IBM DS8000 Performance Configuration Guidelines for Implementing Oracle Databases with Automatic Storage Management (ASM)

Bob Gonzalez
IBM Systems and Technology Group
Open Systems Lab, San Jose, California
October 2008

Version Dec 19, 2008

Table of contents

1 Abstract 1

2 Introduction 1

2.1 Outline 1

2.2 Assumptions 2

2.3 Intended Audience 2

3 Storage Area Network Technology Stack 2

3.1 IBM System Storage DS8000 Hardware Overview 2

3.1.1 Frames 3

3.1.2 Processor Complexes and Servers 4

3.1.3 I/O Cache 4

3.1.4 I/O Enclosures 4

3.1.5 Disk Subsystem 4

3.1.6 Device Adapters 5

3.1.7 Disk enclosures 5

3.1.8 Disk drives 5

3.1.9 Host adapters 5

3.2 Fibre Channel switch overview 7

3.2.1 Hardware 7

3.2.2 Software – Fabric Operating System 7

3.3 Host and HBA (Host Bus Adapter) hardware and software overview 7

4 DS8000 Virtualization Hierarchy 8

4.1 Hierarchical View of the DS8000 Virtualization Architecture 8

4.1.1 Array sites 8

4.1.2 Arrays 8

4.1.3 Ranks 8

4.1.4 Extent Pools 9

4.1.5 Logical Volumes (LUNs) 9

4.1.6 Logical Subsystems (LSS) 9

4.1.7 Address Groups 10

4.1.8 Volume Groups 10

5 Oracle Automatic Storage Management (ASM) 11

5.1 ASM Disk Groups 11

5.2 ASM Disks 12

5.3 ASM Striping 12

5.4 ASM Instances 13

6 I/O Workloads 14

6.1 General I/O Workload Types and Associated Metrics 14

6.1.1 Small Block, random I/O workloads 14

6.1.2 Large Block, sequential I/O workloads 14

6.2 Oracle I/O Workloads 14

6.2.1 Oracle random I/O workloads 14

6.2.2 Oracle sequential I/O workloads 15

6.2.3 Determining the current Oracle I/O profile 15

7 DS8000 Disk and RAID Array Performance Characteristics 15

7.1 DS8000 Disk and RAID Array I/O performance numbers 15
8 DS8000 Configuration Best Practices 16
8.1 Summary of DS8000 Configuration Best Practices 16
8.1.1 General Principles – Workload Isolation, Workload Resource Sharing and Workload Spreading 16
8.1.2 Zoning and paths from host to DS8000 storage 17
8.1.3 RAID level 18
8.1.4 Extent Pool and Volume (LUN) configuration 19
9 Monitoring and Modeling tools 19
9.1 TotalStorage Productivity Center (TPC) V 3.3.2 20
9.2 Disk Magic 20
10 Orion - Oracle I/O Numbers Calibration tool 21
10.1 Orion tool overview 21
10.2 Orion input parameters 22
11 Lab Setup 24
11.1.1 System Storage DS8000 24
11.1.2 Brocade switches 25
11.1.3 Host nodes 25
12 Testing Methodology 25
12.1 Orion test run descriptions 26
12.1.1 OLTP only 26
12.1.2 Data Warehouse only (large sequential reads and writes) 26
12.1.3 Mixed OLTP and Data Warehouse 26
12.2 Storage configuration variations for Orion test runs 27
12.3 Metrics collected for each Orion test run 28
12.4 Analysis of metrics collected for each Orion test run 28
13 Benchmarking Exercise – Comparison of RAID-5 and RAID-10 29
13.1 Prerequisites 29
13.2 Summary and Analysis 30
13.2.1 Performance Assertion #1 – RAID-5 and RAID-10 perform equally for reads 30
13.2.2 Performance Assertion #2 – RAID-5 performs better for sequential writes 32
13.2.3 Performance Assertion #3 – RAID-10 performs better for random writes 34
14 Benchmarking Exercise – Comparisons of LUN distributions across arrays 39
14.1 Prerequisites 39
14.2 Summary and Analysis 41
14.2.1 OLTP Workloads 41
14.2.2 Data Warehouse (Sequential) Workloads 45
14.2.3 Mixed OLTP and Data Warehouse Workloads 49
15 Summary 52
16 Appendix A: Oracle I/O profile scripts 53
17 Appendix B: References 57
18 Trademarks and special notices 59

1 Abstract

This white paper offers guidelines for optimizing storage performance when configuring IBM System Storage DS8000 which is being used to run an Oracle database with Automatic Storage Management (ASM). It documents the DS8000 performance best practices which are detailed in various IBM Redbooks. These configuration best practices are then applied and confirmed via the usage of Oracle's Orion load generation tool.

2 Introduction

This paper discusses some performance configuration guidelines that can be used when deploying IBM® System Storage™ DS8000 for use in an Oracle Automatic Storage Management environment. It attempts to answer some fundamental configuration questions that System Administrators, Storage Administrators, or Oracle DBA's will have when deploying a DS8000 for Oracle Database with ASM.

The load generator used to arrive at the configuration guidelines is Orion - the Oracle I/O Numbers Calibration tool. This tool generates Input/Output (I/O) using the same I/O software stack used by the Oracle server software without having to install the server software and create a database. It can simulate various workload types at different load levels to arrive at performance metrics for I/O's per second (IOPS), Latency (Response Time), and Megabytes per second (throughput). It can also simulate the effect of striping performed by ASM.

2.1 Outline

A general outline of this paper is as follows:

- Overview of all of the layers of the technology stack that are part of a Storage Area Network which includes the DS8000
- Discussion of the system storage virtualization of the DS8000. As defined for the DS8000, virtualization is the abstraction process from the physical disk drives (DDMs) to a logical volume that the hosts and servers see as though it were a physical disk.
- Discussion of the Oracle Automatic Storage Management (ASM) volume manager
- Discussion of general and Oracle-specific I/O workloads
- Coverage of performance characteristics of the DS8000 disk drives and RAID arrays
- Coverage of the usage of monitoring and modeling tools, IBM's TotalStorage® Productivity Center® and IntelliMagic BV's Disk Magic
- Discussion of Orion, the Oracle I/O Numbers Calibration tool, which is used to generate I/O load and gather storage performance metrics
- Description of the testing methodology
- Consolidation and summarization of the extensive information that is available regarding best practices for DS8000 configuration

- Execution of two benchmarking exercises using Orion. One exercise compares the performance of a RAID-5 configuration versus a RAID-10 configuration. The other exercise compares the performance differences resulting from different LUN configurations across RAID arrays.

2.2 Assumptions

This paper will start with the assumption that the best practices that have been documented by the DS8000 performance experts are relevant for a wide range of applications, including Oracle databases using ASM. The performance testing done for the DS8000 assumes that one of the most common usages for storage is to run databases such as Oracle and DB2. Therefore benchmark testing always includes the types of general, and indeed Oracle-like, workloads that will be discussed in the paper. However, this will not skew the results of the testing done in this paper. The performance numbers which result from the test matrix will definitely have to speak for themselves.

Another assumption is that the load generation tool which will be used, Oracle's Orion, is an appropriate tool for doing the type of testing that is going to be done to obtain performance numbers. A detailed description of Orion will be given in a later section. But briefly, Orion generates I/O using the same I/O software stack used by the Oracle server software without having to install the server software. By using Orion, it is possible to avoid addressing the subject of Oracle database server tuning, a subject which is beyond the scope of this paper. We can then conduct a more controlled experiment where it is assumed that no matter how it would be done by an Oracle database, we are going to be generating Oracle I/O at various load levels and with varying workload mixes.

2.3 Intended Audience

The intended audience of this paper is any Technical Lead, System Administrator, Storage Administrator, or Oracle DBA in a production environment that is part of an effort to deploy IBM DS8000 storage for running Oracle with ASM. After reading this paper, the technical staff will have at least some starting point, based on documented and tested best practices, with which to make configuration and deployment decisions for the DS8000.

3 Storage Area Network Technology Stack

It is important to understand all the layers of the storage technology stack in a SAN environment before any configuration decisions can be made. In this section, you will find overviews for the concepts related to DS8000 storage attributes.

3.1 IBM System Storage DS8000 Hardware Overview

The IBM System Storage DS8000 series is a high-performance, reliable, and exceptionally scalable enterprise disk storage system. The IBM System Storage DS8000 series is designed to:

- Deliver robust, flexible, and cost-effective disk storage for the mission-critical workloads of medium and large enterprises
- Enable the creation of multiple Storage System Logical Partitions (LPARs) within a single DS8000 Model 9B2, that can be used for completely separate production, test, or other unique storage environments

- Support high availability, storage sharing, and consolidation for a wide variety of operating systems and mixed server environments
- Help increase storage administration productivity with centralized and simplified management

The following are descriptions of the hardware contained within the DS8000:

3.1.1 Frames

The DS8000 is designed for modular expansion. From a high-level view, there appear to be three types of frames available for the DS8000. However, on closer inspection, the frames themselves are almost identical. The only variations are the combinations of processors, I/O enclosures, batteries, and disks that the frames contain.

Figure 1 is an attempt to show some of the frame variations that are possible with the DS8000 series. The left frame is a base frame that contains the processors (System p POWER5+ servers). The center frame is an expansion frame that contains additional I/O enclosures but no additional processors. The right frame is an expansion frame that contains just disks (and no processors, I/O enclosures, or batteries). Each frame contains a frame power area with power supplies and other power-related hardware.

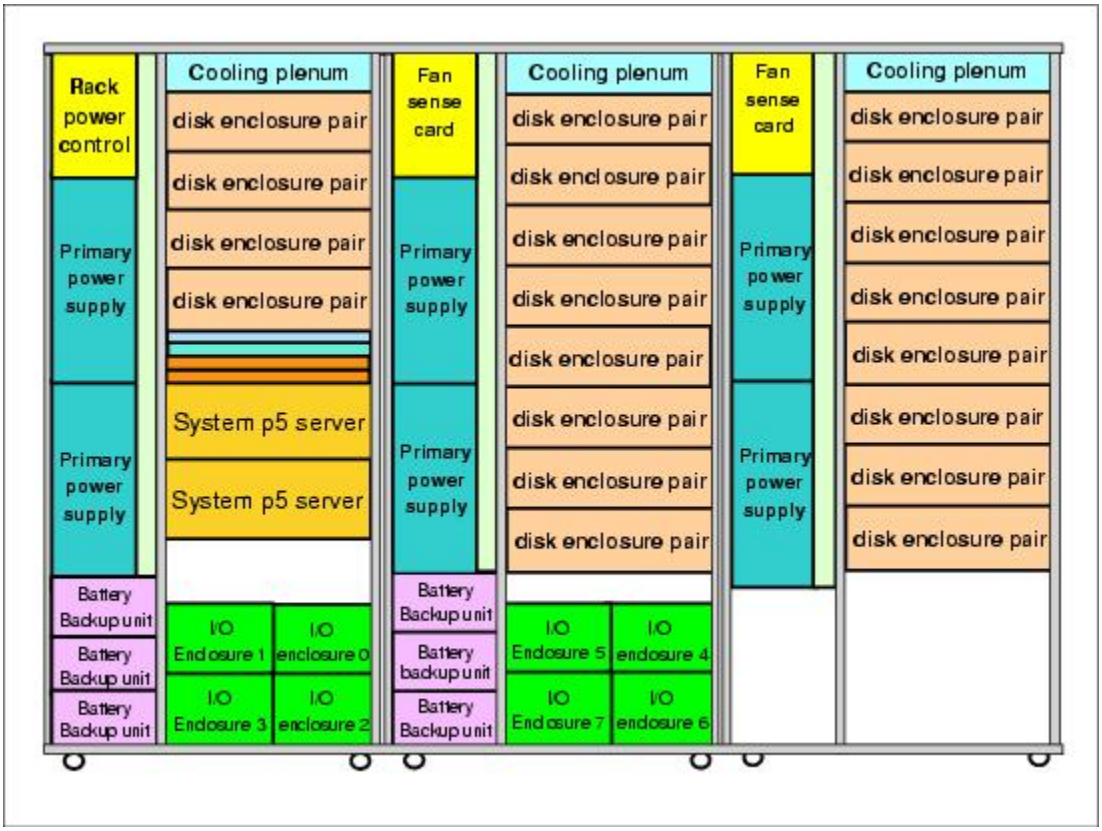


Figure 1 – DS8000 frame possibilities

3.1.2 Processor Complexes and Servers

The DS8000 consists of two processor complexes. Each processor complex has access to multiple host adapters to connect to Fibre Channel, FICON, and ESCON hosts. Attached hosts interact with software which is running on the complexes to access data on logical volumes. Each complex will host at least one instance of this software (which is called a server*),* which runs in a logical partition (an LPAR).

These DS8000 servers contain N-way symmetric multiprocessor (SMP) System p POWER5+ processors. They manage all read and write requests to the logical volumes on the disk arrays.

3.1.3 I/O Cache

The DS8000 Server SMP memory is used as the I/O cache. Like other modern cache, the DS8000 cache contains volatile memory used as a read cache and non-volatile memory used as a write cache. The design decision to use SMP memory as I/O cache is a key element of IBM storage architecture. Although a separate I/O cache could provide fast access, it cannot match the access speed of the SMP main memory.

The DS8100 models 9*x*1 offer up to 128 GB of processor memory and the DS8300 models 9*x*2 offer up to 256 GB of processor memory. Half of this will be located in each processor complex. In addition, the Nonvolatile Storage (NVS) scales to the processor memory size selected, which can also help optimize performance.

To help achieve dramatically greater throughput and faster response times, the DS8000 uses Sequential-prefetching in Adaptive Replacement Cache (SARC). SARC is an efficient adaptive algorithm for managing read caches. The decision of when and what to prefetch is made in accordance with the Adaptive Multi-stream Prefetching (AMP), a new cache management algorithm available with License Machine Code v5.2.400.327 and above.

3.1.4 I/O Enclosures

All base models contain I/O enclosures and adapters. The I/O enclosures hold the adapters and provide connectivity between the adapters and the processors. Device adapters and host adapters are installed in the I/O enclosure. Each I/O enclosure has 6 slots. Each slot supports PCI-X adapters running at 64-bit, 133 MHz. Slots 3 and 6 are used for the device adapters. The remaining slots are available to install up to four host adapters per I/O enclosure.

3.1.5 Disk Subsystem

The disk subsystem consists of three components:

1) First, located in the I/O enclosures are the device adapters. These are RAID controllers that are used by the storage images to access the RAID arrays.

2) Second, the device adapters connect to switched controller cards in the disk enclosures. This creates a switched Fibre Channel disk network.

3) Finally, the disks themselves. The disks are commonly referred to as disk drive modules (DDMs).

3.1.6 Device Adapters

Each DS8000 device adapter (DA) card offers four 2 Gbps FC-AL ports. These ports are used to connect the processor complexes to the disk enclosures. The adapter is responsible for managing, monitoring, and rebuilding the RAID arrays. The adapter provides remarkable performance thanks to a new high function/high performance ASIC. To ensure maximum data integrity, it supports metadata creation and checking.

The DAs are installed in pairs because each storage partition requires its own adapter to connect to each disk enclosure for redundancy. This is why they are referred to as *DA pairs*.

3.1.7 Disk enclosures

Each DS8000 frame contains either 8 or 16 disk enclosures depending on whether it is a base or expansion frame. Half of the disk enclosures are accessed from the front of the frame, and half from the rear. Each DS8000 disk enclosure contains a total of 16 DDMs or dummy carriers. A dummy carrier looks very similar to a DDM in appearance but contains no electronics.

3.1.8 Disk drives

Each disk drive, commonly referred to as a disk drive module (DDM) is hot-pluggable and has two indicators. The green indicator shows disk activity while the amber indicator is used with light path diagnostics to allow for easy identification and replacement of a failed DDM.

The DS8000 allows the choice of these different Fibre Channel DDM types:

- 73 GB, 15k RPM drive
- 146 GB, 15k RPM drive (10k RPM available in older DS8000 models)
- 300 GB, 15k RPM drive (10k RPM available in older DS8000 models)
- 450 GB, 15k RPM drive (announced in September 2008 as this paper was being completed)

and one Fibre Channel Advanced Technology Attachment (FATA) DDM type:

- 500 GB, 7,200 RPM drive

3.1.9 Host adapters

The DS8000 supports two types of host adapters: ESCON and Fibre Channel/FICON. It does not support SCSI adapters. The ESCON adapter in the DS8000 is a dual-ported host adapter for connection to older System z hosts that do not support FICON. The description below will just pertain to Fibre Channel/FICON host adapters.

Each DS8000 Fibre Channel card offers four Fibre Channel ports (port speed of 2 or 4 Gbps depending on the host adapter). The cable connector required to attach to this card is an LC type. Each 2 Gbps port independently auto-negotiates to either 2 or 1 Gbps and the 4 Gbps ports to 4 or 2 Gbps link speed. Each of the 4 ports on one DS8000 adapter can also independently be either Fibre Channel protocol (FCP) or FICON, though the ports are initially defined as switched point-to-point FCP. Selected ports will be configured to FICON automatically based on the definition of a FICON host. Each port can be either FICON or Fibre Channel Protocol (FCP). The personality of the port is

changeable through the DS Storage Manager GUI. A port cannot be both FICON and FCP simultaneously, but it can be changed as required.

The card itself is PCI-X 64 Bit 133 MHz. The card is driven by a new high function, high performance ASIC. To ensure maximum data integrity, it supports metadata creation and checking. Each Fibre Channel port supports a maximum of 509 host login IDs and 1,280 paths.

Figure 2 is a depiction of all of the hardware components in the DS8300.

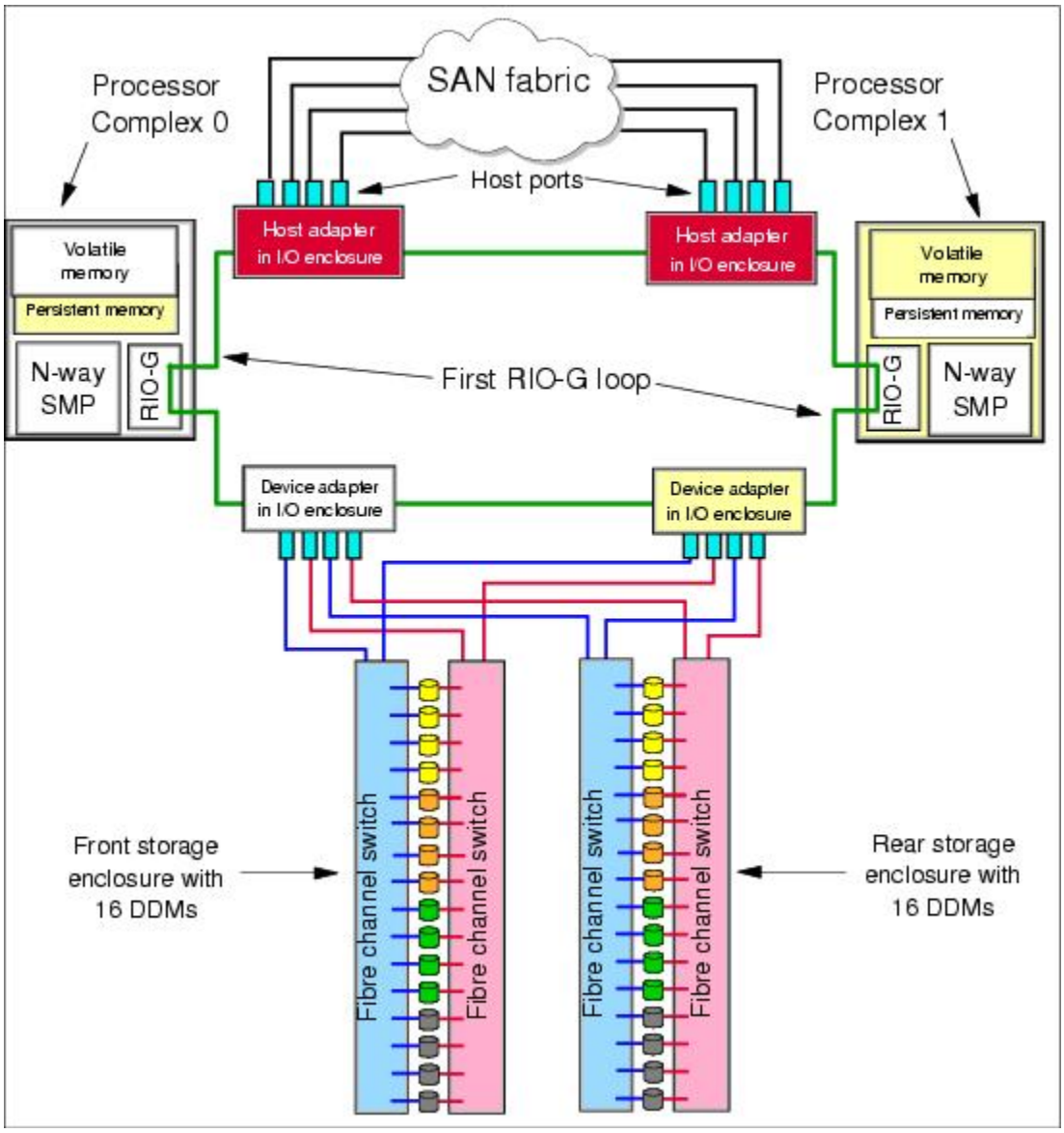


Figure 2: DS8300 Hardware Architecture

3.2 Fibre Channel switch overview

The Fibre Channel switches used in the lab environment are the IBM System Storage SAN32B-3. This is IBM machine type 2005, model number B5K. The Brocade name is Brocade 5000. The following are brief descriptions of both the hardware and software in the IBM Brocade 2005-B5K.

