

Oracle Real Application Clusters 11g  
*Technical Comparison with Microsoft SQL Server 2008*

*An Oracle Competitive White Paper*  
May 2008

Oracle Real Application Clusters 11g  
*Technical Comparison with Microsoft SQL Server 2008*

Introduction .....	2
Oracle Real Application Clusters Architecture .....	3
SQL Server 2008 Federated Databases .....	4
SQL Server 2008 Federated Database Layout.....	4
SQL Server 2008 Federated Database Summary .....	6
How Oracle Real Application Clusters compares .....	7
SQL Server 2008 Failover Clustering .....	9
SQL Server 2008 Failover Cluster Database Layout .....	9
SQL Server 2008 Failover Cluster Database Summary.....	10
How Oracle Real Application Clusters compares .....	10
SQL Server 2008 Mirror Database.....	12
SQL Server 2008 Mirror Database Layout.....	12
SQL Server 2008 Mirror Database Summary .....	13
How Oracle Real Application Clusters compares .....	13
SQL Server 2008 Peer-To-Peer Replication.....	14
SQL Server 2008 Peer-to-Peer Replication Summary .....	14
How Oracle Real Application Clusters compares .....	14
Summary .....	15

# Oracle Real Application Clusters 11g

## *Technical Comparison with Microsoft SQL Server 2008*

### **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 four different database technologies offered by Microsoft as a possible architecture for Microsoft SQL Server 2008:

- Federated Databases
- Failover Clustering (managed by MS Cluster Server)
- Mirror Database
- Peer-to-Peer Replication

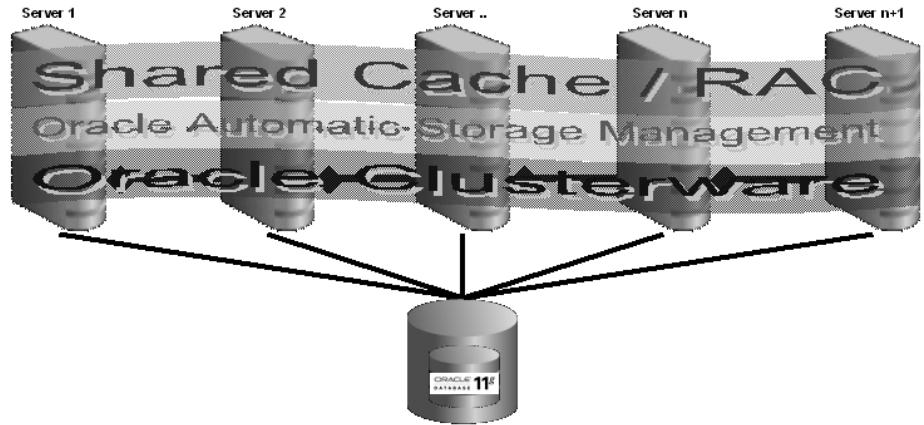
Each of these technologies is either compared to Oracle's clustered architecture – Oracle Real Application Clusters (RAC) –, or the respective Oracle technology. Oracle RAC and related technologies form Oracle's Maximum Availability Architecture (MAA), which enhances protection for the database from disasters.

All of the above architectures have existed in previous versions of Microsoft's SQL Server. Consequently, only improvements can be found for most of the features in SQL Server 2008. For Federated Database Servers no significant improvement could be found at all compared to the previous version.

In turn, all Microsoft solutions discussed in the course of this paper are compared to Oracle Database 11g Release 1 and Oracle Real Application Clusters 11g.

## ORACLE REAL APPLICATION CLUSTERS ARCHITECTURE

It is important to stress that none of the features provided by Microsoft's SQL Server 2008 compare with the combined high availability and scalability features of Oracle's Real Application Clusters.



*Figure 1: Oracle Real Application Cluster Architecture*

Oracle's RAC architecture is unique in the Unix, Windows, and Linux server space. In the above example all 5 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 and 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 and advisories, user sessions can be routed to the 'least loaded' node in a cluster.

Oracle Automatic Storage Management (ASM) and Oracle Clusterware complement the Oracle Real Application Clusters architecture, providing a storage management as well as a cluster software solution for the Grid infrastructure.

