FlexShare™ Design and Implementation Guide

Akshay Bhargava, Network Appliance, Inc., April 2006 | TR-3459

Executive Summary

FlexShare is a Data ONTAP® feature that provides workload prioritization for a storage system. Using FlexShare, storage administrators can confidently host different applications on a single storage system without impacting critical applications – resulting in reduced costs and simplified storage management. This paper provides details on how FlexShare works, FlexShare best practices, and high benefit use cases of FlexShare.

Table of Contents

1. Overview

As storage requirements for enterprises continue to grow, storage administrators constantly strive to maximize return on investment (ROI) while scaling the existing infrastructure. Administrators are consistently looking for creative ways to prevent overprovisioning and to maximize the use of the existing resources. FlexShare, a built-in feature of Data ONTAP, allows storage administrators to accomplish these tasks with ease and flexibility.

FlexShare gives administrators the ability to leverage existing infrastructure and increase processing utilization without sacrificing the performance of critical business needs. With the use of FlexShare, administrators can confidently consolidate different applications and data sets on a single storage system. FlexShare gives administrators the control to prioritize applications based on how critical they are to the business.

1.1 FlexShare Definition

FlexShare is a Data ONTAP software feature that provides workload prioritization for a storage system. It prioritizes processing resources for key services when the system is under heavy load. FlexShare does *not* provide guarantees on the availability of resources or how long particular operations will take to complete. FlexShare provides a priority mechanism to give preferential treatment to higher priority tasks.

1.2 Benefits

The use of FlexShare in an environment can result in many benefits. Some of the key benefits are highlighted in the table below:

BENEFIT	FLEXSHARE DETAILS
Simplification of storage management	▪ Reduces the number of storage systems that need to be managed by enabling consolidation ▪ Provides a simple mechanism for managing performance of consolidated environments ▪ Easy to administer using the same NetApp CLI and Manage ONTAP™ API
Reduction in costs	▪ Allows increased capacity and processing utilization per storage system without impact to critical applications ▪ No special hardware or software required ▪ No additional license required
Flexibility	▪ Can be easily customized to meet performance requirements of different environment workloads

1.3 Supported Configurations

FlexShare works on NetApp storage systems running Data ONTAP version 7.2 or later.

1.4 Features

FlexShare provides storage systems with the following key features:

- Relative priority of different volumes
- Per-volume user versus system priority
- Per-volume cache policies

These features allow storage administrators to tune how the system should prioritize system resources in the event that the system is overloaded.

Recommendation

The ability to control how system resources will be used under load gives the administrator an exceptional level of control. In order to take advantage of FlexShare features, a storage administrator must take the responsibility to fully understand the impact of different configuration options and optimally configure the storage system.

Before proceeding to configure priority on a storage system, it is essential to understand the different workloads on the storage system, the impact of setting priorities on the storage system, and the FlexShare best practices. Improperly configured priority settings can have undesired effects on application and system performance. The administrator should be well versed in the configuration implications and best practices.

This document is meant to help the storage administrator, providing the fundamental knowledge to configure and tune FlexShare. Understanding the key concepts and following the best practices are essential first steps.

2. FlexShare Design

This section provides an overview of the FlexShare design.

2.1 Basic Concepts

FlexShare provides the ability to assign priorities to different volumes. FlexShare also provides the ability to configure certain *per-volume* attributes, including user versus system priority and cache policies.

WAFL® Operation

A read or write request initiated from any data protocol is translated to individual read or write WAFL operations by the file system. Similarly, a system request is translated into individual WAFL operations.

Data ONTAP classifies each WAFL operation as a user or system operation based on its origin. For example, a client read request is classified as a user operation; a SnapMirror® request is classified as a system operation.

Processing Buckets

FlexShare maintains different processing buckets for each volume that has a configured priority setting. FlexShare populates the processing buckets for each volume with WAFL operations as they are submitted for execution. The processing buckets are only used when the FlexShare service is on; when the FlexShare service is off, all WAFL operations are bypassed from processing buckets and sent directly to WAFL.

Data ONTAP maintains a *Default* processing bucket. When the FlexShare service is on, all WAFL operations associated with volumes that do *not* have a FlexShare priority configuration are populated in the *Default* processing bucket; all WAFL operations for a volume that has a FlexShare priority configuration are populated into a dedicated bucket.

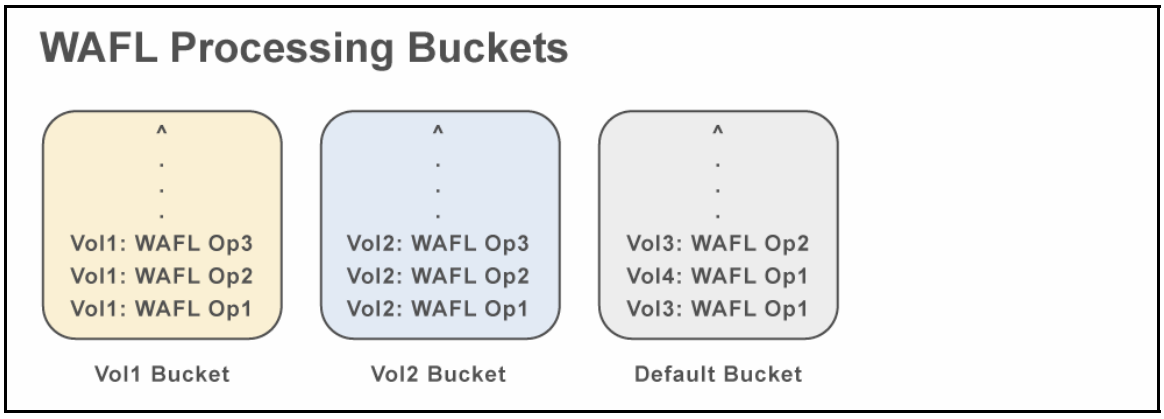

Figure 1) WAFL Processing Buckets

The figure above shows the WAFL processing buckets for Vol1, Vol2, and the Default Bucket. Vol1 and Vol2 have FlexShare priority configurations and as a result have dedicated processing buckets.

User versus System

FlexShare provides a configuration option for user vs. system priority. This allows system-initiated operations to be controlled relative to user-initiated operations. This configuration is available on a per-volume basis.

FlexShare determines whether a WAFL operation is a user or system operation based on the origin of the WAFL operation. If the origin of a WAFL operation is a data access protocol, the operation is considered to be a user operation. All other WAFL operations are considered system operations.

The table below lists some important user and system operations.

USER OPERATIONS	SYSTEM OPERATIONS
Data access operations using: ▪ NFS ▪ CIFS ▪ iSCSI ▪ FCP ▪ HTTP ▪ FTP	▪ SnapMirror ▪ SnapVault® ▪ WAFL Scanners ▪ vol clone/vol split ▪ SnapRestore® ▪ NDMP

Buffer Cache Policies

Data ONTAP uses the cache to store buffers in memory for rapid access. When the cache is full and space is required for a new buffer, Data ONTAP uses a modified least-recently-used (LRU) algorithm to determine which buffers should be discarded from the cache.

FlexShare can modify how the default buffer cache policy behaves by providing hints for the buffers associated with a volume. FlexShare provides hints to Data ONTAP by specifying which information should be kept in the cache and which information should be reused.

FlexShare caching policies, if configured properly based on application workloads, can significantly enhance overall system performance. The buffer cache policy configuration is based on a per-volume setting.

Modes of Operation

The following modes of operation are available with FlexShare.

1. FlexShare service is off.

By default, the FlexShare service is off. The system behavior is identical to previous versions of Data ONTAP when FlexShare was not available.

2. FlexShare service is on; no individual priorities set.

When FlexShare service is enabled, FlexShare provides equal priority to all volumes and equal user versus system priority. FlexShare continues to use the default caching policy.

3. FlexShare service is on; individual volume priorities set.

When FlexShare service is on and one or more individual volume priorities are set, FlexShare begins to prioritize operations between different volumes.

2.2 How FlexShare Schedules WAFL Operations

FlexShare impacts the order in which WAFL operations are processed by the storage system. FlexShare determines the order WAFL operations will be processed based on the priority configuration. FlexShare gives higher priority to WAFL operations originating from higher priority volumes.

When the FlexShare service is on, the prioritization processing described in this section is always in effect.

Volume Level Priorities

The impact of FlexShare volume level priority can best be understood by comparing one storage system with the FlexShare service off with a second storage system with the FlexShare service on.

When the FlexShare service is *off*, the system processes the requests in the order in which they arrive.