3.2.1 Hardware

The IBM System Storage SAN32B-3 is a high performance midrange fabric switch that provides 16, 24, and 32-port, 4 Gbps fabrics switching for Windows NT/2000 and UNIX server clustering, infrastructure simplification and business continuity solutions. The base switch offers Advanced Zoning, Full Fabric License, Fabric Watch, WebTools, NPIV software, dual replaceable power supplies and 16-ports activated. The Ports on Demand features support “pay-as-you-grow” scalability in 8 port increments. B32 optional features include Advanced Inter-Switch Link (ISL) Trunking, Advanced Performance Monitoring, Extended Fabric Activation, and Advanced Security Activation.

With a flexible architecture, the SAN32B-3 support native E_Port connectivity into Brocade and McDATA products. The SAN32B-3 can operate in FOS native, FOS open, M-EOS native and M-EOS open modes without routing and special partitioning. This enables non-disruptive expansion of any Brocade or McDATA fabric based on these operation modes, and provides the ability to implement non-disruptive software upgrades. A special NI code has to be downloaded to set the SAN32B-3 in any McDATA mode.

3.2.2 Software – Fabric Operating System

The Fabric Operating System (FOS) manages the operation of the switch and delivers the same, and compatible, functionality to all the different models of switches and directors. The switch firmware is designed to make the switches easy to install and use, while retaining the flexibility required to accommodate user requirements.

The FOS includes all the basic switch and fabric support software as well as optionally licensed software that is enabled using license keys. It is composed of two major software components: firmware that initializes and manages the switch hardware, and diagnostics. Fabric OS (FOS) Version 5.x is a Linux-based operating system.

3.3 Host and HBA (Host Bus Adapter) hardware and software overview

The discussion in this paper pertains strictly to open systems (Linux/UNIX) hosts. The best practices for configuring the zoning and pathing from the host to storage will be discussed in the section **8.1.2 Zoning and paths from host to DS8000 storage**. But briefly, here are some of the considerations with regard to the tuning, troubleshooting, and performance of a host attached to the DS8000 storage:

- Verify the optimal version and configuration of the Host Bus Adapter and disk devices. For example, on AIX the **queue depth** and **max transfer size** of either the HBA or hdisks should be set properly.
- Understand the optimal load-balancing and failover configuration of the HBA multipathing software. The multipathing software is Device Mapper, or DM, for Linux. For AIX, it is the Subsystem Device Driver Path Control Module, or SDDPCM.

- Be familiar with the host-side I/O performance monitoring tools. The iostat command is the traditional open systems disk performance monitoring tool. In addition, AIX offers the nmon, filemon, and topas commands for disk I/O monitoring.

4 DS8000 Virtualization Hierarchy

This section discusses the DS8000 virtualization hierarchy. Virtualization can be thought of as the process of preparing a bunch of physical disk drives (DDMs) to something that can be used from an operating system, which basically means the creation of LUNs. It is the abstraction from the physical layer to the logical layer in the DS8000 architecture.

4.1 Hierarchical View of the DS8000 Virtualization Architecture

4.1.1 Array sites

An array site is a group of eight DDMs. What DDMs make up an array site is predetermined by the DS8000, but note that there is no predetermined server affinity for array sites. The DDMs selected for an array site are chosen from two disk enclosures on different loops. The DDMs in the array site are of the same DDM type, which means the same capacity and the same speed (rpm).

4.1.2 Arrays

An array is created from one array site. Forming an array means defining it as a specific RAID type. The supported RAID types are RAID-5, RAID-10 and RAID-6 (RAID-6 was announced in September 2008 as this paper was being completed; it will not be discussed in detail in this paper). For each array site, you can select a RAID type. The process of selecting the RAID type for an array is also called defining an array. According to the DS8000 sparing algorithm, from zero to two spares can be taken from the array site.

4.1.3 Ranks

In the DS8000 virtualization hierarchy, there is another logical construct, a rank. When defining a new rank, its name is chosen by the DS Storage Manager, for example, R1, R2, or R3, and so on. You have to add an array to a rank. In the DS8000 implementation, a rank is built using just one array. The available space on each rank will be divided into extents. The extents are the building blocks of the logical volumes.

The process of forming a rank does two things:

- The array is formatted for either fixed block (FB) data (open systems) or count key data (CKD) (System z) data. This determines the size of the set of data contained on one disk within a stripe on the array.
- The capacity of the array is subdivided into equal-sized partitions, called *extents.* The extent size depends on the extent type, FB or CKD.

An FB rank has an extent size of 1 GB (where 1 GB equals 2^{30} Bytes).

4.1.4 Extent Pools

An Extent Pool is a logical construct to aggregate the extents from a set of ranks to form a domain for extent allocation to a logical volume. Typically the set of ranks in the Extent Pool should have the same RAID type and the same disk RPM characteristics so that the extents in the Extent Pool have homogeneous characteristics.

There is no predefined affinity of ranks or arrays to a storage server. The affinity of the rank (and its associated array) to a given server is determined at the point it is assigned to an Extent Pool. One or more ranks with the same extent type (FB or CKD) can be assigned to an Extent Pool. One rank can be assigned to only one Extent Pool. There can be as many Extent Pools as there are ranks.

Storage Pool Striping was made available with License Machine Code 5.30xx.xx and allows you to create logical volumes striped across multiple ranks. This will typically enhance performance.

4.1.5 Logical Volumes (LUNs)

A logical volume is composed of a set of extents from one extent pool. On a DS8000, up to 65,280 (65,536 - 256) volumes can be created.

The DS8000 can have Fixed Block volumes, CKD volumes or System i LUNs. Only Fixed Block volumes are used in an open systems environment, so the discussion from now on will be limited to that type of volume.

A logical volume composed of fixed block extents is called a LUN. A fixed block LUN is composed of one or more 1 GB (2^{30} Bytes) extents from one FB extent pool. A LUN cannot span multiple extent pools, but a LUN can have extents from different ranks within the same extent pool. You can construct LUNs up to a size of 2 TB (2^{40} Bytes).

LUNs can be allocated in binary GB (2^{30} Bytes), decimal GB (10^{9} Bytes), or 512 or 520 Byte blocks. However, the physical capacity that is allocated for a LUN is always a multiple of 1 GB, so it is a good idea to have LUN sizes that are a multiple of a GigaByte. If you define a LUN with a LUN size that is not a multiple of 1 GB, for example, 25.5 GB, the LUN size is 25.5 GB, but 26 GB are physically allocated and 0.5 GB of the physical storage is unusable.

4.1.6 Logical Subsystems (LSS)

A logical subsystem (LSS) is another logical construct. It groups LUNs in groups of up to 256 logical volumes. On the DS8000, there is no fixed binding between any rank and any logical subsystem. The capacity of one or more ranks can be aggregated into an extent pool and logical volumes configured in that extent pool are not bound to any specific rank. Different logical volumes on the same logical subsystem can be configured in different extent pools. As such, the available capacity of the storage facility can be flexibly allocated across the set of defined logical subsystems and logical volumes.

For each LUN, you can choose an LSS. As mentioned above, you can put up to 256 volumes into one LSS. There is, however, one restriction. Volumes are formed from a bunch of extents from an extent pool. Extent pools, however, belong to one server, server 0 or server 1, respectively. LSSs also have an affinity to the servers. All even-numbered LSSs (X'00', X'02', X'04', up to X'FE') belong to server 0 and all odd-numbered LSSs (X'01', X'03', X'05', up to X'FD') belong to server 1. LSS X'FF' is reserved.

For open systems, LSSs do not play an important role except in determining by which server the LUN is managed (and in which extent pools it must be allocated) and in certain aspects related to Copy Services. Some management actions in Copy Services operate at the LSS level. For example, the freezing of FlashCopy pairs to preserve data consistency across all pairs is done at the LSS level.

4.1.7 Address Groups

Address groups are created automatically when the first LSS associated with the address group is created, and deleted automatically when the last LSS in the address group is deleted. LSSs are either CKD LSSs or Fixed Block LSSs. All devices in an LSS must be either CKD or FB. This restriction goes even further. LSSs are grouped into address groups of 16 LSSs. LSSs are numbered X'*ab*', where *a* is the address group and *b* denotes an LSS within the address group. So, for example, X'10' to X'1F' are LSSs in address group 1.

4.1.8 Volume Groups

A volume group is a named construct that defines a set of logical volumes. When used in conjunction with Open Systems hosts, a host attachment object that identifies the HBA is linked to a specific volume group. You must define the volume group by indicating which fixed block logical volumes are to be placed in the volume group. Logical volumes can be added to or removed from any volume group dynamically. (The DS8000 Volume Groups should not be confused with the Volume Groups on AIX Logical Volume Manager).

FB logical volumes can be defined in one or more volume groups. This allows a LUN to be shared by host HBAs configured to different volume groups. An FB logical volume is automatically removed from all volume groups when it is deleted.

Figure 3 shows the DS8000 virtualization hierarchy.

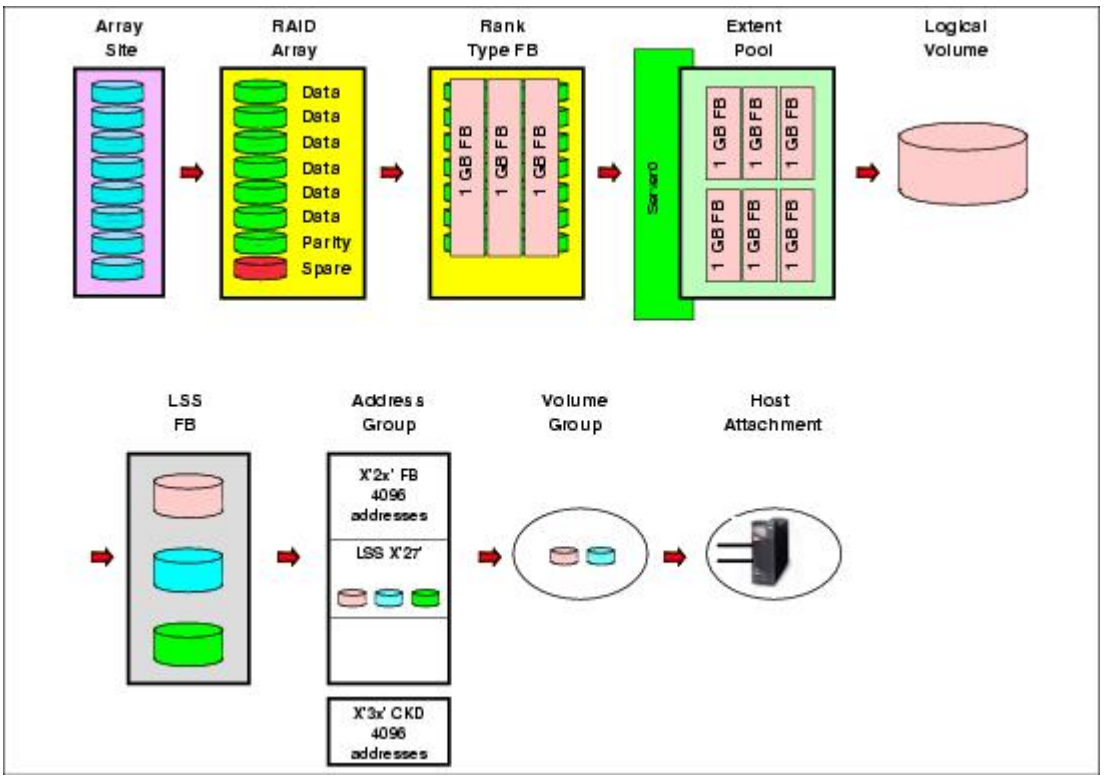


Figure 3 – DS8000 Virtualization Hierarchy

5 Oracle Automatic Storage Management (ASM)

Oracle Automatic Storage Management (ASM) is a volume manager and a filesystem for Oracle database files that supports single-instance Oracle Database and Oracle Real Application Clusters (Oracle RAC) configurations. Oracle ASM is the Oracle recommended storage-management solution that provides an alternative to conventional volume managers, file systems, and raw devices. ASM provides the performance of raw I/O with the management capabilities of a filesystem.

5.1 ASM Disk Groups

Oracle ASM uses disk groups to store data files. An Oracle ASM disk group is a collection of disks that Oracle ASM manages as a unit. Within a disk group, Oracle ASM exposes a file-system interface for Oracle database files. The content of files that are stored in a disk group are evenly distributed, or striped, to prevent hot spots and to provide uniform performance across the disks. The performance is comparable to the performance of raw devices.

You can add or remove disks from a disk group while a database continues to access files from the disk group. When you add or remove disks from a disk group, ASM automatically redistributes the file contents and eliminates the need for downtime when redistributing the content.

When you create a disk group, you specify an ASM disk group type based on one of the following three redundancy levels:

- **Normal** for 2-way mirroring
- **High** for 3-way mirroring
- **External** to not use ASM mirroring, such as when you configure hardware RAID for redundancy

The disk group type determines the mirroring levels with which Oracle creates files in a disk group. The redundancy level controls how many disk failures are tolerated without dismounting the disk group or losing data. This paper assumes that **External Redundancy** is used.

5.2 ASM Disks

ASM disks are the storage devices that are provisioned to ASM disk groups. Examples of ASM disks include:

- A disk or partition from a storage array
- An entire disk or the partitions of a disk
- Logical volumes
- Network-attached files (NFS)

When you add a disk to a disk group, you either assign a disk name or the disk is given an ASM disk name automatically. This name is different from the name used by the operating system. In a cluster, a disk may be assigned different operating system device names on different nodes, but the disk has the same ASM disk name on all of the nodes. In a cluster, an ASM disk must be accessible from all of the instances that share the disk group.

If the disks are the same size, then ASM spreads the files evenly across all of the disks in the disk group. This allocation pattern maintains every disk at the same capacity level and ensures that all of the disks in a disk group have the same I/O load. Because ASM load balances among all of the disks in a disk group, different ASM disks should not share the same physical drive.

Every ASM disk is divided into allocation units (AU). An AU is the fundamental unit of allocation within a disk group. A file extent consists of one or more AU. An ASM file consists of one or more file extents.

When you create a disk group in Oracle Database 11g, you can set the ASM AU size to be between 1 MB and 64 MB in powers of two, such as, 1, 2, 4, 8, 16, 32, or 64. Larger AU sizes typically provide performance advantages for data warehouse applications that use large sequential reads. Oracle Database 10g AUs are 1 MB, although this can be changed by modifying some Oracle hidden initialization parameters.

5.3 ASM Striping

ASM striping has two primary purposes:

- To balance loads across all of the disks in a disk group

- To reduce I/O latency

Coarse-grained striping provides load balancing for disk groups while fine-grained striping reduces latency for certain file types by spreading the load more widely.

To stripe data, ASM separates files into stripes and spreads data evenly across all of the disks in a disk group. The stripes are equal in size to the effective AU. The coarse-grained stripe size is always equal to the AU size. The fine-grained stripe size always equals 128 kB; this provides lower I/O latency for small I/O operations such as redo log writes.

The ASM stripe size is automatically set when a particular type of Oracle file is created since the stripe size is defined in the Oracle ASM file templates. ASM file templates exist for datafiles, online redo logs, archive log files, control files, tempfiles, and parameter files. The Oracle documentation mentioned in the References contains a complete list of file types that are supported.

5.4 ASM Instances

An ASM instance is built on the same technology as an Oracle Database instance. An ASM instance has a System Global Area (SGA) and background processes that are similar to those of Oracle Database. However, because ASM performs fewer tasks than a database does, an ASM SGA is much smaller than a database SGA. In addition, ASM has a minimal performance effect on a server. ASM instances mount disk groups to make ASM files available to database instances; ASM instances do not mount databases. The ASM instance executes only a small portion of the code in the Oracle kernel, thus it is less likely to encounter failures or contention.

The ASM instance creates an extent map which has a pointer to where each 1MB extent of the data file is located. When a database (RDBMS) instance creates or opens a database file that is managed by ASM, the database instance messages the ASM instance and ASM returns an extent map for that file. From that point, the RDBMS instance performs all I/O directly to the disks unless the location of that file is being changed. Therefore, during normal operation the ASM instance is not in the I/O path. The three things that might cause the extent map for a database instance to be updated are:

1) Rebalancing the disk layout following a storage configuration change (adding or dropping a disk from a disk group)
2) Opening of a new database file
3) Extending an existing database file when a tablespace is enlarged

ASM metadata is the information that ASM uses to control a disk group and the metadata resides within the disk group. The RDBMS instances never directly update ASM metadata. ASM metadata is written only by the ASM instance. ASM metadata includes the following information:

- The disks that belong to a disk group
- The amount of space that is available in a disk group
- The filenames of the files in a disk group
- The location of disk group datafile data extents

- A redo log that records information about atomically changing data blocks

ASM and database instances require shared access to the disks in a disk group. ASM instances manage the metadata of the disk group and provide file layout information to the database instances.

6 I/O Workloads

This section gives a general overview of I/O workload types and the metrics used to gauge performance for these workloads. It then discusses the specifics of Oracle I/O workloads.

6.1 General I/O Workload Types and Associated Metrics

I/O workloads are typically characterized as belonging to one of the following types:

1) Small block, random I/O's with a relatively high transaction rate: This includes OLTP databases, mail servers, Web servers, and file servers.
2) Large block, sequential I/O's: This includes Data Warehouse databases, video servers, and backup servers.
3) Mixed workloads: This is a combination of the workload types in 1) and 2) above.

6.1.1 Small Block, random I/O workloads

The performance of these OLTP-type workloads is measured in terms of two metrics, I/O's per second (IOPS) and latency (or response time). IOPS are dependent on two characteristics of disk drives, the average seek time and the rotational latency. The response time is the actual service time for a given I/O. Calculations for the IOPS that can be expected from the DS8000 are given in the section **7 DS8000 Disk and RAID Array performance numbers**.

6.1.2 Large Block, sequential I/O workloads

The performance of sequential I/O workloads is measured in throughput or Megabytes per second (MBPS). Response times are generally not important for sequential I/O workloads as long as throughput objectives are met.

6.2 Oracle I/O Workloads

Oracle I/O workloads can be small-block, random I/O's or large-block sequential I/O's:

6.2.1 Oracle random I/O workloads

The I/O size for Oracle random I/O is the database block size, which is set when the database is created. This is usually set to 8 kB for Oracle databases that are going to be used for OLTP applications or for mixed workloads. However, the database block size can be set up to 32 kB. Starting with Oracle Database 9i, it became possible to create tablespaces that have block sizes that differ from the base block size set at database creation time. So the block size can be 8 kB for tablespaces with OLTP data and 16 kB or 32 kB for tablespaces containing the data warehouse data.

6.2.2 Oracle sequential I/O workloads

For sequential I/O, Oracle will create an I/O size that is composed of several database blocks. Oracle will use I/O sizes of up to 1 MB for sequential I/O. This is the default size used for the allocation units (AUs) in Oracle Database 10g and 11g ASM. Allocation units are the fundamental unit of allocation within an ASM disk group. In Oracle Database 11g, you can modify the ASM AU size to be between 1 MB and 64 MB in powers of two, i.e., 1, 2, 4, 8, 16, 32, or 64.

6.2.3 Determining the current Oracle I/O profile

Appendix A contains two SQL*Plus scripts which can be used to determine the I/O profile, with regard to IOPS and throughput (MBPS), of a currently running Oracle database. They can be run on either single-instance or RAC databases.

7 DS8000 Disk and RAID Array Performance Characteristics

This section discusses some basic performance characteristics of the DS8000 DDMs and RAID arrays.

7.1 DS8000 Disk and RAID Array I/O performance numbers

The 15K RPM Fibre Channel disks in the DS8000 provide an average seek time of approximately 3.5 milliseconds. The (rotational) latency is (15000 rotations/minute) = (15000 rotations/60 seconds) = (1 rotation/.004 seconds) or 4 milliseconds. The average latency is arrived at by dividing this number by 2, so the average latency is 2 ms. For transferring only a small block, the transfer time can be neglected. Therefore this is an average 5.5 ms per random disk I/O operation. This calculates out to (1 I/O / .0055 seconds) or about 180 I/O's per second (IOPS). A combined number of 8 disks (as is the case for a DS8000 array) will thus potentially sustain 1,440 IOPS when spinning at 15K rpm. Reduce the number by 12.5% when you assume a spare drive in the 8 pack. This should be reduced by another 12.5% when using a RAID-5 configuration over the 8 DDM pack because the capacity equivalent to one disk will be used for parity.

Back at the host side, consider an example with 1,000 IOPS from the host, a read-to-write ratio of 3 to 1, and 50% read cache hits. This leads to the following IOPS numbers:

- 750 read IOPS
- 375 read I/Os must be read from disk (based on the 50% read cache hit ratio)
- 250 writes with RAID-5 results in 1,000 disk operations due to RAID-5 write penalty (read old data and parity, write new data and parity)
- This totals to 1375 disk I/O's

With 15K RPM DDMs, doing 1000 random IOPS from the server we actually do 1375 I/O operations on disk compared to a maximum of 1440 operations for 7+P configurations or 1260 operations for 6+P+S configurations. Hence 1000 random I/Os from a server with a standard read-to-write ratio and a standard cache hit ratio saturate the disk drives. These numbers are based on the assumption that

server I/O is purely random. When there are sequential I/Os, track-to-track seek times are much lower and higher I/O rates are possible. We also assumed that reads have a hit ratio of only 50%. With higher hit ratios higher workloads are possible. This shows the importance of intelligent caching algorithms as used in the DS8000.

