Maintaining End-to-End Service Levels for VMware Virtual Machines Using VMware DRS and EMC Navisphere QoS

Applied Technology

Abstract

This white paper describes tests in which Navisphere® QoS Manager and VMware's Distributed Resource Scheduler ran in a VMware ESX virtual environment using EMC® CLARiiON® storage systems. These tests clearly demonstrate that, with QoS Manager and Distributed Resource Scheduler, multiple applications running in a VMware ESX virtual machine can maintain exceptional mission-critical performance while sharing a CLARiiON storage system.

January 2008

Maintaining End-to-End Service Levels for VMware Virtual Machines Using VMware DRS
And EMC Navisphere QoS
Applied Technology

Table of Contents

Executive summary

With the increasing deployment of VMware ESX servers for mission-critical applications, it becomes even more important to maintain the performance of these applications in virtual machines. The complexity of virtualization, especially when server and storage resources are shared across multiple components, makes it more difficult for an administrator to detect and solve performance problems.

Using CLARiiON®'s Navisphere® Quality of Service (NQM) product and VMware's Distributed Resource Scheduler allows you to maintain application service levels in virtual machines. This greatly minimizes the time and resources needed to manage and fix performance issues.

Introduction

Shared usage of a server and storage system by multiple applications allows consolidation and better utilization of resources. However, sharing server and storage resources also creates workload contentions and the challenge of maintaining sustained performance for mission-critical applications.

To minimize workload contention impacts across a shared infrastructure, VMware introduced Distributed Resource Scheduler (DRS) with VirtualCenter 2.0. DRS automatically monitors CPU and memory resources on virtual machines to guarantee application service levels.

Beginning with CLARiiON release 24, EMC® Navisphere® Management Suite includes Navisphere QoS Manager (NQM). NQM allows you to cost-effectively maintain storage I/O performance for critical applications running in virtual machines.

NQM (which measures, monitors, and controls application I/O performance) and DRS (which prioritizes CPU and memory resources) provide an exceptional service level guarantee for applications in virtual machines.

This white paper highlights the benefits of using NQM with DRS. It also describes tests (using Oracle as the application) that illustrate how to achieve superior service levels across storage and server components.

Audience

This white paper is intended for any EMC customer, partner, or employee seeking information about how to use NQM in Oracle database deployments. This paper assumes the reader is knowledgeable about Oracle performance characteristics, CLARiiON storage system fundamentals, and NQM functionalities. Information about NQM functionalities can be found in the *Navisphere Quality of Service Manager (NQM) – Applied Technology* white paper, available at EMC.com and the EMC Powerlink® website.

Readers should also have a good understanding of VMware Infrastructure 3. For information on VMware Infrastructure 3, please see www.vmware.com.

Terminology

Bandwidth: The average amount of read/write data in megabytes that is passed through the storage system per second.

Logical unit number (LUN): A unique identifier that is used to distinguish logical storage objects in a storage system.

Oracle Automated Stress Testing (OAST): An automated test suite designed to build OLTP-type workloads for systems using an Oracle database. It creates tables, performs stress test runs, and outputs transaction-related performance data.

Queue depth: The average number of requests within a polling interval that are waiting to be serviced by the storage, including the one currently in service.

Redundant Array of Independent Disks (RAID): A way of storing the same data in several places on multiple hard disks. I/O operations can overlap in a balanced way, thereby improving performance. Since multiple disks increase the mean time between failures, storing data redundantly also increases fault tolerance. A RAID storage system appears to the operating system as a single logical hard disk device.

Response time: The average time in milliseconds it takes for one request to pass through the storage system, including any waiting time.

Throughput: The average number of read/write requests in I/Os that are passed through the storage system per second.

Utilization: The percentage of time during which the storage system is servicing requests.

Virtual machine: A virtualized x86 PC environment on which a guest operating system and associated application software can run. Multiple virtual machines can operate on the same physical machine concurrently.

Guest operating system: An operating system running on a virtual machine.

VMFS: A clustered file system that stores virtual disks and other files that are used by virtual machines.

Raw device mapping (RDM) – Raw device mapping volumes consist of a pointer in a .vmdk file and a physical raw device. The pointer in .vmdk points to the physical raw device. The .vmdk file resides on a VMFS volume, which must reside in shared storage.

Overview

In VMware environments, virtual machines (VMs) running on ESX servers have different service-level requirements. Two tools, DRS and NQM, help to meet these requirements. DRS maintains CPU and memory resources by setting reservations and shares for VMs in a *resource pool*, while NQM helps you manage storage system I/O.

Most storage systems treat each VM's I/O with the same priority. As a result, the I/O of an application might disrupt the performance of a mission-critical application.

However, NQM allows you to measure and prioritize I/O characteristics in a storage system. This functionality is implemented with three control methods; each method uses a different algorithm and approach for achieving desired service-level goals. The control methods are:

- *Cruise Control*: A specific performance target is defined for a mission-critical application. NQM prioritizes I/O so that the application's performance meets the defined target within a specified tolerance range. A target can be set for bandwidth, throughput, or response time.

