White Paper

EMC XTREMIO HIGH-PERFORMANCE CONSOLIDATION SOLUTION FOR ORACLE

EMC XtremIO, VMware vSphere, Oracle Database, Oracle RAC

- Optimize high throughput for OLTP and OLAP/DW workloads
- Virtualize and consolidate database instances with superior performance
- Accelerate database copying
- Reduce storage footprint for multiple copies of the same database

EMC Solutions

Abstract

This white paper describes the performance and operational advantages of a virtualized Oracle Database with VMware vSphere deployed on an EMC® XtremIO™ all-flash array, and describes how the solution enhances consolidation and virtualization of Oracle Database environments.

May 2014

EMC² PROVEN SOLUTIONS

vmware® PARTNER

TECHNOLOGY ALLIANCE

EMC²

Part Number H12117

Table of contents

Executive summary

Business case

Business needs are driving growth in the volume and velocity of data being collected. At the same time, the demand to quickly turn this data into information and gain insight into opportunities and risks is increasing.

Databases, such as Oracle Database 11*g*, are used to support business-critical applications. To deliver fast response times across the range of applications, these databases require storage designed for both low-latency transactional I/O and high-throughput analytic workloads.

Virtualization enables greater consolidation of types of database workloads. Often, via consolidation, both online transaction processing (OLTP) and online analytical processing (OLAP) workloads share the same servers and storage. Therefore, for optimal performance, the underlying storage infrastructure must also be designed to handle the different workloads in a consolidated infrastructure.

The EMC® XtremIO™ all-flash array effectively addresses the effects of virtualization on I/O-intensive database workloads with impressive random I/O performance and ultra-low latency. This applies equally to OLTP and OLAP workloads, and to the consolidation of multiple workloads on a common storage platform. XtremIO also provides new levels of speed and provisioning agility to virtualized environments with space-efficient snapshots, inline copy deduplication, and accelerated provisioning via VMware vStorage APIs for Array Integration (VAAI). The results include storage and database license savings, breakthrough simplicity for storage management and provisioning, and new capabilities for real-time analytics and development/test cycles.

Solution overview

Virtualizing database servers is a proven strategy for consolidating databases, but it can present unique challenges. When multiple applications are consolidated on fewer physical hosts, the I/O workload can become highly random to the back-end storage because the virtual machines share physical resources such as host bus adapters (HBAs). The XtremIO all-flash storage array is built to perform in these demanding virtualized environments with very high and consistent random I/O performance with the best cost-per-database economics.

This solution demonstrates the benefits of deploying Oracle Database 11*g* on VMware vSphere using XtremIO storage.

Benefits of the solution include:

- Consolidated virtualized Oracle databases and other applications on the same physical hosts and storage
- Consistent performance and up-time for application service-level agreements (SLAs)
- Ease of use and simplified growth planning with XtremIO
- Optimal storage footprint for the entire database environment, including all production and test/dev copies

Key results

This solution shows that XtremIO delivers:

- Fast and simple setup with little or no storage tuning required
- Large-block random I/O and high bandwidth for a large data warehouse with multiple concurrent users
- Increased I/O per second (IOPS) and consistent latency for scaling OLTP workloads
- Consistent performance for consolidated and virtualized database environments with high random I/Os
- Substantial storage footprint savings by using XtremIO inline deduplication
- High fault tolerance/protection for high-availability, clustered systems such as Oracle Real Application Clusters (RAC) and VMware vSphere High Availability (HA)

Introduction

Purpose

This white paper describes a highly available and scalable solution for Oracle Database 11*g* deployed in a virtualized vSphere environment with XtremIO storage.

Scope

The scope of the white paper is to:

- Introduce the key enabling technologies
- Describe the solution architecture and design
- Describe and validate the key components and processes
- Identify the key business benefits of the solution

Audience

This white paper is intended for Oracle database administrators (DBAs), VMware administrators, storage administrators, IT architects, and technical managers responsible for designing, creating, and managing Oracle databases, infrastructure, and data centers.

Technology overview

Overview

The key technology components used in this solution are:

- EMC XtremIO with VAAI support
- VMware vSphere
- VMware vCenter
- EMC PowerPath®/VE
- Oracle Database 11*g* R2 Enterprise Edition
- Oracle RAC 11*g* R2

EMC XtremIO

The EMC XtremIO storage array is an all-flash system based on a scale-out architecture. The system uses building blocks, called X-Bricks, which can be clustered together to grow performance and capacity as required. This solution uses two X-Bricks.

XtremIO uses flash to deliver value across the following main dimensions:

- **Performance:** Regardless of how busy the system is, and regardless of storage capacity utilization, latency and throughput remain consistent, predictable, and constant. Latency within the array for an I/O request is typically far less than one millisecond (ms). Figure 1 shows an example of the XtremIO dashboard used to monitor performance.

Figure 1. XtremIO Storage Management Application dashboard

- **Scalability:** The XtremIO storage system is based on a scale-out architecture. The system begins with a single X-Brick. When additional performance and capacity are required, the system scales out by adding X-Bricks. Performance scales linearly, ensuring that two X-Bricks supply twice the IOPS, three X-Bricks supply three times the IOPS, and four X-Bricks supply four times the IOPS of the single X-Brick configuration. Latency remains consistently low as the system scales out. XtremIO arrays scale out to any required performance or capacity level, as shown in Figure 2.

Figure 2. X-Bricks scaling units

- **Efficiency:** The core engine implements content-based inline data reduction. XtremIO automatically reduces (deduplicates) data as data enters the system. This reduces the amount of data written to flash, improving longevity of the media and reducing cost. XtremIO allocates capacity to volumes on-demand in granular 4 KB chunks. Volumes are always thin-provisioned without any loss of performance, over-provisioning of capacity, or fragmentation.
- **Data protection:** XtremIO uses a proprietary flash-optimized data protection algorithm, XtremIO Data Protection (XDP), which provides RAID 6 protection for data, while enabling performance that is superior to any existing RAID algorithms. Optimizations in XDP also result in fewer writes to flash media for data protection purposes.
- **Functionality:** XtremIO supports high–performance, space-efficient snapshots, inline data reduction, thin provisioning, and full VMware vSphere VAAI with support for Fibre Channel (FC) and iSCSI protocols.

VAAI support

XtremIO is fully integrated with vSphere through the vSphere VAAI plug-in for virtual machine provisioning and cloning, virtual machine disk (VMDK) provisioning, and overall seamless deployment of large-scale virtualization. XtremIO delivers high-performance, low-latency response times, and low provisioning times for all storage provisioning choices at the VMDK level.

XtremIO supports the VAAI block-zero primitive and writes zero blocks in a way that removes the performance drawbacks of provisioning eager-zero-thick (EZT) volumes for virtual disks.

VMware vSphere

The virtualization layer decouples the application from the underlying physical resources. This enables greater flexibility in the application layer by eliminating hardware downtime for maintenance and changes to the physical system without affecting the hosted applications. In a server virtualization use case, this layer enables multiple independent virtual machines to share the same physical hardware.

VMware vSphere is a complete and robust virtualization platform, virtualizing business-critical applications with dynamic resource pools for unprecedented flexibility and reliability. It transforms the physical resources of a computer by virtualizing the CPU, RAM, hard disk, and network controller. This transformation creates fully functional virtual machines that run isolated and encapsulated operating systems and applications.

VMware vCenter

VMware vCenter is a centralized management system for a VMware virtual infrastructure. This system provides you with a single interface that you can access from multiple devices for all aspects of monitoring, managing, and maintaining a virtual infrastructure.

vCenter also manages some advanced features of a VMware virtual infrastructure, such as vSphere HA, VMware vSphere Distributed Resource Scheduler (DRS), VMware vMotion, and VMware Update Manager.

EMC PowerPath/VE

EMC PowerPath/Virtual Edition (VE) for vSphere delivers multipathing features that optimize vSphere virtual environments. PowerPath/VE is installed as a kernel module on the VMware ESXi host and works as a multipathing plug-in (MPP) that provides enhanced path management and load-balancing capabilities for ESXi hosts.

Oracle Database 11*g* Enterprise Edition

Oracle Database 11*g* Enterprise Edition delivers performance, scalability, security, and reliability on a choice of clustered or single servers running Windows, Linux, or UNIX. Oracle Database 11*g* Enterprise Edition supports advanced features, either included or as extra-cost options, that are not available with Oracle Database 11*g* R2 Standard Edition. These include security features, such as Virtual Private Database, and data warehousing options, such as partitioning and advanced analytics.