For a single disk drive, various disk vendors provide the disk specifications on their Internet product sites. Since the access times for the Fibre Channel disks (not FATA) are the same, but they have different capacities, the I/O density is different. 146 GB 15k RPM disk drives can be used for access densities up 1 I/O/GBps. For 73 GB drives, the access density can be 2 I/O/GBps and for 300 GB drives, it is 0.5 I/O/GBps. While this discussion is theoretical in approach, it provides a first estimate. It is important when sizing a storage subsystem; you should not only consider the capacity but also the number of disk drives needed to satisfy the performance requirements.

8 DS8000 Configuration Best Practices

This section consolidates and summarizes the best practices gathered from the DS8000 Redbooks cited in the References section. As stated in the introduction, one of the assumptions for this paper is that the best practices documented for the DS8000 have been arrived at by doing testing which included database-type I/O workloads. Therefore it is critical to review these documented best practices and use them as the starting point in any DS8000 deployment for Oracle with ASM. They are the foundation upon which any configuration decisions must be made.

8.1 Summary of DS8000 Configuration Best Practices

This subsection summarizes best practices for DS8000 configuration.

8.1.1 General Principles – Workload Isolation, Workload Resource Sharing, and Workload Spreading

General principles for DS8000 configuration are summarized below.

1) **Workload Isolation** is the dedication of a subset of hardware resources to one workload. The hardware resources can be processor complexes (Server0 and Server1), ranks (disk drives), Device Adapters (DA's), or I/O ports. LUNs and host connections would be isolated to these resources. Workload isolation can prevent less important workloads from impacting more important workloads. Some factors to consider with Workload Isolation:
   (a) Provides increased probability of consistent response time.
   (b) However maximum potential performance is limited to the set of dedicated resources.
   (c) Contention is still possible for resources which are not dedicated, e.g., processors, cache, or persistent memory.
   (d) A good approach if experience or analysis identifies:
      - a workload which tends to consume 100% of resources
      - a workload which is more important than other workloads
      - workloads with conflicting I/O demands
   (e) Disk drive (rank) level isolation may be appropriate for heavy random workloads.

(f) Device Adapter (DA) level isolation may be appropriate for large blocksize, heavy sequential workloads.

2) **Workload Resource Sharing** is the sharing of a common set of resources for more than one workload. The hardware resources can be processor complexes (Server0 and Server1), ranks (disk drives), Device Adapters (DA's) or I/O ports. LUN's and host connections are assigned to the shared set of resources. This provides a higher potential performance by making a larger set of resources available to a given workload. Some factors to consider:

   (a) There can be possible contention for all shared hardware resources, i.e., ranks, I/O ports, DA's, processors, cache, and persistent memory.

   (b) This is a good approach when:

   - there is not enough workload information to identify isolation requirements
   - the workload will not try to consume all of the hardware available
   - the workloads doing the sharing peak at different times

3) **Workload Spreading** is when a given workload is balanced and distributed evenly across all allowed hardware. This applies both to workload isolation and workload resource sharing. LUN's and host connections should be spread across all dedicated resources or shared resources:

   (a) **Logical Volume spreading** entails:

   - Spreading the workload across Ranks
   - Spreading the workload across DA's
   - Spreading the workload across processor complexes (Server0 and Server1)
   - Allocating new LUNs on least-used shared resources
   - Possibly using Host volume striping

   (b) **Exceptions to LUN spreading**:

   - Files or datasets which will never be accessed simultaneously
   - Multiple log files for a single application may be placed on the same rank, i.e., members of "one side" of Oracle online redo log groups.

   (c) **Host connection spreading** entails:

   - Spreading host connections for a workload across I/O ports.
   - Spreading host connections for a workload across Host Adapters.
   - Spreading host connections for a workload across I/O enclosures.
   - Spreading host connections for a workload across Server0 and Server1 (processor complexes).
   - Spreading host connections for a workload across Left side and Right side I/O enclosures (related to spreading across Server0 and Server1).
   - Allocating new host connections on least-used shared resources.
   - Using multipathing software.

8.1.2 Zoning and paths from host to DS8000 storage

This section summarizes best practices for zoning and host-storage paths.

1) There should be at least two but no more than four paths to a LUN. Increasing paths increases availability in case of outages but it also increases the amount of CPU used by the multipathing software choosing among all of the paths for any given I/O. Therefore a good compromise is between two and four paths per LUN.
2) Use the dynamic load balancing option of the Subsystem Device Driver Path Control Module (SDDPCM).
3) Use different storage Host Adapter's for the host port connections.
4) It is best to not use all four ports on the HA's since there is some possibility of overloading the PCI-X bus behind these cards. Use at most three ports from the HA, but it might be better to even just use two ports.
5) Spread the connections from the host among the various I/O enclosures and I/O buses. An expected throughput would be about 250 MBPS per port. Using two ports will scale throughput a bit less than linearly.

8.1.3 RAID level

This section summarizes considerations for choosing the RAID level.

1) DS8000 supports RAID-5, RAID-10, and RAID-6 (RAID-6 became available in September 2008 as this paper was being completed; it will not be discussed in detail in this paper). JBOD is not supported.
2) A RAID-5 array can be 6+P+S (6 data, 1 parity, and 1 spare) or 7+P (7 data and 1 parity).
3) A RAID-10 array can be 3x2+2S (3 data mirrored, 2 spares) or 4x2 (4 data mirrored).
4) The first four array sites each contribute one spare to DA pair. So each DA pair has access to 4 spares.
5) Be wary of rules of thumb regarding performance differences between RAID-5 and RAID-10. Performance depends on many factors, so there can always be exceptions to some of the general guidelines. Either the SQL*Plus script in Appendix A or Disk Magic can be used to determine workload profile. But here are some very general guidelines regarding both performance and availability:
    (a) For reads from disk, either random or sequential, there is no significant difference between RAID-5 and RAID-10.
    (b) For sequential writes, RAID-5 performs better. DS8000 can detect sequential workload and if an entire stripe is in cache, it switches to a RAID3 algorithm to calculate parity and does not have to read the old data and parity from disk before de-staging.
    (c) For random writes, RAID-10 performs better.
    (d) There may be performance differences between RAID Arrays containing some spares and those without spares, because the Arrays without spares contain more capacity and also have more disks for RAID data striping.
    (e) Elapsed time to rebuild RAID-5 is more than for RAID-10. RAID-10 array rebuilds have less impact on other disks on the same disk loop.
    (f) RAID-10 offers higher availability than RAID-5 because it can tolerate multiple DDM failures and can continue to function. RAID-5 can tolerate one DDM failure.
    (g) RAID-10 uses almost twice as many DDMs as RAID-5 for the same capacity. This can be expensive, so Disk Magic should be run to see if RAID-10 would actually offer performance enhancement over RAID-5.

(h) Many features reduce the potential need for, and the resulting possible performance impact of, an array rebuild made necessary by a DDM failure. This is especially important for anyone concerned about the performance impact of DDM failures and the resulting array rebuild in RAID-5 (see item (e) above):

- Arrays Across Loops (AAL)
- smart policy for floating spares
- hot-pluggable DDMs
- Predictive Failure Analysis (PFA)
- Periodic automatic disk scrubbing

8.1.4 Extent Pool and Volume (LUN) configuration

This section summarizes configuration of extent pools and LUN configuration.

1) It is highly recommended to use storage pool striping, i.e., striping over multi-rank extent pools. This is new with LMC 5.30xx.xx. Previous LMC versions allowed multi-rank extent pools, but the ranks were effectively concatenated, not striped. The guidelines for configuring storage pool striping are:
   (a) Use 4 – 8 ranks per extent pool.
   (b) The extent pools should alternate between servers, i.e., evenly intermix, in order, even-numbered and odd-numbered extent pools.
   (c) The ranks should have the same RAID characteristics and disk RPM's.
   (d) Balance the ranks across DA's.
2) Having more ranks per extent pool increases the possibility that a rank outage will impact the entire extent pool. Hence the recommendation to limit the number of ranks per extent pool to four to eight.
3) Allow the DS8000 to decide how to allocate extents to new volumes, i.e., it uses an algorithm where it picks up the initial extent from where it last left off, called extent rotation. It is possible to specify which rank extents to use when creating a volume, but do not use that option without good reason.
4) There is no reorganization function for extents in an Extent Pool. So if you add more ranks to a full Extent Pool, it is best to add several at a time to still be able to utilize storage pool striping.
5) Internal to the DS8000, there is no performance difference for LUN sizes.

9 Monitoring and Modeling tools

This section describes two important tools that should be used in any SAN environment which includes the DS8000. TotalStorage Productivity Center (TPC), which as of this writing is at Version 3.3.2, is IBM's tool for monitoring all layers of the technology stack in a SAN environment. It is not a monitoring tool in the traditional sense, where polling is constantly taking place for the hardware and software in the technology stack. Rather it collects configuration and performance data at regularly scheduled intervals so that the data can be analyzed and reported on.

Disk Magic is a modeling tool which has been licensed to IBM by the owner of the product, IntelliMagic BV. This tool can be used to predict the effect that storage configuration changes will have on performance.

9.1 TotalStorage Productivity Center (TPC) V 3.3.2

TotalStorage Productivity Center (TPC) is an essential tool in an IBM SAN environment to ensure the health of the data center. TPC offers the following features:

1) It presents a graphical overview of the entire data center topology, from hosts to Fibre Channel switches to storage.
2) It allows a drill-down into each object in the topology. For example, you can select a given DS8000 and expand it to view all of the layers of the virtualization hierarchy.
3) It collects very detailed performance data on LUNs, RAID arrays, switches, etc. For example, for a given LUN over a specified time period you can see the IOPS, the response time, the throughput and the read or write cache hit ratio.
4) It offers a wide range of reports that can be used to analyze the collected performance data.

TPC is simply the only way to monitor and report on the all of the layers of the technology stack in the IBM SAN environment and is a critical component of the setup in a data center.

9.2 Disk Magic

Disk Magic is a storage configuration modeling tool. It can be used to predict the performance impact of changes in the configuration of IBM SAN storage. It also supports storage from other vendors. Disk Magic can be used both before deploying a new DS8000, to predict the outcome of moving data to the DS8000, or also after deployment to predict the consequences of configuration changes. Disk Magic is an internal IBM modeling tool which is available from IBM Sales and Business Partners. It is also available for purchase from the vendor, IntelliMagic BV.

These are some of the storage configuration changes for which Disk Magic can predict a performance impact:

- Move the current I/O load to a different disk subsystem.
- Merge the I/O load of multiple disk subsysgtems into a single one.
- Insert a SAN Volume Controller into an existing disk configuration.
- Increase the current I/O load.
- Increase the disk subsystem cache size.
- Change to large DDMs.
- User fewer or more LUNs.

It is advisable to run Disk Magic in any installation that is about to deploy a DS8000 to get an idea of the best way to configure the storage.

10 Orion - Oracle I/O Numbers Calibration tool

The Orion tool can be used to generate Oracle-specific I/O load and to gather the resulting performance statistics. The overview and description of the input parameters for Orion given below is copied directly from the Orion User's Guide. That guide should be referenced for the most complete understanding of the tool.

10.1 Orion tool overview

Orion is a tool for predicting the performance of an Oracle database without having to install Oracle or create a database. Unlike other I/O calibration tools, Orion is expressly designed for simulating Oracle database I/O workloads using the same I/O software stack as Oracle. It can also simulate the effect of striping performed by ASM.

The following types of I/O workloads are supported:

1) Small Random I/O: OLTP applications typically generate random reads and writes whose size is equivalent to the database block size, typically 8 kB. Such applications typically care about the throughput in I/Os Per Second (IOPS) and about the average latency (I/O turn-around time) per request. These parameters translate to the transaction rate and transaction turn-around time at the application layer.

   Orion can simulate a random I/O workload with a given percentage of reads vs. writes, a given I/O size, and a given number of outstanding I/Os. The I/Os are distributed across all disks.

2) Large Sequential I/O: Data warehousing applications, data loads, backups, and restores generate sequential read and write streams composed of multiple outstanding 1 MB I/Os. Such applications are processing large amounts of data, like a whole table or a whole database and they typically care about the overall data throughput in MegaBytes Per Second (MBPS).

   Orion can simulate a given number of sequential read or write streams of a given I/O size with a given number of outstanding I/Os. Orion can optionally simulate ASM striping when testing sequential streams.

3) Large Random I/O: A sequential stream typically accesses the disks concurrently with other database traffic. With striping, a sequential stream is spread across many disks. Consequently, at the disk level, multiple sequential streams are seen as random 1 MB I/Os, which we also call Multi-User Sequential I/O.

4) Mixed Workloads: Orion can simulate 2 simultaneous workloads: Small Random I/O and either Large Sequential I/O or Large Random I/O. This enables you to simulate, for example, an OLTP workload of 8 kB random reads and writes with a backup workload of 4 sequential read streams of 1 MB I/Os.

For each type of workload, Orion can run tests at different levels of I/O load to measure performance metrics like MBPS, IOPS and I/O latency. Load is expressed in terms of the number of outstanding asynchronous I/Os. Internally, for each such load level, the Orion software keeps issuing I/O requests as

fast as they complete to maintain the I/O load at that level. For random workloads (large and small), the load level is the number of outstanding I/Os. For large sequential workloads, the load level is a combination of the number of sequential streams and the number of outstanding I/Os per stream. Testing a given workload at a range of load levels helps the user understand how performance is affected by load.

10.2 Orion input parameters

This section lists mandatory and optional input parameters to the Orion tool

MANDATORY INPUT PARAMETERS

run: Test run level. This option provides simple command lines at the simple and normal run levels and allows complex commands to be specified at the advanced level. If not set as -run advanced, then setting any other non-mandatory parameter (besides -cache_size or -verbose) will result in an error.

simple: Generates the Small Random I/O and the Large Random I/O workloads for a range of load levels. In this option, small and large I/Os are tested in isolation. The only optional parameters that can be specified at this run level are -cache_size and -verbose. This parameter corresponds to the following invocation of Orion:

```
# ./orion -run advanced -testname mytest  \
          -num_disks NUM_DISKS  \
          -size_small 8 -size_large 1024 -type rand  \
          -simulate concat -write 0 -duration 60  \
          -matrix basic
```

normal: Same as -simple, but also generates combinations of the small random I/O and large random I/O workloads for a range of loads. The only optional parameters that can be specified at this run level are -cache_size and -verbose. This parameter corresponds to the following invocation of Orion:

```
# ./orion -run advanced -testname mytest  \
          -num_disks NUM_DISKS  \
          -size_small 8 -size_large 1024 -type rand  \
          -simulate concat -write 0 -duration 60  \
          -matrix detailed
```

advanced: Indicates that the test parameters will be specified by the user. Any of the optional parameters can be specified at this run level.

testname: Identifier for the test. The input file containing the disk or file names must be named <testname>.lun. The output files will be named with the prefix <testname>_.

num_disks: Actual number of physical disks used by the test. This number is used to generate a range for the load.

OPTIONAL INPUT PARAMETERS

help: Prints Orion help information. All other options are ignored when help is specified.

size_small: Size of the I/Os (in kB) for the Small Random I/O workload. (Default is 8).

size_large: Size of the I/Os (in kB) for the Large Random or Sequential I/O workload. (Default is 1024).

type: Type of the Large I/O workload. (Default is rand):

rand: Large Random I/O workload.

seq: Large Sequential I/O workload.

num_streamIO: Number of outstanding I/Os per sequential stream. Only valid for -type seq. (Default is 1).

simulate: Data layout to simulate for Large Sequential I/O workload.

concat: A virtual volume is simulated by serially chaining the specified LUNs. A sequential test over this virtual volume will go from some point to the end of one LUN, followed by the beginning to end of the next LUN, etc.

raid0: A virtual volume is simulated by striping across the specified LUNs. The stripe depth is 1M by default (to match the Oracle ASM stripe depth) and can be changed by the -stripe parameter.

write: Percentage of I/Os that are writes; the rest being reads. This parameter applies to both the Large and Small I/O workloads. For Large Sequential I/Os, each stream is either read-only or write-only; the parameter specifies the percentage of streams that are write-only. The data written to disk is garbage and unrelated to any existing data on the disk. **WARNING: WRITE TESTS WILL OBLITERATE ALL DATA ON THE SPECIFIED LUNS**.

cache_size: Size of the storage array's read or write cache (in MB). For Large Sequential I/O workloads, Orion will warm the cache by doing random large I/Os before each data point. It uses the cache size to determine the duration for this cache warming operation. If not specified, warming will occur for a default amount of time. If set to 0, no cache warming will be done. (Default is not specified, which means warming for a default amount of time).

duration: Duration to test each data point in seconds. (Default is 60).

matrix: Type of mixed workloads to test over a range of loads. (Default is detailed).

basic: No mixed workload. The Small Random and Large Random/Sequential workloads will be tested separately.

detailed: Small Random and Large Random/Sequential workloads will be tested in combination.

point: A single data point with S outstanding Small Random I/Os and L outstanding Large Random I/Os or sequential streams. S is set by the **num_small** parameter. L is set by the **num_large** parameter.

col: Large Random/Sequential workloads only.

row: Small Random workloads only.

max: Same as detailed, but only tests the workload at the maximum load, specified by the **num_small** and **num_large** parameters.

num_small: Maximum number of outstanding I/Os for the Small Random I/O workload. This can only be specified when **matrix** is **col**, **point**, or **max**.

num_large: Maximum number of outstanding I/Os for the Large Random I/O workload or number of concurrent large I/Os per stream. This can only be specified when **matrix** is **row**, **point**, or **max**.

verbose: Prints progress and status information to standard output.

The offsets for I/Os are determined as follows:

For Small Random and Large Random workloads:

- The LUNs are concatenated into a single virtual LUN (VLUN) and random offsets are chosen within the VLUN.

For Large Sequential workloads:

- With striping (-simulate raid0). The LUNs are used to create a single striped VLUN. With no concurrent Small Random workload, the sequential streams start at fixed offsets within the striped VLUN. For n streams, stream i will start at offset VLUNsize * (i + 1) / (n + 1), except when n is 1, in which case the single stream will start at offset 0. With a concurrent Small Random workload, streams start at random offsets within the striped VLUN.
- Without striping (-simulate CONCAT). The LUNs are concatenated into a single VLUN. The streams start at random offsets within the single VLUN.

11 Lab Setup

This section lists the hardware and software configurations that were used in the lab.

11.1.1 System Storage DS8000

Table 1 describes the DS8000 image used in the performance exercise. This DS8000 storage unit has two separate images, and behaves like two separate DS8000's. However, only one image was used for the testing.

IBM System Storage DS8000 Model	DS83000 Turbo 9B2
Storage Unit	IBM.2107-75DF160
Storage Image	Image2
Storage id	IBM.2107-75DF162
Code levels	
License Machine Code (LMC)	5.3.1.116
Storage Manager	6.1.3.20071026.1
DSCLI	5.3.1.101
WWNN	5005076307FFCF7D
DDMs	300GB, 15K RPM
Number of arraysites	8
Number of DDMs	64 (8 per arraysite)
Non-Volatile Storage	2.0 GB
Cache Memory	54.3 GB
Processor Memory	62.7 GB

Table 1 - DS8000 Storage Unit

11.1.2 Brocade switches

Table 2 describes the four IBM Brocade switches used in the exercise. The switches are configured into two fabrics with two switches per fabric.

IBM name	IBM System Storage SAN32B-3
IBM machine type	2005
IBM machine model	B5K
Brocade name	Brocade 5000
Ports per switch	thirty-two 4 Gbs ports
Kernel	2.6.14
Fabric OS	v5.3.1
BootProm	4.6.5

Table 2 - Brocade switches

11.1.3 Host nodes

Table 3 describes the host nodes that will be used in the exercise.

Server type	IBM System x3550
Processor	4 x Dual-Core Intel® Xeon® processor 5160 @ 3.00 GHz
Memory	8 GB
Host bus adapter (HBA) model	QLE2462
Operating system	Red Hat Enterprise Linux® (RHEL) AS4 U6
Kernel version	2.6.9-67.ELsmp
Multipath software	device-mapper-multipath-0.4.5-27.RHEL4
HBA driver	8.01.07-d4
HBA firmware	4.00.150

Table 3 - host nodes

12 Testing Methodology

This section describes the methodology used for testing performance. It will note which metrics will be gathered from both Orion and TPC and how they will be correlated. It will also list the matrix of possible variations with respect to the following:

(1) The parameters used for each Orion test run

(2) The storage configuration used for each run in (1)

The permutations of (1) and (2) that are actually used for the testing will be noted in the introduction to the exercises.