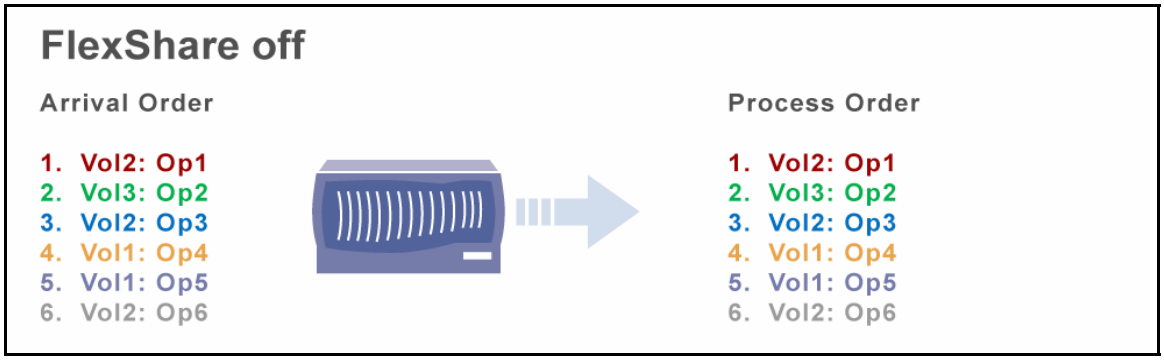

Figure 2) FlexShare off

The figure above depicts the order in which tasks arrive to be processed and the order in which they are processed by a storage system. The order of tasks processed is exactly the same as the order in which tasks arrive.

When FlexShare service is *on*, FlexShare intelligently chooses the order tasks are processed to best meet the priority configuration. On average, FlexShare is more likely to pick a WAFL operation originating from a high priority volume than a WAFL operation originating from a low priority volume. FlexShare ensures that all WAFL operations will be processed regardless of the priority configuration, but FlexShare is more likely to choose higher priority operations to be processed before lower priority operations.

Figure 3 provides a simple example of how FlexShare can impact the order in which tasks are processed based on the priority level configurations.

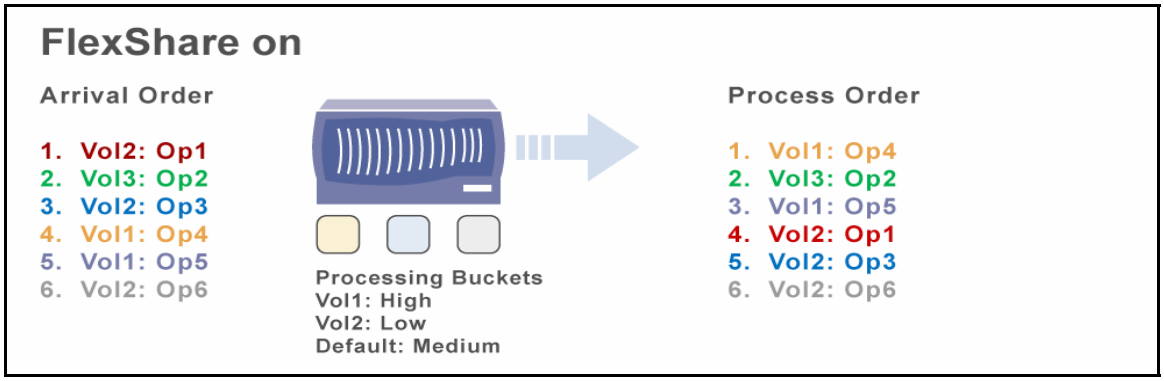

Figure 3) FlexShare on

The figure depicts a *possible* ordering of tasks when FlexShare service is on. The order tasks arrive is different than the order tasks are processed by the storage system. FlexShare orders tasks for processing taking into account the priority configuration. In this example, Vol1 has higher priority configuration than the other volumes and therefore its WAFL operations are preferentially processed.

Volume Level and System Priorities

FlexShare orders WAFL operations to be processed based on the following:

1. The configured volume priority
2. The configured user versus system priority

The order of the steps above is important in determining when WAFL operations are executed. First, the WAFL operations are prioritized based on the volume priorities. The priority of the processing buckets, which contain the WAFL operations, is the first factor that is considered. Second, the WAFL operations are prioritized based on the configured user versus system priority. The items in the individual processing buckets are ordered with respect to the user versus system priority.

Figure 4 and Figure 5 depict an example of how FlexShare chooses WAFL operations to execute based on the priority level and system configurations.

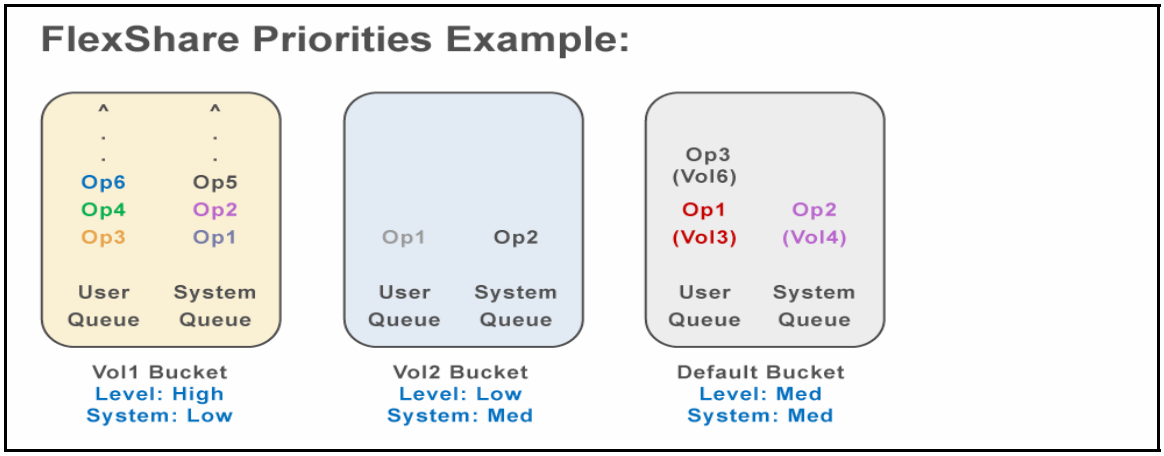

Figure 4) FlexShare Priorities

The figure above depicts what the processing buckets could look like for a storage system as they arrive for processing. Vol1 is configured with a high priority level and low system priority. Vol2 is configured with low priority level and medium system priority. Vol1 and Vol2 are the only volumes that have FlexShare priority configurations and as a result have dedicated processing buckets.

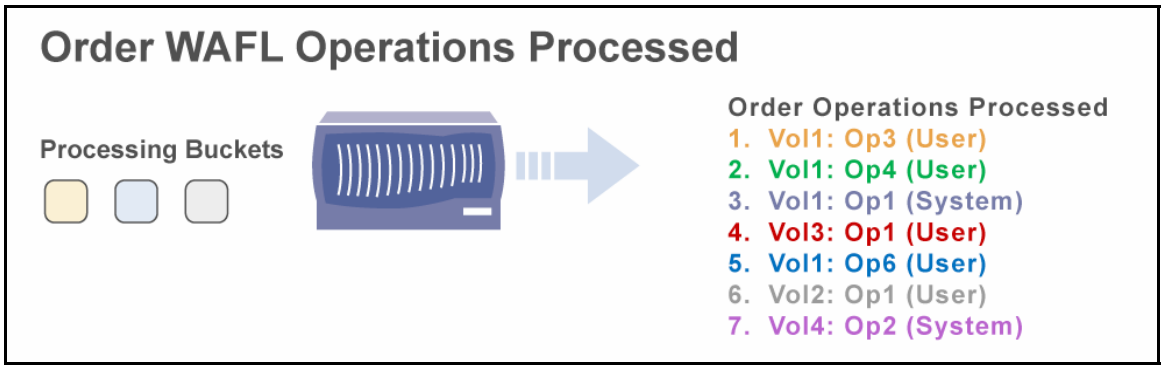

Figure 5) Order WAFL Operations Processed

The figure shows a *possible* order FlexShare would choose to process the WAFL operations from Figure 2. FlexShare orders the operations to be processed given the relative volume priorities and per-volume user versus system priority.

The example in Figure 4 and Figure 5 highlights some important points about the heuristics that FlexShare uses.

- FlexShare provides relative priority based on the volume priority configurations for the different volumes. FlexShare will preferentially choose WAFL operations to be processed from Vol1 before any other volume since Vol1's priority is set the highest.
- FlexShare takes into account the User versus System priority on a per-volume basis. Out of the WAFL operations processed for Vol1, FlexShare will preferentially choose User operations before System operations since the system priority for Vol1 was set to Low. FlexShare can choose User operations from the Vol1 processing bucket even if they were added to the bucket *after* System operations. For example, FlexShare chose to process "Vol1: Op3 (User)" earlier even though this operation was added to the Vol1 Bucket after "Vol1: Op1 (System)".
- FlexShare will choose lower priority operations before higher priority operations, but this happens less frequently.

Impact of WAFL Operation Scheduling

The impact of FlexShare rescheduling WAFL operations is generally only noticeable when the system is under heavy load. If the system is not loaded, the number of outstanding operations is small enough that FlexShare prioritization will not noticeably impact the system performance. To better understand this, imagine there is only one outstanding WAFL operation that needs to be processed out of all the processing buckets. In this case, FlexShare will not have to do any intelligent prioritization. It will simply pick the one outstanding WAFL operation to be processed. The order of processing items can have a significant impact to end users only when the system is loaded and there are sufficient items in the different processing buckets.

FlexShare does not impact the running time of WAFL operations. Once a WAFL operation is dispatched to execute, FlexShare work with the WAFL operation is complete. If there is a WAFL operation that has been dispatched or is already in progress, FlexShare will not interrupt that WAFL operation even if higher priority WAFL operations arrive in the system. FlexShare only controls the order in which WAFL operations are dispatched to be processed, but once they are dispatched they are out of the control of FlexShare.

