

Oracle Real Application Clusters 10gRelease 2
Technical Comparison with Microsoft SQL Server 2005

An Oracle Competitive White Paper
May 2005

Oracle Real Application Clusters 10gRelease 2

Technical Comparison with Microsoft SQL Server 2005

Introduction.....	3
Oracle Real Application Clusters Architecture	4
SQLServer 2005 Federated Database	5
SQLServer 2005 Federated Database Layout	5
SQLServer 2005 Federated Database Summary.....	7
How Oracle Real Application Clusters compares.....	8
SQLServer 2005 Failover Clustering	10
SQLServer 2005 Failover Cluster Database Layout.....	10
SQLServer 2005 Failover Cluster Database Summary	10
How Oracle Real Application Clusters compares.....	11
SQLServer 2005 Mirror Database.....	12
SQLServer 2005 Mirror Database Layout.....	12
SQLServer 2005 Mirror Database Summary.....	12
How Oracle Real Application Clusters compares.....	12
Summary	13

Oracle Database 10gReal Application Clusters

Technical Comparison with Microsoft SQL Server 2005

INTRODUCTION

The cluster database market is rife with competing marketing claims, with each vendor touting the benefits of its own architecture. The buyer has to make the choice of a mission-critical software platform while sifting through a mass of rapidly evolving benchmark results, conflicting analyst reviews and uniformly positive customer testimonials.

This paper is a technical evaluation of three different database technologies from Microsoft: the 'Federated Architecture', the 'Failover Clustering Architecture' (managed by MS Cluster Server) and the 'Database Mirroring Architecture' as represented by Microsoft SQL Server 2005. Each of these technologies is compared to Oracle's clustered architecture: Oracle Real Application Clusters10g Release 2. Oracle RAC forms part of Oracle's High Availability Architecture, which further enhances protection for the database from disasters.

The Federated and Failover Clustering architectures have existed in previous versions of SQLServer. The Microsoft Database Mirroring technology is a new SQLServer 2005 feature.

ORACLE REAL APPLICATION CLUSTERS ARCHITECTURE

It is important to stress that none of the features provided by Microsoft's SQLServer 2005 compare with the combined high availability and scalability features of Real Application Clusters.

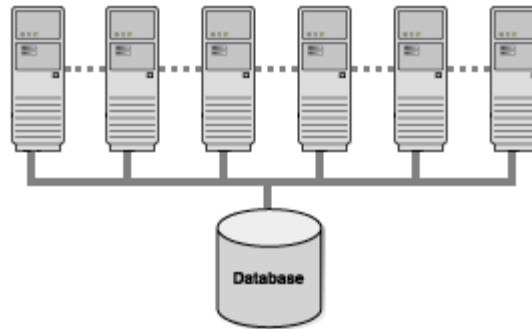


Figure 1 Oracle Real Application Cluster Architecture

Oracle's RAC architecture is unique in the Unix, Windows & Linux server space. In the above example all 6 nodes can process client requests for data from the one database. One of RAC's key differentiators is the inherent ability of the architecture to seamlessly survive a node failure. In the above situation, should a node fail, sessions that were connected to the failed node get migrated to the surviving nodes, balancing the connections to best use the available resources on the remaining five nodes. The other nodes in the cluster continue processing requests, sessions connected to these surviving nodes do not get disconnected.

It is worth noting that the nodes in the cluster do not have to be configured in exactly the same way. An example of this would be mixed workload environments, where the database is shared between OLTP & decision support style users. The database instances can be optimally configured to process requests for the connected user.

Oracle RAC also enables simultaneous use of all machines in a cluster to further enhance performance. As an example in this environment data warehousing queries can be automatically parallelized over all the CPU's available in the cluster boosting the performance of decision support style applications.

Using sophisticated load balancing algorithms users sessions can be routed to the 'least loaded' node in a cluster.