Oracle RAC 11*g* R2

In Oracle RAC 11*g* R2, the Oracle Grid Infrastructure combines Oracle Automatic Storage Management (ASM) and Oracle Clusterware into a single set of binaries, separate from the database software. This infrastructure now provides all the cluster and storage services required to run an Oracle RAC-enabled database.

Oracle ASM

Oracle ASM is an integrated, cluster-aware database file system and disk manager. ASM file system and volume management capabilities are integrated with the Oracle database kernel. In Oracle RAC 11*g* R2, ASM has also been extended to include support for Oracle Cluster Registry (OCR) and voting files to be placed within ASM disk groups.

Oracle Clusterware

Oracle Clusterware is a portable cluster management solution that is integrated with the Oracle database. Oracle Clusterware provides the infrastructure necessary to run Oracle RAC, including Cluster Management Services and High Availability Services. You can also use it to make a non-Oracle application highly available across the cluster.

Solution architecture

Overview

This section describes the solution architecture.

The solution provides an optimal cost-to-performance ratio for Oracle mission-critical application environments. The Oracle Database is deployed as a virtualized four-node RAC database on an XtremIO storage array consisting of two XtremIO X-Bricks.

Architecture diagram

Figure 3 shows the layout of the solution.

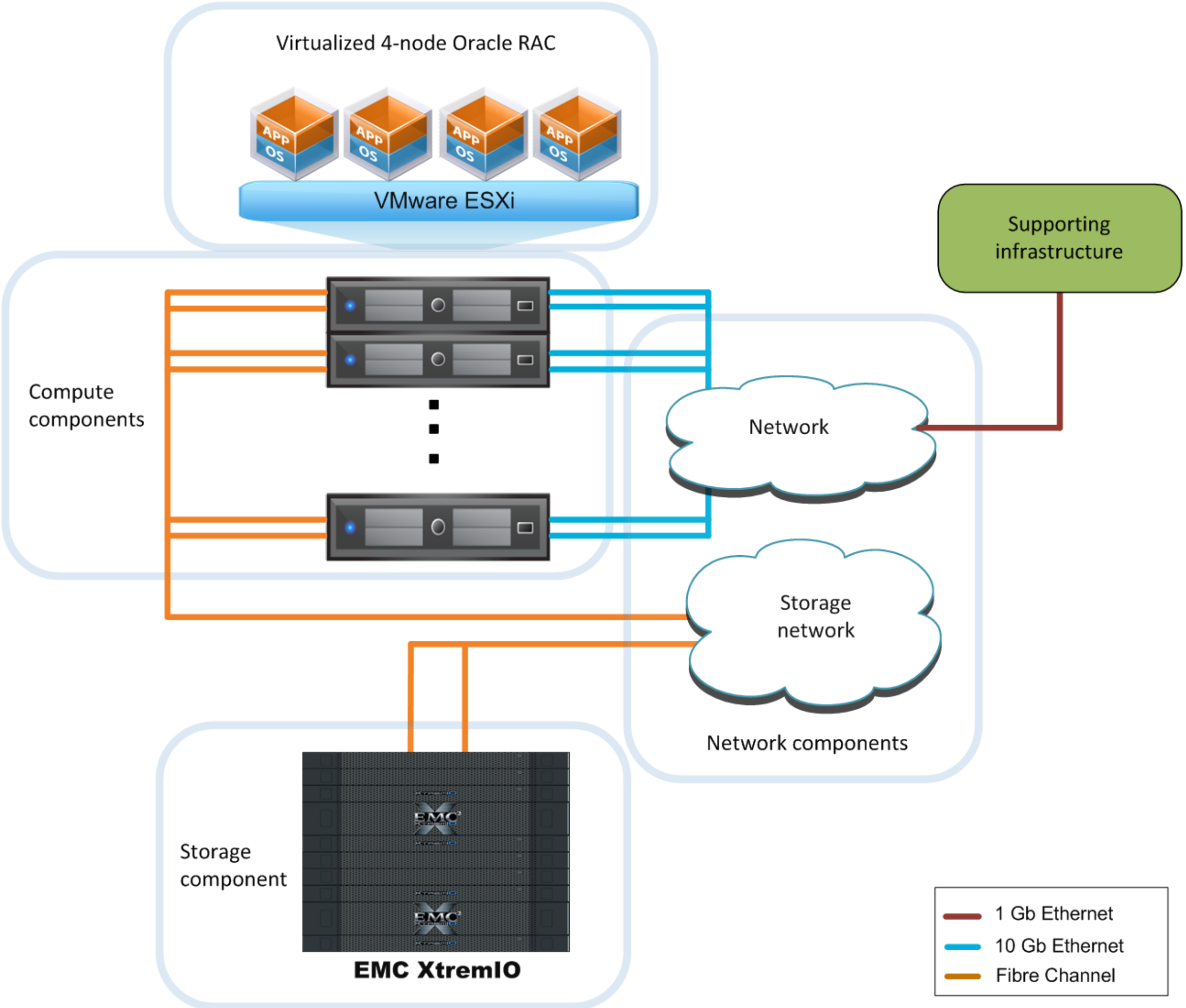

Figure 3. Solution architecture

The architecture is composed of the following:

- **Compute layer:** Comprises four servers that use a total of 32 cores with 2.9 GHz processors and a total of 2 TB RAM. The rack server enables a high-performing, consolidated, virtualized approach to an Oracle infrastructure, resulting in deployment flexibility without the need for application modification.

- **Network layer:** Comprises two IP switches and two director-class SAN switches, which we[1] configured to produce 108 GB/s active bandwidth. The SAN switches are designed for deployment in storage networks supporting virtualized data centers and enterprise clouds.
- **Storage layer:** Comprises two X-Brick arrays (12U) with 14.94 TB of usable physical capacity

The servers are installed with vSphere and configured as a VMware ESXi cluster.

Four virtual machines are created on this ESXi cluster, which runs a four-node Oracle RAC database. We examined the performance by separately running OLTP and data warehouse (DW) workloads against the Oracle RAC database.

Hardware resources

Table 1 lists the hardware resources used in the solution.

Table 1. Hardware resources

Hardware	Quantity	Configuration
Storage array	1	XtremIO X-Brick cluster consisting of two units
Servers	4	8 cores, 2.9 GHz processors, 512 GB RAM, including: • 1 x 1 Gb Ethernet (GbE) network interface card (NIC) • 1 x 10 GbE NIC
LAN switches	2	10 GbE
SAN switches	2	FC

Software resources

Table 2 lists the software resources used in this solution.

Table 2. Software resources

Software	Version	Notes
VMware vSphere	5.0	Hypervisor hosting all virtual machines
VMware vCenter	5.1	vSphere manager
Red Hat Enterprise Linux	6.3	Operating system for database servers
Oracle Database 11*g* Release 2	Enterprise Edition 11.2.0.3	Database
Oracle Grid Infrastructure 11*g* Release 2	Enterprise Edition 11.2.0.3	Clusterware with ASM for volume management
Silly Little Oracle Benchmark (SLOB)	2	OLTP benchmark tool

[1] In this white paper, "we" refers to the EMC Solutions engineering team that validated the solution.

Storage layer: EMC XtremIO

Cluster design

EMC XtremIO includes a scale-out cluster design that adds capacity and performance in balance together to meet any storage requirement. Each cluster building block has highly available, fully active/active storage servers with no single point of failure. As you expand a cluster, XtremIO automatically balances the workloads of all hosts and clusters to maintain performance.

The XtremIO operating system (XIOS) manages storage clusters and provides the following functionality:

- Ensures that all solid-state drives (SSDs) in the cluster are evenly loaded to provide the highest possible performance and endurance required for high-demand workloads throughout the life of the array.
- Eliminates the need to perform complex configuration steps for traditional arrays. It negates the need to set RAID levels, determine drive-group sizes, set stripe widths, set caching policies, build aggregates, or do any other manual configuration.
- Automatically and optimally configures volumes and ensures that I/O performance on existing volumes and on data sets automatically increases when a cluster scales out.
- Manages the process of expanding clusters and ensures that data remains balanced across new X-Bricks. It eliminates the need to restripe data if application requirements change. Every volume receives the full performance potential of the entire XtremIO cluster.

