single virtual high performance cluster server. Adding nodes to the database is easy and manual intervention is not required to partition data when processor nodes are added or when business requirements change. Cluster scalability for all applications out-of-the box — without modification.
- A higher availability solution than traditional cluster database architectures; this architecture provides customers near continuous access to data with minimal interruption by hardware and software component failures. The system is resilient to multiple node failures and component failures are masked from end-users.
- A single management entity; a single system image is preserved across the cluster for all management operations. DBAs perform installation, configuration, backup, upgrade, and monitoring functions once. Oracle then automatically distributes the management functions to the appropriate nodes. This means the DBA manages *one* virtual server.
- Cache Fusion preserves the investment that all Oracle customers have already made in learning and exploiting Oracle for their e-business applications; all the single node database functionality is preserved and application clients connect to the database using the same standard Oracle application interfaces.

SCALABILITY

E-business application users, or mid tier application server clients, connect to the database by way of a virtual database service name. Oracle automatically balances the user load among the multiple nodes in the cluster. The Real

Application Clusters database instances on the different nodes subscribe to all or some subset of database services. This provides DBAs the flexibility of choosing whether specific application clients that connect to a particular database service can connect to some or all of the database nodes.

While each node has a different physical internet protocol address, application clients connect at a logical database service name level. The clients are therefore not concerned with irrelevant issues such as multiple addresses for the multiple server machines.

E-businesses can painlessly add processing capacity as they grow. The Cache Fusion architecture immediately utilizes the CPU and memory resources of the new node. DBAs do not need to manually re-partition data. This benefit is a by-product of the architecture since transparent data access is fundamental to Cache Fusion.

Nodes added to the cluster are automatically utilized; no manual re-allocation of resources is needed. Cluster resources are dynamically re-balanced for optimal cluster utilization.

The Cache Fusion architecture automatically accommodates the rapidly changing e-business requirements and resulting workload changes. DBAs also do not need to manually re-partition data based on the changed workloads. Real Application Clusters dynamically reallocates database resources to optimally utilize the cluster system resources with minimum disk I/O and low latency communication among the nodes. This allows Real Application Clusters to deliver increased application throughput and optimal response time.

In traditional shared disk databases, synchronizing database resources across the cluster database poses significant challenges with regard to achieving cluster scalability. So, how does Cache Fusion solve these complex and age-old problems?

Cache Fusion Processing

There are primarily five key breakthroughs that make the Cache Fusion architecture scalable.

- Utilizing the collective (fast) database caches of all the nodes in the system to satisfy application requests to any one node
- Removing (slow) disk operations from the critical path of inter-node synchronization
- Greatly reducing the required number of messages for inter-node synchronization
- Using scalable inter-node data block transfers that greatly benefit from the above breakthroughs
- Exploiting low latency cluster interconnect protocols for both database messages and data shipping

For purposes of discussion, we divide these five breakthroughs into two sub-topics:

- The Global Cache Service management of the shared cache
- Efficient inter-node messaging and resource management

Global Cache Service Management of the Shared Cache

Cache Fusion exploits the caches of all nodes in the cluster for serving database transactions. This is a significant speedup over slow disk IO operations.

Traditional shared disk database systems use disk I/O for synchronizing data access across multiple nodes. Typically, when two or more nodes contend for the same data block, the node that has a lock on the data block writes it to disk before the other nodes can access the same data block. This requires disk I/O that involves moving components; it is inherently slow. It also requires synchronization in the form of messages that nodes need to exchange to communicate about lock status. While inter-node message communication can be performed more efficiently with a good design and optimal implementation strategy, the costs of disk I/O for synchronization pose significant challenges for scaling non-partitioned workloads or high contention workloads on traditional shared disk database systems.