SQLSERVER 2005 FEDERATED DATABASE

SQL Server's Federated Database provides a level of scalability but at the cost of availability. As nodes are added to a SQL Server database the projected availability decreases as the database is reliant on all nodes being available to process requests

SQLServer's Federated Database model is a collection of independent servers, sharing no resources, connected by a LAN. Federated Databases are complex to implement. The following is a representation of a six node Federated Database.

Remember in the figure below there are six individual SQLServer database, each requiring independent backup & recovery. All of these databases must be online to satisfy requests from client applications.

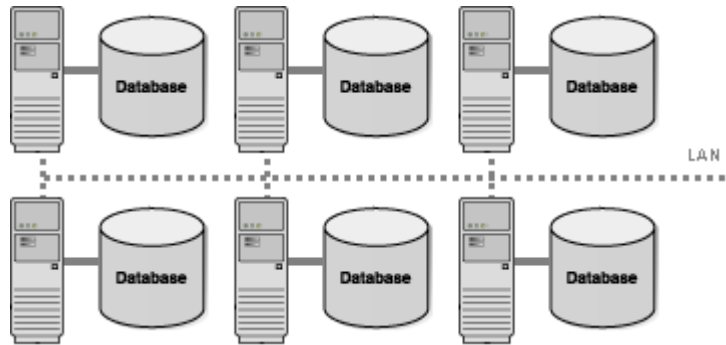


Figure 2 Microsoft's Federated Database Architecture

There are very few packaged applications that support this architecture.

SQLServer 2005 Federated Database Layout

The following describes the process of setting up a Federated Database.

Data is distributed across each participating server. For both the DBA as well as the Application Developer, there is a clear distinction between "local" data, which is on the disk attached to a particular server, and "remote" data, which is owned by another server in the federated database. Applications see a logical single view of the data through UNION ALL views and Distributed SQL – Microsoft calls this technology Distributed Partitioned Views (DPVs). The DPV is constructed differently at each node - it must explicitly consider which partitions are local and which are remote.

For example, here's a summary of how you would partition your customers across multiple servers in Microsoft SQL Server.

For each table in your application:

- First create independent tables on each node

```

-- On Server1:
CREATE TABLE Customers_33
(CustomerID    INTEGER PRIMARY KEY
        CHECK (CustomerID BETWEEN 1 AND 32999),
... -- Additional column definitions)

-- On Server2:
CREATE TABLE Customers_66
(CustomerID    INTEGER PRIMARY KEY
        CHECK (CustomerID BETWEEN 33000 AND 65999),
... -- Additional column definitions)

-- On Server3:
CREATE TABLE Customers_99
(CustomerID    INTEGER PRIMARY KEY
        CHECK (CustomerID BETWEEN 66000 AND 99999),
... -- Additional column definitions)

```

code from http://msdn.microsoft.com/library/en-us/createdb/cm_8_des_06_17zr.asp

- Then create connectivity information.

Linked Server definitions are required together with query optimization options on each participating server.

- Finally create a DPV at each node. Note that the view will be different at each node

```

CREATE VIEW Customers AS
SELECT * FROM CompanyDatabase.TableOwner.Customers_33
UNION ALL
SELECT * FROM Server2.CompanyDatabase.TableOwner.Customers_66
UNION ALL
SELECT * FROM Server3.CompanyDatabase.TableOwner.Customers_99

```

Benchmarks

In the past Microsoft have used DPV's to produce TPC-C benchmark figures. The TPC-C schema, unlike real-world application,s consists of only 9 tables, of which 7 of these have Warehouse_ID as part of their primary key. It is a trivial task to provide DPV's for each of the tables and create the associated indexes.

If we contrast this simple OLTP schema to real world applications

	<i>Tables</i>	<i>Primary Key Indexes</i>	<i>Alternate Key Indexes</i>
Peoplesoft	7,493	6,438	900
Oracle eBusiness (ERP)*	8,155	800	5,100
SAP	16,500	16,329	2,887

* Oracle eBusiness Suite does not support SQLServer. It is used here as a measure of the size of the schema used to support such enterprise scale applications

These applications require global unique indexes on non-primary key columns both for speedy data access as well as for ensuring data integrity. An example of this type of index would be the unique index on Customer_Number in the RA_Customers table in the Oracle eBusiness Suite, which ensures that there is only one customer with a particular value of the unique business key – a key which is not the primary key for the table. Without these indexes, mission critical application data can be corrupted, duplicated or lost.