Inline data reduction

The XtremIO all-flash array performs inline data deduplication based on an algorithm that checks to ensure that each 4 KB data block being stored on the array is not the same as existing content. The result is that every storage I/O is deduplicated in real time on ingest with only unique blocks ever being written to the flash storage. Moreover, deduplication on XtremIO aids performance, as SSD I/O cycles are never consumed by writing duplicate blocks or processing undeduplicated data. This preserves the maximum I/O capability for serving host requests.

When archivelog mode is enabled, once an online redo log file is full, the archiver process copies the redo data from this online redo log file to the archive log file destination. The filled redo log is not available for writing by the log writer process until archiving is complete. Because the archive log is an exact copy of the online redo log, the array does not perform physical write I/O to the back-end SSDs; instead, it sends updates to in-memory pointers, and the write operation is completed instantly.

After the online redo log file being archived is written with new redo data by Log Writer (LGWR), the in-memory pointer of the online redo log file is updated and points to the newly allocated physical redo blocks that store the newly written redo data. The in-memory pointer of the archived redo log file is intact, which means the physical blocks consumed by the archived redo log file are not deprecated as long as they exist in the archive log destination.

XtremIO also provides deduplication-aware caching where blocks that are held in cache can be served for any logical reference to those blocks. Deduplication-aware caching combined with inline deduplication dramatically decreases latencies for handling challenging situations, such as multiple concurrent virtual machine boots, providing consistent sub-millisecond data access.

Thin provisioning

Along with delivering high performance, XtremIO offers thin provisioning capabilities to allocate on-demand capacity as applications need it without any post-reclamation operations or impact on array-storage I/O performance. XtremIO thin provisioning is also granular, with capacity allocated in 4 KB blocks to ensure thrifty use of flash capacity storage, which is consistent with how vSphere uses I/O block sizes.

Fault protection

XtremIO delivers reliability and availability with completely redundant components and the ability to tolerate any component failure without loss of service. XtremIO includes the following fault–protection features:

- Dual power supplies in controllers and disk array enclosures (DAEs) to support loss of a power supply while keeping the controller/DAE in service
- Redundant active/active controllers to support controller failures
- Redundant serial-attached SCSI (SAS) interconnect modules in the DAEs
- Redundant inter-controller communication links
- Multiple host connections with multipath capabilities to survive path failures
- XDP to tolerate SSD failures
- Multiple techniques to ensure initial and ongoing data integrity

This fault tolerant design is ideal for deploying high-availability cluster systems such as Oracle RAC 11*g*.

Scalability

XtremIO storage clusters support a fully distributed, scale-out design that enables linear increases in both capacity and performance for infrastructure agility. XtremIO uses a building-block approach in which the array is scaled with additional X-Bricks.

XtremIO provides host access using N-way active/active controllers for linear scaling of performance and capacity for simplified support of growing virtualized environments. As a result, as capacity in the array grows, performance also grows with the addition of storage controllers.

In-memory metadata operations

The XtremIO cluster distributes metadata evenly across all storage controllers, maintaining metadata in memory during runtime. Metadata is hardened to SSD to enable the array to tolerate failures and power loss, but during normal operations all metadata lookups are memory-based. This is possible only by segmenting the metadata tables and spreading them evenly across all storage controllers. In contrast, a dual-controller design might not contain enough RAM to store all metadata in memory and would require de-staging large amounts of metadata to flash, with several associated performance drawbacks.

The XtremIO in-memory metadata and unique in-line deduplication model combine to deliver unprecedented capabilities in virtualized data centers.

XtremIO Management Server

XtremIO Management Server (XMS) enables you to control and manage the system. XMS functionality includes:

- Initializing and formatting new systems
- Monitoring system health and events
- Monitoring system performance
- Maintaining a performance statistics history database
- Providing GUI and CLI services to clients
- Implementing volume management and data protection operations
- Maintaining (stopping, starting and restarting) the system

XMS is preinstalled with and accessed via an easy to use GUI, XtremIO Storage Management Application (XSMA), and connects to the management ports on the X-Brick storage controllers via TCP/IP. The XtremIO cluster serves storage unaffected by a disconnected XMS.

XMS is deployed as a virtual machine on the system's management environment, which is built on a VMware HA cluster. This means that the XMS virtual machine can quickly restart in the event of failure.

Storage configuration

For this solution, XtremIO is deployed in a two X-Brick cluster with built-in, redundant 40 Gb/s QDR InfiniBand switches providing back-end connectivity between the storage controllers. This ensures a highly available, ultra-low latency network.

The XtremIO cluster contains fifty 400 GB SSDs configured with XDP to provide a physical capacity of 14.94 TB as shown in Figure 4.

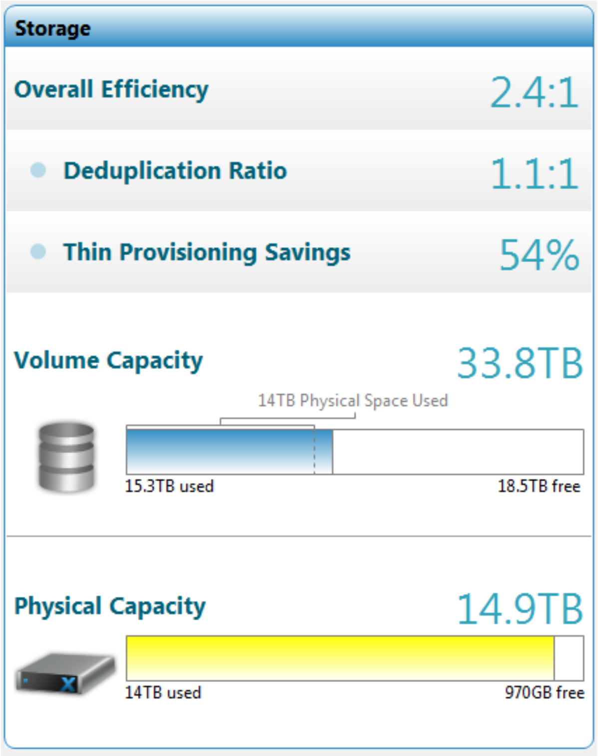

Figure 4. XSMA dashboard storage panel

Storage design

With the traditional storage design for Oracle Database, multiple RAID groups of different drive types are created, each with different levels of protection and distributed across multiple controllers.

With XtremIO, all drives are under XDP protection, and data in the array is distributed across the X-Bricks to maintain consistent performance and equivalent flash wear levels.

Databases generate both random and sequential I/O as shown in Figure 5. With XtremIO, these are treated equally as data is randomized and distributed in a balanced fashion throughout the array.

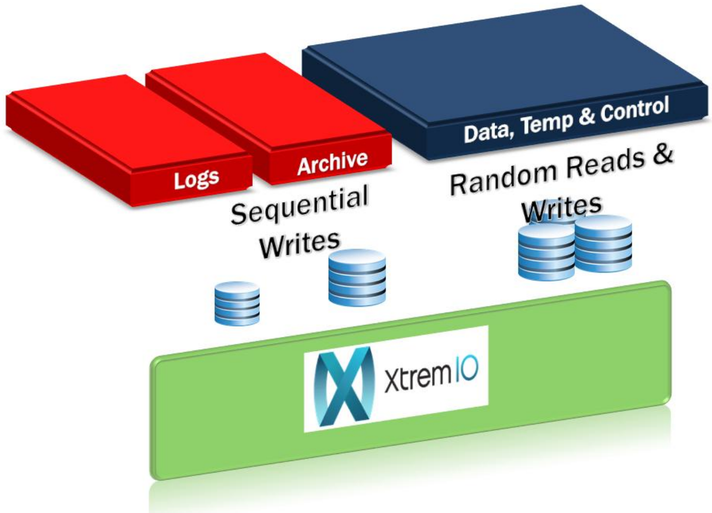

Figure 5. Database front-end random and sequential I/O

For this solution, the volume size is standardized based on data use as shown in Table 3.

Table 3. Oracle storage design on XtremIO X-Bricks

Purpose of volume	Size (GB)
Oracle ASM disks for data files	700
Oracle ASM disks for redo logs	40
Oracle ASM disks for Cluster Ready Service files (Oracle Cluster Registry and Voting disk files)	10
Oracle ASM disks for Oracle Automatic Storage Management Cluster File System (ACFS) for data load of files	200

For the OLTP database, volumes were created and presented to the ESXi servers for use with Oracle Database 11*g*, as shown in Table 4.