- *Limit*: This method limits the performance of a LUN group to a certain level. NQM queues I/O requests for the LUN group to keep its performance under the defined limit. The limit can be set as bandwidth, throughput, or response time.

- *Fixed Queue Depth*: The array actively maintains a specified queue depth for a LUN set. This function is for expert users only, since it may prevent users from fully utilizing a storage system's resources.

NQM control methods work on a LUN-by-LUN basis, so it is important to properly provision LUNs through NQM. Also, since NQM works on a LUN-by-LUN basis, its service is not restricted to one application; NQM policies can be defined on multiple applications (LUNs) intermittently with diverse settings. For example, you can schedule a NQM policy that optimizes or limits the throughput for one virtual machine, and schedule another NQM policy that optimizes or limits bandwidth for another virtual machine.

The test described later in this paper (see "Testing NQM with DRS on VMware ESX servers") illustrates how DRS can be configured with NQM. In this example, Oracle OAST, RMAN, and ORION were run on VMs to generate stable and high-performance workloads. NQM's cruise control and limit methods were used to optimize or limit the performance of the applications.

Best practices for deploying NQM in VMware environments

Since, NQM works at the LUN level, all virtual machines residing on that LUN are affected by NQM control options. Hence, when using NQM in VMware environments, it is important to properly provision CLARiiON LUNs assigned to the ESX server and virtual machines.

EMC recommends the following best practice to apply NQM controls to an application:
1. Configure an entire LUN as an RDM or VMware file system (VMFS).
2. Assign the entire LUN to the virtual machine in which the application resides.
3. Apply NQM controls to the LUN.

If this is not feasible, ensure that I/O-intensive applications running on virtual machines do not share LUNs with *non*-I/O intensive applications running on other virtual machines. These applications *can* share the same disks in a CLARiiON RAID group; however, they must be on separate LUNs to benefit from NQM.

To provide a complete service-level guarantee for a given virtual machine running on an ESX server, it is also important to make sure that NQM's I/O parameters match VMware DRS CPU and memory parameters. For example, assume you configure a resource pool for a VM (or VMs) to have a certain number of CPU or memory reservations; define an NQM class on the LUN where the VM(s) resides; and set a NQM cruise control policy to maintain a certain I/O profile. If the CPU or memory reservations cannot be met on one ESX server, the VM(s) might migrate to a different ESX server to balance CPU and memory resources. However, the VM(s) will still maintain a given I/O profile that was configured in NQM.

Testing NQM with DRS on VMware ESX servers

In the following test, EMC investigated a use case with an Oracle database running on a virtual machine that shared a set of spindles with two other applications running on different virtual machines. Oracle Automated Stress Testing (OAST) was used as an OLTP database application to continuously make multiple-user online transactions for a set period of time. OAST was running on virtual machine 1 (VM1).

The second application was Oracle's Recovery Manager (RMAN), which backs up and recovers the Oracle database. A second database was created on virtual machine 2 (VM2), which was then backed up using RMAN. RMAN was also installed on VM2, which was hosted on a different ESX server.

The third application is ORION. This benchmark tool generates an I/O load that helps users compare performance and throughput, for different tiers and types of disk, when using an Oracle database. ORION is running on a virtual machine 3 (VM3).

Environment

This section describes the storage and server components used to test NQM with VMware ESX servers.

CLARiiON storage configuration

EMC CLARiiON CX3-20f

Processors	2
Memory size	2 GB per SP
Number of disks	30 FC 73 GB @ 15k rpm
Base software	FLARE® 26

CLARiiON RAID groups layout
2 RAID 5 (4+1)
2 RAID 1/0 (2+2)

CLARiiON LUN layout
3 metaLUNs across two RAID 5 (4 +1) group
1 metaLUN for VM boot – 200 GB
1 metaLUN for Oracle data – 200 GB
1 metaLUN for APP2 (application 2) – 100 GB

Created 2 metaLUNs across 2 RAID 1/0 (2 +2) group
1 metaLUN for Oracle Log – 100 GB
1 metaLUN for APP1 (application 1) – 100 GB