The Cache Fusion architecture overcomes this fundamental weakness in traditional shared disk systems by utilizing the collective caches of all the nodes in the cluster to satisfy database requests. The Global Cache Service manages the status and transfer of data blocks across the buffer caches of the instances. The Global Cache Service is tightly integrated with the buffer cache manager to enable fast lookup of resource information in the Global Resource Directory. This directory is distributed across all instances and maintains the status information about resources including any data blocks that require global coordination. Query requests can now be satisfied by the local cache or any of the other caches. This reduces disk I/O. Update operations do not require disk I/O for synchronization since the local node can obtain the needed block directly from any of the cluster database node caches. Expensive disk I/Os are only performed when none of the collective caches contain the necessary data and when an update transaction performs a COMMIT operation that requires disk write guarantees. This implementation effectively expands the working set of the database cache and reduces disk I/O to dramatically speed up database operation. Let's look, however, at the specific types of cross-node contention for data and how the Cache Fusion architecture addresses each of them.

Cache Fusion utilizes both local and remote caches for serving all possible types of inter-node transactions

There are four situations that warrant consideration when multiple Oracle instances access the same data. For reasons of simplicity, the examples refer to a two-node cluster with "node 1" and "node 2."

- *read/read* — User on node 1 wants to *read* a block that user on node 2 has recently *read*.

- *read/write* — User on node 1 wants to *read* a block that user on node 2 has recently *updated*.
- *write/read* — User on node 1 wants to *update* a block that user on node 2 has recently *read.*
- *write/write* — User on node 1 wants to *update* a block that user on node 2 has recently *updated*.

The read/read case typically requires little or no coordination, depending the on specific database implementation. In a traditional shared disk implementation the request by the user on node 1 will be satisfied either via local cache access or by way of disk read operations. In the Cache Fusion implementation, the read request may be served by any of the caches in the cluster database where the order of access preference is local cache, remote cache, and finally disk I/O. If the query request is served by a remote cache, the block is transferred across the high speed cluster interconnect from one node's cache to another and expensive disk I/O is avoided.

Both the read/write and write/write cases, in which a user on node 1 updates the block, coordination between the instances becomes necessary so that the block being read is a read consistent image (for read/write) and the block being updated preserves data integrity (for write/write). In both cases, the node that holds the initially updated data block ships the block to the requesting node across the high speed cluster interconnect and avoids expensive disk I/O for read. It does this with full recovery capabilities.

In the case of the write/read case scenario, node 1 wants to update a block that's already read and cached by are remote instance or node 2. An update operation typically involves reading the relevant block into memory and then writing the updated block back to disk. In this scenario disk I/O for read is avoided and performance is increased as the block is shipped from the cache of node 2 into the cache of node 1.

This "fusing of the caches," to provide users an extended database cache for queries and updates, makes the Cache Fusion architecture a promising one for on-line e-business applications. However, efficient inter-node messaging is key to coordinating fast block transfers between nodes.

Efficient Inter-node Messaging and Resource Management

The efficiency of inter-node messaging depends on three primary factors:

- The number of messages required for each synchronization sequence
- The frequency of synchronization -the less frequent, the better
- The latency, or speed, of inter-node communications

The Cache Fusion architecture dramatically reduces the number of inter-node messages.

Number of Messages For Synchronization

The number of messages required for coordination is intricately linked with the Global Enqueue Service implementation in the cluster database. The Global Enqueue Service tracks which database resources are mastered on which nodes. It also handles resource allocation and de-allocation to synchronize data operations made by the user processes.

The Global Cache Service implementation enables coordination of a fast block transfer across caches by way of at most two inter-node messages and one intra-node message.

The user process that needs access to some piece of data is able to quickly locate it if the associated block is currently mastered on the local or remote node via an in-memory look-up in the Global Resource Directory. Afterwards, the processing branches in the following manner:

- If the data is not already available in a local or a remote cache, Oracle performs regular disk access—this is slow, but inevitable.
- If the data is in the local cache, it is accessed immediately without any message generation—clearly the best case.
- If the data is in a remote cache, generally:

1. The user process generates an inter-node message to the Global Cache Service (LMD) process on the remote node.
2. The Global Cache Service and instance processes in turn update the in-memory Global Resource Directory and send the block to the requesting user process. In some cases a third instance may be involved.

Each synchronization operation occurs with minimal communication and thus delivers fast response. However, does this scheme scale to several thousand user processes that may potentially access the system in an e-business application environment? Oracle's inter-node resource model is the key to providing this capability.