Table 4. Volume/LUN assignment for OLTP database

Folder name	Volume size (GB)	Number of volumes	ASM disk group	Number of initiator groups
Data	700	8	+DATA	4
Log	40	4	+REDO	4
CRS	10	2	+CRS	4

Figure 6 shows the volumes **Data11** to **Data08** grouped under the **Data** folder. These volumes comprise the ASM disk group **+DATA** for the OLTP database. In this solution, a single initiator group comprises a single physical server and its FC ports . Volume **Data11** is presented as a LUN and mapped to all four initiator groups.

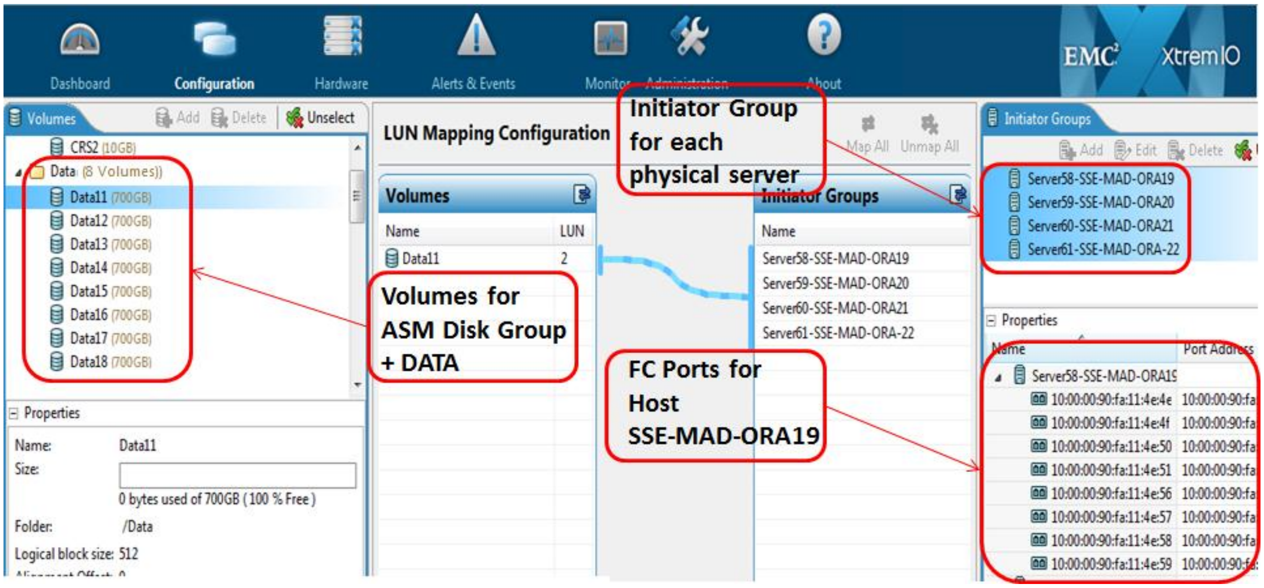

Figure 6. XSMA volume/LUN mapping to servers

For the DW database, the volumes are created and grouped in folders as listed in Table 5. Each volume is presented as a LUN to the initiator groups shown in Figure 6.

Table 5. ASM disk group design for DW database

Folder name	Volume size (GB)	Number of volumes	ASM disk group	Number of initiator groups
DWDATA	700	8	+DATA	4
DWLog	40	4	+REDO	4
DWCRS	10	2	+CRS	4
DWACFS	200	1	+CSV	4

Network layer

Overview

This section briefly describes the best practices used in this solution for SAN and IP network configuration, and for an ESXi server network configuration. When you deploy a virtualized clustered solution, such as Oracle RAC, EMC recommends that you consider both compute and network redundancy when designing your network for fault tolerance.

SAN best practices

EMC recommends that you use the following SAN best practices:

- Use multiple HBAs on the ESXi servers and two SAN switches to provide multiple paths between the hosts and the XtremIO cluster.
- Zone each FC port from the database servers to the ports on the XtremIO X-Bricks for high availability and performance.
- Use path management and dynamic multipathing software, such as PowerPath/VE, on the hosts to enable the failover process to alternate paths and to provide load balancing.

IP network best practices

EMC recommends that you use the following IP network best practices:

- Use multiple network cards and switches for network redundancy.
- Use 10 GbE for network connection, if it is available.
- Use virtual local area networks (VLANs) to logically group devices that are on different network segments or subnetworks.
- Enable and configure jumbo frames[2] throughout the physical or virtual stack for 10 GbE networks.

vSphere network best practices

Networking in virtual and physical environments uses the same concepts and many best practices apply to both. However, additional considerations are applicable for traffic segmentation, availability, and throughput when virtualizing.

This solution was designed to efficiently manage multiple networks and redundancy of network adapters on ESXi hosts. The key best practice guidelines are to:

- Isolate infrastructure traffic from virtual machine traffic for security.
- Use the VMXNET3 family of paravirtualized network adapters.
- Aggregate physical network cards for network redundancy and performance. For example, use pairs of physical NICs per server/vSwitch and uplink each physical NIC to separate physical switches.

For more information on networking with vSphere, refer to the instructions in *VMware vSphere Networking*.

---

[2] Maximum Transfer Unit (MTU) sizes of greater than 1,500 bytes are referred to as jumbo frames. Jumbo frames require gigabit Ethernet across the entire network infrastructure, including servers, switches, and database servers.

Compute layer

Overview

The choice of a server platform for a virtualized infrastructure is based on both the supportability of the platform and the technical requirements of the environment. In production environments, the servers must have:

- Sufficient cores and memory to support the required number and workload of the virtual machines
- Sufficient connectivity, both Ethernet and FC, to enable redundant connectivity to the IP and storage network switches
- Sufficient capacity to withstand a server failure and support failover of the virtual machines

In this test environment, we used four physical servers configured as a vSphere HA cluster and each running a vSphere ESXi server. We then configured four virtual machines to create a four-node virtualized Oracle RAC database deployment.

Oracle Database 11*g* R2 and later versions are fully supported when deployed on vSphere ESXi technology[3]. This includes deployment of an Oracle RAC configuration.

Compute and storage resources

EMC recommends that you implement the following VMware compute resource best practices as explained in the *Oracle Databases on VMware Best Practices Guide*:

- Use Non-Uniform Memory Access (NUMA) on the ESXi servers, a computer architecture in which memory located closer to a particular processor is accessed with less delay than memory located farther from that processor.
- Allocate virtual machine memory (vRAM) in a virtual machine to be less than or equal to the local memory accessed by the NUMA node (processor).
- Install VMware Tools, including several utilities that enhance the performance of the virtual machine's guest operating system and improve the ability to manage the virtual machine.
- Allocate vRAM to be at least twice the size of the Oracle System Global Area (SGA).
- Configure the virtual machine memory reservations to be, at a minimum, the size of the Oracle SGA and operating system overhead.
- Use multiple paravirtualized SCSI (PVSCSI) controllers for the database virtual machines. The use of multiple virtual SCSI controllers enables the execution of several parallel I/O operations inside the guest operating system. Configure the PVSCSI controller for the disks that are used to store the database data files.
- Isolate the Oracle redo log I/O traffic from the data file I/O traffic through separate virtual SCSI controllers. As a best practice, you can use one controller for the operating system and swap file LUNs, another controller for database redo log file LUNs, and one or more additional controllers for database data file LUNs.

---

[3] My Oracle Support Doc ID 249212.1 defines Oracle's VMware support policy.

Virtualization

Network virtualization

On each ESXi server, we created two standard vSwitches with a common configuration as listed in Table 6.

Table 6. vSwitch configuration

Name	Purpose
vSwitch0	Management and public virtual machine traffic
vSwitch1	Fault tolerant configuration for Oracle RAC interconnect

We used the high-performance VMXNET3 driver to assign two vNICs (1 GbE and 10 GbE) to each virtual machine. We mapped the 1 GbE vNIC to vSwitch0 to deliver public traffic, and the 10 GbE vNIC to vSwitch1 to deliver Oracle RAC interconnect traffic.

Virtual machine template configuration

VMware templates minimize deployment time and automate repetitive installation and configuration tasks for each virtual machine that is required. Using the template, you can deploy a virtual machine with a preconfigured operating system and application users, and with software that is configured and ready to use with minimal user intervention.

Customization specifications stored in vCenter simplify the rollout of virtual machines. A deployment wizard, automation tool, or script can use these templates to automatically create or amend server settings (such as server name, time zone, and network configuration) before you build the new virtual machine.

