Technical Report

Storage Performance Primer

Clustered Data ONTAP 8.3

Bob Allegretti, NetApp Technical Marketing Performance Management
August 2015 | TR-4211

Abstract

This paper describes the basic performance concepts as they relate to NetApp® storage systems and the clustered Data ONTAP® operating system. It also describes how operations are processed by the system, how different features in clustered Data ONTAP can affect performance, and how to observe the performance of a cluster.

Data Classification

Version History

Version	Date	Document Version History
Version 1.2	July 2015	Clustered Data ONTAP 8.3.1 revisions Bob Allegretti. Added Performance Manager 2.0. Minor revisions throughout.
Version 1.1	April 2015	Clustered Data ONTAP 8.3 revisions. Authors: Bob Allegretti and Roy Scaife.
Version 1.0	July 2013	Initial version for clustered Data ONTAP 8.2. Authors: Paul Updike, Roy Scaife, and Chris Wilson.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Overview

As demand for storage continues to increase and budgets decrease, IT departments need to get more out of their storage infrastructures in both capacity and performance. This document covers the performance principles and architecture of clustered Data ONTAP® operating system and how it efficiently provides data storage.

Performance is notoriously known for its inherent complexity. NetApp provides simple and capable tools for performance management. This document also introduces an off-box performance management tool specifically designed for clustered Data ONTAP systems. OnCommand® Performance Manager is a part of the OnCommand product portfolio and is provided at no additional cost.

2 Introduction to Clustered Data ONTAP Performance

This document contains a general overview of system architecture, the basic principles of operation, and an introduction to performance management. Do not expect a tuning or a deep troubleshooting guide. Before reading this guide, a good understanding of NetApp clustered Data ONTAP concepts is required. For an introduction to clustered Data ONTAP, see [TR-3982: NetApp Clustered Data ONTAP 8.3 and 8.2.x: An Introduction](#).

2.1 Performance Fundamentals

The fundamental unit of work performed by storage systems is a data operation (typically shortened to simply "op") that either reads or writes data to or from storage systems. The complexities surrounding performance come from the many variables that affect performance. In addition, there are many different types of derived measurements describing performance called metrics. Among those metrics, two are considered most significant and believed to accurately characterize performance at its highest level: throughput and latency. The first, throughput, describes the amount of work the system is doing by expressing units of work over time: for example, megabytes per second (MBps) or input/output operations per second (IOPS). The second, latency, describes the time it takes to complete a unit of work: for example, a user read or write operation expressed in milliseconds per operation (ms/op) units. Note that the term *latency* in the context of this document is functionally equivalent to round trip response time. This terminology, though technically questionable, is a longstanding tradition in the storage industry and would be prohibitive to change. Tens of thousands or hundreds of thousands of operations take place every second so throughput and latency are typically expressed as averages normalized over a given time range (for example, per second) and a unit of work (for example, per operation).

The Data ONTAP operating system and the underlying cluster hardware work efficiently to make sure data is secure, reliable, and always available. The operations a storage system performs are a direct function of the client operations requested by applications supporting an enterprise. Collectively, the operations mix generated by applications is uniquely referred to as an application set workload, often shortened to simply "workload." Workload characteristics that can affect and be used to describe performance include:

- **Throughput.** The number of operations or amount of data payload over a given period of time.
- **Concurrency.** The number of operations in flight (or resident) at a given point in time.
- **Operation size.** The size of the operations requested. The data portion of the operation often referred to as block size or payload.
- **Operation type.** The type of operation requested of the storage system (for example, read, write).
- **Randomness.** The distribution of data access across a dataset in an unpredictable pattern.
- **Sequentiality.** The distribution of data access across a dataset in a repeatable pattern. Many patterns can be detected: forward, backward, skip counts, and others.

- **Working set size.** The amount of data considered to be active and frequently used to complete work.
- **Dataset size.** The amount of data that exists in a system that is both active and at rest.

Varying any of these workload characteristics ultimately ends up affecting the performance of the system and can be observed through measured changes in either latency or throughput. In many production environments application workload almost always increases over time often without warning. Therefore, the performance of the storage system must be known. Armed with this knowledge, plans to allocate more resources or rebalance workloads can be made to meet the demands placed upon the system.

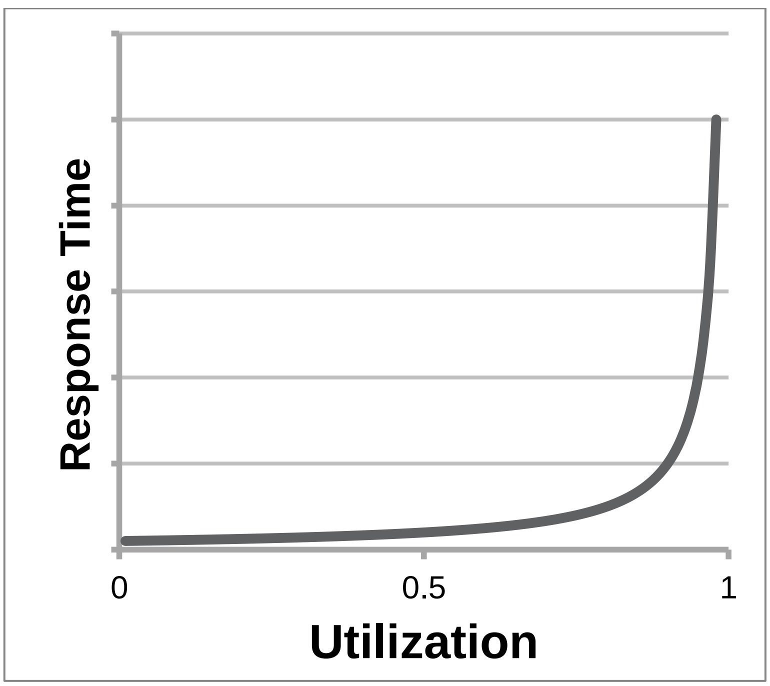
2.2 Normal Performance Relationships

There are some guiding principles behind performance useful in day-to-day operations. These can be stated as relationships between the fundamental characteristics of a workload and their impact on performance:

- Throughput is a function of latency.
- Latency is a function of throughput.
- Throughput is a function of concurrency.
- Throughput is a function of operation size.
- Throughput is a function of randomness of operations.
- Host applications control the operation mix, operation size, randomness, concurrency.

These relationships can be summarized by an exponential growth curve as depicted in Figure 1, where response time (or latency) increases nonlinearly as utilization (or throughput) increases.

Figure 1) Response time exponential growth curve as utilization is saturated.



Throughput and Latency

Workloads can be defined as either closed-loop or open-loop systems. In closed-loop systems, a feedback loop exists. Subsequent operation requests from applications are dependent upon the completion of previous operations and, when bounded by the number of concurrent operation requests, limit the offered load. In this scenario the number of concurrent requests is fixed, and the rate that operations that can be completed depends on how long it takes (latency) for previous operations to be completed. Simply put, in closed-loop systems, throughput is a function of latency; if latency increases, throughput decreases.

In open-loop systems, operations are performed without relying on feedback from previous operations. This configuration can be a single enterprise-class application generating multiple asynchronous requests or hundreds of independently running servers issuing a single threaded request. This fact means that the response time from those operations doesn't affect when other operations are requested. The requests will occur when necessary from the application. As offered load to the system increases, the utilization of the resources increases. As the resource utilization increases, so does operation latency. Because of this utilization increase, we can say that latency is a function of throughput in open-looped systems although indirectly.

Concurrency

Storage systems are designed to handle many operations at the same time. In fact, peak efficiency of the system can never be reached until it is processing a large enough number of operations such that there is always one waiting to be processed behind another process. Concurrency, the number of outstanding operations in flight at the same time, allows the storage system to handle the workload in the most efficient manner. The effect can be dramatic in terms of throughput results.

Concurrency is often a difficult concept to grasp because it is very abstract. One way to picture it is to imagine a single application sending a thousand asynchronous requests in one second (open-loop) and to also imagine a thousand applications sending one synchronous request (closed-loop) in one second. The concurrency experienced on the storage system is identical in either case.

Little's Law: A Relationship of Throughput, Latency, and Concurrency

Little's Law describes the observed relationship between throughput (arrival rate), latency (residence time), and concurrency (residents):

$L = A \times W$

This equation says that the concurrency of the system (L) is equal to the throughput (A) multiplied by latency (W). This implies that for higher throughput, either concurrency would have to increase and/or latency to decrease. This explains why low-concurrency workloads (single-threaded workloads), even with low latencies, can have lower than expected throughput. Thus, to increase throughput with low latency require more workloads to be added to the environment or more concurrency added to the workload.

Operation Size

A similar effect on concurrency is observed with the size of operations on a system. More work, when measured in megabytes per second (MBps), can be done with larger operations than can be done with smaller operations. Each operation has fixed overhead associated with it. By increasing the operation size (or data payload), the ratio of overhead to data is decreased, which allows more throughput over the same time period. Similarly, when work depends on latency in low-concurrency workloads, a larger operation size increases the data throughput efficiency of each individual operation.

Small operations might have a slightly better latency than large operations, so the operations per second could be potentially higher, but the measured data throughput suffers with smaller operations.

Data Access (Random or Sequential)

Data operations sent to a storage system access a logical location within a data file or LUN. This logical location is ultimately translated into an actual physical location on the permanent storage media. The order of operations and the access pattern of the data over time determine the randomness of a workload. When the logical addresses are ordered (next to one another), access patterns are considered sequential.

Sequentially read data exhibits better performance characteristics because fewer drive seeks and operations are required from permanent storage media.

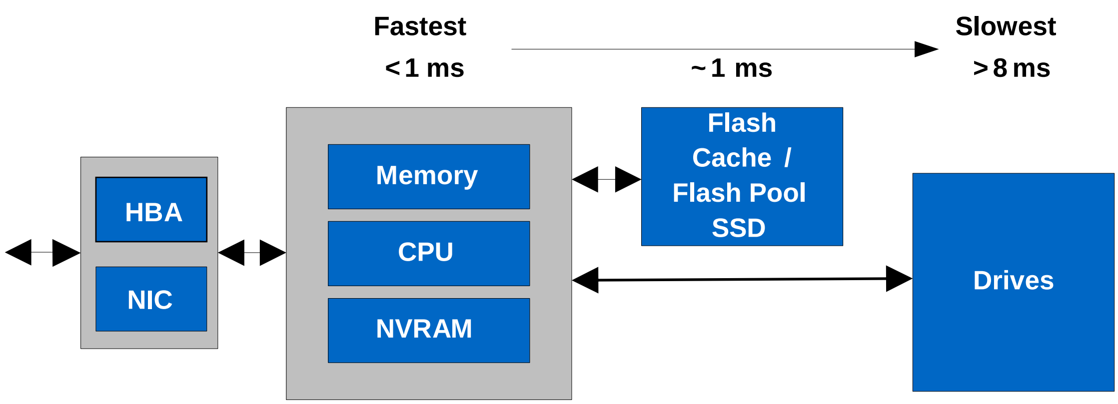
Data ONTAP is highly write-optimized. Due to the way writes are written to storage, almost all writes behave as if they are sequential writes. Thus, we see less improvement in random versus sequential writes. For more information see section 2.6.1 Basic Workload Characterization.

2.3 Cluster Node System Architecture Overview

Storage systems are designed to store and retrieve large amounts of data reliably, inexpensively, and quickly. Unfortunately, the costs associated with storing large amounts of data lead to slow storage media: the mechanical (spinning) disk drive. The fastest storage media such as silicon-based random access memory (RAM) are not persistent and are expensive. It is important to recognize that every workload interacts with the system differently and there are many different workloads. This fact creates a technical challenge around providing the best performance for workload conditions that are largely unknown. NetApp meets this challenge through innovative technologies combining the use of spinning disk, flash, and RAM.

A NetApp storage system may be logically divided into three main areas when discussing performance. Those are connectivity, the system itself, and the storage subsystem. Connectivity refers to the network interface card (NIC) and host bus adapter (HBA) that attach the storage system to the clients and hosts. The system itself is the combination of CPU, memory, and NVRAM. Finally, the storage subsystem consists of the disks, and also Flash Cache™ and Flash Pool™ intelligent caching. Figure 2 logically represents a NetApp system.

Figure 2) High-level cluster node system architecture.



A system running clustered Data ONTAP consists of individual nodes joined together by the cluster interconnect. Every node in the cluster is capable of storing data on disks attached to it, essentially adding "copies" of the preceding architecture to the overall cluster. Clustered Data ONTAP has the capability to nondisruptively add additional nodes to the system to scale both system performance and capacity.

