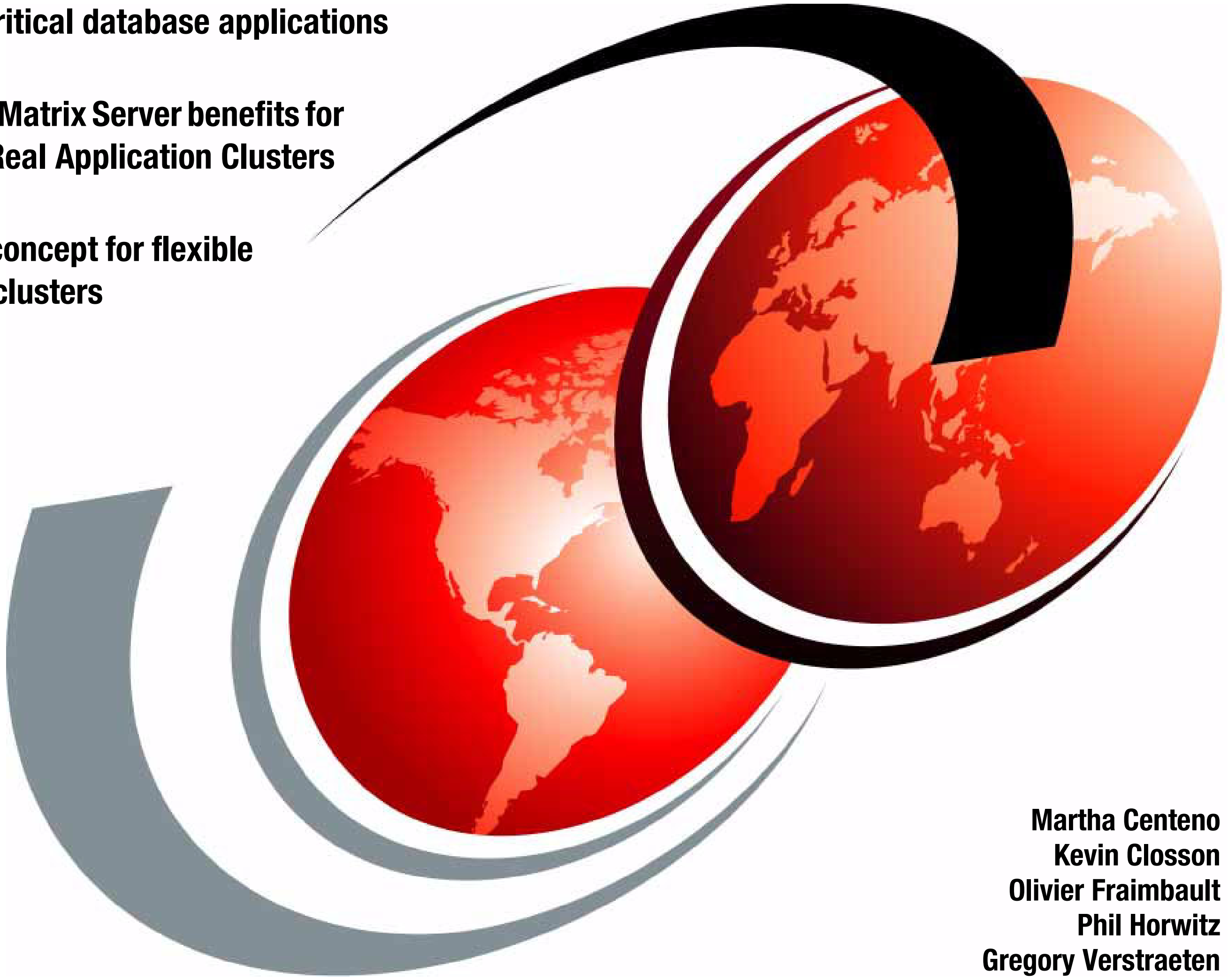
IBM eserver

Oracle9*i* Real Application Clusters and PolyServe Matrix Server

on IBM eserver xSeries and BladeCenter

- **Flexible database clusters for mission-critical database applications**
- **PolyServe Matrix Server benefits for Oracle9*i* Real Application Clusters**
- **Proofs of concept for flexible database clusters**



Martha Centeno
Kevin Closson
Olivier Fraimbault
Phil Horwitz
Gregory Verstraeten

Redpaper

ibm.com/redbooks

International Technical Support Organization

Oracle9i Real Application Clusters and PolyServe Matrix Server on IBM @server xSeries and BladeCenter

November 2004

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (November 2004)

This edition applies to Oracle9*i* Real Application Clusters 9.2.0.3, PolyServe Matrix Server 1.2.1, SUSE LINUX Enterprise Server 8 and Red Hat Enterprise Linux AS 2.1.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- @server®
- @server®
- Redbooks (logo) ™
- ibm.com®
- xSeries®
- BladeCenter™
- Chipkill™
- DB2®
- FlashCopy®
- Home Director™
- IBM TotalStorage Proven™
- IBM®
- Lotus®
- Notes®
- NUMA-Q®
- PartnerWorld®
- POWER™
- Redbooks™
- Sequent®
- ServerProven®
- Tivoli®
- TotalStorage Proven™
- TotalStorage®
- X-Architecture™

The following terms are trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Preface

How do I implement a database server that can dynamically scale to meet my changing IT requirements? How do I improve the response time and throughput of my database server? How do I deliver availability while still meeting my budget targets? How do I manage large user populations and high volumes of transactions, and still respond quickly to client requests?

These are some of the questions businesses ask as they try address the challenges of managing mission-critical database applications, data, and servers.

This IBM® Redpaper introduces the concept of flexible database clusters as a proven solution to these challenges. This type of cluster is built on the IBM @server® xSeries® servers or BladeCenter™ servers, and it features Oracle9*i* Real Application Clusters running on a PolyServe Matrix Server cluster file system.

Proofs of concept and the joint IBM and Oracle certification process for assuring the interoperability of flexible database clusters are presented in this publication. It also features two case studies that demonstrate the benefits of the PolyServe Matrix Server for Oracle9*i* Real Application Clusters when implemented on xSeries or BladeCenter servers.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Raleigh Center.

Martha Centeno is an Advisory Software Engineer in the xSeries Performance Lab in Research Triangle Park in North Carolina. Over the past 10 years as a member of the performance group, she has been involved in the publication of leading benchmark results of xSeries servers in the SAP and TPC-W environments. Her current focus is on the performance analysis of Oracle RAC on xSeries and BladeCenter servers in Linux® environments with focus on the effect of interconnect technologies.

Kevin Closson is a Chief Software Architect with PolyServe, Inc. His 17-year career has included Engineering, Technical Support, and Application Development positions specializing in Oracle and Clustered platforms. Prior to his tenure with PolyServe, Kevin held positions within IBM, Sequent® Computer Systems, and Veritas, where his main engineering focus was throughput and scalability enhancements of the Oracle server on high-end UNIX® SMP and Clustered Systems. Kevin holds patents in shared memory (SM) locking algorithms and database caching methods. Kevin is a frequent speaker at industry trade shows such as Oracle OpenWorld, IBM PartnerWorld®, IOUG-A, and regional Oracle User Groups. In addition to book collaborations, Kevin's written works have appeared in *Oracle Magazine*, *Oracle Internals Magazine* and *SELECT.*

Olivier Fraimbault is an Advanced Technical Support - Products and Solutions Support Center Independent Software Vendor Leader and works at IBM Montpellier, France.

Phil Horwitz is a Senior Engineer with over 14 years of experience in the xSeries Server Performance Lab. His responsibilities include the measurement and analysis of the performance of xSeries servers versus the competition in real-world application environments. His current projects include the analysis of xSeries servers in database cluster and scale-out environments. He has presented xSeries Server Performance analysis to customers and at trade shows worldwide. He earned a Bachelor of Science degree in Electrical Engineering from the University of Florida and Master of Computer Engineering from Florida Atlantic University.

Gregory Verstraeten graduated with a degree in Computer Sciences (from Technologie de Compiegne-France) and joined IBM in 1999 as an IT Specialist at the IBM-Oracle Joint Solution Center located in Montpellier-PSSC. Gregory has developed strong skills in xSeries and Oracle solutions by delivering Level 2 technical support, customer competitive benchmarks, and education. Since 2002, Gregory has been responsible for the OASB (Oracle Application Standard Benchmark) on xSeries, and has delivered top results for x440, x445, and x365 clusters. He has been involved in the development of two IBM Redbooks™ and has written several white papers and IBM Redpapers. Gregory has participated as a speaker in several international conferences across the globe (U.S.A, Australia, Japan, and France).

Thanks to the following people for their contributions to this project:

Ross Gale, Director, Business Development
PolyServe Inc Boston

John McAbel, WW Solutions Marketing Manager
IBM Beaverton

Rufus Credle, ITSO Project Leader
IBM Research Triangle Park

Janelle Adams
PolyServe Inc

Forsyth Alexander
Tamikia Barrow
Jeanne Tucker
ITSO, Raleigh Center

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, and customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this IBM Redpaper or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HQ7 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

1

Flexible database clusters

This chapter introduces the concept of flexible database clusters as a method for addressing the challenges inherent in the management of mission-critical database applications. The chapter provides an overview of the hardware and software components selected for building a flexible database cluster. These components include:

- xSeries and BladeCenter servers
- IBM TotalStorage® products
- Oracle9*i* Real Application Clusters running on a PolyServe Matrix Server cluster file system

1.1 Oracle9*i* Real Application Clusters

Relational databases have become the primary repositories for businesses' mission-critical data. As such, database servers must support both large user populations and high volumes of transactions, as well as respond quickly to client requests. The database has become the key performance and scalability factor for large-scale online transaction processing (OLTP) and data warehousing (DW) systems.

With Oracle9*i* Real Application Clusters, a database can be horizontally scaled; that is, the database workload can be distributed across several cost-effective smaller servers (a cluster) instead of being located on a single large system. IT departments, however, face multiple challenges when implementing a database cluster:

- Scalability: How do I implement a database server that can dynamically scale to meet my changing IT requirements? How do I provision capacity so that I can handle increased demand? How do I scale the system cost effectively?
- Performance: How do I improve the response time and throughput of my database server?
- Availability: How do I deliver five nines? How do I build redundancy into the system so that there is no single point of failure? How do I minimize downtime in the event of a system crash or hardware malfunction? How do I deliver availability while still meeting my budget targets?
- Manageability: How do I manage the system with my same level of resources? How do I centralize computing resources so that I can realize operating efficiencies? How do I decouple the provisioning and management of storage and computing capacity?

Flexible database clusters are the solution to these challenges. This type of cluster is built on the xSeries or BladeCenter servers, a fault-resilient Fibre Channel storage area network (SAN), and the PolyServe Matrix Server software, and it features Oracle9*i* Real Application Clusters running on a PolyServe Matrix Server cluster file system.

The xSeries and BladeCenter servers are ideal platforms for deploying Oracle9*i* Real Application Clusters. The newest xSeries servers utilize the groundbreaking IBM Enterprise X-Architecture™ technology to provide outstanding advances in performance, scalability, and availability that deliver value to Oracle customers.

The IBM TotalStorage DS4400 (formerly IBM TotalStorage FAStT700) Storage Server, with its 2 Gbps technology for faster response times, allows large amounts of highly available disk storage to be connected to the cluster. This solution accommodates a variety of storage requirements and is cost effective and scalable.

Oracle9*i* Real Application Clusters provide a cost-effective path for horizontally scaling the database server tier, managing it efficiently, and delivering five nines of availability (less than six minutes of downtime a year).

PolyServe Matrix Server greatly simplifies the deployment and management of an Oracle9*i* Real Application Clusters database on a SAN. It has been specifically designed to complement the Oracle9*i* Real Application Clusters architecture and seamlessly scales out with Oracle9i running on it.

PolyServe Matrix Server provides full use of all Oracle9*i* features, including a shared Oracle Home, Oracle Disk Manager (ODM), and Oracle Managed Files (OMF), and it dramatically improves the speed of Extract, Transform, and Load (ETL) operations.

The PolyServe Matrix Server cluster file system enables servers to concurrently access and share files on the SAN. It provides performance that is equal to shared raw devices with the added manageability of a file system.

1.2 Advantages of xSeries

The xSeries servers range from affordable, workgroup systems to enterprise-class systems providing true choice for the client. xSeries servers are built on IBM Enterprise X-Architecture™ technology, a blue-print for building IBM mainframe muscle into Intel® processor-based systems.

xSeries servers deliver revolutionary advancements to input/output (I/O), memory, and processors. They also offer a pay-as-you-grow approach to the Intel Xeon, Intel Itanium2, AMD Opteron, and now Intel EM64T. xSeries systems are designed to match nearly any application workload environment, from the entry-level, x205 uniprocessor to the enterprise-powered, up to 32-way x445 for a complete scale-out or scale-up solution.

IBM Enterprise X-Architecture™ provides the infrastructure for continuous computing in enterprise-class database environments. With features that highlight failure masking, failure avoidance, and minimized outages, end users can expect higher mean time between failures (MTBF) and decreased mean time to repairs (MTTR).

IBM Enterprise X-Architecture™ provides value to Oracle customers by improving database performance, availability, scalability, and manageability as follows:

- Optimized memory access from the x445 XceL4TM Cache can add up to 64MB (per 4-way) of high-speed dedicated cache memory and provide high-speed communication between memory and processors. This can lead to a10-15% improvement in performance.
- Active Memory, Chipkill™, Hot Swap, and Hot Add increase uptime through additional memory reliability with operating-system-independent memory mirroring and continuous system operations, thereby increasing availability.
- XpandOnDemand™ and its modular node building blocks for pay-as-you-go easily accommodate growing database solution environments without the need to purchase costly infrastructure up front provide increased scalability.
- IBM Director and its integrated remote management capabilities provide proactive and predictive tools to help increase availability and ease administration for database servers. The result is greater manageability.

IBM works closely with operating system vendors in both the Windows® and Linux environments to provide outstanding performance and robust functionality for tested and validated configurations. xSeries servers support a range of offerings from select products loaded with the Windows operating system for clients who desire one-stop shopping to the distribution of the Datacenter Edition for clients that demand scalability and reliability.

IBM also works closely with the leading Linux distribution companies: Red Hat, Inc., SUSE LINUX AG, and Turbolinux, Inc. The range of Linux offerings available on xSeries servers gives customers the flexibility to choose the optimum Linux implementation. The mission of the IBM Linux Technology Center's (LTC) is to leverage IBM resources to accelerate the growth of Linux as an enterprise operating system and to work as a trusted, valued member of the Linux community. More information about the Linux LTC may be found at this Web site:

http://www.ibm.com/linux/ltc

The IBM LTC contributed key enhancements to the Linux 2.5/2.6 kernels in the areas of:

- Scalability: IBM LTC provided 8-way and 16-way symmetric multiprocessing (SMP), O(1) scheduler, improved resource locking, and per-CPU timers, counters, statistics, and data structures.
- Virtual Memory Manager enhancements: IBM LTC provided support for 32 GB memory on IA32, large page (4 MB/2 MB) VM support, RMAP (reverse physical to virtual address mapping), and large page in-memory file system support.
- I/O enhancements: IBM LTC provided large-block raw I/O, asynchronous I/O, and vectored block/raw I/O.
- Non-Uniform Memory Access (NUMA): IBM LTC provided NUMA topology infrastructure and performance enhancements, and a NUMA-aware scheduler.

These enhancements to Linux have allowed xSeries to reach unforeseen levels of performance for software solutions running on Linux. An example of this performance can be seen in the leading results xSeries has posted on the Oracle Applications Standard Benchmark (OASB). A quick look at these results shows that xSeries has maintained the leading clustered Linux result for Oracle9*i* Real Application Clusters implementations since November 2003. It is likely xSeries will continue this history of leading the industry. OASB results are posted and maintained by Oracle Corporation at their Web site:

http://www.oracle.com/apps_benchmark/html/results.html

The entire xSeries product line features advanced servers that help lower costs, speed innovation, and solve problems rapidly with improved efficiency. These Intel and AMD processor-based servers offer solutions that feature hands-off reliability through mainframe-inspired technologies and smart management tools.

1.3 Advantages of SAN technology

IBM TotalStorage SAN products deliver scalable hardware and software storage solutions that are designed to be resilient and modular, along with worldwide services and support. In short, IBM is a world leader in storage technology, services, and support and is continually delivering new technologies to help improve the way clients acquire and manage their storage infrastructure.

With the complete IBM TotalStorage product line, IBM can provide a solution to help address the unique storage requirements of today and those of the future. Solutions range from affordable, stand-alone tape drives to others that are designed for e-business on demand™ and applications with unpredictable growth requirements.

The IBM TotalStorage DS4000 Series (formerly IBM TotalStorage FAStT) is a cost-effective, scalable, flexible mid-range disk system that can help small and large enterprises with their storage requirements. The IBM TotalStorage DS4000 Series family of products have been enhanced with the following features to complement enterprise disk system offerings:

- DS4000 Storage Manager V9.10
- An enhanced remote mirror option
- DS4100 option for larger capacity configurations
- Support for EXP100 serial ATA expansion units that can be attached to DS4400s.

The IBM TotalStorage DS4400 Series is a full Fibre Channel SAN solution designed to provide the necessary performance at the right price. Large amounts of data can be consolidated with 2Gbps technology for quicker, more efficient access and management. The scalability design of the IBM TotalStorage DS4400 Series is an example of how IBM

TotalStorage products adhere to the commitment IBM has made to the pay-as-you-go philosophy that allows clients to add to their initial investments. The DS4400 scales from 36 GB to greater than 32 TB of storage using 16 IBM TotalStorage DS4000 EXP700 expansion enclosures; the actual usable capacity will vary based on the redundant array of independent disks (RAID). Each expansion enclosure is designed to support as many as 14 2Gbps Fibre Channel disk drives. Moreover, you can select the appropriate RAID level (from RAID levels 0, 1, 3, 5, and 10) to match an application or suit particular needs, helping customers extend and leverage their investments now and in the future.

DS4000 Storage Manager improves the flexibility of a client's rapidly changing storage needs and integrates with storage infrastructure management software to centralize volume, and device management. Clients can use this tool to manage multiple DS4400s from a single console. The IBM TotalStorage DS4000 Storage Manager offers the following tools and features:

- FlashCopy®: This tool creates a single point-in-time copy of all stored data quickly to support uninterrupted performance of normal SAN operations.
- VolumeCopy: This optional feature can be used to create a full-volume copy of a point-in-time logical drive generated by FlashCopy on a single DS4000.
- Dynamic Volume Expansion: With this tool, users can resize logical volumes, typically without interrupting other logical volumes.
- Enhanced Remote Mirroring (supported with v9.1): This feature includes Global Mirror with Asynchronous Write-order Consistency, which is critical for mirroring multi-logical unit number (LUN) applications, Global Copy with Asynchronous, and Metro Mirror with Synchronous.

IBM TotalStorage SAN products provide the connection of servers and storage devices into high-availability storage networks, supporting scalability to help address the demands of rapid and unpredictable growth.

1.4 Advantages of Oracle9*i* Real Application Clusters

The advantages of Oracle9*i* Real Application Clusters are:

- Scale-out versus scale-up

  Oracle9*i* Real Application Clusters offers users a cost-effective means for horizontally scaling the database server tier, managing it efficiently, and to delivering five nines of availability (less than six minutes of downtime a year). Oracle9*i* Real Application Clusters is a shared disk architecture. All nodes running the Oracle server have shared read/write access to a single shared copy of the database on disk.

  To maintain coherency between instances, the Oracle9*i* Real Application Clusters server relies on its own distributed lock manager infrastructure. This infrastructure is called Global Cache Services. It uses high-speed server-to-server interconnections to communicate among Oracle instances and to ensure data integrity. The Oracle implementation of this architecture has matured through 10 years of data center deployment on UNIX systems.

- Cache Fusion performance

  Oracle9*i* Real Application Clusters includes a new technology called Cache Fusion. Each Oracle9*i* Real Application Clusters database instance maintains a database block buffer cache that is a component of the Shared Global Area (SGA). Retrieving information from the SGA is much faster than retrieving that information from disk.

With Cache Fusion, Oracle processes can retrieve information from their own SGA and from the SGA of any other node in the cluster without going to disk. Because obtaining a piece of information from another node's memory is at least 10 times faster than retrieving it from disk, Cache Fusion represents marked progress in the performance of Oracle on clustered systems.

1.5 Deployment options for Oracle9*i* Real Application Clusters

Oracle9*i* Real Application Clusters can be deployed in three scenarios:

- On shared raw devices
- On a network-attached storage (NAS) appliance
- On a clustered file system

Although shared raw devices have historically been used and NAS devices have recently been supported, clustered file systems provide ease of management and performance benefits not available with other approaches. The PolyServe Matrix Server clustered file system delivers both manageability and performance benefits and enhances them with the ability to use the full power of the new Oracle9i features. This section examines the three scenarios and demonstrates the benefits of the clustered file system.

1.5.1 Using raw devices with Oracle9*i* Real Application Clusters

Raw devices are unformatted disks or disk partitions. When deployed on raw devices, the Oracle9*i* Real Application Clusters cluster uses direct (unbuffered) I/O to read and write blocks, thereby bypassing the native file system. Traditionally, a clustered Oracle model such as Oracle Parallel Server required that a raw shared device be used. This restriction arose for two reasons:

- The basic requirement for Oracle shared storage is that I/O transfers must be direct not cached at the operating system cache level. Previously, there were no clustered file systems that offered direct I/O. However, this requirement was naturally achieved when using raw disk partitions.
- Raw partitions were thought to be an absolute requirement for performance.

Today, the raw shared device restriction is obsolete. Clustered file systems such as those provided by PolyServe Matrix Server now support both direct I/O and asynchronous I/O and provide performance equal to that of raw partitions.

Problems with using raw devices and Oracle9*i* Real Application Clusters

Although Oracle9*i* Real Application Clusters can be used with shared raw partitions, there are several drawbacks to this type of configuration:

- Existing database files must be migrated. Typically, SMP (non-clustered) databases reside in regular file systems. When Oracle9*i* Real Application Clusters is deployed on shared raw partitions, the DBA must migrate the database files from the non-CFS file systems to raw devices.
- Certain file types cannot be shared. Only Oracle9*i* Real Application Clusters data such as data files, redo logs, and control files can be stored on the raw devices. The Oracle9*i* Real Application Clusters binaries and configuration files (Oracle Home) must be stored in the local file system of each node in the cluster. Oracle9*i* has over 100,000 files in each Oracle Home. An administrator must individually configure and maintain each node in the Oracle9*i* Real Application Clusters cluster, making sure that the same patches are applied

on each node and that the nodes are configured in the same manner. This process can become unwieldy as nodes are added to the cluster.

- Key Oracle9i features cannot be used. Using shared raw devices prevents administrators from using several key features of Oracle9*i*:

  - OMF

    With OMF, Oracle9*i* provides new and improved functionality to manage the physical files associated with tablespaces, redo logs, and other file features. This capability is not available when an Oracle9*i* Real Application Clusters cluster is run on raw devices. If additional storage capacity needs to be provisioned without OMF, the administrator must monitor the storage on each node and perform the task manually.

    When OMF can be used, the administrator does not need to constantly increase the storage capacity of each node in the cluster. Instead, OMF monitors the storage and increases it at the time the data files expand for an Oracle9*i* Real Application Clusters instance. Additionally, if desired, OMF can relieve the DBA of many of the mundane tasks associated with data files. For example, OMF can completely eliminate the need to name data files.

  - Full ETL capabilities

    Oracle9*i* has new and improved ETL capabilities that allow a data warehouse to be loaded directly from external files called External Tables. When shared raw devices are used, DBAs cannot take advantage of these features. Also affected is the placement of the regular files associated with Import/Export and SQL*Loader. The question becomes which node of the cluster the DBA will use for these files. Choosing only one node for these files creates an inherent single point of failure.

    With External Tables, the power of Oracle Parallel Query Option can be used to rapidly transform and load decision support data in a single, parallel step. However, this is only possible with a clustered file system.

  - Inferior I/O fencing

    An arcane approach to I/O fencing must be used when Oracle9*i* Real Application Clusters is deployed on shared raw devices. Fencing ensures that a server cannot perform I/O to the database when it is no longer a viable member of the cluster.

    The fencing method used with shared raw devices is to automatically turn off the server. This approach, known as STOMITH, an acronym for shoot the other machine in the head, has several problems. There may be applications running on the fenced server that have nothing to do with Oracle9*i* Real Application Clusters. Furthermore, the problem that caused the fencing might persist across reboots, making diagnosis extremely difficult.

    By contrast, PolyServe Matrix Server provides fabric fencing, which excludes the failed server from the cluster and disables its access to the SAN. The fenced server cannot threaten data integrity but is kept functioning as an isolated unit, which allows it to be recovered more quickly.

  - Data duplication and back-up overhead

    Because Oracle Home and certain other files cannot be stored on shared raw devices, there is needless duplication of storage. Duplicate direct attached storage (DAS) pools result in the deployment of more megabytes of storage, which increases storage hardware costs. Additionally, each server node with DAS represents an island of direct-attached disk that must be backed up by the storage administrator.

    If all storage is centralized, backup operations can be streamlined. Finally, centrally managed storage is easier and less costly to manage.

1.5.2 Limitations of deploying Oracle9*i* Real Application Clusters with NAS

The following limitations exist when Oracle9*i* Real Application Clusters is used with NAS:

- Cluster size limitations

  NAS devices fulfill their promise of being easy to deploy and manage. At the same time, their capability and flexibility are extremely in the Oracle9*i* Real Application Clusters environment. In terms of node count, NAS devices are explicitly certified to support clusters of only two nodes. By contrast, PolyServe Matrix Server is explicitly certified to support clusters of four nodes, the highest node count certification test available. In some cases, PolyServe Matrix Server has been proven to support clusters of up to 16 nodes.

- Throughput limitations

  NAS devices are limited in terms of their I/O bandwidth relative to the number of spindles in each filer, a key determinant of Oracle9*i* performance. A high-end filer offers only six 1Gb Ethernet connections, providing no more than approximately 768 megabytes of theoretical bandwidth. Upgrades to these filers require doubling capacity to a clustered filer configuration, which adds to the expense of the configuration.

  Also, the aggregate bandwidth per Ethernet connection on an NAS device is much lower than the aggregate bandwidth per Fibre Channel connection on a SAN. Because PolyServe Matrix Server uses a SAN, the number of Fibre Channel connections is much higher. Also, the speed of each connection is much higher and has much lower latency, important factors in optimizing Oracle9*i* performance. When additional bandwidth is needed in a PolyServe Matrix Server environment, it can be added without affecting other nodes in the cluster.

- Inferior I/O fencing and server recovery

  The I/O fencing model available for NAS devices is STOMITH, the same fencing used by shared raw devices. STOMITH automatically turns off the affected server. This fencing method has several problems. First, there may be applications running on the fenced server that have nothing to do with Oracle9*i* Real Application Clusters. Also, the problem that caused the fencing might persist across reboots, making diagnosis extremely difficult.

  By contrast, PolyServe Matrix Server provides fabric fencing, which excludes the failed server from the cluster and disables its access to the SAN. The fenced server cannot threaten data integrity but is kept functioning as an isolated unit, which allows it to be recovered more quickly.

1.5.3 Cluster file systems: comparing PolyServe Matrix Server with OCFS

Oracle offers the Oracle Cluster File System (OCFS) for basic Oracle data file usage. It is available in the open source community. By contrast, the PolyServe Matrix Server cluster file system offers enhanced Oracle support that is not featured with OCFS. It also offers broader data center support than OCFS. Its general purpose clustered file system also supports:

- Non-Oracle applications such as Network File Services
- Web serving of rich media types
- Custom applications requiring high-performance read and write access to file-based applications.

PolyServe Matrix Server has been tested under some of the most stringent quality assurance processes in the industry and is proven to offer linear scalability and solid reliability for Oracle clusters of up to 16 nodes.

Table 1-1 compares the features of PolyServe Matrix Server and Oracle's Cluster File System (OCFS) for Linux.

Table 1-1 Comparison of PolyServe Matrix Server file system and OCFS file system

Feature	PolyServe Matrix Server 2 (all releases) or later	OCFS 1.0
Oracle-specific functionality:		
Host Oracle tablespaces in the file system	Yes	Yes
Support for Direct I/O	Yes (integrated)	If provided by OS (O_DIRECT compile-time flag)
Support for shared Oracle Home	Yes	No
Support for Oracle Disk Manager (ODM)	Yes	No
Support for Parallel ETL for Warehousing	Yes	No
General purpose file system:		
Support for NFS and CIFS	Yes	No
Support for Web and application serving	Yes	No
Support for other databases	Yes	No
File system resizing	Yes	No
Dynamic I-node allocation	Yes	No
Dynamic user and group privileges	Yes	No
File system API available	Yes (flock)	No
Node specific files	Yes (CDSL)	No
Data availability and coherency:		
Distributed Lock Manager Architecture	In-memory	Disk-based
I/O Fencing (Fabric)	Yes	No (STOMITH)
Support for third party online backup solutions	Yes	No
Multi-path I/O	Yes	No
Cluster management:		
Integrated clusterware for membership and wellness	Yes	No
Consistent device naming across cluster	Yes	No
Storage and fabric management:		
Discovery and mapping of LUNs to the file system	Yes	No
Integrated fabric management with Brocade	Yes	No
Integrated fabric management with McDATA	Yes	No
Auto-discovery of devices in the fabric	Yes	No

1.6 PolyServe Matrix Server benefits

The PolyServe Matrix Server clustered file system provides the following benefits for Oracle9*i* Real Application Clusters:

- Simplified cluster deployment and management

  The PolyServe Matrix Server infrastructure allows companies to build highly available, scalable, and manageable database clusters with Oracle9*i* Real Application Clusters and storage on an existing SAN. The PolyServe Matrix Server clustered file system is fully journaled, recoverable, and highly optimized for Oracle9*i* Real Application Clusters. Its uses include not only data files, but also any files associated with Oracle.

  The performance of the PolyServe Matrix Server clustered file system is equal to shared raw devices but with the manageability of a file system. PolyServe Matrix Server has been specifically designed to complement the Oracle9*i* Real Application Clusters architecture and seamlessly scales out with Oracle9*i* Real Application Clusters running on it.

- I/O performance

  PolyServe Matrix Server supports shared-block access with full data integrity and blocking. Furthermore, it does not redundantly cache or buffer I/O, and unlike most file systems, it does not enforce locking for block-level access. PolyServe Matrix Server allows Oracle9*i* Real Application Clusters to perform direct I/O against the disk. It also enables distributed lock management at the database level, where system-wide performance can best be optimized.

- Shared files

  All Oracle9*i* Real Application Clusters file types are shared. PolyServe Matrix Server enables multiple servers to concurrently access and share files on a SAN. It also guarantees data integrity by implementing a sophisticated, distributed-lock management system. This system arbitrates access to regular file data while offering Oracle9*i* unencumbered direct I/O to database files. The shared data access provides significant manageability benefits to the administrator.

- Full use of Oracle9*i* features

  PolyServe Matrix Server allows administrators to take full advantage of all Oracle9i features:

  - Shared Oracle Home

    All nodes in the cluster can use a single shared Oracle Home. The full Oracle9*i* Home directory has over 100,000 files. With PolyServe Matrix Server, one copy of these files can be shared across all nodes in the cluster, enabling administrators to configure and run an Oracle9*i* Real Application Clusters cluster from a single unified application image. Administrators are no longer burdened with the need to maintain a separate Oracle Home on each node in the cluster and keep their files synchronized.

  - ODM

    ODM is an Application Programming Interface (API) created by Oracle. It replaces the non-optimized I/O and file management API calls traditionally available on UNIX and Linux systems. ODM offers the following features:

    - Asynchronous, direct I/O
    - Specialized I/O priorities
    - Fully optimized scatter/gather I/O
    - Atomic file creation/deletion
    - Cluster-wide file keys that eliminate all risk of file mismanagement in complex clustered environments

PolyServe Matrix Server supports ODM with MxODM monitoring. MxODM is a feature-rich cluster-wide I/O monitoring package that allows monitoring at the cluster, database, instance or node level. MxODM monitoring delivers numerous features:

- A clear single-image, real-time view of desirable statistics such as Log Writer and Database Writer I/O operations and latency
- Parallel query activity
- Table scan I/Os
- Temp segment writes
- Read/write percentages
- Synchronous I/O latency, a feature not available without ODM

– OMF

PolyServe Matrix Server allows administrators to have full OMF capability in an Oracle9*i* Real Application Clusters cluster. OMF features new and improved methods for monitoring and managing data files. These methods greatly reduce the complexity of creating, configuring, and managing the storage for an Oracle database. Administrators running Oracle9*i* Real Application Clusters with a shared raw device cannot use OMF functionality.

– ETL

PolyServe Matrix Server dramatically improves the speed of ETL operations on Oracle9*i*. The ETL capabilities provided with Oracle9*i* allow a data warehouse to be loaded directly from External Tables. In a cluster, this would be a bottleneck if only one server could read and process the External Table. However, with PolyServe Matrix Server, the Oracle Parallel Query Option can process the external file in parallel, dramatically reducing the time it takes to load a data warehouse.

- Improved availability

  PolyServe Matrix Server provides sophisticated clusterware that complements the high availability features of Oracle9*i* Real Application Clusters. PolyServe Matrix Server seamlessly integrates with the Oracle Cluster Management Services (OCMS). Moreover, PolyServe Matrix Server can be extended with custom scripting to handle even more sophisticated high availability scenarios.

Table 1-2 summarizes the scalability, availability, and manageability benefits that PolyServe Matrix Server provides for Oracle9*i* Real Application Clusters deployments.