Configure the virtual machine template in the vSphere client according to the requirements and prerequisites for the Oracle software, which are shown in Table 7.

Table 7. Virtual machine template configuration

Part	Description
CPU	8 vCPUs for OLTP workload
Memory	128 GB
Operating system	Red Hat Enterprise Linux Server Release 6.3
Kernel	2.6.32-279.el6
Virtual network interfaces	Eth0: Public/management IP network Eth1: Dedicated to cluster interconnect
OS user (user created and password set)	Username: oracle UserID:1101
OS groups	Group: oinstall GroupID:1000 Group: dba GroupID:1031

Part	Description
Software preinstalled	Oracle Grid Infrastructure (for standalone server) Oracle Database
RPM packages installed (as Oracle prerequisites)	As specified in the Oracle installation guide
Disk configuration	128 GB virtual disk for root, /tmp, the swap space, and Oracle 11*g* R2 Database binaries

After deploying the virtual machines, we added the disk for the database storage as raw device mapping (RDM).

Enabling access to the storage devices

On each virtual machine, we added the LUNs for database storage from the XtremIO array as RDM and spread them across four PVSCSI controllers to balance I/O, as shown in Figure 7.

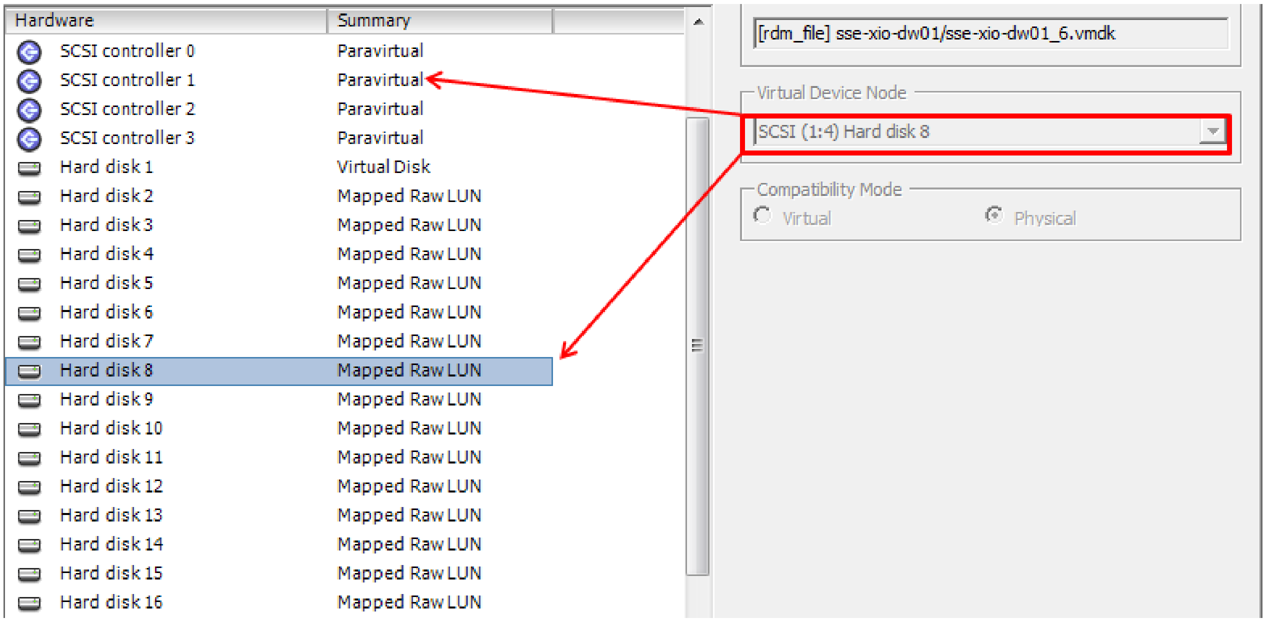

Figure 7. Disk configuration in the virtual machine

SCSI Bus Sharing on the SCSI controllers was set to **Physical** so that virtual machines would have direct and shared access to the raw LUNs across the ESXi servers, as shown in Figure 8.

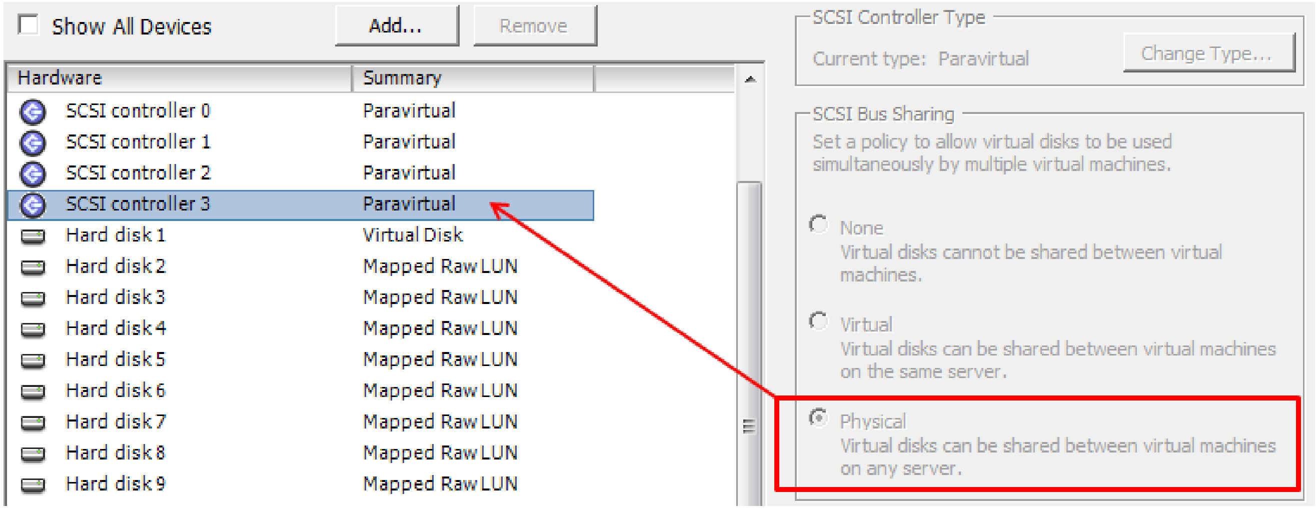

Figure 8. SCSI Bus Sharing settings

Oracle Database

Overview

In this solution, we created a virtualized four-node Oracle OLTP RAC and DW database, both using Oracle ASM and Oracle ACFS, on vSphere.

In Oracle Database11*g*, Oracle ASM and Oracle Clusterware are integrated into the Oracle Grid Infrastructure. ACFS extends ASM by providing a robust, general-purpose, and extent-based journaling file system. In this solution, we used ASM to store the database files and ACFS to store the comma separated values (CSV) files for the data-loading test.

Figure 9 shows the Oracle OLTP RAC ASM instances.

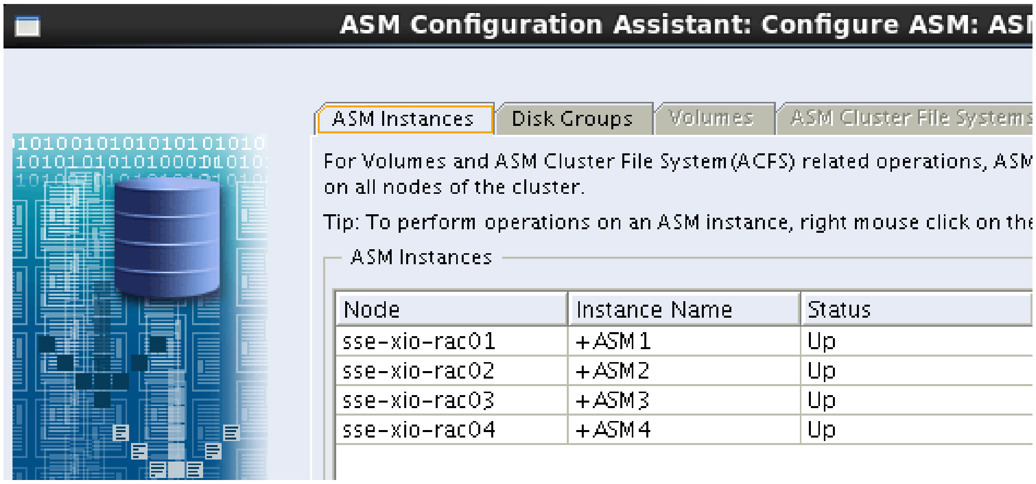

Figure 9. Oracle ASM instances

Note: The database used for either OLTP or DW testing was configured with archive logging disabled to achieve optimal performance.