2.3.1 Connectivity: NICs and HBAs

NICs and HBAs provide the connectivity to client, management, and cluster interconnect networks. Adding more or increasing the speed of NICs or HBAs can scale client network bandwidth.

2.3.2 Controller Subsystem: Memory, CPU, and NVRAM

The controller subsystem contains CPU and some amount of memory, depending on the controller model. As with any computer, the CPU provides the processing power to complete operations for the system. In addition to holding the Data ONTAP operating system, the memory in a NetApp controller also acts as a cache. Incoming writes are staged in main memory prior to being written to disk. Memory is also used as a read cache to provide extremely fast access time to recently read data.

NetApp systems also contain NVRAM. NVRAM is battery-backed memory used to protect in-bound writes as they arrive. This fact allows write operations to be safely committed without having to wait for a disk operation to complete greatly reducing write latency. High-availability (HA) pairs are formed by mirroring NVRAM across two controllers.

Increasing the capacity and performance of these components requires upgrading to a higher controller model. Clustered Data ONTAP allows nodes to be evacuated and upgraded nondisruptively to clients.

2.3.3 Storage Subsystem: Disks, Flash Cache, and Flash Pool

Spinning disk drives are the slowest persistent storage media available. The typical response times for spinning disks are on the order of several milliseconds. Disk drive performance varies depending on the disk type and rotation speed: 7.2K RPM SATA disks have higher latency than 10K RPM SAS disks. Solid-state disks significantly reduce the latency at the storage subsystem. Ultimately, the type of disk needed for a specific application depends on capacity, performance requirements, and workload characteristics. For more information about disk drives, see TR-3838: Storage Subsystem Configuration Guide.

With the introduction of NetApp Flash Cache™ and Flash Pool™ technology, it is possible to combine the performance of solid-state flash technology with the capacity of spinning media. Flash Cache typically operates as an additional layer of read cache for the entire system. It caches recently read, or "hot," data for future reads. Flash Pool serves as a read cache in a similar fashion to Flash Cache at the aggregate level as opposed to the system level. This fact allows improved cache provisioning for specific workloads. Flash Pool also caches random overwrites improving write latency as well.

For more information about Flash Cache, see TR-3832: Flash Cache Best Practices Guide. For more information about Flash Pool, see TR-4070: Flash Pool Design and Implementation Guide.

2.4 Data Storage and Retrieval

The fundamental purpose of a storage system is to provide services to access data reliably (without error), persistently (always available), securely, and fast. A NetApp storage system does this through presenting storage abstractions, such as volumes, LUNs, and file systems, that are physically hosted on a pool of resources referred to as a cluster. Clusters are composed of individual nodes connected through an internal cluster interconnect. Every node is an autonomous system managing it's dedicated resources running technologically advanced software that flawlessly orchestrates these services called clustered Data ONTAP.

2.4.1 Cluster Operations

In clustered Data ONTAP operating system, data does not need to reside on the node connected to the client and can reside anywhere within the cluster. This fact gives great flexibility in adding resources and rebalancing load as business demands change requiring no modifications to the application layer configuration. Thus, data can be accessed directly when residing on local nodes or indirectly across the cluster through application requests, generally referred to as operations, often shortened to simply "ops." Operations can take on many forms, such as the commonly known read and write operation, and lesser known types often categorized as "metadata operations" or "other ops."

Generally speaking, clustered Data ONTAP is composed of four major architectural components:

- **Network.** Transports the operation request and response to and from the application.

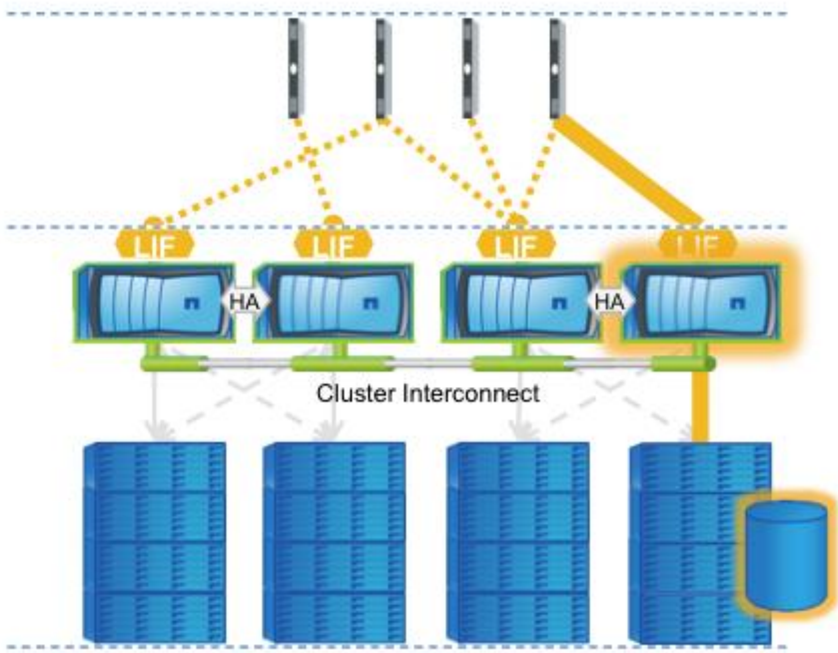
- **Cluster interconnect (indirect access only).** Transports the operation to the node that has responsibility to execute the operation.
- **Data access and layout.** Optimizes the execution of the operation requested in context with all other operations taking place (otherwise known as the WAFL® [Write Anywhere File Layout] system).
- **Disk.** Stores data to permanent media.

Operations can traverse each of these components across a cluster. The average amount of time an operation takes to traverse these components is the latency or response time metric.

Direct Data Access

Direct data access occurs when a client connected to a logical interface (LIF) assigned to a node accesses data stored on disks directly connected to that node. When accessing data in this fashion there is no traversal of the cluster interconnect. Note in Figure 3 how data flows (solid yellow line) directly to the disk connected to local node bypassing the cluster interconnect (floating above).

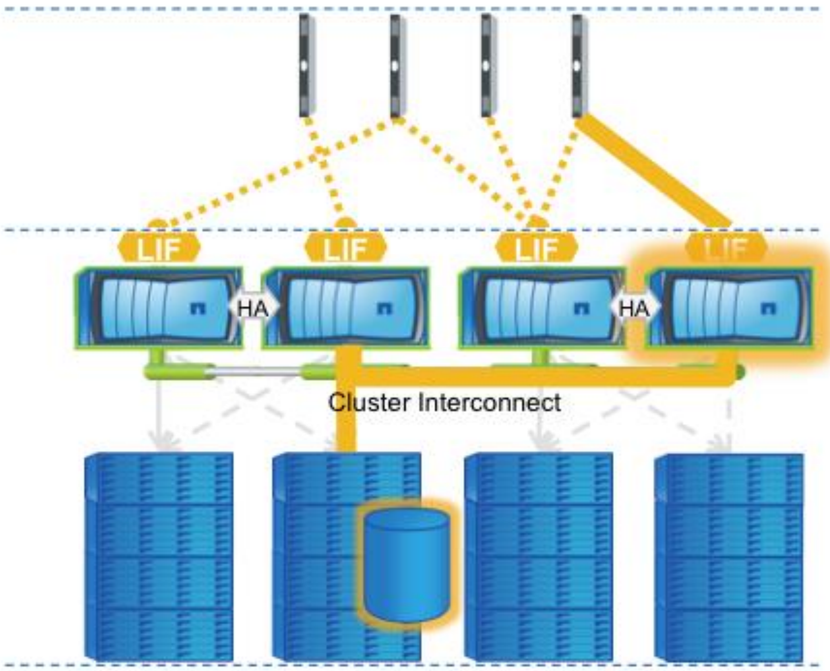
Figure 3) Direct data access on local node.



Indirect Data Access

Indirect data access occurs when a client connected to a LIF assigned to a node accesses the data stored physically on another node using the cluster interconnect (see Figure 4). Indirect data access allows data to live physically on any node without the need to force clients to mount more than a single location to access the data.

Figure 4) Indirect data access to remote node.



Protocol Considerations

Accessing data directly on the node where it is stored is ultimately considered the "shortest path" to the data. Some of the protocols supported by clustered Data ONTAP operating system have the ability to automatically discover the optimal path to the data providing direct data access. Independent of protocols, the management features of clustered Data ONTAP can always be used to override the data access path.

Certain protocols have the capability to automatically direct traffic to the node with direct data access. In the case of NAS protocols, NFS version 4 (NFSv4) can direct clients to local nodes through a variety of capabilities. NFSv4 referrals point the client to the directly attached node during mount. Another capability with NFSv4.1 is parallel NFS (pNFS). pNFS enables clients to connect to any node in the cluster for metadata work while performing direct data operations. To learn more about NFS capabilities in clustered Data ONTAP, read TR-4067: Clustered Data ONTAP NFS Best Practices and Implementation Guide and TR-4063: Parallel Network File System Configuration and Best Practices for Clustered Data ONTAP 8.2 and Later. Similarly, the SMB 2.0 and 3.0 protocols support a feature called autolocation. This capability automatically directs a client to the direct node when mounting a share. More information is available in the Data ONTAP documentation.

In SAN environments, the ALUA protocol enables optimal pathing to a LUN. Even if volumes are moved around in the cluster, the host always accesses the LUN through the optimal path. To learn more about using SAN with clustered Data ONTAP, read TR-4080: Best Practices for Scalable SAN in Clustered Data ONTAP.

2.4.2 Node Operations

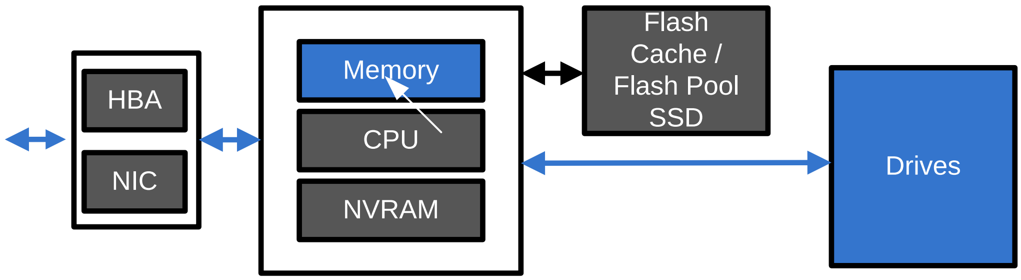
After an operation has been directed to the proper node, that node becomes responsible for completing the read or write operation. The read and write paths in clustered Data ONTAP operating system are very different. In this section, we examine how reads and writes are completed on a node and how the components within the storage system are used.

Reads

Recall the storage system architecture presented in section 2.3 “Cluster Node System Architecture Overview”; read operations can be serviced from memory, flash-based cache, or spinning disk drives. The workload characteristics and capabilities of the system determine where reads are serviced and how quickly. Knowing where reads are serviced can help set expectations as to the overall performance of the system. In the following diagrams, components and links in blue highlight the activity described.

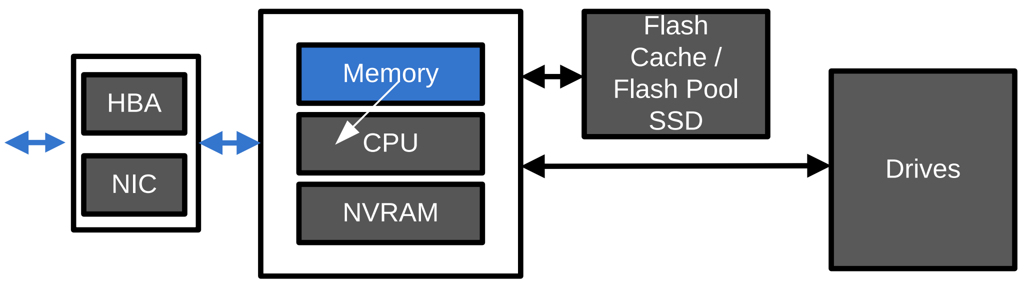
In the simple yet slowest case (Figure 5), read requests that are not cached anywhere must come from disk. After being read from disk, the data is kept in main memory.

Figure 5) Read from disk.



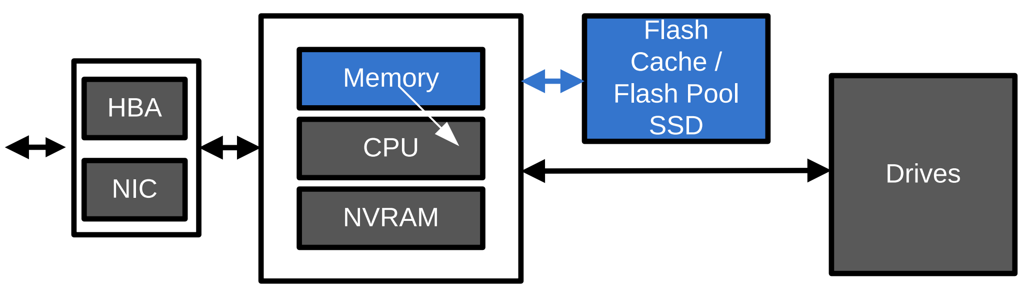
If this data is read again soon, it is possible for the data to be cached in main memory, making subsequent access extremely fast because no disk access would be required (Figure 6).