Table 1-2 PolyServe Matrix Server Matrix Server benefits for Oracle 9i Real Application Clusters

PolyServe Matrix Server feature	**Benefits for Oracle9*i* Real Application Clusters deployment**
Independent scale out of computing and storage Assets	Enables dynamic provisioning of capacity to meet changing load requirements of Oracle9*i* Real Application Clusters. Computing and storage systems can be added independently.
I/O optimizations improved over raw devices	Uses Oracle Disk Manager to deliver superior performance over raw devices. Eliminates the need for write locks and metadata updates and provides asynchronized, direct I/O.
Better manageability than raw devices	Appears as a standard file system to third-party utilities and tools. Administrators can use familiar tools and utilities to provision, backup, and restore information. No new tools are needed.
Single shared Oracle Home	Reduces management cost of maintaining and configuring a highly available cluster of many small Oracle9*i* Real Application Clusters servers. Helps make many small servers manageable with the same cost benefits as a large UNIX server.

PolyServe Matrix Server feature	**Benefits for Oracle9*i* Real Application Clusters deployment**
Enables the use of OMF with Oracle9*i* Real Application Clusters	Enables full use of the OMF capability with an Oracle9*i* Real Application Clusters cluster. OMF delivers significant manageability benefits to Oracle administrators. OMF cannot be used with a shared raw device.
Improves the performance of Oracle9i ETL functionality	Allows all nodes in an Oracle9*i* Real Application Clusters cluster to participate in parallel ETL processing. A cluster of Oracle9*i* Real Application Clusters nodes can load the data much faster than a single node processing the entire workload by itself. External Tables combined with Parallel Query can reduce data movement by 50% during a typical ETL procedure.
Supports OCMS	Seamlessly integrates with OCMS. Provides complete high-availability, failover, and reliability through improved I/O fencing.
Seamless integration with the ODM	Tightly couples PolyServe Matrix Server and Oracle9*i*. Ensures improved performance, full data integrity, and forward compatibility with Oracle9*i* Real Application Clusters.
Centralized management of storage assets	Centralized file system management simplifies SAN administration. In fact, SAN-based architectures are less costly to manage than DAS.
Multi-path I/O	Supports the ability to use multiple data paths to shared storage. Supports multiple Host Bus Adapters (HBAs) per server with connection to different switches. If a data path is lost,PolyServe Matrix Server seamlessly fails over all data traffic from one HBA to another, while maintaining data coherency and integrity. The administrator can create a fully redundant, highly available switched SAN fabric with no single point of failure.
Guaranteed data integrity	Data is protected from corruption or error because the file system enforces a distributed locking and cache coherency mechanism that guarantees data integrity and validity for file access. PolyServe Matrix Server provides direct I/O for block level access and guaranteed data integrity.
No single point of failure	PolyServe Matrix Server can endure N-1 server failures and continue to process requests.

2

IBM certification

This chapter describes the certification process conducted jointly by IBM and Oracle to verify and assure the interoperability of the Oracle9*i* Real Application Clusters (RAC) database server using PolyServe Matrix Server on an IBM @server xSeries cluster running Red Hat Enterprise Linux AS 2.1 or SUSE Linux Enterprise Server (SLES) 8.

2.1 Integration testing

The IBM/Oracle Joint Solution Center performed integration testing of PolyServe Matrix Server running on an IBM cluster with Oracle9*i* Real Application Clusters at IBM labs in Montpelier, France. Three configurations were tested:

- An xSeries 360 cluster, Red Hat Enterprise Linux AS 2.1, PolyServe Matrix Server 1.2.1, and Oracle9*i* Real Application Clusters version 9.2.0.3
- An xSeries 440 cluster, Red Hat Enterprise Linux AS 2.1, PolyServe Matrix Server 1.2.1, and Oracle9*i* Real Application Clusters version 9.2.0.3
- An xSeries 440 cluster, SLES 8, PolyServe Matrix Server 1.2.1, and Oracle9*i* Real Application Clusters version 9.2.0.3

The main goals of the testing were:

- To verify the interoperability of all software and hardware elements usually needed to meet the requirements for Oracle9*i* Real Application Clusters deployments in business-critical environments.
- To document the required implementation steps, any potential operability issues identified during the process, and appropriate solutions.

The remainder of this chapter summarizes the integration testing. For detailed descriptions of the cluster implementations and tests, see the following white papers:

- *Installing and Testing Oracle9i* Real Application Clusters *on IBM x360 and RHLAS*
- *Installing and Testing Oracle9i* Real Application Clusters *on IBM x440 and RHLAS*
- *Installing and Testing Oracle9i* Real Application Clusters *on IBM x440 and SUSE*

The papers are available at the following Web site:

http://www.polyserve.com/products_literature.html

2.2 Cluster implementation

Each cluster was implemented as follows:

1. Planning, including obtaining the necessary drivers, firmware, and software
2. Cabling servers, Fibre Channel fabric, and storage arrays
3. Installing the operating system and building the kernel
4. Installing PolyServe Matrix Server
5. Installing Oracle9*i* Real Application Clusters and creating a shared Oracle Home

After implementation, each cluster ran perfectly.

2.3 Tests performed

The purpose of these tests was twofold:

- To verify that the behavior of the Oracle9*i* Real Application Clusters database server using PolyServe Matrix Server on an xSeries cluster running Red Hat Enterprise Linux AS 2.1 or SLES 8 was correct.
- To confirm that the operation of the Oracle database was not adversely affected by changes or upgrades to any of the hardware or software components of the operating environment.

The following Oracle9*i* Real Application Clusters and related functionality were tested:

- Startup and shutdown under heavy load.
- Multi-instance stress tests
- Multi-instance RAC functionality, including free list groups, resource inquiry, locking, and basic functionality in a RAC environment
- Multi-instance backup
- Multi-instance recovery involving crash recovery
- Multi-instance recovery with multiple-instance deaths
- Multi-instance quiescence I0O
- Multi-instance startup and shutdown under various error conditions
- Multi-instance datafile and tablespace management
- Multiple LMS failure tests
- Scalable message memory stress tests with multiple LMDs
- Parallel Query and Parallel data life cycle management (DLM) tests

The tests were completed successfully on all three configurations.

2.4 Conclusions

After a straightforward PolyServe Matrix Server installation, the xSeries servers running Red Hat or SLES were ready for the installation of Oracle9*i* Real Application Clusters. Once the installation was complete, it quickly passed the full set of Oracle database-related test cases.

Manageability is critical in the deployment of large production databases in a clustered systems environment. It would be nearly impossible to deploy and manage a large Oracle9*i* Real Application Clusters database on a cluster of servers without the following robust cluster file system capabilities of PolyServe Matrix Server:

- Storing Oracle components, Oracle database files (for example, Oracle tablespaces), and Oracle applications
- Enabling the key features of Oracle9*i* Real Application Clusters, such as Oracle Shared Home, Oracle Managed Files (OMFs), External Tables, and parallel ETL

The PolyServe Matrix Server cluster file system also provides all servers in a cluster with scalable concurrent read and write access to all data on a SAN.

PolyServe Matrix Server provides all of the benefits of deploying Oracle9*i* Real Application Clusters on a file system with the performance associated with raw devices. PolyServe Matrix Server also extends the high-availability features in Oracle9*i* Real Application Clusters. It provides comprehensive wellness capabilities for monitoring all system hardware and software components (for example, applications, middleware, servers, networks and file system storage). This is done with automated failover that eliminates single points of failure and increases system reliability and uptime.

The tests conducted by IBM emphasize the quality and uniqueness of PolyServe Matrix Server and its cluster file system capabilities for Oracle9*i* Real Application Clusters on Linux. After Linux has been installed, PolyServe Matrix Server provides all the benefits for Oracle9*i* Real Application Clusters deployments described in this paper, and is also easy to deploy, configure, monitor and manage.

One other benefit to this testing is that PolyServe Matrix Server was recognized as both IBM ServerProven® as well as IBM TotalStorage Proven™. Achieving this level of certification provides an end user with even more confidence that an Oracle-based solution using PolyServe Matrix Server will help to meet their performance and manageability IT goals.

3

Proofs of concept

This chapter presents proofs of concept for flexible database clusters implemented on IBM @server xSeries and BladeCenter servers. Comprehensive descriptions of the hardware and software components used to implement these flexible database clusters are provided.

3.1 Flexible database clusters

Modern clusters may rival the established UNIX- based server, but are they manageable? What are the ramifications of performance and availability at the application level? What is the impact on total cost of ownership (TCO)? Answering these questions is the focus of this section.

Oracle9*i* Real Application Clusters is designed to help build flexible, high-performance, highly available, clustered database solutions on Linux. Connecting such clusters to a fault-resilient Fibre Channel storage area network (SAN) lays the foundation for a computing infrastructure known as a flexible database cluster.

IBM and PolyServe have performed exhaustive proofs of concept to verify that flexible database clusters are the perfect platform for Oracle9*i* Real Application Clusters.

The proofs of concept include:

- A flexible database cluster based on the BladeCenter and PolyServe Matrix Server, with Oracle9*i* Real Application Clusters running on a PolyServe Matrix Server cluster file system. A complete analysis of this testing is presented in 3.2, “Flexible database clusters with BladeCenter” on page 18.
- A flexible database cluster based on xSeries servers and PolyServe Matrix Server, with Oracle9*i* Real Application Clusters running on a PolyServe Matrix Server cluster file system. The key points from this proof of concept are described in 3.4, “Flexible database clusters with xSeries” on page 64.

3.2 Flexible database clusters with BladeCenter

The synergy of BladeCenter, PolyServe Matrix Server, and Oracle9*i* Real Application Clusters makes the flexible database cluster a powerful platform for supporting multiple applications. This section describes a proof-of-concept that validates the architecture and technology of the flexible database cluster and confirms that:

- PolyServe Matrix Server and Oracle9*i* Real Application Clusters perform extremely well on BladeCenter.
- The BladeCenter architecture and technology help provide an unparalleled high-availability platform for implementing flexible database clusters.
- IBM and PolyServe are leading the development of technology for scale-out computing.
- The architecture and technology of the flexible database cluster helps enable on-demand computing. Cluster nodes provide a pool of flexible resources for use among applications. The availability of Oracle9*i* Real Application Clusters is enhanced because nodes can be dynamically reprovisioned using PolyServe Matrix Server to cover the loss of another node.
- The flexible database cluster provides strong management tools such as PolyServe Matrix Server for performance and availability. A single large cluster is now easier to manage than many small clusters.
- A general-purpose cluster file system such as the one included with PolyServe Matrix Server provides a single-system feel and greatly enhances manageability. A shared Oracle home used by all nodes also simplifies management. Support is available for all database operations that require a file system.

- Improved manageability, scalability, expandability, availability, and asset utilization in a flexible database cluster configuration also can help dramatically lower TCO relative to a UNIX-based IT environment.

3.2.1 Clustering and Oracle9*i* Real Application Clusters

Since the release of Oracle9*i* Real Application Clusters during Oracle Open World Europe in June 2001, it has had the capability to scale a single application horizontally to clusters as large as eight nodes or greater without application modification. This scalability has allowed Oracle9*i* Real Application Clusters to gain mindshare within the clustered database community.

Another trend that has been gaining momentum is server consolidation, whereby large, flexible Intel® processor-based clusters run Linux under Oracle9*i* Real Application Clusters. Emerging clustered architectures such as the BladeCenter lend themselves to configuring large clusters in reasonable physical space at low cost. Connecting such clusters to a fault-resilient Fibre Channel SAN lays the basic foundation for the computing infrastructure known as a flexible database cluster.

Flexible database clusters are the perfect combination of server consolidation and Oracle9*i* Real Application Clusters. When compared to several small and separate clusters, each running Oracle9*i* Real Application Clusters, the management, availability, and performance characteristics offered by a large flexible database cluster are difficult to ignore.

Significant testing and proof of concept were required to advance the flexible database cluster concept. To that end, IBM and PolyServe joined forces to build a 14-node BladeCenter running SLES and then attached it to a formidable SAN configured with over 100 physical disk drives. This large cluster was the target of a series of tests that examined, in detail, the running and management of not only a single application, but also several applications simultaneously.

Unlike most Oracle9*i* Real Application Clusters studies to date, this testing was much more than the typical exercise of measuring throughput for a single application as nodes are added to the cluster. Those studies are valuable proof points; however, they lack information about the management aspects of large clusters, what happens when things go wrong, and how applications can benefit from dynamically shifting servers from under-utilized applications to applications where they are more useful.

To that end, the tests included turning off servers while running large numbers of connected Oracle users and dynamically shifting servers from one application to another. Critical measurements such as recovery times and throughput impact were also analyzed. The tests and measurements are described in detail in 3.4, “Flexible database clusters with xSeries” on page 64.

This chapter also includes performance information, but in a slightly different form than the norm. The flexible database cluster serves as a platform on which nodes can be shifted between applications without halting service to client systems. Therefore, tests were conducted to measure speedup to applications that were granted additional nodes dynamically.

3.2.2 Advantages of BladeCenter

BladeCenter is an innovative modular technology that can achieve outstanding performance density and affordable availability. BladeCenter is designed to gather computing resources into high-density enclosures that support hot-swappable, high-performance, 2-way and 4-way Intel processor-based and 2-way POWER™ processor-based blade servers.

The modular design of BladeCenter addresses customer issues such as space constraints, manageability, scalability, capacity, performance, cooling, and power. A data center can be deployed and managed from a 7U enclosure that integrates industry-leading Ethernet and Fibre Channel technology.

The BladeCenter chassis options offer clients the flexibility of choosing components for a comprehensive solution. The support for these options varies by server and chassis platform. They include:

- Cisco Systems Intelligent Gigabit Ethernet Switch Module: This component provides four external copper Gigabit Ethernet ports and 14 internal Gigabit Ethernet ports that interface with the blade centers through the BladeCenter midplane.
- Brocade Entry and Enterprise Switch Modules: These connect to SAN fabrics that consist of two switches with the entry switch and up to 239 switches with the enterprise module.
- Optical Pass-thru Module: This module provides each blade server with an unswitched, unblocked network connection. The blade servers must use either the integrated Gigabit Ethernet, or one these expansion cards: Gigabit Ethernet, Myrinet, or Fibre Channel.
- 4-port Gigabit Ethernet Switch Modules: These modules provide high-speed, Layer 2 Ethernet connections between each blade server and the outside network environment.
- Nortel Networks Layer 2-7 Gigabit Ethernet Switch Module: This component integrates advanced Ethernet functionality into the chassis.
- 2-port Fibre Channel Switch Module: This provides two Fibre Channel uplinks capable of supporting up to 2 Gbps.
- Redundant KVM/Management Module: This option provides remote management and connectivity for the most critical applications. Power Supply Modules provide both power and redundancy. The result is greater reliability and availability.

BladeCenter supports a 14 hot-swap, 2-way Intel Xeon processor (HS20) or a 7 hot-swap, 4-way Intel Xeon MP processor (HS40). With this support, it addresses the changing business needs of clients and meets the increased workload requirements driven by database solutions. The blade servers can be customized to meet end user needs by adding options such as:

- Myrinet Cluster Expansion Card working with the Optical Pass-thru Module: This option provides high-speed interconnect for high performance computing applications.
- Gigabit Ethernet Expansion Card expanding the Ethernet subsystem to enable more bandwidth: This card works with the Optical Pass-thru, Nortel Networks Switch, and 4-port Gigabit Ethernet chassis module.
- Fibre Channel Expansion Card: This card adds dual-port Fibre Channel connectivity to each blade and works with the 2-port Fibre Channel Switch Module.
- SCSI or IDE Hard Disk Drives (HDDs) support: This support includes ATA-100 HDD or SCSI drive support through the SCSI Storage Expansion Unit.
- SCSI Storage Expansion Unit: This option provides integrated mirroring capabilities and uses standard hot-swap Ultra320 SCSI HDDs.
- PCI I/O Expansion Unit: This unit supports legacy telecom network interfaces.

With IBM Director, customers can take advantage of a comprehensive remote management tool that can automate and simplify IT tasks from a single graphical console. Advanced management capabilities boost administrator productivity and reduce skill-level requirements, which can help reduce costs, improve overall productivity, and make administration easier.

Because installing a new blade server is fast and easy, expanding capacity for applications can be completed in a few minutes and not hours. IBM Director provides a rich and broad systems management capability with tools such as:

- IBM Deployment Wizard: With functions such as Automatic Chassis Discovery and Event Action Plans, this wizard makes infrastructure deployment easy.
- IBM Remote Deployment Manager: This tool can be used to create, maintain, and deploy system images from a single drag-and-drop user interface.

The following optional enhancements are available in the IBM Director Server Plus Pack:

- Capacity Manager to help predict future server bottlenecks and proactively alert administrators to minimize downtime
- Software Rejuvenation to predict pending software failures that could lead to costly down time
- System Availability to track and provide graphical views of system down/up time helping save labor costs
- Rack Manager to simplify rack hardware management

With today's IT challenges of containing costs, handling a shortage of skilled workers, and keeping up with the demands of e-business innovation, the BladeCenter offers a high-performance and manageable solution. BladeCenter helps businesses be responsive to customers, focus on core competencies, and adapt to variable cost structures with the flexibly and resiliency to meet any challenge.

3.2.3 Flexible database clusters concepts

The concept of a flexible database cluster is based on building a cluster that is large enough to support several databases. Instead of using one cluster for an online transaction database and two other clusters for decision support and development databases, a flexible database cluster can support all three.

For many years, large symmetric multiprocessing (SMP) systems have supported more than one Oracle database in what is routinely referred to as server consolidation. A trend that started in the late 1990s, server consolidation is the concentration of the workloads from several small SMPs into one large SMP. In the past, it was seen as a way to increase efficiency in the datacenter by reducing administrative overhead.

If consolidation adds value in an SMP environment, why would it not add value in a cluster environment? The answer is not simply that it does add value. In fact, consolidation also delivers more value in clustered environments than in SMP environments, particularly when supporting Oracle9*i* Real Application Clusters.

The Hard Partitions technology brought to market by Sun and Sequent has been recognized as valuable for consolidation. This technology offers fixed system provisioning to given applications within a single SMP system. With these systems, administrators can dedicate, for example, 16 processors to application A and another 16 processors to application B.

Dynamic partitioning (for example, logical partition, or LPAR, and virtual partition, or VPAR) is also viewed as critical for supporting consolidation of the flexibility it offers.

These SMP-oriented server consolidation systems support the dedication of CPUs to important workloads. Likewise, less important workloads (for example, untested applications still under development) are cordoned off to other CPUs to limit any disturbance they might cause to other applications. Arguably, clusters do this even better than SMPs with domains or partitions. After all, with SMP systems, there are usually many system components, such as memory controllers and I/O adapters, that are shared among partitions or domains. With clusters, however, an application running on nodes 1 and 2 does not share server-level resources with applications on nodes 3 and 4. The SAN is shared, but partitionable.

Server consolidation is beneficial and, because of architectural differences, clusters perform even better in some ways than single SMP systems. Why then are so many IT organizations supporting a small dedicated cluster for each application? If several different clusters all support Oracle9*i* Real Application Clusters based applications, why not consolidate?

Some consumer groups further credit server consolidation with IT savings through reductions in the cost of support and skills, capital equipment, and system management. Fewer systems require less management.

If consolidating simple, single servers into one large server yields IT savings, how much more would be yielded by consolidating complex clusters into one large cluster?

The flexible database cluster concept lowers administrative overhead and offers higher availability and on-demand scalability beyond that of several small clusters. It is an architecture well worth considering. A flexible database cluster is more than just a big cluster. The prime ingredients are systems software and deployment methodology.

3.2.4 Large database clusters with Oracle9*i* Real Application Clusters at a glance

The thought of assembling a 14-node cluster, much less managing one, conjures up visions of cabling and OS configuration nightmares. This mentality is likely rooted in the UNIX-based clustered systems of the 1990s. Clusters of that era were configured with very few nodes primarily due to limitations in Oracle Parallel Server.

Oracle9*i* Real Application Clusters has changed that. The scalability and availability characteristics of Oracle9*i* Real Application Clusters are compelling reasons to build large clusters. The economic benefit of combining powerful Intel processor-based servers running Linux in a large cluster makes the idea even more palatable.

So, what is required to make a large database cluster and what are the management considerations? The answer hinges on the technology that is being assembled into the cluster.

The base requirement for an Oracle9*i* Real Application Clusters cluster is a set of servers with shared disk access (for example, just a bunch of disks, or JDOB) and LAN or interconnect connectivity. Strictly speaking, nothing more is required. For reasons explored throughout this chapter, it is unlikely that such a cluster will be as manageable at a large node count, nor would such a cluster be configured to offer the management and flexibility attributes of the flexible database cluster model.

Concerns over building and maintaining large clusters for Oracle9*i* Real Application Clusters generally fall into six categories, although not necessarily in this order:

- Storage configuration and management
- Storage connectivity and configuration
- OS configuration and management

- Oracle product installation, configuration, and maintenance
- Database file locations and space management
- Operational methodology

Storage configuration and management

Today, building large clusters such as the flexible database cluster in this proof of concept is actually quite simple, due in part to Fibre Channel SAN technology. Even simpler is building large clusters with advanced Intel processor-based clusters such as BladeCenter.

Storage management is also much simpler and more powerful with today's technology. An example is IBM TotalStorage intelligent storage array.

Storage connectivity and configuration

Modern technology is also making it easier to connect and configure SAN switches. In the flexible data clusters proof-of-concept system, the switch was easily configured with the IBM TotalStorage SAN Fibre Channel Switch Specialist management tool. The switch features an embedded Web browser interface for configuration, management, and troubleshooting. BladeCenter integrates Fibre Channel switch hardware packaged to almost eliminate cabling.

OS configuration and management

Configuring Linux to support a large cluster is much simpler than configuring legacy UNIX. Enhanced configuration tool kits are available with Linux distributions such as SLES 8, which provides the KDE Konsole utility. This utility feeds key input to all or some nodes of a cluster, which is useful for redundant administrative tasks. This is but one example of the added value that SLES 8 offers.

> **Note:** For information about using KDE Konsole when configuring large clusters for Oracle9*i* Real Application Clusters, see the following paper on the SUSE Web site:
>
> http://www.SUSE.com/en/business/certifications/certified_software/oracle/docs/9iR2_RAC_sles8_polyserve.pdf

SLES 8 also provides a helpful package in RPM form called **orarun**. This package assists administrators with node-level Oracle9*i* Real Application Clusters requirements such as environment variables, kernel parameters, and automated startup and shutdown of key Oracle components such as Oracle Cluster Management Service (OCMS), SQL*Net listeners, and global services daemon (GSD).

Every small improvement that makes a cluster resemble a single system is critically important, even more so in the case of the flexible database cluster.

Oracle product installation, configuration, and maintenance

On standard clusters, the Oracle9*i* Enterprise Edition software must be installed on each node where instances of Oracle9*i* Real Application Clusters will be executed. During installation, the Oracle Universal Installer prompts for a list of nodes on which to install the product. It then copies the files for Oracle Home, more than 100,000 of them, to each node in the list.

Installing Oracle is not difficult. However, difficulty arises when Oracle is installed on, for example, 14 nodes. The Oracle Universal Installer provides for a list of eight nodes for installing Oracle, while the maximum supported node count on Linux is 32.

Installing Oracle on a 14-node cluster requires manual file propagation if a shared Oracle Home is not available. Configuration must occur in 14 different locations, one for each node.

For instance, configuring init.ora requires logging in to 14 different systems. Applying patches is difficult as well. For example, the 9.2.0.3 upgrade patch for Linux is more than 280 MB in size. This is a lot of software to apply to a lot of Oracle Home locations.

With a general-purpose cluster file system, such as the PolyServe Matrix Server cluster file system used in this test, it is quite simple to set up a shared Oracle Home. First, the Oracle Universal Installer is instructed to install on only one node (a single-node cluster install). Then, after Oracle is fully installed on the cluster file system as a single-node install, it is very simple to convert that Oracle Home so that it is shared by any number of other nodes.

Converting to a shared Oracle Home is merely providing for like-name directories and files with node-specific directories and files. For example, it may be advantageous to have the directory $ORACLE_HOME/network/admin set up as a different physical directory when you log in to different nodes, yet this indirection needs to be automatic. The PolyServe Matrix Server cluster file system provides Context Dependent Symbolic Links (CDSLs) to do this. Only a few objects require a CDSL, so this is really quite simple.

With a shared Oracle Home, all nodes in the cluster can use the same executables. Also, configuration files are located in the cluster file system and can be edited from any node in the cluster. In fact, all configuration files for all nodes can be edited from any node in the cluster. $ORACLE_HOME points to the same directory on all nodes, and that directory is in the cluster file system.

Database file locations and space management

Managing a large number of Oracle data files can become difficult when several databases are housed in one large cluster, such as those in the flexible database cluster proof of concept. Perhaps the most significant technology available for assisting in this area is an Oracle9*i* feature known as Oracle Managed Files (OMF). Using OMF with Oracle9*i* Real Application Clusters requires a cluster file system such as the PolyServe Matrix Server cluster file system.

The OMF feature is preferred in a large flexible database cluster because it simplifies administration. Other benefits of OMF are:

- Simplified deployment of databases: Using OMF minimizes the time spent making decisions about file structure and naming and reduces file management tasks overall.
- Reduction of the corruption that can be caused when administrators specify the wrong file: Oracle provides each OMF file with a unique name.
- Reduction of wasted disk space consumed by obsolete files: The OMF feature removes unused files automatically.

When a database administrator (DBA) must think about tablespaces as a collection of files, it takes time away from actual database administration. The file content is the actual database. Databases consist of rows in blocks grouped into extents and tracked as segments. Those objects reside in data files. Deploying with OMF reduces the need for physical administration and so that a DBA can concentrate on logical administration, which is exactly where the biggest return on DBA effort lies.

For example, why be concerned about what sectors of disk are used by a file system when that particular detail is the responsibility of the file system? Likewise, DBAs can leave data file management to OMF. More time can be dedicated to query optimization and other such tasks that may actually improve performance.

Examples of how flexible database clusters use OMF are provided in Chapter 4, “Proven benefits for Oracle9i Real Application Clusters” on page 69.

Operational methodology

Monitoring is a primary concern in a large cluster environment. If administrators must execute numerous commands to obtain the status of many nodes, the value proposition quickly dwindles.

Monitoring cluster operations with the type of technology used in the flexible database cluster proof of concept is vastly superior to the technology of the recent past. Comprehensive I/O monitoring at the storage level is possible through modern storage management software. Oracle9*i* Real Application Clusters and Enterprise Manager offer a great deal of instance and global database monitoring. PolyServe Matrix Server offers monitoring of the entire cluster from a central console that can be executed either remotely or on any system in the datacenter. In addition, the implementation of the Oracle Disk Manager (ODM) offered by PolyServe provides unprecedented Oracle-centric, cluster-aware I/O monitoring.

Beyond monitoring, other issues have traditionally affected cluster operations. For example, administrators have had to connect to specific nodes in the cluster to perform operations such as configuring files in an Oracle Home. Also, requiring administrators to remember which node they used to perform an export or which node has a SQL*Plus report on it can be bothersome in today's hectic IT environment.

Concerns such as these have served as roadblocks to deploying the type of large cluster that can be used as a flexible database cluster.

3.2.5 Infrastructure for a flexible database cluster

This section describes how the flexible database cluster was set up for testing. The core technology for this aspect of the proof of concept was the PolyServe Matrix Server cluster file system. Two PolyServe Matrix Server product features were key in this testing: the cluster file system and the MxODM I/O monitoring package.

PolyServe Matrix Server cluster file system

The PolyServe Matrix Server cluster file system is both general-purpose and optimized for Oracle in the following ways:

- General-purpose

  A single shared Oracle Home was configured for all 14 nodes. Archived redo logging was performed in a cluster file system location and compressed.

  Some of the data was loaded with External Tables (an Oracle9*i* feature supported only in a PolyServe Matrix Server cluster file system with Oracle9*i* Real Application Clusters).

- Optimized for Oracle

  All data files, control files, online logs, and so on were located in file systems mounted with the PolyServe Matrix Server "DBOPTIMIZED" mount option, which implements Direct I/O.

  ODM was used for asynchronous I/O and improved clusterwide I/O monitoring. For more information about ODM, see the white paper on the Oracle Technology Network:

  `http://otn.oracle.com/deploy/availability/pdf/odm_wp.pdf`

Shared Oracle Home

As described earlier in this paper, a general-purpose cluster file system such as the PolyServe Matrix Server cluster file system supports setting up a single directory for Oracle Home. This functionality is key to the flexible database cluster architecture.

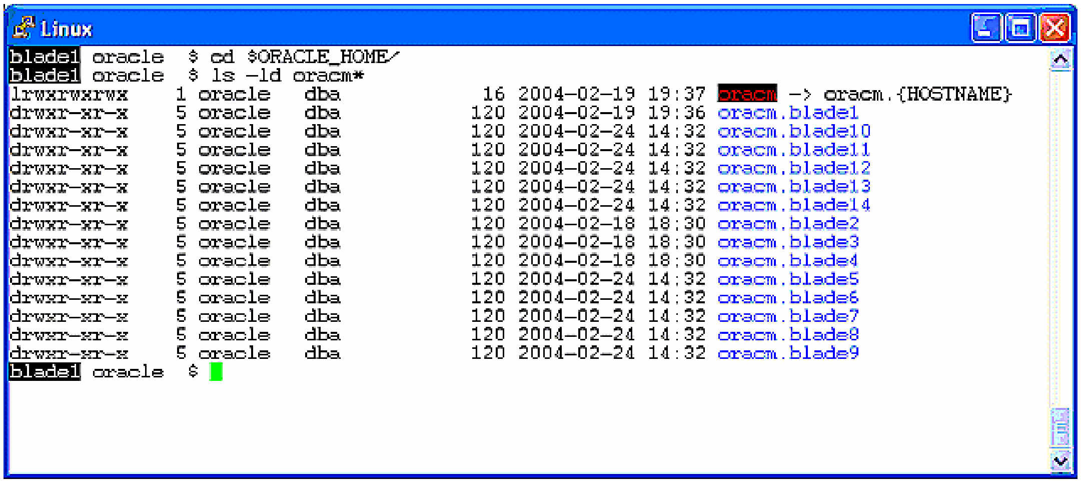
Context Dependent Symbolic Links

Context Dependent Symbolic Links (CDSL) are a feature of the PolyServe Matrix Server cluster file system. They are links to files or directories that are resolved based on the host name of the node on which the current process is executing. CDSL is an essential feature for the enabling of a shared Oracle Home. It also facilitates management aspects such as setting up SQL*Net, archived redo logging, instance logging (for example, alert logs and trace files), and so on.

Oracle Cluster Management Services setup

Using a shared Oracle Home and CDSL simplified the process of setting up Oracle Cluster Management Services (OCMS) for 14 nodes on the flexible database cluster.

In Figure 3-1, the OCMS subdirectory (oracm) is actually a CDSL that links the directory `.oracm.${HOSTNAME}` to oracm. For example, logging in to blade6 will automatically link oracm to `oracm.blade6`. It is easy, however, to modify the contents of any of the oracm CDSL directories because they are simple directories.



