Flexible Database Clusters

Implementing Large Clusters with Oracle9i Real Application Clusters and PolyServe Matrix Server

An IBM-PolyServe White Paper

Oracle9*i* Real Application Clusters (RAC) is the cornerstone for building flexible, high performance, highly-available, clustered database solutions on Linux. Connecting such clusters to a fault-resilient FibreChannel SAN lays the foundation for the computing infrastructure known as Flexible Database Clusters (FDC).

While a cluster of this magnitude comprised of modern Intel-based servers may rival the established UNIX server, is it manageable? What is the performance impact at the application level? How does it affect Total Cost Ownership (TCO)?

To answer these questions, IBM and PolyServe joined forces to build a 16-node cluster running SuSE Linux and attached it to a formidable SAN configured with 206 physical disk drives. Oracle9*i* RAC was then installed on PolyServe Matrix Server.

The FDC cluster was the target of a series of tests that took an in-depth look at running and managing not just a single application, but three separate applications. The results of the testing confirmed that Flexible Database Clusters provide:

- A means to consolidate and deploy multiple databases and associated applications in a single, easily managed cluster environment.
- Simplified management of large database clusters, made possible by the PolyServe Matrix Server clustered file system.
- Dynamic "scalability on demand" architecture, enabling near-linear speedup to running applications— with little or no interruption.
- Dynamic repurposing of server resources on demand to quickly and easily move processing capacity to where it is most needed.
- An autonomic, always-on operating environment with immediate self-healing and little or no performance degradation (and therefore increased utilization rates).
- Dramatic incremental TCO benefits from improved manageability, scalability, expandability, availability and asset utilization.

A second "Flexible Database Clusters – Technical Findings" white paper is available at http:/www.polyserve.com for those readers interested in a deeper technical discussion of the findings of this paper.

Flexible Database Clusters

Economics of Consolidation

The cost factors associated with building Intel-based clusters running Linux in support of Oracle9*i* RAC are both proven and substantial. The common deployment method for Oracle on Linux is a small cluster per database application. However, just as consolidating applications onto a large SMP system is a proven cost-saving action, consolidating applications onto a large cluster provides great benefits. Consolidation reduces administrative overhead and enables the power of RAC to be fully exploited—the result is a "Flexible Database Cluster."

Flexible Database Cluster Components

The components used in the Flexible Database Cluster make it a powerful platform for supporting multiple applications.

Oracle9*i* RAC

Oracle9*i* RAC can exploit a large, flexible cluster consolidation. It includes capabilities not available with other database products:

- Availability— Oracle9*i* RAC is fault resilient and provides the ability for nodes to join an application in the event of a down server.
- Scalability—Applications scale well due to Oracle's Cache Fusion.
- Flexibility—Multiple Oracle database applications can share a SAN from within a single cluster, reducing administrative overhead, and nodes can be reprovisioned from one application to another.

Hardware

Today's technology makes it easy to build large clusters such as the Flexible Database Cluster. Modern Intel-based servers running Linux are very capable platforms when coupled with the unmatched capabilities of Oracle9*i* RAC. FibreChannel SANs and switches can be easily assembled in a large cluster.

Managing storage is also easier and more powerful. For example, the IBM FastT intelligent storage array used in the FDC test system completely handles RAID issues and reduces administrative overhead dramatically. The SAN switch was also easily configured via the IBM TotalStorage SAN FibreChannel Switch Specialist management tool.

Single points of failure were reduced in the FDC test system through the combination of a fully redundant switched FibreChannel SAN and the Multipath I/O feature provided by PolyServe Matrix Server.

PolyServe Matrix Server

Matrix Server shared data clustering software, which includes a true symmetric cluster file system, is both general purpose and Oracle optimized. It provided the following advantages in the FDC tests:

- Improved management of applications, and shared Oracle Home.

- Simple, contained database movement between applications (for example, transportable tablespace from OLTP to DSS without accessing the network).
- Large database loads with External Tables and Parallel Query.

The "shared Oracle Home" functionality is one of the keys to the Flexible Database Cluster architecture. The cluster file system component of Matrix Server supports setting up a single directory for Oracle Home. Oracle needs to be installed only once—on the Matrix Server cluster file system as a single-node install. That Oracle Home is then "converted" to a shared Oracle Home, allowing all nodes in the cluster to use the same executables. Also, configuration files are located in the Matrix Server cluster file system and can be edited from any node in the cluster.

In addition, of course, Matrix Server can provide "shared home" for applications and middleware other than Oracle9*i* RAC running alongside Oracle (or in separate clusters), and can also provide high availability for all of those applications and middleware services as well.