Figure 6) Read from memory.



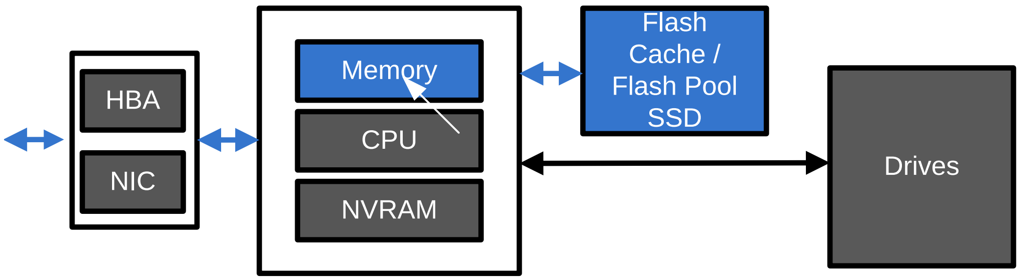
When more room is needed in main memory cache, as is common with working sets larger than the memory cache, data is evicted. If Flash Cache or Flash Pool is in the system, that block could be inserted into the flash-based cache. In general, only randomly read data and metadata are inserted into flash-based caches (Figure 7).

Figure 7) Write (insert) to flash.



After data is inserted into Flash Cache, subsequent reads of this block unable to be serviced from the buffer cache would be served from the flash-based cache (Figure 8) until they are evicted from the flash-based cache. Flash access times are significantly faster than those of disk, and adding cache in random read–intensive workloads can reduce read latency dramatically.

Figure 8) Read from flash.

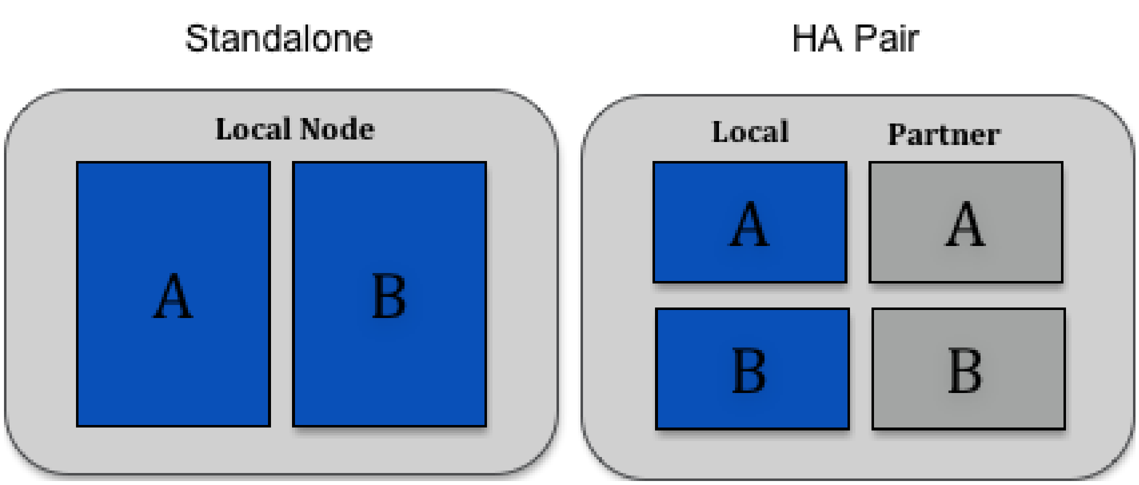


Incoming reads are continually being checked for access patterns. For some data access patterns, such as sequential access, Data ONTAP predicts which blocks a client may want to access prior to the client ever requesting. This “read-ahead” mechanism preemptively reads blocks off disk and caches them in main memory. These read operations are serviced at faster RAM speeds instead of waiting for disk when the read request is received.

Writes

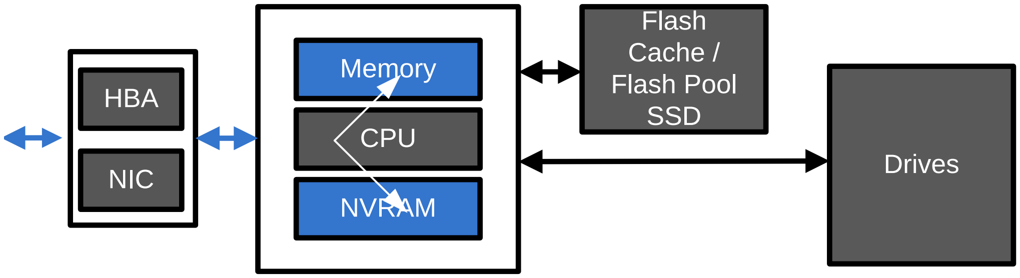
Next, consider how data is written on the storage system. For most storage systems, writes must be placed into a persistent and stable location prior to acknowledging to the client or host that the write was successful. Waiting for the storage system to write an operation to disk for every write could introduce significant latency. To solve this problem, NetApp storage systems use battery-backed RAM to create nonvolatile RAM (NVRAM) to log incoming writes. NVRAM is divided in half, and only one half is used at a time to log incoming writes. When controllers are in highly available pairs, half of the NVRAM is used to mirror the remote partner node’s log, while the other half is used for logging local writes. The part that is used for logging locally is still split in half, just like a single node (Figure 9).

Figure 9) NVRAM segmenting: standalone and HA pair



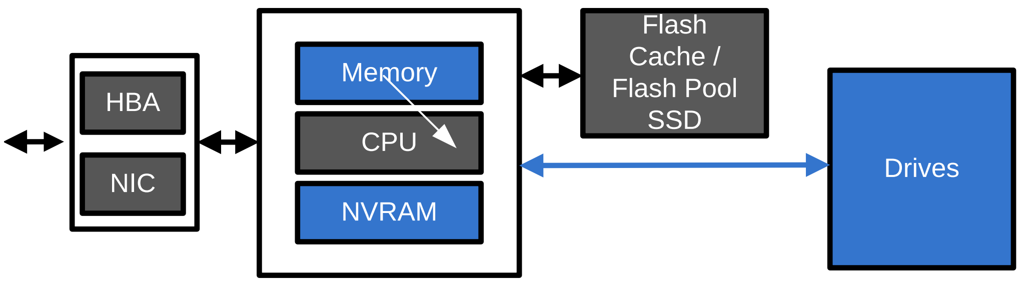
When a write enters a NetApp system, the write is logged into NVRAM and is buffered in main memory. After the data is logged in persistent NVRAM, the client is acknowledged (Figure 10). NVRAM is accessed only in the event of a failure.

Figure 10) Accepting a write



At a later point in time, called a consistency point, the data buffered in main memory is efficiently written to disk (Figure 11). Consistency points (CP) can be triggered for a number of reasons, including time passage, NVRAM utilization, or system-triggered events such as a Snapshot® copy.

Figure 11) Consistency point.



In general, writes take a minimal amount of time, on the order of low milliseconds to sub-milliseconds. If the disk subsystem is unable to keep up with the client workload and becomes too busy, write latency can begin to increase. When writes arrive too quickly for the provisioned back-end storage, both sides of the NVRAM can fill up and lead to a scenario called a back-to-back CP. This fact means that both segments of NVRAM log are full, a CP is currently occurring, and another CP will immediately follows the current CP's completion. This scenario affects performance because the system can't immediately acknowledge the write because NVRAM is full, and the client must wait until the operation can be logged. Improving the storage subsystem often alleviates the back-to-back CP scenario. Increasing the number of disks, moving some of the workload to other nodes, and considering flash-based caching can help solve write performance issues. Remember, only randomly overwritten data is written to the SSD portion of a Flash Pool aggregate and that Flash Cache is only a read cache but that offloading any type of operation can reduce disk utilization.

2.5 Controlling Workloads: Introduction to Quality of Service

NetApp storage quality of service (QoS) gives the storage administrator the ability to monitor and control storage workloads delivering consistent performance that meets service objectives.

Consistent Performance

Storage QoS is a clustered Data ONTAP feature designed to help address the need for consistent workload performance. Storage resources are provisioned based on assumptions about workload IOPS or data throughput. These assumptions can be based on estimates or on empirical data collected from existing deployments. When these assumptions prove false, it is possible for some low-priority workloads to negatively affect high-priority workload service levels. Storage QoS instruments workload performance metering where it can be used to throttle offered load coming from less important workloads. This throttling limits low-priority workload resource consumption and assures adequate resources are always available for important work.

Workload Isolation

Some workloads should be isolated from others in the cluster. Rogue workloads, those that behave badly due to configuration errors or software defects, can consume a disproportional amount of shared resources, affecting all others. The ability to monitor and then isolate rogue workloads is valuable in environments where new applications are deployed with little to no control. Using a simple command line interface, storage QoS can be set up and applied immediately when the need arises.

2.5.1 Storage QoS Concepts

Before showing examples and use cases for storage QoS, it is important to understand some basic QoS concepts and terminology.

Workload

A workload is the set of I/O requests sent (or targeted) to one or more storage objects. In clustered Data ONTAP, QoS workloads include I/O operations and data throughput, measured in IOPS and MBps, respectively, that are targeted to a storage object

Storage Objects

A storage object is the target assigned to a QoS policy group for monitoring and control. QoS storage objects can be any of the following:

- Storage virtual machines (SVMs)
- FlexVol® volumes
- LUNs
- Files

Policies

QoS policies are defined by a QoS policy group and applied to storage objects. A QoS policy limits throughput to one or more storage objects in the QoS policy group. The throughput limit is applied collectively to the group. QoS policies may be configured to control IOPS or MBps throughput. The QoS policy limit may be configured to `none` to allow only instrumentation of performance metrics to the storage objects in the QoS policy group without limiting throughput.

Limits

The storage administrator can limit the workload throughput specifying IOPS or MBps limits. When the workload throughput exceeds the QoS policy limit, the workload is throttled at the protocol layer before entering the system. Actively throttling the workload increases I/O latency and some applications may time out. When this timeout occurs, it appears to applications that the system has run out of performance headroom. Throttling a workload in the protocol stack limits consumption of cluster resource that have been provisioned for other workloads on the cluster.

When the QoS policy is configured to throttle IOPS or MBPS, the specified policy value is a hard limit. The storage administrator should be aware that the workload IOPS or MBps throughput may exceed the value set in the QoS policy by up to 10% while the I/O operations are queued and throttled. As a general rule, the lower the QoS policy limits, the higher the deviation from the limit while the policy takes effect. I/O operations queued as a result of hitting the QoS policy limit do not affect cluster resources. QoS policies apply to all supported protocols, including NFS, SMB, SAN, iSCSI, and FCoE. Starting in clustered Data ONTAP 8.3, NFS 4.1 is supported.

Note: QoS is not compatible with NFS 4 prior to clustered Data ONTAP 8.3.

QoS is not compatible with pNFS in clustered Data ONTAP.

When to Use MBps

For large block I/O workloads, NetApp recommends configuring the QoS policy using MBps.

When to Use IOPS

For transactional workloads, NetApp recommends configuring the QoS policy using IOPS.

Policy Groups

QoS policy groups allow monitoring and controlling a set of storage objects (that is, SVMs, volumes, LUNs, or files). One QoS policy (behavior) can be assigned to a QoS policy group. The storage administrator can monitor storage object workloads by assigning the storage objects to a policy group without applying a QoS policy.

Note: Only one QoS policy may be applied to a QoS policy group.

Storage objects assigned to QoS policy groups are SVM scoped. This scoping means that QoS policy groups can only be assigned to one or more FlexVol volumes, LUNs, and files within the same SVM. The I/O limits are applied collectively across one or more storage objects in a policy group using a fair-share methodology.

Note: The QoS policy throughput limit is applied to the *combined* throughput of all storage object workloads assigned to the policy group.

Nested storage objects cannot be assigned to the same or a different QoS policy group. For example, a VMDK file and its parent volume may not both be assigned to a QoS policy group.

Note: Nested QoS policy groups are not supported. This fact means when a QoS policy group is assigned to an SVM the contained objects, volume, LUNs, and files, cannot be assigned to policy groups. The one exception to this restriction is Autovolumes discussed later.