12.1 Orion test run descriptions

The following describes the individual Orion test runs that will be considered and the workload that the runs are simulating. The **-advanced** parameter will always be used so that all of the parameters which are used in a run are explicitly stated:

12.1.1 OLTP only

This will use 8 kB block sizes and varying percentage combinations of reads and writes. The combinations tested will be:

- 100% reads (not a realistic production load, but gives a baseline for pure reads)
- 90% reads and 10% writes
- 70% reads and 30% writes
- 50% reads and 50% writes
- 100% writes (not a realistic production load, but gives a baseline for pure writes)

The combination of 70% reads and 30% writes is often cited as fairly representative of a typical OLTP workload. But the particular profile for any running database must be determined by running the SQL given in Appendix A. The parameters used with Orion for RAID-10 with 70% reads and 30% writes would be:

```
# ./orion -run advanced -testname oltp_raid10_read70_write30 \
        -num_disks 4 -size_small 8 -cache_size 50000 \
        -num_large 0 -write 30 -matrix row -duration 60 -verbose
```

12.1.2 Data Warehouse only (large sequential reads and writes)

This will use 1 MB block sizes and varying percentage combinations of reads and writes. The combinations tested are as follows:

- 100% reads
- 70% reads and 30% writes
- 50% reads and 50% writes
- 100% writes

The parameters used with Orion for RAID-5 with 100% reads would be:

```
# ./orion -run advanced -testname seq_raid5_read100_write0 \
        -num_disks 4 -type seq -size_large 1024 -simulate raid0 \
        -stripe 1024 -write 0 -num_small 0 -matrix col -duration 60 \
        -verbose
```

12.1.3 Mixed OLTP and Data Warehouse

This will simulate a mixed OLTP and Data Warehouse load. The block size for the small, random I/O will be 8 kB and the block size for the large, sequential I/O will be 1 MB. The combinations tested are as follows:

- 70% reads and 30% writes
- 50% reads and 50% writes

The parameters used with Orion for RAID-10 with 70% reads and 30% writes are:

```
# ./orion -run advanced -testname mixed_raid10_read70_write30 \
        -num_disks 4 -size_small 8 -size_large 1024 -type rand \
        -simulate raid0 -stripe 1024 -write 30 -matrix detailed \
        -duration 60 -verbose
```

As noted in the Orion User's Guide, in a mixed environment with both small-block, random I/O's and large-block, sequential I/O's, the large-block sequential I/O's actually get interpreted by the storage as large-block, random I/O's. That is the reason that the **-type rand** parameter will be used in this test as opposed to the **-type seq** parameter that was used in the Data Warehouse only test.

12.2 Storage configuration variations for Orion test runs

Below are the storage configurations which will be considered. The assumption is that each LUN in these configurations would map to one ASM disk:

(1) Four 100 GB LUNs in one extent pool which consists of one rank. Therefore all LUNs are actually on only one RAID array. This simulates a configuration that may exist in a data center where there is not the luxury of allocating LUNs from separate RAID arrays when adding space to a specific database, i.e., where the SAN storage is considered a "disk farm" and space is allocated from whichever array has free space. It might also be said to exist in a data center where perhaps the storage planning has not followed best practices as closely as possible.

Although ASM striping will be simulated with this configuration for the Data Warehouse workloads, i.e. **-simulate raid0**, this should not perform as well as configuration (2) below because the same backend RAID array is being used for all of the LUNs.

(2) Four 100 GB LUNs in four separate extent pools that each consist of one rank. This puts each LUN on a separate RAID array. This will spread the workload across DS8000 servers since extent pools have a server affinity. It should also spread the workload across Device Adapters (DA's). This is an optimal configuration for database space allocation since it implements full workload spreading.

(3) Four 100 GB LUNs from one multi-rank extent pool that is using storage pool striping. The results from this test should be compared to configuration (2) to see the performance characteristics of this relatively new feature (introduced with LMC 5.30xx.xx in October 2007). Since extent pools have an affinity for a specific DS8000 server, only one server will be utilized and therefore this may not perform quite as well as configuration (2) for some workloads.

The actual storage configurations used for each Oracle I/O workload type from the above list will be noted in the introduction to the exercises.

12.3 Metrics collected for each Orion test run

The following are the metrics that will be collected for each Orion run:

- **Orion metrics**:
  - IOPS
  - MBPS
  - Latency

  As discussed in the excerpt from the Orion User's Guide, IOPS and Latency will only be generated if random I/O's are part of the Orion run and MBPS will only be generated if large sequential I/O's are part of the Orion run.

- **TPC metrics –** the following will be collected at the RAID array level. If an array contains more than one LUN used as a target in the Orion test run, the sum of the statistics from all of the LUNs in a given array always equals the statistics gathered for that array as a whole:
  - **IOPS:**
    - Total I/O rate (overall)
  - **MBPS:**
    - Total Data Rate
  - **Response Time (Latency):**
    - Overall Response Time

12.4 Analysis of metrics collected for each Orion test run

There will be no way to correlate the data points from Orion exactly with the 5-minute interval average data collection points of TPC. Therefore, the comparison of the collected metrics will be done as follows:

- **Overall Summary Comparison** – There will be a comparison of the metrics for the entire run period. Orion provides a <testname>_summary.txt file that contains the peak metrics observed over the entire run time. This is measured at a discrete data point. The reports in TPC show average metrics per 5 minute sample period. So the peak average metric collected by TPC over the entire run time will be compared to the peak discrete data point metric for Orion.

  This is not a perfect basis for comparison, since it is comparing a discrete data point (Orion) to a maximum average over a 5 minute time period (TPC), but it is the best comparison available.

- **Comparison over a continuum during the same time period** – One way to compensate for the fact that a comparison cannot be done at exactly each data point for Orion and TPC is to graph the results and do a visual comparison over the entire run time period. However, since the sample data points for Orion do not map exactly to TPC, and since Orion collects data at more data points than TPC for the run period, a graph will only be included for Orion. As mentioned above, the TPC metrics are averaged over 5 minute intervals whereas the Orion metrics are gathered at discrete data points.

- **Comparison by absolute metric number, not by percentage difference** – The most meaningful comparison of the collected metrics at a given data point is to compare by the difference in absolute numbers. In other words, it is more helpful to be able to say for a given data point "RAID-10 offers 1500 more IOPS at this data point than RAID-5" as opposed to saying "RAID-10 offers 10% more IOPS than RAID-5 at this data point". The same goes for

the comparisons of Latency and MBPS. This approach can contribute in a more meaningful manner to being able to do a cost-benefit analysis when making a purchasing decision. You are buying “horsepower” in absolute terms, not by comparison of percentage differences.

The metrics gathered from Orion will be included in this document in graphical format. The metrics gathered from TPC will not be included in this document. The TPC metrics are going to be used for comparison with, and verifiability of, the Orion metrics. The important point to remember is that any data center which is deploying a DS8000 should have TPC installed and running as part of the storage monitoring setup.

13 Benchmarking Exercise – Comparison of RAID-5 and RAID-10

This section will run through an example of how Orion and TPC can be used to evaluate one configuration, RAID-5 versus RAID-10, in an Oracle ASM environment. It will compare the results to the best practices guidelines given in the section **8.1.3 RAID Level**. This will confirm the differences in performance depending on the particular Oracle I/O workload used.

The storage configuration used for all of the runs in this exercise is configuration (1) in the section **12.2 Storage configuration variations for Orion test runs**. This has all 4 LUNs on one extent pool which consists of one rank. This is considered the least optimal performance configuration since it does not employ full workload spreading. However, the metrics from the various test runs can be compared given that they all have this configuration in common. The comparison of storage configuration (1) to the highly preferable storage configurations (2) and (3) will be discussed in the section **14 Benchmarking Exercise – Comparisons of LUN distributions across Arrays**.

13.1 Prerequisites

The prerequisite setup for the exercise is as follows:

- Orion has been installed on the host node.
- DS8000 arrays, ranks and extent pools have been created. Neither the RAID-5 nor the RAID-10 arrays will have spare DDMs for this testing.
- DS8000 LUNs have been created and made visible to the host.
- TPC is collecting performance metrics at 5 minute intervals.

Examples of the DSCLI commands used to create the RAID-5 and RAID-10 arrays are:

```
dscli > mkarray -type raid5 –arsite S6

dscli> mkarray –type raid10 -arsite S7
```

Examples of the DSCLI commands used to create the extent pools are:

```
dscli> mkextpool -rankgrp 0 -stgtype fb "Extent Pool 0"

dscli> mkextpool -rankgrp 1 -stgtype fb "Extent Pool 1"
```

Examples of the DSCLI commands used to create the ranks are:

```
dscli> mkrank -extpool p5 -array A5
dscli> mkrank -extpool p6 -array A6
```

Examples of the DSCLI commands used to create the fixed block volumes (LUNs) are:

```
dscli> mkfbvol -extpool P5 -cap 100 -volgrp V2 -name ORION_#h 0500-0503
dscli> mkfbvol -extpool P6 -cap 100 -volgrp V2 -name ORION_#h 0600-0603
```

13.2 Summary and Analysis

This section will summarize and analyze the results of the Orion test runs.

13.2.1 Performance Assertion #1 – RAID-5 and RAID-10 perform equally for reads

This assertion states that for reads from disk, either random or sequential, there is no significant difference between RAID-5 and RAID-10.

Test Runs used for Assertion #1

Two test runs were used to verify this assertion:

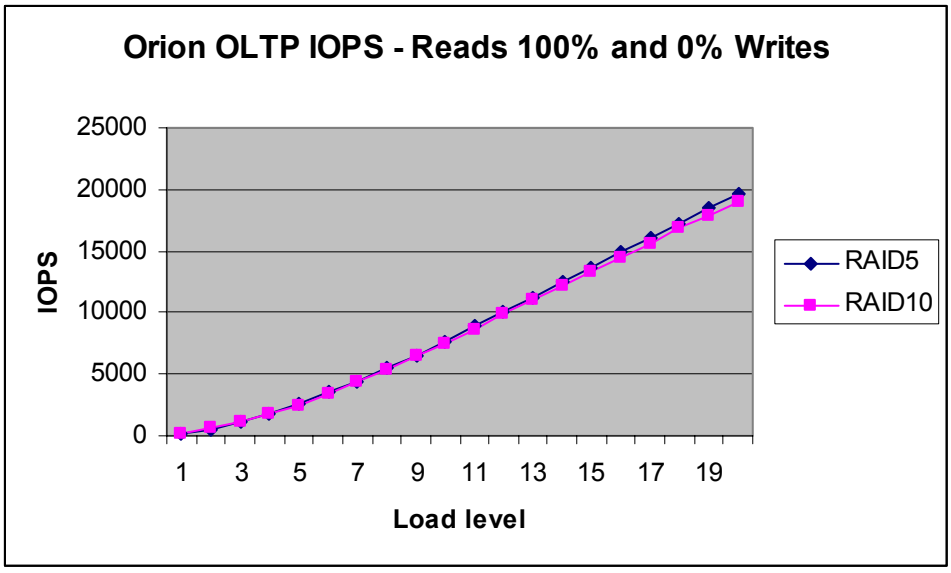
- OLTP 100% Read and 0% Write
- Data Warehouse (Sequential) 100% Read and 0% Write

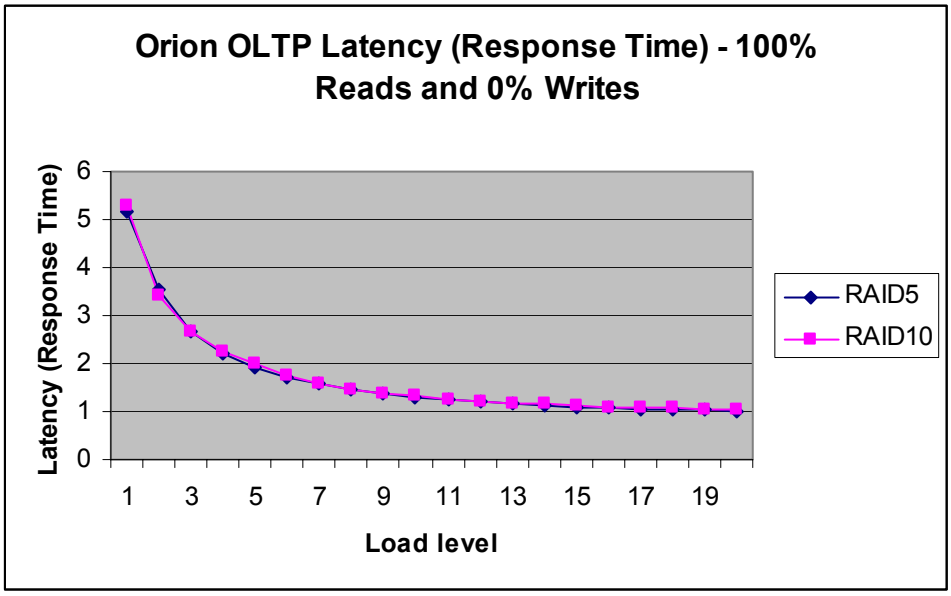
The 100% Read and 0% Write OLTP I/O mix is an admittedly unrealistic OLTP production environment. However, there was no other way to control for the fact that only reads were to be tested in isolation. In addition, the IOPS for the OLTP in this test fit the description in the Orion User's Guide of "unbelievably good" results. As the Orion User's Guide mentions, this is due to the read cache. For these tests, unlike any of the other tests, the read cache hit ratio was over 99%. Using the -cache parameter did not change this result. The only time that the IOPS were more in line with the other tests was if the test was run immediately after a RAID array had been created and no other testing had yet been done on that array.

The Data Warehouse (Sequential) 100% Read and 0% Write workload is a realistic workload for a reporting or business intelligence database when there are no data loads occurring concurrently.

Test Results and Analysis for Assertion #1

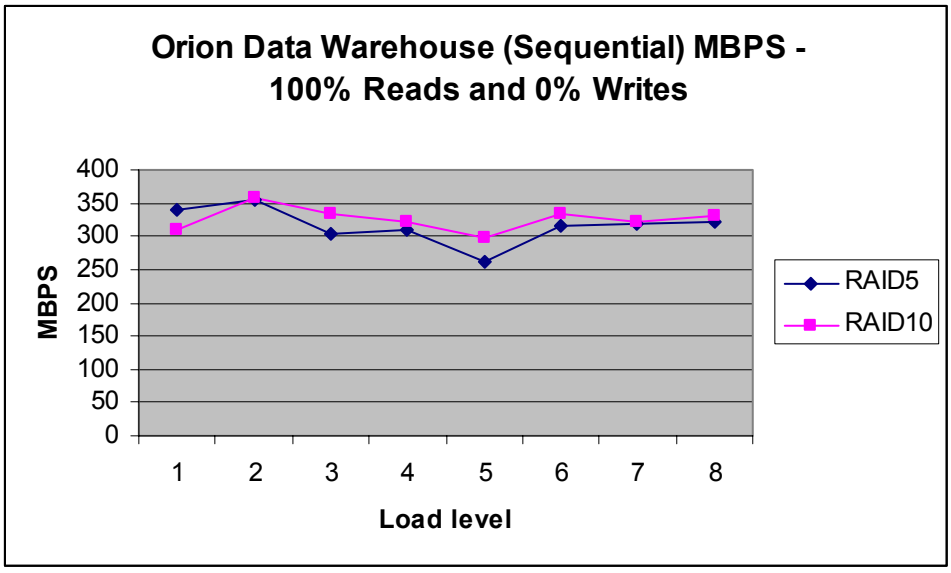
The results of the tests confirm the assertion that there is no significant difference between RAID-5 and RAID-10 for either random reads or sequential reads. The IOPS and Latency graphs for random reads are shown below.





For the small-block, random I/O OLTP peak workload, Orion shows the IOPS to be only 600 more for RAID-5 over RAID-10 when the IOPS are in the 19,000 IOPS range. And TPC showed the IOPS difference to be only 400 more for RAID-10 when the IOPS are in the 16,000 IOPS range. The graphs above show how closely the RAID-5 and RAID-10 IOPS and Latency follow each other over the test run.

The Orion results for the Data Warehouse Read (Sequential) workload in the graph below show how closely the RAID-5 and RAID-10 MBPS track each other over the course of the test run. There is no significant difference between RAID-5 and RAID-10 for large sequential reads.



13.2.2 Performance Assertion #2 – RAID-5 performs better for sequential writes

This assertion states that for sequential writes to disk, RAID-5 performs better.

Test Runs used for Assertion #2

Three test runs were used to verify this assertion:

- Data Warehouse (Sequential) 0% Read and 100% Write
- Data Warehouse (Sequential) 50% Read and 50% Write
- Data Warehouse (Sequential) 70% Read and 30% Write

The Data Warehouse (Sequential) 0% Read and 100% Write workload would correspond to an environment where a data load is occurring for a data warehouse and there are no queries occurring concurrently.

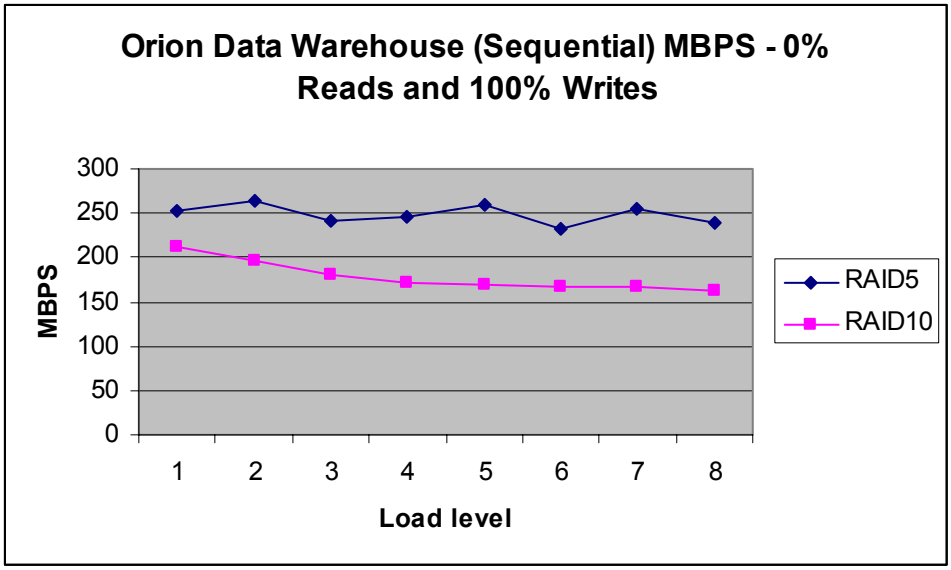
The Data Warehouse (Sequential) 70% Read and 30% Write and the Data Warehouse (Sequential) 50% Read and 50% Write would correspond to a data warehouse where reporting queries and data loads may occur concurrently.

Test Results for Assertion #2

The results of the tests show that RAID-5 performs significantly better than RAID-10 for large, sequential writes when the I/O mix is either 0% Read and 100% Write or when it is 50% Read and 50% Write. The picture is slightly different when the ratio is 70% Read and 30% Write. For that ratio, at a couple of the lower load levels, RAID-10 actually performs slightly better. But as the load increases, RAID-5 performs better, although not quite to the extent as the other two workload mixes.

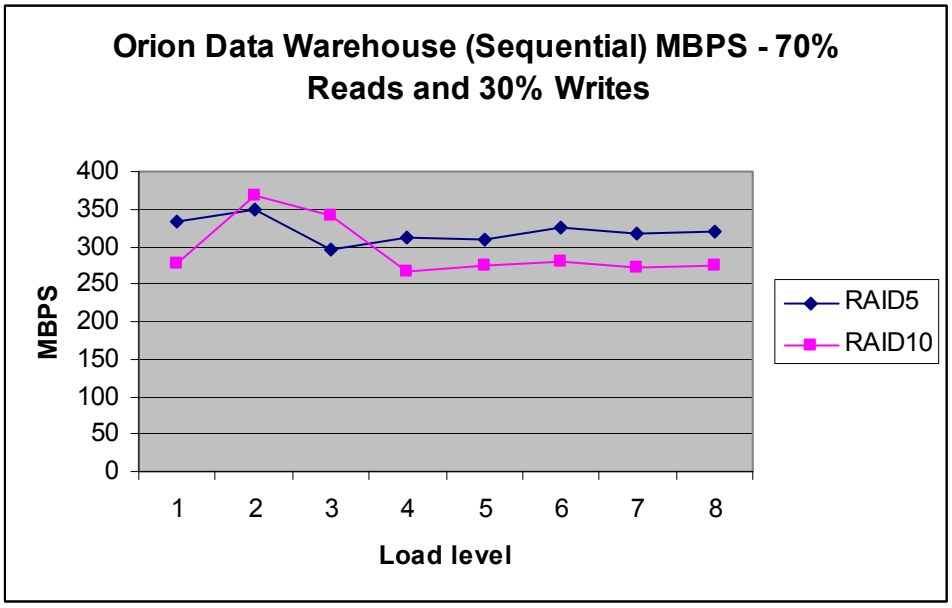
For the 0% Read and 100% Write workload, the Orion RAID-5 data points are anywhere from approximately 230 MBPS to approximately 265 MBPS throughput. The Orion RAID-10 data points are anywhere from approximately 160 MBPS to approximately 210 MBPS throughput. The performance

improvement for RAID-5 over RAID-10 ranges from around 40 MBPS up to 90 MBPS, with most data points showing a 60 – 70 MBPS throughput improvement. Below is the graph of the Orion data points for the 0% Read and 100% Write sequential workload.



For the 50% Read and 50% Write workload, the Orion RAID-5 data points are anywhere from approximately 286 MBPS to 339 MBPS throughput. The Orion RAID-10 data points are anywhere from approximately 235 MBPS to 255 MBPS throughput. The performance improvement for RAID-5 over RAID-10 ranges from around 40 MBPS up to 85 MBPS, with most data points showing improvements in the 50 – 80 MBPS range. Below is the graph of the Orion data points for the 50% Read and 50% Write sequential workload.