Matrix Server also provides these additional benefits in an Oracle9*i* RAC environment:

- With Matrix Server, all Oracle files can be stored in the file system. This includes, but is not limited to, the Oracle Cluster Management quorum disk, srvconfig file, control files, online and archived redo logs, datafiles, imp/exp files, SQL*Loader source files, External Tables.
- Matrix Server provides cluster-wide uniform device naming, which reduces "device slippage" and related problems. Device slippage complicates cluster administration and, if not handled carefully, can threaten to corrupt data.
- Matrix Server enables the Oracle Managed Files feature in an Oracle9*i* RAC environment. With Oracle Managed Files, the database itself creates and extends tablespaces dynamically, as needed, simplifying database administration.
- Matrix Server enables database tablespaces to be stored in standard file system files, and supports access by standard backup tools and utilities. This permits standard third-party backup tools to be used for Oracle database tables.
- Matrix Server enables Oracle's External Table feature to allow all cluster nodes to access data stored in flat files, and it permits Extract/Transform/Load (ETL) processes to run in parallel across all nodes in the cluster.
- Matrix Server extends Oracle's availability capabilities by providing system-wide wellness and failover for: applications, middleware, servers, networking, and file system storage.
- Matrix Server also improves availability by supporting integrated multi-path I/O for multiple Fibre Channel connections from servers to the SAN and multiple switches with the SAN. In such a configuration, all cluster nodes can continue to operate even in the presence of multiple cable, HBA, or switch failures.
- Matrix Server integrates with fabric access control mechanisms to enable only correctly-functioning cluster members access shared data.

For more information on the PolyServe Matrix Server value proposition for Oracle9*i* RAC visit: http://www.polyserve.com/products_literature.html

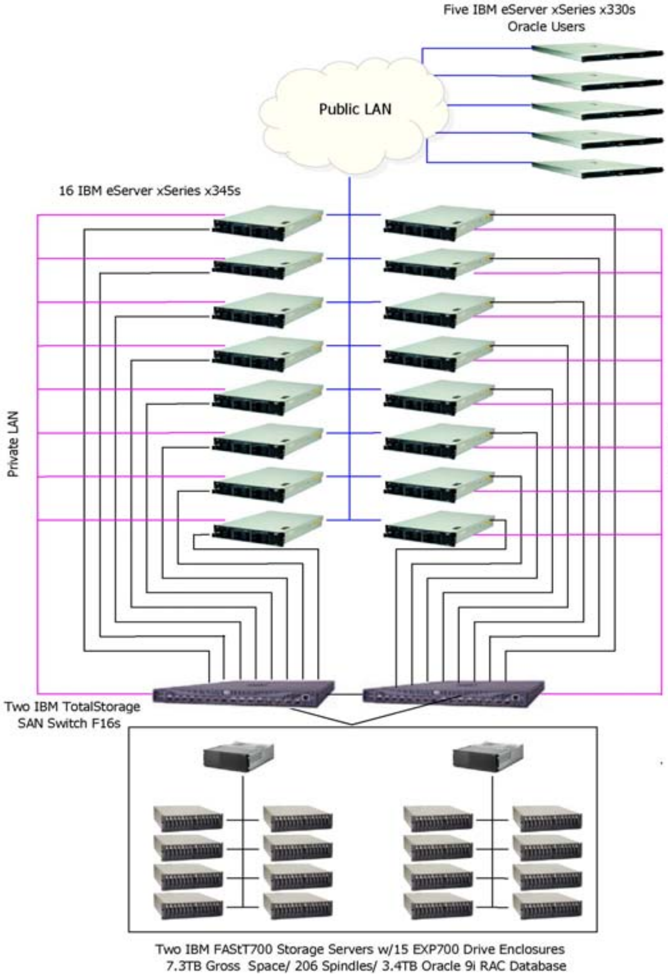
Matrix Server Oracle Disk Manager

PolyServe Matrix Server provides MxODM, an implementation of the Oracle Disk Manager (ODM) interface. MxODM offers improved datafile surety through cluster-wide file keys for access and enables Oracle9*i* with asynchronous I/O on the direct I/O mounted file systems where it stores datafiles and other database files such as redo logs. The monitoring capability of MxODM is a major   benefit in a FDC architecture deployment.

The MxODM I/O statistics package provides I/O performance information at a cluster-wide level, database global level, instance, or node level. Because MxODM understands Oracle file, process, and I/O types, its offers specialized reporting that focuses on key Oracle “subsystems” such as the Parallel Query Option (PQO), Log Writer, and Database Writer.

Proof of Concept

The following illustration shows the components used in the FDC test system.


