NEXSAN

HYPER–UNIFIED STORAGE

Nexsan Unity

Performance Best Practices Guide

Firmware Version Unity v. 6.0

NEXSAN | 325 E. Hillcrest Drive, Suite #150 | Thousand Oaks, CA 91360 USA
Printed Tuesday, August 20, 2019 | www.nexsan.com

Trademarks

Patents

This product is protected by one or more of the following patents, and other pending patent applications worldwide:

United States patents US8,191,841, US8,120,922;

United Kingdom patents GB2466535B, GB2467622B, GB2467404B, GB2296798B, GB2297636B

About this document

Contents

Figures

About this document

This Performance Best Practices Guide explains the relevant concepts and tradeoffs to be made in an effort to allow a Unity Storage administrator to get the maximum level of performance for use cases. It is organized from bottom to top – starting with back-end storage design (RAID-sets and storage pools) and moving up through the Unity Storage System stack all the way to the hosts and the applications.

Audience

This guide has been prepared for the following audience:

- Sales Engineers
- Partners/Resellers
- Any qualified NST/Unity administrator.

Conventions

Here is a list of text conventions used in this document:

Convention	Description
underlined blue	Cross-references, hyperlinks, URLs, and email addresses.
boldface	Text that refers to labels on the physical unit or interactive items in the graphical user interface (GUI).
`monospace`	Text that is displayed in the command-line interface (CLI) or text that refers to file or directory names.
`monospace bold`	Text strings that must be entered by the user in the command-line interface or in text fields in the graphical user interface (GUI).
italics	System messages and non-interactive items in the graphical user interface (GUI) References to Software User Guides

Notes, Tips, Cautions, and Warnings

Note Notes contain important information, present alternative procedures, or call attention to certain items.

Tip Tips contain handy information for end-users, such as other ways to perform an action.

CAUTION: In hardware manuals, cautions alert the user to items or situations which may cause damage to the unit or result in mild injury to the user, or both. In software manuals, cautions alert the user to situations which may cause data corruption or data loss.

WARNING: Warnings alert the user to items or situations which may result in severe injury or death to the user.

Related documentation

The following Nexsan product manuals contain related information:

- Nexsan Unity Online Help
- *Nexsan Unity Hardware Reference Guide*
- *Nexsan Unity Hardware Maintenance Guide, Unity Next Generation*
- *Nexsan Unity Software User Guide*
- *Nexsan Unity nxadmin Command-line Interface Reference Guide*
- *Nexsan Unity nxcmd Command-line Interface Reference Guide*
- *Nexsan Unity Snapshots and Replication Guide*
- *Nexsan Unity Storage Expansion Reference Guide*
- *Nexsan Unity VMware Best Practices Guide*
- *Nexsan Unity NFS Interoperability*
- *Nexsan Unity Networking Best Practices Guide*
- *Nexsan Unity Performance Best Practices Guide*
- *Nexsan Unity Microsoft Best Practices Guide*

Chapter 1

Performance Best Practices overview

Predicting and tuning the performance of storage systems is a complex undertaking given the variety of workloads that enterprise applications and virtualized environments have. At every turn in the storage sub-system design process, many tradeoffs present themselves. Unity is built to help and guide you to make the most appropriate decisions and to select parameters that meet most of your needs, most of the time. However, no amount of auto-tuning can meet all your needs all of the time.

This document explains the relevant concepts and tradeoffs to be made in an effort to allow a storage administrator to get the maximum level of performance for use cases. It is organized from bottom to top – starting with back-end storage design (raid-sets and storage pools) and moving up through Unity stack all the way to the hosts and the applications.

This document covers the following main sections:

Chapter 2

Storage design considerations

Unity virtualizes storage enclosures and the most basic variable in Unity's performance is the performance of the disks that it virtualizes. Although storage pools can be expanded after they have been created, various design decisions made initially are very difficult to change after the fact. It is thus important to get these decisions right, right from the start.

This section covers the concepts and best practices related to properly designing storage pools:

Storage pools

- Pools are virtualized disks
- LUNs are presented as blocks
- Up to 30 pools on Unity
- Cannot decrease size or reconfigure after pool creation
- Can assign FASTier Read and/or Write cache at any time

On Unity, storage is organized into storage pools which are striped over storage volumes. Storage volumes are RAID sets made up of virtualized disks—either individual disks or LUNs presented as a storage blocks. Storage pools can have arbitrary size and there can be up to 30 different pools in a single Unity. After creation, storage pools can be expanded in capacity and performance by adding new volumes and extending the top-level stripe across those volumes. Storage pool capacity cannot be decreased, nor can used volumes be reconfigured.

Each pool can be assigned different amounts and types of FASTier read and write cache in order to reduce response time to I/Os and also to reduce the load on the pool storage and any associated disks.