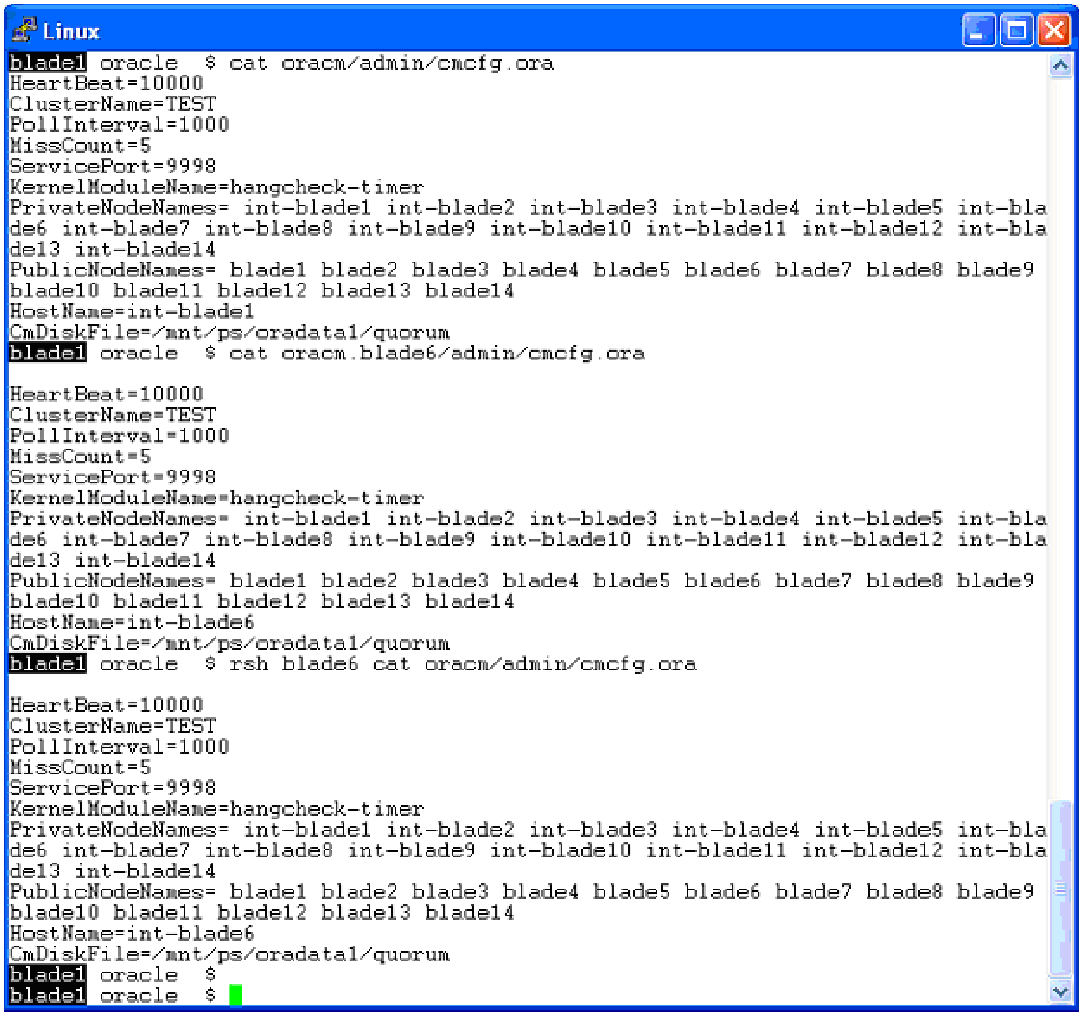
```
blade1 oracle  $ cd $ORACLE_HOME/
blade1 oracle  $ ls -ld oracm*
lrwxrwxrwx    1 oracle   dba              16 2004-02-19 19:37 oracm -> oracm.{HOSTNAME}
drwxr-xr-x    5 oracle   dba             120 2004-02-19 19:36 oracm.blade1
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade10
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade11
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade12
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade13
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade14
drwxr-xr-x    5 oracle   dba             120 2004-02-18 18:30 oracm.blade2
drwxr-xr-x    5 oracle   dba             120 2004-02-18 18:30 oracm.blade3
drwxr-xr-x    5 oracle   dba             120 2004-02-18 18:30 oracm.blade4
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade5
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade6
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade7
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade8
drwxr-xr-x    5 oracle   dba             120 2004-02-24 14:32 oracm.blade9
blade1 oracle  $
```

Figure 3-1 OCMS subdirectory

Figure 3-2 on page 27 shows a shell process on blade1 that lists the contents of the cmcfg.ora file in its CDSL-resolved $ORACLE_HOME/admin/cmcfg.ora file. It also shows that, while executing on blade1, the cmcfg.ora file is visible with the physical directory path of:

```
$ORACLE_HOME/oracm.blade6/admin/cmcfg.ora
```

Finally, using the **`rsh(1)`** command, the cmcfg.ora file for blade6 is listed as a simple path because the CDSL resolves it properly once the shell is generated on blade6.



```
blade1 oracle  $ cat oracm/admin/cmcfg.ora
HeartBeat=10000
ClusterName=TEST
PollInterval=1000
MissCount=5
ServicePort=9998
KernelModuleName=hangcheck-timer
PrivateNodeNames= int-blade1 int-blade2 int-blade3 int-blade4 int-blade5 int-bla
de6 int-blade7 int-blade8 int-blade9 int-blade10 int-blade11 int-blade12 int-bla
de13 int-blade14
PublicNodeNames= blade1 blade2 blade3 blade4 blade5 blade6 blade7 blade8 blade9
blade10 blade11 blade12 blade13 blade14
HostName=int-blade1
CmDiskFile=/mnt/ps/oradata1/quorum
blade1 oracle  $ cat oracm.blade6/admin/cmcfg.ora

HeartBeat=10000
ClusterName=TEST
PollInterval=1000
MissCount=5
ServicePort=9998
KernelModuleName=hangcheck-timer
PrivateNodeNames= int-blade1 int-blade2 int-blade3 int-blade4 int-blade5 int-bla
de6 int-blade7 int-blade8 int-blade9 int-blade10 int-blade11 int-blade12 int-bla
de13 int-blade14
PublicNodeNames= blade1 blade2 blade3 blade4 blade5 blade6 blade7 blade8 blade9
blade10 blade11 blade12 blade13 blade14
HostName=int-blade6
CmDiskFile=/mnt/ps/oradata1/quorum
blade1 oracle  $ rsh blade6 cat oracm/admin/cmcfg.ora

HeartBeat=10000
ClusterName=TEST
PollInterval=1000
MissCount=5
ServicePort=9998
KernelModuleName=hangcheck-timer
PrivateNodeNames= int-blade1 int-blade2 int-blade3 int-blade4 int-blade5 int-bla
de6 int-blade7 int-blade8 int-blade9 int-blade10 int-blade11 int-blade12 int-bla
de13 int-blade14
PublicNodeNames= blade1 blade2 blade3 blade4 blade5 blade6 blade7 blade8 blade9
blade10 blade11 blade12 blade13 blade14
HostName=int-blade6
CmDiskFile=/mnt/ps/oradata1/quorum
blade1 oracle  $
blade1 oracle  $
```

Figure 3-2 Shell process listing the contents of cmcfg.ora

Using a shared Oracle Home and CDSL reduces administrative overhead. The administrator can view and edit all of the configuration files for all nodes on any node in the cluster while logged in. Without a shared Oracle Home, an administrator would have to log in to each of the 14 nodes and edit the cmcfg.ora file to set up OCMS.

It is important to note that if a given node name does not appear in the list assigned to the PrivateNodeNames and PublicNodeNames parameters in the cmcfg.ora file, it cannot join the Oracle cluster. That is, it cannot join until all instances of OCMS have been shut down, a task that requires all database instances throughout the cluster to be shut down.

For example, if there is another node called blade15 in the cluster, it cannot dynamically join the Oracle9*i* Real Application Clusters cluster (OCMS) until all the following tasks have been completed:

- Database instances are shut down.
- All copies of the cmcfg.ora file are edited to add blade15 in the PrivateNodeNames/PublicNodeNames parameter assignments.
- OCMS has been rebooted on each node.

This impacts sites that rely on a cold-standby server in the event of a node failure. In a four-node cluster scenario, losing node 4 and replacing it with the cold-standby node means that cmcfg.ora needs to be configured in advance to accommodate its node name in the active cluster.

The flexible database cluster architecture is much more flexible. All nodes in the cluster are included in the OCMS member list. They all run Oracle9*i* Real Application Clusters; the only thing that varies is which instances run on the various nodes.

PolyServe Matrix Server Oracle Disk Manager

PolyServe Matrix Server provides an ODM implementation called MxODM that supports the ODM interface. Although MxODM offers improved data file integrity with clusterwide file keys for access, its main benefit in the flexible database cluster architecture is monitoring. MxODM also enables Oracle9*i* with asynchronous I/O on the direct I/O-mounted file systems, where it stores data files and other database files such as redo logs.

The MxODM I/O statistics package provides basic I/O performance information. These reported items are referred to as the Core Reporting Elements:

- Number of file read and write operations
- Read and write throughput per second in kilobytes
- Count of synchronous and asynchronous I/O operations
- I/O service times
- Percentages

The Core Reporting Elements can be provided at the following levels:

- Clusterwide level: At this level, aggregate information is provided for all database instances on all nodes.
- Database global level: At this level, information is limited to a named database (for example, PROD, DEV, FIN, or DSS).
- Instance level: At this level, information is limited to a named instance (for example, PROD1, PROD8, DEV1, FIN4, or DSS_6).
- Node level: At this level, information is limited to a named node (for example, rhas1.acme.com or rhas6.acme.com). This information is the aggregate of all instance activity on the named node.

  If a node hosts instances that access different databases (for example, $ORACLE_SID=PROD1, or $ORACLE_SID=DEV1), the Core Reporting Elements will reflect the combined information for all instances on the named node.

Because MxODM understands Oracle file, process, and I/O types, the **`mxodmstat(8)`** command offers very specialized reporting capabilities. On complex clustered systems, it is nearly impossible to take a quick look at the clusterwide or per-instance activity for a given subsystem of the Oracle Server.

For example, on an 8-node cluster with six PROD instances, two DEV instances, and Parallel Query Slaves active only on nodes 1 through 4 in the PROD database, a DBA will have difficulty associating clusterwide impact to the Parallel Query Option (PQO) activity. Likewise, quickly determining the DBWR activity for only the PROD instances on nodes 1 through 6 is nearly impossible without MxODM.

MxODM offers canned reporting that focuses on the following key Oracle subsystems:

- PQO

  This query returns the Core Reporting Elements for only the PQO slaves (for example, ora_p000_PROD1, ora_p001_PROD3, and so on). DBAs can use this information to obtain a top-level view of the impact PQO is having on the cluster, either as a whole or at the node level.

- Log writer

  This query focuses on only the lgwr processes and their activity at the cluster level, database level, or node level. Because all Core Reporting Elements can be returned in this query, DBAs can maintain streaming output of this query showing lgwr activity at either the cluster level or broken down by database, instance, or node.

- Database writer

  This query is of the utmost value. It too can return all Core Reporting Elements at all Reporting Levels; however, it can also limit reporting to only dbwr process activity. DBAs can glance at **mxodmstat(8)** output and easily determine the average dbwr I/O service times for all databases clusterwide, or can focus on specific databases, nodes, or instances.

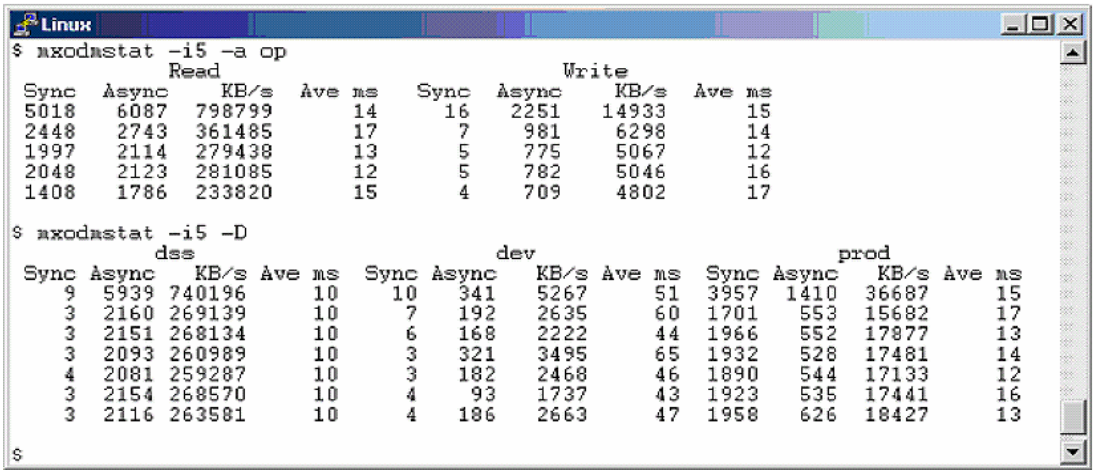
The examples discussed in this section demonstrate are only a few samples of the helpful monitoring used during the flexible database cluster project.

> **Note:** More monitoring functionality than is described in this chapter is available in the mxodmstat package and at this Web site:
>
> http://www.polyserve.com

Monitoring several databases that have instances spread across a 14-node BladeCenter is a daunting task. A high-level view can be obtained with **mxodmstat**, as was done on the flexible database cluster test system.

The first command in Figure 3-3 shows top-level I/O activity for all databases in aggregate categorized into reads and writes.

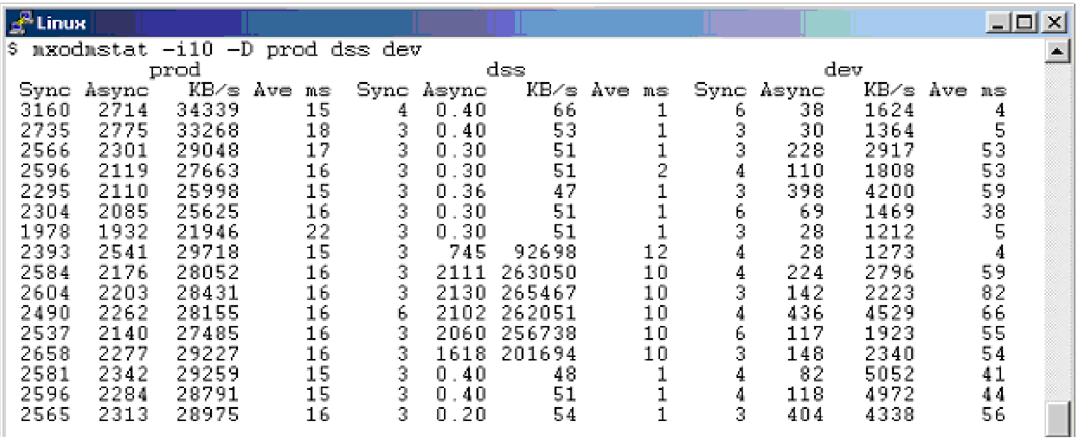
The second command shows top-level I/O for all three databases by database. This is the aggregated total I/O with breakout for the count of synchronous and asynchronous I/O and I/O service times. Note that without the -a op argument/option pair, the I/O is not broken out by reads and writes.



```
$ mxodmstat -i5 -a op
           Read                          Write
 Sync  Async    KB/s  Ave ms   Sync  Async    KB/s  Ave ms
 5018   6087  798799      14     16   2251   14933      15
 2448   2743  361485      17      7    981    6298      14
 1997   2114  279438      13      5    775    5067      12
 2048   2123  281085      12      5    782    5046      16
 1408   1786  233820      15      4    709    4802      17

$ mxodmstat -i5 -D
          dss                        dev                        prod
 Sync Async   KB/s Ave ms  Sync Async    KB/s Ave ms  Sync Async    KB/s Ave ms
    9  5939 740196     10    10   341    5267     51  3957  1410   36687     15
    3  2160 269139     10     7   192    2635     60  1701   553   15682     17
    3  2151 268134     10     6   168    2222     44  1966   552   17877     13
    3  2093 260989     10     3   321    3495     65  1932   528   17481     14
    4  2081 259287     10     3   182    2468     46  1890   544   17133     12
    3  2154 268570     10     4    93    1737     43  1923   535   17441     16
    3  2116 263581     10     4   186    2663     47  1958   626   18427     13

$
```

Figure 3-3 Monitoring databases with mxodmstat

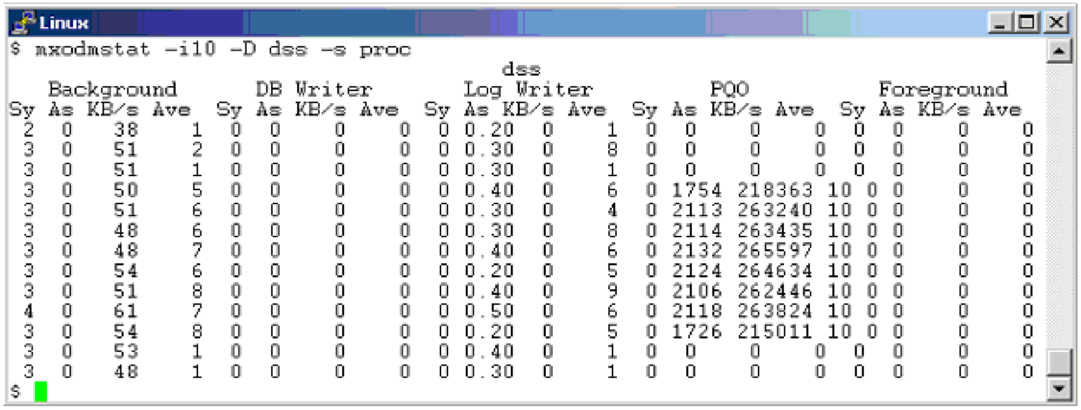
The mxodmstat output in Figure 3-4 shows a flurry of DSS database activity.



```
$ mxodmstat -i10 -D prod dss dev
           prod                      dss                       dev
 Sync Async   KB/s Ave ms  Sync Async   KB/s Ave ms  Sync Async   KB/s Ave ms
 3160  2714  34339     15     4  0.40     66      1     6    38   1624      4
 2735  2775  33268     18     3  0.40     53      1     3    30   1364      5
 2566  2301  29048     17     3  0.30     51      1     3   228   2917     53
 2596  2119  27663     16     3  0.30     51      2     4   110   1808     53
 2295  2110  25998     15     3  0.36     47      1     3   398   4200     59
 2304  2085  25625     16     3  0.30     51      1     6    69   1469     38
 1978  1932  21946     22     3  0.30     51      1     3    28   1212      5
 2393  2541  29718     15     3   745  92698     12     4    28   1273      4
 2584  2176  28052     16     3  2111 263050     10     4   224   2796     59
 2604  2203  28431     16     3  2130 265467     10     3   142   2223     82
 2490  2262  28155     16     6  2102 262051     10     4   436   4529     66
 2537  2140  27485     16     3  2060 256738     10     6   117   1923     55
 2658  2277  29227     16     3  1618 201694     10     3   148   2340     54
 2581  2342  29259     15     3  0.40     48      1     4    82   5052     41
 2596  2284  28791     15     3  0.40     51      1     4   118   4972     44
 2565  2313  28975     16     3  0.20     54      1     3   404   4338     56
```

Figure 3-4 mxodmstat shows database activity

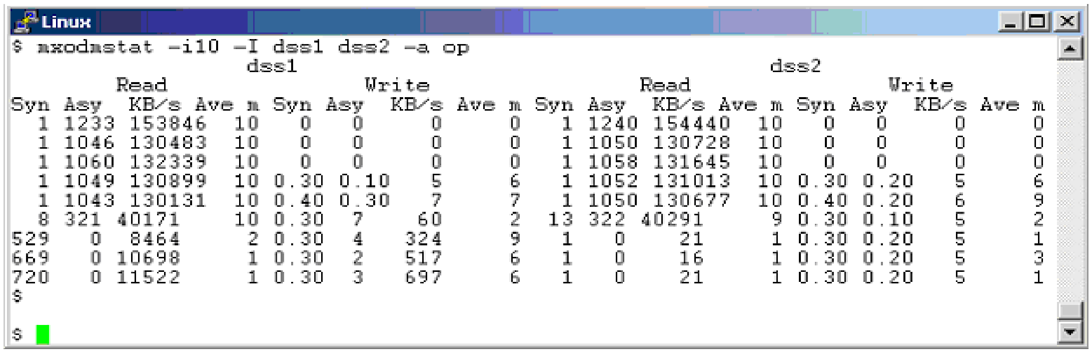
Figure 3-5 shows the use of a drill-down command to identify which processes are performing the I/O on the DSS database. Note that the I/O is mainly performed by PQO processes.



```
$ mxodmstat -i10 -D dss -s proc
                                      dss
   Background      DB Writer       Log Writer         PQO          Foreground
Sy As KB/s Ave  Sy As KB/s Ave  Sy As KB/s Ave  Sy As KB/s Ave  Sy As KB/s Ave
 2  0   38    1  0  0    0    0  0 0.20  0    1  0  0    0    0  0  0    0    0
 3  0   51    2  0  0    0    0  0 0.30  0    8  0  0    0    0  0  0    0    0
 3  0   51    1  0  0    0    0  0 0.30  0    1  0  0    0    0  0  0    0    0
 3  0   50    5  0  0    0    0  0 0.40  0    6  0 1754 218363 10 0 0    0    0
 3  0   51    6  0  0    0    0  0 0.30  0    4  0 2113 263240 10 0 0    0    0
 3  0   48    6  0  0    0    0  0 0.30  0    8  0 2114 263435 10 0 0    0    0
 3  0   48    7  0  0    0    0  0 0.40  0    6  0 2132 265597 10 0 0    0    0
 3  0   54    6  0  0    0    0  0 0.20  0    5  0 2124 264634 10 0 0    0    0
 3  0   51    8  0  0    0    0  0 0.40  0    9  0 2106 262446 10 0 0    0    0
 4  0   61    7  0  0    0    0  0 0.50  0    6  0 2118 263824 10 0 0    0    0
 3  0   54    8  0  0    0    0  0 0.20  0    5  0 1726 215011 10 0 0    0    0
 3  0   53    1  0  0    0    0  0 0.40  0    1  0  0    0    0  0  0    0    0
 3  0   48    1  0  0    0    0  0 0.30  0    1  0  0    0    0  0  0    0    0
$ 
```

Figure 3-5 mxodmstat drill-down command

Figure 3-6 shows a focused monitoring of instances DSS1 and DSS2 only, broken out by reads and writes. After six lines of output (60 seconds), the query plan changed from100% asynchronous large reads by both instances evenly to only DSS1 performing small synchronous reads and only a few asynchronous writes.



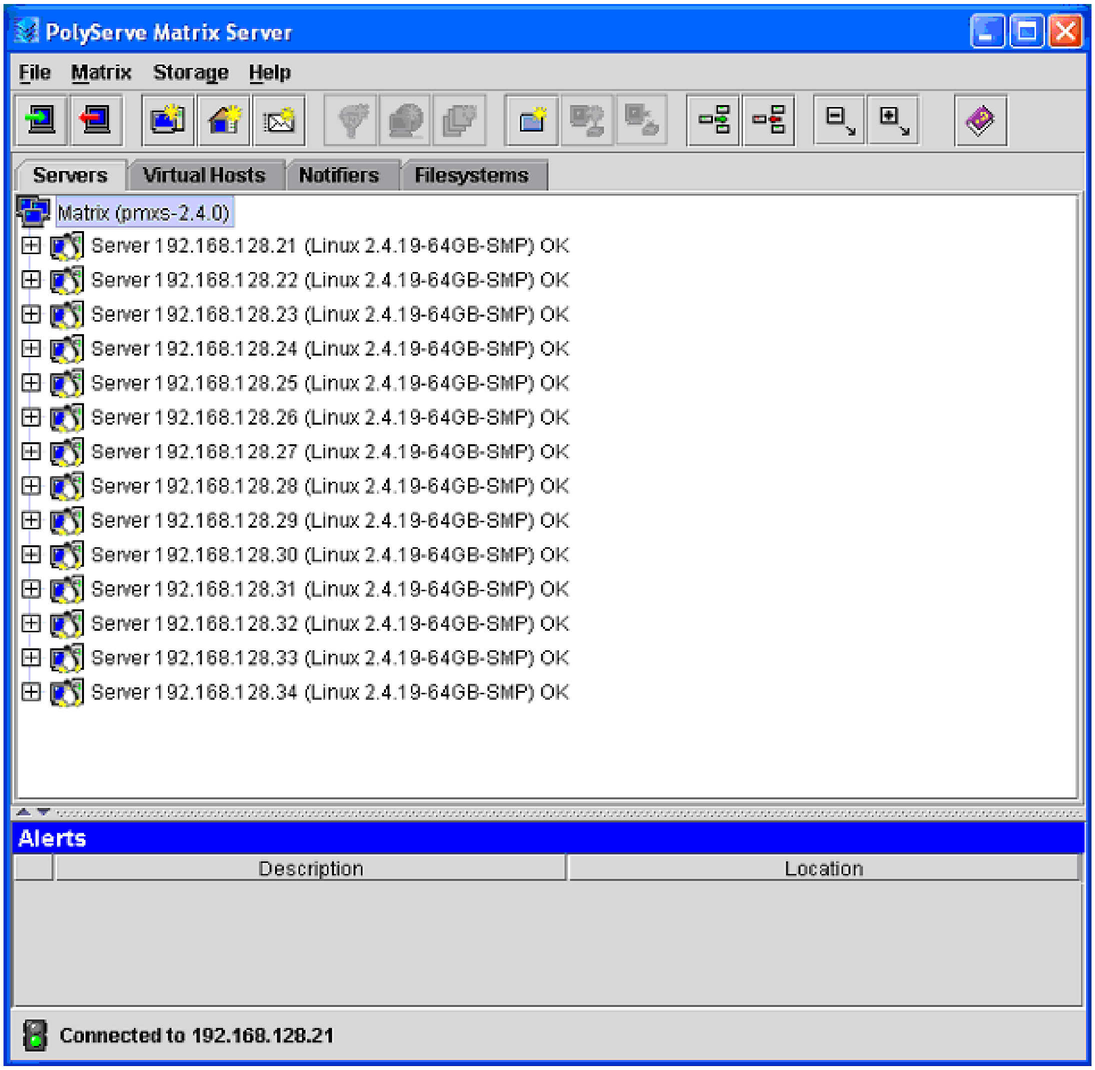
```
$ mxodmstat -i10 -I dss1 dss2 -a op
                  dss1                                    dss2
        Read               Write                Read               Write
Syn Asy  KB/s Ave m Syn Asy  KB/s Ave m Syn Asy  KB/s Ave m Syn Asy  KB/s Ave m
  1 1233 153846  10   0   0     0     0   1 1240 154440  10   0   0     0     0
  1 1046 130483  10   0   0     0     0   1 1050 130728  10   0   0     0     0
  1 1060 132339  10   0   0     0     0   1 1058 131645  10   0   0     0     0
  1 1049 130899  10 0.30 0.10   5     6   1 1052 131013  10 0.30 0.20   5     6
  1 1043 130131  10 0.40 0.30   7     7   1 1050 130677  10 0.40 0.20   6     9
  8 321 40171    10 0.30  7    60     2  13 322 40291     9 0.30 0.10   5     2
529   0  8464     2 0.30  4   324     9   1   0     21    1 0.30 0.20   5     1
669   0 10698     1 0.30  2   517     6   1   0     16    1 0.30 0.20   5     3
720   0 11522     1 0.30  3   697     6   1   0     21    1 0.30 0.20   5     1
$

$ 
```

Figure 3-6 Instances DSS1 and DSS2

Cluster management

PolyServe Matrix Server offers more than a cluster file system; it also offers a high-availability framework and SAN management. Furthermore, PolyServe Matrix Server offers multi-path I/O, a pivotal component in reducing and eliminating single points of failure.

Figure 3-7 shows the PolyServe Matrix Server console, which provides a high-level view of cluster status. The console can be used to set up file systems, monitor systems for failover, mount options, and advanced high-availability processes. A command-line interface is also available for cluster configuration.

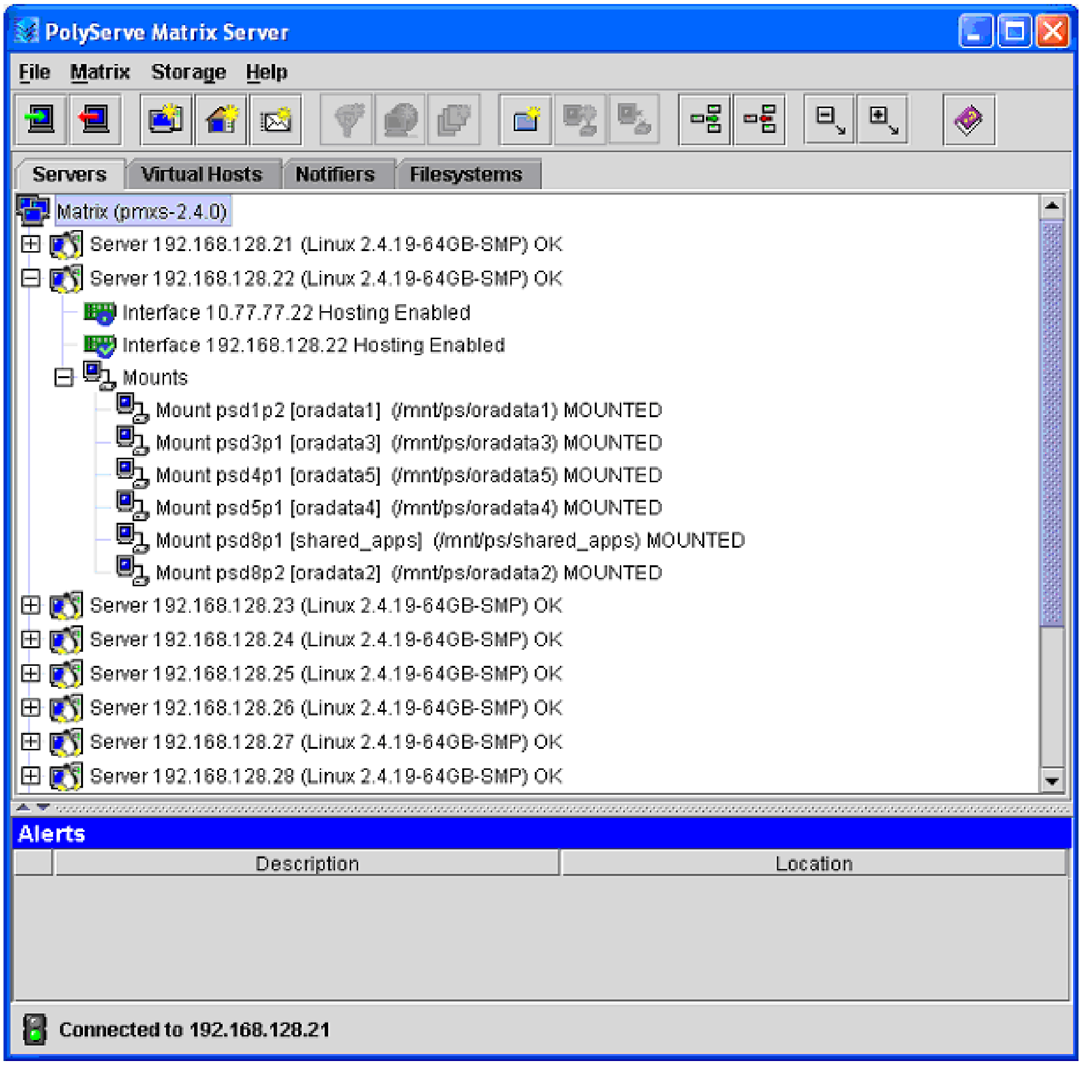


```
Matrix (pmxs-2.4.0)
Server 192.168.128.21 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.22 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.23 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.24 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.25 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.26 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.27 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.28 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.29 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.30 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.31 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.32 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.33 (Linux 2.4.19-64GB-SMP) OK
Server 192.168.128.34 (Linux 2.4.19-64GB-SMP) OK
```

Connected to 192.168.128.21

Figure 3-7 PolyServe Matrix Server Management Console

Figure 3-8 on page 32 illustrates another view of the PolyServe Matrix Server Management Console. This view demonstrates its drill-down capability for obtaining the status of file systems with a clusterwide view.



```
Matrix (pmxs-2.4.0)
⊞ Server 192.168.128.21 (Linux 2.4.19-64GB-SMP) OK
⊟ Server 192.168.128.22 (Linux 2.4.19-64GB-SMP) OK
    Interface 10.77.77.22 Hosting Enabled
    Interface 192.168.128.22 Hosting Enabled
  ⊟ Mounts
      Mount psd1p2 [oradata1]  (/mnt/ps/oradata1) MOUNTED
      Mount psd3p1 [oradata3]  (/mnt/ps/oradata3) MOUNTED
      Mount psd4p1 [oradata5]  (/mnt/ps/oradata5) MOUNTED
      Mount psd5p1 [oradata4]  (/mnt/ps/oradata4) MOUNTED
      Mount psd8p1 [shared_apps]  (/mnt/ps/shared_apps) MOUNTED
      Mount psd8p2 [oradata2]  (/mnt/ps/oradata2) MOUNTED
⊞ Server 192.168.128.23 (Linux 2.4.19-64GB-SMP) OK
⊞ Server 192.168.128.24 (Linux 2.4.19-64GB-SMP) OK
⊞ Server 192.168.128.25 (Linux 2.4.19-64GB-SMP) OK
⊞ Server 192.168.128.26 (Linux 2.4.19-64GB-SMP) OK
⊞ Server 192.168.128.27 (Linux 2.4.19-64GB-SMP) OK
⊞ Server 192.168.128.28 (Linux 2.4.19-64GB-SMP) OK
```

Connected to 192.168.128.21

Figure 3-8 Drill-down on the PolyServe Matrix Server Management Console

3.3 Proof of concept components

State-of-the-art and robust technologies were key to creating a suitable test system to prove the flexible database cluster architecture.

Figure 3-9 on page 33 shows the cluster system components used for the flexible database cluster proof of concept.

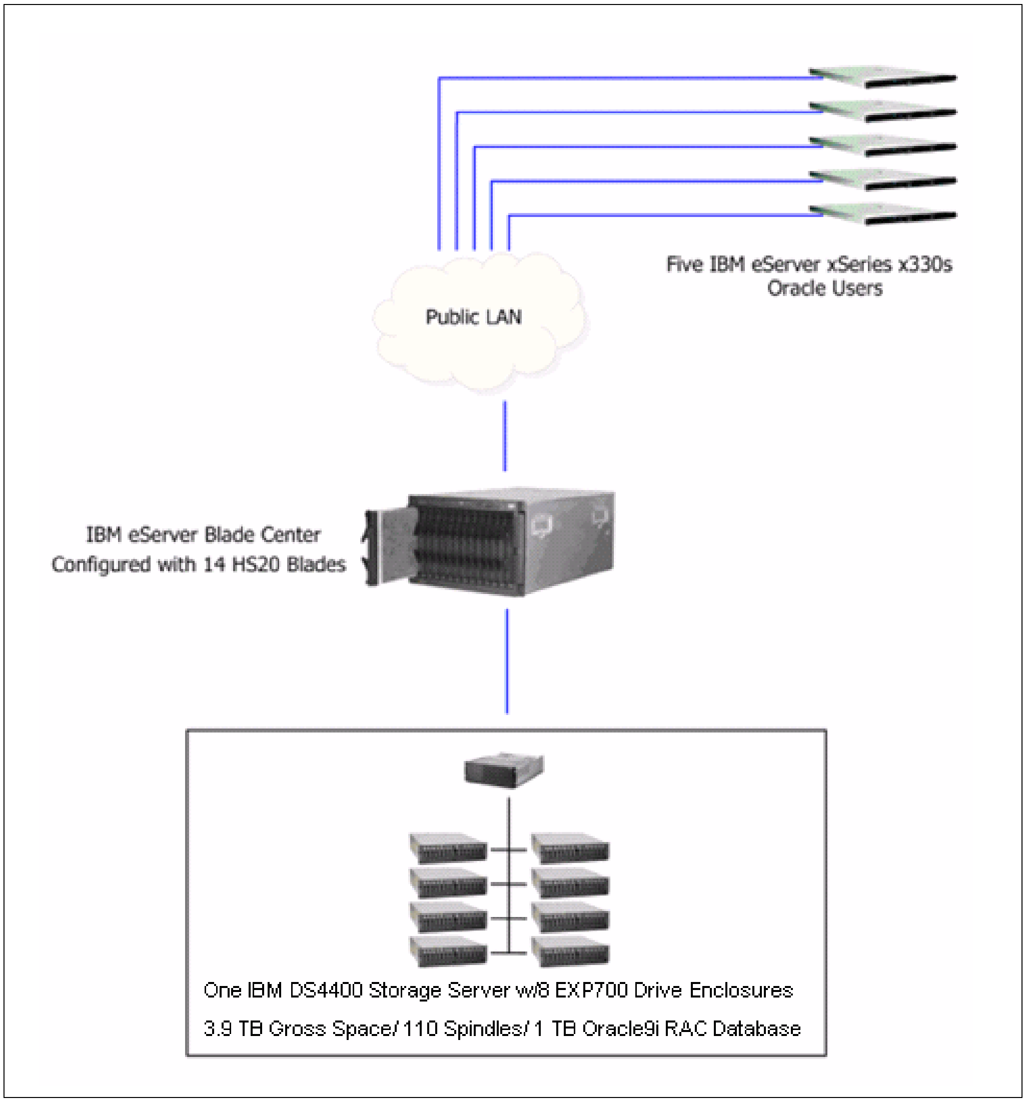

Five IBM eServer xSeries x330s
Oracle Users
Public LAN
IBM eServer Blade Center
Configured with 14 HS20 Blades
One IBM DS4400 Storage Server w/8 EXP700 Drive Enclosures
3.9 TB Gross Space/ 110 Spindles/ 1 TB Oracle9i RAC Database

Figure 3-9 The proof-of-concept flexible database cluster

3.3.1 Overview of BladeCenter

To support the basic computing infrastructure needed by the flexible database cluster proof of concept, a hardware platform that would showcase flexibility and manageability was necessary. BladeCenter provides both.