Five IBM eServer xSeries x330s
Oracle Users
Public LAN
16 IBM eServer xSeries x345s
Private LAN
Private LAN
Two IBM TotalStorage
SAN Switch F16s
Two IBM FAStT700 Storage Servers w/15 EXP700 Drive Enclosures
7.3TB Gross Space/ 206 Spindles/ 3.4TB Oracle 9i RAC Database

System Overview

The following cluster system components were used in the FDC test system:

- **Client nodes**: Five IBM xSeries 330 1U rack-optimized servers.
- **Server nodes:** 16 IBM eServer xSeries x345 servers. The architecture of these servers contributed to the high availability of the FDC environment and, when combined with the Multipath I/O feature provided by Matrix Server, made it possible to build a SAN subsystem free of single points of failure.
- **Storage:** Two IBM TotalStorage FAStT700 Storage Servers and 15 FAStT EXP700 Storage Expansion Units with 206 36.4GB HDDs, providing over 7 TB of storage space.
- **LAN:** All public LAN traffic between clients and cluster nodes as well as Oracle interconnect traffic was exchanged through a Catalyst 6509 switch. Additional traffic on the switch included Ethernet access to the FAStT700 Storage Server for configuration and management of the storage subsystem.
- **SAN switch:** Two IBM TotalStorage SAN Switch F16s were used to interconnect the cluster nodes and the SAN.

Database Overview

To test the Flexible Database Cluster architecture, three databases, OLTP, DSS, and DEV, were created in the Matrix Server cluster file system using Oracle9*i* Release 2 version 9.2.0.3. The particular workloads chosen for the Flexible Database Cluster tests were not as important as the fact that there were three (3) of them. The goal was to have a realistic mix of processing running on the system while testing the manageability of the FDC architecture.

OLTP Database (PROD)

The OLTP database schema was based on an order entry system and contained Customers, Orders, Line Items, Historical Line Items, Product, and Warehouse application tables. The total database size was approximately 810 GB.

The application workload accessing the PROD database initially connected 100 users per node and ramped up to 500 users per node. The nodes under test were evenly loaded. Each user cycled through a set of transactions. At the end of each transaction the client process slept for a small random period of time to simulate human interaction.

DSS Database

The DSS database schema simulated a sales productivity decision support system. It contained space for both an extraction of the Customers table from the PROD database and several smaller datamarts.

The DSS workload was set up as a stream of queries in a queue serviced in serial order by the Parallel Query Option (PQO). Changes to the number of instances did not affect the current query being executed, but at the beginning of each query PQO considered how many instances there were. An increase from, say, 2 to 4 nodes caused a speed-up on the next query that was executed!

Most queries pushed through the queue had an access method of full table scan. A few included index range scans. The goal was to keep the PQO busy as instances were dynamically added, and these queries generated sufficient demand on the I/O subsystem.

DEV Database

The DEV database was a simple insert engine designed to test scalability while inserting records 2 KB in size. The database was approximately 10 GB. It had only two threads defined; therefore, only two instances could access this database at one time.

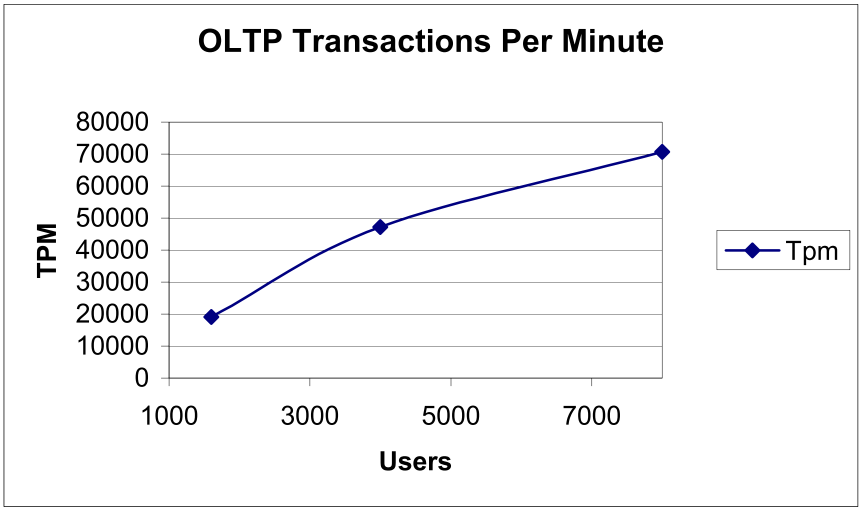
The DEV workload was a zero think time program that inserted 2K rows via pipe to SQL*Loader. The streams of loader processes executed on up to two nodes when DEV was being tested along with the other workloads.

FDC Test Results

The goal of the FDC case study was to test and prove the architecture and to ascertain value-add in key areas such as "on-demand" scalability and higher availability.