Unity Storage Systems

- Choose a storage system for the storage pools
- Select the storage system based on performance density requirements of the target applications
- FASTier can be used to raise the total effective performance of the storage pool

The following Unity Storage Systems are available. These are the fundamental building blocks of any Unity storage solution:

UNITY2200 and UNITY2200X

Disk drive specifications for these Unity storage expansions:

Feature	Description
Form factor	3U 3.5” front-loading
Disk drives	Up to 15 HDDs and one FASTier read drive
	HDD drives: 7.2K 600 GB | 900 GB | 1.2 TB | 1.8 TB 10K: 2TB | 4TB | 6TB | 8TB | 10TB | 12TB
	FASTier SSD drives One 800 GB drive per drive pack, option to upgrade to 1.92 TB FASTier or 3.84 TB drives
Cascadable?	No
Hot-swap/hot-plug support	Yes
Interface	SAS 3
Transfer rate	12.0 Gb/sec
Upgradable disk drive firmware?	Yes

US316

Disk drive specifications for these Unity storage expansions:

Feature	Description
Form factor	3U 3.5” front-loading
Disk drives	Up to 15 HDDs and one FASTier read drive
	HDD drives: 7.2K 600 GB | 900 GB | 1.2 TB | 1.8 TB

Feature	Description
	10K: 2TB | 4TB | 6TB | 8TB | 10TB | 12TB
	FASTier SSD drives One 800 GB drive per drive pack, option to upgrade to 1.92 TB FASTier or 3.84 TB drives
Cascadable?	Yes
Hot-swap/hot-plug support	Yes
Interface	SAS 3
Transfer rate	12.0 Gb/sec
Upgradable disk drive firmware?	Yes

UNITY4400, UNITY6900, and US224

Disk drive specifications for these Unity storage expansions:

Feature	Description
Form factor	2U 2.5” front-loading
Disk drives	Supports up to 24 drives (SSD option)
	SSD drives: 800 GB | 1.9 TB | 3.8 TB | 7.6 TB
	HDD drives 7.2K: 600 GB | 900 GB | 1.2 TB | 1.8 TB 10K: 2TB | 4TB | 6TB | 8TB | 10TB | 12TB
	FASTier SSD drives: 800 GB per drive pack, option to upgrade to 1.92 TB FASTier or 3.84 TB drives. Not used when all data drives are SSDs.
Cascadable?	Yes.
Hot-swap/hot-plug support	Yes
Interface	SAS 3
Transfer rate	12.0 Gb/sec
Upgradable disk drive firmware?	Yes

US424

Disk drive specifications for these Unity storage expansions:

Feature	Description
Form factor	4U 3.5” front-loading
Disk drives	Supports up to 24 drives
	HDD drives .2K: 2TB | 4TB | 6TB | 8TB | 10TB | 12TB 10K: 2TB | 4TB | 6TB | 8TB | 10TB | 12TB
	FASTier SSD drives: 800 GB per drive pack, option to upgrade to 1.92 TB FASTier or 3.84 TB drives
Cascadable?	Yes
Hot-swap/hot-plug support	Yes
Interface	SAS 3
Transfer rate	12.0 Gb/sec
Upgradable disk drive firmware?	Yes

US460

Disk drive specifications for these Unity storage expansions:

Feature	Description
Form factor	4U 3.5” top loading
Disk drives	
	Supported drives: HGST or Sandisk 2TB | 4TB | 6TB | 8TB | 10TB | 12TB
	FASTier SSD drives: 800 GB FASTier per drive pack. Option to upgrade to 1.92 TB FASTier or 3.84 TB drives
Cascadable?	Yes
Hot-swap/hot-plug support	Yes
Interface	SAS 3
Transfer rate	12.0 Gb/sec
Upgradable disk drive firmware?	Yes

Figure 2-1: Choosing a storage system

Regardless of the Unity Storage System chosen for your design, the first storage provisioning task will be to create volumes (RAID sets) and to assign those volumes to storage pools. The following sections will explain the concepts and tradeoffs involved in this phase of provisioning.

► **Related topics:**

Monitoring disk performance below

Monitoring disk performance

For any disk connected to Unity, performance can be monitored using the following CLI commands:

- `nstpool iostat <period>`
- `iostat -xtncM <period>`

These statistics allow you to learn how many IOPS each disk is handling, the average number of I/Os that Unity is keeping outstanding to them, and the average response time of the disks themselves. The `%busy` statistic provides a simple metric to assess the load on the disks, but it can be misleading.

Disks that show 100% busy may not be pushed to the limit. 100% simply means that they always have at least one command outstanding to them. However, higher IOPS may be possible by keeping a larger number of commands outstanding to them, so there may still be performance capacity left when a disk reports 100% busy.