OLTP database storage design

As shown in Table 4 on page 19, we used three ASM disk groups to store the relevant database files, including data files, control files, online redo log files, voting disk files, CRS files, and temporary files.

We used a CSV disk group to store the CSV files for the data-loading test. External redundancy was used for the ASM disk groups, and default settings were used for the remaining ASM disk groups' attributes. When we created the DATA and REDO disk groups, we set the sector size attribute to 4 KB to maximize the I/O performance on the XtremIO storage. We also set the block size of the online redo log files to 4 KB to match the sector size of the REDO disk group.

Oracle ASM disk group layout for OLTP database

Multiple ASM disk groups with external redundancy were used in the configuration:

- +CRS contains the OCR and voting files installation location.
- +DATA contains the data files, control files, and temporary files.
- +REDO holds the online redo log files.

OLTP database and workload profile

Table 8 lists each OLTP database workload profile for the solution.

Table 8. Database workload profile for each OLTP database

Profile characteristic	Details
Database type	OLTP
Database size	4 TB
Oracle Database	11*g* R2 RAC on ASM
Instance configuration	SGA size: 16 GB **Note:** Considering that larger database cache size will buffer more data, we configured a very small buffer cache to generate a stable and high physical I/O workload.
Workload profile	OLTP workload simulated by SLOB
Data block size	8 KB

Configuring ASM disk group for DW database

Table 9 details the ASM disk group design used for the DW database. As with the OLTP ASM disk groups layout, +DATA and +REDO were created. An additional ASM disk group, +CSV, was created to act as a network share using the Oracle ACFS file system. Flat data files are stored on this ACFS share for loading into the database.

Table 9. Storage design to ASM disk group mapping for the DW database environment

Folder name	Volume size (GB)	Number of volumes	ASM disk group name
DATA	700	8	+DATA
REDO	40	4	+REDO
CSV	200	1	+CSV
CRS	10	2	+CRS

DW database and workload profile

Table 10 details the database and workload profile for the solution. We used a DW workload generator to create a DW database and deliver the DW workloads, including the query and data-loading workloads required for the solution.

Table 10. Database and workload profile for each DW database

Profile characteristic	Details
Database type	Data warehouse
Database size	8 TB
Oracle Database	11*g* R2 RAC on ASM
Instance configuration	• sga_target = 60 G • pga_aggregate_target = 40 G
Workload profile	DW workload
Data load source	External flat files on Oracle ACFS

Profile characteristic	Details
Data block size	16 KB

Performance testing and validation

Overview

To characterize both OLTP and DW systems, the solution test environment consisted of two main workloads. We created four virtual machines running an Oracle RAC database for the OLTP workload, and we created another four virtual machines to run an Oracle RAC database for the DW workload in the consolidated VMware environment.

We ran an OLTP workload generated by SLOB against the database to simulate a real-world OLTP workload. We used the DW workload generator to create the DW workload.

We gathered the system I/O performance metrics (IOPS, MBPS, and latency) primarily from the Oracle Automatic Workload Repository (AWR) report. In addition, we gathered metrics for I/O throughput at the server/database and storage levels.

Notes on results

Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, this workload should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments could vary significantly.

EMC Corporation does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute.

Test objectives

The overall test objectives were to:

- Demonstrate the high performance achieved when virtualized Oracle databases were run on XtremIO.
- Demonstrate how XtremIO significantly simplified storage operations for Oracle RAC clusters.
- Demonstrate sustained storage array IOPS for OLTP workloads on an Oracle four-node RAC database.
- Demonstrate sustained storage array I/O bandwidth for DW/OLTP workload on an Oracle four-node RAC database.
- Demonstrate significant storage space savings using inline data reduction on XtremIO when production volumes are cloned for test/dev environments.

Test scenarios

The following test scenarios were conducted on the solution and are described in subsequent sections:

- SLOB random single-block read/write workload test
- DW query workload test

- DW data-loading test
- Test/dev copy creation test

SLOB random single-block read/write workload test

The solution characterizes the Oracle OLTP system performance on the XtremIO array. We created a four-node Oracle RAC database in a consolidated VMware virtualized environment.

SLOB was used to generate the workload because it is the best SQL workload to drive maximum physical random I/O from a database platform.

SLOB is an I/O-intensive SQL generator, instead of a synthetic I/O generator with the following characteristics:

- Supports testing Oracle logical read (SGA buffer gets) scaling
- Supports testing physical, random single-block reads (database file sequential read/DB file parallel read)
- Supports testing random single-block writes (database file parallel write)
- Supports testing extreme REDO logging I/O
- Consists of simple PL/SQL
- Is free of all application contention

Read-only test: SLOB random single-block query-only test

This test was used to measure the performance when concurrent SLOB "zero-think-time" sessions (simulated concurrent users) ran queries against the RAC database.

The sessions were generated by the SLOB toolkit with each user running similar queries to validate the read IOPS.

Test methodology

We ran the fixed number of concurrent sessions with each user running similar OLTP queries simultaneously against the four-node RAC database, then measured the performance statistics. During the test, we controlled the number of concurrent sessions to keep the I/O latency to about 1.5 ms.

Test procedure

The test included running 32 concurrent sessions on the four-node RAC database with eight sessions on each node. Each session ran the OLTP query workload against the cluster database.

Test results

Performance statistics were captured using AWR RAC reports. We observed the physical reads value in the AWR report to assess read IOPS statistics.

Query average response time was calculated from **db file parallel read** and **db file sequential read**, under **Top Timed Events** in the AWR report, as shown in Figure 10.

	Wait		Event		Wait Time			Summary Avg Wait Time (ms)				
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	Cr
*	User I/O	db file parallel read	568,066	0.00	1,450.58	2.55	49.65	2.57	2.18	2.72	0.26	
	User I/O	db file sequential read	986,577	0.00	1,060.18	1.07	36.29	1.08	0.96	1.13	0.08	
		DB CPU			583.22		19.96					
	User I/O	utl_file I/O	140	0.00	1.96	13.99	0.07	14.29	10.79	15.48	2.33	
	System I/O	control file sequential read	2,408	0.00	1.69	0.70	0.06	0.70	0.66	0.76	0.05	

Figure 10. Query-only average response time measurement - AWR report

We used the following formula to calculate the I/O latency:

- For **db file sequential read**:
  - The total wait time is T1, which is 1,060.18 seconds.
  - The total number of waits is N1, which is 986,577.
- For **db file parallel read**:
  - The total wait time is T2, which is 1,450.58 seconds.
  - The total number of waits is N2, which is 568,066.
- The average read response time is: (T1+T2)/(N1+N2), which is (1,060.18 + 1,450.58) * 1,000/(986,577 + 568,066), therefore, the average response time is 1.615 ms.

Table 11 shows the IOPS and the corresponding I/O latency in the test.

Table 11. Read IOPS in the test

Performance metric	Four-node RAC database
IOPS	197,601
I/O latency (ms)	1.615

We achieved a total of 197,601 read IOPS and an average latency of 1.615 ms with a four-node RAC when running 32 reader sessions on four nodes.

One-hundred percent UPDATE transaction test: SLOB random single-block test

This test was used to measure the performance when concurrent SLOB zero-think-time sessions (simulated concurrent users) ran UPDATE transactions against the RAC database.

The sessions were generated by the SLOB toolkit with each user running similar UPDATE transactions to validate the read-write IOPS.

Test methodology

We ran the fixed number of concurrent sessions with each user running similar SLOB UPDATE SQL statements simultaneously against the four-node RAC database, then measured the performance statistics. During the test, we controlled the number of concurrent sessions to keep the I/O latency to about 1 ms.

We decreased the buffer cache for each database instance to push consistent physical write I/O workload to the back-end storage. The write workload was driven by the UPDATE SQL statements that generally produce the following operations:

1. Reads the data blocks to be updated into the buffer cache.
2. Updates the rows in the data blocks.
3. Commits the updated rows and triggers LGWR flushing redo entries to the online log files.