## SQL SERVER 2008 FEDERATED DATABASES

SQL Server's Federated Database model is a collection of independent servers, sharing no resources, connected by a LAN. Federated Databases are complex to implement.

The Federated Database model was available with Microsoft SQL Server 2005 already. There are no new features in this area that would be worth mentioning in this paper. Therefore, the following is a classical representation of a six node Federated Database.

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

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.

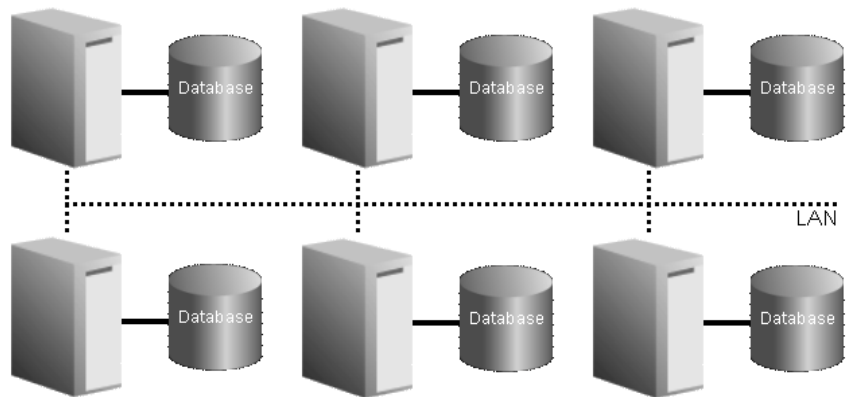


Figure 2: Microsoft's Federated Database Architecture

There are only very few packaged applications that support this architecture.

## SQL Server 2008 Federated Database Layout

The following briefly 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.

The example below shows how the 'customers' table would be partitioned 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 example taken from <http://msdn.microsoft.com/en-us/library/ms188299.aspx>