Figure 2-2: iostat –xtncM 5

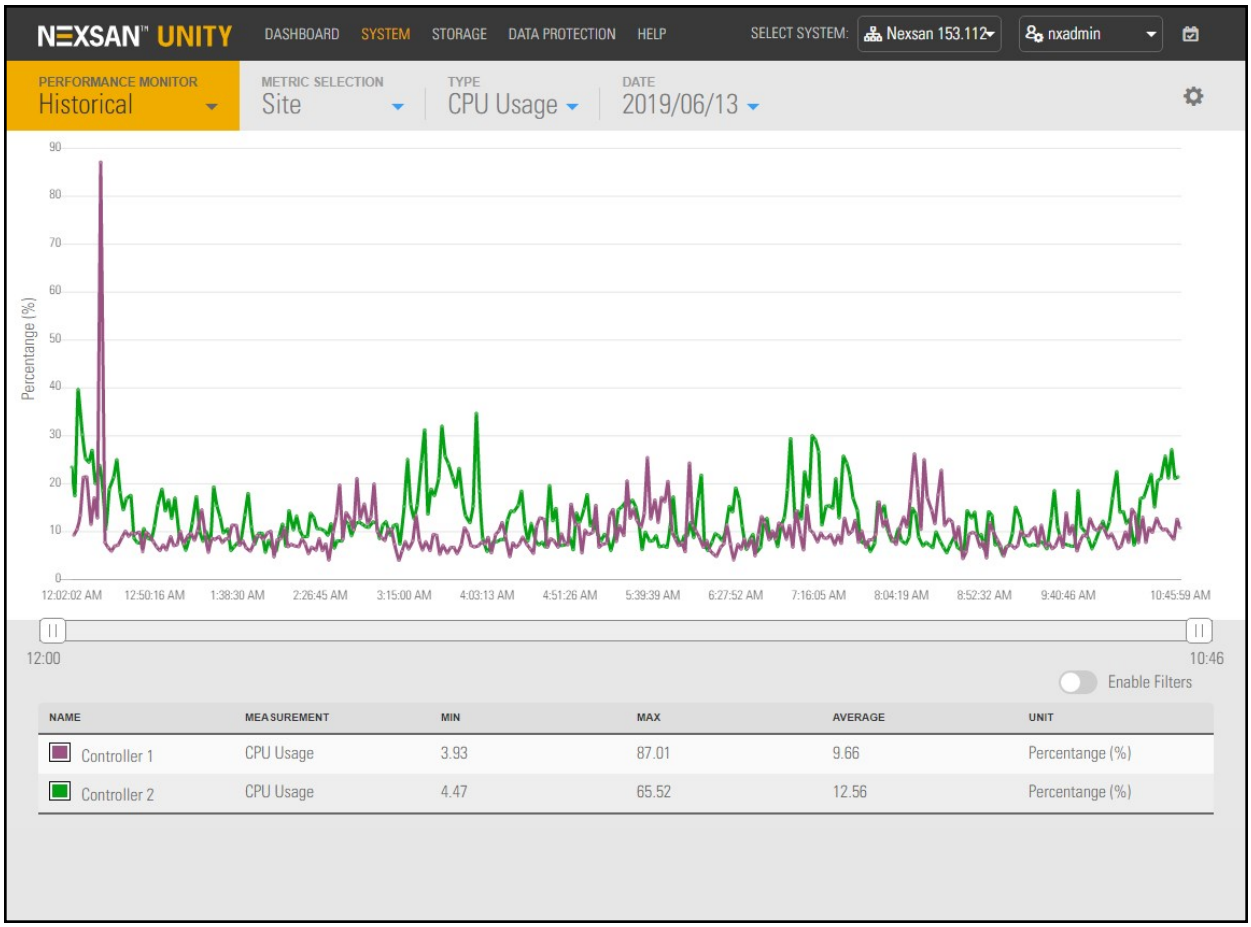
```
  tty         cpu
tin tout  us sy wt id
  0  983   1 18  0 81
                  extended device statistics
   r/s    w/s   Mr/s   Mw/s wait actv wsvc_t asvc_t  %w  %b device
  16.0   15.0    0.0    0.0  0.0  0.0    0.0    0.6   0   1 c0t0d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c0t1d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394FC890EF8d0
   1.8    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC890F10d0
   2.2    1.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC8901F0d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394FC8848A0d0
   3.2    1.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC88DE70d0
   7.6    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC890F00d0
   6.0    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC8848C0d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394EC8810D0d0
   1.0    1.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC88DE64d0
   1.0    1.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC890F04d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394EC882F84d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394FC8901F4d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394FC890F14d0
   0.4    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC890F08d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394FC8848A8d0
   9.4    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC88DE58d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394FC890F18d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394FC8848B8d0
   0.6    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC88DE68d0
   7.0    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC8901CCd0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t50000394FC88DE5Cd0
   1.8    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394EC88018Cd0
   2.0    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC890EFCd0
   1.0    1.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC890F0Cd0
   1.0    1.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 c2t50000394FC8848ACd0
 204.0  128.0    0.1   10.5  0.0  3.5    0.0   10.5   0  82 c2t5000C5005F8D4543d0
 180.4  130.2    0.1   10.2  0.0  2.9    0.0    9.4   0  76 c2t5000C5005F8D46E3d0
 160.6  132.2    0.1   10.2  0.0  2.7    0.0    9.1   0  71 c2t5000C5005F8D4823d0
 179.2  126.0    0.1   10.4  0.0  2.8    0.0    9.0   0  74 c2t5000C5005F8CE1A3d0
 155.8  125.8    0.1   10.4  0.0  2.8    0.0   10.1   0  66 c2t5000C5005F8D4863d0
 200.0  140.0    0.1   10.4  0.0  3.1    0.0    9.0   0  80 c2t5000C5005F8D41E3d0
 182.6  130.2    0.1   10.5  0.0  3.3    0.0   10.5   0  75 c2t5000C5005F858AF7d0
 174.0  108.0    0.1   10.5  0.0  3.2    0.0   11.5   0  75 c2t5000C5005F8CEF97d0
 198.8  122.0    0.1   10.5  0.0  2.8    0.0    8.7   0  77 c2t5000C5005F858C37d0
 176.4  129.2    0.1   10.4  0.0  2.9    0.0    9.5   0  70 c2t5000C5005F8D5167d0
 198.0  127.8    0.1   10.5  0.0  3.3    0.0   10.1   0  79 c2t5000C5005F858EFBd0
 192.6  125.0    0.1   10.5  0.0  3.8    0.0   12.0   0  78 c2t5000C5005F858FBBd0
 201.0  124.2    0.1   10.5  0.0  3.8    0.0   11.7   0  80 c2t5000C5005F8D562Bd0
 190.4  123.6    0.1   10.5  0.0  3.1    0.0    9.9   0  75 c2t5000C5005F8CE02Bd0
 182.0  131.4    0.1   10.5  0.0  3.2    0.0   10.3   0  73 c2t5000C5005F8CD9ABd0
 189.0  123.0    0.1   10.5  0.0  2.9    0.0    9.2   0  74 c2t5000C5005F8D484Bd0
 158.8  123.4    0.1   10.5  0.0  2.5    0.0    8.9   0  67 c2t5000C5005F8CDBEBd0
 171.8  128.6    0.1   10.2  0.0  2.9    0.0    9.5   0  74 c2t5000C5005F8D506Bd0
 204.4  127.8    0.1   10.2  0.0  3.1    0.0    9.3   0  80 c2t5000C5005F8D0FABd0
 223.4  124.6    0.1   10.2  0.0  3.7    0.0   10.7   0  82 c2t5000C5005F8D45ABd0
 214.4  125.2    0.1   10.4  0.0  3.8    0.0   11.1   0  78 c2t5000C5005F858DFBd0
 224.4  122.0    0.1   10.5  0.0  3.6    0.0   10.3   0  84 c2t5000C5005F858E6Fd0
 196.4  127.2    0.1   10.4  0.0  3.3    0.0   10.3   0  78 c2t5000C5005F8D508Fd0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t600144F09CF4800000005259542D0001d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t600144F084708600000005259547100001d0
 171.4  114.2    0.1   10.2  0.0  3.8    0.0   13.3   0  81 c2t5000CCA01D03D5C8d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t500003954C881620d0
   0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 c2t5000C5005E64CE4Fd0
   2.0    0.0    0.0    0.0  0.0  0.0    0.0    0.1   0   0 127.0.0.1:///nest_syspool/config
```