Frequency of Synchronization

The Cache Fusion architecture greatly reduces the frequency of required inter-node synchronization operations in the following two ways:

- Dynamically migrating resources based on data access patterns
- Employing efficient inter-node resource management

Dynamic Remastering of Resources

Resources dynamically migrate to nodes that demonstrate a favored access pattern. This yields faster performance by avoiding remote IPC communications in favor of local operations.

The Global Cache Service tightly integrated with the buffer cache enables the database to automatically adapt and migrate resources in the Global Resource Directory to a node that establishes a frequent access pattern for a particular data set. Dynamic resource remastering provides the key benefit of greatly increasing the likelihood of local cache access without additional IPC traffic. This is a dramatic improvement over manual partitioning and provides efficiencies in the Cache Fusion architecture by reducing both the number of resource-related inter-node messages and data block transfers across the cluster interconnect. Dynamic remastering works for data that has clear access patterns established, in other words, a set of data is most often accessed by database clients connecting to a particular node. The buffer cache implements the policy that determines when dynamic remastering should be invoked based on observed access patterns. This improves e-business application response by providing local resource management. Furthermore, it reduces the frequency of inter-node communications for synchronization.

Efficient Inter-node Row Level Locking

Oracle supports efficient row level locks. These row level locks are created when DML operations such as SELECT FOR UPDATE are executed by an application. These locks are held until the application commits the transaction. Any other application process will block if it requests a lock on the same row. Cache Fusion block transfers operate independently of these user visible row-level locks. The transfer of data blocks by the Global Cache Service is a low level process that can occur without waiting for row level locks to be released. Blocks may be transferred from on instance to another while row level locks are held. As long as there are no row level locks the Global Cache Service provides access to data blocks allowing multiple transactions to proceed in parallel.

Inter-node Message Latency

Cache Fusion exploits low latency communication protocols for high performance.

Most hardware systems vendors provide high bandwidth, low-latency interconnects today or are actively developing these cluster interconnect features. Virtual Interface Architecture (VIA) is the most popular low latency protocol for Intel-based systems. These protocols, also known as “memory mapped protocols,” enable shipping data blocks among the memory caches of distinct nodes without using operating system kernel operations. This accelerates application performance and saves CPU resources. The Oracle Cache Fusion architecture exploits VIA for Intel-based systems for purposes of both inter-node IPC messages as well as for data block transfers across nodes. In Unix systems, Real Application Clusters can exploit optimized IPC interfaces such as RDG in Compaq TruCluster. All of these Cache Fusion features result in improved application response times.

HIGH AVAILABILITY

Real Application Clusters provide a true high availability solution as opposed to a failover solution. The key breakthrough is providing full database access during most of the database recovery period. This makes Real Application Clusters an excellent platform for e-business applications requiring 24x7 availability.

Real Application Clusters excel in three primary areas with regard to high availability:

- Providing database access during database recovery
- Masking failures from end-users
- Resilient to N-1 node failures

The following diagram illustrates how Oracle maintains data access during database recovery.