During the UPDATE workload run, the background database writer process flushed the "dirty" blocks out of the buffer cache into data files. Considering we used a very small buffer cache, the data blocks were read into the buffer cache and written out of the buffer cache soon after the rows were updated. As a result, the execution of each UPDATE operation caused both physical reads and physical writes from the back-end array.

Test procedure

The test included running 32 concurrent sessions on the four-node RAC database with eight sessions on each node. Each session ran a similar UPDATE SQL workload against the cluster database.

Test results

We used the following performance statistics from the AWR reports to calculate the workload, as shown in Figure 11.

	Wait		Event		Wait Time			Summary Avg Wait Time (m			
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev
*	User I/O	db file sequential read	11,618,228	0.00	11,100.11	0.96	94.71	0.96	0.95	0.96	0.00
		DB CPU			978.01		8.34				
	System I/O	log file parallel write	48,686	0.00	90.51	1.86	0.77	1.86	1.84	1.88	0.02
	System I/O	db file parallel write	109,143	0.00	58.40	0.54	0.50	0.54	0.52	0.55	0.01
	Cluster	gc cr grant 2-way	37,106	0.00	10.39	0.28	0.09	0.28	0.27	0.29	0.01
	Other	reliable message	5,837	0.00	5.59	0.96	0.05	1.66	0.73	3.53	1.27

Figure 11. UPDATE-only write average response time measurement - AWR report

- Physical writes were used for physical write IOPS.
- Physical reads were used for physical read IOPS incurred by UPDATE transactions. The average read response time was retrieved from the **db file sequential read** wait event, which was 0.96 ms in the AWR sample shown in Figure 11.
- Redo size was used to calculate the redo write I/O bandwidth.
- Statistics of the **log file parallel write** wait event were used to calculate the average latency for the LGWR background process.

In the AWR sample shown in Figure 11, the total wait time of the **log file parallel write** wait event was 90.51 seconds, which equals 90,510 ms. The number of waits was 48,686. The write average latency for LGWR was 90,510/48,686 = 1.86 ms.

- Statistics of the **db file sequential read** wait event were used to calculate the average physical read response time for the database foreground processes.
- Statistics of the **db file parallel write** wait event were used to calculate the average physical write response time for the database write background processes.

  In the AWR sample shown in Figure 11, the total wait time of the **db file parallel write** wait event was 58.40 seconds, which equals 58,400 ms. The number of waits was 109,143. The average response time for database read/write was 58,400/109,143 = 0.535 ms.

Table 12 shows the performance statistics of the test.

Table 12. Scaling of RAC nodes and sessions with resulting increases in IOPS

Performance metric	Four-node RAC database
Read IOPS	31,646
Write IOPS	33,136
Aggregate IOPS (Write + Read)	64,782
Redo throughput (MB/s)	26
LGWR response time (ms)	1.86
Database writer write response time (ms)	0.535
Foreground process read response time (ms)	0.96

As shown in Table 12, when running 32 write sessions on each of the nodes, we achieved 64,782 aggregated IOPS including 33,136 write IOPS and 31,646 read IOPS, which were part of the UPDATE transaction.

We used a very small buffer cache (only 64 MB, almost no data cached in the server) to generate a high write I/O workload. When we tested the IOPS, we controlled the workload to keep the read I/O latency to about 1 ms.

Seventy-five/twenty-five QUERY/UPDATE ratio test: SLOB random single-block test

This test was used to measure the performance during concurrent SLOB zero-think-time sessions (simulated concurrent users) with 75 percent of sessions running queries and 25 percent of sessions running UPDATE transactions against the RAC database. We used this to simulate a realworld workload and validate the performance of the XtremIO storage.

Test methodology

We ran the fixed number of concurrent sessions with 75 percent of sessions running similar OLTP queries and the remaining 25 percent of sessions running similar UPDATE transactions simultaneously against the four-node RAC database, then measured the performance statistics. During the test, we controlled the number of concurrent sessions to keep the I/O latency to about 1 ms.

Test procedure

The test included running 64 concurrent sessions on the RAC database, with 75 percent of sessions running similar queries and 25 percent of sessions running similar UPDATE SQL statements on the cluster database.

Test results

To calculate the workload, we used the performance statistics from the AWR reports, shown in Figure 12, just as we did with the 100 percent UPDATE transaction test.

	Wait		Event		Wait Time			Summary Avg Wait Time (ms				
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	C...
*	User I/O	db file sequential read	7,196,550	0.00	8,153.29	1.13	70.71	1.13	1.11	1.14	0.02	
	User I/O	db file parallel read	936,985	0.00	2,416.68	2.58	20.96	2.58	2.52	2.60	0.04	
		DB CPU			1,233.61		10.70					
	System I/O	log file parallel write	244,207	0.00	254.49	1.04	2.21	1.04	1.04	1.05	0.01	
	System I/O	db file parallel write	66,278	0.00	38.73	0.58	0.34	0.58	0.56	0.60	0.02	
	Other	DFS lock handle	7,499	100.00	14.19	1.89	0.12	1.90	1.54	2.42	0.37	

Figure 12. Average response time - 75/25 QUERY/UPDATE ratio test - AWR report

Table 13 shows the XtremIO array IOPS and the corresponding I/O response time in the test.

Table 13. Scaling of RAC nodes and sessions with resulting increases in read IOPS

Performance metric	Performance data
Read IOPS	112,124
Write IOPS	28,830
Aggregate IOPS (write + read)	140,954
Redo throughput (MB/s)	21
LGWR response time (ms)	1.042
Database read/write (DBWR) response time (ms)	0.584
Foreground process read response time (ms)	1.299

For the mixed workload including query and UPDATE transactions, our tests results show that XtremIO storage can maintain a high-level IOPS. In our 75/25 QUERY/UPDATE ratio, the read IOPS reached 112,124 and the write IOPS was 28,830, which kept the read I/O latency to about 1 ms.

DW query workload test

The DW workload generator provided an Oracle DW test workload to test and validate the performance of typical Oracle DW workloads on the XtremIO storage platform.

The schema in the kit had 12 tables including two fact tables—sales and returns. The remaining tables acted as dimension tables. The fact tables were range-partitioned by date and sub-partitioned by hash on their join key. The database size was 3.8 TB. Multiple concurrent users ran a series of typical queries against the data. The throughput was measured during the test.

Test methodology

This test was used to measure the performance when running the fixed number of concurrent users, with each user running similar queries. The concurrent users were generated by the DW workload generator. Direct path read was chosen at runtime for all serial full-table scans in this test.

Test procedure

The test included running 32 concurrent users on each node to get a total of 128 concurrent users running on the four-node RAC database, with each session running a similar DW query workload using the scripts in the DW workload generator.

Test results

To assess the query throughput (GB/s), we used **physical read bytes** in the AWR report.

During testing, the throughput reached 5 GB/s when we ran the DW workload on the four-node RAC database.

DW data-loading test

Modern enterprise DWs require frequent and large data loads periodically throughout the day. The 24/7 nature of the DW no longer allows DBAs a lot of time for data loading. Therefore, simulating the impact of data extract, transform, and load (ETL) processes on the performance of the database is important.

This test scenario demonstrates the ETL processes on the production database and records the performance data, especially the throughput (physical write total MB/s), during the ETL load.

We used Oracle external tables, which used the ORACLE_LOADER access driver, to load data from external tables to internal tables. The data came from flat data files.

This test scenario shows the throughput scalability when data was loaded from external tables located on the Oracle ACFS file system into the database.

Test methodology

This test shows the throughput of the XtremIO array, with the disk storage configuration used in this solution, while the sessions ran the data loading on the four-node RAC database. Each session ran a similar ETL workload by loading CSV flat files into the database.

Test procedure

This test demonstrated the performance of the X-Bricks by loading data from external tables. The test included running one user on each node of the four-node RAC database to load data from one external table, which equaled four sessions loading data simultaneously.

Each session loaded one CSV file with a size of 120 GB. The CSV file was located on the Oracle ACFS file system. The external table was created as follows:

```
create table sales_ext (
id integer,
…)
organization external(
type oracle_loader
default directory EXT_DIR
access parameters (fields terminated by "|")
location ('sales.csv'))
parallel reject limit unlimited;
```

The data was loaded from the external table as follows:

```
alter session enable parallel dml;
alter table sales parallel;
alter table sales_ext parallel;
insert into /*+ append */ sales select * from sales_ext;
```

Note: The **sales** table has the same structure as the **sales_ext** table. The data is loaded directly with the **append** hint, and multiple parallel slaves are used for data loading.