Figure 2-3: nstpool iostat –v <pool name> 5

```
                          capacity     operations    bandwidth
pool                    alloc   free   read  write   read  write
----------------------  -----  -----  -----  -----  -----  -----
sgt-3d21a2fc-5eea-464a-9602-a328586c8b9f  6.73T  19.3T  1.20K  28.8K  1.67M   404M
  raidz2                1.69T  4.81T    303  6.34K   427K  43.3M
    c2t5000C5005F8D508Fd0      -      -    158    209  81.7K  11.0M
    c2t5000C5005F8D41E3d0      -      -    120    202  62.7K  10.9M
    c2t5000C5005F8D5167d0      -      -    113    178  59.2K  11.0M
    c2t5000C5005F8D4863d0      -      -    106    208  56.3K  11.0M
    c2t5000C5005F8CE1A3d0      -      -    157    179  81.4K  10.9M
    c2t5000C5005F858DFBd0      -      -    167    213  90.8K  11.0M
  raidz2                1.69T  4.81T    303  6.43K   426K  43.7M
    c2t5000C5005F8D45ABd0      -      -    156    115  80.2K  11.1M
    c2t5000C5005F8D0FABd0      -      -    135    118  69.0K  11.1M
    c2t5000C5005F8D4823d0      -      -    123    119  63.0K  11.1M
    c2t5000C5005F8D46E3d0      -      -    129    123  65.7K  11.1M
    c2t5000C5005F8D506Bd0      -      -    148    112  75.7K  11.1M
    c2t5000CCA01D03D5C8d0      -      -    140    112  72.2K  11.1M
  raidz2                1.68T  4.82T    312  6.43K   434K  43.5M
    c2t5000C5005F8CDBEBd0      -      -    138    129  70.5K  11.0M
    c2t5000C5005F858C37d0      -      -    154    120  78.6K  11.0M
    c2t5000C5005F858E6Fd0      -      -    152    126  77.8K  11.0M
    c2t5000C5005F8D484Bd0      -      -    133    132  67.9K  11.0M
    c2t5000C5005F8CD9ABd0      -      -    133    115  67.7K  11.0M
    c2t5000C5005F8CE02Bd0      -      -    138    124  71.0K  11.0M
  raidz2                1.68T  4.82T    307  6.21K   428K  42.2M
    c2t5000C5005F8CEF97d0      -      -    133    120  68.0K  10.7M
    c2t5000C5005F8D562Bd0      -      -    128    121  65.8K  10.7M
    c2t5000C5005F8D4543d0      -      -    149    126  76.2K  10.7M
    c2t5000C5005F858AF7d0      -      -    137    124  69.1K  10.7M
    c2t5000C5005F858FBBd0      -      -    150    122  75.7K  10.7M
    c2t5000C5005F858EFBd0      -      -    142    123  72.9K  10.7M
logs                        -      -      -      -      -      -
  :nvram8                 366M   650M      0    436      0  28.9M
  :nvram9                 366M   650M      0    436      0  29.0M
  :nvram10                366M   650M      0    436      0  28.9M
  :nvram11                366M   650M      0    436      0  28.9M
  :nvram12                366M   650M      0    436      0  29.0M
  :nvram13                366M   650M      0    436      0  28.9M
  :nvram14                366M   650M      0    436      0  28.9M
  :nvram15                366M   650M      0    436      0  28.9M
----------------------  -----  -----  -----  -----  -----  -----
```