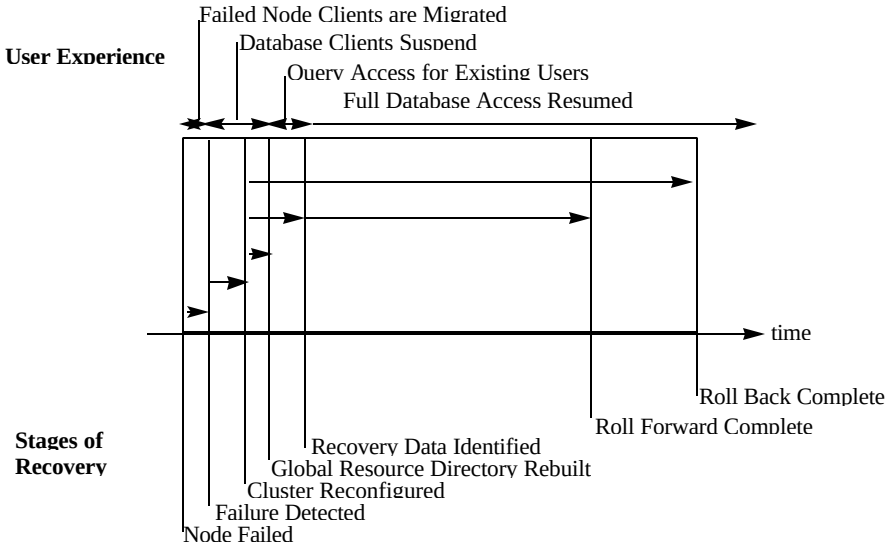

Figure 4: Database Recovery

Data Access During Database Recovery

Full database access occurs at a very early stage in the recovery process. In recovery, rebuilding memory structures is fast and performing I/O based recovery is slow. Thus, the phases of cluster reconfiguration and Global Resource Directory rebuilding occur relatively quickly. The cluster reconfiguration process excludes the failed node from the cluster map. The Global Resource Directory rebuilding process re-establishes resources from the failed node onto surviving nodes. Once these processes complete, query access is permitted to all existing clients who have block access. Concurrent with the

Full database access is resumed once the blocks needing recovery are identified.

directory rebuild operations, the recent transactions of the failed node are replayed. This happens in two stages.

- A Scan operation identifies data blocks needing recovery
- Replays of recent transactions occur

Once the scan operation completes, all database nodes are fully operational and the replaying of transactions on the failed node occurs in a deferred queue in the background. New transactions that conflict with data blocks that are identified as needing recovery temporarily suspend until their associated replay operations are complete.

Transparent Application Failover

Database clients do not need to reconnect, re-logon with automatic session failover.

Real Application Clusters provide near-continuous availability by hiding failures from end-user clients and application server clients. This provides continuous, uninterrupted data access. Transparent Application Failover in the database transparently re-routs application (query) clients to an available database node in the cluster when the connected node fails. Application clients do not see error messages describing loss of service. Failures are also hidden from update clients, in a similar fashion, by way of a simple application coding technique. The failover routine calls the appropriate client library function to re-rout the connection. Furthermore, you can configure database clients to pre-connect, or to have redundant idle connections. These redundant connections with another database node avoid delays if thousands of users must migrate servers during a node failure. These rich capabilities provide great benefits to e-business applications that cannot afford downtime.

Fault Resilience

Full data access is maintained as long as there is one surviving node in the cluster.

Real Application Clusters provide full database access and relatively uninterrupted operation as long as there is at least one surviving database node. In this event, all database resources are re-mastered onto the surviving node and all clients connect to that node for full data access. In this way, the Real Application Clusters database with the Cache Fusion architecture is resilient to N-1 node failures in an N node cluster.

MANAGEABILITY

All the same tools used for with the regular Oracle database work with Real Application Clusters. A single system operational view is presented.

Real Application Clusters is the realization of a true single system access database, providing seamless data access from any node to all disk devices and remote caches. This single system image extends to all database administrative operations, as well. Installation, configuration, backup, upgrade, and monitoring operations are performed once and are automatically distributed to all nodes in the cluster. The various Oracle tools such as Oracle Universal Installer, the Database Configuration Assistant, and Recovery Manager discover the different nodes in the cluster database and present them as targets for the desired tasks.

By selecting multiple node targets for a given administrative operation, the management task executes on multiple nodes in the database cluster. This presents e-businesses with tremendous economies of scale for managing their environment. For example, adding a node to the database cluster poses minimal incremental effort in managing it. In this manner, Real Application Clusters supports both online e-business applications and decision support applications alike and presents a single virtual high performance server for data access and administration.

E-BUSINESS APPLICATIONS

Cache Fusion technology makes Oracle Real Application Clusters an attractive database for online e-business applications. E-businesses need very high system availability with rapid application growth capability and the flexibility to dynamically alter application workloads in real-time based on changing business requirements; all this in a manageable environment. While all the prior discussions on availability, scalability, and manageability apply strongly to meeting these requirements, the dynamic resource allocation capabilities in the Cache Fusion architecture stand out in providing e-businesses the flexibility they need to move at the internet speed of e-business.