2.3 How FlexShare Manages System Resources

FlexShare automatically controls how system resources are used by the storage system based on the volume priority level and per-volume buffer cache policy configurations. The storage administrator does not have to configure any other options to take advantage of the system resource management FlexShare provides.

FlexShare has several mechanisms to control how system resources are used. The WAFL operation ordering contributes to how system resources get used, but it is not the only means. FlexShare employs cache management and other intelligent schemes to control the different system resources.

FlexShare prioritizes, but does not guarantee the availability of system resources. FlexShare does not pre-partition or exclusively reserve system resources.

Cache Management

FlexShare provides hints to the Data ONTAP buffer cache manager by specifying which information should be kept in the cache and which information should be reused. FlexShare provides the following important information to the buffer cache manager:

- FlexShare recommends that data items in the cache that originated from a volume with a *reuse* setting be the first to be removed from the cache. Buffers containing user data are proactively aged as soon as the data has been sent to the client.
- FlexShare recommends that data items in the cache with a *keep* setting be preferentially kept in the cache.

The buffer cache manager preferentially keeps the items in the cache marked with a keep setting. However, if the cache is full and all items in the cache have a keep setting, the least-recently-used data item will be removed from the cache. It is important to note that cached data with a keep setting can be removed from the cache if the cache is full and all items in the cache belong to volumes with a keep configuration.

For optimal performance, it is recommended to set the volume cache policies appropriately. Refer to *Section 4: FlexShare Best Practices* for more information.

System Resource Usage

FlexShare prioritization controls the following system resources:

- CPU
- Disk I/O
- NVRAM
- Memory

This section highlights the mechanism by which FlexShare controls the critical system resource usage.

CPU

Higher priority volumes have their WAFL operations preferentially scheduled for CPU processing. This is primarily impacted based on how FlexShare controls the order in which the WAFL operations are chosen to execute. Refer to *Section 2.2: How FlexShare Schedules WAFL Operations* for more details on how FlexShare processes the volume and system priorities.

Disk I/O

Higher priority volumes are allowed more concurrent disk reads than lower priority volumes. FlexShare maintains a maximum number of concurrent disk reads allowed per-volume. The higher the priority of the volume, the higher the maximum number of concurrent disk reads allowed.

The amount of concurrent disk reads is automatically set based on the volume level priority. The amount of concurrent disk reads for a volume can be viewed using the advanced counters described in Section 5.

WAFL uses NVRAM to keep a log of write requests that need to be written to disk. During a Consistency Point (CP), the same set of data is copied from system memory to disk. FlexShare prioritizes disk writes by controlling how NVRAM is used. This is described in the next section.

NVRAM

FlexShare controls the amount of NVRAM consumption based on volume priority. A volume's priority dictates how much NVRAM can be consumed relative to other volumes. This is essential in maintaining priority for writes during a Consistency Point (CP) operation. If a low priority volume has exhausted its amount of writes allocated for NVRAM, it will have to wait until the current CP is completed. High priority volumes have significantly larger NVRAM limits and, therefore, their writes are generally unaffected during a CP.

The amount of NVRAM consumption is automatically set based on the volume level priority. The amount of NVRAM consumption for a volume can be viewed using the advanced counters described in Section 5.

Memory

The memory consumption is dictated by the configured buffer cache policy for the volume. This is described in detail with the description of the cache management.

3. FlexShare Administration

FlexShare can be administered using the CLI or the Manage ONTAP API. This section describes the important configuration and status commands for the CLI, the CLI commands that impact FlexShare configuration, and details about the Manage ONTAP API.

The content in this section provides an overview of the typical commands and options. Comprehensive details on the FlexShare CLI are provided in the *System Administration Guide*. Refer to the *System Administration Guide* or the *na_priority* man page for additional information.

The default values that are assigned when FlexShare is initially enabled are:

- Volume Level: Medium
- System: Medium
- Cache: default

FlexShare configuration can be dynamically changed at any time the system is running. Configuration changes take effect as soon as they are issued on the system. There is no overhead to change configuration options. Configuration changes stay active across system reboots. The default values assigned by FlexShare can be modified as well.

3.1 FlexShare CLI Overview

The *priority* command is the CLI command that provides all configuration and status information related to FlexShare.

Basics

Issue the *priority* command without any arguments to display the priority command options.

```
NetApp1> priority
The following commands are available; for more information
type "priority help <command>"
delete              off                 set                 show
help                on
```

Use the *help* option to find out more information about a command.

```
NetApp1> priority help on
priority on
- Start priority scheduler.
```

Refer to the *na_priority* man page or the *System Administration Guide* for detailed information.

```
NetApp1> man na_priority
na_priority(1)                                    na_priority(1)

NAME
       na_priority - commands for managing priority scheduling.

SYNOPSIS
       priority command argument ...

DESCRIPTION
       The  priority  family  of  commands  manages  the priority
       scheduling policy on an appliance.
.
.
.
```

Enable Service

To see the status of the FlexShare service, use the *show* command.

```
NetApp1> priority show
Priority scheduler is stopped.

Priority scheduler system settings:
        io_concurrency: 8
```

The FlexShare service is off by default.

Note: The io_concurrency setting displayed in the *priority show* output represents the average number of concurrent suspended operations per disk for a volume. This is an advanced option and should not be modified unless recommended by NetApp personnel.

To enable FlexShare service, use the *on* command.

```
NetApp1> priority on
Priority scheduler starting.
NetApp1> Fri Mar 17 22:07:11 GMT [wafl.priority.enable:info]: Priority scheduling is
being enabled
```

To verify the FlexShare service is enabled, use the *show* command.

```
NetApp1> priority show
Priority scheduler is running.

Priority scheduler system settings:
        io_concurrency: 8
```

To disable FlexShare service, use the *off* command.

```
NetApp1> priority off
Priority scheduler has stopped.
NetApp1> Fri Mar 17 22:52:28 GMT [wafl.priority.disable:info]: Priority scheduling
is being disabled
```

Priority Settings

The *set* command is used to configure volume priorities. Configuration for *level*, *system*, and *cache* can be specified. At least one configuration option from *level*, *system*, or *cache* must be specified. Options that are not explicitly set inherit the default setting.

The *level* option is configured on a per-volume basis. A volume with a higher priority level will be given more resources than a volume with a lower priority level.

The *system* option is configured on a per-volume basis. It controls the balance of system versus user priority given to a volume.

Valid *level* and *system* options include:

- VeryHigh
- High
- Medium
- Low
- VeryLow

The *system* option can also take a number as a numeric percentage from 1 to 100 for the system priority.

The *cache* option is configured on a per-volume basis. It controls the buffer cache policy for the volume. Valid cache options include:

- reuse
- keep
- default

The example below sets the volume level priority to High. The volume will inherit the default settings for *system* and *cache*.

```
NetApp1> priority set volume vol1 level=High
NetApp1> priority show volume -v vol1
Volume: vol1
        Enabled: on
          Level: High
         System: Medium
          Cache: n/a
```

Note: The *Cache: n/a* output represents the default cache configuration.

The example below explicitly sets the level, system, and cache configuration.

```
NetApp1> priority set volume vol2 level=Low system=Low cache=reuse
NetApp1> priority show volume -v vol2
Volume: vol2
        Enabled: on
          Level: Low
         System: Low
          Cache: reuse
```

FlexShare maintains a default configuration that applies to the *Default* processing bucket. All the volumes with priority configurations inherit the default settings unless explicitly configured.

```
NetApp1> priority show default -v
Default:
          Level: Medium
         System: Medium
```

The default configuration can be modified, if desired. Default values for *level*, *system*, *nvlog_limit*, *system_read_limit*, and *user_read_limit* can be modified.

```
NetApp1> priority set default system=Low
NetApp1> priority show default -v
Default:
          Level: Medium
         System: Low
```

Priority configuration can be deleted, if desired.

```
NetApp1> priority delete volume vol2
NetApp1> priority show volume vol2
Unable to find priority scheduling information for 'vol2'
```

3.2 Expected Behavior with other CLI commands

This section provides common CLI commands and how they do or do not impact FlexShare priority configuration.

CLI COMMAND	PRIORITY CONFIGURATION OUTCOME
vol rename	Priority configuration is unchanged.
vol copy	The destination volume will be assigned the *default* priority configuration. The source volume's priority configuration will be unchanged.
vol clone	The cloned volume will be assigned the *default* priority configuration. The source volume's priority configuration will be unchanged.
vol online/vol offline	Priority configuration is unchanged. A volume online or offline will automatically trigger FlexShare to re-balance system resource limits based on the current online volumes in the aggregate.
vol destroy	Priority configuration for the volume is permanently removed.

3.3 Manage ONTAP API

FlexShare configuration and status can be administered using the Manage ONTAP API. The complete functionality to configure and retrieve status is available from the Manage ONTAP API.

Refer to the Manage ONTAP SDK API Reference for documentation on the individual APIs. The Manage ONTAP Developer Portal is available from the NOW™ site and contains links to the SDK.

3.4 Upgrade and Revert

FlexShare configuration is safely stored in the Data ONTAP registry. An upgrade preserves the FlexShare priority configuration and the configuration becomes active automatically. If the Data ONTAP version is reverted to a previous version that does not support FlexShare, the FlexShare configuration will be ignored without any impact.