- 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

```

Code example taken from <http://msdn.microsoft.com/en-us/library/ms188299.aspx>

### Benchmarks

In the past Microsoft have used DPV's to produce TPC-C benchmark figures. The TPC-C schema, unlike real-world applications, consists of only 9 tables, of which 7 have a Warehouse\_ID as part of their primary key. It is a trivial task to provide DPV's for each of those tables and create the associated indexes. This impression changes when comparing this simple OLTP schema with 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 SQL Server. It is used here as a measure of the size of the schema used to support such enterprise scale applications

The applications listed in the table above require global unique indexes on non-primary key columns for both, fast 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 that 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, porting 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.

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.

## **SQL Server 2008 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 to partition the data to avoid creating a ‘hot’ node. The data cannot simply be partitioned on a percentage of the database, as this would not take into consideration the distribution of queries and might cause a hot node. This node would then become a bottleneck, restricting throughput.

Also, even if there was a way to perfectly partition the data initially so that load was spread over all nodes, over time, as data is 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 of database data is not an easy task, there is a tendency 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 its own database. This duplication causes multiple copies of data to be held in multiple SQL Server databases.

### ***Adding nodes***

As a workload grows there will be a time when a new node needs to be added to the SQL Server Federated Database. The process is: Install the OS, Install SQL Server, 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 SQL Server DPV database is actually a number of separate databases, of which each one needs to be backed up separately. More importantly, should there ever be a 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 a node failure the node 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**

The Oracle Real Application Clusters architecture is radically different from the Federated Database Architecture used by Microsoft SQL Server. The shared disk approach provided by Oracle copes with these issues mentioned above as follows:

### ***Hot Nodes***

With Oracle RAC, data is not partitioned on a per-node basis. Connections are routed to the least loaded node. The hot node syndrome does not exist in 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 of 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. Consequently, 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. Consequently losing a node in a RAC cluster does not mean that any data becomes inaccessible. With RAC failovers occur 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.

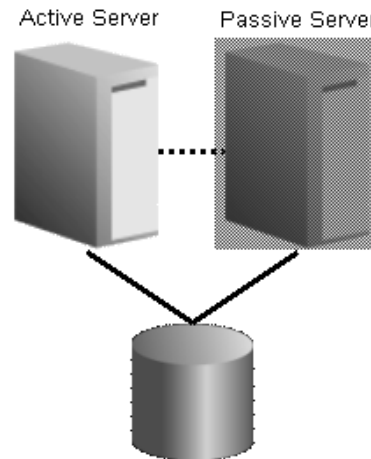
#### ***Benchmarking***

Oracle regularly benchmarks RAC clustered databases on various platforms. Oracle does not have to segment the data onto individual nodes for those benchmarks: [http://www.oracle.com/corporate/press/2007\\_nov/tpcc-hp-ow.html](http://www.oracle.com/corporate/press/2007_nov/tpcc-hp-ow.html) (Linux), [http://www.oracle.com/solutions/performance\\_scalability/11g-windows.html](http://www.oracle.com/solutions/performance_scalability/11g-windows.html).

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.

## SQL SERVER 2008 FAILOVER CLUSTERING

Microsoft Cluster Server (MSCS) is a technology Microsoft supports with the Microsoft SQL Server to provide a slightly enhanced level of availability compared to a standalone node. Microsoft calls this solution 'Failover Clustering'. Figure 3 shows a simple representation of a MSCS system that can be used to manage a SQL Server Database in a Failover Clustering manner:



*Figure 3: Microsoft's Failover Clustering Architecture*

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

Microsoft's Failover Clustering has been available for a number of SQL Server versions. With Microsoft SQL Server 2008, Microsoft enhanced the solution to support 16 nodes per cluster.

In addition, Microsoft supports 'Geographically Dispersed Failover Clustering', which enables a failover cluster to be setup across two different locations with one or more storage arrays at each site, eliminating the risk introduced by a single shared disk system. Finally, a Cluster Validation Tool can be used to ensure that one has adequate hardware resources to run a Microsoft Failover Cluster solution.

### SQL Server 2008 Failover Cluster Database Layout

A SQL Server database installed on this architecture looks and acts just like a single SQL Server 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. Even in clusters with more than two nodes, any additional node is of no immediate use for the database, since a certain database can only run on a specific node at a time. Having a 16 nodes cluster may be beneficial for 8 databases, but is of no further use for 1 database.

The files that make up the database reside on a file system on the central disk that is made available to the node that is running the SQL Server Database. Using the new 'Geographically Dispersed Failover Clustering' feature, the data can be copied over to a second storage system, protecting from a complete data loss in case of a failure of the primary storage. However, the data still needs to be made available using the secondary storage after the failure occurred, which is not uninterrupted.