Flexible Implementation

E-business systems requirements and workloads change frequently. In fact, e-commerce systems requirements change so rapidly that labor and time-intensive data partitioning efforts to accommodate such changes may well be obsolete by the time they are phased into the system. More serious, however, is the sheer complexity and impracticality of partitioning packaged e-business applications such as one of the popular Oracle Financials applications. These applications have hundreds, or even thousands, of tables. The data localities and access patterns are a “black box,” for the most part.

Real Application Clusters provides the flexibility needed for e-business applications. Resources are automatically and dynamically balanced across to cluster for higher performance.

Technology consultants most often find partitioning such applications a daunting task. They usually recommend that customers use a larger computer or that they segment the application and/or application modules across distinct databases. However, segmenting applications can fragment data and obscure a global enterprise-wide view of information. This problem is addressed in Cache Fusion which dynamically migrates database resources to adapt to changing business requirements and workloads.

The Global Cache Service and buffer cache dynamically allocates and migrates database resources to nodes based on policies with regard to data access patterns. This makes the data available when you want it, where you want it. In other words, in the local cache — and as fast as you can possibly get it. The Cache Fusion architecture is flexible and can quickly adapt to changing business workloads. When system resources and requirements change, Cache Fusion

migrates resources to speed up access and make Real Application Clusters a high performance database platform for e-business.

DECISION SUPPORT APPLICATIONS

Today's e-businesses use decision support as a strategic weapon for acquiring customers and expanding markets. For example, a company that wants to run a product promotion gathers information using decision support systems to fine-tune a customer list that best fits the profile of the target demographic.

The deployment of decision support applications has been a traditional strength of Oracle Parallel Server. Extremely large customer implementations of Oracle Parallel Server with tens of nodes have been successful. Real Application Clusters continues to provide significant speed up for decision support applications in three primary ways. Cache Fusion:

- Enables cache access for large data sets and reduces disk I/O
- Employs flexible parallelism that efficiently uses cluster resources
- Makes maximum use of Dynamic Parallel Aware Optimization

Cache Access for Large Data Sets and Reduced Disk I/O

Cache Fusion facilitates rapid access to multiple caches on multiple remote nodes. Decision support applications receive a significant speed up from cache access because it minimizes the number of expensive disk I/Os that are needed to satisfy read requests. Cache Fusion extends the working set of the cache to be the aggregate sum of all the caches on all the nodes. While remote cache access is slower than local cache access, it delivers significant savings over disk access. This enables application users to enjoy significantly improved response times.

Flexible Parallelism

Real Application Clusters utilizes both function shipping and data shipping for load distribution and exploits all available cluster resources.

Each database table (or index) is assigned a degree of parallelism that is appropriate for operations on that object. The degree of parallelism can be a user-defined attribute of the table, specified as a parameter at table creation time, or you can assign it later using a SQL statement.

An appropriate degree of parallelism takes into account factors such as the number of instances that are active at run time, available processors, the number of operations running on each processor, table size, and the number of disk devices that store the table data. If set the degree of parallelism to use the system default, Oracle computes an intelligent default value.

Because table sizes and available processing resources can change over time, Oracle's dynamic architecture also provides easy ways to dynamically change the degree of parallelism. If the table is large such that a full table scan requires more than one minute, set the degree of parallelism to the default. If you are

using Oracle in a multi-user environment with many concurrent parallel operations, use the Adaptive Multi-User feature.

If you compare this flexibility to the static partitioning-based approach in which you must physically re-partition tables to alter the degree of parallelism, Oracle is clearly the superior choice. This flexible Oracle parallel execution technology is fully exploited and further complemented by Real Application Clusters in two important ways.

- Function shipping for load distribution
- Exploitation of data locality on shared nothing hardware

Function Shipping

Oracle's parallel execution on loosely coupled systems extensively uses the function shipping strategy to perform work on remote nodes. Parallel execution server processes located on remote nodes are sent efficient messages, often in the form of modified SQL sub-queries, to indicate the work that needs to be done. This may surprise many purists because function shipping is traditionally associated with shared nothing database systems. On systems where shared disk access is available, data shipping is the typical approach.

Oracle's parallel architecture, with its combination of key elements of both systems, makes intelligent use of function shipping when the data to be accessed is located at the remote node. This delivers efficient parallel execution and eliminates unneeded inter-node data transfers over the inter-connect.