Finally, below is the graph of the Orion data points for the 70% Read and 30% Write sequential workload showing the less significant difference between RAID-5 and RAID-10 for this workload mix.



13.2.3 Performance Assertion #3 – RAID-10 performs better for random writes

This assertion states that for random writes, RAID-10 performs better than RAID-5.

Test Runs used for Assertion #3

Six test runs were used to verify this assertion:

- OLTP 90% Read and 10% Write
- OLTP 70% Read and 30% Write
- OLTP 50% Read and 50% Write
- OLTP 0% Read and 100% Write
- Mixed OLTP and Data Warehouse (Sequential) 70% Read and 30% Write
- Mixed OLTP and Data Warehouse (Sequential) 50% Read and 50% Write

The first three OLTP workloads listed above, 90% Read and 10% Write, 70% Read and 30% Write and 50% Read and 50% Write are realistic OLTP workloads for production shops. The 0% Read and 100% Write workload is similar to the OLTP 100% Read and 0% Write tested for Performance Assertion #1, i.e., it is not a realistic production workload but is used to test writes in isolation for at least one random write I/O test run.

The two Mixed OLTP and Data Warehouse (Sequential) workloads listed above, 70% Read and 30% Write and 50% Read and 50% Write correspond to a production environment where the database is

operating in a hybrid mode, i.e., both OLTP workloads and Data Warehouse reports are occurring concurrently. This is a realistic workload for many production databases.

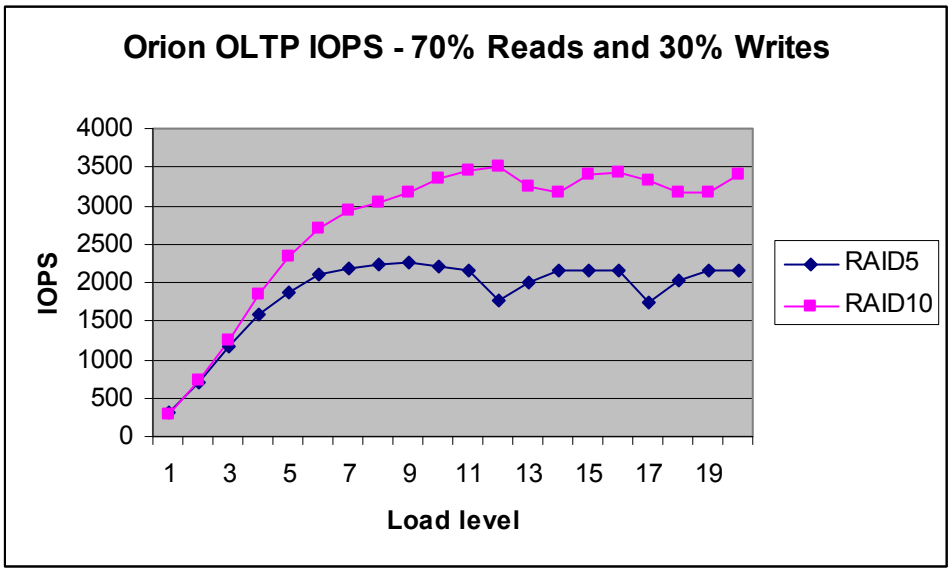
Test Results for Assertion #3

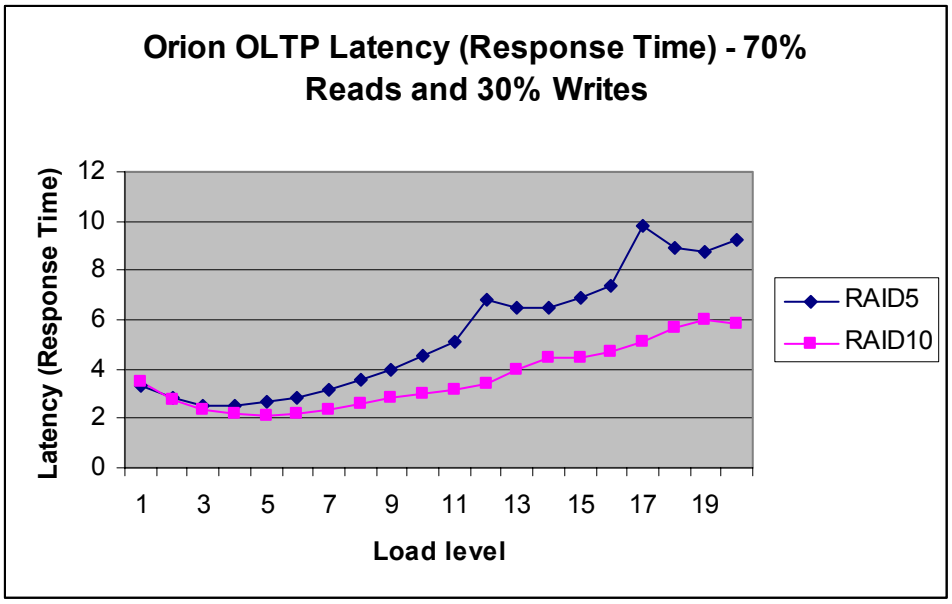
The results of the tests show that, with one minor exception, RAID-10 performs significantly better than RAID-5 whenever any OLTP random writes are occurring. The RAID-10 over RAID-5 performance improvement increases markedly whenever the percentage of writes increases and also whenever the I/O load starts increasing past the lowest load levels. The one minor exception is at the lowest Orion load levels for an OLTP 90% Read and 10% Write workload. In that situation, the RAID-5 and RAID-10 IOPS closely track each other for those lowest load levels. But once Orion starts using higher load levels for that workload mix, it shows the largest RAID-10 performance improvement over RAID-5 than any of the other workload mixes. At the highest load level for OTLP 90% Read and 10% Write, the RAID-10 offers 2800 more IOPS than RAID-5.

Below are three I/O workload mixes that represent the significant IOPS and Response Time advantage of RAID-10 over RAID-5 whenever any OLTP random writes are occurring at all:

OLTP 70% Reads and 30% Writes:

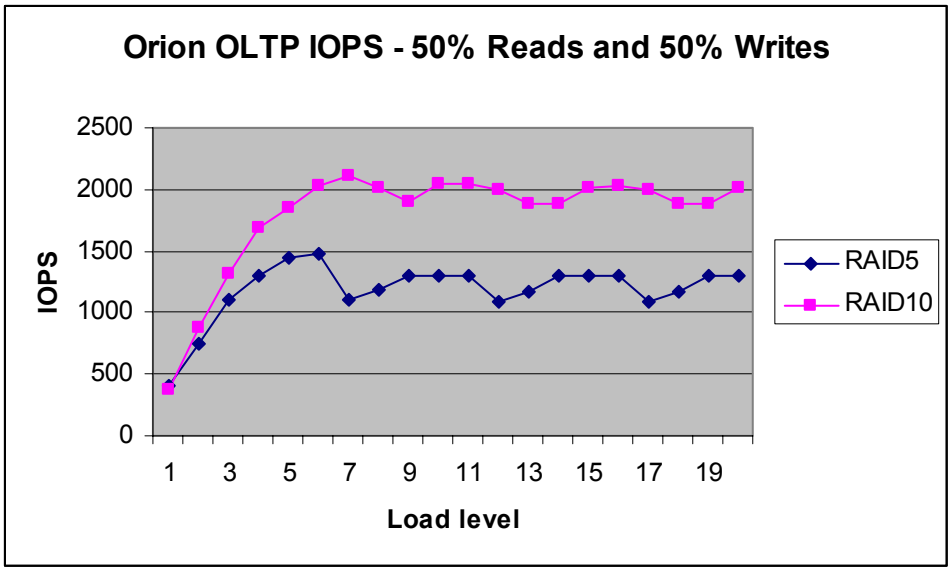
- RAID-5 IOPS range from 231 to 6598.
- RAID-10 IOPS range from 194 to 9357.
- RAID-10 generally offers an improvement at higher load levels of at least 1000 IOPS over RAID-5 with the largest improvement being 1600 IOPS at one data point.
- RAID-5 Latency (Response Time) ranges from 2.52 ms/op to 9.81 ms/op.
- RAID-10 Latency (Response Time) ranges from 2.17 ms/op to 5.87 ms/op.
- RAID-10 generally offers an improvement at higher load levels of from 2 ms/op to 3 ms/op with the largest improvement being 4.71 ms/op.

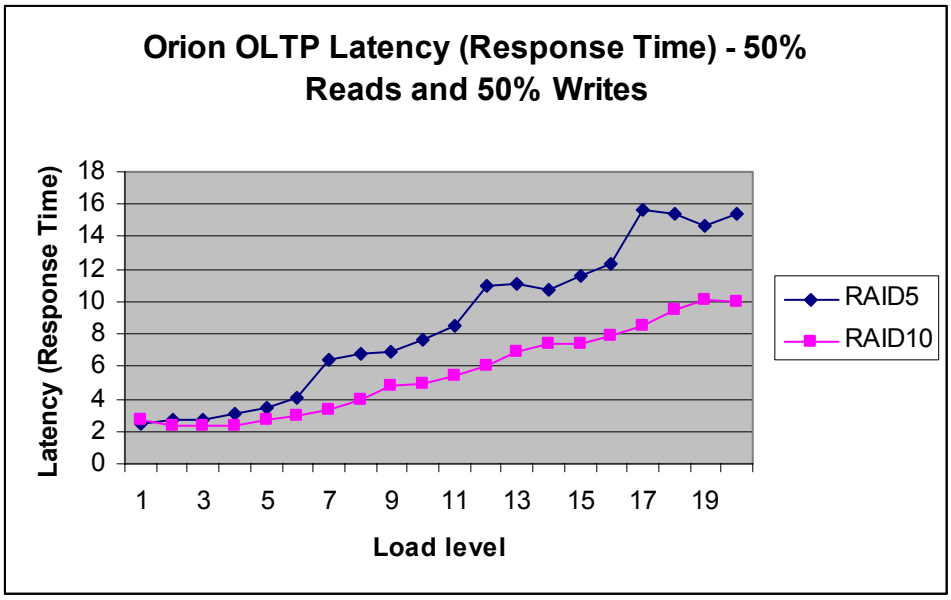




OLTP 50% Reads and 50% Writes:

- RAID-5 IOPS range from 405 to 1479.
- RAID-10 IOPS range from 374 to 2103.
- RAID-10 generally offers an improvement at higher load levels of between 500 - 900 IOPS over RAID-5 with the largest improvement being 1004 IOPS at one data point.
- RAID-5 Latency (Response Time) ranges from 2.47 ms/op to 15.66 ms/op.
- RAID-10 Latency (Response Time) ranges from 2.29 ms/op to 10.08 ms/op.
- RAID-10 generally offers an improvement at higher load levels of from 3 ms/op to 7 ms/op with the largest improvement being 5.86 ms/op.

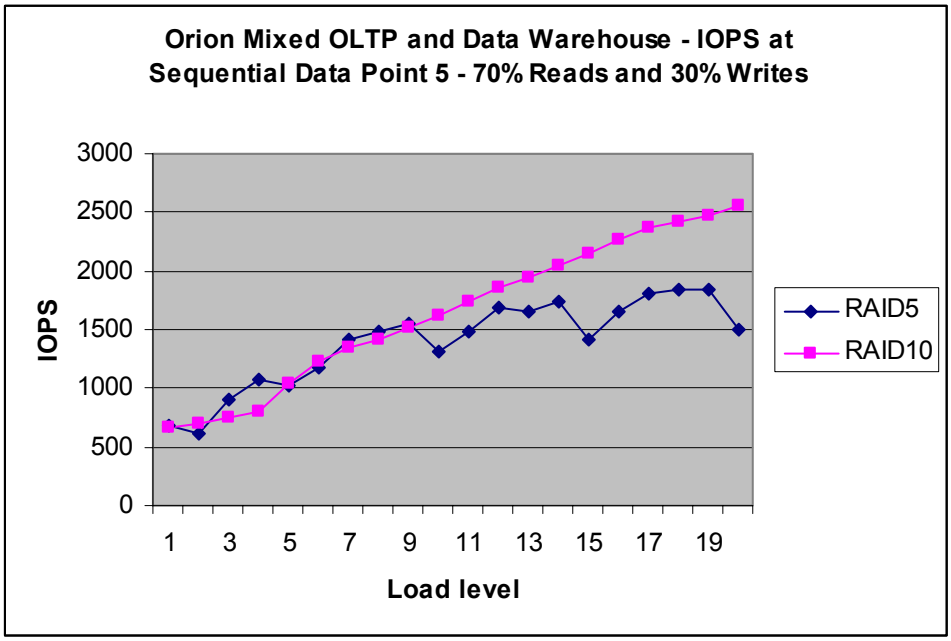


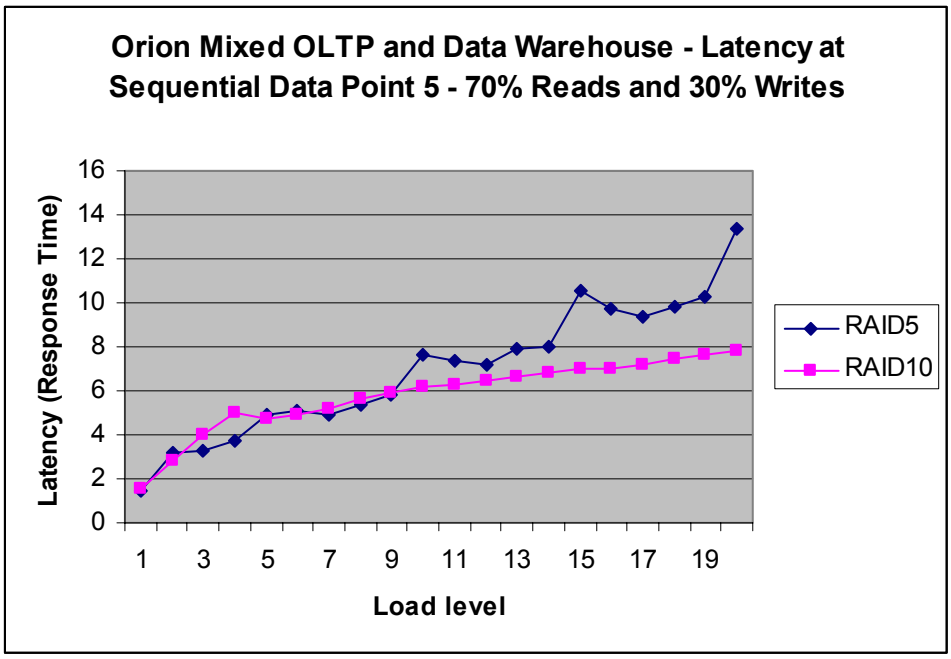


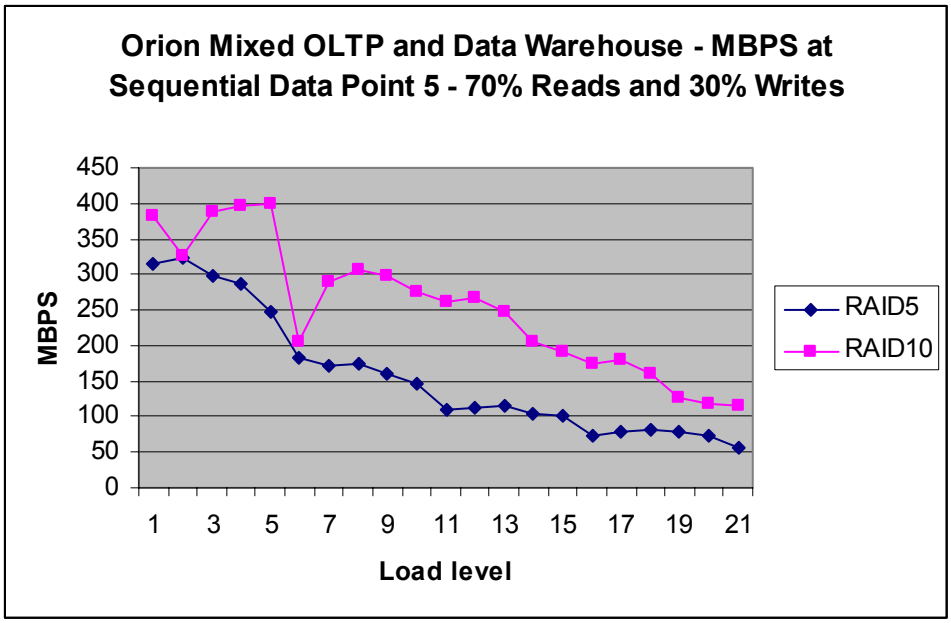
Mixed OLTP and Data Warehouse (Sequential) 70% Reads and 30% Writes at Sequential Data Point 5:

- RAID-5 IOPS range from 620 to 1848.
- RAID-10 IOPS range from 660 to 2562.
- RAID-10 generally offers an improvement at higher load levels of between 300 - 600 IOPS over RAID-5 with the largest improvement being 1070 IOPS at one data point.
- RAID-5 Latency (Response Time) ranges from 1.46 ms/op to 13.39 ms/op.
- RAID-10 Latency (Response Time) ranges from 1.51 ms/op to 7.8 ms/op.

- RAID-10 generally offers an improvement at the highest load levels of from 1 ms/op to 2 ms/op with the largest improvement being 5.59 ms/op.







14 Benchmarking Exercise – Comparisons of LUN distributions across arrays

This section will compare the performance of all three storage configurations in section **12.2 Storage configuration variations for Orion test runs**. To recap, these configurations are:

(1) All 4 LUNs on one extent pool which consists of one rank. This is considered the least optimal approach to LUN configuration since there is no workload spreading at all. This was the configuration used in the RAID-5 versus RAID-10 benchmarking exercise.

(2) Each of the 4 LUNs on its own array where each array consists of one rank. This implements full workload spreading since it will use two even-numbered extent pools that have an affinity for the DS8000 server0 and two odd-numbered extent pools that will have an affinity for server1.

(3) All 4 LUNs on one extent pool that consists of four ranks where all LUNs are using storage pool striping. For a LUN to be able to use storage pool striping, the mkfbvol command must be invoked with the "-eam rotateexts" parameter.

14.1 Prerequisites

The prerequisite setup for the exercise is as follows:

(1) Orion has been installed on the host node.

(2) DS8000 arrays, ranks and extent pools have been created. Neither the RAID-5 nor the RAID-10 arrays will have spare DDMs for this testing. For storage pool striping, one extent pool will be used which consists of four ranks.

(3) DS8000 LUNs have been created and made visible to the host.

(4) TPC is collecting performance metrics at 5 minute intervals.