CLARiiON storage group layout
Both ESX servers were in a cluster, hence a single storage group was created with all five LUNs shared
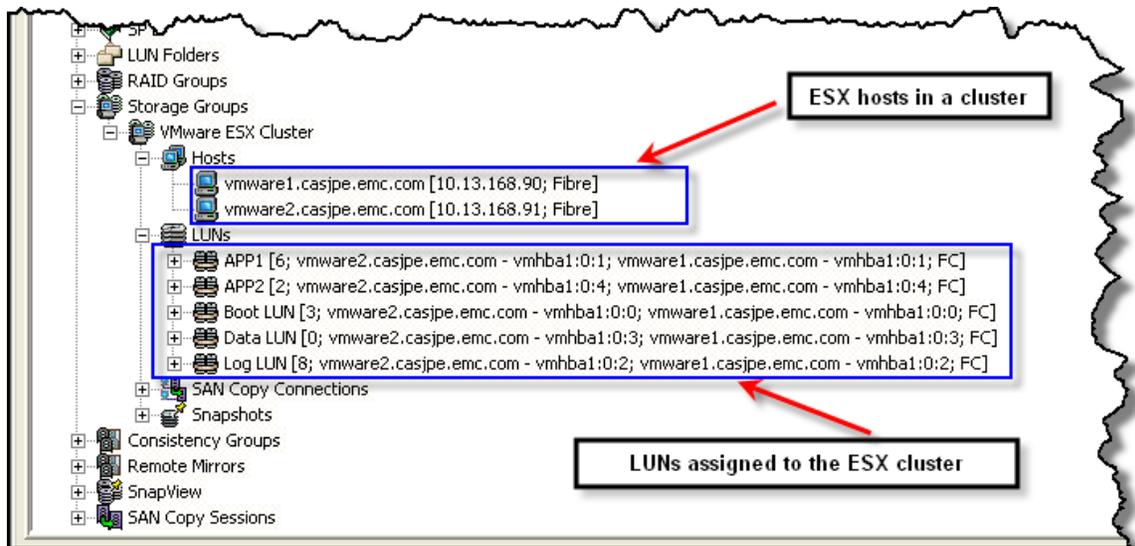across the two servers as shown in Figure 1.

Figure 1. Storage group configuration for VMware ESX servers

ESX Server configuration

2 ESX Servers	Dell 2950
ESX version	3.0.2
Memory	16 GB
Number of CPUs	8 CPU x Xeon 1.862 GHz (2 x Quad Core Xeon 1.862 GHz)
Two Fibre Channel HBAs	QLA2432

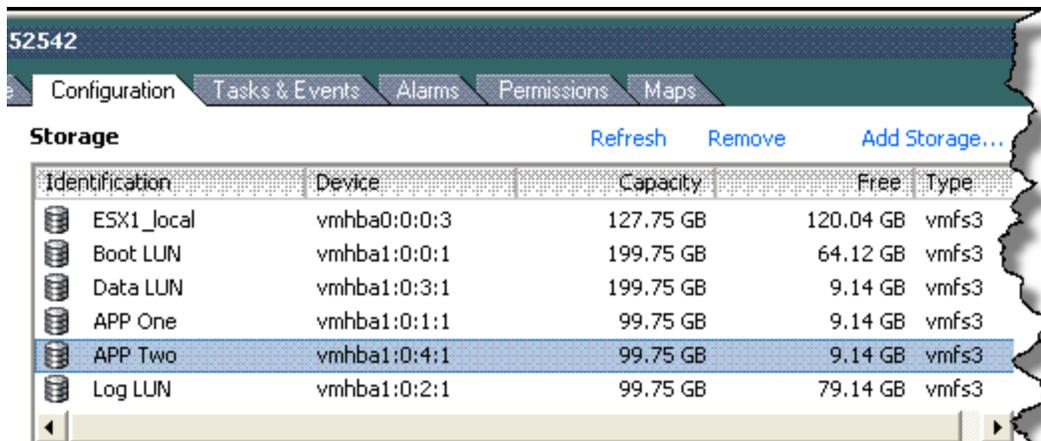LUNs assigned to ESX servers were configured as VMFS-3 volumes as shown in Figure 2.

Figure 2. LUNs configured as VMFS-3 volumes at the ESX server level

Virtual machine configuration

3 Linux virtual machines (VM1, VM2, and VM3)
VM1 on ESX Server 1
VM2 and VM3 on ESX Server 2

Table 1. Virtual machine configuration

Virtual machine	CPU\Memory config		Virtual disk layout	Application
VM1	OS Memory No. of CPUs	RHEL 5 AS 16 GB 4	20GB Boot LUN – sda 95GB Data LUN – sdb 95GB Data LUN – sdc 10GB Log LUN – sdd 10GB Log LUN – sde	**Oracle OAST**
VM2	OS Memory No. of CPUs	RHEL 5 AS 16 GB 2	20GB Boot LUN – sda 45GB APP1 LUN – sdb 45GB APP2 LUN – sdc	**RMAN**
VM3	OS Memory No. of CPUs	RHEL 5 AS 16 GB 4	20GB Boot LUN – sda 45GB APP2 LUN – sdb 45GB APP1 LUN – sdc	**ORION**

VMware Distributed Resource Scheduler (DRS) Configuration

When ESX Servers are configured in a cluster using VMware Virtual Center, DRS monitors key metrics associated with virtual machines, resource pools, and ESX servers. Virtual machines are configured for default access to certain physical resources, and in the event resources are not readily available, are dynamically and seamlessly load balanced to another node in the cluster. This process can be *Fully Automated, Partially Automated*, or *Manually Controlled*, but for the sake of these tests was set to *Fully Automated.*

To facilitate the test, a rule in the DRS configuration was created to ensure that VM1 and VM3 always ran on different ESX servers. In addition, VM1, the Oracle database server, was given a reserved minimum of 13 GB of memory out of the pool. As such, the RMAN backup and recovery test server, VM2, could benefit from dynamic load balancing, in the event it experienced contention for its assigned resources on one ESX server or the other.