In addition, the Nexsan Unity Web interface provides a performance monitoring view where you can see whether the controllers are busy and whether the disks are busy (both metrics are a percentage).

Figure 2-4: Nexsan Unity performance page



Resource groups

- Resource groups are load-balanced across controllers in a cluster
- Any one resource group or storage pool can be owned by only a single controller at any one time
- Create at least 2 storage pools and assign these pools to separate resource groups

Storage Pools are assigned to resource groups, of which there are 2 in today's product. Resource groups are load-balanced across controllers in an Unity cluster. Any one resource group or individual storage pool can be owned by only a single controller at any one time. All I/O to a resource group will depend on one controller's:

- ports for access to storage
- CPU for processing of I/Os
- memory for first level read cache
- memory for write cache (if write FASTier is assigned to the pool)

These are resources that can be starved and can act as a bottleneck on performance in some configurations. **A basic principle is to organize a system's storage into at least 2 storage pools and to assign these pools to separate resource groups.** This way, I/O to the system can leverage the ports, CPU, and memory of both controllers and these three important resources are effectively doubled as compared to a system utilizing a single storage pool.

Write operations into free space and transactional I/O

- Write operations are collected into transaction groups and later written sequentially to their final location in the storage pool
- Keep at least 20% free capacity in the storage pool for better performance

In Unity architecture, write operations are collected into transaction groups and later written sequentially to their final location in the storage pool. Typically, the transaction group is temporarily stored on fast solid-state disk until it is committed. When the transaction group is committed to disk, Unity writes it out sequentially to the pool media (typically spinning media). This allows for much better performance from the pool media than other architectures are able to achieve.

However, for this to work well, enough contiguous free capacity must exist in the pool. For this reason, **it is recommended to keep at least 20% free capacity in the storage pool.** As the pool fills up, it is possible that performance will degrade.

FASTier cache

Unity includes support for Nexsan's FASTier™ cache devices which provide non-volatile high-speed cache memory. The FASTier read, write, or read/write caching mechanism enhances the performance and speed for a storage pool.

- FASTier read cache stores frequently-read chunks of data to accelerate read operations on the system.
- FASTier write cache accelerates synchronous write operations to disk.
- FASTier read/write cache can be used for either read or write cache for the storage pool.