4. FlexShare Best Practices

Following the best practices outlined in this section will help ensure that the FlexShare configuration meets the highest level of performance and robustness.

4.1 Set Priority Configuration for All Volumes in an Aggregate

While volumes in an aggregate can have different priority configurations, it is important to explicitly set priority configuration for *all* volumes in an aggregate. If any volume in an aggregate requires a priority configuration, it is recommended to explicitly set priority configuration for *all* volumes in the aggregate.

Setting individual priorities is required because the performance of the storage system is more balanced if all volumes have priority configuration. This best practice is based on what happens when some volumes in an aggregate have priorities and others do not. Volumes that do *not* have a priority configuration are treated with the default priority. The default priority processes WAFL operations from a common default processing bucket. As a result, all the tasks from the default priority volumes are processed by the same bucket. This can result in undesired performance constraints on the default priority volumes.

For example, consider a large aggregate that has 100 volumes. The large aggregate enables the 100 volumes to utilize all the available disk resources in the aggregate. One volume in the aggregate has a High priority configured; the other 99 volumes do not have an explicit priority configuration. FlexShare creates an independent processing bucket to prioritize operations for the High priority volume. The 99 remaining volumes are serviced by the default bucket. With this configuration, the 99 volumes can easily be strained for resource time.

Now consider modifying the example to meet the best practice. The configuration would consist of 99 volumes with a Medium priority and one volume with a High priority. In this case, FlexShare would create a dedicated processing bucket for all the 100 volumes – one with High priority and 99 with Medium priority. This will result in better load distribution across all the volumes.

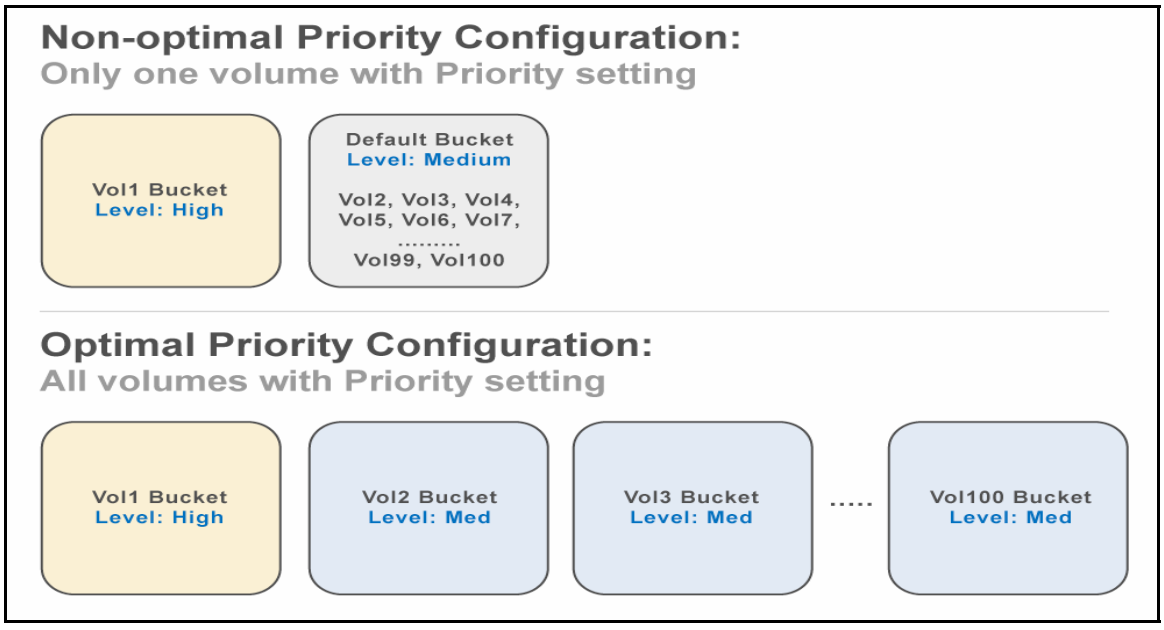

Figure 6) Priority Configuration with Processing Buckets

The figure above shows the difference in the FlexShare processing buckets between a non-optimal priority configuration and an optimal priority configuration, following the best practice. The aggregate has 100 volumes, labeled vol1 to vol100. vol1 has a High priority configuration.

In the non-optimal case, the Default bucket processes WAFL operations for vol2 through vol100. In the best practice configuration, each volume in the aggregate has its own processing bucket.

Review *Section 2.2: How FlexShare Schedules WAFL Operations* for an overview on the processing buckets and how operations are prioritized amongst them.

4.2 Configure Cluster Configuration Consistently

There are some important precautions that should be taken into account in a clustered deployment:

1. Both nodes of a cluster must have the same global priority on or off setting.
2. The priority configuration of the individual nodes in a cluster should be configured to meet the desired behavior in case a cluster failover occurs.

Priority Setting

Set the service on or off identically on both nodes. Verify the configuration using the *priority show* command.

Cluster Failover

In the event of a cluster failover, the priority schedules are merged. The priority configuration from the failed cluster node is inherited by the healthy cluster node after the cluster failover.

The priority configurations should take into account that the priorities will be merged in the event of a cluster failover. In planning the priority configuration, the administrator should consider -- if all the volumes from both storage systems were hosted on a single storage system, how should their relative priority be? The best approach is to compile a complete list of volumes in the cluster, prioritize amongst them, and then set the priority configuration.

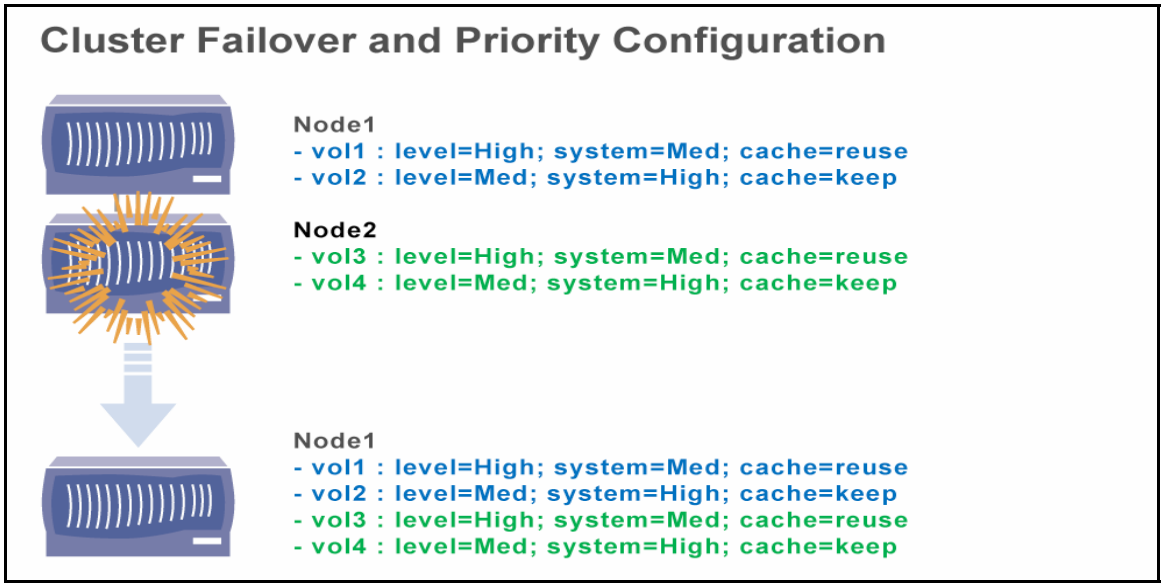

Figure 7) Cluster Failover and Priority Configuration

The figure above depicts how the priority configurations get merged in the event of a cluster failover. Prior to cluster failover, each Node has its own independent priority configuration for its volumes. After Node2 fails, Node1 acquires the priority configuration from the original Node1, merging Node1 and Node2's priority configuration. After a fail back, the priority configuration will be exactly like it was before the cluster failover.

4.3 Set Volume Cache Usage Appropriately

A properly configured buffer cache policy can improve the cache hit rate, significantly improving overall system performance. Here are some guidelines to make sure the buffer cache policy is optimally configured for an environment:

1. Select the right workloads for *keep* versus *reuse.*
2. Don't over allocate the number of keep volumes.

Keep versus Reuse

Configure data sets that will benefit from caching with a *keep* policy. Data sets that make good candidates for a *keep* buffer cache policy typically have active read or write workloads to a small working set relative to the storage system's buffer cache. A database that is frequently accessed with queries involving the same tables could make a good candidate for a keep policy.

It is equally important to identify and properly configure data sets that are not going to benefit from caching with a *reuse* policy. This will allow space in the cache to be optimally used for the data sets that will benefit from caching. Data sets that are read once or infrequently should use the *reuse* policy. For example, a volume with database logs is generally sequentially written, but infrequently read. Therefore, caching database logs generally does not improve the cache hit rate. As a result, database log volumes often make good candidates for a *reuse* policy.