Figure 3 illustrates the layout of the entire configuration with all its components.
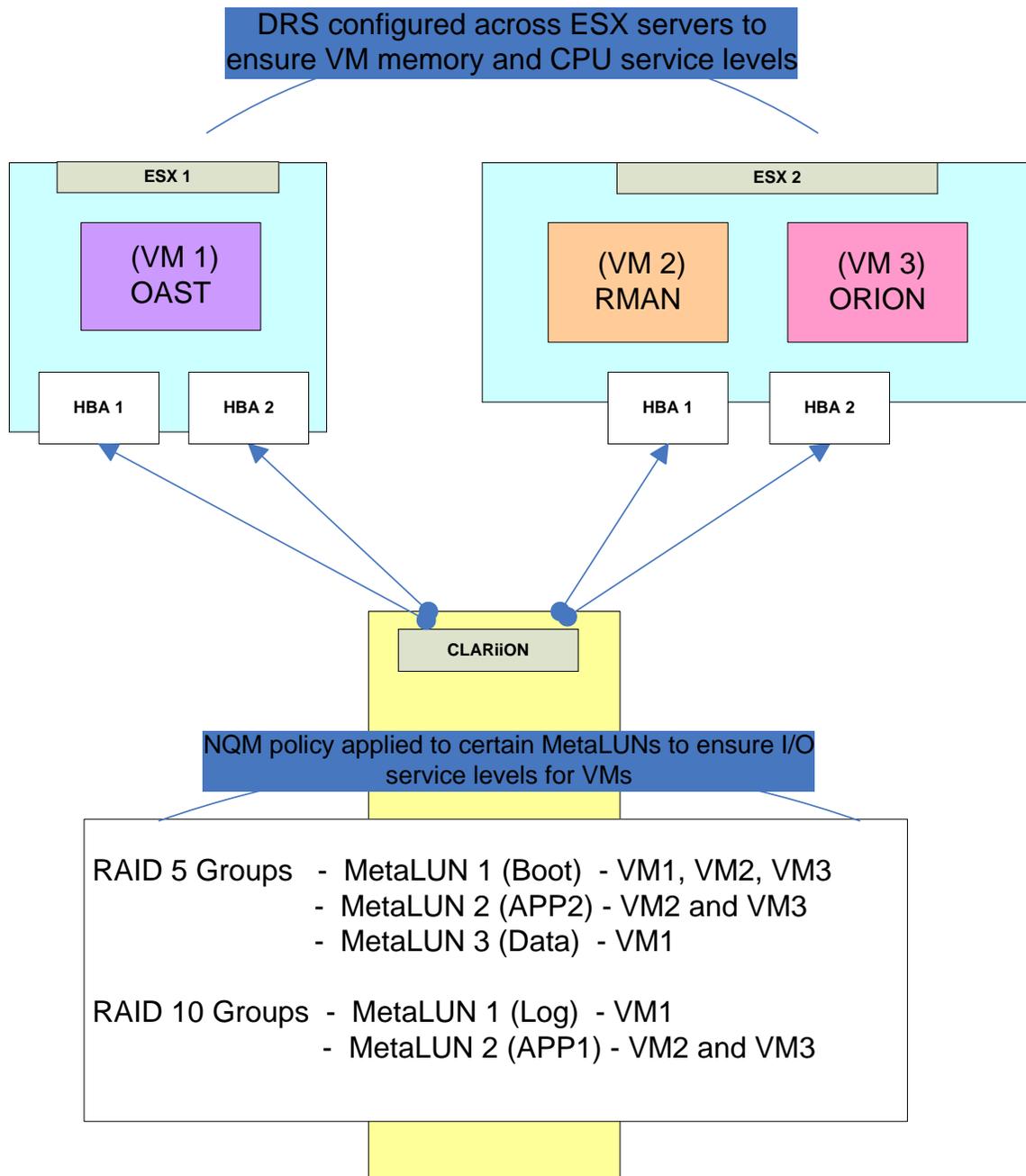
Figure 3. Test configuration

Methodology

The two main scenarios for this test were:

1. Test the cruise control option for bandwidth and throughput:

   - Run OAST on VM1.

   - Introduce contention to VM1 by running RMAN on VM2.

   - Apply NQM's cruise control option on the Data and Log LUNs for VM1.

   - Analyze test results with NQM.

2. Test the limit control option for bandwidth and throughput:

   - Run OAST on VM1, RMAN on VM2, and ORION on VM3.

   - Apply NQM's limit control option on Data and Log LUNs to provide a certain service level for RMAN backup.

   - Analyze test results with NQM.

   - Remove the limit control option on VM1 after RMAN backup completes.

   - Analyze test results after NQM controls have been removed.

This tested NQM's cruise control and limit functions. Tests were repeated three or four times (with the same settings) to ensure consistency. For each test, data was collected for all LUNs: bandwidth (MB/s), throughput (IO/s), response time (ms), utilization (%), and queue length.

Test 1: Using NQM's cruise control function

This test involved applying NQM's cruise control function to LUNs presented to VM1 (running OAST) in order to stabilize bandwidth and throughput of VM1. After a certain time interval, contention was introduced in the form of the RMAN application running on VM2. The contention was caused because the LUNs presented to both VM1 and VM2 were using the same set of disks on the CLARiiON storage system.