FASTier Read cache

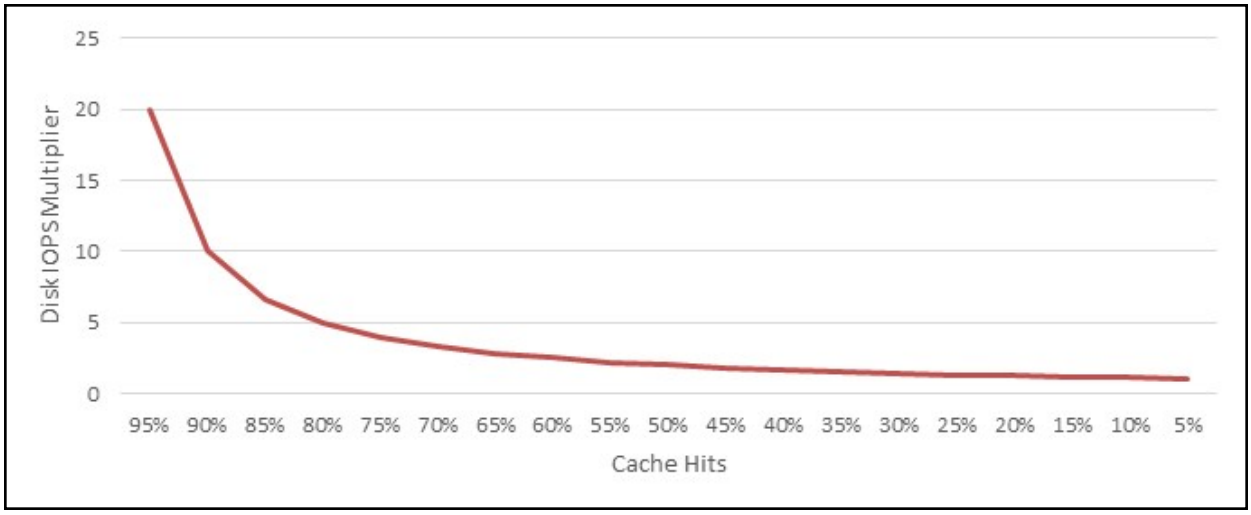
- FASTier Read cache is applied at the storage pool level
- FASTier Read cache is most effective at accelerating workloads featuring random I/Os
- Use eMLC drives for FASTier Read cache

FASTier Read cache is applied at the storage pool level and is used to cache the most frequently accessed blocks from that storage pool. This has the effect both of serving those particular blocks with lower latency as well as offloading operations from the pool disks, increasing their available performance to service other I/O operations.

The effectiveness of FASTier will vary with the workloads applied to the pools. Generally, it is most effective at accelerating workloads featuring random I/Os—email, database, virtualized servers, etc. This is because flash or SSD technology is significantly faster than spinning media at servicing random I/Os, but only slightly faster at servicing sequential I/Os.

The less uniform and the less random the I/O pattern is, the more effective FASTier can be at anticipating which blocks/files will be read next and the more cache hits you can generally expect. Cache hits have a multiplier effect on effective system IOPS—the multiplier is relative to the number of IOPS that the disk itself is capable of servicing. This effect is shown in the table below.

Figure 2-5: Effect of cache hits on IOPS



It is recommended to use eMLC drives for FASTier Read cache. Their endurance is high enough that you should expect them to last longer than 5 years in this role.

FASTier Write cache

- Always configure FASTier Write cache on any pool to which there will be regular write traffic
- The amount of FASTier Write cache required depends on the number of pools and the expected volume of write traffic; calculate 10s worth of writes
- The FASTier device type to use depends on the expected average volume of writes, the peak write performance desired, and Unity hardware platform

Incoming writes to Unity are collected into transaction groups that are first stored in high-speed DRAM-based memory. If FASTier Write cache is configured, incoming writes get written to FASTier before being acknowledged to the host. This way, those writes will not be lost in the event of a power or controller failure. If FASTier Write cache is not present, incoming writes are written to a transaction log on the pool media itself so that it is kept safe. In both cases, the transaction group is later de-staged to the pool media. **Hence, when no FASTier Write cache is configured, each incoming write is written to the pool twice. This is referred to as the double-write penalty.**

It is recommended to always configure FASTier Write cache on any pool to which there will be regular write traffic.

The appropriate amount of FASTier Write cache to configure for a pool depends on the number of pools in the system and the expected volume of write traffic. As a general rule, consider that you should have enough FASTier Write cache to absorb up to 10s worth of writes. For example, if the write throughput is estimated at 200 MB/s, then 2 GB of FASTier Write cache should be enough.

The type of FASTier device to use depends on the expected average volume of writes, the peak write performance desired, and Unity hardware platform.

Transaction groups

Transaction groups in cache are closed either:

- when they are full (the system allows a global maximum amount of memory to be dedicated to caching writes), or
- after a 5-second interval.

When a transaction group closes, a new one is opened to accept new incoming writes. The closed transaction group is committed sequentially to the pool media.

Chapter 3

Dataset considerations

Unity unified storage platform provides access to data either in block or file format. For block LUNs, access can be via Fibre Channel or iSCSI. File shares can be accessed via CIFS, NFS, or FTP.

For both types of datasets, a block size (or record size) can be chosen, compression can be enabled or disabled on the fly, and the VAAI primitives are available. Finally, datasets can be replicated asynchronously via a TCP/IP link for disaster recovery purposes.