Don't Over Allocate Number of Keep Volumes

The benefit of having the *keep* buffer cache policy is that critical data can achieve a high percent of cache hits and also minimize the amount of data that is swapped in and out of the cache. In order to minimize the amount of data that is swapped in and out, it is important that the active data sets that are configured with a *keep* policy be smaller than the available cache size. If the active working data sets with keep setting are larger than the available cache, all of the data cannot fit in the cache. As a result, the benefit of the *keep* policy would be diminished.

4.4 Tuning For SnapMirror and Backup Operations

SnapMirror and backup operations including NDMP are system operations that should be prioritized by configuring the *system* level for a volume.

The per-volume *system* setting impacts all system activities including SnapMirror and backup operations. FlexShare treats all SnapMirror and backup operations pertaining to a volume as a group, not as individual entities. For example, if a volume has many QSM relationships, the group of QSM relationships is prioritized by FlexShare, not each individual QSM relationship. In other words, FlexShare does not prioritize individual SnapMirror transfers; all SnapMirror transfers for a volume are prioritized together.

In some cases, storage administrators may want to control the SnapMirror or backup operations priority for an entire storage system in a generic way, without having to configure individual volume system priorities. In other cases, individual volumes will have varying requirements and will demand that the system priority be set individually for particular volumes.

Understand Expected Behavior

Storage administrators will be interested in prioritizing user activity compared to system activity. Some will want to give higher priority to user activity while minimizing SnapMirror and backup operation impact. This can be accomplished by setting the *system* priority to be lower. Keep in mind that when the system priority is reduced, the amount of time that SnapMirror transfers or other backup operations generally take can increase. For example, if there is a lot of user activity and the system priority is low, then the user activity will be prioritized for processing above the system activity. As a result, the system activity will take longer to complete.

If you have strict timelines for particular SnapMirror and backup operations to complete, you will want to tune the system priorities with caution. It is advised to closely monitor and tune the priority configuration to meet the desired behavior. See *Section 5: Understanding FlexShare Behavior and Troubleshooting* for more information.

Configuring SnapMirror or Backup Operation Priority across a Storage System

Storage administrators can set global priority for SnapMirror or backup operations across an entire storage system. For example, a storage administrator may want to generally give higher priority to user activity compared to system activity. To accomplish this, it is recommended to configure the default system value to meet the desired behavior. The default system value applies to the default bucket. All volumes that have priority configuration will need to be explicitly configured to meet the user versus system configuration. Note that priority configuration options that are not explicitly configured inherit the original default settings (level: Medium, system: Medium, cache: default). *Refer to Section 3: FlexShare Administration* for details.

Review the *Section 6: FlexShare High Benefit Use Cases* for examples of Backup/Disaster Recovery Throttling.

Configuring SnapMirror or Backup Operation Priority for a Particular Volume

For volumes that have different requirements from the global system priority configuration, storage administrators will want to manually change the configuration on a per-volume basis. Many environments will want to take advantage of this level of control FlexShare provides. By individually configuring system priorities, storage administrators can give SnapMirror or backup operations different levels of priority.

For example, imagine two volumes have the same priority level, but have very different requirements for backup. User access to VolA is critical at all hours of the day, but user access to VolB is critical only during peak hours. The data in VolA is copied with SnapMirror hourly; the data in VolB is copied with SnapMirror nightly during off-peak hours. A delay in the amount of time it takes the backup to complete for VolA is a tradeoff that has been considered and is acceptable. The backup window for VolB happens during the off-peak hours and it is essential the backup finishes before critical users come online. For this situation, an administrator may choose to have different system policies for VolA and VolB. The diagram below depicts the scenario.

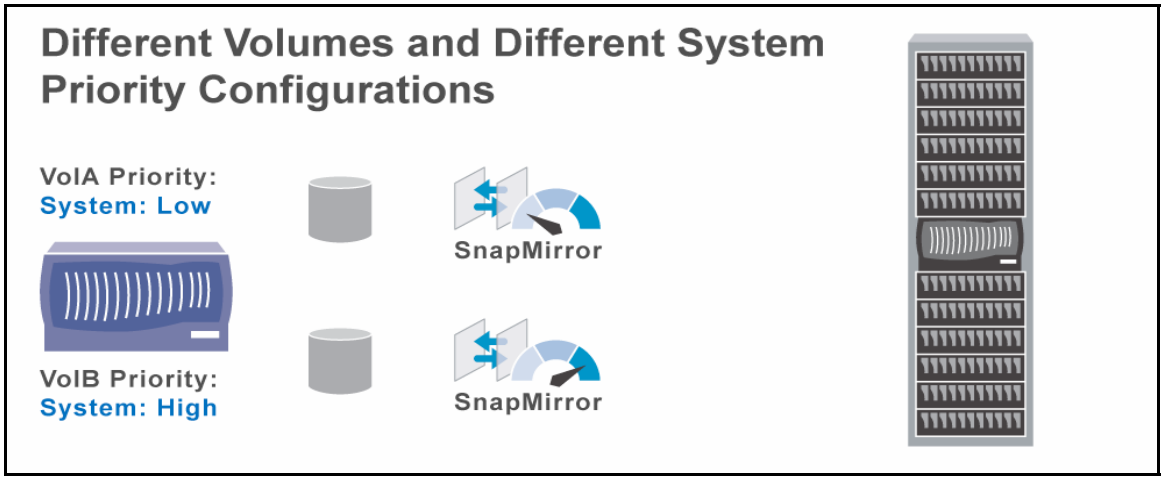

Figure 8) Different Volumes and Different System Priority Configurations

The figure above shows an example where it makes sense to have different system priority configuration for different volumes. It is essential for VolB SnapMirror operations to finish in a timely manner; therefore the system priority is higher. VolA SnapMirror operations can take place at a slower rate since user activity is higher priority.

5. Understanding FlexShare Behavior and Troubleshooting

The previous sections provide information on how to plan and configure FlexShare: how FlexShare works, how to administer FlexShare, and best practices to follow. This section focuses on aspects after FlexShare has been configured:

- Understanding how to analyze FlexShare behavior
- Troubleshooting FlexShare
- Maintaining optimal FlexShare configuration

5.1 FlexShare Counters

FlexShare has a number of advanced diagnostic counters that are useful in understanding FlexShare behavior and troubleshooting. These counters provide valuable insight into how FlexShare is operating. These counters are advanced and only available in the advanced mode.

Terminology

The following terminology is frequently used in the counters:

- Pending: Waiting to run in FlexShare
- Scheduled: Dispatched by FlexShare to WAFL
- Queued: Waiting to be scheduled; received by FlexShare and intentionally being queued

Commands

The FlexShare counters are only available in the advanced mode.

The following commands can be issued to retrieve the counters described in this section:

- *stats show prisched*
- *stats show priorityqueue*

Counters from *stats show prisched* are referred to as *prisched* object counters; counters from *stats show priorityqueue* are referred to as *priorityqueue* object counters.

The *prisched* object counters provide information on the total number of operations queued by FlexShare. The *priorityqueue* object counters provide detailed information on each individual processing bucket, or priority queue, including its configuration and performance statistics.

Below is a sample output of the FlexShare counters:

```
NetApp1*> stats show prisched
prisched:prisched:queued:0
prisched:prisched:queued_max:5

NetApp1*> stats show priorityqueue
priorityqueue:vol1:weight:76
priorityqueue:vol1:usr_weight:78
priorityqueue:vol1:usr_sched_total:0/s
priorityqueue:vol1:usr_pending:0
priorityqueue:vol1:avg_usr_pending_ms:0ms
priorityqueue:vol1:usr_queued_total:0/s
priorityqueue:vol1:sys_sched_total:5/s
priorityqueue:vol1:sys_pending:0
priorityqueue:vol1:avg_sys_pending_ms:0.00ms
priorityqueue:vol1:sys_queued_total:5/s
priorityqueue:vol1:usr_read_limit:14
priorityqueue:vol1:max_user_reads:0
priorityqueue:vol1:sys_read_limit:4
```

```
priorityqueue:vol1:max_sys_reads:3
priorityqueue:vol1:usr_read_limit_hit:0
priorityqueue:vol1:sys_read_limit_hit:0
priorityqueue:vol1:nvlog_limit:33470630
priorityqueue:vol1:nvlog_used_max:0
priorityqueue:vol1:nvlog_limit_full:0
priorityqueue:(default):weight:50
priorityqueue:(default):usr_weight:22
priorityqueue:(default):usr_sched_total:0/s
priorityqueue:(default):usr_pending:0
priorityqueue:(default):avg_usr_pending_ms:0ms
priorityqueue:(default):usr_queued_total:0/s
priorityqueue:(default):sys_sched_total:5/s
priorityqueue:(default):sys_pending:0
priorityqueue:(default):avg_sys_pending_ms:0.00ms
priorityqueue:(default):sys_queued_total:5/s
priorityqueue:(default):usr_read_limit:7
priorityqueue:(default):max_user_reads:0
priorityqueue:(default):sys_read_limit:24
priorityqueue:(default):max_sys_reads:6
priorityqueue:(default):usr_read_limit_hit:0
priorityqueue:(default):sys_read_limit_hit:0
priorityqueue:(default):nvlog_limit::22020151
priorityqueue:(default):nvlog_used_max:60848
priorityqueue:(default):nvlog_limit_full:0
```

Counters Explained

The counters can be broken into two categories:

- Configuration: These counters provide information on internal configuration including how FlexShare translates user configured priority settings and limits on system resources.
- Performance: These counters provide information on how the system is performing.