VMware DRS was configured to use the "fully automated" policy for the ESX server cluster in order to load balance CPU and memory resources across VM1 and VM2. The goal of this use case was to show how DRS along with NQM provide the needed performance for VM1, even when a backup (VM2) application is introduced that uses the same set of disks.

Setting NQM's cruise control option to stabilize bandwidth

As shown in Figure 4, two NQM I/O classes were created within Navisphere. The "DATABASE" class indicates the bandwidth of the OAST on VM1 while the "Background Class" indicates the bandwidth of RMAN on VM2. OAST on VM1 ran first, and then RMAN was introduced on VM2. As shown in the graph, the database performance decreased when RMAN and database LUNs were contending for resources. The bandwidth of the database decreased from about 14 MB/s to 2 MB/s after RMAN was introduced.

At 10:50, the NQM cruise control option was applied to stabilize the bandwidth of the DATABASE on VM1 to about 15 MB/s. After about 10 minutes[1] (at approximately 11:01), we can see the result of the cruise control option. The bandwidth of the Background Class went down, and the database recovered its bandwidth.
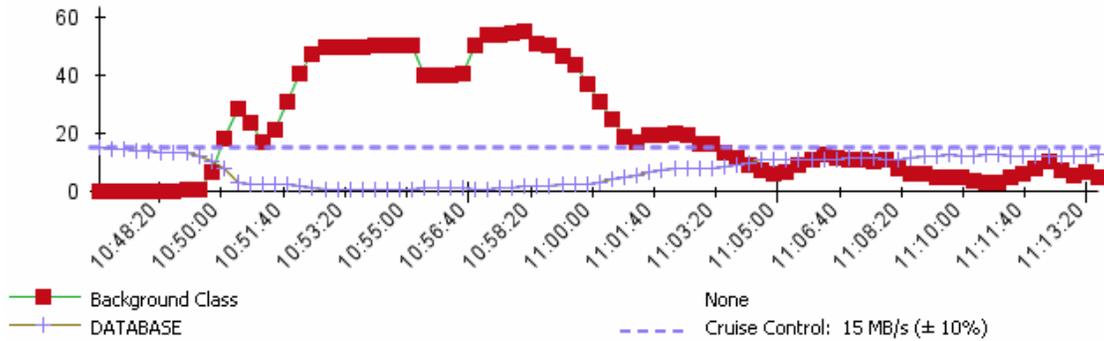

Figure 4. Data from the CLARiiON storage system depicted the effect of the NQM cruise control function on bandwidth

Figure 5 and Figure 6 show a decrease in transactions per minute (TPM), as well as a decrease in response time, with the introduction of RMAN, during the 3-5 minute time intervals. After the NQM cruise control bandwidth option was applied to the LUN given to VM1, the TPM numbers and the response time stabilized.
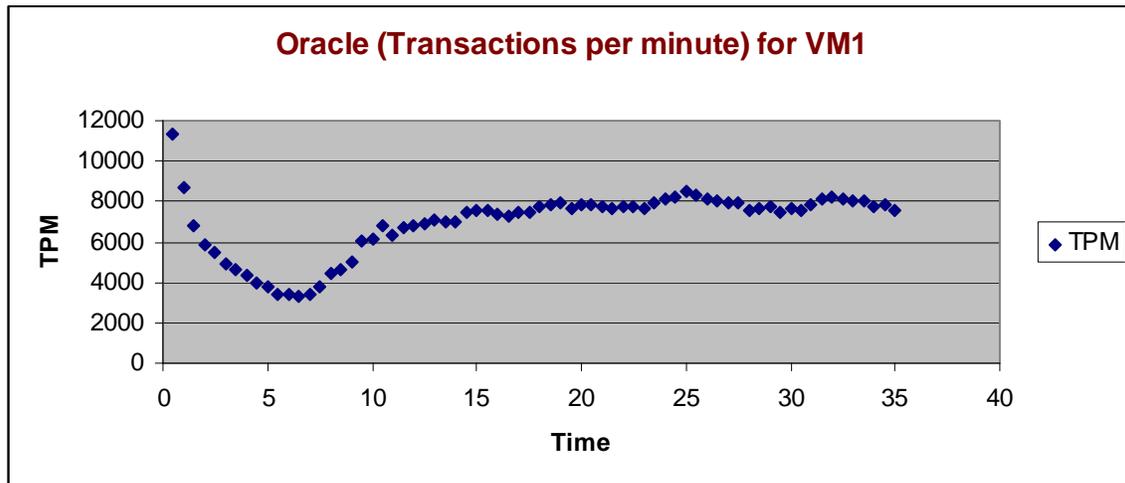
Figure 5. Transactions per minute chart for Oracle OAST on VM1