Exploitation of Data Locality

This capability, while powerful, was more relevant during the popularity of shared nothing hardware systems. Almost all popular cluster systems today are based on a shared disk architecture.

Real Application Clusters exploits the data locality on shared nothing hardware systems.

Each node on a shared nothing hardware system has direct hardware connectivity to a subset of disk devices. In this case, it is more efficient to access these local devices from the 'owning' node. Real Application Clusters exploit this affinity of devices to nodes and delivers superior performance on these multi-computer systems. As with other elements of the Cache Fusion architecture, this strategy works transparently, without data partitioning. The system dynamically detects the locality of data and makes intelligent use of it in two ways:

- Spawns query server processes on nodes where the data to be processed is located
- Assigns local data partitions to each query server process to eliminate or minimize inter-node data movement

Such dynamic exploitation of data locality maximizes local data access and minimizes sub-optimal inter-node data transfer over the interconnect, delivering optimal performance on shared nothing hardware systems.

Dynamic Parallel-aware Query Optimization

Oracle's cost-based optimizer incorporates parallel execution considerations as a fundamental component in arriving at optimal execution plans. The optimizer dynamically computes intelligent heuristic defaults for parallelism based on the number of processors and the number of disk devices that store a given table. Evaluation of the costs of alternative access paths — table scans versus indexed access, for example — takes into account the degree of parallelism available for the operation. This results in execution plans optimized for parallel execution.

Intelligent decisions are made with regard to intra-node and inter-node query parallelism.

In a Real Application Clusters environment, intelligent decisions are made with regard to intra-node and inter-node parallelism. For example, if a particular query requires six query processes to complete the work and six CPUs are idle on the local node (the node that the user connected to), then the query is processed using only local resources. This demonstrates efficient intra-node parallelism and eliminates the query coordination overhead across multiple nodes. However, if there are only two CPUs available on the local node, then those two CPUs and four CPUs of another node are used to process the query. In this manner, both inter-node and intra-node parallelism are used to provide speed up for query operations.

In real world decision support applications, queries are not perfectly partitioned across the various query servers. Therefore, some CPUs complete their processing and become idle sooner than others. The Oracle parallel execution technology dynamically detects idle CPUs and assigns work to these idle CPUs from the queue tables of the over-loaded CPUs. In this way, Oracle efficiently re-distributes the query workload across all CPUs. Real Application Clusters further extends these efficiencies to clusters by enabling the re-distribution of work across all the CPUs of a cluster.

CACHE FUSION AT WORK

Oracle Application Standard Benchmark

Real Application Clusters scales real world applications transparently and out-of-the box.

One example of Real Application Clusters scalability is the Oracle Standard Benchmark that was designed to provide a workload representing a typical customer implementation of Oracle Application's most commonly used modules. Four of these are from Oracle Financial Applications: Account Payable (AP), Accounts Receivable (AR), Fixed Assets (FA) and General Ledger (GL) and three are from Supply Chain Management: Inventory (INV), Order Entry (OE) and Purchase Order (PO).

The workload simulates a mid-market company running a daytime load of both transactions, consistent with the modules mix, and a high frequency of relatively light batch jobs. The benchmark measures the user count with a response time of less than 4 seconds.

One audited result from Compaq running a two node ProLiant 6400R with Microsoft Windows NT 4.0 shows 3,248 users on Real Application Clusters. This can be directly compared to Compaq's Single Node benchmark result of 1,792 users running on a single ProLiant 6400R database. The result of this apple-to-apple comparison shows a 1.81 scalability factor for Real Application Clusters with its Cache Fusion technology. This details of this benchmark and its independent audited results are available at **http://www.oracle.com/apps_benchmark.**

Oracle Global Mail Server

The rapid growth of the internet and corporate intranets and extranets call for new message scaling requirements while supporting a broad range of user types. At the same time, cost controls for such a system are of primary importance to companies wishing to remain competitive. Consolidation is one way to go, but reducing the amount of hardware is only part of the solution to Oracle's messaging needs. To be able to run a company-wide, enterprise messaging system on a minimum number of servers requires software designed for that environment.