With emerging blade technologies that enable clients to reverse their server sprawl and collapse the complexity of their distributed IT infrastructure, we felt that this proof point would be an ideal opportunity to showcase this technology in a database cluster environment. Blades also deliver better management software, have less cable snarl, offer more expansion possibilities, and have smaller footprint requirements.

With those features in mind, it is easy to see why BladeCenter, which is rack-optimized and high-density in an innovative 7U form-factor design, was a perfect choice for deploying the Oracle9*i* Real Application Clusters installation. The BladeCenter chassis accommodates up to 14 hot-swap 2-way Intel Xeon™ DP processor-based blade servers. It also integrates with the chassis key infrastructure components such as Layer 2-7 Gigabit Ethernet Switching, SAN switching and centralized management tools.

Large amounts of highly available disk storage had to be connected to BladeCenter, so the storage subsystem was designed around the IBM TotalStorage DS4400 Storage Server. The DS4400 is a redundant array of independent disks (RAID) storage subsystem that contains Fibre Channel (FC) interfaces that connect both the host systems and the disk drive enclosures. With its dual 2 Gbps controllers and high- availability design, IBM TotalStorage DS4400 delivers the necessary throughput to support this high-end proof point.

BladeCenter chassis

The BladeCenter chassis (Figure 3-10 and Figure 3-11) can accommodate up to 14 blade servers in its 7U form factor. Resources are shared among all of the blades and include power, switch, management, and blower modules. The chassis provides high-speed I/O capabilities for all of the modules, thereby reducing the amount of cabling required in the datacenter. The management module, through remote access, can be used to control components in the enclosure.

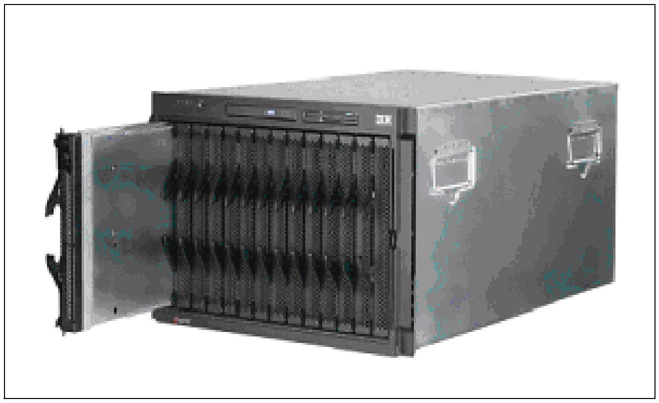


Figure 3-10 BladeCenter - front

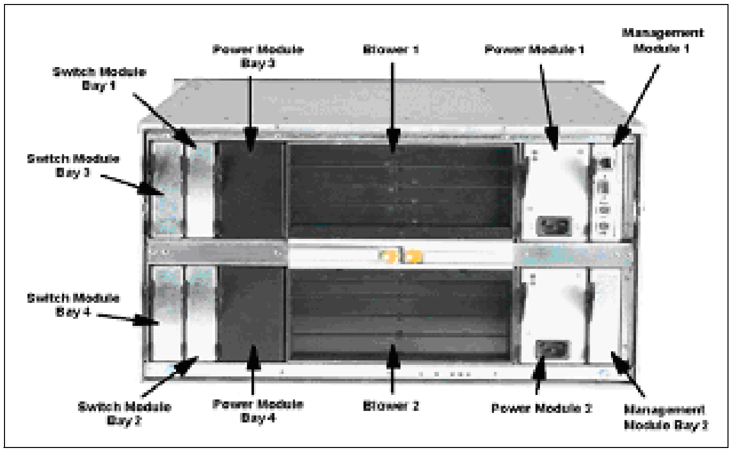


Figure 3-11 BladeCenter - rear

The BladeCenter chassis was configured as follows:

- Standard 48X CD-ROM and 1.44MB floppy accessible from all blades in the Media Tray.
- Management module: The center for systems management on BladeCenter, the management module is responsible for monitoring all components in BladeCenter and each individual blade. It can detect the condition and state of any of the installed components.

- Two additional 1200-watt hot-swap power modules (two are standard). These modules were required to power blade slots 7 through 14. Installed as pairs, the power modules provide redundancy and power for robust configurations.
- Two 4-port Gigabit Ethernet Switch Modules: Although not standard on the BladeCenter unit, the modules were necessary to provide the interconnectivity between the blades, the management module, and the external network. The module is a fully functional Ethernet switch with four external gigabit ports, two internal 10/100 links to the management module, and 14 internal gigabit links to the blades. Two 4-port Gigabit Ethernet Switch Modules were used in this proof point to support access to the external, public network (eth0) and the internal, private interconnect traffic (eth1).
- One 2-port Fibre Channel Switch Module: With the Fibre Channel Expansion Card in each blade server, the optional 2-port Fibre Channel Switch Module completes the required Fibre Channel connectivity to the SAN. Each port is capable of supporting transmission speeds of up to 2 Gbps after auto-negotiating with the DS4400 Storage Server.
- 14 BladeCenter HS20 blades.

BladeCenter HS20 Blade Server

The BladeCenter HS20 blades (Figure 3-12) are high-throughput, 2-way SMP-capable Xeon processor-based blade servers. With support for up to 8 GB PC2100 ECC DDR Chipkill™ SDRAM and processor speeds of 2.4 GHz to 3.2 GHz, these blade servers are highly scalable. An integrated service processor on each blade server communicates with the BladeCenter management module for remote control of server tasks.

Also integrated on the HS20 are two Ethernet controllers that can be configured for either fault-tolerance or increased throughput with adapter teaming. With blade server expansion card options such as Myrinet Cluster, Gigabit Ethernet, Fibre Channel, and support for both EIDE and SCSI drives, a blade can be tuned to create customized solutions that match application needs.

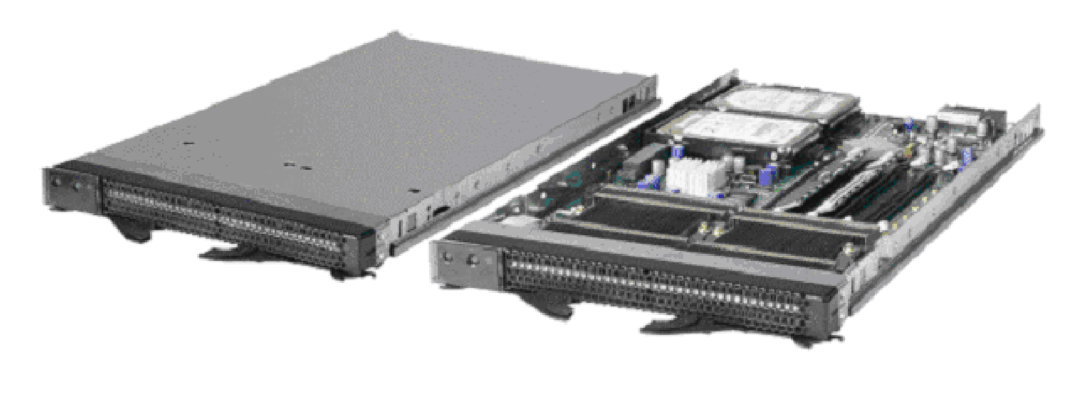


Figure 3-12 View of an HS20 Blade Server

For the flexible database cluster proof of concept, each blade server was configured with:

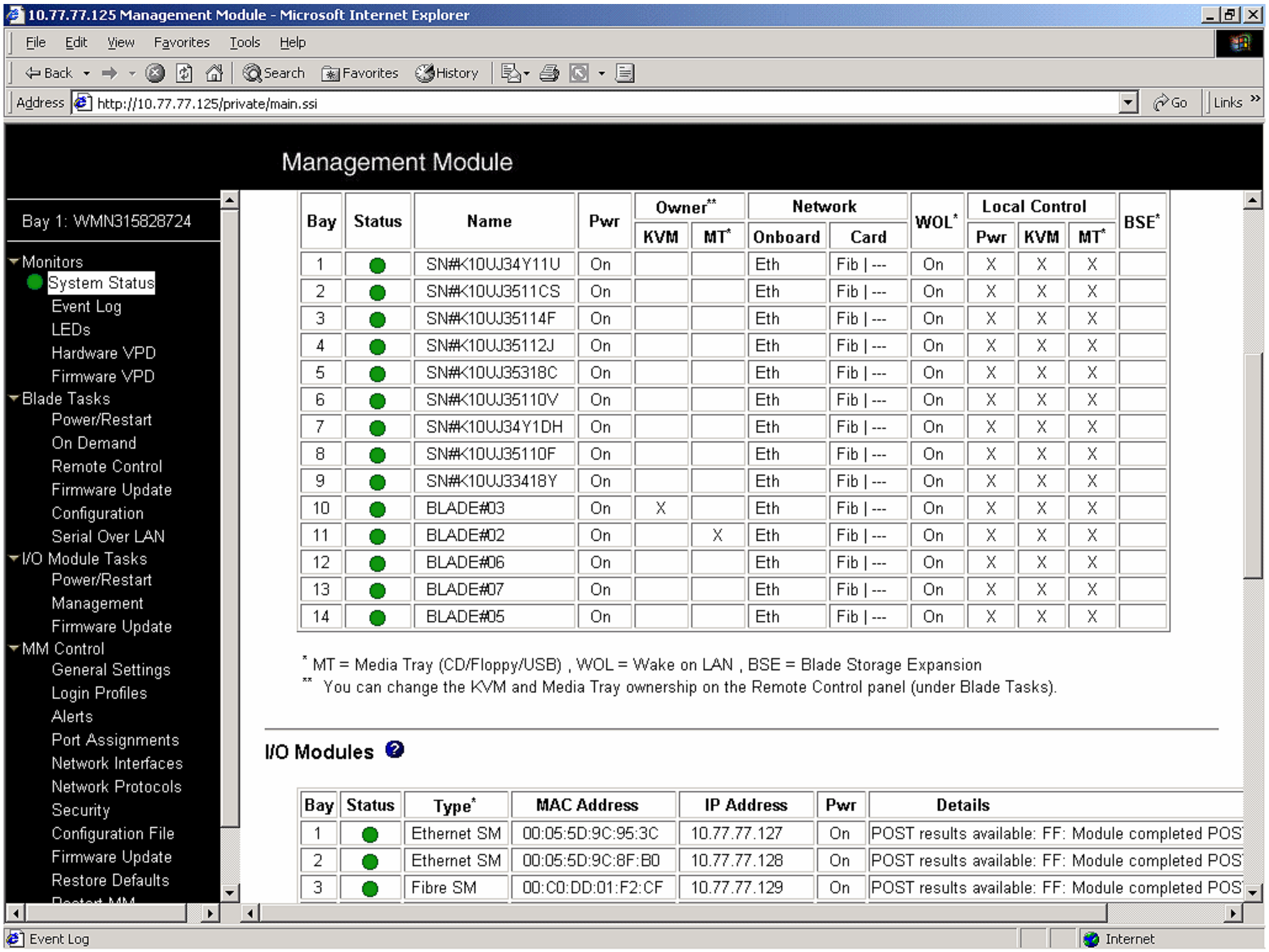
- 2 x 2.4GHz Xeon processors
- 2GB PC2100 DDR ECC Chipkill memory
- 40GB IDE drive
- Integrated dual Broadcom Gigabit Ethernet controllers
- Fibre Channel Expansion card
- SLES 8 (Service Pack 1)

3.3.2 Managing BladeCenter

As IT environments become more complex to manage and support, it is important that users have the tools and processes for monitoring and maintaining the infrastructure. To reduce the occasionally time-consuming task of managing high-density computing environments, BladeCenter has a built-in, Web-based graphical user interface (GUI) that allows remote access to BladeCenter. With this GUI, users can remotely turn blades off and on and manage I/O modules. Access is gained through a standard Ethernet port and a standard Web browser.

The BladeCenter main menu has four main sections: System Monitors, Blade Tasks, I/O Module, Management Module Control. With the Monitors menu, users can view status, settings, and other information about each of the key components configured in BladeCenter. The information displayed includes:

- System Status of blade servers, I/O modules, management modules, and power modules
- Event log
- Front panel and blade server LEDs
- Vital Product Data for blades, I/O module, and management module



Bay	Status	Name	Pwr	Owner**		Network		WOL*	Local Control			BSE*
				KVM	MT*	Onboard	Card		Pwr	KVM	MT*	
1	●	SN#K10UJ34Y11U	On			Eth	Fib | ---	On	X	X	X	
2	●	SN#K10UJ3511CS	On			Eth	Fib | ---	On	X	X	X	
3	●	SN#K10UJ35114F	On			Eth	Fib | ---	On	X	X	X	
4	●	SN#K10UJ35112J	On			Eth	Fib | ---	On	X	X	X	
5	●	SN#K10UJ35318C	On			Eth	Fib | ---	On	X	X	X	
6	●	SN#K10UJ35110V	On			Eth	Fib | ---	On	X	X	X	
7	●	SN#K10UJ34Y1DH	On			Eth	Fib | ---	On	X	X	X	
8	●	SN#K10UJ35110F	On			Eth	Fib | ---	On	X	X	X	
9	●	SN#K10UJ33418Y	On			Eth	Fib | ---	On	X	X	X	
10	●	BLADE#03	On	X		Eth	Fib | ---	On	X	X	X	
11	●	BLADE#02	On		X	Eth	Fib | ---	On	X	X	X	
12	●	BLADE#06	On			Eth	Fib | ---	On	X	X	X	
13	●	BLADE#07	On			Eth	Fib | ---	On	X	X	X	
14	●	BLADE#05	On			Eth	Fib | ---	On	X	X	X	

* MT = Media Tray (CD/Floppy/USB) , WOL = Wake on LAN , BSE = Blade Storage Expansion
** You can change the KVM and Media Tray ownership on the Remote Control panel (under Blade Tasks).

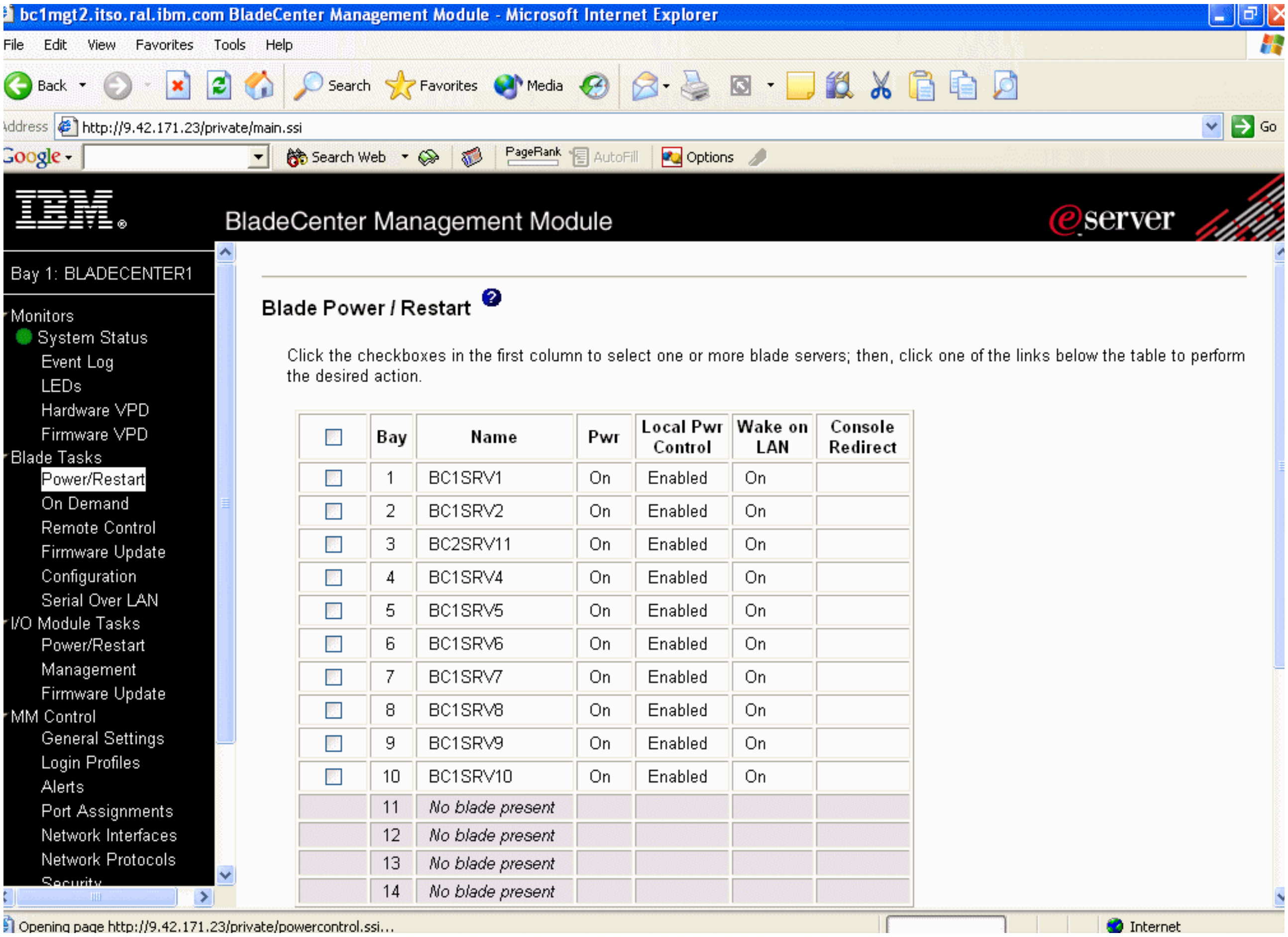
I/O Modules

Bay	Status	Type*	MAC Address	IP Address	Pwr	Details
1	●	Ethernet SM	00:05:5D:9C:95:3C	10.77.77.127	On	POST results available: FF: Module completed POS
2	●	Ethernet SM	00:05:5D:9C:8F:B0	10.77.77.128	On	POST results available: FF: Module completed POS
3	●	Fibre SM	00:C0:DD:01:F2:CF	10.77.77.129	On	POST results available: FF: Module completed POS

Figure 3-13 Portion of the System Status summary window

The settings for each blade server can be configured and controlled with the Blade Tasks section of the menu, which is illustrated in Figure 3-14. The following tasks can be performed:

- Start and restart of individual or all blade servers
- Remote control of an individual blade to associate the media tray ownership
- Updates of firmware
- Configuration for each blade for KVM control, media tray control, wake on LAN, and boot sequence



Blade Power / Restart

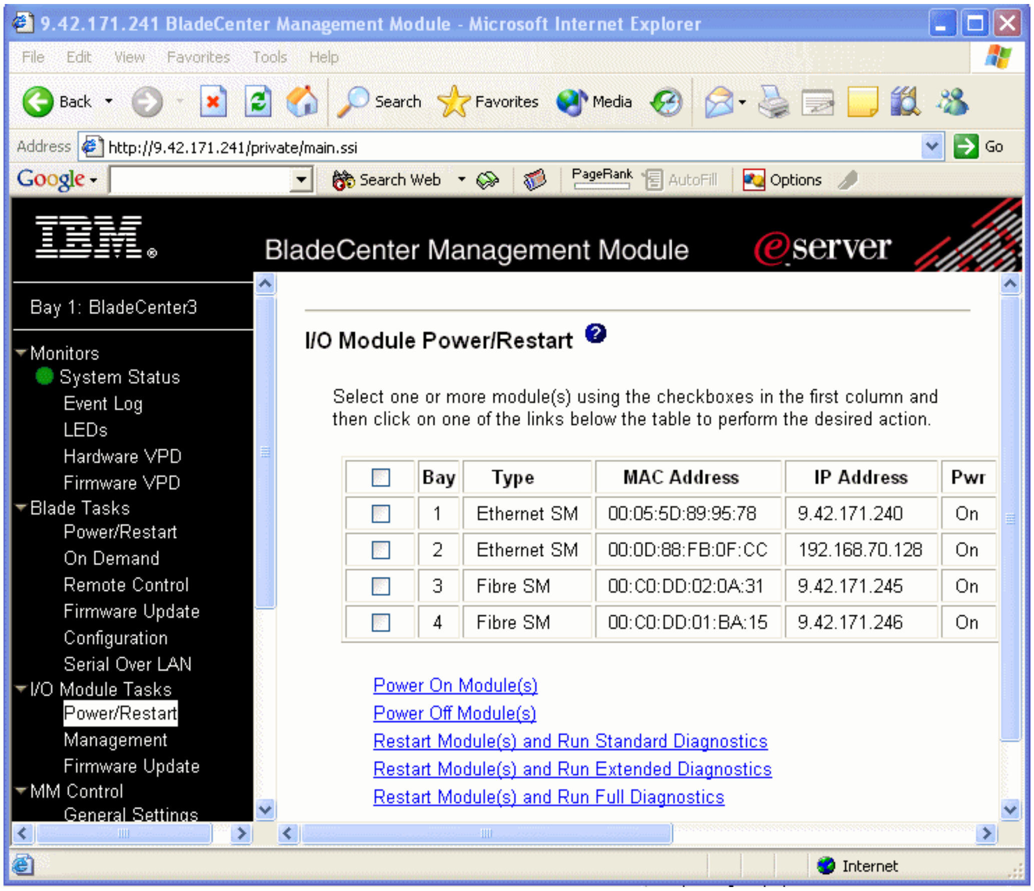
Click the checkboxes in the first column to select one or more blade servers; then, click one of the links below the table to perform the desired action.

☐	Bay	Name	Pwr	Local Pwr Control	Wake on LAN	Console Redirect
☐	1	BC1SRV1	On	Enabled	On	
☐	2	BC1SRV2	On	Enabled	On	
☐	3	BC2SRV11	On	Enabled	On	
☐	4	BC1SRV4	On	Enabled	On	
☐	5	BC1SRV5	On	Enabled	On	
☐	6	BC1SRV6	On	Enabled	On	
☐	7	BC1SRV7	On	Enabled	On	
☐	8	BC1SRV8	On	Enabled	On	
☐	9	BC1SRV9	On	Enabled	On	
☐	10	BC1SRV10	On	Enabled	On	
	11	*No blade present*				
	12	*No blade present*				
	13	*No blade present*				
	14	*No blade present*				

Figure 3-14 Portion of Blade Task menu selection

The heading of the I/O Module section (illustrated in Figure 3-15 on page 38) of the menu depends on which optional switch modules are installed on BladeCenter. In this particular installation, there were two 4-port Gigabit Ethernet Switch Modules and two 2-port Fibre Channel Switch Modules. The heading was displayed as I/O Module Tasks and allowed the following activities:

- Start and restart of modules
- Individual switch management setting of IP network addresses
- A drill-down for advanced management into each module to:
  - Further configure the switch modules
  - Generate a Telnet session or Web-based GUI to monitor and control the device
- Firmware upgrade capabilities



I/O Module Power/Restart

Select one or more module(s) using the checkboxes in the first column and then click on one of the links below the table to perform the desired action.

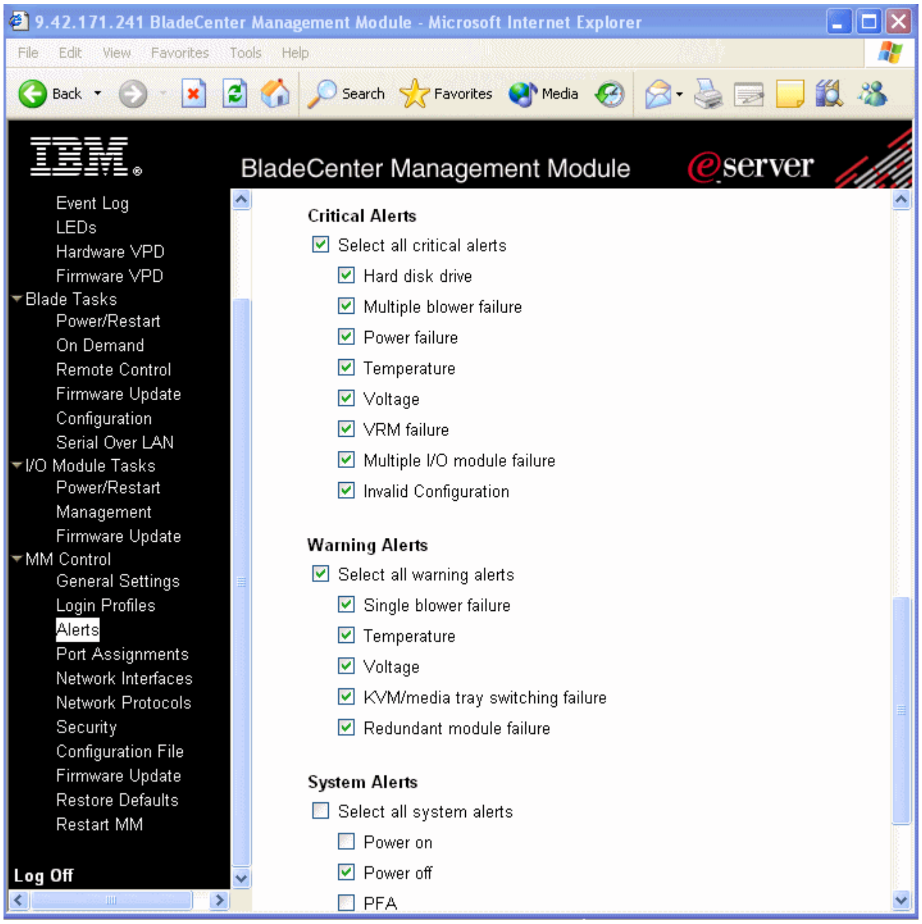
□	Bay	Type	MAC Address	IP Address	Pwr
□	1	Ethernet SM	00:05:5D:89:95:78	9.42.171.240	On
□	2	Ethernet SM	00:0D:88:FB:0F:CC	192.168.70.128	On
□	3	Fibre SM	00:C0:DD:02:0A:31	9.42.171.245	On
□	4	Fibre SM	00:C0:DD:01:BA:15	9.42.171.246	On

Power On Module(s)
Power Off Module(s)
Restart Module(s) and Run Standard Diagnostics
Restart Module(s) and Run Extended Diagnostics
Restart Module(s) and Run Full Diagnostics

Figure 3-15 Partial display of I/O Module Tasks

The final module controlled with the Web-based system management interface is the BladeCenter management module (Figure 3-16 on page 39). The tasks include:

- General module settings such as name, date, and time
- Login profiles with user ID and password control
- Alert level settings and user targets for receiving notification
- Port assignments
- Network interfaces
- Network protocols
- Security
- Configuration file management with backup and restore options
- Default restoration
- Management module



Critical Alerts

☑ Select all critical alerts

- ☑ Hard disk drive
- ☑ Multiple blower failure
- ☑ Power failure
- ☑ Temperature
- ☑ Voltage
- ☑ VRM failure
- ☑ Multiple I/O module failure
- ☑ Invalid Configuration

Warning Alerts

☑ Select all warning alerts

- ☑ Single blower failure
- ☑ Temperature
- ☑ Voltage
- ☑ KVM/media tray switching failure
- ☑ Redundant module failure

System Alerts

☐ Select all system alerts

- ☐ Power on
- ☑ Power off
- ☐ PFA

Figure 3-16 Sampling of alerts that can be monitored

> **Note:** BladeCenter is also tightly integrated to the IBM Director V4 systems management tool and the Rapid Deployment Manager; however, these system management products are beyond the scope of this paper.

3.3.3 Customizing BladeCenter I/O modules to match application needs

One of the strengths of a BladeCenter deployment is the flexibility to add modules that can enable the customized solutions that match application needs. From a physical perspective, adding these modules inside the chassis reduces the complexity of the external infrastructure by reducing cabling, power, and space requirements. From a management perspective, complexity is reduced by a centrally administered installation. Current I/O module options for BladeCenter include:

- Optical Pass-thru module that allows unswitched, unblocked network connections to each blade server
- 4-port Gigabit Ethernet Switch Module (Figure 3-17) that provides high-speed Ethernet connections to each blade server and the outside network environment
- Nortel Networks Layer 2-7 Gigabit Ethernet Switch module (Figure 3-17) integrating advanced Ethernet functionality into the chassis
- 2-port Fibre Channel Switch Module supporting two FC uplinks at transmission rates up to 2 Gbps

The requirements for standard gigabit Ethernet connectivity and connection to a fault-resilient Fibre Channel SAN for the proof-point were easily met by sliding the 4-port Gigabit Ethernet Switch Module and the 2-port Fibre Channel Switch Module into BladeCenter. These modules were immediately recognized by the BladeCenter management module and were available for configuration.

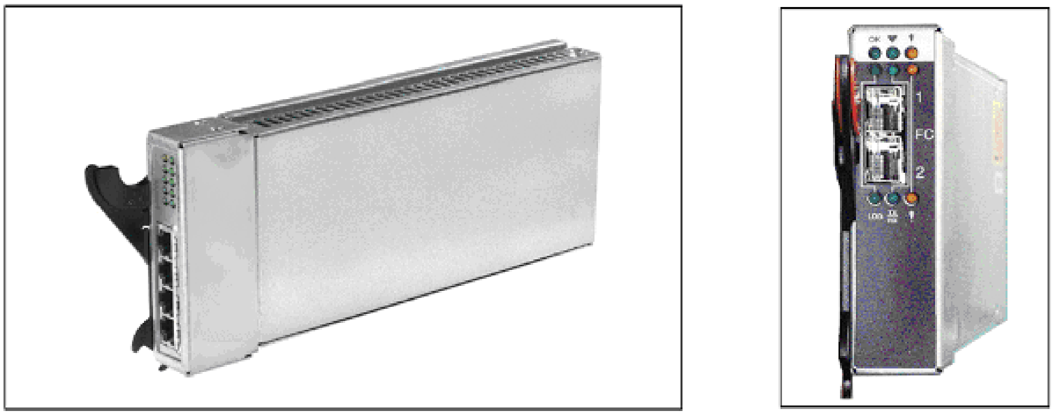


Figure 3-17 View of 4-port Ethernet Switch Module and 2-port Fibre Channel Switch Module

Managing the BladeCenter 4-port Gigabit Ethernet Switch Module

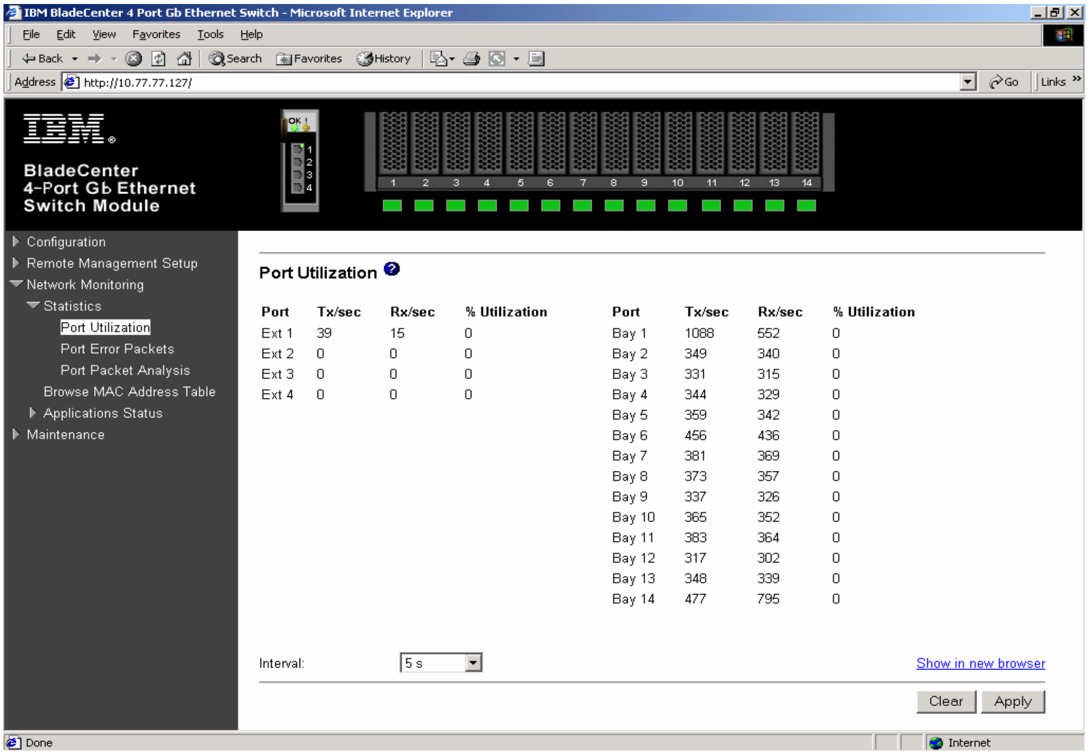
The Ethernet Switch Module can be managed through a Telnet session or a Web interface, which is the preferred method. Each module on the BladeCenter is assigned a TCP/IP address that corresponds to the module slot in which the switch is installed. This makes it easy to start a Telnet session directly or access the Web interface from the assigned address.