### **SQL Server 2008 Failover Cluster Database Summary**

This clustering technology does not provide additional scalability. The individual database only runs on 1 node and is therefore limited by the scalability of a single node. Supporting 16 nodes in a cluster does not increase high availability, either, since the database can only run on 1 node at a time. Once this node fails, all connections to this database are lost.

#### ***Failover causes an application blackout***

Regardless of the number of nodes in the cluster, fact is that should the node that hosts the SQL Server database (e.g. node1) fail, there is a delay due to the steps required to restart the SQL Server database on a surviving node (e.g. node2). Those steps can be summarized as follows

1. Node2 has to recognize that node1 has 'gone away'
2. The disks (file systems) 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 used on node 1 needs to be created on node2
5. The Services (here especially the SQL Server Database, plus related services) that were being managed on node1 need to be started on node2
6. The SQL Server database then needs to recover on node2 before the database can be opened and be fully operational again.
7. Applications need to reconnect and then restart their transactions.

Note: most the above steps must be done sequentially (SQL Server cannot start until the disks and the network have been started. The network name cannot be created until the IP address has been instantiated.

### **How Oracle Real Application Clusters compares**

A failover cluster solution in general cannot be compared with Oracle Real Application Clusters. With Oracle RAC an instance of the same database runs on all nodes in the cluster, enabling all nodes to actively share workload in the cluster. This means, Oracle Real Application Clusters can make use of any new node in the cluster, scaling out the workload immediately across all nodes.

A RAC cluster does not require MSCS (it comes with Oracle Clusterware as its cluster software solution) and is therefore not limited by the node restrictions MSCS imposes. Oracle RAC from 10g Release 2 onwards supports up to 100 nodes and most operating systems certified for the Oracle Database.

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

With RAC, failovers occur in seconds rather than minutes. Connections to the node that failed can be automatically reconnected to the surviving nodes. Application tiers can be notified that a node has died in a timely manner using Fast Connection Failover, enabling them to immediately invalidate connections to the failed node in their connection pools.

Using Oracle's Automatic Storage Management (ASM), mirroring data either within a storage subsystem or between more than one storage system is totally transparent to the RAC database. ASM provides normal redundancy to maintain a single mirror (either on another disk or another storage systems) and high redundancy to maintain 2 data copies in different locations.

ASM stripes the data across all disks, ensuring an optimized IO utilization. In addition, all ASM diskgroups, the logical unit in which ASM organizes the disk available to ASM, are open and accessible on every node of the cluster at any point in time. In case of a node failure, no data needs to be made visible to another node, since all the data is already accessible to any node. In case of a storage failure, the RAC database remains fully operational without any service interruption.

## SQL SERVER 2008 MIRROR DATABASE

Microsoft's Mirror Database was a new feature in Microsoft SQL Server 2005. It was provided in an early version as part of the initial SQL Server 2005 release. Microsoft changed the status as part of the SP1 upgrade.

With SQL Server 2008, the Mirror Database feature has undergone some further enhancements. For example, only in the 2008 version of SQL Server, a manual failover can be performed without restarting the database.

In addition, the mirroring protection is extended to pages of data in a way that once a page is found to be corrupt on either the principal server or mirror server, the corresponding page is retrieved from its partner server and the database operation can continue seamlessly. Using a newly available compression technology, Microsoft furthermore allows compressing the data flow between the primary server and the mirror server to reduce network traffic.

While the classic Mirror Database approach could only have 1 mirror server, Microsoft SQL Server 2008 introduces a variant allowing periodical Log Shippings between more than 1 warm standby site. Log shipping occurs on a schedule and therefore, there is a time delay between the data changes on the master server and these changes transferred to the secondary server(s).

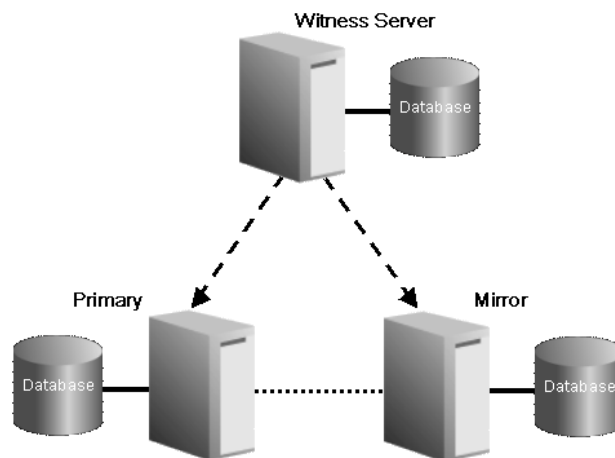
**SQL Server's Mirror Database is comparable to Oracle Data Guard.**

**Mirror Database offers a certain level of protection, but once again, does not improve the scalability of the SQL Server database itself.**

### SQL Server 2008 Mirror Database Layout

Microsoft typically explains their Mirror Database Technology as a high availability solution. As indicated in the diagram below, three separate SQL Server databases are required to make full use of this feature.

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'. While the Witness Server is per definition optional, it is the only way to enable an automatic failover between the primary and the mirror database. Without the Witness Server, a failover always needs to be initialized manually.