Examples of the DSCLI commands used to create the RAID-5 and RAID-10 arrays are:

```
dscli > mkarray -type raid5 –arsite S6

dscli> mkarray –type raid10 -arsite S7
```

Examples of the DSCLI commands used to create the extent pools are:

```
dscli> mkextpool -rankgrp 0 -stgtype fb "Extent Pool 0"

dscli> mkextpool -rankgrp 1 -stgtype fb "Extent Pool 1"
```

Examples of the DSCLI commands used to create the ranks are:

```
dscli> mkrank -extpool p5 -array A5

dscli> mkrank -extpool p6 -array A6
```

Examples of the DSCLI commands used to create the fixed block volumes (LUNs) that are not using storage pool striping (the default configuration):

```
dscli> mkfbvol -extpool P5 -cap 100 -volgrp V2 -name ORION_#h 0500-0503

dscli> mkfbvol -extpool P6 -cap 100 -volgrp V2 -name ORION_#h 0600-0603
```

Examples of the DSCLI commands used to create the fixed block volumes (LUNs) that are using storage pool striping:

```
dscli> mkfbvol -extpool P5 -cap 100 -eam rotateexts -volgrp V2 -name ORION_#h 0500-0503
```

All of the I/O workload tests that were run in **13 Benchmarking Exercise – Comparison of RAID-5 and RAID-10** were run for this exercise. These are:

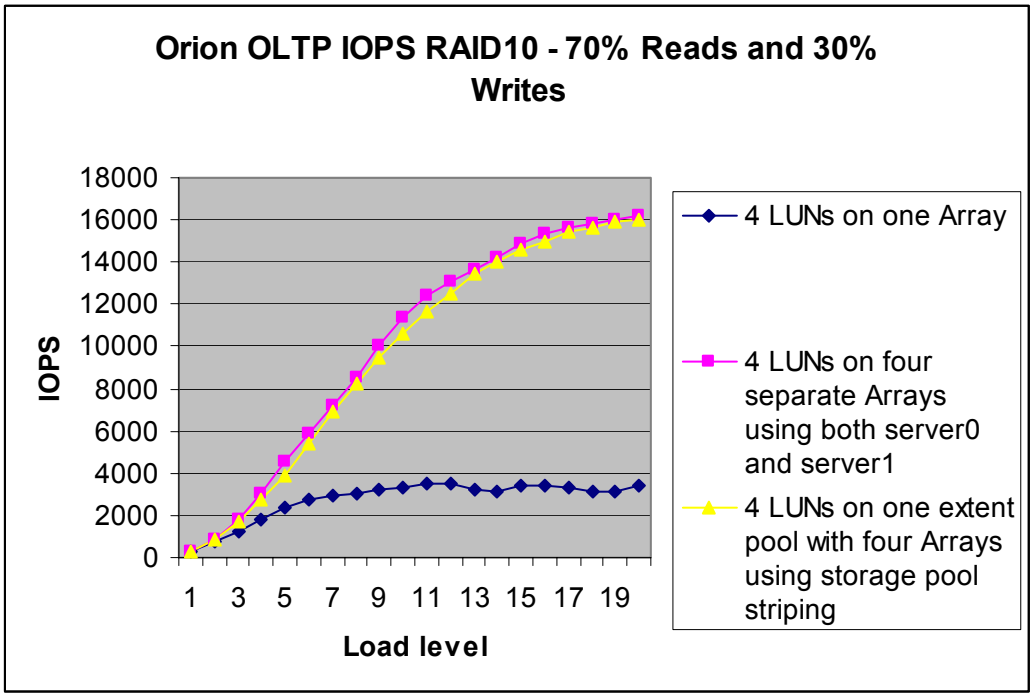
- OLTP 100% Read and 0% Write
- OLTP 90% Read and 10% Write
- OLTP 70% Read and 30% Write
- OLTP 50% Read and 50% Write
- OLTP 0% Read and 100% Write
- Data Warehouse (Sequential) 100% Read and 0% Write
- Data Warehouse (Sequential) 70% Read and 30% Write
- Data Warehouse (Sequential) 50% Read and 50% Write
- Data Warehouse (Sequential) 0% Read and 100% Write
- Mixed OLTP and Data Warehouse (Sequential) 70% Read and 30% Write
- Mixed OLTP and Data Warehouse (Sequential) 50% Read and 50% Write

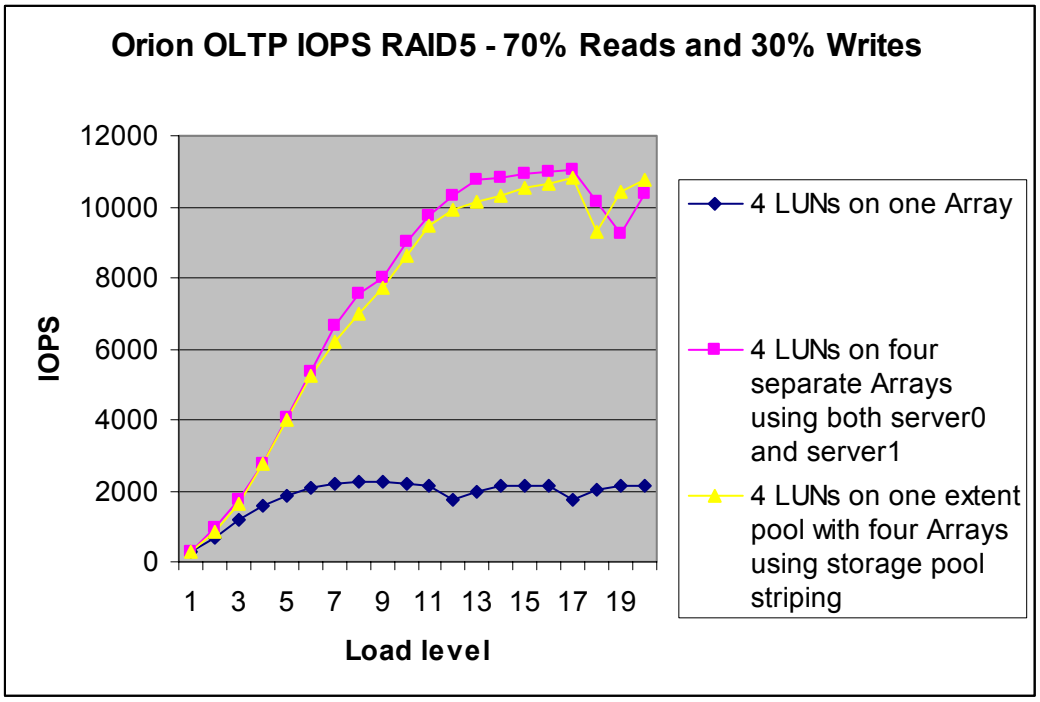
14.2 Summary and Analysis

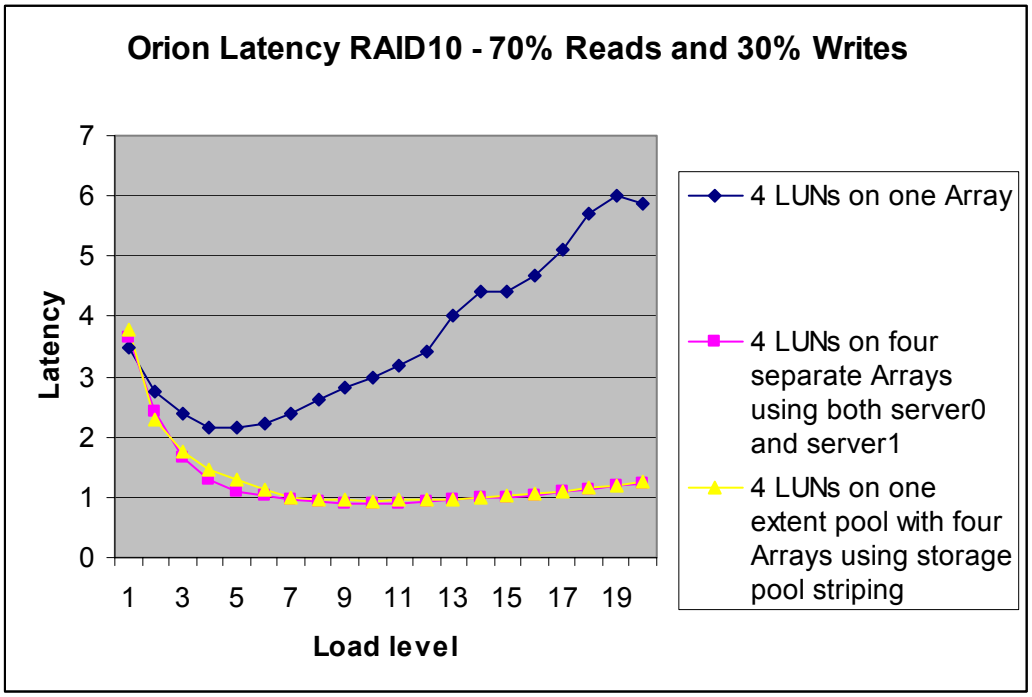
This section will compare the results of running Orion using the three different LUN configurations. The test results from putting all four LUNs on one array, which is storage configuration (1), were discussed in detail in the section **13 Benchmarking Exercise – Comparison of RAID-5 and RAID-10**. Those results will now be compared to running the exact same tests on storage configurations (2) and (3). As mentioned previously, storage configuration (1) is the least optimal configuration. The point of this exercise is to show the dramatic performance improvement obtained by implementing the principles of Workload Isolation and Workload Spreading with regard to LUN configuration.

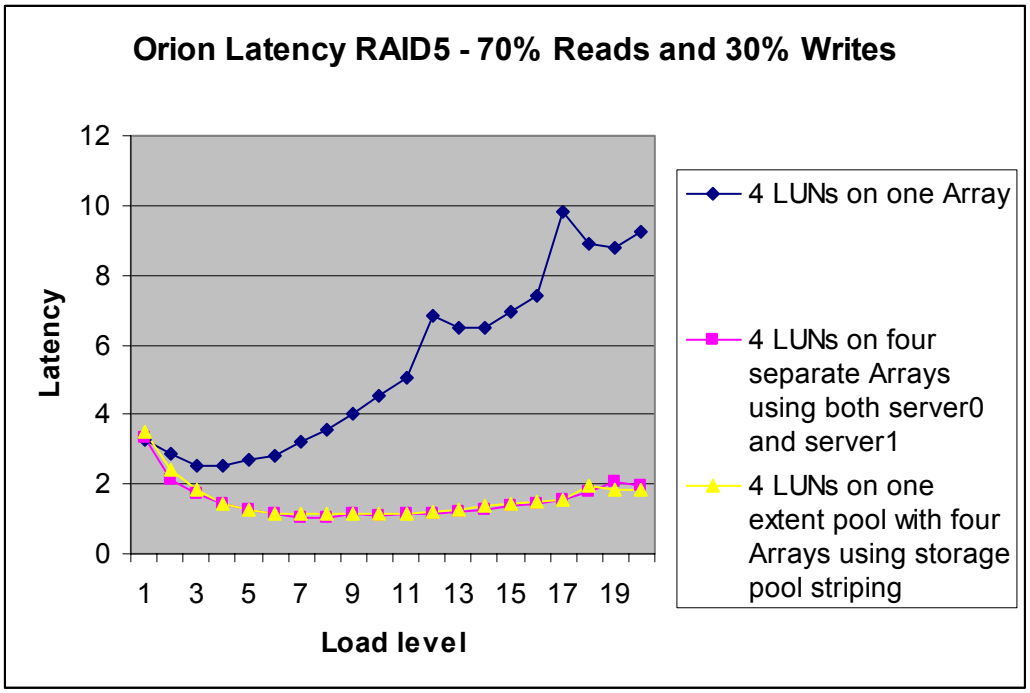
14.2.1 OLTP Workloads

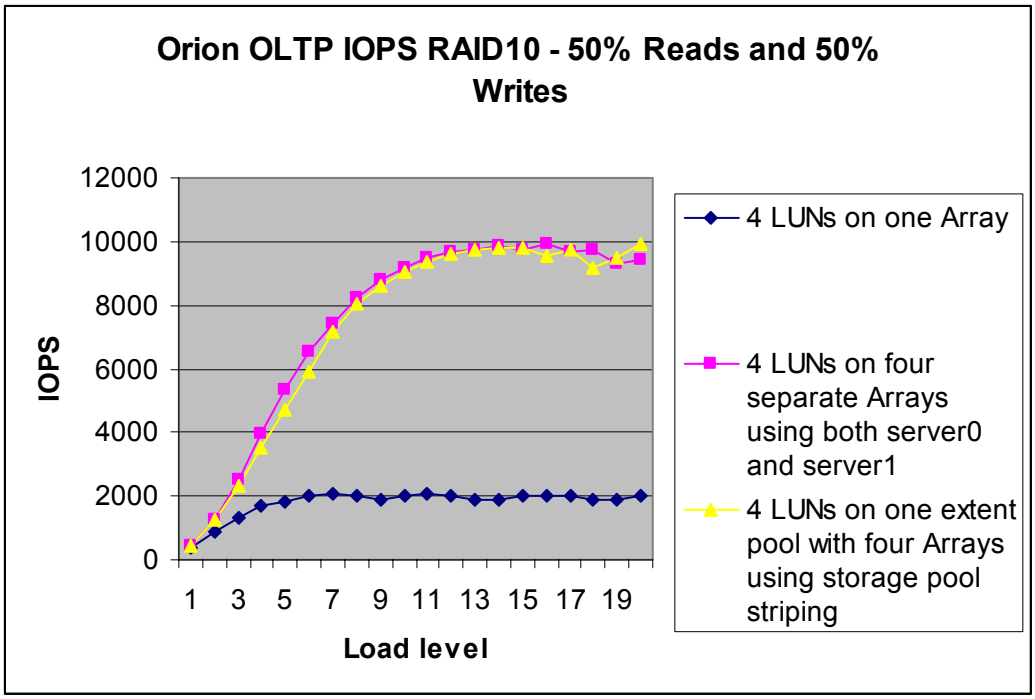
- For pure OLTP workloads, the IOPS and Latency for storage configurations (2) and (3) are almost identical at all load levels. There is no significant difference at all.
- For pure OLTP workloads, the IOPS and Latency performance numbers for storage configurations (2) and (3) are dramatically better than for storage configuration (1), i.e., at anything but the lowest load levels the IOPS and Latency benefits are 400% - 600%. This corresponds to the fact that four arrays are being used as opposed to one array and Workload Spreading at the array level is being used to great advantage.
- The significant advantage of RAID-10 over RAID-5 in pure OLTP environments becomes even more pronounced at anything but the very lowest load levels when using storage configurations (2) and (3). For the OLTP 70% Read and 30% Write workload, there is a 2,000 - 6,000 IOPS advantage for RAID-10 over RAID-5 at the middle to highest load levels. For the OLTP 50% Read and 50% Write workload, there is a 3,500 - 4,500 IOPS advantage for RAID-10 over RAID-5 at the middle to highest load levels.
- The fact that storage configurations (2) and (3) are identical in performance for pure OLTP workloads would favor storage configuration (3) (storage pool striping) as the solution for such an OLTP environment. With storage pool striping, when ASM diskgroup space expansion is required, a LUN could be carved out of the multi-rank extent pool and added as an ASM disk and no array-level hot spots would be created. This would not be the case with storage configuration (2) where adding just one ASM disk (LUN) would entail obtaining additional space from one RAID array. This would immediately load that array more than the other arrays being used in the ASM disk group. The only workaround for this characteristic of storage configuration (2) would be to always add equally sized LUNs from all of the arrays at the same time. This is not as simple an administrative task as adding space when using storage configuration (3).
- Below are the RAID-10 and RAID-5 IOPS and Latency graphs of the Orion data points for the OLTP 70% Read and 30% Write and the 50% Read and 50% Write workloads:

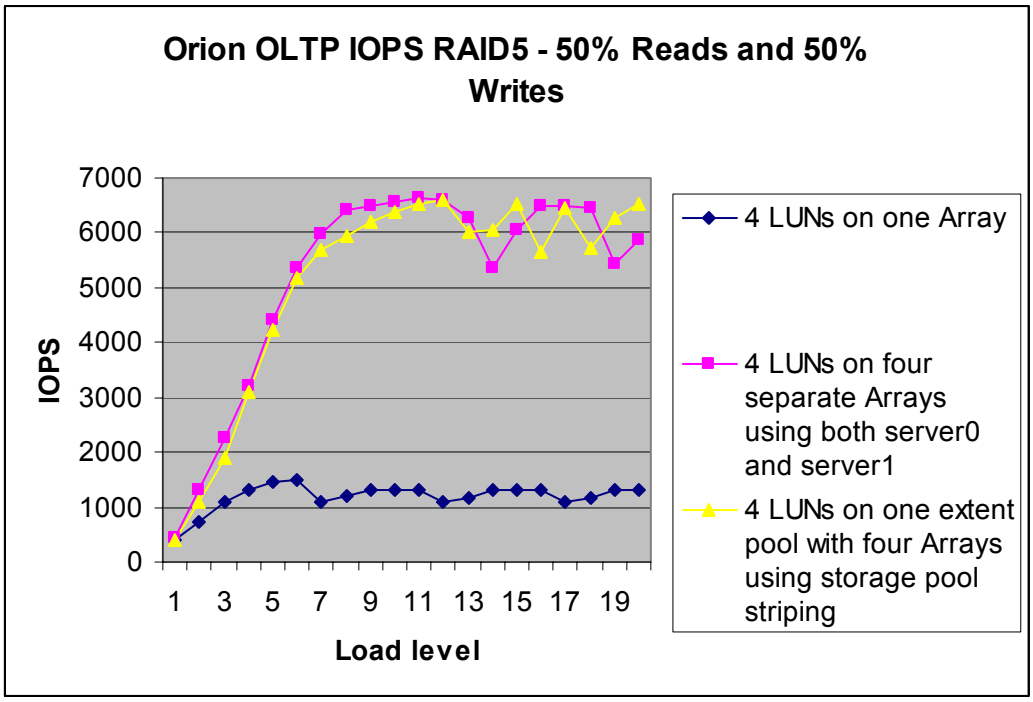


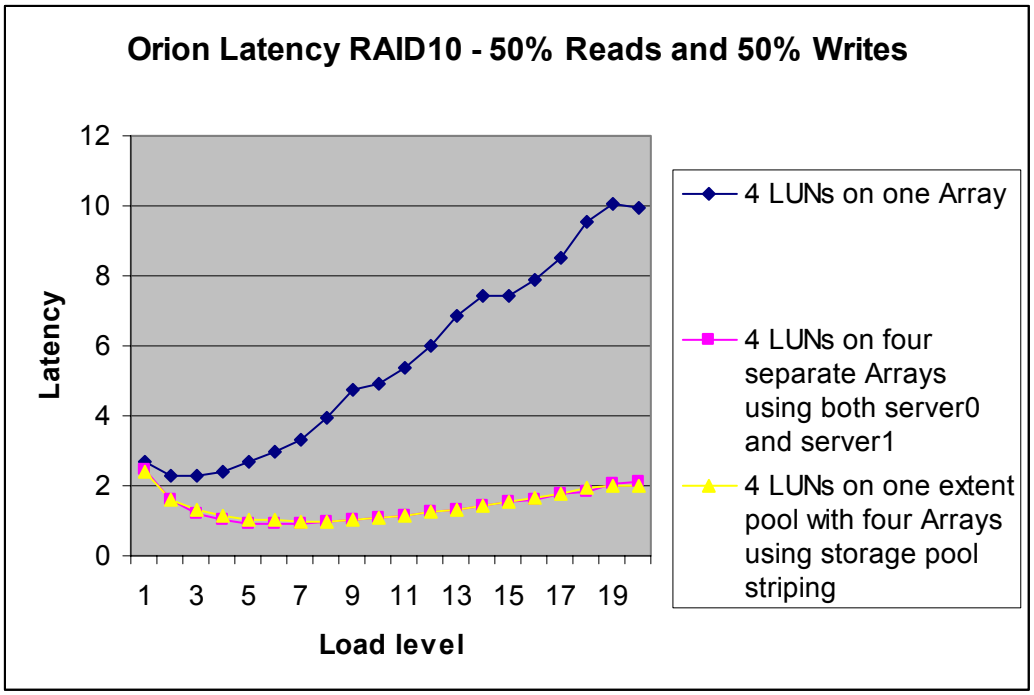


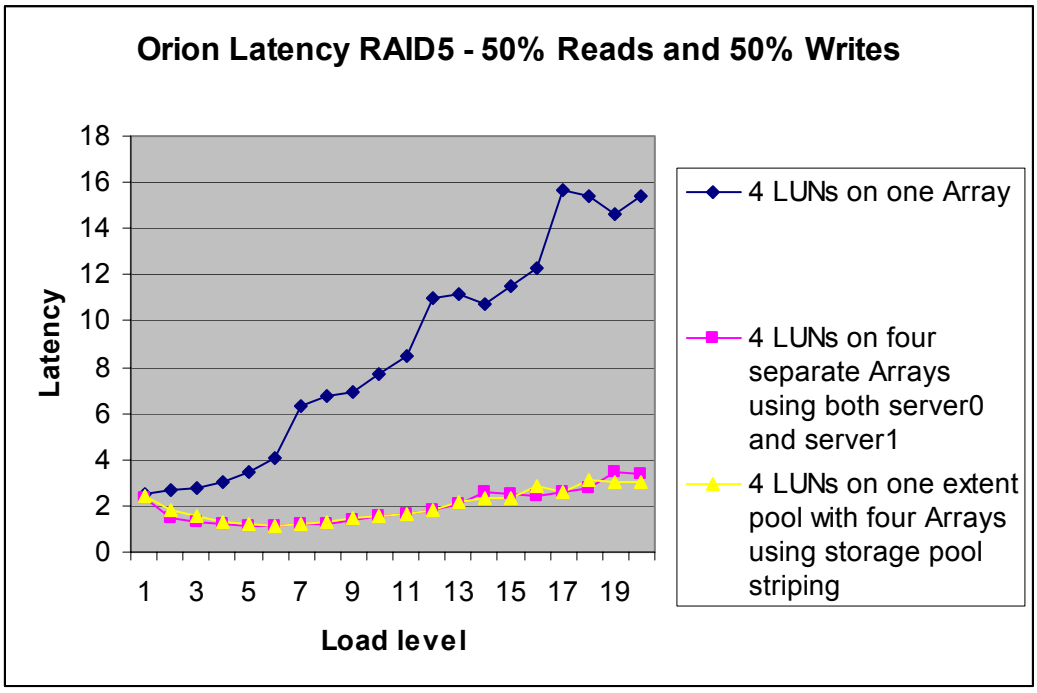












14.2.2 Data Warehouse (Sequential) Workloads

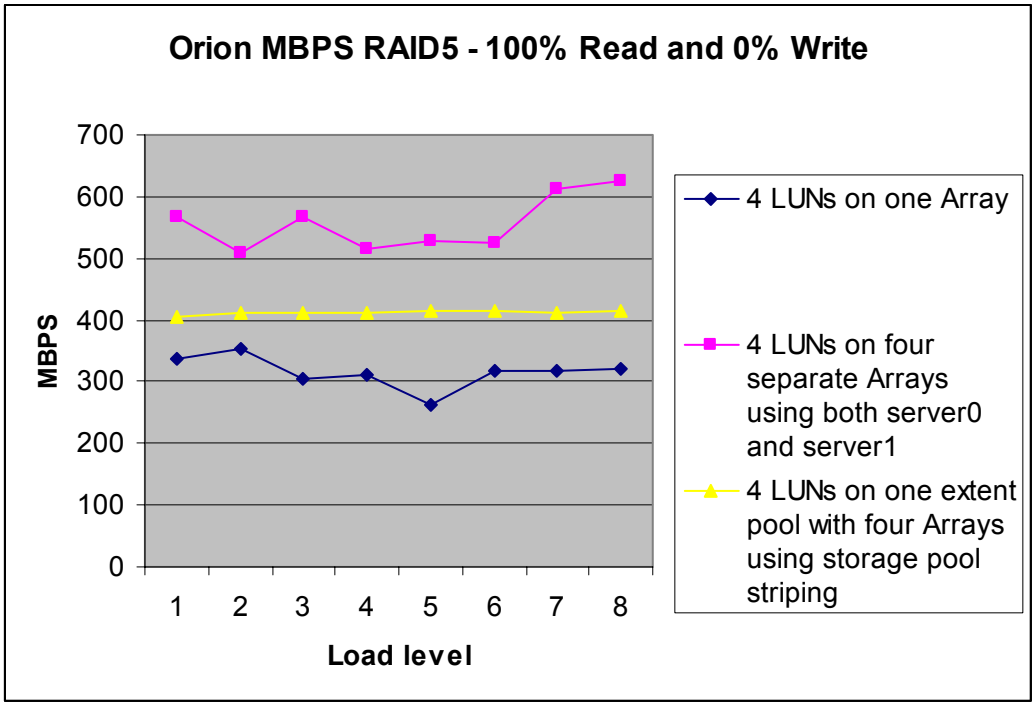
- For pure Data Warehouse (Sequential) workloads, storage configuration (2) performs best, followed by storage configuration (3) and then lastly storage configuration (1). This shows that sequential workloads benefit the most from the total Workload Spreading that is offered by storage