Test results

To assess the throughput (TB/hour), we used **physical write bytes** in the AWR report.

During testing, the throughput reached 3.59 TB/hour when we ran the data-loading workload on the four-node RAC database.

Test/dev copy creation test

In this scenario, we used the SCSI EXTENDED COPY (XCOPY) to copy data in the LUNs of the production database to other newly created LUNs that were used for test/dev environment provisioning. The XCOPY utility is a small tool developed by EMC to send read/write SCSI commands from the host to the XtremIO storage array.

The XtremIO array automatically deduplicated data as the database was copied. This reduced the amount of data written to flash, which extended the flash lifetime. XtremIO data reduction did not negatively affect performance of the array at any time and was always inline.

Data deduplication occurs globally across the array. The test methodology featured is just one option that can be used to create replicas of a primary database on XtremIO without increasing space utilization. Other methods include:

- OS/ASM/RMAN copy and restore
- RMAN backup and restore
- Oracle Enterprise Manager (cloning)
- Data Guard setup
- vCenter server cloning for VMDKs on VMFS

There is no performance difference between accessing the primary volumes versus accessing the cloned volumes. The metadata resolves to the same set of actual physical data blocks. This is preferred in certain situations when performance validation is required against the replica. However, it might create unwanted results if the development environment is allowed to take a larger share of the performance capacity of the array.

To keep various entities separate, you can employ a variety of techniques, such as Linux Control Groups (Cgroups). Host quality of service (QoS) can be used when you want to set IOPS, bandwidth (reads, writes, or both), and CPU limits to mitigate unwanted increased performance for the development host.

Test methodology

This test was used to measure the deduplication ratios of the XtremIO array when more than one clone of the production database was created for test/dev environment provisioning. We used the physical capacity and thin provisioning space savings of the array to verify the efficiency of the XtremIO deduplication feature.

Test procedure

We used the following steps for the deduplication test:

1. Before copying the production database, captured the current deduplication ratios, used physical capacity, and thin provisioning space savings as a baseline.
2. Created the first clone (to create a copy of the database for test/dev purposes) by cloning seven 750 GB data volumes in parallel using XCOPY, that is, copied about 6 TB of data (about 4 TB of database data). Captured the deduplication ratios, used physical capacity, and thin provisioning space savings on the XtremIO array management console. After we created the clone, it was mounted on other servers and operated for test or development purposes.
3. Created the second test/dev database by cloning the production database using XCOPY, then captured the deduplication ratios, used physical capacity, and thin provisioning space savings on the XtremIO array management console.

Test results

Figure 13 shows that as more clones of the production database were created, the deduplication ratios increased steadily, while the increase of the array’s used physical capacity was negligible.

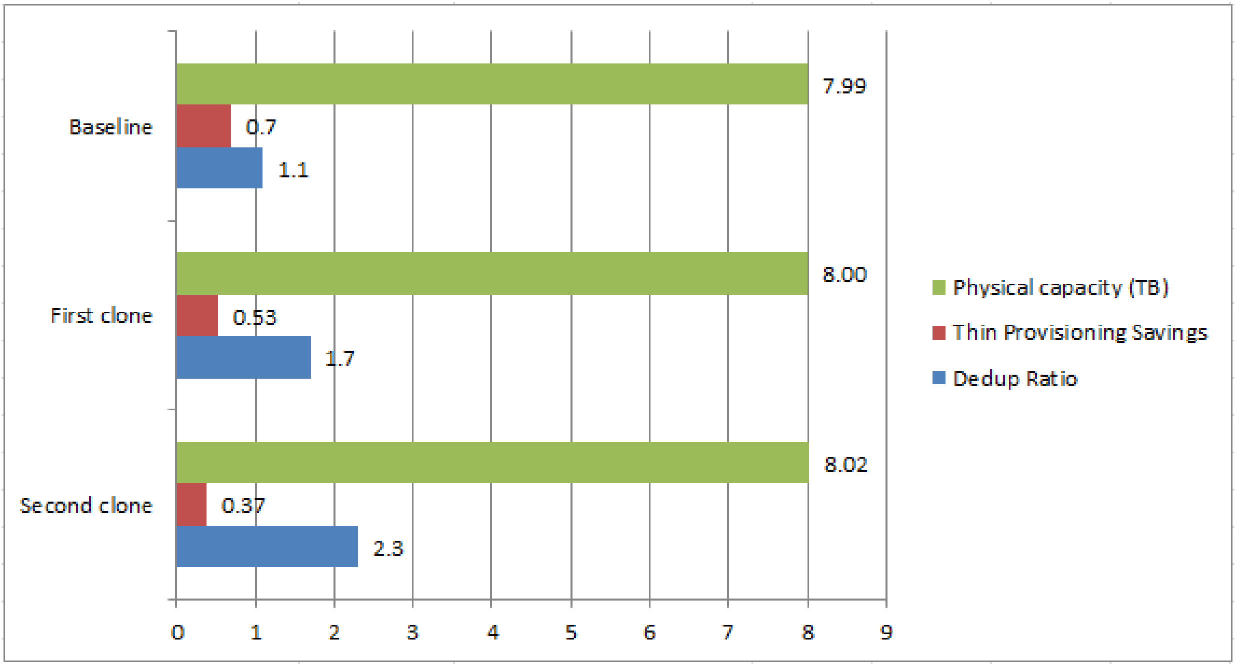

Figure 13. XtremIO deduplication efficiency

XtremIO has a highly efficient implementation of XCOPY integrated with in-memory metadata and inline data reduction. Each time the database is cloned, there is no new space occupied on the SSDs.

Summary of validation and testing

Databases deployed on XtremIO achieve high performance regardless of workload type or configuration. Both OLTP and DW workloads perform well and scale consistently. No tuning or database refactoring is required.

Unlike traditional and hybrid arrays, virtualized and consolidated databases, and the associated I/O blender effect, do not have any inherent disadvantage on XtremIO compared to non-virtualized databases running on physical bare-metal servers. That consolidation and adaptability to change protects your capital investment, and can be flexible as your requirements change and databases grow, without sacrificing agile data center operations. It is simple to linearly and predictably scale the performance of databases on XtremIO, as shown in the following results:

- Transactions per minute (TPM) for OLTP workloads increased when servers were added.
- Average query throughput for DW workloads increased when servers were added.
- Data load speeds increased when additional database servers were added.
- With inline deduplication, the physical capacity increase caused by database cloning was negligible.

Conclusion

Summary

The Oracle Database 11*g* with EMC XtremIO and VMware vSphere solution offers true scalability for consolidated and virtualized Oracle instances across multiple workloads. This solution can be used as the foundation for a system that can be scaled in a flexible, predictable, and near-linear way.

Storage can keep up with linear scaling on the host side. For storage, XtremIO N-way active/active scale-out architecture linearly scales capacity, increases IOPS, and maintains superior latency. For the servers, scaling comes when you add additional compute resources including CPUs, memory, HBA ports, and front-end ports, to provide higher IOPS and throughput for OLTP and DW environments.

This solution enables you to achieve efficient resource utilization through virtualization while providing database and application consolidation. You can also independently scale capacity and processing capability without the limitations imposed by a single-purpose appliance. As you undergo changes from any level, such as applications, database software, and non-database software, the solution stack can align with the shifting demands imposed by your business needs. In turn, new workload approaches such as realtime analytics are now possible, with the consolidation of production and reporting instances.

Findings

This solution provides:

- Fast and simple XtremIO setup with little or no storage tuning required
- High-bandwidth DW with multiple concurrent users and large-block random workload
- Higher performance and flexibility with reduced latency and increased IOPS
- Virtualized clustered databases with high I/Os, such as Oracle RAC
- Substantial storage footprint savings by using XtremIO inline deduplication
- High fault tolerance/protection for highly available clustered systems such as Oracle RAC and vSphere HA.

References

White paper

For additional information, refer to the following EMC documents. Access to these documents depends on your login credentials. If you do not have access to a document, contact your EMC representative.

- *EMC Infrastructure for High Performance Microsoft and Oracle Database Systems*
- *Introduction to the EMC XtremIO Storage Array*

VMware documentation

For additional information, refer to the following VMware documents.

- *Windows XP Deployment Guide*
- *Oracle Databases on VMware Best Practices Guide*

Oracle documentation

For additional information, refer to the *Oracle Database Client Installation Guide 11g Release 2 (11.2) for Linux*.