Applications also usually do not partition their data accesses cleanly. It is generally not feasible to find partitioning keys for application tables that yield a high proportion of “local” data accesses. Local accesses are those in which the requirements of a query can be satisfied exclusively by the contents of a single partition of data. Most significant queries in SAP, PeopleSoft or the Oracle eBusiness Suite join multiple tables, and different queries use different alternate keys in the join predicates. And non-local data accesses incur the unacceptable performance overhead of frequent distributed transactions.

Even if a suitable partitioning key could be found, thousands of application tables would have to be partitioned. Thus, to port PeopleSoft or SAP to a federated database SQL Server configuration would require the creation and management of thousands of DPVs (one per partitioned table) – a Herculean task. And, since DPVs cannot support global unique indexes on alternate keys, this effort would guarantee serious violations of the integrity of critical business information. Hence, anything other than simplistic OLTP applications cannot be ported to run on federated databases.

SQLServer 2005 Federated Database Summary

There are a number reasons a Federated approach fails for ‘real world’ applications:

Hot Nodes

A DBA needs to be very careful how they partition their data to avoid creating a ‘hot’ node. They can’t simply partition on a percentage of the database, as this would not take into effect the distribution of queries and would cause a hot node. This node would then become a bottleneck, restricting throughput. Also if a DBA did manage to partition perfectly such that load was spread over all nodes, over time, as data was added and changed, query profiles change and what started out as a perfectly distributed system would end up unbalanced with hot nodes.

No Single point of truth

Because partitioning all of a database’s data is not an easy task. DBA’s tend to partition only the large tables and choose to duplicate the smaller tables amongst all the nodes. This means that any changes to the smaller tables need to be replicated to all the nodes in the cluster, as each node has it’s own database. This duplication causes multiple copies of data to be held in multiple SQLServer databases.

Adding nodes

As a workload grows there will become a time when a DBA needs to add a new node to their SQLServer Federated database. The process is: Install the OS, Install SQLServer, decide on a new partitioning scheme, unload the data from existing nodes, repartition the data, load the data into the new collection of nodes, bring the database back online, possibly have to make change to the application.

Consistent backups

A Microsoft SQLServer DPV database is actually a number of separate databases, each one needs backing up. More importantly, should there ever need to recover a DPV database then each of the individual databases need to be recovered to the same point in time and finally all of them would need to be brought online.

Coping with node failure

On failure of a node that section of data becomes unavailable to the application. Few real-world applications can tolerate a segment of their data being taken offline.

Benchmarking

Microsoft's DPV architecture became known as a benchmark special. It should be feasible for Microsoft to take the data for the TPC-C benchmark and, as the queries are all predefined, engineer a TPC-C result. They do not have any current TPC (-C or -H) benchmarks published using this technology.

How Oracle Real Application Clusters compares

Oracle Real Application Clusters Architecture is radically different. The shared disk approach provided by Oracle copes with these issues as follows:

Hot Nodes

With RAC data is not partitioned on a per-node basis. Connections are routed to the least loaded node. The hot node syndrome is not a feature of an Oracle RAC database.

Single Point of the Truth

Oracle only requires one copy the database. No additional copies are required using a RAC architecture. 'One copy the data' = 'A single point of the Truth'.

Adding Nodes

There could come a time when the number of nodes in an Oracle RAC database is insufficient for the workload. In Oracle RAC's case the procedure to be followed is: Install the OS, Install Oracle RAC, Bring the new instance online, The instance registers automatically with the database listeners and applications can make use of the new node instantly with no changes to either the application schema or the Application.