The *priorityqueue* object refers to an instance name, which is the priority queue name. The priority queue name is either the volume name or '(default)' for the default priority queue.

Configuration

OBJECT	COUNTER	DESCRIPTION
priorityqueue	weight	This is the relative weight of this queue compared to other queues. Value can be in the range of 0 to 100.
priorityqueue	usr_weight	This is the relative weight of user operations compared to system operations. Values can be in the range of 0 to 100.
priorityqueue	nvlog_limit	This is the maximum amount of NVLOG, measured in bytes, the queue can use during a CP. (See also *nvlog_used_max.*)
priorityqueue	usr_read_limit	This is the maximum number of concurrent user-reads allowed. (See also *max_user_reads.*)
priorityqueue	sys_read_limit	This is the maximum number of concurrent system-reads allowed. (See also *max_sys_reads.*)

Performance

OBJECT	COUNTER	DESCRIPTION
prisched	queued	The number of operations currently queued in FlexShare waiting to be scheduled.
prisched	queued_max	The maximum number of operations queued in FlexShare at the same time.
priorityqueue	nvlog_used_max	The maximum amount of NVLOG the queue has used during a CP. (See also *nvlog_limit.*)
priorityqueue	max_user_reads	The maximum number of user reads that have been outstanding on the queue since FlexShare was enabled or the queue was created. (See also *usr_read_limit*.)
priorityqueue	max_sys_reads	The maximum number of system reads that have been outstanding on the queue since FlexShare was enabled or the queue was created. (See also *sys_read_limit*.)
priorityqueue	usr_sched_total	The total number of scheduled user operations per second.
priorityqueue	usr_queued_total	The total number of queued user operations per second.
priorityqueue	avg_usr_pending_ms	The average pending time for user operations in milliseconds.
priorityqueue	usr_pending	The current number of pending user operations.
priorityqueue	sys_sched_total	The total number of scheduled system operations per second.
priorityqueue	sys_queued_total	The total number of queued system operations per second.
priorityqueue	avg_sys_pending_ms	The average pending time for system operations in milliseconds.
priorityqueue	sys_pending	The current number of pending system operations.

5.2 Troubleshooting

The motivation of most, if not all, FlexShare troubleshooting is to validate that the FlexShare configuration is impacting the appropriate tasks and with the appropriate level of priority. There are a number of tips that can assist in any troubleshooting effort.

When FlexShare Impact Is Expected

FlexShare is designed to change performance characteristics when the storage system is under load. If the storage system is not under load, it is expected that the FlexShare impact will be minimal and can even be unnoticeable.

Knowing how FlexShare works and assessing the expected behavior are important first steps in any FlexShare diagnosis. Make sure to understand the key concepts highlighted in *Section 2: FlexShare Design*.

Leverage the Diagnostic Counters

The diagnostic counters provided in *Section 5.1: FlexShare Counters* give the most in-depth details into how FlexShare is internally configured and how FlexShare is performing.

Review the counters and look for the following cases:

- General System Performance: Check *usr_sched_total* and *sys_sched_total* for each queue to see how many operations are being dispatched by FlexShare to WAFL per second. The sum of *usr_sched_total* and *sys_sched_total* for each queue provides the total number of operations being scheduled per second for the queue. Reviewing this information will give a general overview of how many operations are executing relative to each queue.
- General System Performance: Review the *avg_usr_pending_ms* and *avg_sys_pending_ms* counters. Higher priority volumes will typically have values of zero or close to zero for *avg_usr_pending_ms* and *avg_sys_pending_ms* counters. Lower priority volumes can expect to have higher *avg_usr_pending_ms* and *avg_sys_pending_ms*, especially when the system is under load.
- Read performance troubleshooting: Compare *max_user_reads* with *usr_read_limit* and compare *max_sys_reads* with *sys_read_limit* to see if the storage system is frequently running into an I/O limitation. Volumes with lower priority are more likely to reach the respective thresholds for read and write operations. A storage system can have no volumes encountering a read or write threshold if FlexShare determines that the current system performance does not require restrictions on I/O performance.
- Write performance troubleshooting: Compare *nvlog_used_max* with *nvlog_limit* to see if the NVLOG throttling is impacting writes. If FlexShare is restricting write performance of a particular queue, the *nvlog_used_max* will be greater than or equal to the *nvlog_limit* for the respective queue.
- User versus System Troubleshooting: Check *avg_usr_pending_ms* and *avg_sys_pending_ms* to see how FlexShare is preferentially processing user versus system operations for an individual queue. Volumes with higher system priority can expect the *avg_sys_pending_ms* will be smaller than *avg_usr_pending_ms.* Volumes with lower system priority can expect that the *avg_usr_pending_ms* will be smaller than *avg_sys_pending_ms.* This behavior will be more noticeable when the queue has many simultaneous operations arriving in the system.

FlexShare off versus on

In some troubleshooting scenarios, it may be a useful option to observe the difference when FlexShare is off versus when FlexShare is on. The administrator will need to assess if this is a viable troubleshooting option for the environment.

Follow the steps outlined below to isolate potential problems on a storage system:

1. Review the system performance characteristics when the FlexShare service is turned off.
    a. Outline the FlexShare configuration that will yield the desired system priority configuration.
    b. Outline the expected performance changes.
2. Turn the FlexShare service on.
    a. Verify FlexShare configuration matches the designed configuration from Step 1a.
3. Review the system performance characteristics when the FlexShare service is turned on.
    a. Review the diagnostic counters, paying particular attention to the difference in volumes that have priority configurations.
    b. Identify any performance changes in the system performance.
    c. Verify the performance changes meet the expectations from Step 1b.

If there are no performance bottlenecks on the storage system in Step 1, it is unlikely that any major changes will occur when the FlexShare service is enabled.

5.3 Maintaining Priority Configurations

Maintaining a storage system to perform at its optimal performance level is an ongoing task. To meet the existing priority requirements can sometimes take a few iterations to optimize for an environment. In addition, as existing priority requirements change due to new application deployments or data consolidations, a storage administrator will need to appropriately tune the FlexShare priority configurations.

Adhering to a systematic methodology for tuning priority configurations will result in the fewest misconfigurations. Follow the steps outlined below for tuning the FlexShare priority configuration:

1. Review the current system performance characteristics.
    a. Review the existing FlexShare priority configuration.
    b. Review the diagnostic counters.
    c. Outline the FlexShare configuration changes that are required to yield the desired system priority configuration.
    d. Outline the expected performance changes.
2. Make FlexShare configuration changes.
3. Review the effect of the FlexShare configuration changes.
    a. Review the FlexShare priority configuration to make sure it matches the outlined plan from Step 1c.
    b. Review the diagnostic counters, paying particular attention to the differences from Step 1b.
4. Assess if priority performance meets desired goals.
    a. If yes: Plan to re-evaluate priority configuration tuning at recurring times in the future and at major changes including new application deployments and data consolidations.
    b. If no: Review desired goals to make sure they are realistic. Go back to Step 1.

6. FlexShare High Benefit Use Cases

Some common high benefit use cases are described in this section:

- Consolidated Environment
- Mixed Storage including FC and ATA
- Backup/Disaster Recovery Throttling
- Multiple Application Instances

A single storage system can take advantage of more than one of the use cases described. For example, a storage system can incorporate several use cases provided in this section – FlexShare can be used to prioritize applications in the consolidated environment, prioritize mixed storage between applications on FC and ATA disk drives, and prioritize backup/disaster recovery all on the same storage system.

FlexShare use cases can give insights to storage administrators on the ways in which FlexShare can be used in their environment to reach optimum usage of processing power.

6.1 Consolidated Environment

FlexShare enables storage administrators to consolidate different applications and data sets on a single storage system without impacting critical applications.

Examples of data sets that can be consolidated include:

- Mail
- Database
- Home Directories

A high benefit use case is to consolidate an application, such as database or mail, and home directories on the same storage system. The application can be set to a higher priority than the home directories. This prioritization protects the application workload to be preferentially treated in case the system is overloaded.

FlexShare can also be used to prioritize among multiple instances of an application. For example, in an environment that has many database instances, FlexShare can be used to prioritize among the different instances. This use case is described in *Section 6.4: Multiple Application Instances*.

The priority configuration must be set for all the volumes of the application as well as all the volumes for the home directories. Refer to *Section 4: FlexShare Best Practices* for more information.