QoS policy group membership remains unchanged as storage objects are moved within the cluster. However, as previously discussed, storage objects cannot be nested. For example, if a VMDK file, which is part of a policy group, is moved to a different datastore (volume), which is already part of a policy group, then the VMDK file is no longer assigned to the policy group.

Monitor

Starting in clustered Data ONTAP 8.3, volume workloads are automatically created and monitoring volume workload performance can begin without any configuration (see "Autovolumes"). In clustered Data ONTAP 8.2 and earlier or to monitor and control storage objects other than volumes in clustered Data ONTAP 8.3, storage objects must be assigned to policy groups as shown in this section.

Assigning storage objects to a QoS policy group without a QoS policy, or modifying an existing QoS policy limit to `none`, provides the ability to monitor the workload targeted to the storage objects with no throughput limit. In this configuration the storage administrator can monitor workload latency, IOPS, and data throughput. Storage QoS measures latency from the network interface to and from the disk subsystem.

Creating a QoS policy group to monitor workload latency and throughput:

```
Cluster1::> qos policy-group create -policy-group monitor_workload -vserver vserver1
```


Assigning volumes to a QoS policy group:

```
Cluster1::> vol modify -vserver vserver1 -volume vol1 -qos-policy-group monitor_workload
  (volume modify)

Volume modify successful on volume: vol1

Cluster1::> vol modify -vserver vserver1 -volume vol2 -qos-policy-group monitor_workload
  (volume modify)

Volume modify successful on volume: vol2
```

Assigning an SVM to a QoS policy group:

```
Cluster1::> vserver modify -vserver vserver2 -qos-policy-group vserver2_qos_policy_group
```

Displaying the QoS policy group configuration and the number of workloads assigned to the policy group:

```
Cluster1::> qos policy-group show
Name             Vserver     Class        Wklds Throughput
---------------- ----------- ------------ ----- ------------
monitor_workload vserver1    user-defined 2     0-INF
vol1_qos_policy  vserver1    user-defined 0     0-500IOPS
vol2_qos_policy  vserver1    user-defined 0     0-100MB/S
3 entries were displayed.
```

Note: QoS policy groups that do not have a throughput limit are shown with `0-INF`. This represents an infinite QoS policy limit.

Viewing the QoS policy group latency statistics:

```
cluster1::> qos statistics latency show
Policy Group     Latency    Network    Cluster    Data     Disk     QoS
-------------- ---------- ---------- ---------- -------- -------- -------
-total-             16ms        6ms        2ms      3ms      4ms     1ms
```

Viewing the QoS policy group performance statistics:

```
cluster1::> qos statistics performance show
Policy Group           IOPS      Throughput   Latency
-------------------- -------- --------------- ----------
-total-                 12224       47.75MB/s   512.45us
rogue_policy             7216       28.19MB/s   420.00us
prevent_policy           5008       19.56MB/s    92.45us
```

Autovolumes

Starting in clustered Data ONTAP 8.3, volume workloads are instantiated automatically. Thus, QoS volume workload statistics can be viewed without configuring a QoS policy group. This functionality is called autovolumes. For more information about using autovolumes for monitoring performance see Monitoring Workload Performance from Command Line Interface (CLI).

```
cluster1::> qos statistics volume latency show
Policy Group     Latency    Network    Cluster    Data     Disk     QoS
-------------- ---------- ---------- ---------- -------- -------- ------
vol1-wid12        170ms       15ms        0ms     10ms    120ms     0ms
vol3-wid302        30ms        5ms        0ms     25ms      0ms     0ms
cluster1::> qos statistics volume performance show
Policy Group           IOPS      Throughput    Latency
-------------------- -------- --------------- ----------
vol3-wid302               120          127MB/s   170.34ms
vol7-wid1441               30           10KB/s    30.13ms
```

Note: Autovolume commands require advanced privileges.

For more information about the QoS policy group monitoring commands, see the [Clustered Data ONTAP 8.3 Commands: Manual Page Reference](#).

Control

In clustered Data ONTAP 8.2 or later, QoS policy groups with a policy can control and limit the workloads of the storage objects assigned to the policy group. This capability provides the ability to manage and, when appropriate, throttle storage object workloads.

Creating a QoS policy group to control workload IOPS:

```
Cluster1::> qos policy-group create -policy-group vol1_qos_policy_group -max-throughput 500iops
            -vserver vserver1
```

Creating a QoS policy group to control workload data throughput:

```
Cluster1::> qos policy-group create -policy-group vol2_qos_policy_group -max-throughput 1000MBPS
            -vserver vserver1
```

Assigning volumes to QoS policy groups:

```
Cluster1::> vol modify -vserver vserver1 -volume vol1 -qos-policy-group vol1_qos_policy_group
  (volume modify)

Volume modify successful on volume: vol1

Cluster1::> vol modify -vserver vserver1 -volume vol2 -qos-policy-group vol2_qos_policy_group
  (volume modify)

Volume modify successful on volume: vol2
```

Assigning LUNs to QoS policy groups:

```
Cluster1::> lun modify -vserver vserver1 -lun lun1 -vol vol2
            -qos-policy-group lun_qos_policy_group
```

Assigning files to QoS policy groups:

```
Cluster1::> volume file modify -vserver vserver1 -vol vol2 -file log.txt
            -qos-policy-group file_qos_policy_group
```

Displaying the QoS policy group configuration and the number of workloads assigned to the policy group:

```
Cluster1::> qos policy-group show
Name             Vserver     Class        Wklds Throughput
---------------- ----------- ------------ ----- ------------
monitor_workload vserver1    user-defined 0     0-INF
vol1_qos_policy_group
                 vserver1    user-defined 1     0-500IOPS
vol2_qos_policy_group
                 vserver1    user-defined 1     0-100MB/S
3 entries were displayed.
```

For more information about the QoS policy group monitoring commands, see the [Clustered Data ONTAP 8.3 Commands: Manual Page Reference](#).

Storage QoS Summary

The storage QoS capability in NetApp clustered Data ONTAP 8.2 or newer improves utilization of storage resources by consolidating multiple workloads in a single shared storage infrastructure, while minimizing the risk of workloads affecting each other's performance. Administrators can prevent tenants and applications from over-consuming provisioned resources in the storage infrastructure, improving the end-user experience and application uptime. In addition, predefining service-level objectives allows IT to provide different levels of service to different stakeholders and applications, making sure that the storage infrastructure continues to meet the business needs.

Storage QoS adds new capabilities for the storage administrator to monitor and control user workloads. Following is a brief summary of the QoS functionality delivered starting in clustered Data ONTAP 8.2:

- Monitor and manage storage object workloads
- Control I/O and data throughput workloads on SVMs, volumes, LUNs, and files
- Provide multiprotocol support, including SMB, SAN, iSCSI, FCoE, and NFS
- Provision policy groups in Workflow Automation (WFA) 2.1 and newer
- Provide QoS support for V-Series

However, there are a few caveats to remember when you consider QoS:

- QoS is not supported on Infinite Volumes.
- Alerts may not be configured for QoS.
- QoS does not provide workload guarantees.
- QoS is not supported with pNFS.

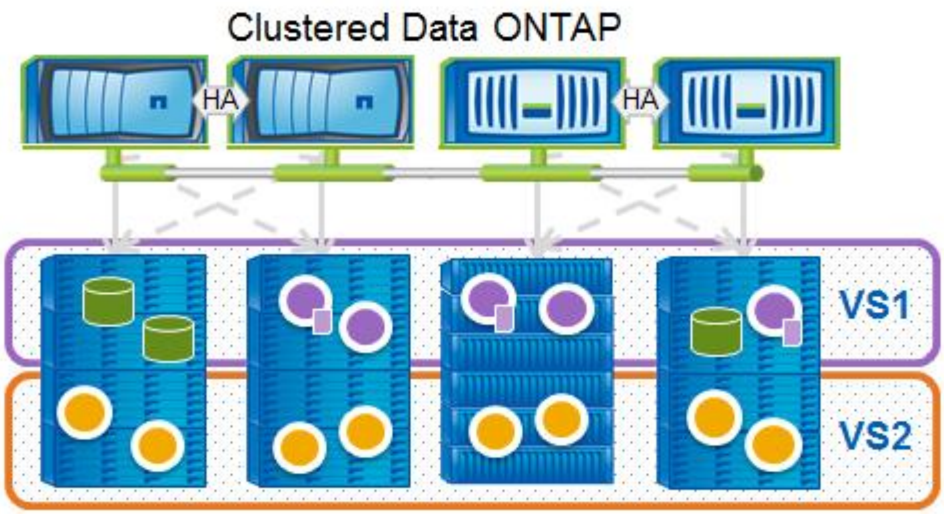
Table 1) QoS limits.

QoS Feature Area	Maximum	
	Per Node	Per Cluster
QoS policy groups supported	3,500	3,500
Number of controllers supported by QoS	N/A	8
Storage objects assigned to a QoS policy group	10,000	10,000

2.5.2 Examples of Using Storage QoS

Storage QoS has many applications for the storage administrator. The following are a few scenarios that illustrate QoS capabilities. The first use case is an example in which the storage administrator throttles a “rogue” workload that is affecting other workloads. The second scenario describes how a storage administrator may prevent runaway (or rogue) workloads by proactively setting QoS policies. The final use case looks at managing workloads so that service providers can meet their service-level agreements (SLAs).

Figure 12) Storage QoS.



Reactively Respond

In this scenario the storage administrator has not applied any storage objects to a QoS policy group. By default, Data ONTAP treats all storage objects on a best-effort basis. However, one of the storage objects (that is, a volume) has a rogue workload affecting the performance of other workloads on the system. Using Data ONTAP `statistics` and `qos statistics` commands, the storage administrator can identify the rogue workload. After the rogue workload is identified, the storage administrator can use storage QoS to isolate the workload by assigning it to a QoS policy group and applying a throughput limit.

Throttle Rogue Workloads

After identifying the rogue workload, the storage administrator creates a new QoS policy group and sets the I/O throughput limit to 1,000 IOPS.

```
Cluster1::> qos policy-group create vol1_rogue_qos_policy -max-throughput 1000iops
            -vserver vserver1
```

To view the QoS policy group configuration, the storage administrator may use the `qos policy-group show` command.

```
Cluster1::> qos policy-group show -policy-group vol1_rogue_qos_policy

              Policy Group Name: vol1_rogue_qos_policy
                        Vserver: vserver1
                           Uuid: a20df2c2-c19a-11e2-b0e1-123478563412
             Policy group class: user-defined
                Policy Group ID: 102
             Maximum Throughput: 1000IOPS
            Number of Workloads: 0
              Throughput Policy: 0-1000IOPS
```

Next, the offending volume is assigned to the QoS policy group to begin throttling the rogue workload. The `modify` option of storage objects is used to assign an existing storage object to a QoS policy group.

```
Cluster1::> volume modify -vserver vserver1 -volume vol1 -qos-policy-group vol1_rogue_qos_policy
```

The storage administrator can verify that the volume has been assigned to the QoS policy group using the `volume show` command.

```
Cluster1::> volume show -vserver vserver1 -volume vol1 -fields qos-policy-group
```



```
vserver  volume qos-policy-group
-------- ------ ---------------------
vserver1 vol1   vol1_rogue_qos_policy
```

Proactively Prevent Runaway Workloads

This scenario is one in which the storage administrator proactively sets a QoS policy group for the storage objects to prevent the impact of new, and possibly runaway, workloads. This situation may arise in a large virtualized environment in which the storage administrator needs to prevent a development or test application from affecting other production applications.

Apply Limits Before a Problem Occurs

The first step is to create a QoS policy group and apply a throughput limit.

```
Cluster1::> qos policy-group create -policy-group vmdk_13_qos_policy_group
            -max-throughput 100iops -vserver vserver1
```

After the QoS policy group has been created, the storage objects are assigned to the policy group. It is important to remember that the QoS limit is applied to the aggregate throughput of all storage objects in the QoS policy group.

```
Cluster1::> volume file modify -vserver vserver1 -vol vol2 -file vmdk-13.vmdk
            -qos-policy-group vmdk_13_qos_policy_group
```

Lastly, the storage administrator should monitor the storage object workload and adjust the policy as needed. Changes to the policy throughput limit can be completed quickly without affecting other workloads.

```
Cluster1::> qos statistics performance show
Policy Group           IOPS      Throughput   Latency
-------------------- -------- --------------- ----------
-total-                   867       47.75MB/s   512.45us
vol1_rogue_qos_policy     769       28.19MB/s   420.00us
vmdk_13_qos_policy_group   98       19.56MB/s    92.45us
```

After reviewing the required IOPS resources for vmdk-13 with engineering, the storage administrator agrees to increase the policy throughput limit to 200 IOPS.

```
Cluster1::> qos policy-group modify -policy-group vmdk_13_qos_policy_group
            -max-throughput 200iops
```

Isolate Tenants with Per-SVM Throughput Limits

In our final use case we look at a service provider who needs to isolate customer workloads to meet the service-level agreements. A new SVM is created in the cluster for each customer, and the service provider must enable workloads to be controlled based on the SLA level. This service provider has three SLA levels—Bronze, Silver, and Gold—corresponding to the maximum data throughput allowed.