Consistent Backups

An Oracle RAC database is a single database image, irrespective of the number of nodes. So a single backup, and restore, backs up and recovers the database in a consistent way

Coping with Node Failure

No single node is responsible for a portion of the data, therefore losing a node in a RAC cluster does not mean that any data becomes inaccessible. With RAC failover occurs in seconds, connections to the node that failed can be automatically reconnected to the surviving nodes. Application tiers can be advised in a timely manner using 'Fast Connection Failover' that a node has died and can invalidate the connections in their connection pools relating to the failed node.

Benchmarking

Oracle regularly benchmark RAC clustered databases. The most recent being on HP hardware using Linux. See :
http://www.tpc.org/tpcc/results/tpcc_result_detail.asp?id=103120803. For this benchmark Oracle did not have to segment the data onto individual nodes.

SQL Server's Failover Clustering database offers a slightly higher level of availability compared to a 'standalone' database but offers nothing in terms of scalability.

SQLSERVER 2005 FAILOVER CLUSTERING

Microsoft Cluster Server (MSCS) is a technology Microsoft supports to provide a slightly enhanced level of availability compared to a standalone node. Microsoft call this 'Failover Clustering'. The following is a representation of an MSCS managed SQLServer Database

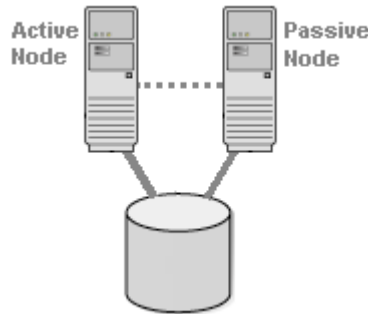


Figure 3 Microsoft's Failover Clustering Architecture

The hardware architecture looks similar to a RAC environment. There are 2 nodes, connected by a network interconnect. There is also what appears to be 'shared disk'. In fact the disk in a Failover Cluster database is not shared and the SQLServer database only runs on one of the nodes. The second node is provided as a backup should the first node fail.

SQLServer 2005 Failover Cluster Database Layout

A SQLServer database installed on this architecture looks and acts just like a single SQLServer database. It suffers from the same limitations that the hardware and operating system impose on it. It is restricted by the scalability of a single server. The files that make up the database reside on the central disk which is made available to the node that is running the SQL Server Database.

SQLServer 2005 Failover Cluster Database Summary

This clustering technology does not provide additional scalability. The individual database only ever runs on 1 node and is therefore limited by the scalability of a single node.

Failover causes an application blackout

Should the running node (node1) fail there is a delay whilst the steps required to restart the SQLServer database on a surviving node (node2) are completed. The steps can be summarized as follows

1. The node2 has to recognize that node1 has 'gone away'
2. The disks that were visible to node1 need to be made visible, via software to node2
3. The IP address that was in use on node1 needs to be created on node2
4. The network name that was in use on node 1 needs to be created on node2
5. The Services (e.g. SQLServer) that were being managed on node1 need to be started on node2
6. SQLServer on node2 then needs to recover and then open the database
7. Applications can reconnect and then restart their transactions

Note most the above steps must be done in serial (SQLServer can't start until the disks & Network have been started, The network name can't be created until the IP address has been instantiated

How Oracle Real Application Clusters compares

With Oracle RAC an instance of the same database runs on all nodes in the cluster, all nodes in the cluster share workload.

A RAC cluster does not require MSCS and is therefore not limited by the node restrictions MSCS imposes, Oracle RAC 10g Release 2 supports up to 100 nodes.

If a node fails other nodes in the cluster keep on providing service to existing connections. Transactions that were 'in flight' get rolled back by the other node in the cluster immediately, there is no need to wait for resources to be restarted and the database to be opened on other nodes ... it's already open.

With RAC, failover occurs in seconds rather than minutes. Connections to the node that failed can be automatically reconnected to the surviving nodes. Application tiers can be advised in a timely manner using Fast Connection Failover that a node has died and can invalidate connections to the failed node in their connection pools.