Using the Web interface, a DBA can manage and monitor performance of the Ethernet Switch Module with a mouse.

From the main BladeCenter Management menu, select **I/O Module Tasks** → **Management** and then click **Advanced Management** for the module of choice. This action provides a drill down into a Web-based, graphical tool that monitors the 4-port Gigabit switch (see Figure 3-18 on page 41 and Figure 3-19 on page 41 for examples).

For ease of navigation, the menu is broken into four distinct categories:

- Switch configuration for port settings, VLANs, link aggregation, and so on
- Remote management setup
- Network monitoring to display port utilization statistics
- Maintenance tasks such as upgrading firmware, downloading configuration, and restarting modules



Port Utilization

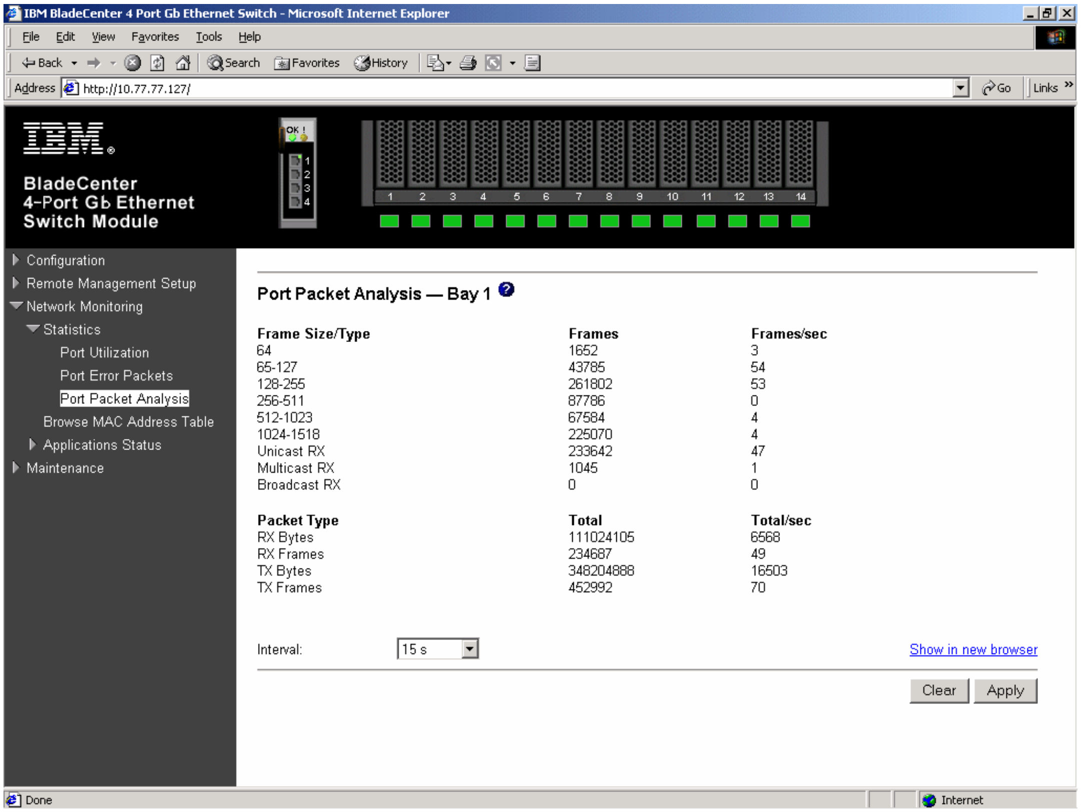
Port	Tx/sec	Rx/sec	% Utilization	Port	Tx/sec	Rx/sec	% Utilization
Ext 1	39	15	0	Bay 1	1088	552	0
Ext 2	0	0	0	Bay 2	349	340	0
Ext 3	0	0	0	Bay 3	331	315	0
Ext 4	0	0	0	Bay 4	344	329	0
				Bay 5	359	342	0
				Bay 6	456	436	0
				Bay 7	381	369	0
				Bay 8	373	357	0
				Bay 9	337	326	0
				Bay 10	365	352	0
				Bay 11	383	364	0
				Bay 12	317	302	0
				Bay 13	348	339	0
				Bay 14	477	795	0

Interval: 5 s

Show in new browser

Clear Apply

Figure 3-18 Port Utilization view for the 4-port Gigabit Ethernet Switch Module



Port Packet Analysis — Bay 1

Frame Size/Type	Frames	Frames/sec
64	1652	3
65-127	43785	54
128-255	261802	53
256-511	87786	0
512-1023	67584	4
1024-1518	225070	4
Unicast RX	233642	47
Multicast RX	1045	1
Broadcast RX	0	0

Packet Type	Total	Total/sec
RX Bytes	111024105	6568
RX Frames	234687	49
TX Bytes	348204888	16503
TX Frames	452992	70

Interval: 15 s

Show in new browser

Clear Apply

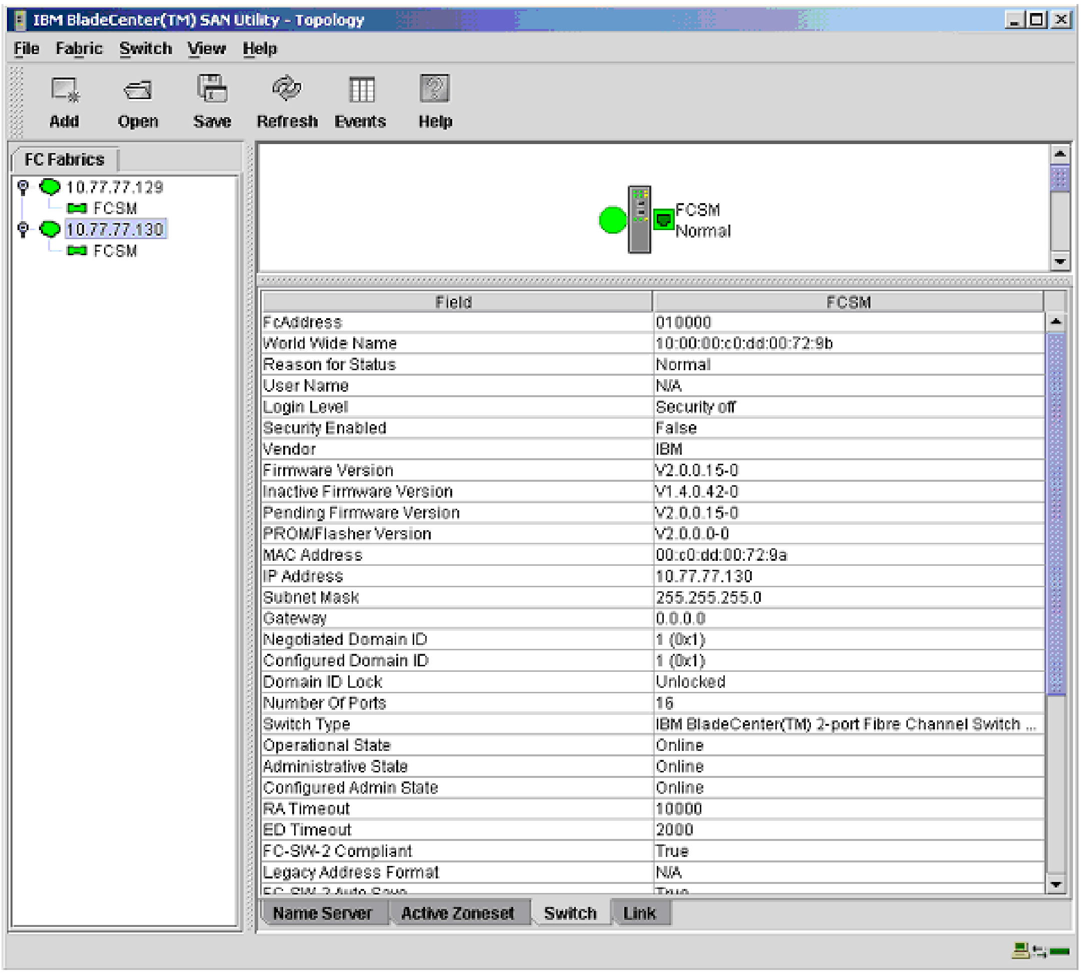
Figure 3-19 Port Packet Analysis for the 4-port Gigabit Ethernet Switch Module

Managing the BladeCenter Fibre Channel Switch Module

The SAN utility tool facilitates the configuration and management of the BladeCenter Fibre Channel Switch Module. This tool is a Java™-based graphical user interface that can be used to view and configure ports, zoning, and network setup. It also features diagnostic functionality for troubleshooting.

The SAN utility can be installed and executed from one of the blade servers or from an external monitoring system. It is supported in both Windows and Linux environments. The SAN utility provides a graphical view of the Fibre Channel subsystem. It can be used to view, monitor or change network, switch module, and port configurations for one or more fabrics concurrently. (A *fabric* is defined as one or more connected switches.)

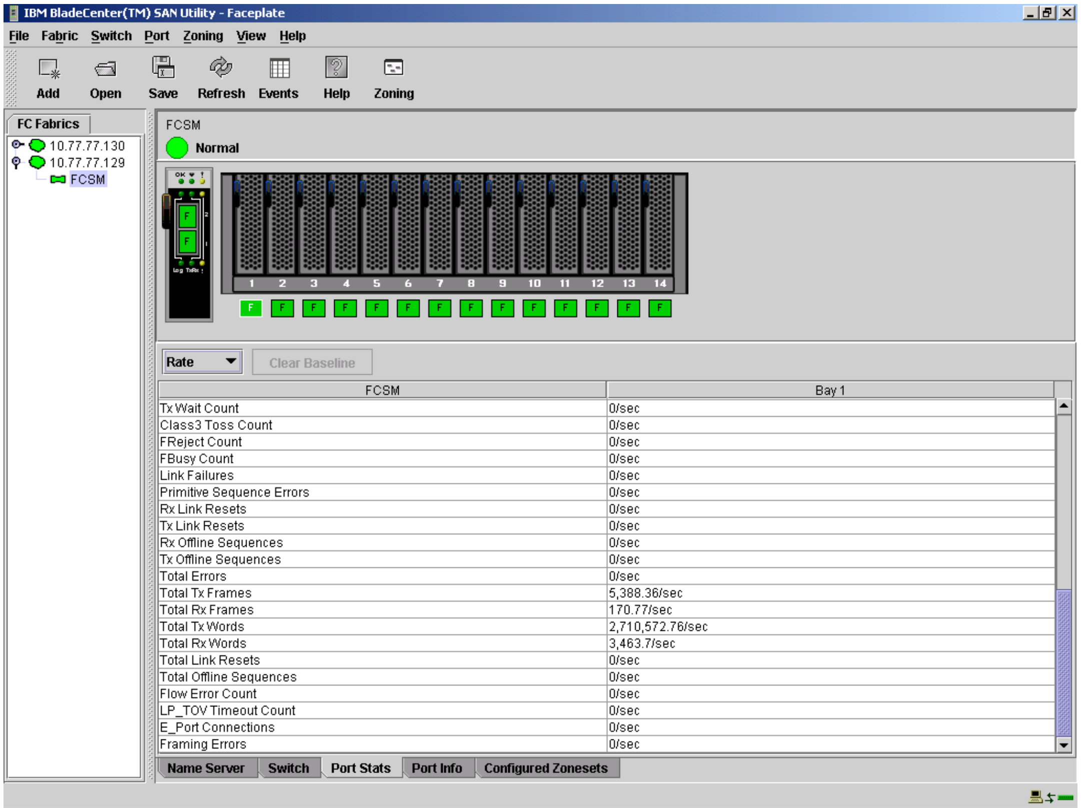
There are two basic views available with the SAN utility: the Topology display and the Faceplate display. The Topology display (Figure 3-20) shows all switches that are able to communicate and all the connections between the switches. The Topology display provides status information for switches, interswitch links, and the Ethernet connection.



Field	FCSM
FcAddress	010000
World Wide Name	10:00:00:c0:dd:00:72:9b
Reason for Status	Normal
User Name	N/A
Login Level	Security off
Security Enabled	False
Vendor	IBM
Firmware Version	V2.0.0.15-0
Inactive Firmware Version	V1.4.0.42-0
Pending Firmware Version	V2.0.0.15-0
PROM/Flasher Version	V2.0.0.0-0
MAC Address	00:c0:dd:00:72:9a
IP Address	10.77.77.130
Subnet Mask	255.255.255.0
Gateway	0.0.0.0
Negotiated Domain ID	1 (0x1)
Configured Domain ID	1 (0x1)
Domain ID Lock	Unlocked
Number Of Ports	16
Switch Type	IBM BladeCenter(TM) 2-port Fibre Channel Switch ...
Operational State	Online
Administrative State	Online
Configured Admin State	Online
RA Timeout	10000
ED Timeout	2000
FC-SW-2 Compliant	True
Legacy Address Format	N/A
FC-SW-2 Auto Save	True

Figure 3-20 SAN Utility Topology display

BladeCenter Fibre Channel Switch Module information can also be viewed through the Faceplate display (Figure 3-21). It shows the front of the switch and its related information: switch name, switch operational state, and port status.


IBM BladeCenter(TM) SAN Utility - Faceplate
File Fabric Switch Port Zoning View Help
Add Open Save Refresh Events Help Zoning
FC Fabrics
10.77.77.130
10.77.77.129
FCSM
FCSM
Normal
1 2 3 4 5 6 7 8 9 10 11 12 13 14
F F F F F F F F F F F F F F
Rate
Clear Baseline
Name Server Switch Port Stats Port Info Configured Zonesets

FCSM	Bay 1
Tx Wait Count	0/sec
Class3 Toss Count	0/sec
FReject Count	0/sec
FBusy Count	0/sec
Link Failures	0/sec
Primitive Sequence Errors	0/sec
Rx Link Resets	0/sec
Tx Link Resets	0/sec
Rx Offline Sequences	0/sec
Tx Offline Sequences	0/sec
Total Errors	0/sec
Total Tx Frames	5,388.36/sec
Total Rx Frames	170.77/sec
Total Tx Words	2,710,572.76/sec
Total Rx Words	3,463.7/sec
Total Link Resets	0/sec
Total Offline Sequences	0/sec
Flow Error Count	0/sec
LP_TOV Timeout Count	0/sec
E_Port Connections	0/sec
Framing Errors	0/sec

Figure 3-21 Faceplate display from the SAN Utility

BladeCenter also includes a management tool called Fabric View. This application provides a method to visually monitor real time traffic performance for each port on a switch. The graphs can be set up to display either kilobytes per second or number of frames per second.

3.3.4 Storage subsystem

The family of IBM TotalStorage DS4000 Series products is designed to support the large and growing data storage requirements of business-critical applications. The IBM TotalStorage DS4000 Series Storage Server is a RAID controller device that contains Fibre Channel interfaces that connect the host systems and the disk drive enclosures.

The IBM TotalStorage DS4400 can support up to 224 FC disks. To avoid single points of failure, it also supports the following high-availability features:

- Hot-swap RAID controllers
- Two dual redundant Fibre Channel disk loops
- Write cache mirroring
- Redundant hot-swap power supplies, fans, and dual AC line cords.

The DS4400 Storage Server used in this proof point is illustrated in Figure 3-22. It has controllers that use the 2 Gbps Fibre Channel standard on both the host and the drive.

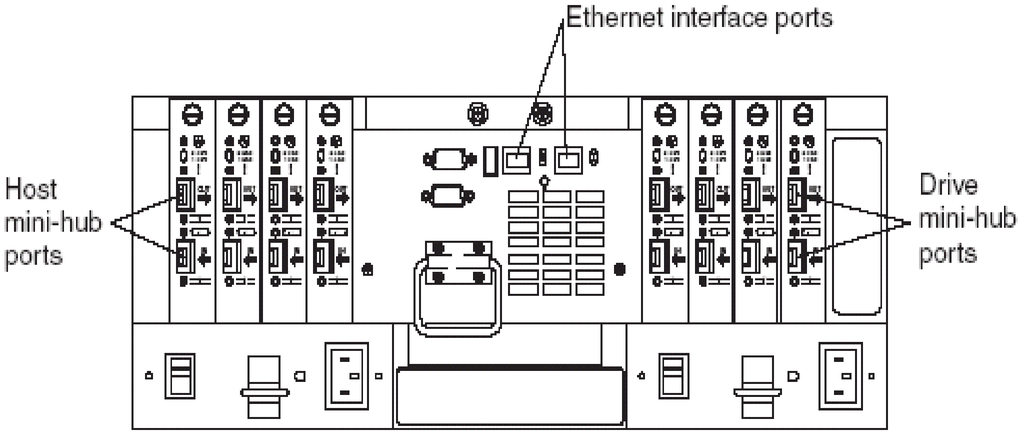


Figure 3-22 Rear view of the DS4400

For this implementation, the IBM TotalStorage DS4400 Storage Server had eight EXP700 drive enclosures attached with a total of 110 15K rpm 36.4GB drives. The storage manager software was used to automatically lay out each array across the multiple controllers and drive loops. The IBM TotalStorage DS4400 allocation heuristic can distribute I/O traffic across available resources, so there was no need to manually define the arrays. Because the IBM TotalStorage DS4400 Storage Server can support two redundant drive loops, the drive enclosures were set up to take advantage of this redundancy. If one data path fails, the controller uses the other data path to maintain the connection to the drive group.

The view of the storage server and drive enclosures in Figure 3-23 shows two redundant drive loops. Loop A and Loop B make up one redundant pair of drive loops. Loop C and Loop D make up a second redundant pair.

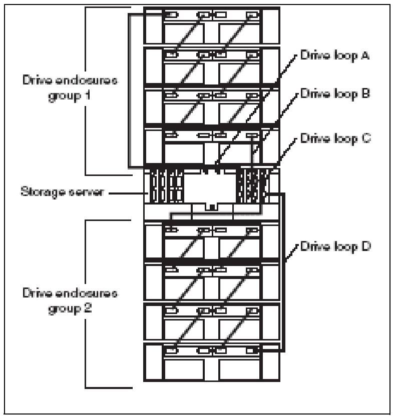


Figure 3-23 View showing drive loops

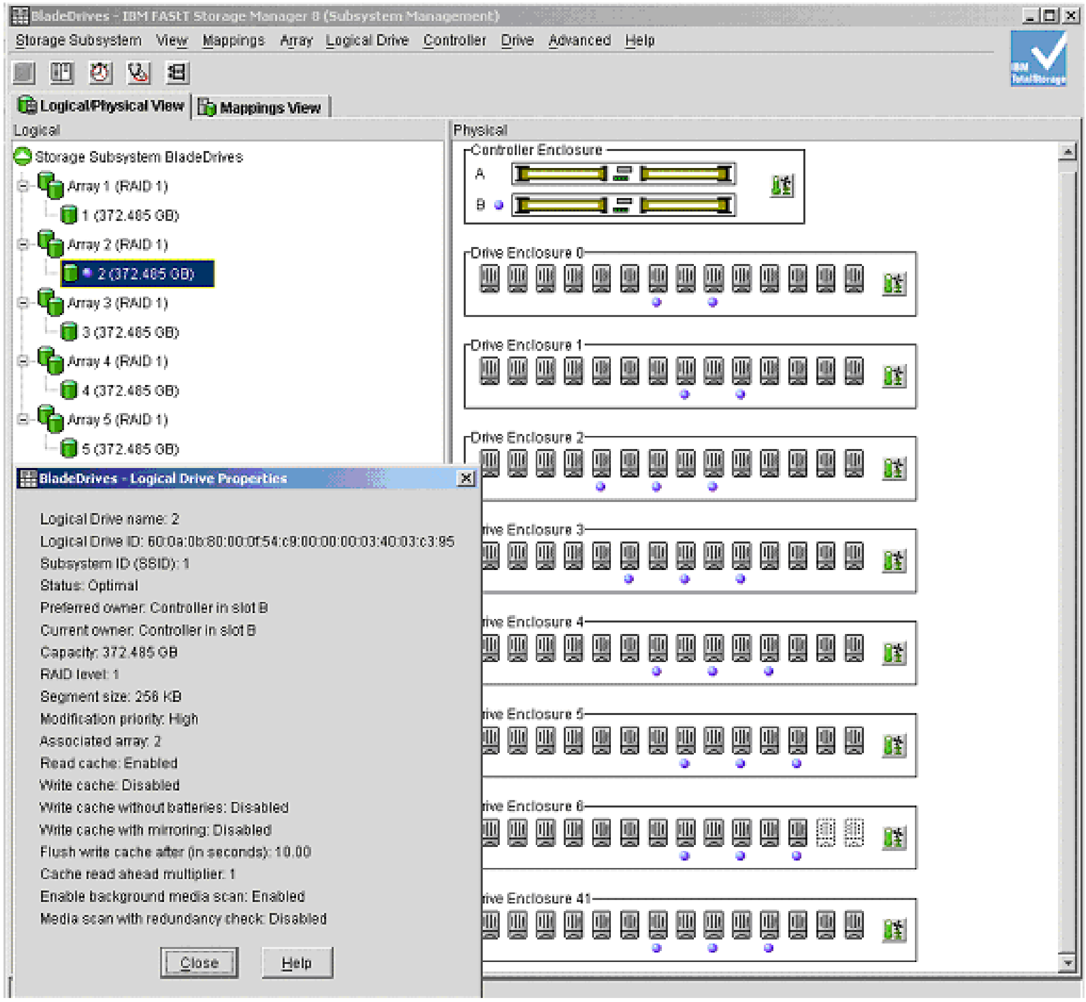
The 110 drives were automatically distributed across five arrays of 22 drives each with a RAID-1 configuration and a 256K stripe size.

Managing the storage system

The IBM TotalStorage DS4400 Storage Server management software simplifies the management of extensive SAN installations. As illustrated in Figure 3-24, DBAs can use it to:

- Configure arrays and logical drives
- Assign logical drives into storage partitions
- Replace and rebuild failed disk drives
- Expand the size of arrays and logical volumes
- Convert from one RAID level to another
- Monitor performance

In addition to array and drive level control, the IBM TotalStorage DS4400 Storage Sever management software can also be used to update the firmware and nonvolatile static random access memory (NVSRAM) on the controllers.



Logical Drive name: 2
Logical Drive ID: 60:0a:0b:80:00:0f:54:c9:00:00:00:03:40:03:c3:95
Subsystem ID (SSID): 1
Status: Optimal
Preferred owner: Controller in slot B
Current owner: Controller in slot B
Capacity: 372.485 GB
RAID level: 1
Segment size: 256 KB
Modification priority: High
Associated array: 2
Read cache: Enabled
Write cache: Disabled
Write cache without batteries: Disabled
Write cache with mirroring: Disabled
Flush write cache after (in seconds): 10.00
Cache read ahead multiplier: 1
Enable background media scan: Enabled
Media scan with redundancy check: Disabled

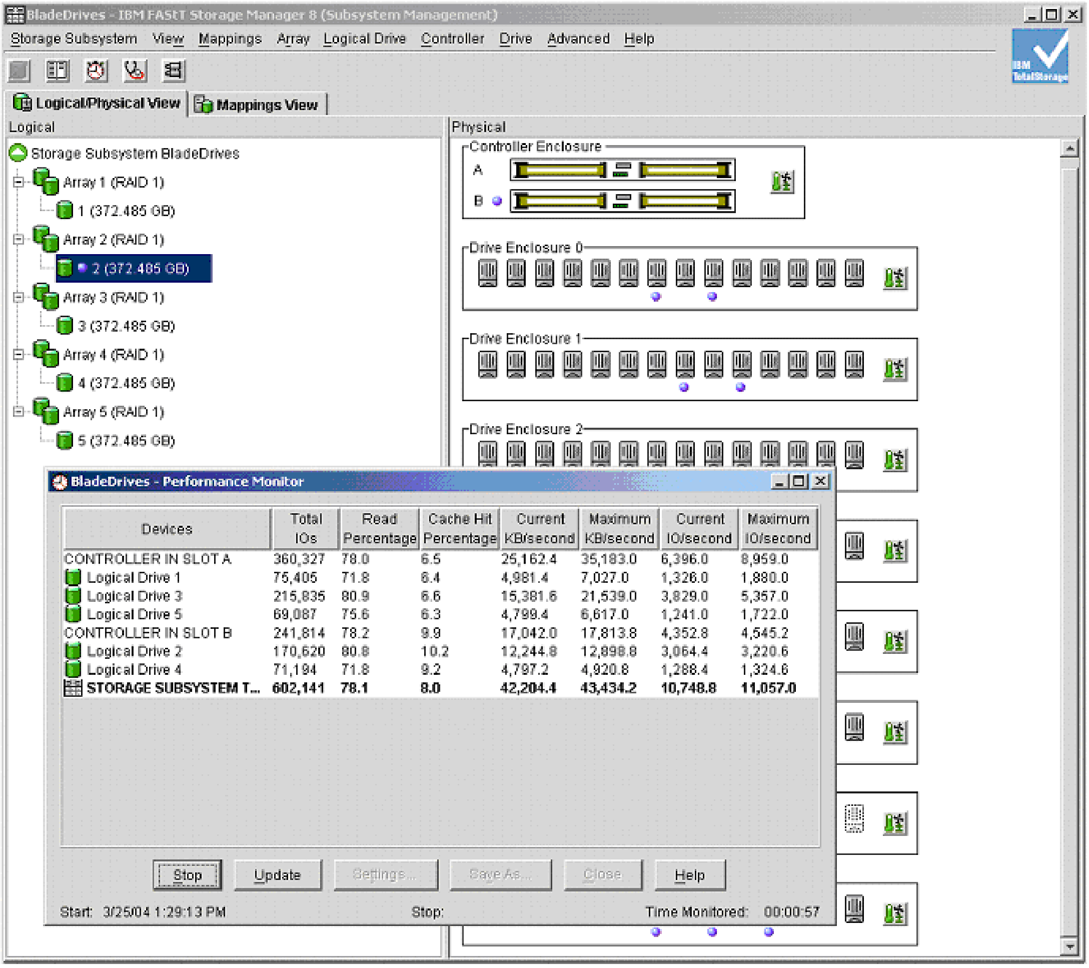
Figure 3-24 IBM TotalStorage DS4400 Storage Server Manager 8 (Subsystem Management)

The IBM TotalStorage DS4400 Storage Server management software is supported in both Windows and Linux environments and provides two methods for managing storage subsystems:

- Host-agent (in-band) management through the Fibre Channel I/O path to the host
- Direct (out-of-band) management over the network

The performance monitor data can be used to make storage subsystem tuning decisions. There are many settings in the IBM TotalStorage DS4400 Storage Server that can impact performance. These include cache parameters, controller ownership, segment size, RAID levels, logical drive modification priority, and remote volume mirroring. To assist with overall storage tuning, the IBM Storage Manager has a built-in performance monitor that displays statistics for total I/Os, read percentage, cache hit percentage, current kilobytes per second, maximum kilobytes per second, current I/Os, and maximum I/Os.

Tuning the IBM TotalStorage DS4400 Storage Server successfully, like other server components, depends on obtaining the correct balance of availability and high performance. The performance monitor, illustrated in Figure 3-25, in the IBM TotalStorage DS4400 Storage Server management software is just one of the tools available to help end users improve SAN performance.



Devices	Total IOs	Read Percentage	Cache Hit Percentage	Current KB/second	Maximum KB/second	Current IO/second	Maximum IO/second
CONTROLLER IN SLOT A	360,327	78.0	6.5	25,162.4	35,183.0	6,396.0	8,959.0
Logical Drive 1	75,405	71.8	6.4	4,981.4	7,027.0	1,326.0	1,880.0
Logical Drive 3	215,835	80.9	6.6	15,381.6	21,539.0	3,829.0	5,357.0
Logical Drive 5	69,087	75.6	6.3	4,799.4	6,617.0	1,241.0	1,722.0
CONTROLLER IN SLOT B	241,814	78.2	9.9	17,042.0	17,813.8	4,352.8	4,545.2
Logical Drive 2	170,620	80.8	10.2	12,244.8	12,898.8	3,064.4	3,220.6
Logical Drive 4	71,194	71.8	9.2	4,797.2	4,920.8	1,288.4	1,324.6
STORAGE SUBSYSTEM T...	**602,141**	**78.1**	**8.0**	**42,204.4**	**43,434.2**	**10,748.8**	**11,057.0**

Start: 3/25/04 1:29:13 PM Stop: Time Monitored: 00:00:57

Figure 3-25 Performance monitor

3.3.5 Overview of databases

To test the flexible database cluster architecture, three databases were created in the PolyServe Matrix Server cluster file system using Oracle9*i* Release 2 version 9.2.0.4. The main databases were called OLTP and DSS. The third, smaller database was called DEV.

These databases were not intended to convey best practices for operations or performance, with the notable exception of the use of OMF. OMF is Oracle9*i* file management simplification, which deserves close consideration in any deployment scenario. The databases were intended only to be realistic in size and structure for the purpose of testing the functionality and added value of flexible database cluster architecture.

OLTP database (PROD)

The OLTP database schema is based on an order-entry system similar to but not compliant with that defined in the TPC-C benchmark specified by the Transaction Processing Performance Council. At a high level, the database schema contains the following application tables:

- Customers

  The database contains more than 159,000,000 customer rows in the customer table. This table contains customer-centric data such as a unique customer identifier, mailing address, e-mail contact information, and so on. The customer table is indexed with a unique index on the custid column and a non-unique index on the name column.

- Orders

  The database contains an orders table with more than 200,000,000 rows of data. The orders table has a unique composite index on the custid and ordid columns.

- Line items

  Simulating a customer base with complex transactions, the line item table contains as many as 15 line items per order for a total of nearly 1,700,000,000 rows. The item table has a three-way unique composite index on custid, ordid, and itemid. Creating this index was one of the timed PQO tasks described in “DSS” on page 60.

- Product

  This table describes products available for ordering. Along with such attributes as price and description, there are up to 140 characters available for a detailed product description. There are more than 140,000,000 products. The product table is indexed with a unique index on its prodid column.

- Warehouse

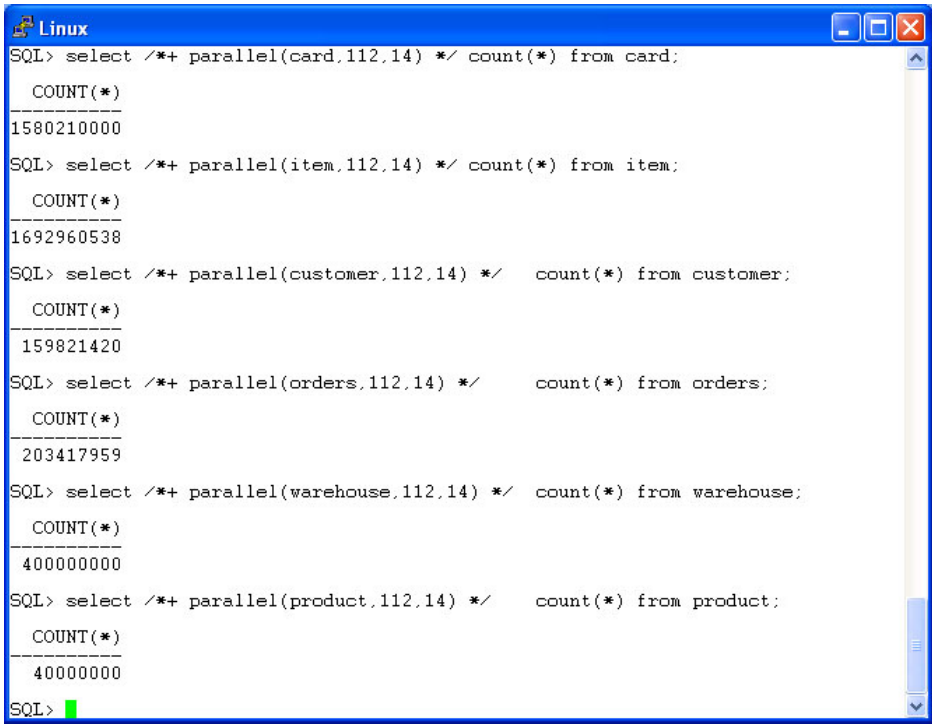
  This table maintains product levels at the various warehouse locations and detailed information about the warehouses. This table is critical for order fulfillment. The warehouse table is indexed with a unique composite index of two columns.

- Card

  The card table holds credit card transaction detail. This table is also used by the DSS database instances as a transportable tablespace. The card table has nearly 1,600,000,000rows of data as shown in Figure 3-26 on page 48.