Table 2) SLA levels.

SLA Level	Data Throughput
Bronze	100MBPS
Silver	200MBPS
Gold	400MBPS

After the service provider determines the SLA throughput limits, the storage administrator can create the storage QoS policy group with the appropriate limit and assign the SVM storage object to the policy group. For this example, we use three fictional service provider customers—Acme, Bravos, and Trolley—who have purchased the Bronze, Silver, and Gold service levels, respectively.

Create a policy group with the appropriate throughput limit (determined by service level) and assign the SVM for each customer to the policy group:

```
Cluster1::> qos policy-group create -policy-group acme_svm_bronze -max-throughput 100MBPS
            -vserver acme_svm

Cluster1::> qos policy-group create -policy-group bravos_svm_silver -max-throughput 200MBPS
            -vserver bravos_svm

Cluster1::> qos policy-group create -policy-group trolley_svm_gold -max-throughput 400MBPS
            -vserver trolley_svm
```

Apply the SVM for each customer to the QoS policy group:

```
Cluster1::> vserver modify -vserver acme_svm -qos-policy-group acme_svm_bronze

Cluster1::> vserver modify -vserver bravos_svm -qos-policy-group bravos_svm_silver

Cluster1::> vserver modify -vserver trolley_svm -qos-policy-group trolley_svm_gold
```

2.6 Performance Management with Clustered Data ONTAP

Assuring storage system performance is essential throughout its deployed lifecycle. Some applications have more stringent performance requirements than others, but there is typically some level of performance requirement. The ability to understand workloads, identify problems, and relate them back to the system's operation is essential to achieving performance goals.

This section introduces the capabilities of Data ONTAP operating system and other NetApp software to complete performance management functions, including looking at statistics and using features to alter the performance of the system or workloads.

2.6.1 Basic Workload Characterization

As mentioned earlier, the workload characteristics and system architecture ultimately define the performance of the system. Also mentioned earlier were the storage QoS capabilities available in clustered Data ONTAP 8.3. You can use the statistics generated by the QoS command line interface (CLI) to monitor and gain a basic understanding of workloads in the system. These insights can then be used to confirm initial sizing estimations, refine sizing forecasts, or simply set performance baselines and expectations. When reviewing this data, keep in mind the relationships introduced in section 2.2.

The following command shows the workload characteristics of the busiest workloads on the system:

```
Cluster1::> qos statistics workload characteristics show
Workload          ID     IOPS      Throughput      Request size    Read  Concurrency
--------------- ------ -------- ---------------- --------------- ------- -----------
-total-              -     5076        37.88MB/s           7825B     65%           2
volume_a         14368     4843        37.82MB/s           8189B     68%           2
...
```

Although this is just an example, the preceding data shows that the system is processing about 5,000 IOPS, roughly 8KB in size, with 65% being reads.

Entire policy group characteristics can be viewed by eliminating the workload part of the command, as in the following:

```
Cluster1::> qos statistics characteristics show
```

These examples are basic, and more statistics than are presented here are available in Data ONTAP.

Overview of Application Workload Characteristics

Ultimately every production workload is unique due to the many variables that contribute to individual behavior. However, to properly set expectations and for instructional purposes, some basic generalizations about application workloads are presented here.

Write Work

As mentioned earlier, clustered Data ONTAP operating system is highly optimized to efficiently perform writes by taking advantage of NVRAM, system memory, and consistency point logic to optimize the on-disk layout of blocks written. This reduces the effects of writing to and later reading from slower disk storage media. Thus sequential and random writes are, for all practical purposes, instantly recorded in memory, permanently logged in local and partner NVRAM, and the response immediately sent to the application. Then based on time thresholds or NVRAM usage thresholds, writes are flushed to a slower persistent medium while Data ONTAP continues to service user operations. Exceptions can occur when unexpected situations are encountered, such as CPU or disk utilization issues causing resource constraints, excessive loads (along with concurrency) causing back-to-back CPs, or file system disk layout issues resulting in unnecessary disk I/O. None of these exceptions should occur in a properly operating and designed system.

Sequential Read Work

The Data ONTAP read-ahead engine detects common sequential workload read patterns to efficiently cache data before it is requested from disk. This caching in combination with the previously written layout optimizations contributes to greatly reducing delays associate with disks. Thus workloads with highly sequential read patterns, given adequate resources, should experience low service times by avoiding costly disk accesses.

Random Read Work

Some workloads are inherently more difficult to handle than others. Repeated random access for data is rarely a problem provided most of the working set fits in caches. Random read workloads should experience a large percentage of cache hits and thus fewer disk reads resulting low average service times. However, caches are shared resources and can be oversubscribed when shared by too many applications. In addition, random read workloads with large working sets and even larger datasets make it very difficult or even impossible to predict what data will be needed. Thus, under some unusual circumstances, this causes the storage system to frequently reach for data on a slow disk medium, increasing service times.

Indirect I/O Work

When considering indirect workloads, it is tempting to conclude that the cluster interconnect is a potential source of additional service delay (or latency). However, when observed under normal operating conditions, the actual effects are minimal and contribute negligible delays in service time.

2.6.2 Observing and Monitoring Performance

Monitoring performance avoids potential problems and aids in determining whether to add more load to a cluster. Latency is used as a key indicator to determine if there are performance issues. In other words, if there is no latency issue, there is no performance issue. Corroborating metrics for latency include throughput and resource utilizations which means that if unacceptable latency is observed alongside increased throughput, a workload may have grown and thus be saturating a resource in the system (confirmed by observing resource utilization). Low throughput is not necessarily a problem, because clients and applications may simply not be requesting that work be done.

Data ONTAP operating system collects statistics that can be collected through graphical tools as well as through the cluster command line interface (CLI) and application programming interface (API). The following subsections introduce methods to monitor performance metrics using NetApp tools and Data

ONTAP features. In advanced or more complex environments, these CLI commands and related APIs from the NetApp SDK can be used to create custom monitoring tools.

OnCommand Performance Manager

NetApp makes the OnCommand® portfolio suite of management tools available to customers. OnCommand Performance Manager, an integrated component of OnCommand Unified Manager and part of the OnCommand portfolio, is designed for clustered Data ONTAP® operating system. There is an entire section later dedicated to showing the capabilities of Performance Manager Version 2.0 (see section 3 Performance Management with NetApp OnCommand Portfolio). It is made available with clustered Data ONTAP 8.3.1. Performance Manager is designed to eliminate much of the complexity surrounding performance management and automatically alerts when significant performance events occur.

Best Practice
It is highly recommended that OnCommand Performance Manager monitor all clustered Data ONTAP systems.

Simplicity is achieved through minimizing performance configuration settings and automation of typically complex tasks. Upon pointing Performance Manager to one or more clusters, it will automatically:

- Discovers storage objects of interest
- Establishes performance baselines and thresholds

Note: Retains 13 months of performance data: 30 days at five-minutes granularity and 12 months at one hour granularity

- Detects performance threshold breaches and alerts administrator through integration with OnCommand Unified Manager and e-mail
- Identifies root cause performance degradation by correlating source of resource contention
- Suggests remediation tasks to resolve issue
- Exports retained data for external monitoring, reporting, and auditing

As a simple example, the screen shot in Figure 14 shows a previously encountered incident where the volume under consideration was identified by Performance Manager as a victim.

Figure 13) Performance Manager victim workload with annotations 1 and 2.

The sample screen shot in **Error! Reference source not found.** depicts an Performance Manager victim incident with:

1. Increased workload latency (or response time). The latency graph plots the point in time (red dot) when the metrics exceeded the automatically established threshold (gray bands). The line color of the graph line also changes to red and remains for the duration of the incident.
2. Lower workload throughput (or operations) correlates with incident detection.

More information about Performance Manager can be found at the NetApp Support site documentation portal in the [OnCommand Performance Manager User Guide].

More information about Performance Manager integration with OnCommand Unified Manager can be found in the [OnCommand Unified Manager Administration Guide]..

Storage QoS CLI

Starting in clustered Data ONTAP 8.2, the QoS throughput policy is configured by setting either an I/O limit (IOPS) or a data throughput limit (bytes per second). Table 3 provides a list of available QoS CLI labels. Continuing in clustered Data ONTAP 8.3, volume workloads are automatically created and can be used without configuration (see Monitoring Workload Performance from Command Line Interface (CLI)).

Table 3) QoS throughput labels.

QoS Storage Unit	QoS CLI Labels
IOPS	IOPS, iops, io/s
Bytes/second	Mb/s, MB/s, mb/s, MB/S, MBps, mbps, B/s, B/S, b/s, bps

Note: The data throughput limit can only be specified in bytes/second, including megabytes/second.

Observing Throughput and Latency

Workload-level statistics are available after storage objects are assigned to a QoS policy group. These statistics can be displayed using the QoS statistics CLI commands.

As discussed earlier, throughput and latency are significant metrics. Similar to workload characteristics, the throughput and latency of the system, policy group, or workloads can be determined by using the following command:

```
Cluster1::> qos statistics workload performance show
Workload            ID     IOPS       Throughput    Latency
--------------- ------ -------- ---------------- ----------
-total-              -     5060        37.97MB/s   492.00us
volume_a         14368     4847        37.86MB/s   510.00us
...
```

More detailed latency information is also available by using the following command:

```
Cluster1::> qos statistics workload latency show
Workload            ID    Latency    Network    Cluster       Data       Disk        QoS
--------------- ------ ---------- ---------- ---------- ---------- ---------- ----------
-total-              -   608.00us   270.00us        0ms   148.00us   190.00us        0ms
volume_a         14368   611.00us   270.00us        0ms   149.00us   192.00us        0ms
```

The output describes the latency encountered at the various components in the system discussed in previous sections. Using this output, it's possible to observe where most of the latency is coming from for a specific workload:

- **Latency.** Refers to the total latency observed.
- **Network.** The amount of latency introduced by the network-level processing in Data ONTAP.
- **Cluster.** The amount of latency introduced by the cluster interconnect.
- **Data.** The amount of latency introduced by the system, except latency from the disk subsystem.
- **Disk.** The effective amount of latency introduced by the disk subsystem. Note that any reads that were serviced by the WAFL cache do not have a disk latency component because those operations did not go to disk.
- **QoS.** The amount of latency introduced by queuing by QoS if throughput limits have been established.

Observing Resource Utilizations

QoS also enables users to view disk and CPU utilizations for a policy group or workload. This output can help point out workload resource utilization and identify bullies. For instance, if workload A is a very important workload and shows high latencies, and in the resource utilizations shows workload B is using a lot of a system resource, the resource contention between workload A and workload B could be the source of the latency. Setting a limit or moving workload B could help alleviate workload A's latency issue. Resource utilizations are provided on a per-node basis.

```
Cluster1::> qos statistics workload resource cpu show -node Node-01
Workload           ID   CPU
--------------- ----- -----
-total- (100%)      -   29%
volume_a        14368   12%
System-Default      1   10%
...
```

```
Cluster1::> qos statistics workload resource disk show -node Node-01
Workload           ID  Disk  No. of Disks
--------------- ------ ----- ------------
-total-              -    4%           29
volume_a         14368    5%           22
System-Default       1    1%           23
...
```


Viewing Cluster-Level and Node-Level Periodic Statistics

Clustered Data ONTAP operating system includes commands to show statistics beyond logical workloads. To see an overview of physical resource utilization use `statistics show-periodic`. The output from this command provides cluster-level details on the number of operations and resource utilization.

Note: Use Ctrl-C to stop scrolling statistics and print a summary.