This section covers the following topics:

Block size and record size requirements

- Available capacity of a LUN or share depends on the selected block size
- Block size affects performance
- LUNs have a default block size of 8 KB, recommended for transactional workloads
- Choose the LUN block size that best matches the application's intended access pattern
- A LUN's block size cannot be modified once the LUN is created
- File Systems have a default record size of 128 KB
- A share's record size can be modified at any time
- A share's record should not be changed except if the application is database-driven and the database record size is precisely known

Unity keeps a small amount of metadata (including checksum, compression details, and block re-mapping metadata) on a per-block basis. Hence, the available capacity of a LUN or share will depend on the block size chosen, with larger block sizes providing marginally more usable capacity due to lowered overhead. The block size also affects performance as it is the smallest size of data that is addressable on the storage system.

For instance, given a 128 KB block size, if a Read is issued from the host for a 4 KB subset of this block, Unity will read the entire 128 KB before returning the 4 KB subset that was requested. This type of situation leads to reduced performance. The flip side of this is when the application performs large I/Os (that is, 1 MB) and the block size is small (that is, 8 KB). In this case, to perform a single 1 MB I/O, Unity will have to perform several independent data and metadata accesses rather than a single one. This can also lead to reduced performance.

Hence, the primary consideration when choosing a block size is to choose the block size that best matches the application's intended access pattern. Some common applications and their typical block sizes are listed in the table below.

LUNs have a default block size of 8 KB. This is the default choice for transactional workloads. Large-block streaming workloads require a higher block size, such as 128 KB, to obtain better streaming read performance. Generally speaking, if you know the block size that your application typically uses, set Unity LUN block size to match.

Note A LUN's block size cannot be modified once the LUN is created.

File Systems have a default record size of 128 KB. However, this record size is somewhat dynamic and Unity will write in smaller blocks if the application would seem to benefit from it. It is not recommended to change the default record size of shares except in very specific circumstances where the application is database-driven and the database record size is precisely known.

Note The share's record size can be modified at any time.

Refer to this table for recommended block/record size according to the application used.

Application / Data type	Recommended block size
Windows / Linux Boot	64 k
Microsoft Exchange 2010 DB	32 k
Microsoft Exchange 2010 Log	128 k

Application / Data type	Recommended block size
Oracle OLTP	8 k
SQL Server	64 k
Video Streaming	128 k

Compression

- Use compression for datasets that are compressible
- Do not use compression for datasets containing data that cannot be compressed—such as videos, images, music, and any data which is already compressed or encrypted
- If the resulting compression ratio is less than 1.5:1, disabling compression may lead to lower latency on I/Os, at the expense of losing the capacity savings

Unity has the ability to perform in-line compression and decompression on a block by block basis. This compression typically leads to better performance when a dataset is compressible. This is because the compression algorithm is very efficient and the impact on memory and CPU is quite limited. The small added latency that compression introduces has often less impact than the lower volume of I/O that Unity has to push and retrieve from the disk subsystems.

Most Unitys are disk-bound—not CPU-bound. Compression can be enabled or disabled on the fly and when this setting is modified, it applies to any newly written data without affecting already written data. As such, **it is recommended to use compression except for datasets containing data that cannot be compressed—such as videos, images, music, and any data which is already compressed or encrypted**. If the resulting compression ratio is less than 1.5:1, disabling compression may lead to slightly lower latency on I/Os, at the expense of the capacity savings that would be lost. This decision can be taken once the achievable compression ratio is known.

VAAI overview

- VAAI hardware acceleration allows the VMware ESX/ESXi host to perform VM and storage management operations faster and consume less CPU, memory, and storage fabric bandwidth.
- Leave VAAI enabled on all LUNs unless advised otherwise by Nexsan support staff.

Unity includes a built-in VAAI (vStorage APIs for Array Integration) plug-in to provide hardware acceleration on Unity when integrated into a VMware ESX/ESXi environment. VAAI hardware acceleration functionality enables the VMware ESX/ESXi host to offload specific virtual machine and storage management operations to Unity. With storage hardware assistance, the VMware ESX/ESXi host performs these operations faster and consumes less CPU, memory, and storage fabric bandwidth.

There is no known benefit under any circumstances to disabling it, hence **it is recommended to keep enable VAAI enabled on all LUNs unless advised otherwise by Nexsan support staff**.

You must configure and enable VAAI hardware acceleration functionality on your VMware ESX/ESXi to support for the VAAI plug-in to work. In VMware, the VAAI plug-in is visible at the datastore level. Refer to the VMware documentation for further details.

Asynchronous replication

- Asynchronous replication is performed by taking a snapshot and sending the contents of that snapshot to the disaster recovery site
- Asynchronous replication only replicates the data that changed since the last replication

Asynchronous replication can be enabled or disabled on a per-dataset basis. The replication is performed by taking a snapshot and sending the contents of that snapshot to the disaster recovery site. The snapshot is applied at the secondary site automatically—it is either committed there or it is not, in the event of any unrecoverable errors. The contents of a snapshot are the deltas since the last successfully replicated snapshot. Also, replication data can be compressed prior to being transmitted for the purposes of making the most efficient use possible of the WAN link resources.