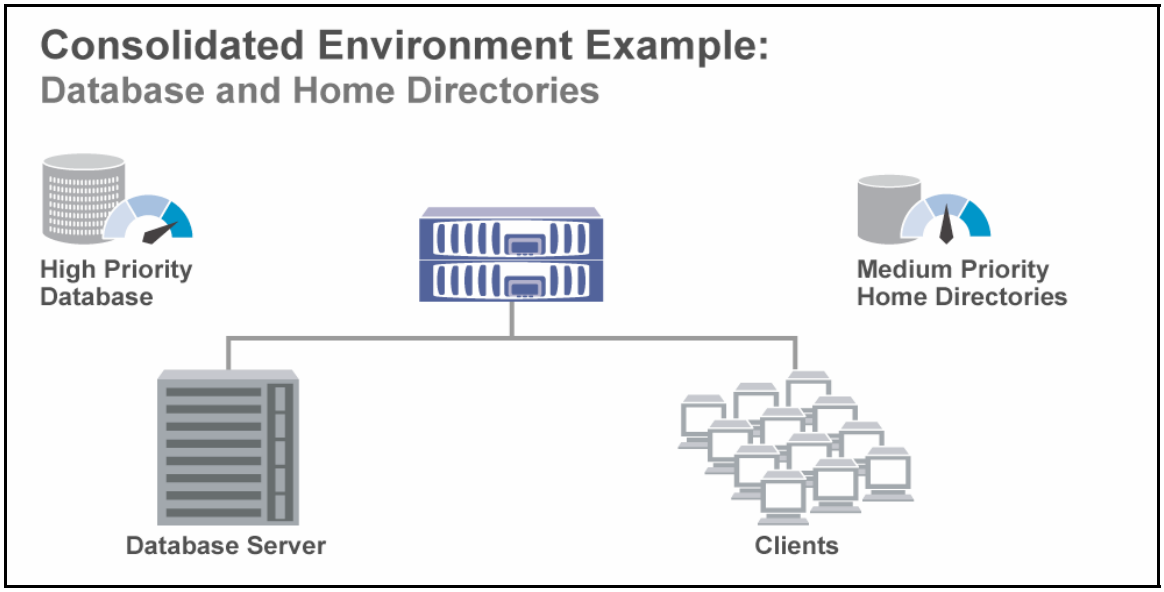

Figure 9) Consolidated Environment Example: Database and Home Directories

The example highlights database and home directory data residing on the same storage system. The database is given higher priority.

6.2 Mixed Storage including FC and ATA

FlexShare enables storage administrators to prioritize data access in a mixed storage environment that includes FC and ATA disk drives so that high-end storage is utilized to its full extent. Storage administrators can choose to prioritize volumes on FC disks over volumes on ATA disks.

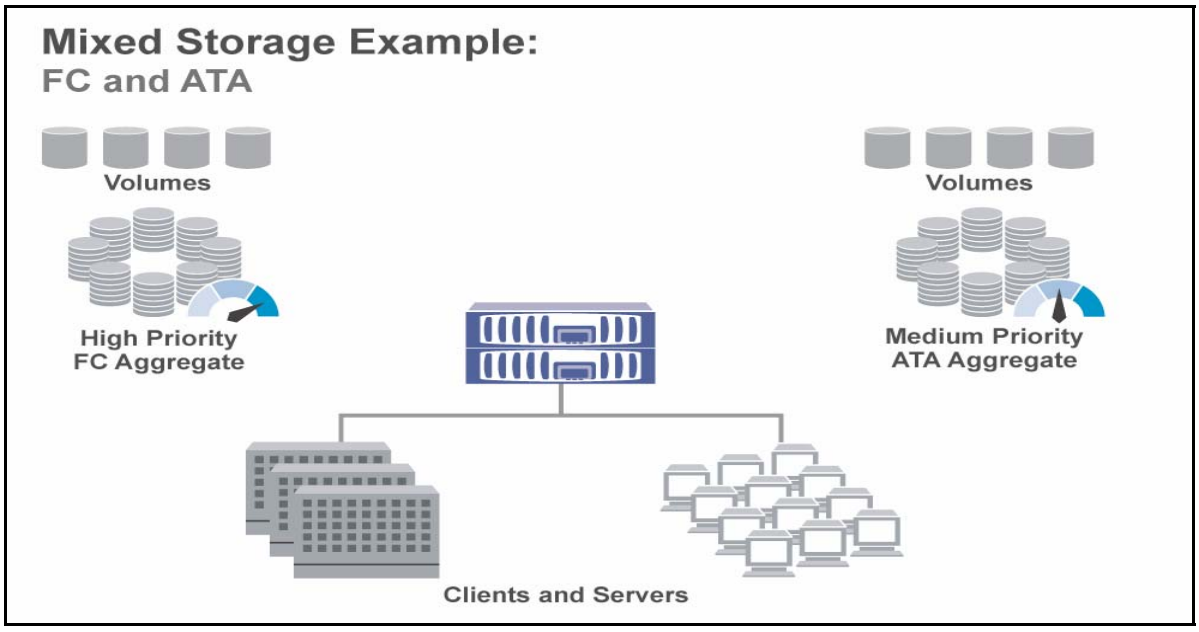

Figure 10) Mixed Storage Example: FC and ATA

6.3 Backup/Disaster Recovery Throttling

FlexShare enables storage administrators to control the priority of backup applications and disaster recovery. This control gives administrators control on how to prioritize backup and disaster recovery operations relative to user initiated tasks.

There are benefits and tradeoffs to modifying the system priority. Lowering the system priority has the benefit of providing higher priority to user access but has the downside that system operations such as backup and SnapMirror transfers can take longer. The environment must be evaluated to see if the benefit of prioritized user access is worth the degraded backup transfer speeds. The degree to which it is acceptable will dictate the configuration.

Some possible use cases are described in the table below.

USE CASE	STORAGE SYSTEM	CONFIGURATION
1. Prioritize user activity higher than backup and system operations	Primary Storage	Set the system priority of the application volumes to be Low or VeryLow depending on the desired behavior.
2. Prioritize user activity higher than system activity during peak hours; prioritize backup and system activity during off-peak hours	Primary Storage	Maintain a different system priority for peak and off-peak hours. This provides optimal performance to user access during peak hours and optimal backup completion time during off-peak hours for system operations. Set the system priority of the application volume to High or VeryHigh for the duration of the backup operations. Set the system priority to the desired level for peak hours.
3. Dedicated storage system for backup	NearStore® Storage	Set the system priority to High or VeryHigh on the NearStore storage system.

Use case #1 is practical for data sets that are constantly backed up throughout the day and have margin for longer backup windows. This use case has the tradeoff that backup operations can take longer. Figure 11 depicts this use case in an environment.

Use case #2 is practical in environments that have backup operations primarily during off-peak hours and user access is not critical during this time. Figure 8 depicts this use case in an environment.

Use case #3 is practical for any dedicated storage system including NearStore. Dedicated storage systems will want to always give highest priority to backup operations. Figure 12 depicts this use case in an environment.