*Figure 4: Microsoft's Mirror Database Architecture*

In figure 4, logs are shipped between the primary SQL Server database and the mirror SQL Server Database. The Witness Database acts as a system monitor. The classic Mirror Database technology supports only 1 mirror server.

### **SQL Server 2008 Mirror Database Summary**

Mirror Database does not provide any kind of scalability. The availability offering of this SQL Server 2008 feature appears to be similar to Oracle Data Guard's Physical Standby database. The Mirror database is in an 'unavailable' state until a failover occurred or was initialized manually.

Considering the new functionality of extended the mirroring protection to pages of data, compressing the data flow between the 2 servers and supporting more than 1 secondary server based on an asynchronous, scheduled log shipping, one has to attest that those capabilities have 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 technology. Oracle RAC and Oracle Data Guard can be combined to provide unparalleled levels of both availability as well as scalability, as described in Oracle's Maximum Availability Architecture<sup>1</sup>.

---

<sup>1</sup>For more information on Oracle's Maximum Availability Architecture (MAA) is available here: <http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm>

## SQL SERVER 2008 PEER-TO-PEER REPLICATION

While Microsoft has fitted their SQL Server with a replication feature for quite some releases, the Peer-to-Peer replication was one of the main new features in SQL Server 2005 and is still one of the most enhanced features in SQL Server 2008.

**SQL Server's Peer-to-Peer Replication is comparable to Oracle's Replication technology.**

**Peer-to-Peer Replication offers a certain level of load balancing, but once again, does not improve the scalability of the SQL Server database itself.**

According to Microsoft Peer-to-Peer Replication is a high availability solution that can also be used to load balance workload across more than one database server, while each database server is independent. The basic idea of a Peer-to-Peer Replication is that each subscriber is also a publisher, so that data can move both ways, like in a bi-directional replication.

New in SQL Server 2008 Peer-to-Peer Replication is the ability to add nodes online. In former versions of this feature any node addition would have required to quiesce the replication process for a certain amount of time.

Another enhancement is the 'Conflict Detection' technology, so one *can protect* against accidental conflicts when multiple replication nodes update the same row. While former SQL Server versions would not even have detected the concurrent update of the same row, which could have caused a "conflict or even a lost update when the row is propagated to other nodes"<sup>2</sup>, enabling Conflict Detection would now prevent lost updates by treating them as critical errors.

The new graphical Topology Viewer used to manage and monitor the replication nodes complements the Peer-to-Peer Replication functionality in SQL Server 2008.

### SQL Server 2008 Peer-to-Peer Replication Summary

Peer-to-Peer Replication can be used to load balance workload, but does not provide any scalability. In general, this SQL Server 2008 feature appears to be similar to Oracle's Multi-Master or Streams Replication, which has been available for quite some Oracle Database releases. Given the new Conflict Detection in SQL Server 2008, it is worth noting that "in the event of a conflict, the topology remains in an inconsistent state until the conflict is resolved and the data is made consistent across the topology."<sup>3</sup>

### How Oracle Real Application Clusters compares

Peer-to-Peer Replication has no comparison to Oracle RAC. It is analogous to the Oracle Replication technology that has been available for many Oracle releases.

---

<sup>2</sup> Quote from the Oracle SQL Server 2005 document: <http://technet.microsoft.com/en-us/library/bb934199%28SQL.100%29.aspx>

<sup>3</sup> Quote from the Oracle SQL Server 2008 document: <http://technet.microsoft.com/en-us/library/bb934199%28SQL.100%29.aspx>

## SUMMARY

Microsoft's Federated Database does not provide any additional 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 its 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 CPUs 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 not provide any 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 relatively new feature 'Mirror Database' mimics some of the technology Oracle has had in its Data Guard product for many years. 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.

Microsoft's Peer-to-Peer Replication, although once of the most enhanced features in SQL Server 2008 does not provide any better scalability or availability than the former versions. Features like online node addition or Conflict Detection have been available in Oracle's Replication technology for many releases.

Oracle RAC should not be compared to any replicated environment. Solutions for replicated or distributed databases have been on the market for quite a while and established their reputation for certain purposes. RAC, however, provides scalability and high availability without the requirement to change the application.



Oracle Real Application Clusters 11g: Technical Comparison with Microsoft SQL Server 2008

May 2008, Version 1.1

Author: Markus Michalewicz

Contributing Authors: Philip Newlan, Barb Lundhild

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](http://oracle.com)

Copyright © 2007, 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, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.