Note: The flexible database cluster proof of concept was not intended to comply with any specification of the Transaction Processing Performance Council. For more information about TPC-C, visit this Web site:

`http://www.tpc.org`

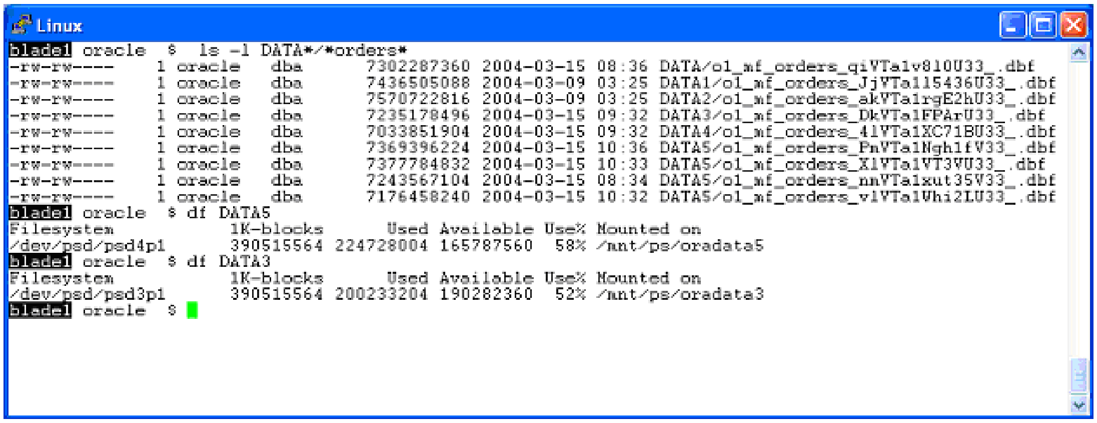


```
SQL> select /*+ parallel(card,112,14) */ count(*) from card;

  COUNT(*)
----------
1580210000

SQL> select /*+ parallel(item,112,14) */ count(*) from item;

  COUNT(*)
----------
1692960538

SQL> select /*+ parallel(customer,112,14) */   count(*) from customer;

  COUNT(*)
----------
 159821420

SQL> select /*+ parallel(orders,112,14) */     count(*) from orders;

  COUNT(*)
----------
 203417959

SQL> select /*+ parallel(warehouse,112,14) */  count(*) from warehouse;

  COUNT(*)
----------
 400000000

SQL> select /*+ parallel(product,112,14) */    count(*) from product;

  COUNT(*)
----------
  40000000

SQL>
```

Figure 3-26   The card table

The database was created using the simplified OMF method. The flexible database cluster proof of concept was meant to prove manageability in a complex environment, so complex tablespace definitions were not necessary. Tablespaces created by OMF are optimized for normal operations; however, specialized tuning may still be required in certain cases.

To help illustrate the simplicity of this type of file management, Figure 3-27 shows a simple **ls(1)** command that displays all of the data files associated with the ORDERS tablespace. The data files are named automatically by OMF, which enables unique naming.



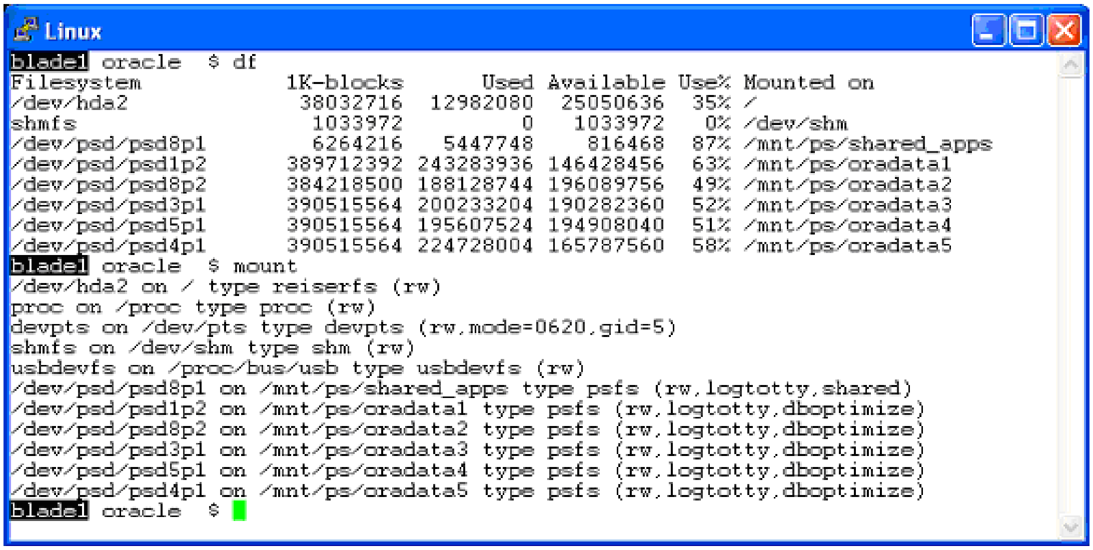
```
blade1 oracle  $  ls -l DATA*/*orders*
-rw-rw----     1 oracle   dba      7302287360 2004-03-15 08:36 DATA/o1_mf_orders_qiVTa1v810U33_.dbf
-rw-rw----     1 oracle   dba      7436505088 2004-03-09 03:25 DATA1/o1_mf_orders_JjVTa115436U33_.dbf
-rw-rw----     1 oracle   dba      7570722816 2004-03-09 03:25 DATA2/o1_mf_orders_akVTa1rgE2hU33_.dbf
-rw-rw----     1 oracle   dba      7235178496 2004-03-15 09:32 DATA3/o1_mf_orders_DkVTa1FPArU33_.dbf
-rw-rw----     1 oracle   dba      7033851904 2004-03-15 09:32 DATA4/o1_mf_orders_41VTa1XC71BU33_.dbf
-rw-rw----     1 oracle   dba      7369396224 2004-03-15 10:36 DATA5/o1_mf_orders_PnVTa1Ngh1fV33_.dbf
-rw-rw----     1 oracle   dba      7377784832 2004-03-15 10:33 DATA5/o1_mf_orders_X1VTa1VT3VU33_.dbf
-rw-rw----     1 oracle   dba      7243567104 2004-03-15 08:34 DATA5/o1_mf_orders_nnVTa1xut35V33_.dbf
-rw-rw----     1 oracle   dba      7176458240 2004-03-15 10:32 DATA5/o1_mf_orders_v1VTa1Vhi2LU33_.dbf
blade1 oracle  $ df DATA5
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/psd/psd4p1      390515564 224728004 165787560  58% /mnt/ps/oradata5
blade1 oracle  $ df DATA3
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/psd/psd3p1      390515564 200233204 190282360  52% /mnt/ps/oradata3
blade1 oracle  $
```

Figure 3-27   ls(1) and df(1) commands

Since the databases were created in the PolyServe Matrix Server cluster file system, ascertaining the amount of free space available for the databases was a simple task. Figure 3-27 also shows a simple **df** command that lists free and used space for some of the files in the databases.

All database files were created in the oradata[1-5] file systems. The **mount(1)** command in Figure 3-28 shows that they are all mounted with the PolyServe Matrix Server dboptimized mount option, which provides cache-coherent direct I/O. This means that database accesses by Oracle are rendered directly.

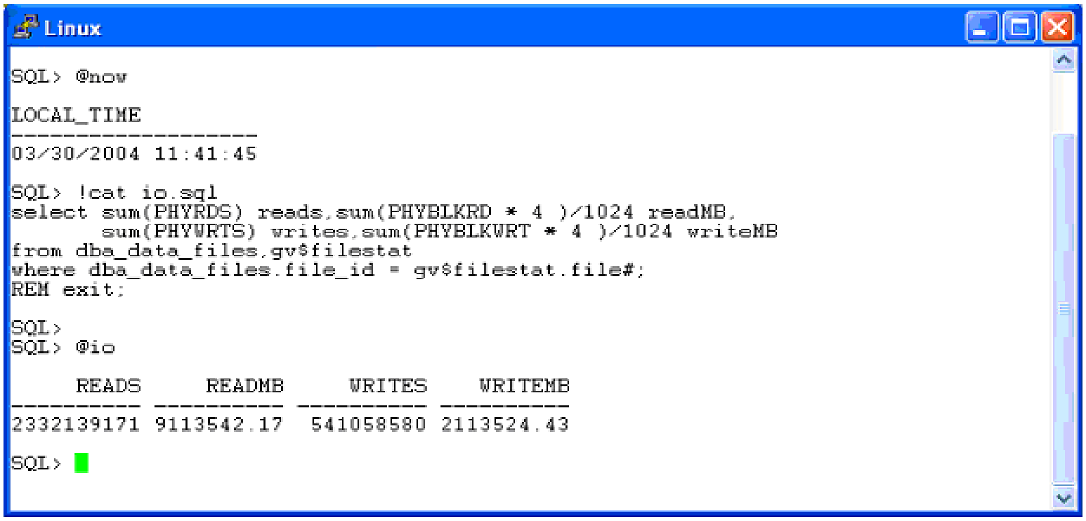
Other tools (for example, backup tools) can also obtain direct I/O without being recompiled (with code changes) to perform the **open(2)** system call with the O_DIRECT flag. With the dboptimized mount option, if I/O can be rendered directly, it will be. I/O that cannot be rendered direct is serviced through the buffered I/O path and cache coherency is maintained throughout.



```
blade1 oracle  $ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda2             38032716  12982080  25050636  35% /
shmfs                  1033972         0   1033972   0% /dev/shm
/dev/psd/psd8p1        6264216   5447748    816468  87% /mnt/ps/shared_apps
/dev/psd/psd1p2      389712392 243283936 146428456  63% /mnt/ps/oradata1
/dev/psd/psd8p2      384218500 188128744 196089756  49% /mnt/ps/oradata2
/dev/psd/psd3p1      390515564 200233204 190282360  52% /mnt/ps/oradata3
/dev/psd/psd5p1      390515564 195607524 194908040  51% /mnt/ps/oradata4
/dev/psd/psd4p1      390515564 224728004 165787560  58% /mnt/ps/oradata5
blade1 oracle  $ mount
/dev/hda2 on / type reiserfs (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
shmfs on /dev/shm type shm (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
/dev/psd/psd8p1 on /mnt/ps/shared_apps type psfs (rw,logtotty,shared)
/dev/psd/psd1p2 on /mnt/ps/oradata1 type psfs (rw,logtotty,dboptimize)
/dev/psd/psd8p2 on /mnt/ps/oradata2 type psfs (rw,logtotty,dboptimize)
/dev/psd/psd3p1 on /mnt/ps/oradata3 type psfs (rw,logtotty,dboptimize)
/dev/psd/psd5p1 on /mnt/ps/oradata4 type psfs (rw,logtotty,dboptimize)
/dev/psd/psd4p1 on /mnt/ps/oradata5 type psfs (rw,logtotty,dboptimize)
blade1 oracle  $ 
```

Figure 3-28 mount(1) options

To illustrate that the proof of concept was more than a typical benchmark, Figure 3-29 shows a query against the Oracle GV$ virtual tables. The query indicates that the database instances had been active without stopping and had amassed nearly 3,000,000,000 physical I/O transfers. The io.sql script also reveals that the instances had collectively transferred roughly 11 TB of data without rebooting.



```
SQL> @now

LOCAL_TIME
-------------------
03/30/2004 11:41:45

SQL> !cat io.sql
select sum(PHYRDS) reads,sum(PHYBLKRD * 4 )/1024 readMB,
       sum(PHYWRTS) writes,sum(PHYBLKWRT * 4 )/1024 writeMB
from dba_data_files,gv$filestat
where dba_data_files.file_id = gv$filestat.file#;
REM exit;

SQL>
SQL> @io

     READS     READMB     WRITES    WRITEMB
---------- ---------- ---------- ----------
2332139171 9113542.17  541058580 2113524.43

SQL> 
```

Figure 3-29 Query showing I/O activity

To put this in perspective, even at the observed peaks of 14,000 I/O per second, it would take that I/O rate two and a half days to reach it. That is not a benchmark. In reality, this test was specifically set up to execute the OLTP test on all 14 blades for an entire weekend.

DSS database: decision support

The DSS database was used to perform analytical queries about customer credit. The table used for this decision support was the card table from the OLTP schema described in “OLTP database (PROD)” on page 47. The card tablespace was simply set up as a transportable tablespace and accessed directly by the DSS database without copying. This is the power of a large cluster on a SAN. It becomes very efficient to take data from one database to another with no copies across a network.

DEV database: development

The DEV database is a simple insert engine designed to test scalability while inserting 2KB records. The database is approximately10GB. Only two threads are defined; therefore, only two instances can access this database at one time.

3.3.6 Workload descriptions

The workloads chosen for the flexible database cluster proof of concept were not as important as the fact that there were three of them. As stated earlier, the databases were called PROD, DSS, and DEV. The sections that follow describe the type of processing each database sustained during the test. The goal was to have a realistic mix of processing running on the system while testing the manageability of the flexible database cluster architecture.

OLTP workload

Simulating an order-entry system, the application accessing the PROD database connects100 users per node using the PROD service defined in the tnsnames service definition. The nodes being tested are evenly loaded due to the load balancing attribute of the PROD SQL*Net service.

Each user cycles through a set of transactions. At the end of each transaction, the client process sleeps for a small random period of time to simulate human interaction. To that end, this testing is not a typical benchmark, whereby all processors are 100% utilized. Such a condition is not desirable in a datacenter scenario; therefore, testing manageability of a proposed architecture under those conditions was not deemed realistic.

Note: The value of traditional benchmarks is not being questioned. Hardware vendors need a fair playing field to establish the capabilities of their offerings.

A key attribute of this testing is that it was completely void of traditional cluster-aware tuning. Most cluster-centric database tests have a form of application-level partitioning. For example, nearly all TPC-C benchmarks executed on a cluster use a method called data-dependent request routing.

This method uses a transaction monitor to route all requests for a given transaction to a node in the cluster provisioned for servicing requests that modify data within a certain key range. For example, node 1 in the cluster accepts new order transaction requests only from customers with customer identifiers in the range of 1-1,000,000; node 2 services the 1,000,001-2,000,000 range; and so on.

The limitation of such partitioning schemes is that they require changes to the application. Applications should not have to change to scale horizontally in a clustered environment, and with Oracle9*i* Real Application Clusters, they do not. Oracle9*i* Real Application Clusters is a radical departure from typical clustered database technology. With its Cache Fusion Technology and shared disk architecture, off-the-shelf applications can fully exploit clustered systems, without cluster-centric tuning or application code modifications. To that end, the flexible database cluster proof of concept used an application test that did not require any cluster-centric tuning.

The test application users connect to any Oracle instance in the cluster and execute transactions as though they were running on a legacy SMP system. In fact, this test application has been used in the past to test SMP scalability.

The transaction details are as follows:

- Orders query: This transaction accounts for 16% of the activity. It provides top-level detail on existing orders for the customer and presents it in descending order from most current to least current.
- Customer attribute update: This transaction represents 26% of the workload. It updates information such as phone numbers, addresses, and credit card information.
- Orders report: This transaction differs from Orders Query because it offers full order detail for a customer that includes shipment status. This transaction is executed 4% of the time.
- Product update: This transaction occurs when a product description changes; 26% of all transactions are product updates.
- New items: This transaction accounts for 11% of the activity; it adds items into stock on hand.
- New orders: This transaction simulates taking an order from an existing customer for stock on hand. It accounts for 17% of the total transactions.

The physical I/O mix for the workload was 79% reads and 21% writes. On average, each transaction has the costs shown in Table 3-1:

Table 3-1 Average cost

Oracle statistics	Average per transaction
SGA Logical Reads	46.9
SQL Executions	2.7
Physical I/O	8.2
Block Changes	5.9

The Oracle statistics show that the workload is quite formidable. We encourage comparisons of these statistics to those taken from production systems with similar CPU count. This is a very realistic workload.

DSS workload

The DSS database performed analytical queries about customer credit. The fact table used for this decision support was the credit card activity table from the OLTP schema. The card tablespace was set up as a transportable tablespace and accessed directly without copy by the DSS database.

This configuration shows the power of a large cluster on a SAN. It becomes very efficient to take data from one database to another without copying across a network.

DEV workload

The DEV workload is simply a program that inserts 2K rows via pipe to SQL*Loader. The streams of loader processes execute on up to two nodes when DEV is being tested with the other workloads.

3.3.7 Measurement results

This section presents the results of a test that was conducted to measure system availability for the flexible database cluster proof of concept.

High availability and manageability

Oracle9*i* Real Application Clusters is known for scalability, but it also offers high availability. When it is combined with the power of SQL*Net, there will almost always be an instance of a database to which to connect.

The full capability of Oracle9*i* Real Application Clusters for high availability is limited, however, in small clustered environments. Once again, the flexible database cluster adds tremendous architectural and operational value.

Consider an application that is sized to require a 4-node cluster. If a node suffers hardware failure, the application is now serviced at no better than 75%. Until the hardware is either repaired or replaced, this condition will persist. Of course, there is still 100% uptime for the application due to the power of Oracle9*i* Real Application Clusters, but there would likely be users that can detect the loss of performance related to the reduced node count.

One form of protection is to fully load a spare node that is ready to be cabled and booted with Oracle[13]. This is a requirement in the absence of a cluster file system-based shared Oracle Home such as that configured for the flexible database cluster testing.

The question is whether it will have the right personality to replace the failed node. That is, with OCMS, the cmcfg.ora parameter called PrivateNodeNames contains a list of host names or IP addresses. Neither can be changed without rebooting OCMS on all nodes in the cluster. Of course, all database instances must be down to reboot OCMS. Similarly, tnsnames.ora, listener.ora, and many other configuration files likely expect the replacement node to possess, at a minimum, the same IP address and hostname.

These issues are not insurmountable, but they do tax the continuity of operations in the face of node failure. Every minute counts when a database is running at 75% capacity.

By contrast, this same application serviced by four nodes in a flexible database cluster environment can rapidly return to 100% capacity after one or even several concurrent hardware failures.

Fortunately, suffering multiple hardware failures is not common, but it can happen. This is one of the reasons Oracle9*i* Real Application Clusters is the right choice for mission-critical deployments.

During the proof of concept, a test was set up to measure the added availability the architecture provides. In summary, the test consisted of the following processes:

- Two thousand users were connected to a 10-node cluster executing OLTP.
- One of the 10 nodes was turned off.
- All users connected to the nine remaining nodes maintained their connection to Oracle.

Figure 3-30 has a time-line graphic that depicts the events that occurred during the test.

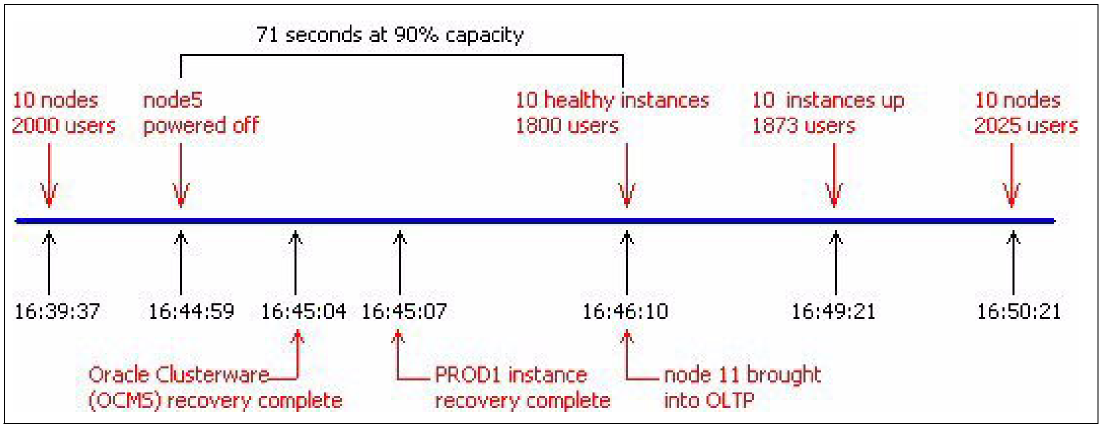
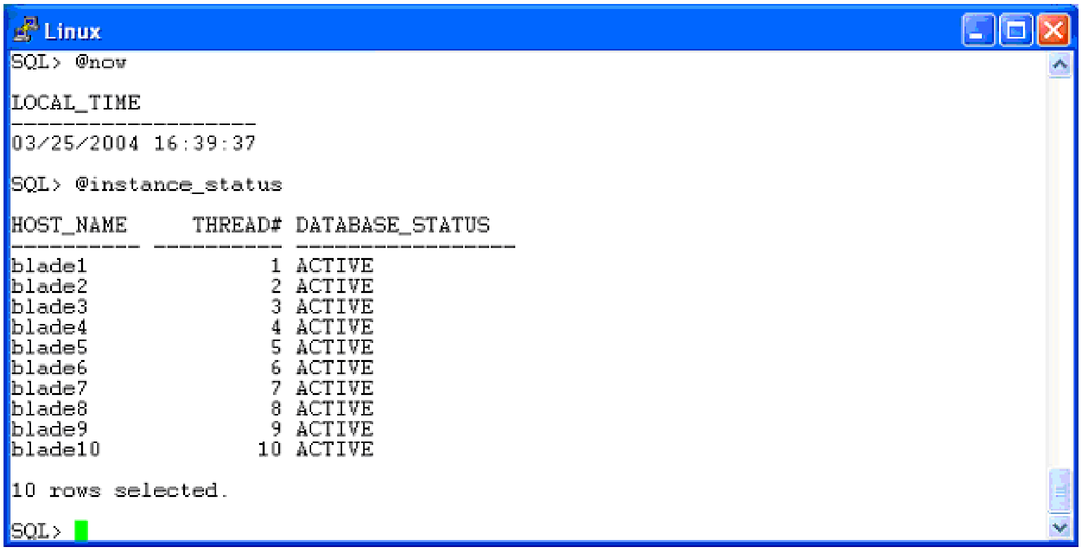


Figure 3-30 Availability time line

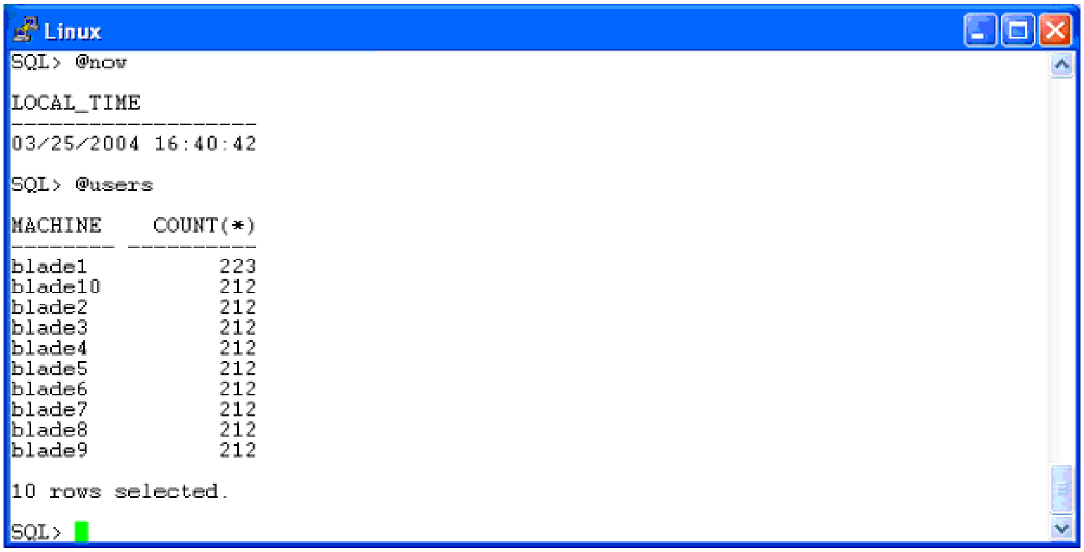
A replacement node was up with an instance of Oracle accepting connections 71 seconds after the node was turned off. The OLTP workload was set up to execute at 200 users per node on blades 1 through10. A light DSS database workload was executing on the remaining four blades. Figure 3-31 shows that at 16:39:37, the instance_status.sql script reported that there were 10 active instances on blades 1 through10.



```
SQL> @now

LOCAL_TIME
-------------------
03/25/2004 16:39:37

SQL> @instance_status

HOST_NAME      THREAD# DATABASE_STATUS
---------- ---------- -----------------
blade1              1 ACTIVE
blade2              2 ACTIVE
blade3              3 ACTIVE
blade4              4 ACTIVE
blade5              5 ACTIVE
blade6              6 ACTIVE
blade7              7 ACTIVE
blade8              8 ACTIVE
blade9              9 ACTIVE
blade10            10 ACTIVE

10 rows selected.

SQL>
```

Figure 3-31 Active instances on blades 1-10

In Figure 3-32 on page 54, the users.sql script shows that there were 200 user connections per node. The script also counted 12 Oracle background processes per node.

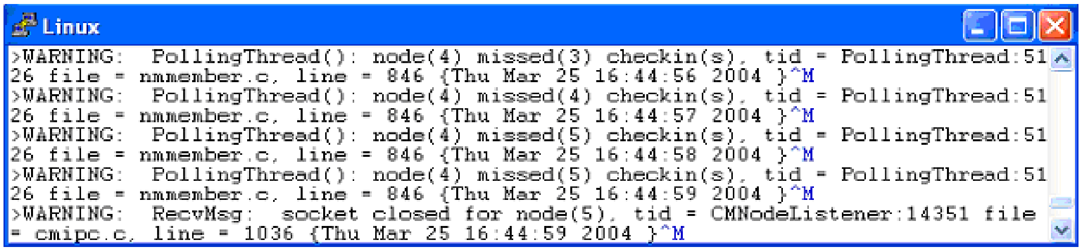


```
SQL> @now

LOCAL_TIME
-------------------
03/25/2004 16:40:42

SQL> @users

MACHINE    COUNT(*)
--------- ---------
blade1          223
blade10         212
blade2          212
blade3          212
blade4          212
blade5          212
blade6          212
blade7          212
blade8          212
blade9          212

10 rows selected.

SQL>
```

Figure 3-32   users.sql script

The workload was allowed to come to a steady state of transaction execution. After some time, blade5 was turned off with the BladeCenter management console to simulate a node failure.

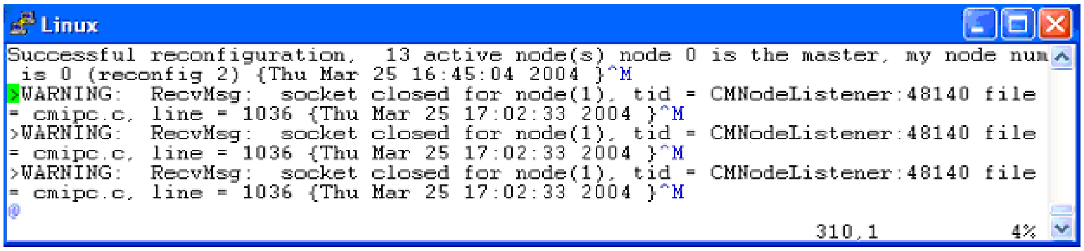
Figure 3-33 shows the OCMS cm.log file from blade1 logging the event that node 4 (counting from zero) was no longer responding at 16:44:59.



```
>WARNING:  PollingThread(): node(4) missed(3) checkin(s), tid = PollingThread:51
26 file = nmmember.c, line = 846 {Thu Mar 25 16:44:56 2004 }^M
>WARNING:  PollingThread(): node(4) missed(4) checkin(s), tid = PollingThread:51
26 file = nmmember.c, line = 846 {Thu Mar 25 16:44:57 2004 }^M
>WARNING:  PollingThread(): node(4) missed(5) checkin(s), tid = PollingThread:51
26 file = nmmember.c, line = 846 {Thu Mar 25 16:44:58 2004 }^M
>WARNING:  PollingThread(): node(4) missed(5) checkin(s), tid = PollingThread:51
26 file = nmmember.c, line = 846 {Thu Mar 25 16:44:59 2004 }^M
>WARNING:  RecvMsg:  socket closed for node(5), tid = CMNodeListener:14351 file
= cmipc.c, line = 1036 {Thu Mar 25 16:44:59 2004 }^M
```

Figure 3-33   cm.log file from blade1

After the Oracle instance on blade5 was turned off, OCMS had to perform cluster membership recovery. Figure 3-34 shows that the cm.log file registered that the OCMS recovery was complete at 16:45:04. This shows that both OCMS and PolyServe Matrix Server were able to respond to node 5 leaving the cluster within15 seconds. For a cluster of 14 nodes, this is remarkable.



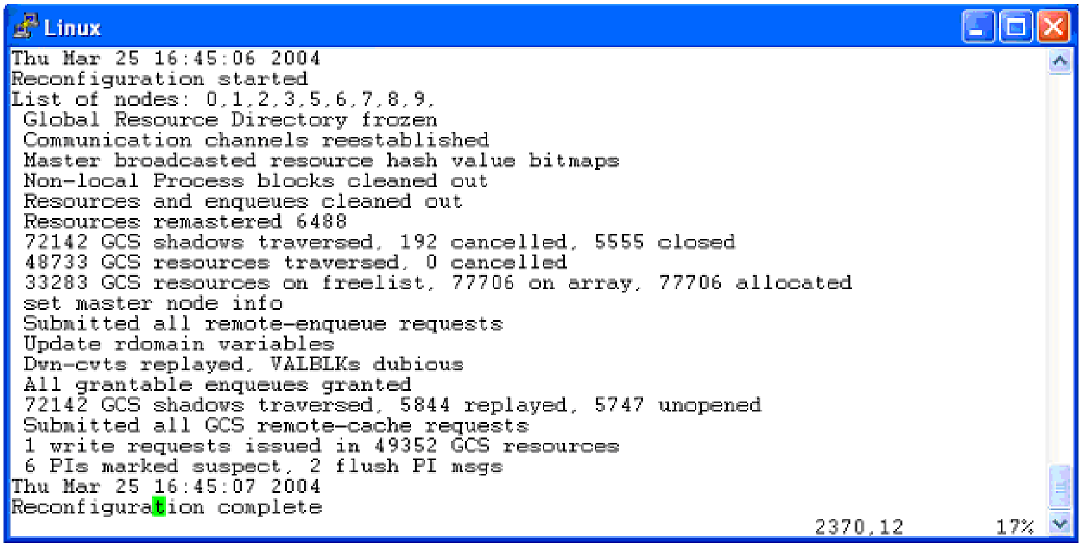
```
Successful reconfiguration,  13 active node(s) node 0 is the master, my node num
 is 0 (reconfig 2) {Thu Mar 25 16:45:04 2004 }^M
>WARNING:  RecvMsg:  socket closed for node(1), tid = CMNodeListener:48140 file
= cmipc.c, line = 1036 {Thu Mar 25 17:02:33 2004 }^M
>WARNING:  RecvMsg:  socket closed for node(1), tid = CMNodeListener:48140 file
= cmipc.c, line = 1036 {Thu Mar 25 17:02:33 2004 }^M
>WARNING:  RecvMsg:  socket closed for node(1), tid = CMNodeListener:48140 file
= cmipc.c, line = 1036 {Thu Mar 25 17:02:33 2004 }^M
@
                                                          310,1          4%
```

Figure 3-34   cm.log file showing recovery

The state of the database instances must also be taken into consideration. Did instances of Oracle on nodes other then blade5 suffer an outage?

This test proved the remarkable cluster recovery of Oracle9*i* Real Application Clusters. Because Oracle executables are stored in the PolyServe Matrix Server cluster file system, Oracle cannot commence instance recovery until the PolyServe Matrix Server recovery is complete. In the test, this occurred in 15 seconds, which is very quick.

PolyServe Matrix Server and OCMS recovery completed at 16:45:04, and the OLTP instance on blade1 (PROD1) started reconfiguration just 2 seconds later. According to the alert log information shown in Figure 3-35, the PROD1 instance finished reconfiguring only 1 second later.

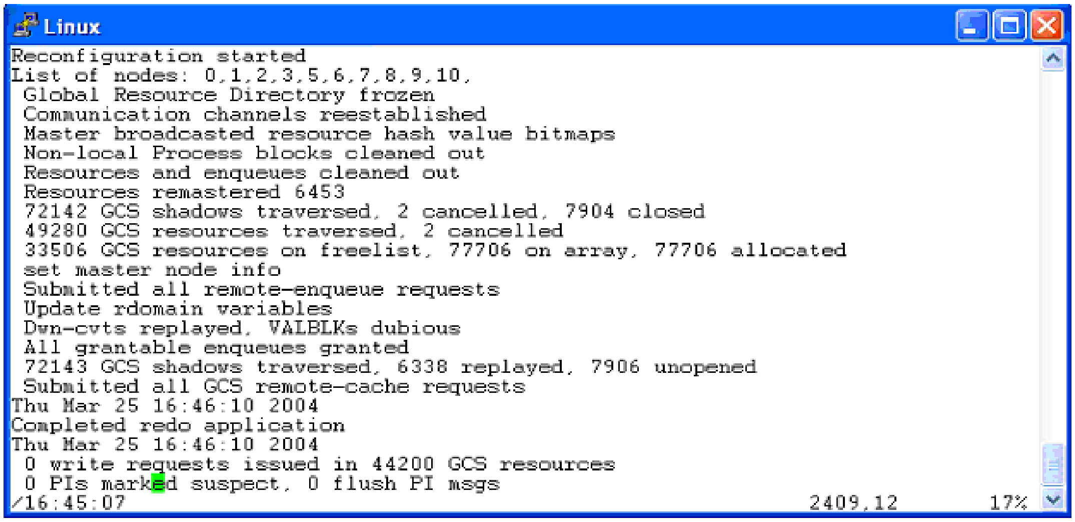


```
Thu Mar 25 16:45:06 2004
Reconfiguration started
List of nodes: 0,1,2,3,5,6,7,8,9,
 Global Resource Directory frozen
 Communication channels reestablished
 Master broadcasted resource hash value bitmaps
 Non-local Process blocks cleaned out
 Resources and enqueues cleaned out
 Resources remastered 6488
 72142 GCS shadows traversed, 192 cancelled, 5555 closed
 48733 GCS resources traversed, 0 cancelled
 33283 GCS resources on freelist, 77706 on array, 77706 allocated
 set master node info
 Submitted all remote-enqueue requests
 Update rdomain variables
 Dwn-cvts replayed, VALBLKs dubious
 All grantable enqueues granted
 72142 GCS shadows traversed, 5844 replayed, 5747 unopened
 Submitted all GCS remote-cache requests
 1 write requests issued in 49352 GCS resources
 6 PIs marked suspect, 2 flush PI msgs
Thu Mar 25 16:45:07 2004
Reconfiguration complete
                                                            2370,12       17%
```

Figure 3-35 PROD1 instance reconfiguration

The manageability of the flexible database cluster architecture proved invaluable. Having such a large cluster in support of various applications means that a node may be repurposed in response to a server failure, which we simulated by turning off blade5.

Figure 3-36 shows that the reconfiguration of OCMS nodes 0-3 and 5-10 (counting from zero) was complete by 16:46:10.