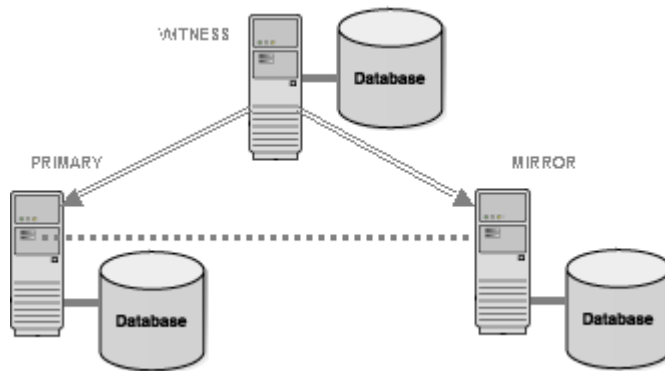
SQLSERVER 2005 MIRROR DATABASE

SQL Servers Mirror database is equivalent to one of the components of Oracle's Data Guard Architecture. Mirror Database on its own offers a level of protection but once again offers nothing to improve the scalability of the database solution.

Microsoft's Mirror Database is a new feature to be made available as part of the 2005 release of SQLServer. It maintains a second server at a remote site with a copy of the production database.

SQLServer 2005 Mirror Database Layout

Microsoft typically explains their new Mirror Database Technology as a high availability solution. As indicated in the diagram below three separate SQLServer databases are required. The first database is the production database, the second database is the mirror database and the third database acts as a monitor or 'Witness Server' In this figure the primary servers logs are written to a remote location: 'the Mirror Server'. The third acts as a monitor: the Witness Server.



SQLServer 2005 Mirror Database Summary

Offers nothing in terms of scalability. It's availability offering is similar in appearance to Oracle's Data Guard's Physical Standby database. The Mirror database is in an 'unavailable' state until it is required. This capability has existed as part of the Oracle database for many releases.

How Oracle Real Application Clusters compares

Database Mirroring has no comparison to Oracle RAC. It is analogous to the Oracle Data Guard Product. RAC & Data Guard combine as described in Oracle's High Availability Architecture¹ to provide unparalleled levels of both availability & scalability.

¹Information relating to Oracle's High Availability Architecture is available here: <http://www.oracle.com/technology/deploy/availability/techlisting.html>

SUMMARY

Microsoft's Federated Database offers nothing in terms of availability. In fact as nodes are added the actual measure of availability decreases. The scalability of the solution may work for TPC-C style benchmarks but it's ability to provide a scalable solution for enterprise applications is severely limited.

Oracle Real Application Clusters offers the opportunity to provide both High Availability and Scalability to an application with no changes to the application code or the application schema. An application that scales well from 2 to 4 to 8 cpu's on a single node would scale well from 2 to 4 to 8 nodes.

The Microsoft Failover Clustering solution may add a level of availability, a 'cold restart' capability, but does nothing in terms of scalability. A SQLServer database is constrained by the limits of the single hardware server.

Oracle RAC allows multiple servers to be combined to offer improved scalability and availability up to 100 nodes in a single cluster.

Microsoft's newest feature 'Mirror Database' mimics some of the technology Oracle has had in its Data Guard product for many years now. Once again a Mirror Database solution is constrained by the hardware limits of a single server.

Oracle Data Guard and Oracle RAC are complementary to each other. RAC addresses system or instance failures. It provides rapid and automatic recovery from failures that do not affect data such as node failures. It also provides increased scalability for an application and the opportunity to take advantage of commodity priced servers. Data Guard, as a complement to RAC, provides data protection through the use of transactionally consistent primary and standby databases, which neither shared disk or run in lock step. This enables recovery from site disasters or data corruptions.



White Paper: Oracle Real Application Clusters 10g Release 2: Technical Comparison with Microsoft SQL Server 2005

May 2005

Author: Philip Newlan

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
oracle.com

Copyright © 2005, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.