This command should be run in "advanced" privilege to get more information:

```
TestCluster::> set -privilege advanced

Warning: These advanced commands are potentially dangerous; use them only when directed to do so
by NetApp personnel.
Do you want to continue? {y|n}: y
```

The following example shows cluster-level performance. The output is too wide to fit in the document, so it is divided between two output blocks.

```
TestCluster::*> statistics show-periodic
cluster:summary: cluster.cluster: 6/7/2013 18:27:39
  cpu  cpu    total                   fcache    total    total data     data     data cluster …
  avg busy      ops  nfs-ops cifs-ops      ops     recv     sent busy     recv     sent    busy …
 ---- ---- -------- -------- -------- -------- -------- -------- ---- -------- -------- ------- …
  58%  91%    17687    17687        0        0    156MB    133MB  59%   68.7MB   47.3MB      4% …
  65%  92%    18905    18905        0        0    199MB    184MB  84%    103MB   74.9MB      6% …
  54%  86%    17705    17705        0        0    152MB    132MB  58%   68.9MB   47.2MB      4% …
cluster:summary: cluster.cluster: 6/7/2013 18:27:47
  cpu  cpu    total                   fcache    total    total data     data     data cluster …
  avg busy      ops  nfs-ops cifs-ops      ops     recv     sent busy     recv     sent    busy …
 ---- ---- -------- -------- -------- -------- -------- -------- ---- -------- -------- ------- …
Minimums:
  54%  86%    17687    17687        0        0    152MB    132MB  58%   68.7MB   47.2MB      4% …
Averages for 3 samples:
  59%  89%    18099    18099        0        0    169MB    150MB  67%   80.3MB   56.5MB      4% …
Maximums:
  65%  92%    18905    18905        0        0    199MB    184MB  84%    103MB   74.9MB      6% …
```

```
… continued

… cluster  cluster     disk     disk     pkts     pkts
…    recv     sent     read    write     recv     sent
…  -------- -------- -------- -------- -------- --------
…  87.6MB   86.3MB   96.4MB    139MB    87861    75081
…  96.2MB    109MB    108MB    261MB   127944   111190
…  84.0MB   85.4MB   69.6MB    101MB    87563    75402

… cluster  cluster     disk     disk     pkts     pkts
…    recv     sent     read    write     recv     sent
…  -------- -------- -------- -------- -------- --------

…  84.0MB   85.4MB   69.6MB    101MB    87563    75081

…  89.3MB   93.8MB   91.6MB    167MB   101122    87224

…  96.2MB    109MB    108MB    261MB   127944   111190
```

The following command shows the same information for a single node within the cluster.

```
TestCluster::*> statistics show-periodic -object node -instance node -node Node-01
```

Note: When reviewing CPU information in Data ONTAP, CPU AVG is a better indicator of overall CPU utilization compared to CPU BUSY.

Monitoring Workload Performance from Command Line Interface (CLI)

Clustered Data ONTAP operating system provides a large set of commands for monitoring performance. This section describes how to use a small set of commands to gain insight into workload performance on a clustered Data ONTAP system. It is important to note that as a performance primer, the information presented here is for monitoring and instructional purposes only. This should not be mistaken for troubleshooting workflows or advanced diagnostics.

The recommended method for monitoring performance on a clustered Data ONTAP system is to use volume workloads. Clustered Data ONTAP 8.3 and later makes this easy through the use of the newly introduced autovolumes feature (see the section on autovolumes for an introduction). At this point, it is worth noting a distinction between volume objects versus volume workloads. A volume object is the target of a volume workload. A volume workload object contains performance metrics that record the time (latency) and resource utilization (processing) of operations across the cluster. In clustered Data ONTAP systems, an operation enters the cluster on one node, can get transported across the cluster interconnect, and can get serviced on a different node. The total time it takes to complete the operation is otherwise known as operation residency time and is recorded in the volume workload as a latency metric. The total amount of resource used to complete the operation is recorded as utilization metrics. The total amount of work completed over a given time is recorded as throughput metrics. The metrics are expressed in average units per second when depicting throughput and in average units of time per operation when depicting latency. Volume objects are irrelevant in this monitoring scenario and are used for alternative workflows beyond the scope of this document.

Thus volume workloads are the primary mechanism for gaining visibility into the service times provided to applications using a volume. The recommended commands to use for CLI performance monitoring are summarized in Table 4.

Table 4) Recommended performance-monitoring commands.

Command	Description
`qos statistics volume performance show`	View volume workload operations per second, data throughput, and clusterwide latency
`qos statistics volume characteristics show`	View volume workload operation payload size and concurrency
`qos statistics volume latency show`	View clusterwide latency contribution at the cluster component level
`qos statistics volume resource cpu show`	View CPU resource consumption attributed to volume workload
`qos statistics volume resource disk show`	View disk resource utilization attributed to volume workload

The following examples provide general guidance on interpreting command output for each of the individual commands listed in Table 4.

```
ontaptme-fc-cluster::*> qos statistics volume performance show
Workload            ID      IOPS       Throughput    Latency
--------------- ------ -------- ---------------- ----------
-total-              -      4124        29.80MB/s  1198.00us
bobvol1-wid13..  13891      2474         6.77MB/s  1015.00us
bobvol5-wid9864   9864      1650        23.03MB/s  1472.00us
```


Use the preceding command to view overall latency on volume workloads and get a sense of work performed on the cluster.

- **Workload.** Concatenation of volume name and internal workload identifier.
- **ID.** Internal workload identifier.
- **IOPS.** Average number of operations processed every second.
- **Throughput.** Average amount of payload data moved into and out of the workload every second.
- **Latency.** Average operation residency time into and out of the workload.
- **-total-.** For throughput metrics, the sum of averages of all workloads on the cluster. For latency, the average of average latencies on the cluster.

```
ontaptme-fc-cluster::*> qos statistics volume characteristics show
Workload             ID     IOPS       Throughput Request Size Read Concurrency
--------------- ------ -------- ---------------- ------------ ---- -----------
-total-               -     4179        30.95MB/s        7766B  45%           5
bobvol1-wid13..  13891     2417         6.58MB/s        2856B  35%           3
bobvol5-wid9864   9864     1761        24.37MB/s       14513B  59%           2
```

More can be learned about the offered load presented to the volume using the QoS volume characteristics command shown earlier. In particular, the concurrency calculation shows the level of application offered load in relation to cluster consumption of that load. Both these factors individually are highly complex and do not provide enough information to draw any major conclusions. However, concurrency does provide insight into application behavior such as the request arrival rate:

- **Workload.** Concatenation of volume name and internal workload identifier.
- **ID.** Internal workload identifier.
- **IOPS.** Average number of operations processed every second.
- **Request size.** Calculation of throughput divided by IOPS. Given that all the metrics available are averages, this calculation is the best that can be done. For more detailed request size information, a histogram is required.
- **Read.** Percentage of workload that is read operations. The remaining percentage is write operations for SAN protocols and is the sum of writes and meta-operations (or other) for NAS protocols.
- **Concurrency.** Product of latency and IOPS; see "Little's Law: A Relationship of Throughput, Latency, and Concurrency." This result is the number of operations resident in the cluster being serviced at a given point in time.
- **-total-.** For throughput and concurrency metrics, the sum of averages of all workloads on the cluster. For remaining metrics, the average of averages on the cluster.

In the following example, the output is too wide to fit in the document, so it is divided between two output blocks.

```
ontaptme-fc-cluster::*> qos statistics volume latency show
Workload             ID    Latency    Network    Cluster       Data       Disk
--------------- ------ ---------- ---------- ---------- ---------- ----------
-total-               -  1391.00us    74.00us    41.00us    56.00us  1203.00us
bobvol5-wid9864    9864     2.17ms    90.00us   162.00us    69.00us     1.83ms
bobvol1-wid13..   13891  1126.00us    69.00us        0ms    51.00us   988.00us
```

```
..continued
        QoS      NVRAM
 ---------- ----------
        0ms    17.00us
        0ms    14.00us
        0ms    18.00us
```


The volume latency command breaks down the latency contribution of the individual clustered Data ONTAP components (see section2.4.1, earlier). Among the monitoring commands presented here, this one is possibly the most useful in that it provides visibility into workload internals across the cluster. It is worth repeating that latency is the equivalent of operation residency time and is the sum of operation wait time and execution time for a given component. In the preceding example, it can be seen that workload bobvol5 is an indirect workload averaging 2.17ms latency. The largest portion of that latency is attributed to the 1.83ms disk contribution. The remaining component contributions which account for very little of the total reported latency, are CPU execution time, internal queuing delays, network and cluster interconnect transmission delay. This command does not show every detail. However, when considering a single workload under normal operating conditions, execution time is almost always far less significant in relation to wait time (in this example disk wait time):

- **Workload.** Concatenation of volume name and internal workload identifier.
- **ID.** Internal workload identifier.
- **Latency.** Overall average operation residency time into and out of the cluster.
- **Network.** Average latency contribution of the server (or client) facing transport component including the delay introduced by front-end SAN or NAS transport stacks.
- **Cluster.** Average latency contribution of the cluster interconnect transport component. This latency measurement is the delay introduced by the cluster interconnect transport protocols.
- **Data.** Average latency contribution of the clustered Data ONTAP proprietary file system known as WAFL. After the operation has been transported by the underlying protocols, it is processed by WAFL and response returned back to the protocols for delivery as rapidly as possible.
- **Disk.** Average latency contribution of the physical disks.
- **QoS.** Applicable only when QoS policy limits are in place. When actively limiting, this latency measurement is the average incurred to enforce the user-defined policy limit.
- **NVRAM.** Average latency incurred to replicate write operations in NVRAM to the high-availability (HA) and/or cluster partner.
- **-total-.** The average of average latencies on the cluster.

```
ontaptme-fc-cluster::*> qos statistics volume resource cpu show -node ontaptme-fc-cluster-03
Workload           ID   CPU
--------------- ----- -----
-total- (100%)      -    5%
bobvol1-wid13.. 13891    3%
bobvol5-wid9864  9864    1%
```

Some workloads are more expensive than others in terms of CPU utilization due to application-specific behavior. Thus it is useful to know how some workloads consume shared CPU resource in relation to others. The preceding volume resource CPU command displays the specific CPU utilization for a given volume workload. Note that this in no way represents the total physical node-level CPU utilization discussed earlier. In addition, there are many internal Data ONTAP processes that can be running not accounted for here. It should also be noted that indirect workloads are present here to account for the transport protocol CPU overhead (see bobvol5-wid9864 earlier):

- **Workload.** Concatenation of volume name and internal workload identifier.
- **ID.** Internal workload identifier.
- **CPU.** Processor resource utilization attributed to the workload.
- **-total-.** Sum of all the workload CPU utilization metrics.

```
ontaptme-fc-cluster::*> qos statistics volume resource disk show -node ontaptme-fc-cluster-03
Workload            ID   Disk Number of HDD Disks
--------------- ------ ------ -------------------
-total-              -    15%                  12
bobvol5-wid9864   9864    26%                   7
```


In a similar fashion to CPU, physical disks are a limited shared resource where some workloads consume more than others. That is where the similarities end, though. In the preceding command context, disk utilization represents the amount of time a disk is busy servicing requests on behalf of the volume workload. This is indeed a major contributing factor to disk component latency described in section 2.4.1, "Cluster Operations." Disk utilization (and thus disk latency) can widely vary among workloads due to many factors previously discussed such as data access patterns, working set size, disk free space, and cache resources. Unlike the volume resource CPU command, the volume resource disk command only includes workloads that are local to the node because it is fundamentally impossible for disk (or aggregate) utilization to be split across multiple nodes:

- **Workload.** Concatenation of volume name and internal workload identifier.
- **ID.** Internal workload identifier.
- **Disk.** Average disk utilization attributed to the workload.
- **-total-.** Average disk utilization of all disks on node.

2.6.3 Managing Workloads with Data Placement

QoS is a very valuable tool to manage workloads within the cluster; however, the location and access path of data in the cluster can also play a role in performance, as was mentioned earlier. Clustered Data ONTAP has features that allow data to be moved, cached, and duplicated across nodes in the cluster to help manage performance.

DataMotion for Volumes

Independent of protocol, volumes can be moved and mirrored in the storage layer. Using volume move (vol move), volumes can be moved to the node handling the most client access to increase direct access. Using the same method, volumes can be moved to different disk types or nodes with different hardware to achieve different performance characteristics. Volume moves should be used to proactively manage performance and not when encountering performance problems, because volume move requires resources to perform the movement.

3 Performance Management with NetApp OnCommand Portfolio

When running a data center and operating storage systems, various questions come to mind:

- What is the performance status of my storage systems?
- What needs my attention now?
- How is my storage system performing in detail?
- Are there trends that may cause future issues?
- Am I meeting all critical service-level objectives and agreements?
- What is abnormal and why?