Every employee uses email to accomplish tasks ranging from obtaining business approvals to distributing documents to sending technical information. Email is also part of the workflow, such as tracking expense approvals and notifying the right people when a software bug is updated. Because email is such a vital part of Oracle's business, it must always be available — 24 hours a day, 7 days a week, 52 weeks of the year.

Oracle Email Server (OES) is a cost-effective messaging system designed to support very large corporate and internet service provider customers. This superior scalability translates to minimum hardware, simple deployment, and reduced administrative costs. OES's scalable performance, support for industry standards, and use of Real Application Clusters technology, all contribute to an economical, scalable messaging system as illustrated by the rollout of OES in Oracle Corporation's data center.

Before consolidation, the messaging system for Oracle employees worldwide consisted of:

- 97 servers
- 120 message stores
- Proprietary email clients, and

- 50 data centers, about 1 per country

The system required 60 administrators because there were multiple systems all over the world. The new globalized messaging system consists of:

- 1 high availability cluster with 2 nodes
- 4 database instances
- 1 global LDAP directory with failover
- 1 SMTP server with failover
- 5 IMAP servers

Real Application Clusters enables corporations to consolidate their applications and reap significant cost benefits.

To provide fault tolerance, another high availability messaging server cluster would be maintained in a different geographic location with hot backups. The reduction from 97 mail servers to just four will save Oracle $11 million dollars this year. Furthermore, the operational staff dedicated for email administration has been reduced from 60 people to 13 people.

Oracle Single Application Image

Oracle is using Real Application Clusters to reap the benefits of consolidation by centralizing multiple application instances across the globe onto a single cluster — Global Single Application Image project. Oracle has embarked on building a single global database encompassing ERP and CRM applications that will provide single global e-business practices for sales, service, finance, human resources, and so on. This will not only reduce the current 60 database instances down to two and consolidate the 14 data centers into two, with an expected annual savings of between $3.5 to $5.0 million, but it will give Oracle management a 360-degree view of the company and simplify our business processes. Consolidating the data is more efficient, it improves the quality and accuracy of data, and it simplifies the complexity and lowers labor costs.

Because so much of Oracle's business is dependent on theses applications, access to these consolidated systems is crucial. With Oracle Parallel Server running on the two high availability clusters, node failure on the back-end does not preclude data access because all data can be accessed from any node. In addition, data integrity is maintained because committed work on a failed node is recovered automatically without administrator intervention and without data loss.

These projects only represent Oracle’s initial consolidation efforts using Real Application Clusters. Other planned projects include using Oracle’s ERP and CRM products with Real Application Clusters — the anticipated result of which is an even greater cost savings over the next five years.

SUMMARY

Real Application Clusters, with its shared cache architecture — Cache Fusion, provides the most highly scalable and available database system for e-business. Cache Fusion provides flexible and effortless scalability for e-business applications because manual intervention is not required to re-partition data to take advantage of the new configuration.

Cache Fusion provides e-businesses a higher availability solution than do traditional cluster databases. This is because the Cache Fusion architecture provides customers with near continuous, uninterrupted data access with minimal interruption as a result of hardware and software component failures. The system is resilient to multiple node losses and end-users never notice component failures.

The Real Application Clusters architecture presents DBAs with a single management entity — a single system image. And this single system image is preserved across the cluster for all management operations. DBAs can perform installation, configuration, backup, upgrade, and monitoring functions once. Oracle then automatically distributes the execution of these management functions to the appropriate nodes within the cluster. The DBA thus manages one virtual server.

Cache Fusion preserves the investment that Oracle customers have already made in learning and exploiting Oracle for their e-business applications. All single node database functionality is preserved and application clients connect to the database using the same Oracle standard application interfaces.

Oracle has received widespread praise for its successful “billion-dollar savings” plan and data consolidation strategies. These successes serve as both an example of Real Application Clusters’ capabilities and as a model for others who want to efficiently manage their organizations in the ever-changing age of the internet.

Oracle9*i* Real Application Clusters
May 2001
Author: Sohan DeMel
Contributing Authors: Merrill Holt, Mark Bauer

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.