Stress Testing

A test suite was set up with 16 instances of the PROD database. 100, 250, and then 500 users were connected per node, each executing the OLTP workload. The following graph shows that the cluster and Oracle9*i* RAC performed nicely as the user count increased from 1600 clusterwide to 4000 and then 8000. In fact, when the workload increased from 100 to 250 users per node, the throughput increased from 19,140 transactions per minute (tpm) to 47,160 tpm—98% of linear scalability.



Fault Resilience Test

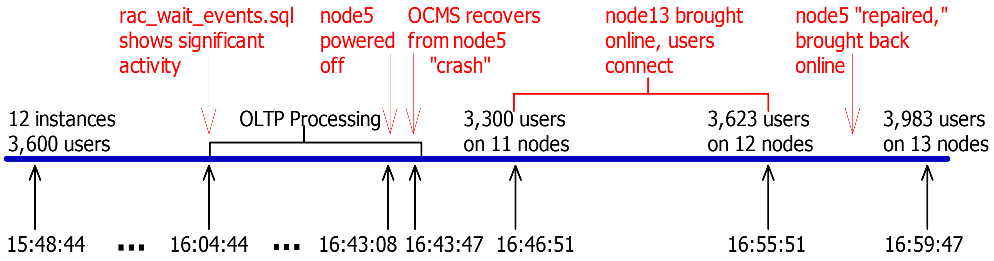
The FDC architecture with PolyServe Matrix Server is arguably the only approach that can fully harness the ability of RAC to sustain operations in light of multiple node failures. This test was intended to measure the ability to quickly reprovision nodes in the event of a failure. An OLTP workload was running on the first 12 nodes while the remaining four nodes were running a light DSS workload.

The test proved the ability of Oracle9*i* RAC to handle a crash of one of the 12 nodes, as well as its ability to reprovision a DSS node to take the place of the crashed node. Highlights of the test were:

- Application reconfiguration was unnecessary.
- The crash and replacement of a node were completely transparent to users.
- The *tnsnames.ora* file was structured to allow load balancing of new requests to the newly added node.

All operations were completely dynamic—the Oracle9*i* RAC way!

The following timeline gives a bird's-eye view of what transpired during the test.



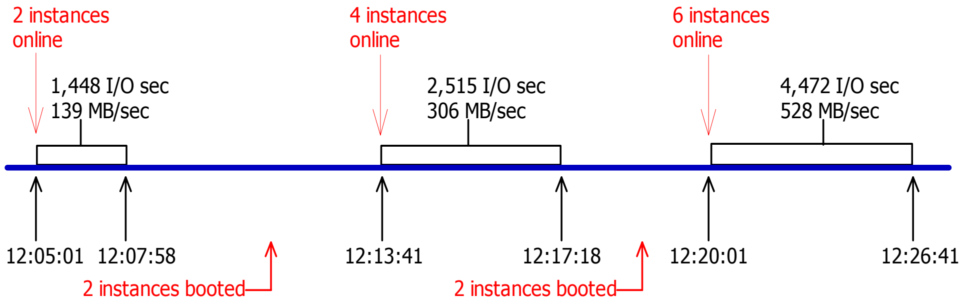
On-Demand Scalability Test

This test was intended to measure the ability of the FDC to quickly add nodes to a running application. The application was a queue of DSS queries being serviced by the Parallel Query Option. The test results show that three times the server resources were added in a 13-minute timeframe—with little or no interruption. Highlights of the test were:

- Application reconfiguration was not necessary.
- The addition of resources was completely transparent to users, except that the queries completed faster!
- The Parallel Query Option simply takes advantage of the bandwidth on the very next query from the queue.

The addition of resources was completely dynamic—again, the Oracle9*i* RAC way!

The following timeline shows what transpired during this test.



Summary

Large cluster configurations such as the Flexible Database Cluster are a natural progression for exploiting the functionality and power of Oracle9*i* RAC. These large clusters offer the following benefits:

- A single large cluster is easier to manage than many small clusters.
- Cluster nodes provide a pool of flexible resources for use among applications.
- The availability of Oracle9*i* RAC is enhanced; nodes can be dynamically reprovisioned to cover for the loss of another node, and on-demand scalability is supported.
- A general-purpose cluster file system like that included with Matrix Server provides a single-system feel and greatly enhances manageability. A shared Oracle Home used by all nodes also simplifies management. Support is also available for all database operations that require a file system.
- Improved manageability, scalability, expandability, availability and asset utilization in a FDC cluster also dramatically improves TCO.

PolyServe, Inc.
20400 NW Amberwood Drive
Suite 150
Beaverton, OR 97006

(877) 476-5973 (within U.S.)
(503) 617-7592 (fax)
(503) 617-7574 (outside U.S.)
www.polyserve.com

FDC_summary 100803 IBM markup sbn v3