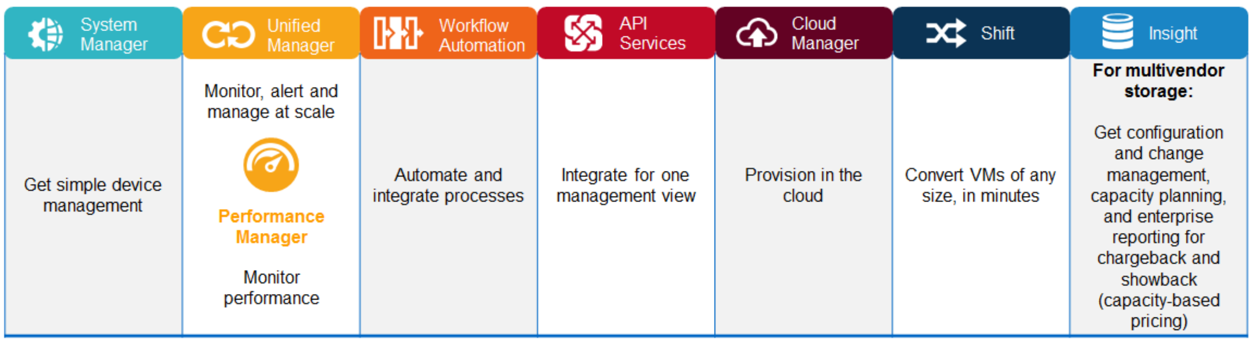
Storage performance management is a critical part of data center infrastructure management. The technical challenges associated with this can be significant, often leading to additional and unwelcome operational costs. Thus, employing automated tools to reduce or eliminate those costs is vital. OnCommand Performance Manager, an integrated component of OnCommand Unified Manager, reduces costs by continuously collecting, analyzing, and retaining performance data from the entire storage environment and facilitating performance visualization.

3.1 OnCommand Performance Manager Overview

Performance Manager does much more than simply monitor and plot data graphs. It is a highly sophisticated performance management tool purpose built for clustered Data ONTAP® operating system.

It requires minimal configuration and setup to get started. It constantly collects performance data at five-minute intervals with minimal operational interference and retains data for 13 months. Performance Manager eliminates the need for retaining in-house clustered Data ONTAP performance subject-matter experts. It encapsulates the subject-matter performance expertise required to run a clustered Data ONTAP storage environment in a single tool included with the purchase of the storage system. It expertly analyzes clustered Data ONTAP performance data and generates attention-grabbing events based upon user-defined and system-defined thresholds.

Figure 14) NetApp OnCommand portfolio

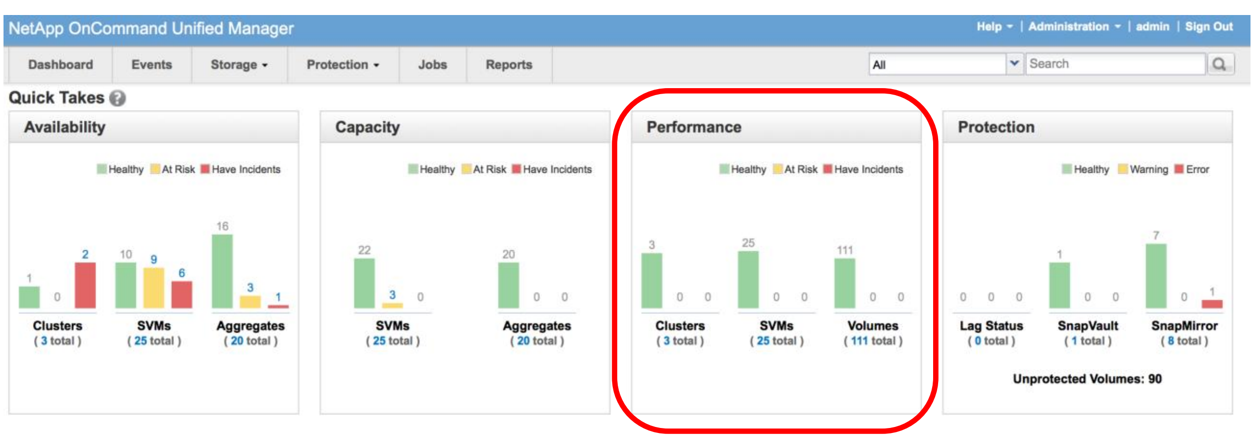


OnCommand Performance Manager is part of the NetApp OnCommand management portfolio (see Figure 14) and is a component of OnCommand Unified Manager (see Figure 15). Many of the features and functions discussed in this document are managed by the tools in this portfolio. For example, QoS policy groups can be defined and applied from within the System Manager graphical user interface.

In general, Performance Manager provides the following benefits:

- Visualization of storage performance
- Alerts on performance events automatically
  - Built-in predefined alerts
  - Automatically established dynamic alerts
  - User-defined thresholds alerts
- Quick elimination of storage as the source of a performance issue

Figure 15) OnCommand Unified Manager dashboard with integrated performance



These benefits are realized through implementation of the following features:

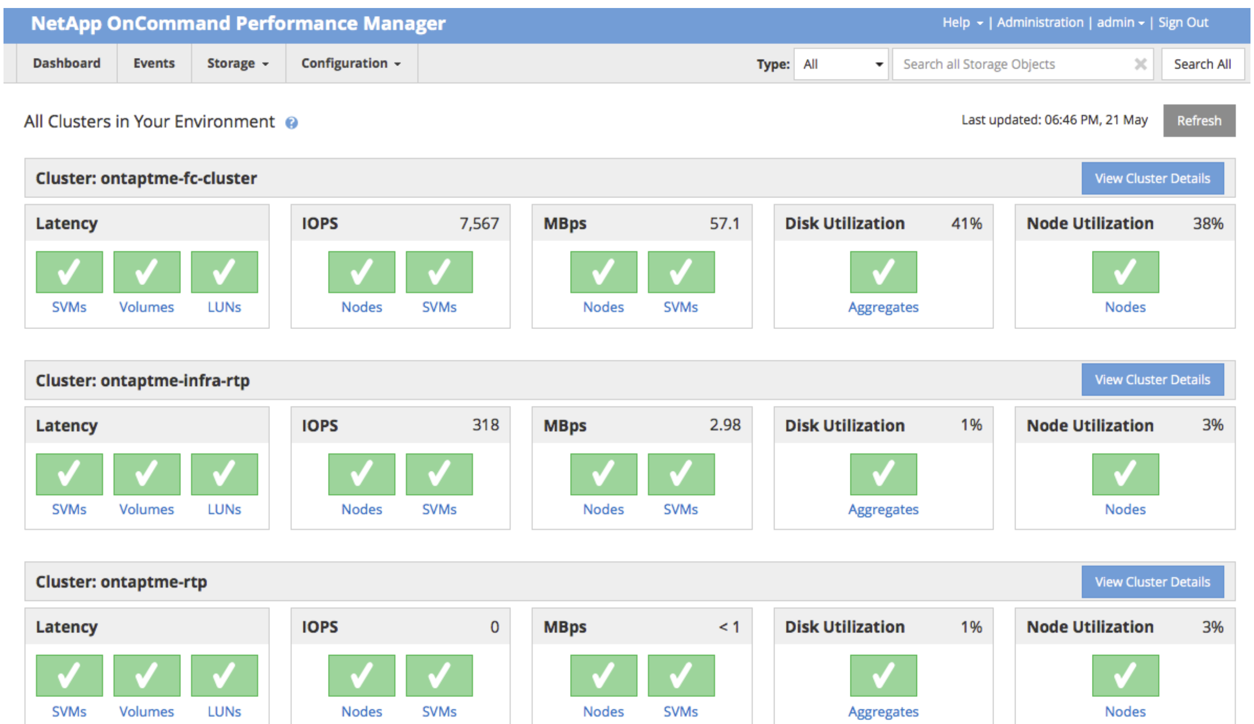
- A comprehensive performance dashboard for entire storage environment
- A landing page for the storage objects of interest (such as cluster, volume, and LUN)
- A performance explorer to view charts comparing related objects and metrics

3.2 OnCommand Performance Manager: Dashboard

Ideally, storage environments would manage themselves without any human intervention. The next best option is a tool indicating what demands attention now. OnCommand® Performance Manager does this managing by composing a cluster dashboard listing clusters that are of most interest first. The order of precedence is unreachable clusters first, actively alerting clusters second, and most active clusters next. The dashboard also presents other high-level information such as key performance metrics, utilization of the most active resources, and simple color-coded alert status indicators.

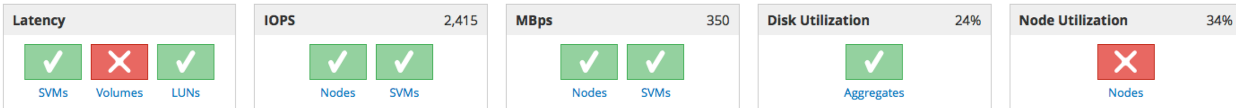
The example dashboard in **Error! Reference source not found.** shows three clusters. From this figure it is learned that there are no active alerts because all color-coded status indicators are green, the cluster called *ontaptme-fc-cluster* is most active at 7,567 IOPS, the most active disk aggregate is running at 41% busy, and the most active node is operating at 38% utilization.

Figure 16) OnCommand Performance Manager dashboard



When new alerts are present, then the color-coded status indicators change to red. For example, the cluster status shown in **Error! Reference source not found.**indicates there are an active volume and a node alert due to a latency threshold breach.

Figure 17) Actively alerting cluster on dashboard



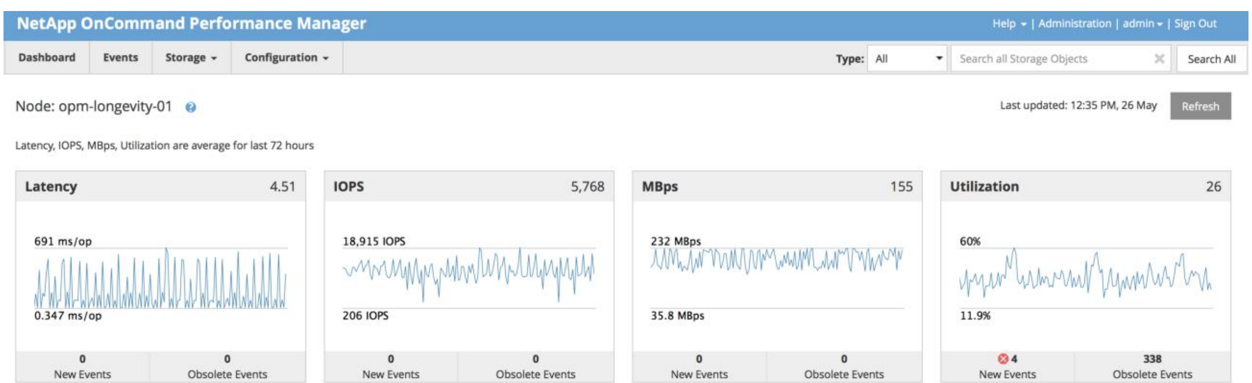
3.3 OnCommand Performance Manager: Performance Visualization

The visualization of storage performance makes it possible to see how storage metrics and objects relate to each other, observe how relationships change over time, and identify trends to avoid potential performance issues. One way Performance Manager accomplishes this task is through the storage object landing page and the performance explorer showing various customizable views.

OnCommand Performance Manager – Storage Object Landing Page

The Performance Manager object landing page focuses on a specific storage object where summary and detailed performance information is presented. All storage objects have a landing page and have a similar look and feel. Obviously not all objects are the same, so a cluster landing page has slightly different metrics and views than a volume landing page. However, all landing pages summarize key high-level metrics and categorizes events for a given storage object over the prior three days. For example Figure 18 shows a cluster node object landing page where summary metrics charts display latency, IOPS, MBps, and utilization. Below each of the metrics charts appear links to any new and obsolete events for the given metrics and object pair. In addition to object-specific information, the landing page provides access to the performance explorer where interactions with other objects can be observed.