---

[1] It can take the cruise control function longer than the limit function to achieve goals.
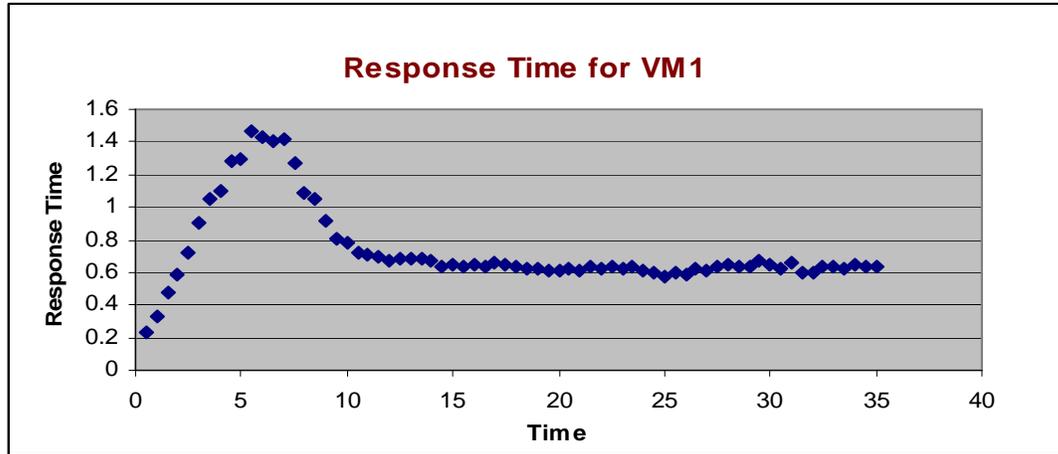
Figure 6. Response Time chart for VM1 (running Oracle OAST)

Setting NQM's cruise control option to stabilize throughput

Along the same lines, a test was conducted in which the cruise control option was applied to stabilize the throughput of VM1 after contention was introduced by running RMAN on VM2. The DATABASE option below indicates the throughput of the OAST on VM1, and the Background Class indicates the throughput of RMAN on VM2.

The contention caused the throughput of VM1 to decrease from about 2000 I/Os per second to about 300 I/Os per second, as shown in Figure 7. The cruise control option of 1000 I/Os per second was applied to the LUN provisioned to VM1 at 13:50, which caused the throughput of VM1 to stabilize after a period of time, in spite of the contention that was introduced.
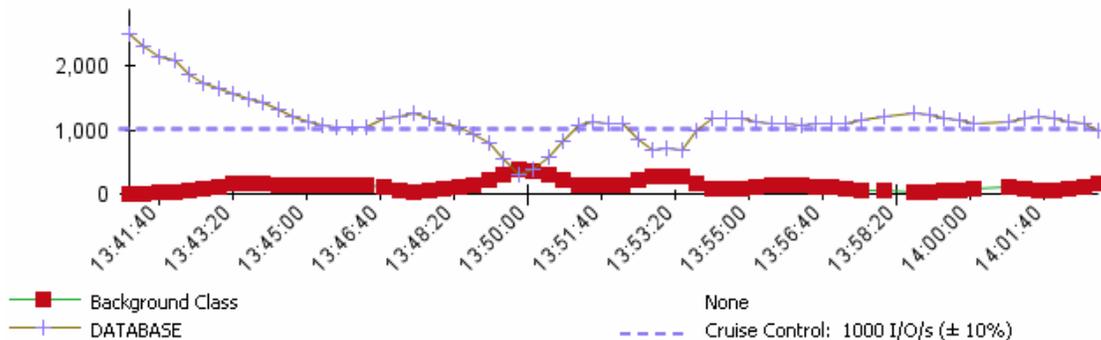

Figure 7. The effect of NQM cruise control on the throughput of the CLARiiON storage system

Figure 8 shows an unstable decrease in TPM, during 8-10 minute time intervals, with the introduction of the RMAN. A few minutes after the NQM cruise control throughput option was applied to the LUN given to VM1 (close to time 10), the TPM numbers gradually increased.
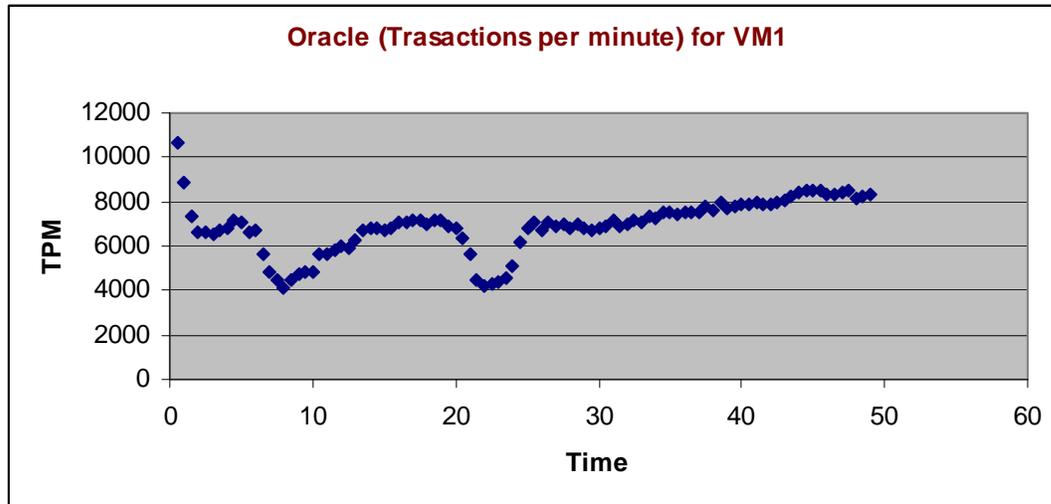
Figure 8. Transactions per minute chart for Oracle OAST on VM1

Cruise control test results on vm1

As shown in both tests, the cruise control option, once triggered, guaranteed a specified service level for VM1. Depending on the application requirements, you can tune either bandwidth or throughput with NQM. This can be generalized to any application that is contending for resources with other applications; thus the cruise control option could be applied to the RMAN application running on VM2.