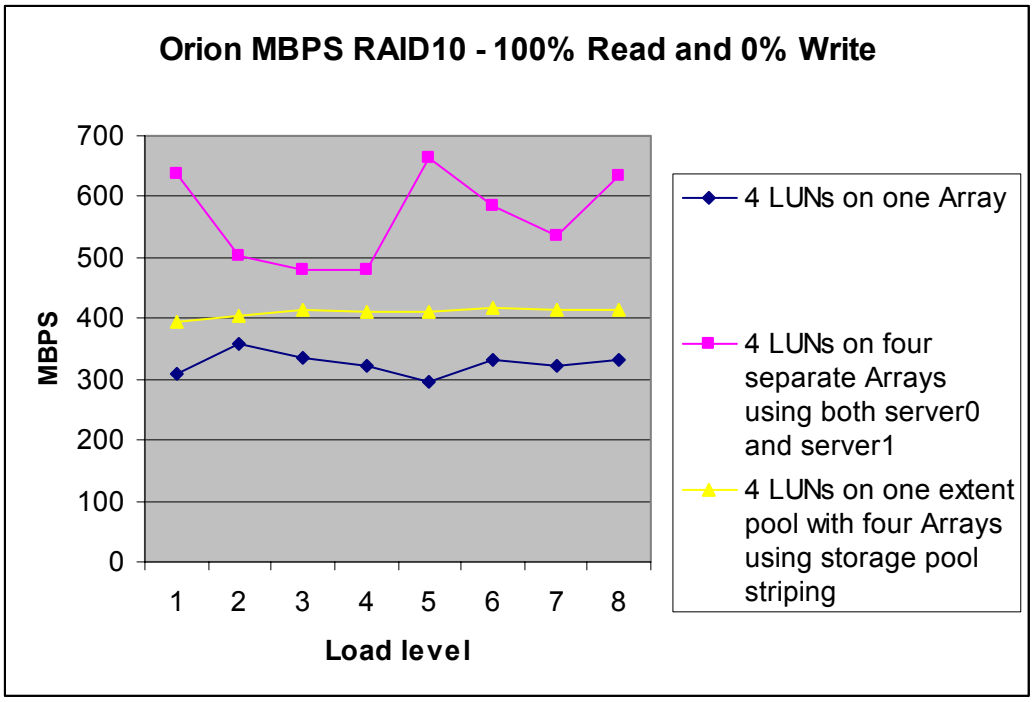
configuration (2) where both DS8000 servers and Device Adapters are being utilized in addition to four RAID arrays. The storage pool striping of storage configuration (3) restricts all four RAID arrays to one DS8000 server so it does not get quite the throughput of storage configuration (2).

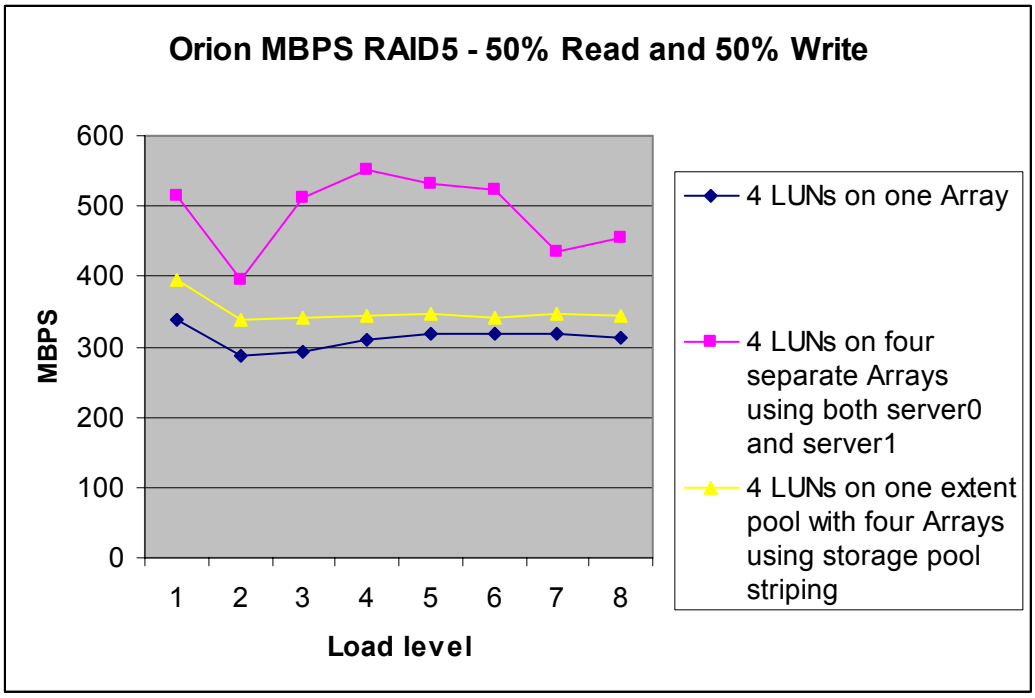
- The performance advantage of RAID-5 over RAID-10 for sequential writes is still very pronounced when using either storage configuration (2) or storage configuration (3) and the workload mix is 0% Read and 100% Write. It is not quite as clear a picture when the mix is 50% Read and 50% Write. The performance numbers for RAID-5 versus RAID-10 are sometimes higher and sometimes lower with that workload mix. The 100% Read and 0% Write workload mix shows the expected relatively close numbers for RAID-5 and RAID-10 that were discussed in the RAID-5 versus RAID-10 benchmarking exercise.

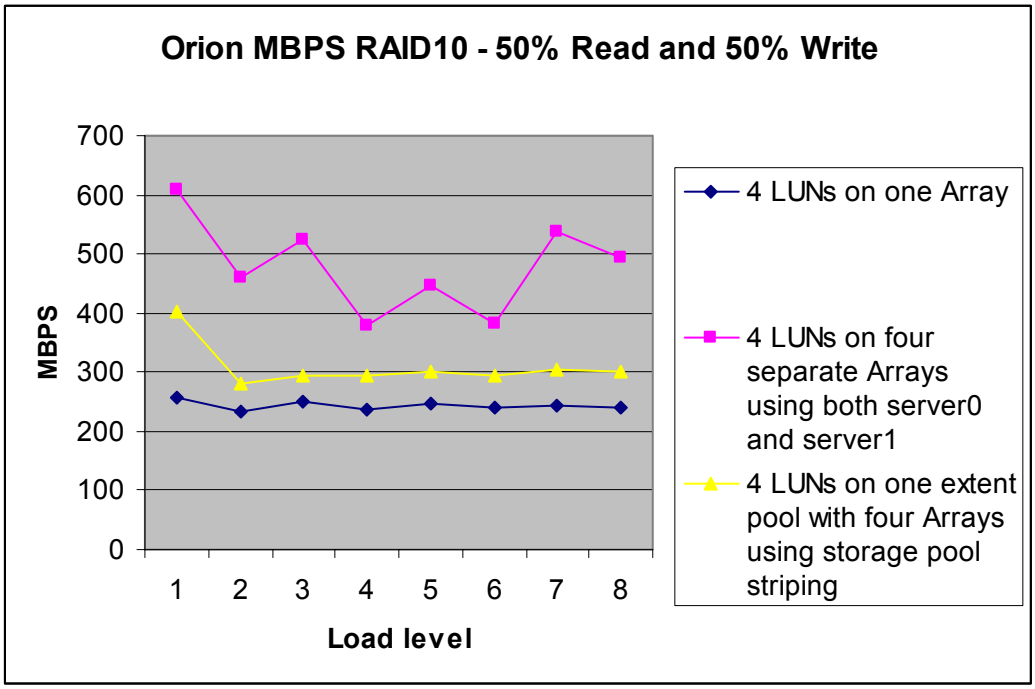
Below are the MBPS in tabular form for RAID-5 and RAID-10 at the various sequential workload mixes. However, it is easier to make comparisons by viewing the graphs included below.

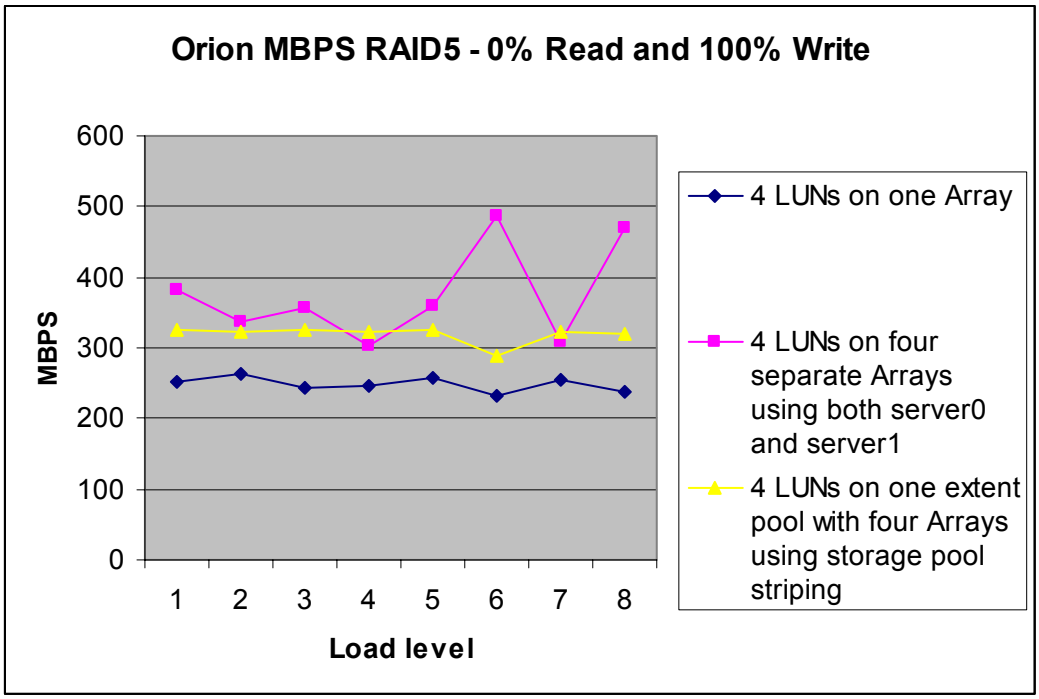
Read/Write ratios	MBPS Config (1)	MBPS Config (2)	MBPS Config (3)
RAID-5 100%/0%	262.45 - 354.03	510.28 - 626.4	405.95 - 414.86
RAID-10 100%/0%	296.32 - 358.74	478.68 - 664.02	395.13 - 416.38
RAID-5 50%/50%	286.47 - 339.62	396.48 - 551.51	339.32 - 394.2
RAID-10 50%/50%	234.38 - 255.54	378.75 - 607.54	279.78 - 401.65
RAID-5 0%/100%	232.85 - 264.38	303.98 - 488	287.67 - 326.46
RAID-10 0%/100%	162.55 - 212.77	287.83 - 420.98	261.55 - 264.3

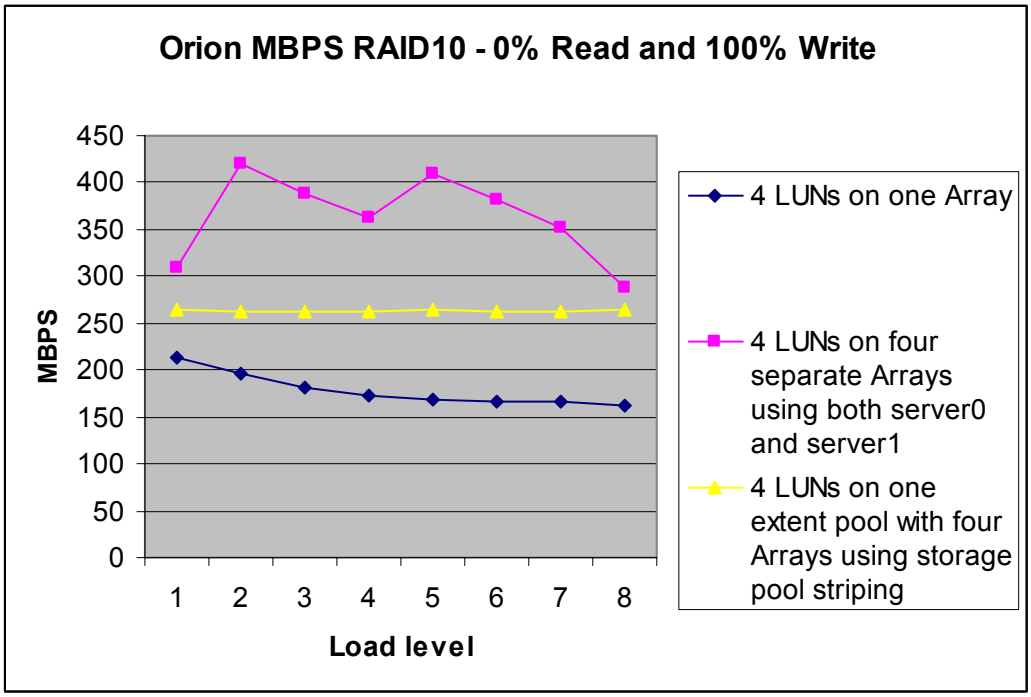






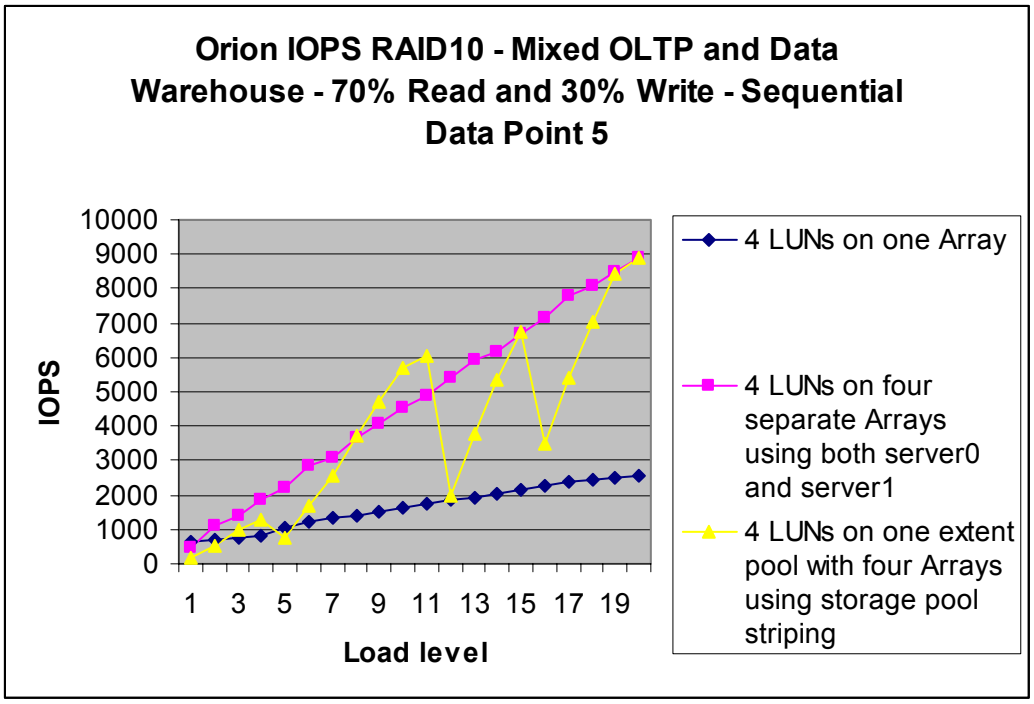


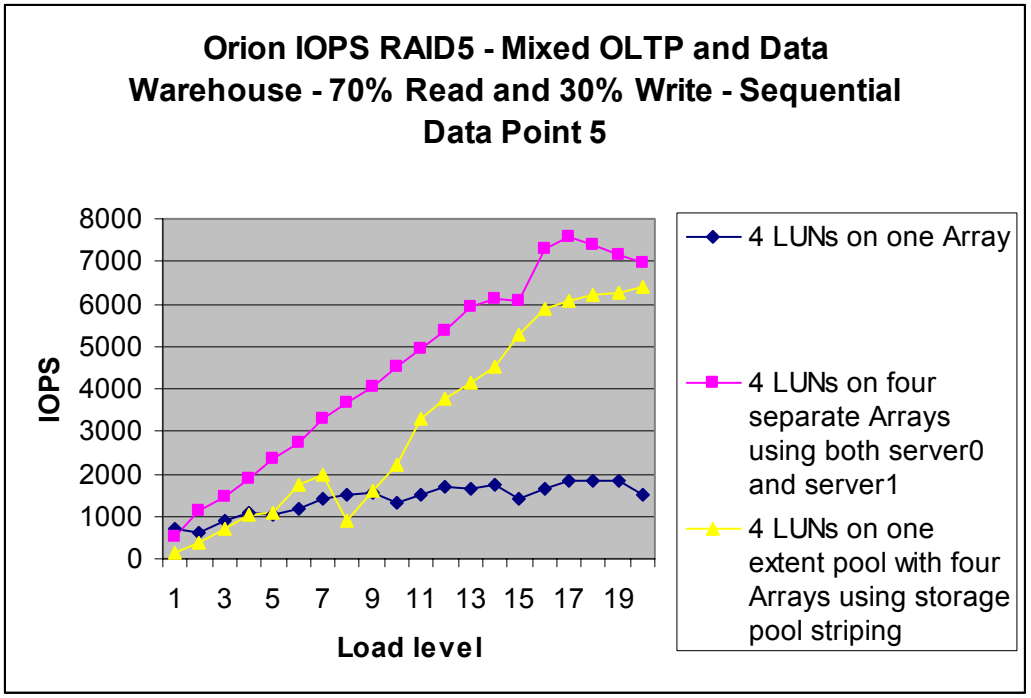


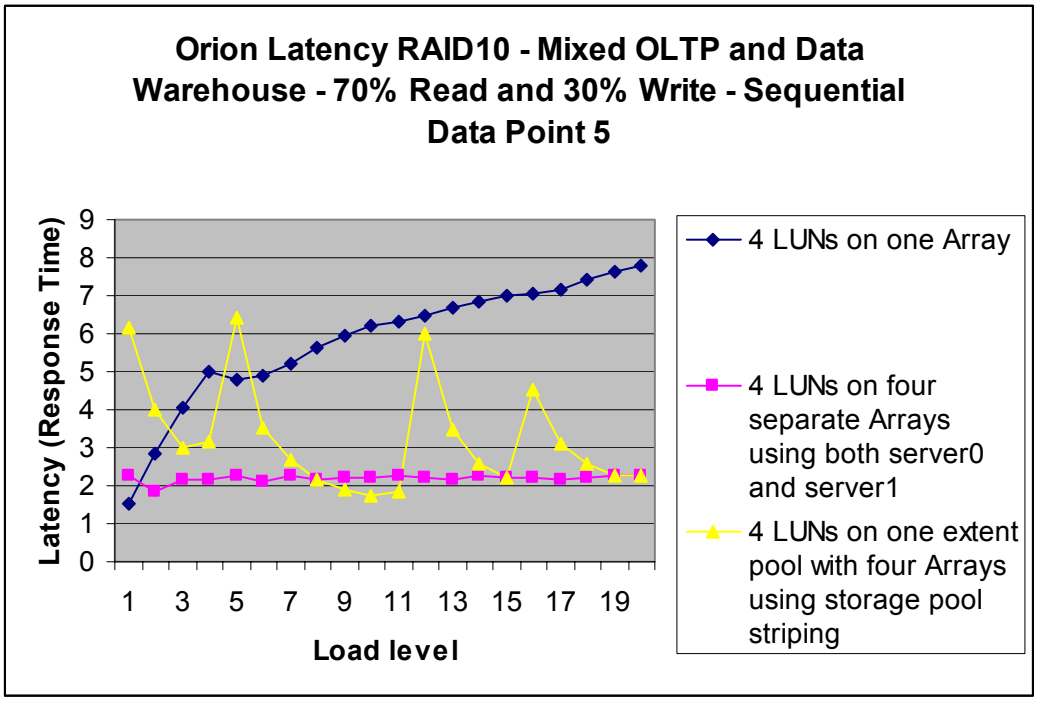


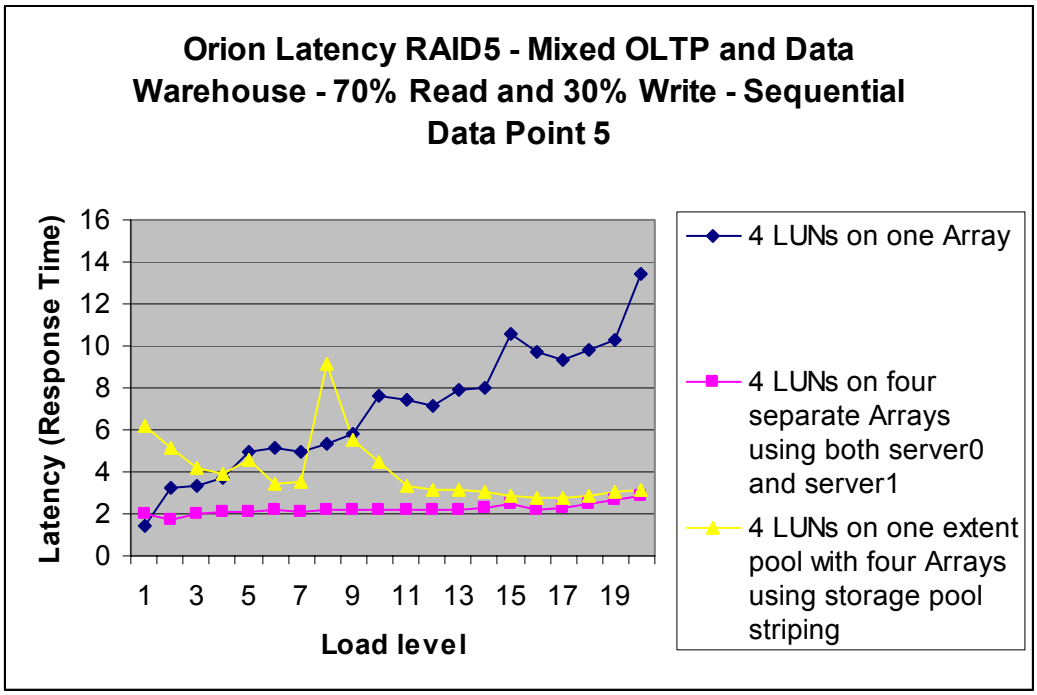
14.2.3 Mixed OLTP and Data Warehouse Workloads