Figure 18) Performance Manager node object landing page summary



OnCommand Performance Manager: Performance Explorer

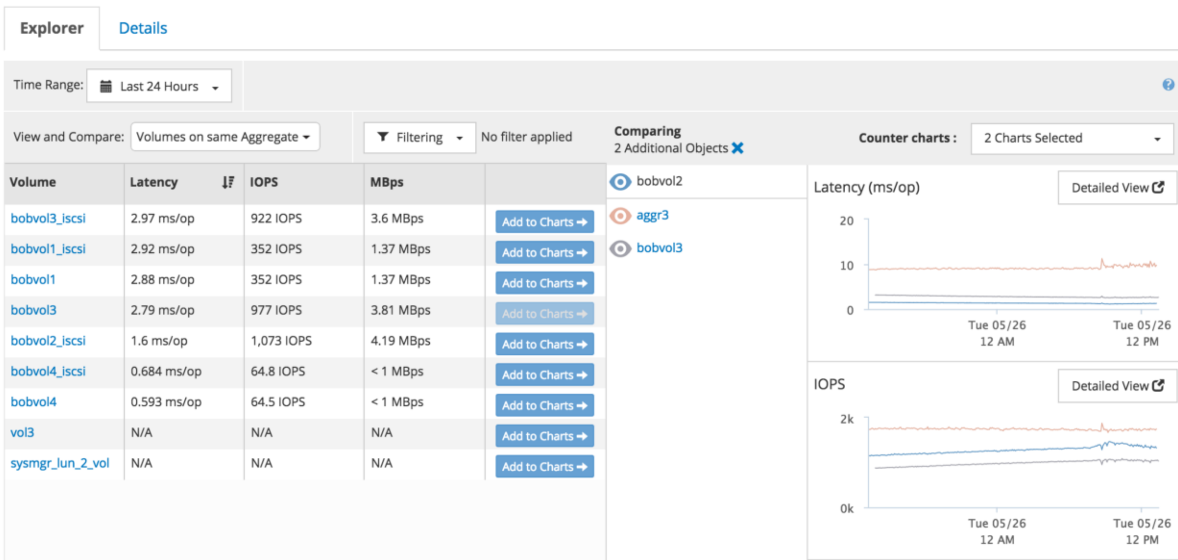
The performance explorer is a modular component of Performance Manager common to all object landing pages. It allows viewing of storage object historical trends over specific time ranges and building views selecting from a menu of metrics comparing related objects.

Performance explorer permits the selection of time ranges over the entire 13 months of retained data. Performance Manager retains metrics for 30 days at 5-minute granularity and 12 months at one-hour granularity. There are several predefined time ranges available such as last 24 hours and last 30 days. However, any time range is permitted over all the retained data at one-hour granularity.

Performance explorer lets users navigate object relationships and select objects for display using metrics charts. One can navigate from the current object to parent container, to sibling, and to child objects. For example, Figure 19 shows performance explorer building charts that compare latency and IOPS over the last 24 hours for both volume and aggregate objects. On the left side of the page it can be seen that additional objects can be easily added to the same charts.

Performance Manager supports nine metrics in total such as IOPS, MBps, latency, cache miss ratio, and utilization. Metrics charts with multiple objects are stacked, effectively overlaying them in time where alert events and performance data can be visually correlated.

Figure 19) Performance explorer comparing volume and aggregate metrics



When more details or higher resolution is needed then a full-page detailed view of any single chart can be launched from performance explorer (see upper left-corner labeled “Detailed View”).

3.4 OnCommand Performance Manager Thresholds and Events

Data centers contain capital equipment shared by numerous business entities. To make sure these resources are consumed in alignment with business priorities, service guarantees are put in place. This fact ultimately leads to the establishment of business policies that are translated to service-level agreements. To assure performance service-level objectives and codify critical business policies, Performance Manager provides threshold policies.

Threshold Policy

A threshold policy template reflects business priorities, storage provisioning, and service-level objectives. For example, a storage system provisioned for a business critical workload has service-level objectives where latency must not exceed 20ms. A storage volume is provisioned for this workload, and Performance Manager is set up to continuously analyze volume performance and alert when the service-level objective is violated.

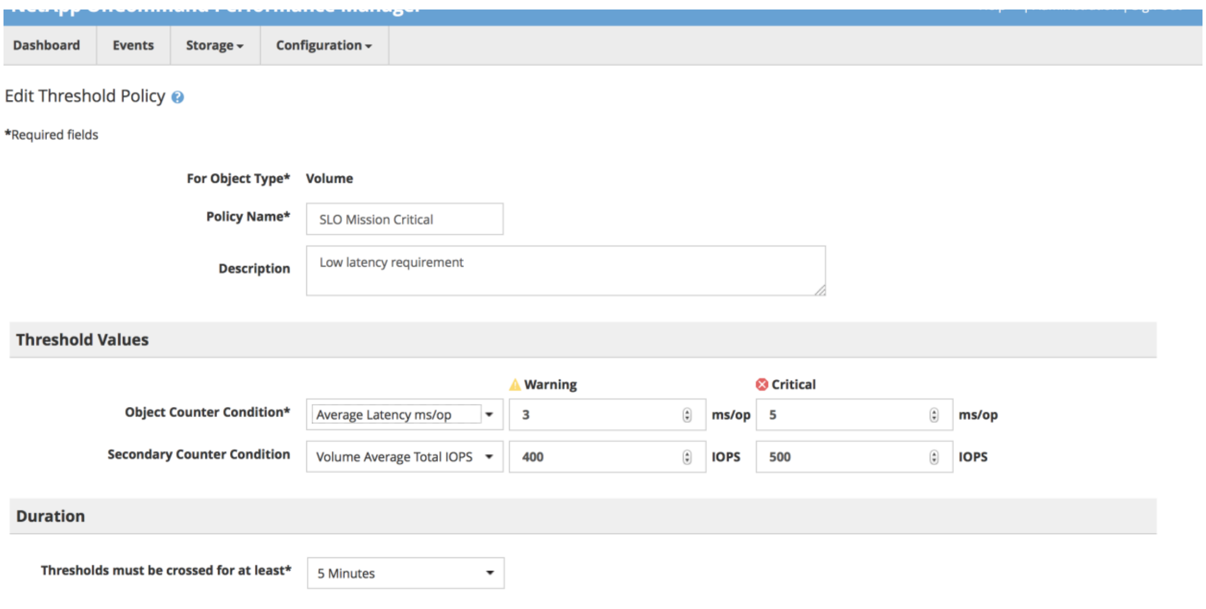
To accomplish this goal, threshold policy templates are created and applied to one or more objects. The following attributes are defined when creating a policy threshold:

- Object type
- Metrics
- Warning threshold

- Critical threshold
- Duration

The object type and metrics define the storage object and metrics pair that is the target of the policy. When creating a volume storage object policy compound metrics are permitted when used in combination with latency metrics. For example, during off hours, when load significantly decreases, latency calculation can skew artificially high and cause false alerts. Thus compound metrics are analyzed where both latency and IOPS must exceed given thresholds (see Figure 20) to generate an event.

Figure 20) Threshold policy creation



The policy definition has both warning and critical threshold levels. When either of these thresholds is breached for the given duration, event alerts are created. The warning threshold level is useful in that it can be used to trigger an early investigation before an issue becomes critical.

A threshold policy is only a template and is not of use until applied to one or more storage objects. Threshold policy templates are applied to objects from the object inventory views (not shown here).

Events

When threshold policies are violated events are generated. Events remain in a new state until the alerting condition subsides and then transition to an obsolete state. Performance Manager always posts events to the dashboard (see **Error! Reference source not found.**) and retains a record in the event inventory view. Events can be set up to appear on the OnCommand Unified Manager dashboard and sent to an e-mail to an administrator.

There are two types of events: system and user-defined. Performance Manager has built-in system event generation where NetApp engineering establishes thresholds. These include thresholds associated with the internal operations of the system such as node busy conditions, file system layout factors, and disk utilization. Take system events seriously and act on them. User-defined events result from violating a user-created threshold policy (discussed previously).

Events appear in chart time lines such that they can be visually correlated with all other metrics. This correlation is helpful in confirming expected correlations or in discovering unexpected correlations between storage object resource consumption and the triggering of an event. In Figure 21 it can be seen that a critical system event is correlated with aggregate utilization (red line) crossing 50% utilization (note that chart key is not shown to conserve space). In the onscreen display, the user can access additional event details by hovering a mouse over the red dot.

When an event is generated, the following attributes are recorded:

- Status: warning or critical
- Type: system or user-defined
- State: new or obsolete
- Duration: how long the alerting condition lasted
- Associated storage object
- Description: why the event was generated.

Figure 21) Event correlation with metric chart

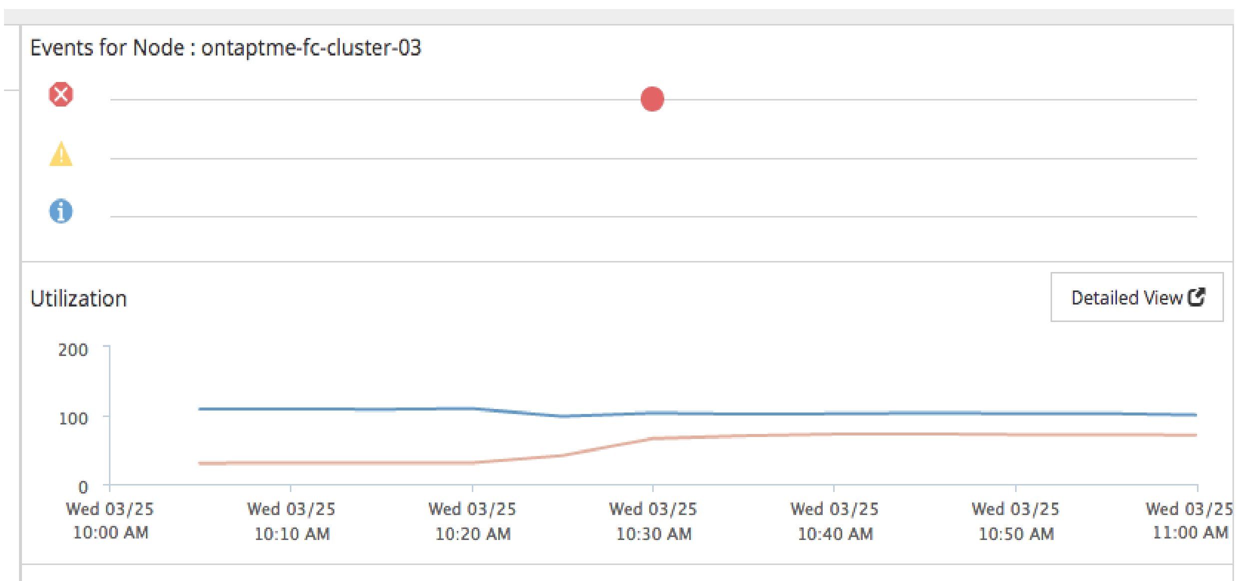


Figure 22 shows details from a user-defined CRITICAL event where volume latency exceeds a 5.00ms/op threshold setting for 3 hours and 10 minutes.

Figure 22) User-defined event details

3.5 OnCommand Performance Manager Key Takeaways

Performance Manager is more than a monitoring and plotting tool. It intimately understands the internal workings of clustered Data ONTAP and helps operating and answering fundamental performance questions about the entire storage environment. This is accomplished through holistic illustrations depicting storage performance on the dashboard and meticulous management of service levels from threshold policies that directly reflect the priorities of the business.

OnCommand Performance Manager is a valuable component of the NetApp OnCommand product portfolio and will be heavily invested in for years to come.

Additional Resources

- OnCommand Performance Manager product page: http://www.netapp.com/us/products/management-software/performance-manager.aspx
- OnCommand Performance Manager documentation http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=61809
- OnCommand community: click the Performance Manager tag to start a discussion or ask a question.: Netapp.com/oncommand_community
- TR-4015: SnapMirror Configuration and Best Practices Guide for Clustered Data ONTAP http://www.netapp.com/us/media/tr-4015.pdf
- TR-4063: Parallel Network File System Configuration and Best Practices for Clustered Data ONTAP 8.2 and Later http://www.netapp.com/us/media/tr-4063.pdf
- TR-4067: Clustered Data ONTAP NFS Implementation Guide http://www.netapp.com/us/media/tr-4067.pdf
- TR-3982: NetApp Clustered Data ONTAP 8.3: An Introduction http://www.netapp.com/us/media/tr-3982.pdf
- TR-4080: Best Practices for Scalable SAN in Clustered Data ONTAP 8.3 http://www.netapp.com/us/media/tr-4080.pdf
- TR-3832: Flash Cache Best Practices Guide http://www.netapp.com/us/media/tr-3832.pdf
- TR-4070: Flash Pool Design and Implementation Guide http://www.netapp.com/us/media/tr-4070.pdf
- TR-3838: Storage Subsystem Configuration Guide http://www.netapp.com/us/media/tr-3838.pdf

Contact Us

Let us know how we can improve this technical report.

Contact us at docfeedback@netapp.com.

Include TECHNICAL REPORT 4211 in the subject line.

Addendum

8.3 Clustered Data ONTAP Upgrade Recommendations

There are no known issues regarding performance when upgrading from 8.2 clustered Data ONTAP to 8.3 at the time of this writing. In all cases the recommendation is to contact your NetApp sales representative and/or utilize the NetApp Upgrade Advisor on the NetApp Support site at:

http://support.netapp.com/NOW/asuphome/

Refer to the [Interoperability Matrix Tool (IMT)]() on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp
www.netapp.com