Test 2: Using NQM's limit function

During this test, NQM's limit function was applied to the LUNs on which the OAST application was running (in VM1) to provide sufficient bandwidth and throughout for the RMAN backup operation running in VM2. A boost in resource requirement for the ORION application running on VM3 was also noted during this test. For this use case, NQM was used to increase the bandwidth and throughput of the backup application in cases where backups must be completed within a certain time window.

Setting NQM's limit function on bandwidth

As shown in Figure 9, two NQM I/O classes were created within Navisphere. The DATABASE class was the database running on VM1. The Background Class was the RMAN backup and ORION benchmark application running on VM2 and VM3, respectively, which were configured on the same set of LUNs. After seeing a performance drop in the Background Class, a bandwidth limit of 2 MB/s was applied at 15:53 on the DATABASE I/O class on VM1. The Background Class (consisting of RMAN on VM2 and ORION on VM3) operations gained bandwidth, along with throughput. The limit goal was reached within 2 minutes. As seen in the graph, the bandwidth for RMAN and ORION increased after NQM controls were applied.
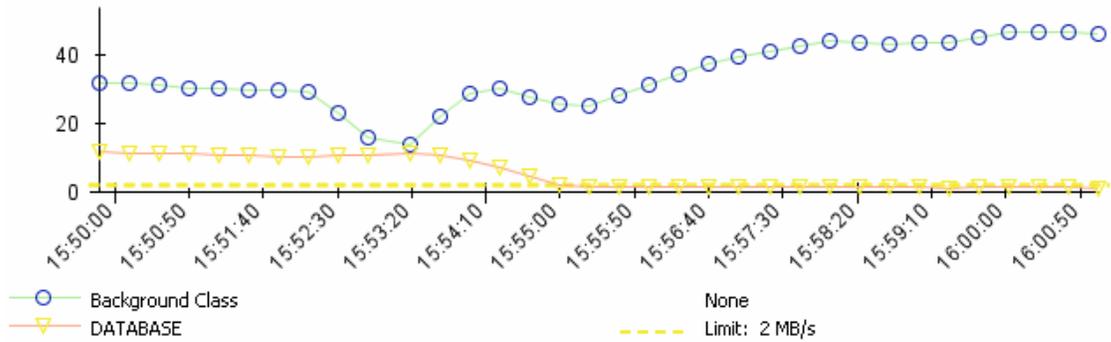
Figure 9. The effect of the NQM limit function on bandwidth

Setting NQM's limit function on throughput

Another test was conducted where the limit function was applied on the throughput to the LUNs presented to VM1. A limit of 100 I/Os per second was applied at 16:11 on the DATABASE I/O class (VM1) as shown in Figure 10. In a few minutes, the throughput of the Background Class, which was composed of a RMAN workload (VM2) and ORION workload (VM3), increased.
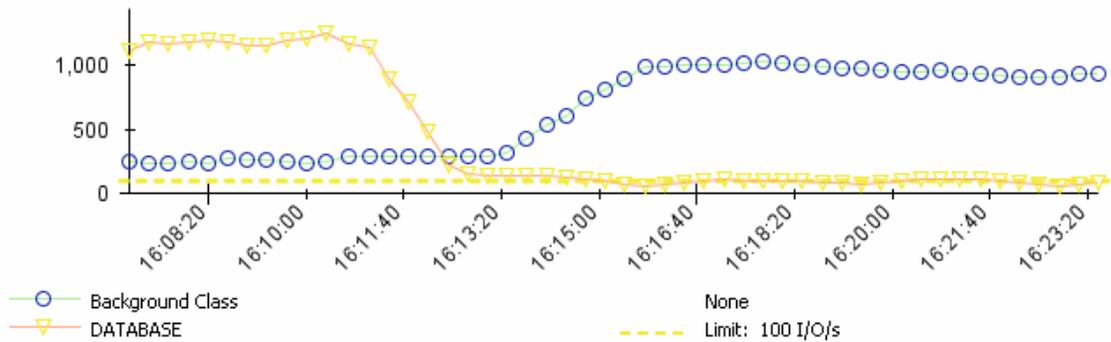

Figure 10. Data from the CLARiiON storage system depicted the effect of the NQM limit on throughput