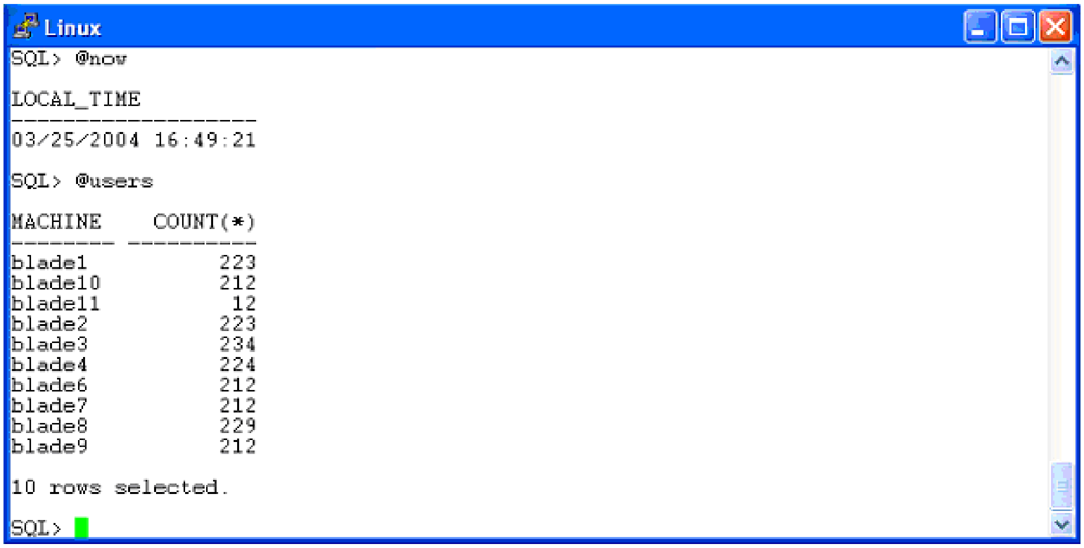


```
Reconfiguration started
List of nodes: 0,1,2,3,5,6,7,8,9,10,
 Global Resource Directory frozen
 Communication channels reestablished
 Master broadcasted resource hash value bitmaps
 Non-local Process blocks cleaned out
 Resources and enqueues cleaned out
 Resources remastered 6453
 72142 GCS shadows traversed, 2 cancelled, 7904 closed
 49280 GCS resources traversed, 2 cancelled
 33506 GCS resources on freelist, 77706 on array, 77706 allocated
 set master node info
 Submitted all remote-enqueue requests
 Update rdomain variables
 Dwn-cvts replayed, VALBLKs dubious
 All grantable enqueues granted
 72143 GCS shadows traversed, 6338 replayed, 7906 unopened
 Submitted all GCS remote-cache requests
Thu Mar 25 16:46:10 2004
Completed redo application
Thu Mar 25 16:46:10 2004
 0 write requests issued in 44200 GCS resources
 0 PIs marked suspect, 0 flush PI msgs
/16:45:07                                                   2409,12       17%
```

Figure 3-36 OCMS reconfiguration

There were nine healthy instances at 16:45:07, and 63 seconds later there were 10 instances. The 63 seconds spanned the time required to shut down the DSS instance that was executing on blade11 and then to start a PROD11 instance on that node. Meanwhile, OLTP continued executing with 200 sessions per node on the nine remaining original PROD instances.

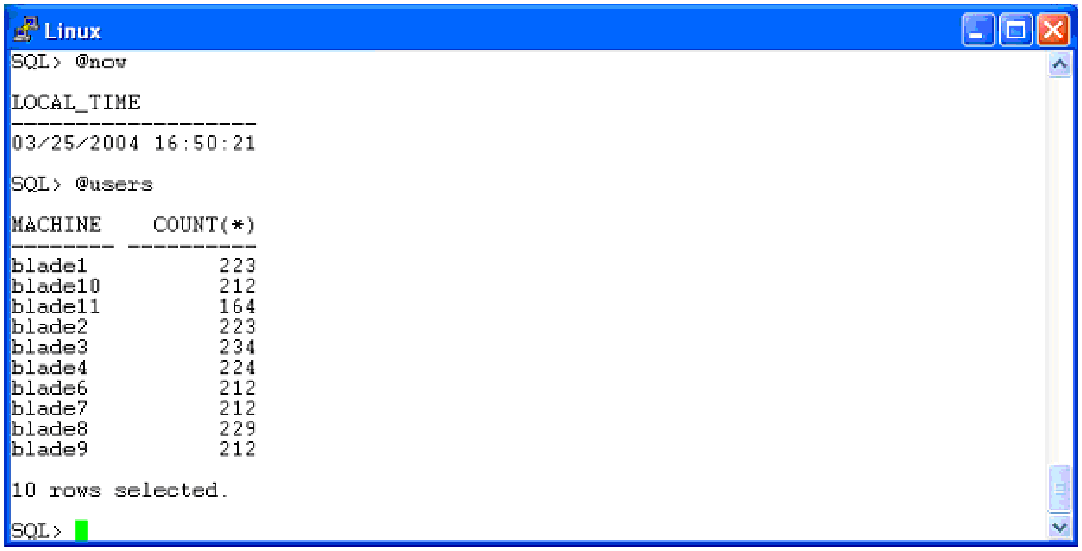
Figure 3-37shows that throughout the recovery period, 100% of the user count on nodes 1-4 and 6-10 remained connected. Figure 3-37 also shows that 62 of the users from blade5 were failed over through Net Services to nodes 2, 3, 4, and 8. These users were executing applications enabled for Transparent Application Failover.



```
SQL> @now

LOCAL_TIME
-------------------
03/25/2004 16:49:21

SQL> @users

MACHINE    COUNT(*)
--------- ----------
blade1          223
blade10         212
blade11          12
blade2          223
blade3          234
blade4          224
blade6          212
blade7          212
blade8          229
blade9          212

10 rows selected.

SQL>
```

Figure 3-37 User count during recovery

The remaining users connected to blade11 once it was online. Figure 3-38 shows that in the 1 minute that elapsed from the activity illustrated in Figure 3-36 on page 55 to that in Figure 3-37, 150 users had connected to PROD11. At that point, 100% of the original user community was back online.



```
SQL> @now

LOCAL_TIME
-------------------
03/25/2004 16:50:21

SQL> @users

MACHINE    COUNT(*)
--------- ----------
blade1          223
blade10         212
blade11         164
blade2          223
blade3          234
blade4          224
blade6          212
blade7          212
blade8          229
blade9          212

10 rows selected.

SQL>
```

Figure 3-38 Users connecting to PROD11

In summary, the impact to the users in this test was a reduction in server bandwidth of 10% for only 71 seconds from 16:44:59 when blade5 went offline to 16:46:10 when blade11 was ready to accept connections. There was no total outage, no lengthy service brown-outs, and no emergency action taken by the administrative and operational staff. This establishes the flexible database cluster architecture as a natural fit for today's demanding IT requirements.

3.3.8 Scalability in an OLTP environment

This section examines the results of the scalability measurements made from the proof of concept that used the BladeCenter platform.

The OLTP workload described in "OLTP workload" on page 50 was used to test the combination of BladeCenter, PolyServe Matrix Server, and Oracle9*i* Real Application Clusters. The test method consisted of connecting 100 users that were executing the Pro*C benchmark code and connected by SQL*Net to dedicated servers.

Scalability on demand was the focus during all flexible database cluster testing. Data points collected at each node count were not preceded by a global reboot. Instead, once a test execution completed, additional servers were started to satisfy the next test node count requirement. That is, there were no database shutdowns after the test began with 100 users on one node.

After that test completed, an instance of the PROD (OLTP) database was started, and the 2-node test commenced, and so on. This method clearly supports the position that the flexible database cluster architecture is a truly flexible pool of server resources that can be added to a workload without interruption. SQL*Net is set up to support connectivity to a PROD instance on any node in the BladeCenter. Therefore, all that has to transpire to support such connections is the start of an instance, a seamless dynamic utilization of the server pool.

In Figure 3-39 on page 58 and Figure 3-40 on page 59, each data point represents an average taken from three executions of the benchmark for 30 minutes. A full test suite from one to 14 nodes represented 12 hours of solid OLTP processing.

Generally, scalability is the first data point to examine in a benchmark scenario. The flexible database cluster proof of concept was more than a benchmark, however. Scalability for Oracle9*i* Real Application Clusters on PolyServe Matrix Server and BladeCenter was very good. Scalability was limited under this workload, as it turns out, by not having enough disk subsystem bandwidth to satisfy the tremendous bandwidth available in the BladeCenter nodes.

Older system architectures were much more prone to performance bottlenecks than the combined technology of BladeCenter and Oracle9*i* Real Application Clusters on PolyServe Matrix Server. At one time, running this OLTP workload on the same number of disk drives would have certainly driven 28 processors to saturation. However, the 28 CPUs in the BladeCenter that were tested never peaked above 82% utilization.

When Oracle9*i* Real Application Clusters are deployed on BladeCenter, it seems the challenge for building balanced, scalable systems running this workload may not be a processor or memory-level concern, but may instead be a storage subsystem issue. That is, this testing has established that Oracle9*i* Real Application Clusters and BladeCenter with PolyServe Matrix Server can scale as high as the underlying disk subsystem can handle.

BladeCenter was attached to a SAN configured with 110 disk drives. The database was placed evenly across 106 of those disks with a Stripe and Mirror Everything (SAME) methodology. Modern drives such as those used for the OLTP database should be expected to service roughly 100 random 4K transfers with acceptable latency. That latency, however, is not static. As the drives approached critical service levels, the latencies seen during this testing approached 40ms per transfer as should be expected.

As seen in Figure 3-39, Oracle9*i* Real Application Clusters was able to drive the physical 4K random I/O rate up to steady states of roughly 12,600 transfers per second. Peaks of nearly 14,000 per second did occur during checkpoints and other burst-related activity.

This is a tremendous amount of random OLTP disk transfers. These I/O rates are not from a simple test program. For Oracle to demand this much I/O, there must be an incredible amount of processor-intensive and memory-intensive activity in the System Global Area (SGA).

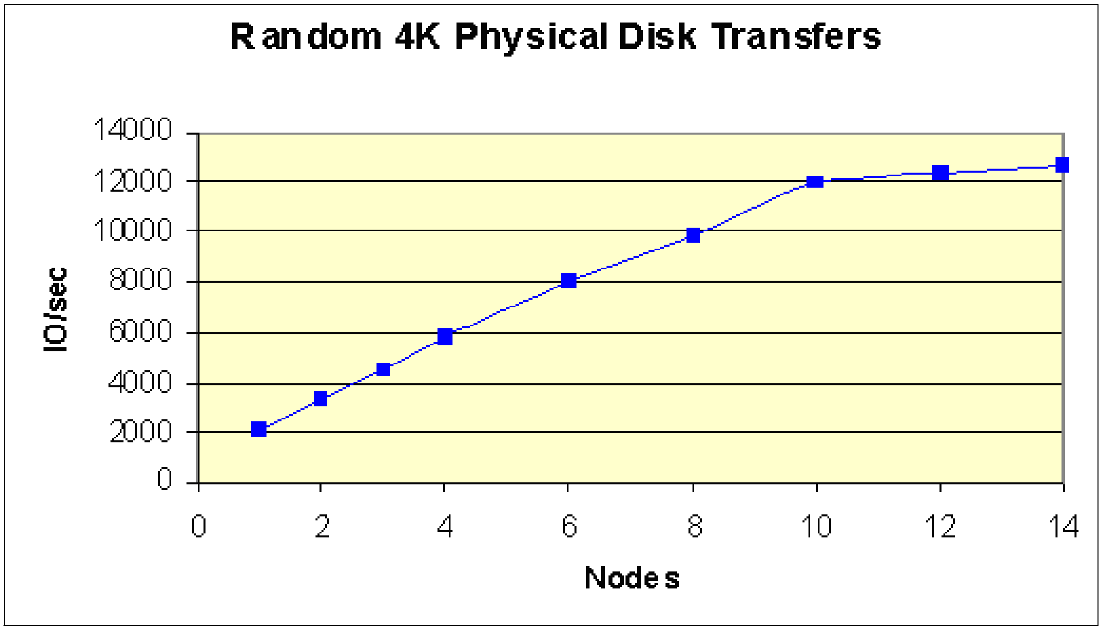


Figure 3-39 Physical disk transfers

For example, Table 3-1 on page 51 shows that for every physical transfer, there are also 5.7 logical reads of buffers in the SGA. Every logical read has associated Oracle internal overhead such as acquiring or freeing latches (for example, cache buffers chains latches and cache buffers least recently used, or lru, latches). The peak-sustained I/O fallout of 12,600 transfers per second indicates how stressed Oracle instances were on each node.

Figure 3-39 also shows that the OLTP workload drove the physical I/O rates from 2,073 per second at one node to 12,656 per second at 14 nodes. Analysis of I/O latency data revealed that the I/O service times remained acceptable through the 8-node test, but beyond that point, the drives were saturated.

Since the sustained I/O rate at eight nodes was 9,829 transfers per second, or 93 per drive, the resulting increase in transfer times was expected. The key point, however, was that the amount of hardware available to allocate to this test held scalability to the level achieved.

Oracle9*i* Real Application Clusters on PolyServe Matrix Server suffered no scalability problems. As was demonstrated by the peak clusterwide CPU utilization of 82%, the BladeCenter certainly had more bandwidth.

The throughput graph in Figure 3-40 clearly indicates that scalability was directly related to I/O throughput. This is good news. Unlike older OLTP system architectures, where the bottleneck was usually at the system level, resolving this performance issue was simple. With the IBM TotalStorage DS4400 SAN architecture, you simply add disks to the array.

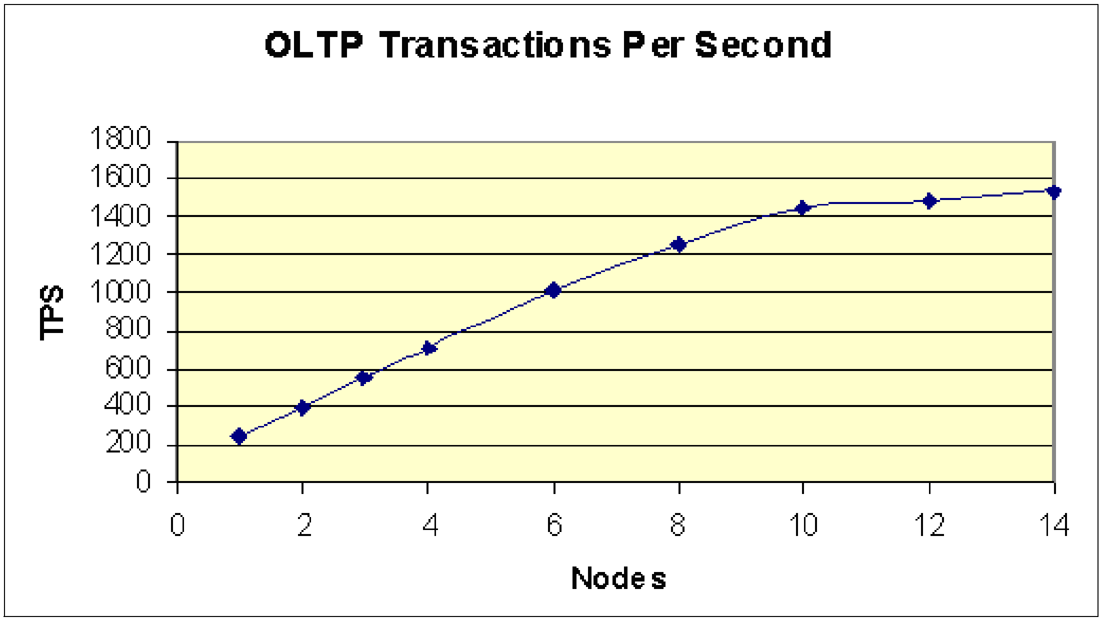


Figure 3-40 Scalability and I/O throughput

3.3.9 Scale on demand, dynamic and transparent

This section discusses the facets of scalability.

DSS

The flexible database cluster architecture has the advantage of being able to use the Oracle PQO for DSS workloads. PQO can decompose DSS-style queries and large data maintenance functions into work units executed by some or all of the CPUs in a cluster.

The DSS-style testing performed on the flexible database cluster also proved the scale on demand theme of Oracle9*i* Real Application Clusters. The testing essentially consisted of adding nodes to increase the speed of a DSS query or large administrative task such as index creation. The DSS database was not shut down during the testing. Instead, as was the case in the OLTP testing, additional servers were allocated and the test was executed again. The DSS database was designed to run on the top eight nodes of the cluster while OLTP was running on nodes 1 through 6, which is not a very high load.

Three main tests were executed at the various node count levels:

- DSS test 1

  This query tested the ability of the flexible database cluster architecture to scan and sort a large amount of data. The query consisted of a **select count(distinct(cardnum))** command from the card table described in 3.3.5, “Overview of databases” on page 46. The table had roughly 1,600,000,000 rows, which required a great deal of sorting and merging to eliminate duplicates. The sorting could not be performed entirely in memory.

- DSS test 2

  After testing a distinct select count from the card table, an index was created on its cardnum column. The index was a simple, non-sorted index that allowed duplicates.

- DSS test 3

  This index creation test was by far the most data-intensive. It created a three-way unique composite index on the item table described in “DSS database: decision support” on page 50. The table had roughly 1,700,000,000 rows that could not be sorted in memory. This index creation consisted of reading rows and writing to both the sort segments and the target index.

Once again, the key point about this testing was that at each step, adding nodes to increase the speed of the workload was a non-intrusive effort. We simply started an instance of the DSS database on other blades, ran the task again, and completion times improved.

Figure 3-41 on page 61 shows query completion times for DSS test 1. When executed on a single node, the query completion time was 101 minutes. Without interruption, another Oracle instance was added and the completion time improved with 100% linear scalability. Further proving the scalability on demand feature of the flexible database cluster architecture, Oracle instances were booted and the test was run again in succession for 4-node, 6-node, and 8-node counts. Scalability to eight nodes was 81%. The 8-node completion time for the test was 15.9 minutes. To put this in perspective, returning this query in 15.9 minutes means Oracle9*i* scanned and sorted roughly 18,000 rows per CPU per second.

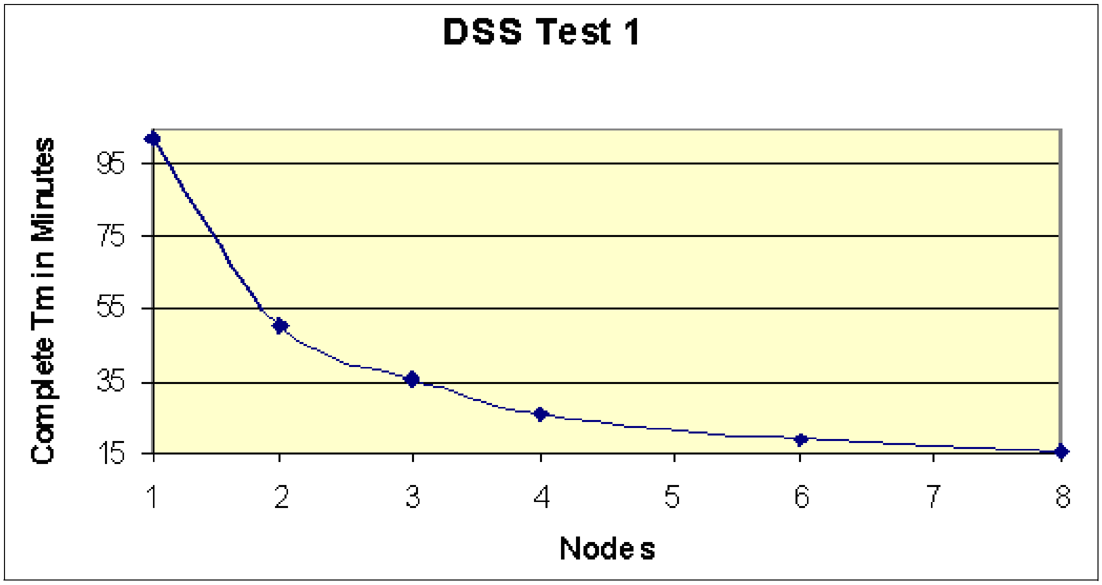


Figure 3-41 Completion times for DSS test 1

After executing DSS test 1, the Oracle instances on the second through eighth nodes were shut down in preparation for DSS test 2. This test created a non-unique index on the credit card table with approximately 1,700,000,000 rows as described in 3.3.5, "Overview of databases" on page 46. Since the index was not sorted or unique, it offered a different profile than DSS test 1.

DSS test 2 and DSS test 3 both prove that Oracle9*i* Real Application Clusters and PolyServe Matrix Server take full advantage of all available disk subsystem bandwidth and do not exhibit scalability limits at the software level.

The most interesting finding from DSS test 2 and DSS test 3 was that they exhibited precisely the same scalability attributes, even though the work being done was dramatically different. The index created in DSS test 2 was a simple, single-column, non-unique index on 1,700,000,000 rows, while DSS test 3 created a unique 3-way composite index on 1,600,000,000 rows. If there were a server-level bottleneck (for example, CPU bandwidth, bus arbitration, memory controller, or I/O adapter), there would have been substantial performance variance between these two large index creations. After all, DSS test 3 is significantly more processor intensive and memory intensive by virtue of what it is creating.

The design of the flexible database cluster architecture naturally provides increased server bandwidth. Each two-CPU blade has its own bus, memory controller, and I/O adapter, all of which reduce the chances of encountering a traditional bus or memory controller bottleneck. These are very balanced servers. This architecture will fully utilize all disk I/O subsystem bandwidth given to it. This architecture is optimized for growth. If you want more server bandwidth, simply add a server. If more disk subsystem bandwidth is needed, simply add it to the SAN.

Flexible database cluster architecture enables the simple addition of BladeCenter nodes to these index-create tasks as shown in Figure 3-42 on page 62. Each data point was collected without interruption. Additional instances of Oracle were simply started on the additional nodes and the index-create task was run again.

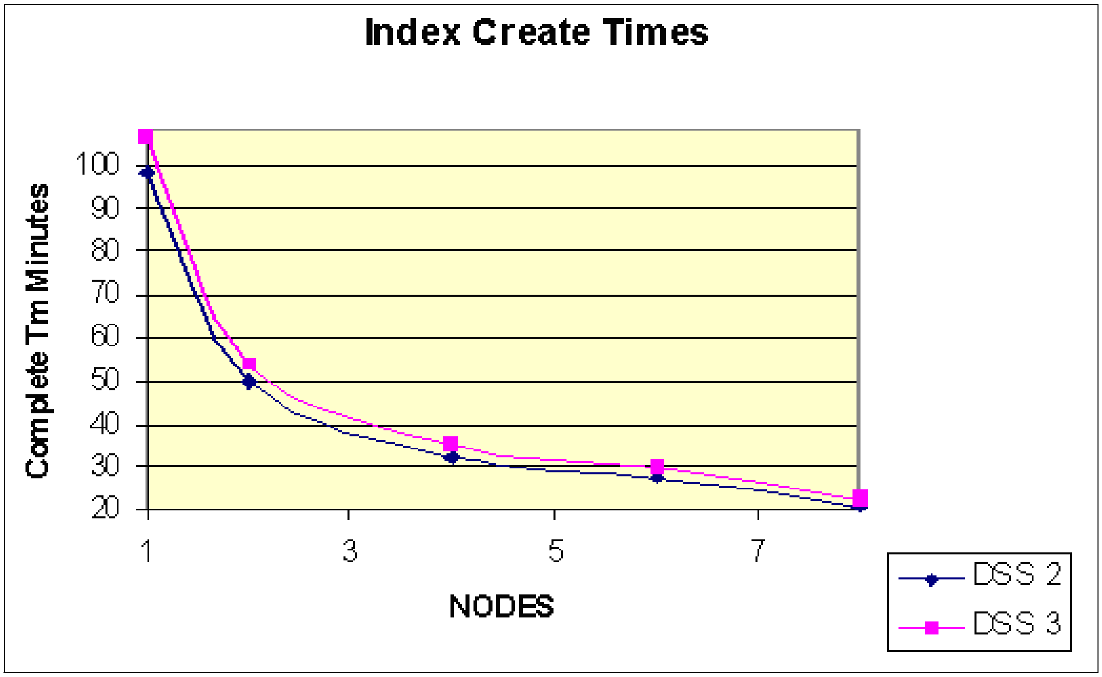


Figure 3-42 Index-create times

The scalability trend exhibited by both queries is summarized in Table 3-2.

Table 3-2 Scalability trend

1 to 2 nodes	2 to 4 nodes	4 to 8 nodes
99% scalability	77% scalability	77% scalability

Analysis of the disk I/O performed by DSS test 3 was conducted using the io.sql script provided in Appendix A, “Implementation overview” on page 77. The physical disk transfer rate from the PQO processes scaled 100% from one to two nodes, 70% from two to four nodes, and 74% from four to eight nodes as shown in Figure 3-43 on page 63.

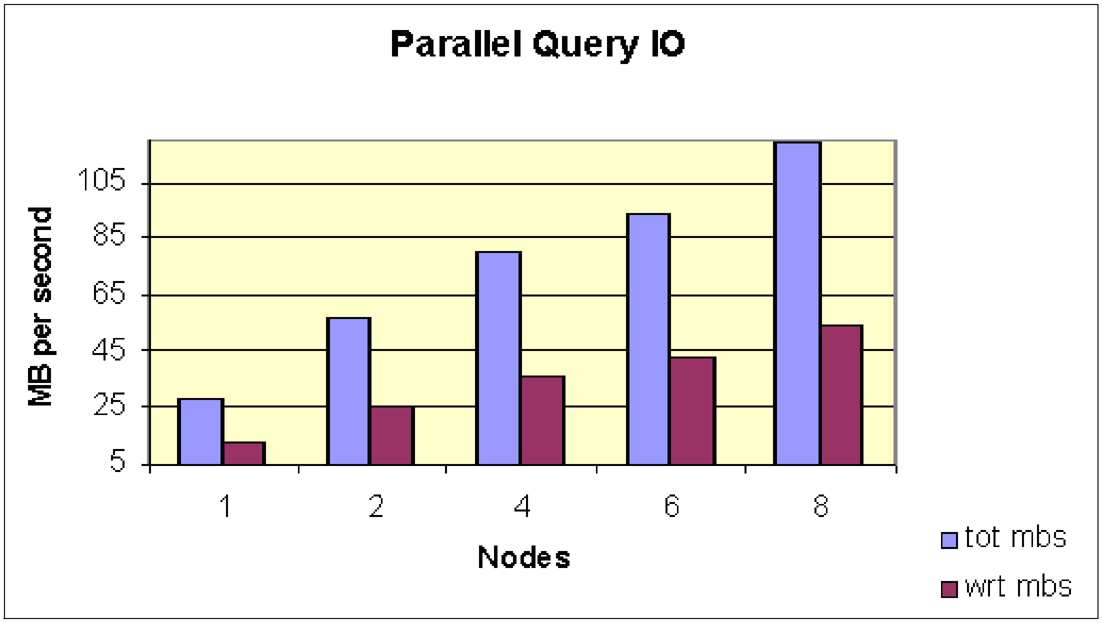


Figure 3-43 Parallel Query IO

Disk I/O

One important aspect of this I/O profile is missing from the data provided in the Oracle GV$ performance virtual tables. Since the storage was striped and mirrored, there were substantially more disk transfers at the array level. At eight nodes, the array-level I/O ratio was actually 37% reads and 63% writes. This is attributable to the fact that DSS test 3 generated 54MB per second of data written to a mixture of sort segments and index blocks.

The GV$ performance virtual tables do not present the 100% overhead associated with the mirror write. The actual array-level write transfer rate was 108 MB per second and the total array-level I/O rate when processing on eight nodes was 118 MB per second as shown in Figure 3-44.

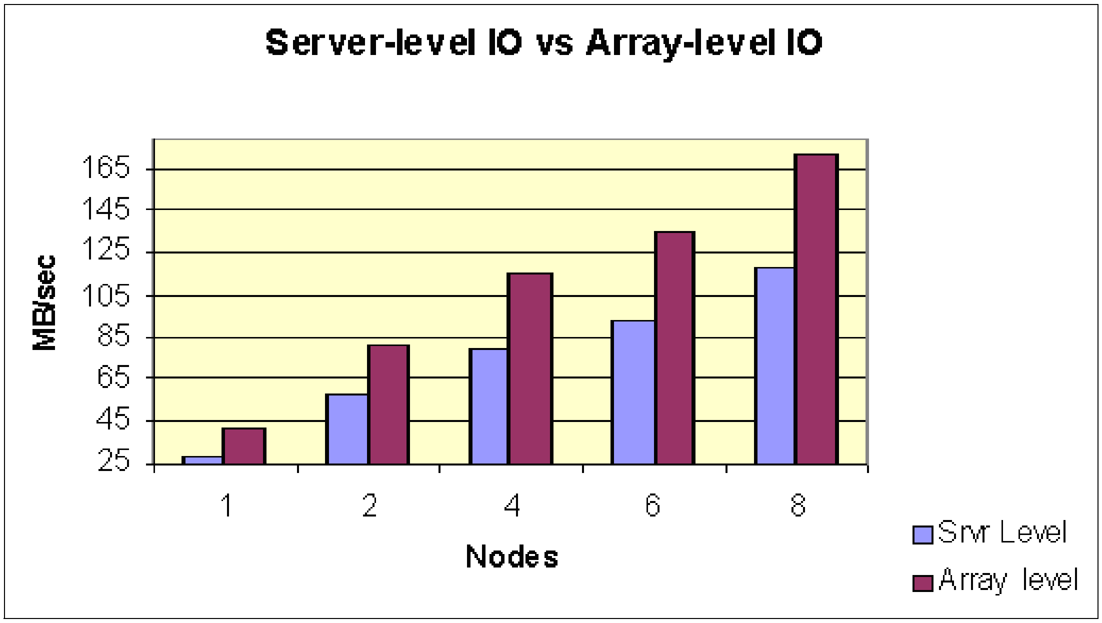


Figure 3-44 Server-level I/O versus array-level I/O

3.3.10 Scripts

The scripts used in the proof of concept were:

- users.sql

```
Column machine format a8
select machine,count(*) from gv$session
group by machine ;
```

- instance_status.sql

```
column  host_name format a10
select host_name,thread#,database_status from gv$instance
order by thread# ;
```

- now.sql

```
select to_char(sysdate,'MM/DD/YYYY HH24:MI:SS') "LOCAL_TIME" from dual
```

- io.sql

```
select sum(PHYRDS) reads,sum(PHYBLKRD * 4 )/1024 readMB,
      sum(PHYWRTS) writes,sum(PHYBLKWRT * 4 )/1024 writeMB
from dba_data_files,gv$filestat
where dba_data_files.file_id = gv$filestat.file#;
REM exit;
```

3.3.11 Summary

The flexible database cluster architecture, with BladeCenter and PolyServe Matrix Server, clearly enhances the value of Oracle9*i* Real Application Clusters, providing:

- 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 in some cases, 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 fast or even 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

3.4 Flexible database clusters with xSeries

IBM and PolyServe joined forces to build a 16-node cluster running SLES and attached it to a formidable SAN configured with 206 physical disk drives. Oracle9*i* Real Application Clusters was then installed on a PolyServe Matrix Server cluster file system.

The flexible database cluster cluster was the target of a series of tests that examined, in depth, the running and management of not only a single application, but also 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, without 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

3.4.1 System overview

The following cluster system components were used in the flexible database cluster test system:

- Client nodes: Five xSeries 330 1U rack-optimized servers
- Server nodes: 16 xSeries x345 servers
- Storage: Two IBM TotalStorage DS4400 Storage Servers and 15 TotalStorage EXP700 Storage Expansion Units with 206 36.4GB HDDs, providing over 7 TB of storage space
- LAN switch: Cisco Catalyst 6509 switch
- SAN switch: Two IBM TotalStorage SAN Switch F16s to interconnect the cluster nodes and the SAN

The architecture of xSeries x345 servers contributed to the high availability of the flexible database cluster environment and, when combined with the Multipath I/O feature provided by PolyServe Matrix Server, made it possible to build a SAN subsystem free of single points of failure.

All public LAN traffic between clients and cluster nodes as well as Oracle interconnect traffic was exchanged through the Cisco Catalyst 6509 switch. Additional traffic on the switch included Ethernet access to the IBM TotalStorage DS4400 Storage Server for configuration and management of the storage subsystem.

3.4.2 Database overview

To test the flexible database cluster architecture, three databases, OLTP, DSS, and DEV, were created in the PolyServe 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 of them. The goal was to have a realistic mix of processing while testing the manageability of the flexible database cluster architecture.

OLTP database (PROD)

The OLTP database schema was based on an order entry system and contained tables for customers, orders, line Items, historical line items, products, and the warehouse application. 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 data marts.

The DSS workload was set up as a stream of queries in a queue serviced in serial order by the 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. For example, an increase from 2 to 4 nodes caused a speedup 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 of 2 KB. 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 by 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.

3.4.3 Test results

Results and analysis are presented for the stress test, the fault-resilience test, and the on-demand scalability test.

Stress test

A test suite was set up with 16 instances of the PROD database. User numbers of 100, 250, and then 500 were connected per node, each executing the OLTP workload. Figure 3-45 on page 67 shows that the cluster and Oracle9*i* Real Application Clusters 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 to 47,160 transactions per minute, or 98% of linear scalability.