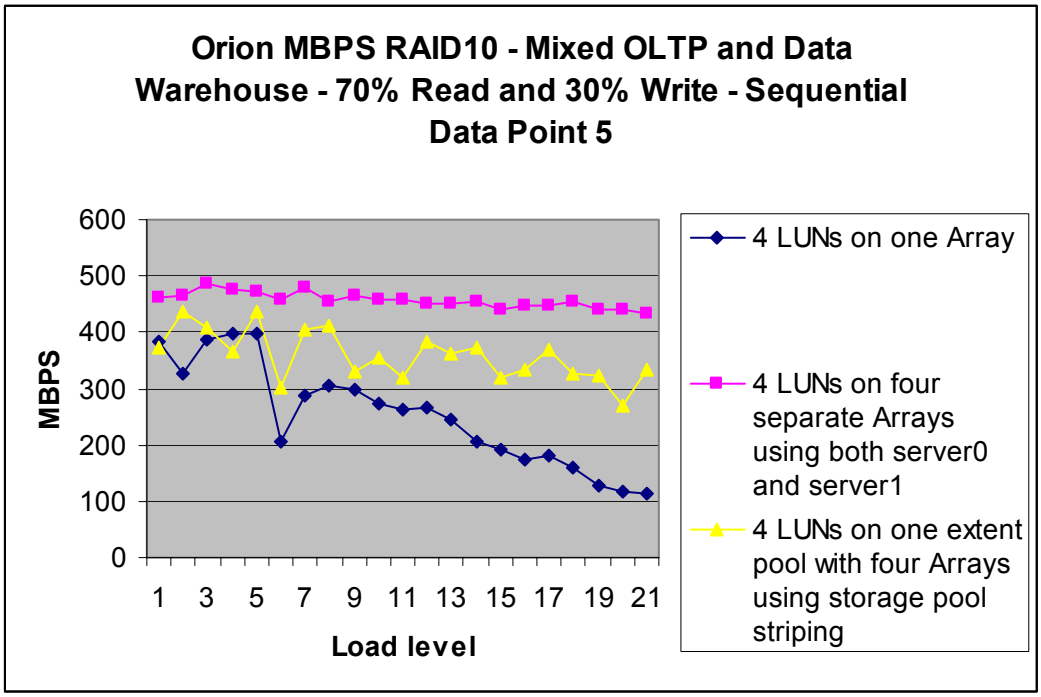
- As in the pure Data Warehouse (Sequential) workloads discussed in section 14.2.2, storage configuration (2) performs best for Mixed OLTP and Data Warehouse workloads, followed by storage configuration (3) and lastly storage configuration (1).
- As can be seen from the graphs included below, the performance of storage configurations (2) and (3) is very close. However, storage configuration (2) shows a much steadier increase in IOPS and MBPS over the full workload range while storage configuration (3) seems to be less consistent and shows up and down spikes over the full workload range
- RAID-5 and RAID-10 seem to perform very similarly for IOPS, Latency and MBPS with this workload mix, i.e., the IOPS for RAID-10 are not significantly better than RAID-5 and the MBPS for RAID-5 are not significantly better than RAID-10. This indicates that using mixed OLTP and Data Warehouse workloads with storage configurations (2) and (3) is not an optimal configuration at high load levels since the respective advantages of RAID-5 and RAID-10 for different types of workloads is nullified. If possible, it is best to separate OLTP and Data Warehouse workloads, although this must be weighed against the possible increased administrative costs of attempting to always keep such workloads separate in a production environment.

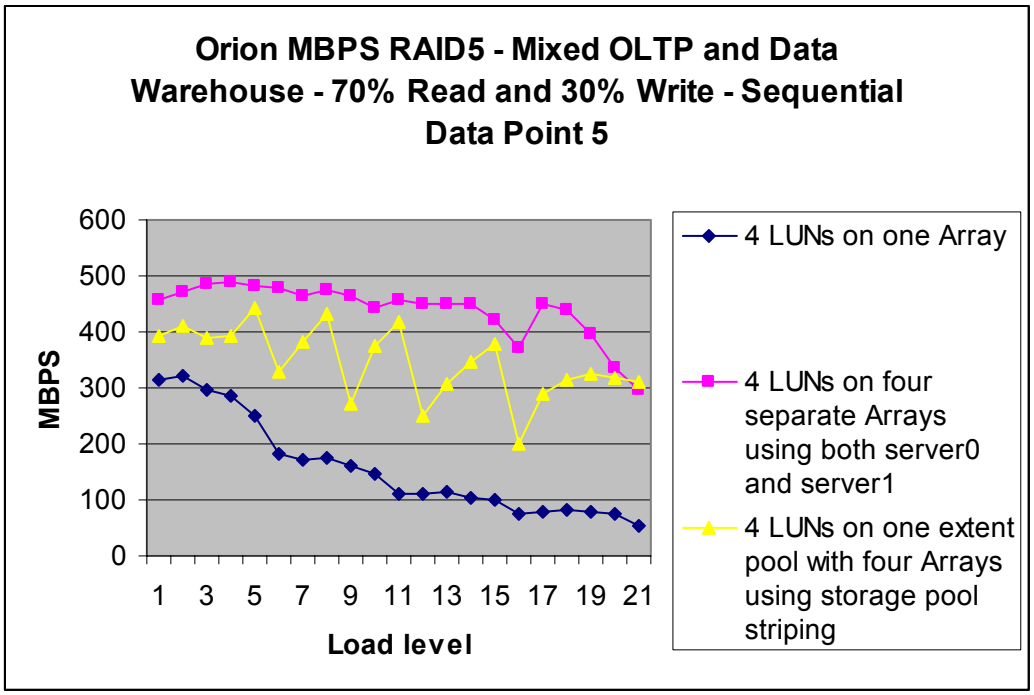












15 Summary

This paper has discussed the performance differences that can be expected between RAID-5 and RAID-10 when running Oracle with Automatic Storage Management on the IBM DS8000. The paper

has also discussed performance variations depending on LUN configurations across RAID arrays. It has attempted to show via the usage of Orion, the Oracle I/O Numbers calibration tool, these performance differences for various types of Oracle workloads. The usage of Orion allows the testing to simulate an ASM deployment without having to install the Oracle database software and without the additional complexity of tuning the Oracle server. Thus it is a controlled experiment which isolates the testing strictly to the I/O layer of an Oracle implementation.

The primary premise of the paper was that the performance testing done by the IBM storage performance group for the DS8000 always includes database-type I/O and that the assertions made in the DS8000 Redbooks concerning the differences between RAID-5 and RAID-10 should be verifiable when an Oracle-specific test is done. The results of the testing done for this paper confirm those assertions regarding performance. In addition, the general principles of Workload Isolation, Workload Resource Sharing and Workload Spreading should pertain as much to Oracle database workloads as to any other type of I/O workload. The results documented in this paper also confirm those general principles. Finally, the benchmark runs have resulted in specific metrics that can be used as a concrete basis for comparison.

16 Appendix A: Oracle I/O profile scripts

The following SQL*Plus scripts can be run on an Oracle database to determine the I/O profile.

This first script is from the Oracle OpenWorld 2007 presentation cited in the References and which originated from Luca Canali at CERN. It can be run on Oracle Database 10g and above. If it is run on a RAC database, it will automatically sum the results for all member instances. It has been modified for formatting and to include the read/write ratios:

```
set lines 120
set pages 9999
col Begin_Time heading 'Begin Time' justify center
col End_Time heading 'End Time' justify center
col Physical_Read_IOPS heading 'Read IOPS' justify center
col Physical_Read_IOPS format 99,999
col Physical_Write_IOPS heading 'Write IOPS' justify center
col Physical_Write_IOPS format 99,999
col IOPS_Read_Percentage heading 'IOPS Read|Percentage' justify center
col Physical_Read_Total_Bps heading 'Read Bytes|per Second' justify center
col Physical_Read_Total_Bps format 9,999,999,999
col Physical_Write_Total_Bps heading 'Write Bytes|per Second' justify center
col Physical_Write_Total_Bps format 9,999,999,999
col Bps_Read_Percentage heading 'Bps Read|Percentage' justify center

alter session set nls_date_format='MM/DD/YYYY HH24:MI';

spool io_profile.lst

select min(begin_time) Begin_Time,
      max(end_time) End_Time,
      sum(case metric_name
```

```
        when 'Physical Read Total IO Requests Per Sec'
        then round(average) end) Physical_Read_IOPS,
    sum(case metric_name
        when 'Physical Write Total IO Requests Per Sec'
        then round(average) end) Physical_Write_IOPS,
    round((sum(case metric_name
         when 'Physical Read Total IO Requests Per Sec'
         then round(average) end) * 100) /
        (sum(case metric_name
              when 'Physical Read Total IO Requests Per Sec'
              then round(average) end) +
         sum(case metric_name
              when 'Physical Write Total IO Requests Per Sec'
              then round(average) end))) IOPS_Read_Percentage,
    sum(case metric_name
        when 'Physical Read Total Bytes Per Sec'
        then round(average) end) Physical_Read_Total_Bps,
    sum(case metric_name
        when 'Physical Write Total Bytes Per Sec'
        then round(average) end) Physical_Write_Total_Bps,
    round((sum(case metric_name
         when 'Physical Read Total Bytes Per Sec'
         then round(average) end) * 100) /
        (sum(case metric_name
              when 'Physical Read Total Bytes Per Sec'
              then round(average) end) +
         sum(case metric_name
              when 'Physical Write Total Bytes Per Sec'
              then round(average) end))) Bps_Read_Percentage
from dba_hist_sysmetric_summary
  group by snap_id
  order by snap_id;
spool off
```

The following SQL*Plus script is from James Koopmann at http://www.jameskoopmann.com/?p=9. It has been modified to work properly on a RAC database and also for some content and formatting:

```
----------------------------------------------------
--- This script is modified for formatting and some content
--- from the following:
--# Script  : wrh_sysstat_ioworkload_ALL.sql
--# Author  : thecheapdba
--# Tested  : Oracle 10.2
--# Version : 2007/08/07
--# Purpose : Report on IOPS & MBPS over a period of time as seen by DB.
--# ---------------------------------------------------------------------
--# NOTES   : cut and paste into a worksheet for graphing
--# ---------------------------------------------------------------------

set echo     off
set feedback off
set linesize 120
set pagesize 55
```

```
set verify   off

set termout off
column rpt new_value rpt
select name || '_wrh_sysstat_ioworkload.lst' rpt
  from v$database;
set termout on
prompt
prompt
prompt ^^^^^^^^^^^^^
prompt Report Name : &&rpt
prompt ^^^^^^^^^^^^^
spool &&rpt

column end_time format a20
column end_time justify center head "Snap End Time"
column beg_id format 9,999
column beg_id justify center head "Snap ID"
column instance_name format A9
column instance_name justify center head "Instance"
column sri justify center head "Small|Read|IOPS" noprint
column swi justify center head "Small|Write|IOPS" noprint
column tsi justify center head "Total|Small IOPS"
column srp justify center head "Read I/O%|Small IOPS"
column swp justify center head "Small|Write I/O%" noprint
column lri justify center head "Large|Read IOPS" noprint
column lwi justify center head "Large|Write IOPS" noprint
column tli justify center head "Total|Large IOPS"
column lrp justify center head "Read I/O%|Large IOPS"
column lwp justify center head "Large|Write I/O%" noprint
column tr  justify center head "Total|Read MBPS"
column tw  justify center head "Total|Written|MBPS"
column tm  justify center head "Total MBPS"

SELECT to_char(end_time, 'MM/DD/YYYY HH24:MI:SS') end_time,
      beg_id,
      instance_name,
      ROUND(sr/inttime, 3) sri,
      ROUND(sw/inttime, 3) swi,
      ROUND((sr+sw)/inttime, 3) tsi,
      ROUND(sr/DECODE((sr+sw), 0, 1, (sr+sw))*100, 3) srp,
      ROUND(sw/DECODE((sr+sw), 0, 1, (sr+sw))*100, 3) swp,
      ROUND(lr/inttime, 3) lri,
      ROUND(lw/inttime, 3) lwi,
      ROUND((lr+lw)/inttime, 3) tli,
      ROUND(lr/DECODE((lr+lw), 0, 1, (lr+lw))*100, 3) lrp,
      ROUND(lw/DECODE((lr+lw), 0, 1, (lr+lw))*100, 3) lwp,
      ROUND((tbr/inttime)/1048576, 3) tr,
      ROUND((tbw/inttime)/1048576, 3) tw,
      ROUND(((tbr+tbw)/inttime)/1048576, 3) tm
FROM (SELECT beg.snap_id beg_id,
            beg.instance_number,
            beg.instance_name,
            end.snap_id end_id,
```

```
       beg.begin_interval_time,
       beg.end_interval_time,
       end.begin_interval_time begin_time,
       end.end_interval_time end_time,
       (extract(day    from (end.end_interval_time - end.begin_interval_time)) * 86400)+
       (extract(hour   from (end.end_interval_time - end.begin_interval_time)) * 3600)+
       (extract(minute from (end.end_interval_time - end.begin_interval_time)) * 60)+
       (extract(second from (end.end_interval_time - end.begin_interval_time)) * 01) inttime,
       decode(end.startup_time, end.begin_interval_time, end.sr, (end.sr - beg.sr))    sr,
       decode(end.startup_time, end.begin_interval_time, end.sw, (end.sw - beg.sw))    sw,
       decode(end.startup_time, end.begin_interval_time, end.lr, (end.lr - beg.lr))    lr,
       decode(end.startup_time, end.begin_interval_time, end.lw, (end.lw - beg.lw))    lw,
       decode(end.startup_time, end.begin_interval_time, end.tbr, (end.tbr - beg.tbr)) tbr,
       decode(end.startup_time, end.begin_interval_time, end.tbw, (end.tbw - beg.tbw)) tbw
  FROM (SELECT dba_hist_snapshot.snap_id,
               gv$instance.instance_number,
               gv$instance.instance_name,
               dba_hist_snapshot.startup_time,
               begin_interval_time,
               end_interval_time,
               sum(decode(stat_name, 'physical read total IO requests', value, 0) -
                  decode(stat_name, 'physical read total multi block requests', value, 0)) sr,
               sum(decode(stat_name, 'physical write total IO requests', value, 0) -
                  decode(stat_name, 'physical write total multi block requests', value, 0)) sw,
               sum(decode(stat_name, 'physical read total multi block requests', value, 0)) lr,
               sum(decode(stat_name, 'physical write total multi block requests', value, 0)) lw,
               sum(decode(stat_name, 'physical read total bytes', value, 0)) tbr,
               sum(decode(stat_name, 'physical write total bytes', value, 0)) tbw
          FROM wrh$_sysstat, wrh$_stat_name, dba_hist_snapshot, gv$instance
          WHERE wrh$_sysstat.stat_id = wrh$_stat_name.stat_id
            AND wrh$_sysstat.snap_id = dba_hist_snapshot.snap_id
            AND wrh$_sysstat.instance_number = dba_hist_snapshot.instance_number
            AND gv$instance.instance_number = dba_hist_snapshot.instance_number
          GROUP BY dba_hist_snapshot.snap_id, gv$instance.instance_number,
                   gv$instance.instance_name, dba_hist_snapshot.startup_time,
                   begin_interval_time, end_interval_time) beg,
        (SELECT dba_hist_snapshot.snap_id,
               gv$instance.instance_number,
               gv$instance.instance_name,
               dba_hist_snapshot.startup_time,
               begin_interval_time,
               end_interval_time,
               sum(decode(stat_name, 'physical read total IO requests', value, 0) -
                  decode(stat_name, 'physical read total multi block requests', value, 0)) sr,
               sum(decode(stat_name, 'physical write total IO requests', value, 0) -
                  decode(stat_name, 'physical write total multi block requests', value, 0)) sw,
               sum(decode(stat_name, 'physical read total multi block requests', value, 0)) lr,
               sum(decode(stat_name, 'physical write total multi block requests', value, 0)) lw,
               sum(decode(stat_name, 'physical read total bytes', value, 0)) tbr,
               sum(decode(stat_name, 'physical write total bytes', value, 0)) tbw
          FROM wrh$_sysstat, wrh$_stat_name, dba_hist_snapshot, gv$instance
          WHERE wrh$_sysstat.stat_id = wrh$_stat_name.stat_id
            AND wrh$_sysstat.snap_id = dba_hist_snapshot.snap_id
            AND wrh$_sysstat.instance_number = dba_hist_snapshot.instance_number
```

```
          AND gv$instance.instance_number = dba_hist_snapshot.instance_number
        GROUP BY dba_hist_snapshot.snap_id, gv$instance.instance_number,
              gv$instance.instance_name, dba_hist_snapshot.startup_time,
              begin_interval_time, end_interval_time) end
    WHERE beg.snap_id + 1 = end.snap_id
     AND beg.instance_number = end.instance_number
)
order by 2, 3;
```

17 Appendix B: References

1. IBM System storage DS8000 Series: Architecture and Implementation
   http://www.redbooks.ibm.com/redbooks/pdfs/sg246786.pdf

2. IBM TotalStorage DS8000 Series: Performance Monitoring and Tuning SG24-7146-00
   http://www.redbooks.ibm.com/redbooks/pdfs/sg247146.pdf

3. Implementing an IBM/Brocade SAN
   http://www.redbooks.ibm.com/redbooks/pdfs/sg246116.pdf

4. SAN Volume Controller: Best Practices and Performance Guidelines SG24-7521-00
   http://www.redbooks.ibm.com/redbooks/pdfs/sg247521.pdf

5. Monitoring Your Storage Using TotalStorage Productivity Center
   http://www.redbooks.ibm.com/redbooks/pdfs/sg247364.pdf

6. Performance Metrics in TotalStorage Productivity Center Performance Reports
   http://www.redbooks.ibm.com/redpapers/pdfs/redp4347.pdf

7. IBM System Storage Command Line Interface User's Guide
   http://www-1.ibm.com/support/docview.wss?uid=ssg1S7001162&loc=en_US&cs=utf-8&lang=en

8. Oracle Orion download site (includes Orion User's Guide)
   http://www.oracle.com/technology/software/tech/orion/index.html

9. Oracle Database Storage Administrator's Guide
   11g Release 1 (11.1)
   Part Number B31107-04
   http://download.oracle.com/docs/cd/B28359_01/server.111/b31107/toc.htm

10. Oracle Automatic Storage Management
    Under-the-Hood & Practical Deployment Guide
    Nitin Vengurlekar, Murali Vallath, Rich Long
    Oracle Press
    ISBN 978-0-07-149607-0

11. Automatic Storage Management Technical Overview
    An Oracle White Paper
    November 2003
    http://www.oracle.com/technology/products/manageability/database/pdf/asmwp.pdf

12. Oracle OpenWorld 2007 Presentation -

RAC PACK
Back-of-the-Envelope Database Storage Design
Nitin Vengurlekar
http://www.oracle.com/technology/products/database/asm/pdf/back%20of%20the%20env%20by%20nitin%20oow%202007.pdf

13. The Optimal Oracle Configuration: Using the Oracle Orion Workload Tool to Accurately Configure Storage
    Technical White Paper – LSI Corporation
    James F. Koopmann
    http://www.lsi.com/documentation/storage/business_solutions/25034-00_RevA_Oracle_Opitmal_Config.pdf
    ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/tsw03006usen/TSW03006USEN.PDF

14. Oracle – Complete history of IOPS & MBPS from workload repository history (snapshots)
    James Koopmann
    http://www.jameskoopmann.com/?p=9

18 Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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

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

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

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Users of this document should verify the applicable data for their specific environment.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

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.