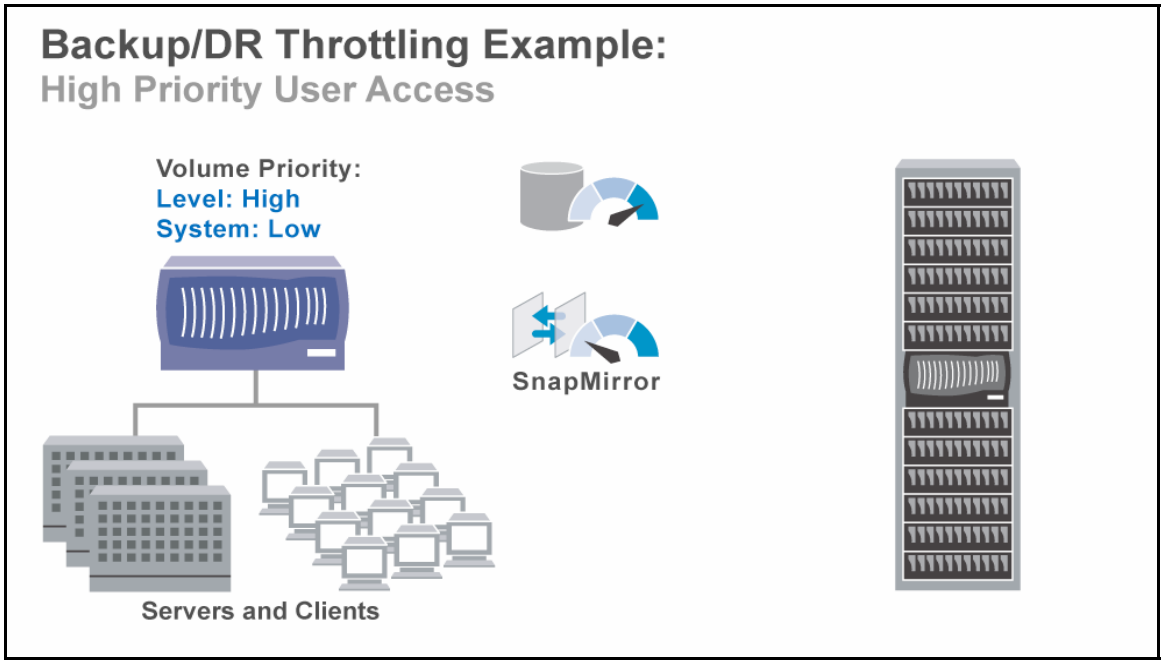

Figure 11) Backup/DR Throttling Example: High Priority User Access

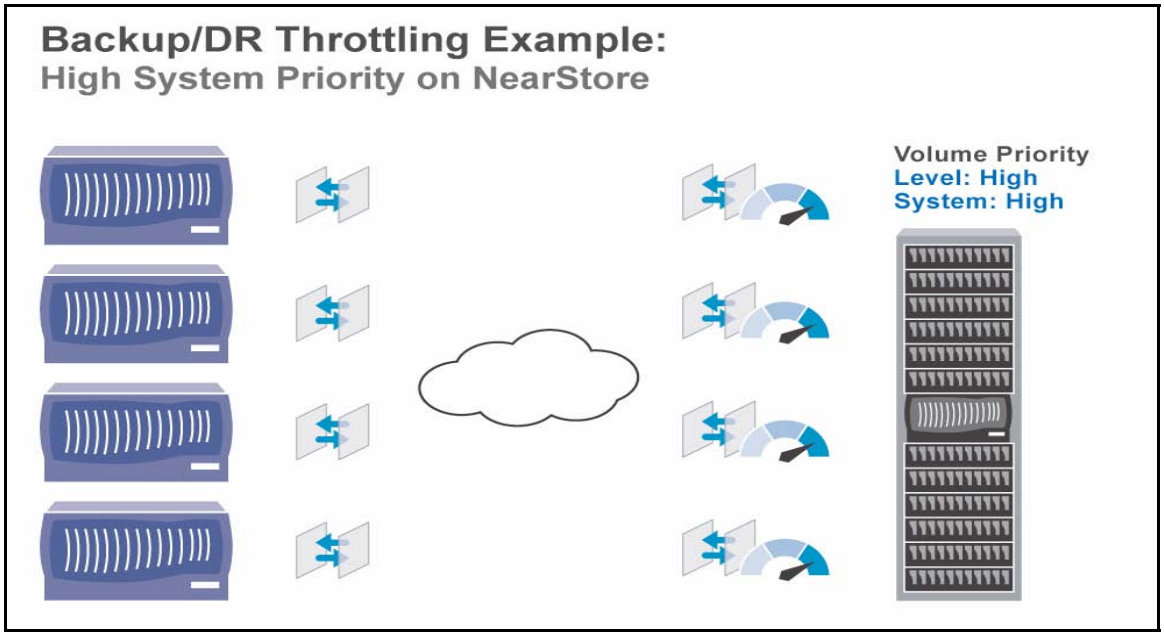

Figure 12) Backup/DR Throttling Example: High System Priority on NearStore

6.4 Multiple Application Instances

FlexShare allows storage administrators to prioritize among multiple application instances deployed on the same storage system. Common applications such as Mail and Database can frequently have multiple instances on the same storage system.

For optimal system resource balancing, it is recommended to evaluate and configure the priority level and buffer cache policy for the individual volumes of the different application instances.

Priority Guidelines

It is generally advised to set the priority level for the individual volumes of an application with the same priority level. For example, if there is an application instance that consists of data in multiple volumes – such as a database volume and a log volume, it is recommended to set the database volume and the log volume with the *same* priority level. There are often dependencies within the applications' data sets that span the different volumes. This is the safest configuration choice for generic application deployments. If a storage administrator decides to deviate from this recommendation, it is strongly advised to understand the inner workings of the particular application to avoid causing performance bottlenecks.

With multiple application instances, it is advised to set the individual volumes of each application instance with the appropriate priority level. The figure below depicts a typical example.

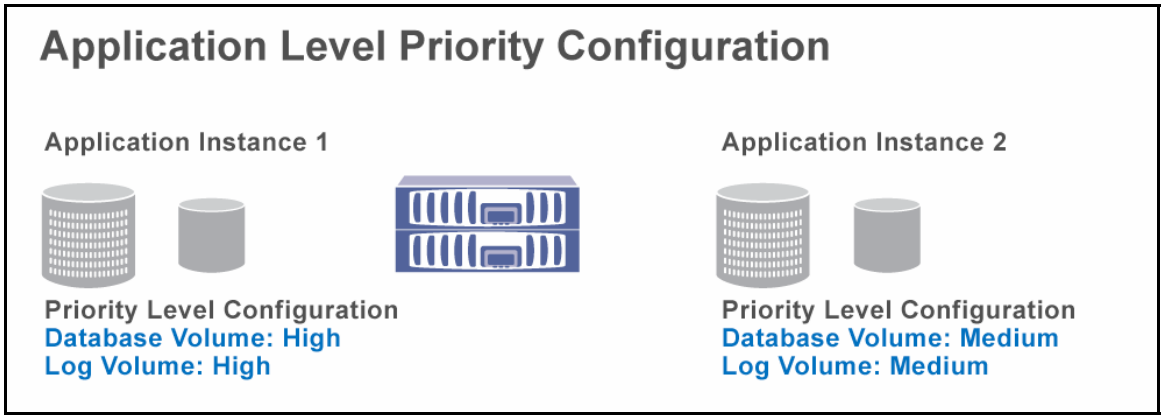

Figure 13) Application Level Priority Configuration

Recommendation on setting the priority levels for all volumes associated with an application with the same level. Prioritize the collective volumes of one application instance relative to the collective volumes of another application instance.

Buffer Cache Policy Guidelines

The application workload will significantly impact how to tune the buffer cache policy. The administrator may likely choose to set the buffer cache policy differently for particular volumes of the application. For example, an application with a database and a log volume may consider setting the database volume with a keep buffer cache policy and the log volume with a reuse buffer cache policy.

Configuring Priority for Multiple Application Instances

Figure 12 depicts two database instances deployed on the same storage system. The second instance could be an independent instance or a FlexClone™ copy of the first instance. The high priority database has a system level configuration of *High* while the lower priority database has a system level configuration of *Medium*. It is important to note that the DB volume and the Log volume for each instance are set with the same priority level.

The cache policies for each database instance should be tailored independently. In the example, the high priority database has a buffer cache policy of *keep*, while the medium priority database has a *reuse* policy. The high priority database's data set will be preferentially cached. This will likely result in a higher cache hit rate for queries on the high priority database and better performance. The log volumes are set to reuse since most of the data written to the logs is sequential. As a result, the log volumes will not benefit significantly from caching, so the cache is reserved for other data.

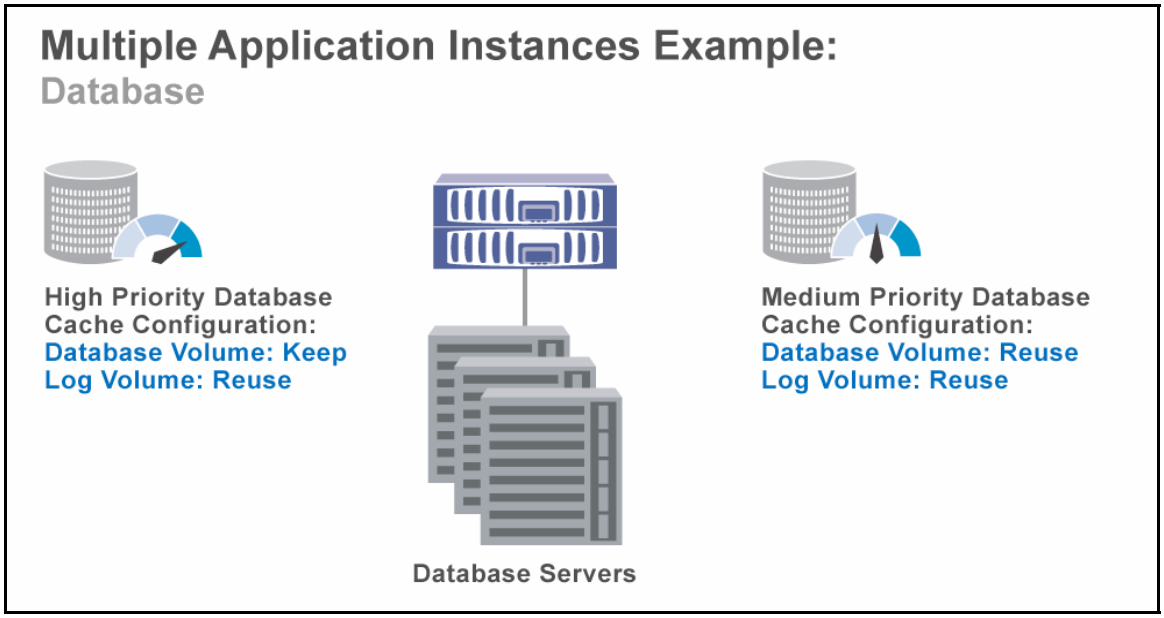

Figure 14) Multiple Applications Instances Example: Database.

The high priority database is set with a High priority level and its DB volume is set with a keep buffer cache policy for optimal performance.

7. Summary

FlexShare is a powerful Data ONTAP feature that enables storage administrators to implement workload prioritization on a storage system. It provides administrators with the ability to configure volume priority levels, user versus system priorities, and caching policies. FlexShare has significant intelligence to control and protect critical system resources.

The information in this technical report provides details on the FlexShare design, administration, best practices, troubleshooting, and high benefit use cases. Administrators are encouraged to review this material and understand the impact of configuring FlexShare. After assessing an environment's performance objectives and reviewing this material, the storage administrator should be better prepared to configure FlexShare and obtain optimally performing storage systems in their environment.

8. Acknowledgements

The following individuals made significant contributions to this paper: Darrell Suggs, Rob Fair, Hoofar Razavi, and Matti Vanninen.

9. Revision History

Date	Name	Description
03/16/2006	Akshay Bhargava	Initial Version
04/27/2006	Akshay Bhargava	Terminology Updates

www.netapp.com