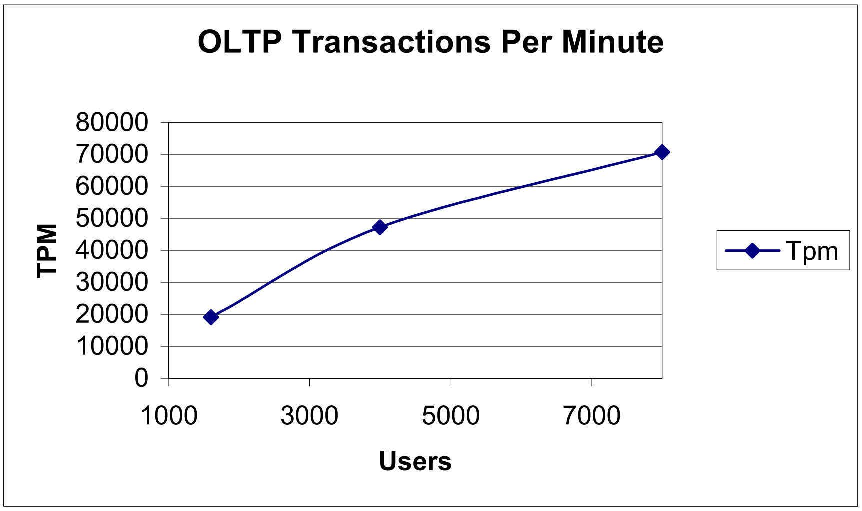


Figure 3-45 Results of stress testing

Fault resilience test

The flexible database cluster architecture with PolyServe Matrix Server is the only approach that can fully harness the ability of Oracle9*i* Real Application Clusters 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* Real Application Clusters 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. Figure 3-46 is a high-level view of what transpired during the test.

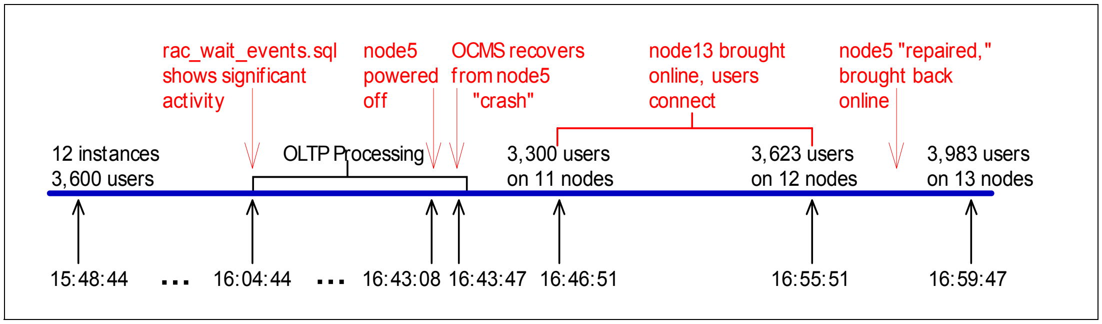


Figure 3-46 Timeline for stress test

On-demand scalability test

This test measured the ability of the flexible database cluster to quickly add nodes to a running application. The application was a queue of DSS queries being serviced by the PQO. The test results show that three times the server resources were added in a 13-minute period, without 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 more quickly.
- The PQO takes advantage of the bandwidth on the next query from the queue.

The addition of resources was completely dynamic. Figure 3-47 illustrates what transpired during this test.

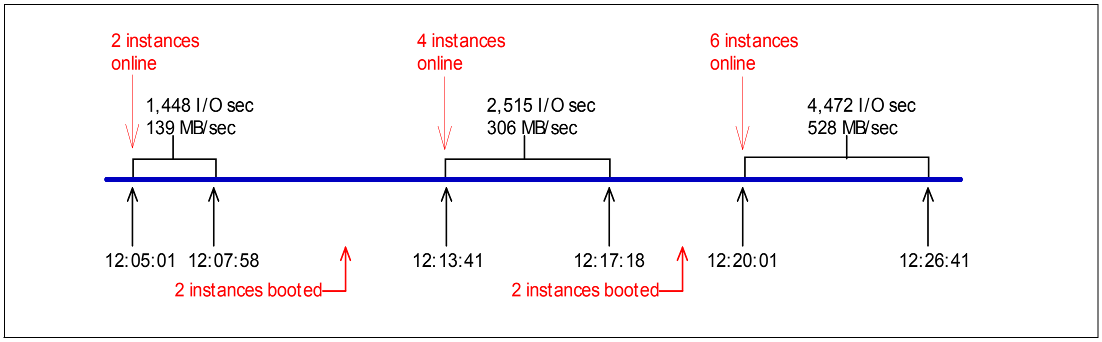


Figure 3-47 Timeline for on-demand scalability test

4

Proven benefits for Oracle9*i* Real Application Clusters

This chapter presents two case studies that demonstrate the benefits of flexible database clusters implemented on xSeries and BladeCenterBladeCenter severs and running Oracle9i and PolyServe Matrix Cluster File System.

4.1 The Benefits

Clusters based on IBM hardware and PolyServe Matrix Server provide a proven solution for Oracle9*i* Real Application Clusters deployments. Numerous client installations have experienced the following benefits:

- Easier Oracle administration and management of multiple Oracle workloads

  PolyServe Matrix Server supports a single shared Oracle Home. Administrators can configure and run an Oracle9*i* Real Application Clusters cluster from a single, unified application image. Multiple Oracle servers and workloads can be consolidated into one cluster.

- Improved performance

  PolyServe provides an implementation of Oracle Disk Manager (ODM) that includes cluster-wide input/output (I/O) monitoring at the cluster, database instance, or node level. It also supports Oracle parallel extract, transform, load (ETL), which greatly reduces data import time. With IBM/PolyServe, the performance of client workloads is improved and smaller Oracle9*i* Real Application Clusters can support larger user populations.

- Improved scalability and capacity on demand

  Server and storage capacity can be added or repurposed as necessary, making the cluster a dynamic resource that can respond to changing workloads immediately.

- System-wide high availability

  IBM/PolyServe clusters provide integrated high availability for applications, middleware, servers, fabric, and storage.

- Reduced total cost of ownership (TCO)

  Deploying Oracle9*i* Real Application Clusters on an IBM and PolyServe cluster dramatically lowers capital and operating costs for the Oracle IT infrastructure. The increased operational efficiencies also result in higher utilization of human and capital resources. Customers have reported a total three-year TCO savings of over 50% by migrating to IBM/PolyServe clusters for their Oracle deployments.

Two case studies that demonstrate the benefits of deploying Oracle9*i* Real Application Clusters on IBM/PolyServe cluster are:

- Burlington Coat Factory: This case study describes how Burlington Coat migrated from UNIX clusters to highly scalable, available IBM/PolyServe clusters for its data center.
- Tractor Supply Company: This case study describes how an IBM/PolyServe cluster running Oracle9*i* Real Application Clusters drastically reduced cost and increased system uptime for Tractor Supply Company.

4.2 Case study: Burlington Coat Factory

Burlington Coat Factory Warehouse Corporation (Burlington Coat Factory) has always been progressive when it comes to the technology that runs its business. An early advocate of open systems, the company has in recent years embraced Linux in its enterprise systems. *Retail Info Systems News* has, for example, acknowledged the company for fusing technology with strategy.

Burlington Coat Factory needed to migrate Oracle databases from their legacy IBM/Sequent NUMA-Q® UNIX servers that were at the end-of-life stage. Burlington Coat Factory evaluated using UNIX servers versus a cluster of xSeries Linux servers with PolyServe Matrix Server software. Burlington Coat Factory chose the IBM/Linux/PolyServe platform because of substantially lower TCO with better reliability and manageability.

It is not surprising, then, that the company is in the process of deploying its next generation IT infrastructure and that the infrastructure is based on open standards and cost-effective processors. Burlington Coat Factory is using PolyServe Matrix Server software to build clusters of xSeries Intel-based servers. These servers will share both storage and application processing resources, creating flexible clusters to run mission critical IT applications.

4.2.1 Decrease infrastructure cost

Burlington Coat Factory is a leading outerwear and apparel retailer that offers name-brand apparel and accessories at prices up to 60% lower than other department stores. This business model, spread across approximately 300 stores in 42 states, requires an efficient, reliable, and highly available IT infrastructure.

The company faces the same problems faced by successful companies everywhere: how will the IT infrastructure handle growing application demands? How can IT continue to manage critical applications in a cost-effective manner? For Burlington Coat Factory, IT efficiency and reliability are business imperatives.

Burlington Coat Factory runs its business on 40 Oracle databases addressing functional areas of the business. Until recently, these business applications have run on a small number of high-end IBM/Sequent NUMA-Q UNIX servers. Because the NUMA-Q product line was reaching the end-of-life stage, it was clear that Burlington Coat Factory would need to replace these servers with other high-end UNIX servers, or adopt a new infrastructure using Intel-architecture servers.

Burlington Coat Factory considered the following factors in their evaluation:

- Ease of software migration
- TCO
- Ease of management
- Scalability
- Performance
- Reliability

Based on these factors, the company decided to adopt a new infrastructure based on xSeries Intel-architecture servers running Linux (see Figure 4-1 on page 72). Because the Burlington Coat Factory IT team had years of experience with Intel-architecture UNIX on the Sequent NUMA-Q series, the choice of Linux on xSeries was a natural one. Also, the operating system is inexpensive and therefore, the risks were manageable.

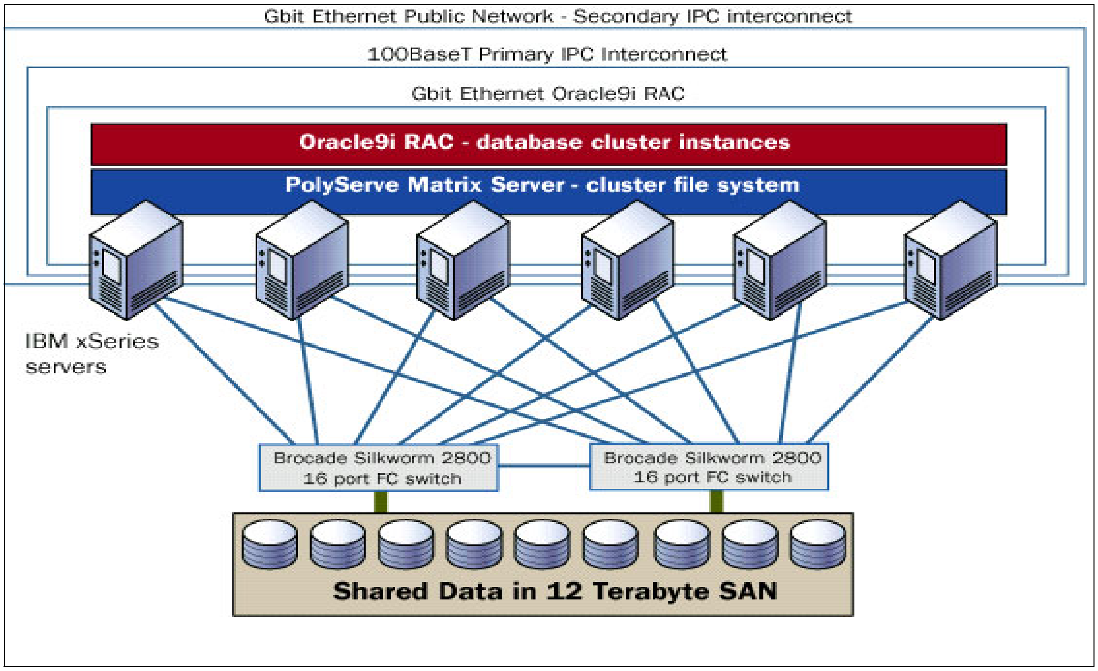

Gbit Ethernet Public Network - Secondary IPC interconnect
100BaseT Primary IPC Interconnect
Gbit Ethernet Oracle9i RAC
Oracle9i RAC - database cluster instances
PolyServe Matrix Server - cluster file system
IBM xSeries servers
Brocade Silkworm 2800 16 port FC switch
Brocade Silkworm 2800 16 port FC switch
Shared Data in 12 Terabyte SAN

Figure 4-1 The implementation at Burlington Coat Factory

> "PolyServe software is a crucial tool in our migration of core data center applications from our legacy UNIX platform to clusters of xSeries. The PolyServe team has proven its ability to deliver enterprise-class system software products as well as the support that's required for mission-critical software in our data center. The IBM team has also been unbelievable in working with us on this project - they've stood on their heads to do things for us." —Mike Prince, Chief Information Officer, Burlington Coat Factory

Linux on Intel-architecture servers offered lower total cost of ownership over proprietary UNIX platforms. The lack of high licensing fees, easier remote administration, and the stability of Linux were also factors.

4.2.2 Using PolyServe Matrix Server to deploy and manage Linux clusters

Burlington Coat Factory is deploying flexible clusters of xSeries servers running Linux and PolyServe Matrix Server cluster file system software. Initially, Burlington Coat Factory is deploying Oracle9*i* Real Application Clusters, but also plans to deploy other applications and services such as Web serving, file serving, and lightweight directory access protocol (LDAP) on the clusters. PolyServe Matrix Server simplifies the deployment and management of applications and middleware. For example, with PolyServe Matrix Server, Oracle9*i* Real Application Clusters can be installed and managed in a single location in the storage area network (SAN). PolyServe Matrix Server also enhances the performance of Oracle9*i* Real Application Clusters in a file system by providing interfaces to Oracle Disk Manager (ODM), the optimized I/O Oracle provides for Oracle9*i*.

All servers in the farm are attached to a SAN. PolyServe Matrix Server software simplifies the deployment and management of the cluster by providing shared access to data stored on the SAN for all servers in the cluster.

By pooling the servers as a common set of resources, this architecture provides better server utilization than using large symmetric multiprocessing (SMP) servers dedicated to specific applications. All active servers in the cluster can, for example, share a common, small pool of standby servers. With appropriate clustering software, such as the built-in high-availability capability of PolyServe Matrix Server, the larger set of standard servers is also inherently more flexible and reliable than a smaller number of proprietary UNIX SMP servers. The failure of a single server in this architecture is a far less significant event than it would be with larger, proprietary UNIX SMP servers.

Using PolyServe Matrix Server, Burlington Coat Factory can easily add servers and storage to their cluster. The fully symmetric architecture of PolyServe Matrix Server software means that the cluster can scale to large numbers of servers. Scalability is critical as Burlington Coat Factory extends its initial deployment to address its entire enterprise IT infrastructure.

With 40 Oracle instances, manageability is a big concern. Bill Ubelacker, Director of System Services for Burlington Coat Factory, said, “We won't even consider running the databases on raw partitions, so file system support was essential. With a shared file system, there is only one Oracle home directory and one location for upgrades. PolyServe makes the whole cluster function as one coherent whole.”

The platform is also inherently highly available. If a server fails, the shared file system and highly available features of PolyServe Matrix Server enable another server to quickly and seamlessly take its place. Such availability is very important to Burlington Coat Factory, particularly during the peak loads associated with the holiday buying season.

The manageability benefits will increase as the company moves more applications onto the flexible cluster architecture. Instead of using dedicated servers for databases, Web servers, and application servers, the company will be able to run them all on clusters of four-way xSeries Intel-architecture servers that can be reprovisioned as dictated by load or other factors. This architecture will make administration of the Burlington Coat Factory data center significantly simpler and less costly.

4.2.3 Next steps for Burlington Coat Factory

Burlington Coat Factory started testing early in 2002 with a simple four-node cluster with PolyServe Matrix Server running Web servers, proxy servers, and a DNS server. They simulated load and conducted testing functionality and reliability testing. “We couldn't break it,” said Ubelacker.

As the company expands its infrastructure, Burlington Coat Factory is assembling and testing all components of the cluster architecture: hardware, database software, server-to-server interconnects, SAN, PolyServe Matrix Server cluster file system, and high availability software. The company is gradually phasing in its new architecture, moving applications from the existing Sequent servers to clusters of xSeries servers.

Burlington Coat Factory is moving with characteristic speed from staged pre-production testing to production operation. Ubelacker says, “We plan to have Oracle9*i*-based applications in production in early 2003 on this infrastructure.” The company has never been one to lag behind if there is a competitive advantage to be gained with new technology.

4.3 Case study: Tractor Supply Company

Tractor Supply Company (Tractor Supply) is the largest retail farm and ranch store chain in the U.S. The company operates more than 430 retail stores in 30 states, employs more than 6,200 team members, and is headquartered in Nashville, Tennessee. Tractor Supply is continuing to grow, opening new stores and improving product offerings while celebrating its 65th anniversary.

The company was founded in 1938 as a mail-order catalog business offering tractor parts to America's family farmers. Today Tractor Supply is a leading-edge retailer with 2002 revenues surpassing $1,000,000,000 for the first time. Last year the company opened 113 new stores after purchasing many of the assets of its former largest competitor.

4.3.1 The challenge

Tractor Supply wanted to install a cost-effective performance infrastructure to host its mission-critical systems to meet the demands of the company's growth. As a test case for the viability of this new architecture, Tractor Supply chose to migrate its mission-critical Oracle data warehouse that collects point of sale (POS) transactions from all its stores nationwide. This system is absolutely critical to running Tractor Supply's business and it must be highly available in order for Tractor Supply to make critical business decisions.

The Tractor Supply data warehouse not only requires a substantial amount of processing power to tabulate the transaction data, but it must also be completely fault tolerant and highly available. Originally, Tractor Supply had considered a two-node failover cluster of eight-processor Intel SMP servers running Oracle8**i** on Windows and a leading clustering product to handle the workload. However, Tractor Supply decided that not only was this approach expensive, but it would also create unacceptable administration costs.

As part of its evaluation, Tractor Supply had explicit and demanding requirements for any new software and hardware platform that it would select for its data warehouse. TSC evaluated each platform solution on the following criteria:

- Reduction in total cost of ownership (TCO) through improved manageability, or operating expenditure (OPEX)
- Reduction in capital expenditure (CAPEX) through the use of industry standard components
- Flexible and easy expansion through horizontal scaling
- System-wide elimination of single points of failure
- High levels of I/O throughput
- Elimination of the costly cycle of forklift upgrades to new, bigger servers

Tractor Supply was very concerned about the usability and ease of management of any cluster configuration. In particular, Tractor Supply evaluated the ease of installation and configuration and the ability to manage the cluster as a single large server.

4.3.2 The solution

Rather than deploying expensive UNIX servers or a Windows failover cluster, Tractor Supply chose to deploy a cluster of Linux servers, whereby each server in the cluster can simultaneously read and write to volumes of shared data on a SAN.

Tractor supply reached this conclusion after conducting a rigorous performance benchmark. To confirm that this architecture would meet its performance needs, TSC set up a three-node Linux cluster of dual-processor Intel servers running Oracle9*i* Real Application Clusters (RAC) and PolyServe Matrix Server. They conducted controlled load testing on the three-node cluster that simulated key resource-intensive processes of their data warehousing application. In addition to costing substantially less than alternative approaches, TSC concluded that the PolyServe/Oracle9*i* Real Application Clusters cluster would exceed the current performance demands of the POS data warehouse and provided a future growth path as demand on the system increased.

Tractor Supply also evaluated several high availability and cluster file system products to provide the resilient, fault tolerant infrastructure for their data center. They needed to find a cluster file system that would allow them to manage their industry standard servers as a unified system. Three cluster file system products purporting to support Oracle9*i* Real Application Clusters and general-purpose file-based applications were subjected to a battery of ease of use, performance, scalability, and high availability tests. The company concluded that one product contained a single-point of failure and was cumbersome to configure and manage. Therefore, it was eliminated.

The second cluster file system was evaluated as part of Tractor Supply's Oracle9*i* Real Application Clusters testing. Tractor Supply concluded that it could not deploy the product across its data center on both Oracle and non-Oracle platforms. Thus, it was discarded because it was not a general-purpose solution.

PolyServe Matrix Server was the final cluster file system tested. Tractor Supply was attracted to PolyServe Matrix Server because of its strong integration with Oracle databases. Stevan Townsend of Tractor Supply said, "PolyServe demonstrated that it had the capabilities and Oracle expertise to get the job done. We were impressed with PolyServe's implementation of the Oracle Disk Manager (ODM) that ensured that PolyServe Matrix Server and Oracle9*i* Real Application Clusters were tightly coupled and worked seamlessly together."

> "We put PolyServe through an extensive set of tests and benchmarks and compared it to other alternatives. Each time, PolyServe stood out from the crowd. It was the best product out there."—Stevan Townsend, Project Manager and Lead Engineer

In addition, PolyServe Matrix Server installed easily, possessed an intuitive graphical user interface for centralized cluster configuration, and, most importantly, offered seamless, highly scalable file system performance using standard Linux system calls. The fully symmetric architecture also eliminated single points of failure and provided high availability functionality and multi-path I/O capabilities. When Tractor Supply outgrows its current configuration, additional servers and storage can be added to the cluster online without affecting existing members of the cluster.

As a consequence of its exhaustive and grueling testing, Tractor Supply has selected the following solution for their mission-critical data center architecture:

- Oracle9*i* Real Application Clusters
- PolyServe Matrix Server
- SUSE LINUX Enterprise Server (SLES)
- BladeCenter
- EMC Fibre Channel Switches
- EMC Storage

4.3.3 The results

Tractor Supply migrated its production POS data warehouse to a PolyServe-powered cluster and went into production in December 2003. Tractor Supply estimates that using the PolyServe solution significantly reduced server costs compared to the alternative of purchasing a failover cluster of two expensive, eight-processor systems. Tractor Supply believes that the six-node cluster delivers higher levels of availability and is easier to manage than a two-node cluster of bigger SMP machines.

Using the PolyServe/Oracle9*i* Real Application Clusters solution, Tractor Supply cab scale out processing power to meet their increasing performance demands. Tractor Supply believes that it will save substantial money in the future because it will never again do a costly forklift upgrade of the servers that support the POS data warehouse.

Building on the successful migration of its POS data warehouse, Tractor Supply is planning to deploy larger configurations for future use cases. The next phase of the project includes multiple clusters of Oracle9*i* Real Application Clusters on servers used for the company's sales audit system, which stores all their POS data, SAP back-end databases, and test and development environments.

4.3.4 The configuration

The six-node Linux cluster shown in Figure 4-2 is comprised of a BladeCenter chassis with dual-processor BladeCenter HS20 blades. Each blade in the cluster can access data on the EMC switched Fibre Channel SAN. The Tractor Supply Oracle9*i* Real Application Clusters data warehouse is deployed on these six blades.

PolyServe Matrix Server runs on each server to provide a highly manageable, highly available environment, whereby all six blades can concurrently read and write the same data on the SAN. Tractor Supply selected SLES for its reliability and scalability. SUSE LINUX is the Linux distribution of choice for all six nodes in the cluster.

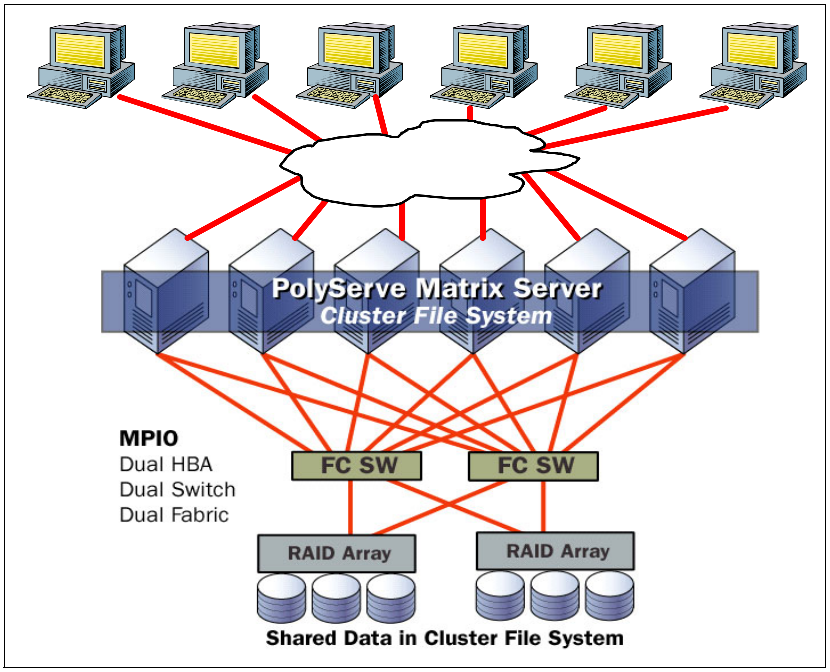


Figure 4-2 The configuration at Tractor Supply Company

A

Implementation overview

This appendix provides a high-level view of the steps for implementing a flexible database cluster on the platforms described in this paper.

New implementations

This section provides a high-level overview of the steps required to implement a flexible database cluster running Oracle9*i* Real Application Clusters.

Planning

During the planning phase, preparations should be made for the hardware and the necessary drivers, software, and firmware should be obtained. These are the steps that must be taken:

- Ensure that all hardware site requirements are met.
- Download drivers and firmware for the IBM platform and the storage area network (SAN) storage array.
- Obtain a supported version of the Linux operating system and kernel.
- Obtain the PolyServe Matrix Server software and documentation.
- Obtain the Oracle9*i* Real Application Clusters software and documentation.

Installing and configuring the cluster

The cluster implementation consists of three phases:

1. Install and configure the IBM hardware platform.
2. Install and configure the operating system, the kernel, and PolyServe Matrix Server.
3. Install Oracle9*i* Real Application Clusters and create a shared Oracle Home.

IBM hardware platform

Installation instructions for the IBM eServer xSeries servers and SAN can be obtained by searching for SAN on the following Web site:

http://www.redbooks.ibm.com

A good Redbook to start with would be the *IBM TotalStorage: Introducing the SAN File System*, SG24-7057-01 and can be obtained at:

http://www.redbooks.ibm.com/abstracts/sg247057.html?Open

Operating system and PolyServe Matrix Server

This phase of the implementation consists of the following steps:

1. Install the operating system according to the operating system documentation. Operating system files should not be installed on the SAN disks. PolyServe Matrix Server requires minimal changes to operating system files.
2. Build the kernel. The PolyServe Matrix Server distribution includes sample configuration files that are recommended for use with the Linux kernel and PolyServe Matrix Server.
3. Configure the SAN. The steps to take depend on the configuration and can include setting up the storage array, configuring FibreChannel switches, and creating logical unit numbers (LUNs) that PolyServe Matrix Server can use.
4. Install PolyServe Matrix Server on each server. PolyServe Matrix Server is provided in RPM format and is easily installed and configured. After the software is installed, a simple interface is used to perform the initial cluster configuration on a single server. That configuration can then be exported to the other servers in the cluster.
5. Add all servers to the cluster and start PolyServe Matrix Server.

At this point, the cluster is installed and running.

Oracle9*i* Real Application Clusters and shared Oracle Home

This phase of the implementation installs Oracle9*i* Real Application Clusters and sets up the shared Oracle Home. The key points are as follows:

- Oracle9*i* Real Application Clusters, including the Oracle Cluster Management Services (OCMS) component, is installed on a PolyServe Matrix Server cluster file system. The installation is performed and configured as thought it were for a single-node cluster.
- PolyServe Matrix Server includes a clustered file system feature called Context Dependent Symbolic Links (CDSLs) that enables administrators to create an Oracle Home directory that can be shared by all nodes in the cluster. A CDSL contains a keyword that identifies a particular location.

  When the operating system resolves the symbolic link, it translates the keyword into the appropriate path name. For example, keyword {HOSTNAME} resolves to the host name of the server where you logged in.
- To create a shared Oracle Home, the OCMS subdirectory (oracm) is converted to a CDSL linking the directory .oracm.${HOSTNAME} to oracm. The shared Oracle Home on the PolyServe Matrix Server file system can then be accessed, by way of the symbolic link, from any node in the cluster.

  For example, logging into blade6 automatically links oracm.${HOSTNAME} to oracm. The administrator can view and edit all of the configuration files for all nodes while logged in on any node in the cluster. (PolyServe provides complete documentation for creating a shared Oracle Home.)
- CDSLs can also be created for other Oracle9*i* directories such as listener.ora.
- Environment variables for the Oracle Owner must be set appropriately. The login directory for the Oracle Owner is the same for all nodes in the cluster, and environment variables such as $ORACLE_SID must be set accordingly.

For more information, refer to Oracle Technote (Doc Id 240963.1) “Managing Node Dependent Files on RAC while Using a Shared ORACLE_HOME” for supplemental information at this Web site:

http://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=240963.1

Upgrading a shared Oracle9*i* Home with minimal downtime

Applying collective patch sets from Oracle is a difficult task in the Oracle9*i* Real Application Clusters because of issues such as:

- Oracle cannot be upgraded while instances are running.
- Without a Shared Oracle Home, the upgrade must take place on every node of the cluster.
- The procedure to apply a Patchset (for example, 9.2.0.4) to a release cannot be rolled back.
- Rolling upgrades are not supported. That is, a database cannot be opened simultaneously by an instance at release levels 9.2.0.3 and 9.2.0.4.
- Applying an Oracle9*i* patch set on Linux requires a baseline release of 9.2.0.1. For example, an Oracle Home at release level 9.2.0.2 cannot be upgraded to 9.2.0.3. Without a Shared Oracle Home, administrators must reinstall 9.2.0.1 on all nodes of the cluster and then proceed to apply the patchset.

These issues are not insurmountable; however, in combinatuib, they severely impact system availability because they must be performed with the database shut down on all nodes.

With PolyServe Matrix Server, a Rapid Patch method can be used to minimize these issues and make it possible to upgrade to a higher patch set level with an absolute minimum of Oracle database downtime.

> **Note:** It is Oracle's stated intent to support "rolling upgrades" for patch installs under Oracle10g. However, the Rapid Patch method has specific value where Oracle9i is concerned.

Oracle subclusters

The Rapid Patch procedure involves creating and manipulating Oracle subclusters. Implementing an Oracle subcluster involves manipulation of the cmcfg.ora file and two OCMS quorum disks. An Oracle subcluster can execute either the same or different versions of OCMS. The pivotal component is the OCMS quorum disk. As long as the quorum disks are different and there are no common server IP addresses among the subclusters, OCMS does not know about the other subcluster.

A running Oracle OCMS cluster can be manipulated into subclusters without database downtime, and the subcluster can rejoin the original cluster without interruption.

Upgrade procedure

The is a high-level overview of the upgrade procedure:

1. Select a single node to be the upgrade node. Stop PolyServe Matrix Server on this node. The remaining nodes should continue Oracle9*i* Real Application Clusters operations.
2. On the upgrade node, change the IP addresses used by OCMS.
3. Create a target directory for the Oracle9*i* Real Application Clusters upgrade (the new Oracle Home) on the PolyServe Matrix Server cluster file system.
4. Start PolyServe Matrix Server on the upgrade node and then create a symbolic link for the Oracle user's login directory to the target directory on the upgrade node.
5. Install Oracle9*i* Real Application Clusters and the appropriate patch set into the new Oracle Home.
6. Stop Oracle and OCMS on the upgrade node and then stop PolyServe Matrix Server.
7. Reverse the temporary LAN assignments made in step 2.
8. Start PolyServe Matrix Server on the upgrade node.
9. On the other nodes, shut down the production Oracle Instances. Downtime begins.
10. On the other nodes, shut down OCMS, gsd, listeners, and so on.
11. Migrate Oracle Home customizations to the new Oracle Home, change symbolic links on all nodes to point to the new Oracle Home, and install the PolyServe ODM software if desired.
12. Start OCMS, gsd, listeners, and Instances on all nodes. Downtime ends!

> **Note:** The complete procedure is available from PolyServe:
>
> `http://www.polyserve.com/products_literature.html`

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information about ordering these publications, see "How to get IBM Redbooks" on page 82. Note that some of the documents referenced here may only be available online.

- *Implementing Oracle9i RAC with Linux on IBM @server xSeries servers*, REDP-0410-00

Other publications

These publications, provided with the products are also relevant as further information sources:

- Oracle documentation about Oracle9*i* Real Application Clusters and Oracle9*i* Real Application Clusters:
  - *Oracle9i Real Application Clusters Installation Guide Release 9.2.0.1 for UNIX Systems*
  - *Oracle9i Real Application Clusters Administrator's Reference 9.2.0.1 for UNIX Systems*
  - *Oracle9i Real Application Clusters Release Notes Release 2 (9.2.0.1.0) for Linux Intel*
  - *Oracle9i Real Application Clusters Online Generic Documentation CD-ROM Installation and Usage Notes*
  - *Oracle9i Real Application Clusters Real Application Clusters Installation and Configuration*
  - *Oracle Enterprise Manager Configuration Guide*
- PolyServe documentation for more information about PolyServe Matrix Server:
  - *PolyServe Matrix Server 1.2.0 for Red Hat Advanced Server 2.1 Installation Guide*
  - *PolyServe Matrix Server 1.2 for Linux Administration Guide*
  - *Create an Oracle Home Directory Using Context Dependent Symbolic Links Tech Tip TT0003*

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM

  `http://www.ibm.com`
- Linux Technology Center

  `http://www.ibm.com/linux/ltc`
- Oracle

  `http://www.oracle.com/apps_benchmark/html/results.html`

- PolyServe

  http://www.polyserve.com/products_literature.html

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Oracle9*i* Real Application Clusters and PolyServe Matrix Server

on IBM *e*server xSeries and BladeCenter

Redpaper

Flexible database clusters for mission-critical database applications

PolyServe Matrix Server benefits for Oracle9*i* Real Application Clusters

Proofs of concept for flexible database clusters

How do I implement a database server that can dynamically scale to meet my changing IT requirements? How do I improve the response time and throughput of my database server? How do I deliver availability while still meeting my budget targets? How do I manage large user populations and high volumes of transactions, and still respond quickly to client requests?

These are some of the questions businesses ask as they try address the challenges of managing mission-critical database applications, data, and servers.

This IBM Redpaper introduces the concept of flexible database clusters as a proven solution to these challenges. This type of cluster is built on the IBM Eserver xSeries servers or BladeCenter servers, and it features Oracle9*i* Real Application Clusters running on a PolyServe Matrix Server cluster file system.

Proofs of concept and the joint IBM and Oracle certification process for assuring the interoperability of flexible database clusters are presented in this publication. It also features two case studies that demonstrate the benefits of the PolyServe Matrix Server for Oracle9*i* Real Application Clusters when implemented on xSeries or BladeCenter servers.