The response time of VM1 increased with the application of the limit function on the LUNs given to VM1, while the response time for RMAN (VM2) and ORION (VM3) applications decreased, as shown in Figure 11.
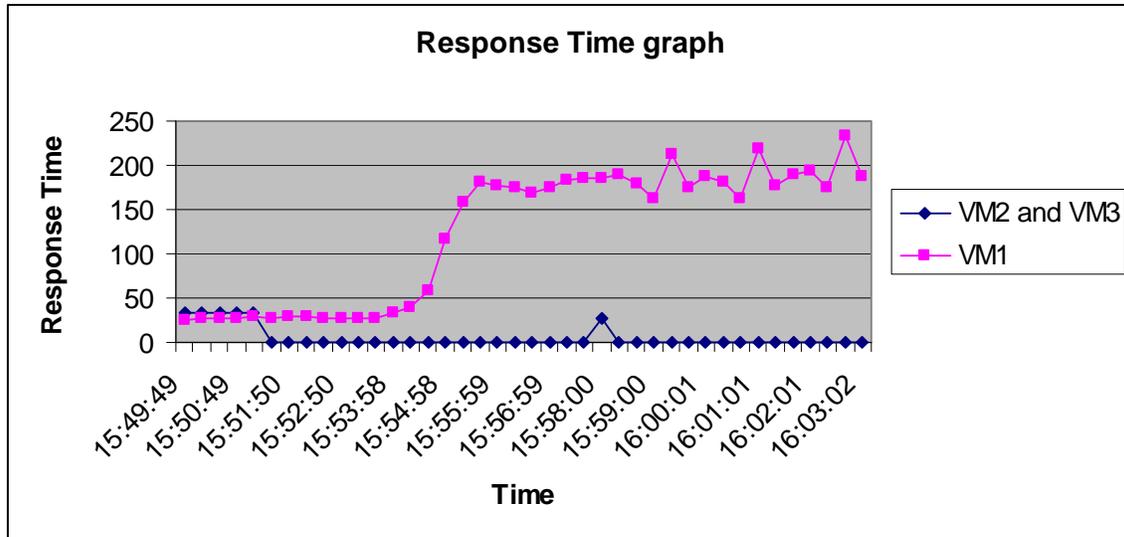
Figure 11. Response time graph comparing VM1, VM2, and VM3

Removing NQM's limit control

After the RMAN backup operation finished in the Background Class, the limit control option was disabled on DATABASE at 16:25, and the DATABASE operation returned to its normal performance level, as shown in Figure 12.
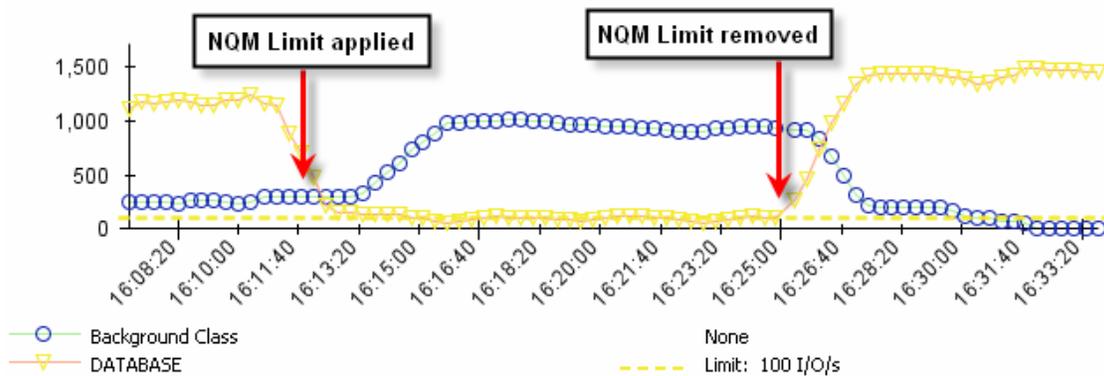

Figure 12. Graph comparing throughput of VM1, VM2, and VM3 when NQM limit is applied and removed.

Limit test results on VM1

As shown in the above test results, the limit control option can limit the bandwidth, throughput, and response time for a given application, and can be removed once the operation completes. This can be generalized to any application that is contending for resources with other applications, thus the limit function could have been applied to the RMAN (VM2) and ORION (VM3) applications that reside on the same set of LUNs, to increase the performance of VM1. The limit goals can be reached much faster than the cruise control goals.

Conclusion

NQM can be a valuable application for enhancing storage utilization and improving user experience in VMware ESX deployments. By using NQM in conjunction with DRS, storage and server resources can maintain end-to-end I/O service levels for critical applications, such as Oracle databases. As a result, performance of critical applications is not compromised by least-critical applications that are sharing the same server or storage resources. The overall server and storage system utilization is improved while reducing the complexity of data layout and its planning.

References

More information can be found at http://www.EMC.com/products/systems/clariion.jsp.

Related documents

- *Navisphere Quality of Service Manager (NQM) – Applied Technology* white paper, available at EMC.com and the EMC Powerlink website
- *CLARiiON Integration with VMware ESX Server – Applied Technology* white paper, available at EMC.com and the EMC Powerlink® website
- *SAN Configuration Guide*, available at VMware.com
- *Resource Management Guide*, available at VMware.com
- *SAN Design and Deployment Guide,* available at VMware.com