The replication process does have some memory, CPU and I/O overhead. Refer to the *Nexsan Unity nxadmin CLI Reference Guide* for more details.

Synchronous replication

- Synchronous replication ensures that the data on Unity is always current and it is performed at the pool level
- Synchronous replication affects performance since every write operation is performed twice (locally and remotely)
- Read IOPS load is reduced on the local storage, but adds network latency to every other I/O request
- Additional latency inserted on Reads or Writes at 10 KM of Fibre Channel distance between sites is only 150 us

In version 2.2 of Unity, Read operations are "round-robined" between the local storage and the remote storage as well. This reduces the Read IOPS load to the local storage, but does add network latency to every other I/O request.

Note The maximum amount of additional latency inserted on Reads or Writes at 10 KM of Fibre Channel distance between sites is only 150 us, less than the typical latency of the disks themselves.

Protocols

LUN access protocols

Due to higher port density and the more efficient protocol layer, Fibre Channel can provide higher levels of performance than iSCSI but it requires a dedicated SAN infrastructure in the customer datacenter.

File access protocols

- Unity uses CIFS, NFS, and FTP
- These MTU and LACP network settings improve performance.

Unity supports file share access through CIFS, NFS, and FTP. There are very few settings within Unity to tune the performance with these protocols. However performance is sensitive to network settings. Two

settings within Unity are important to consider—the network MTU and whether LACP is enabled or not. These settings are covered in Network considerations on page 27.

Chapter 4

Network considerations

This section describes network hardware, cabling, and connectivity considerations. It also provides troubleshooting steps when encountering network issues.

This section covers the following topics:

Network MTU

- Configure data networks with jumbo frames: MTU = 9000
- The MTU setting must also be configured on the switch side and any other machine connected to Unity

Achieving high performance on 10GbE ports requires the use of jumbo frames, or large MTU. As such, it is recommended to configure the data networks with an MTU (payload, not frames) of 9000. This increases the payload size of each packet, leading to reduced metadata overhead. Modifying the MTU is a disruptive operation for the network but it is possible to set it one controller at a time.

The MTU must also be configured on the switch side, as well as any other machines that interact with Unity. Some switches only support changing MTU as a global setting; this means any machine connected to this switch would also need its MTU updated, even if it does not interact with Unity.

Aggregations

- Primary data network interface (nx0)
- Secondary data network interface (nx1)
- Management Interface (nx99)

Unity binds one or more physical ports together into a single virtual port, known as an aggregation. These aggregations are named *nx0*, *nx1*, and *nx99* within Unity.

For details on aggregations and network interfaces, refer to the *Nexsan Unity Software User Guide*.

LACP

- LACP must be enabled on Unity and the switch
- LACP provides load balancing on a connection basis

Unity supports the use of the Link Aggregation Control Protocol (LACP). When LACP is enabled within Unity and within the switch, Unity can balance the load of traffic for a single virtual IP (VIP) over multiple physical Ethernet ports in an aggregation, as well as provide seamless fault tolerance for port/cable failures.

This load balancing operates on a connection basis, not a packet basis. This means multiple clients are typically required to achieve full saturation of an aggregation.

Troubleshooting network issues

Having a healthy network infrastructure is important to ensure optimal operation of your Unity since typically several machines will be communicating with Unity over a variety of protocols (AD, NFS, iSCSI, NDMP, and SMTP to name a few). Networking issues can manifest themselves many ways; some of the more common symptoms are inability to connect to an IP, slow connections, and intermittent networking errors.

Unity provides several mechanisms to monitor networking performance. Throughput can be monitored via Unity's Performance Monitor, or via CLI commands (`nic show-link -s`). The CLI commands can also show per-port granularity to help identify bottlenecks. Every component from the client to Unity should be examined to determine where the problem lies.

To verify network status:

- Verify each controller on Unity can continuously ping its peer controller.
- Verify each Unity controller can ping the gateway.
- Test that a client can ping each controller and the relevant Virtual IPs.
- Check switch configurations; some switches need additional configuration to recognize aggregated links.
- Check link speeds with the `nic show-phys` CLI command.
- If the problem is intermittent (dropped packets or lost pings), try removing links from the aggregation.
- Network complexity should be reduced as much as possible to try and isolate the faulty component/configuration.

To detect a wrong cabling link between the switches and Unity:

- For each network port on Unity, ask to the network administrator to bring down the port one by one on the switch(es).
- Verify on both controllers of Unity which port is down and verify if that corresponds with the wanted configuration.

This image provides an example of a down link.

```
ES200100-001-02:P:/> nic show-link
LINK        CLASS     MTU    STATE    BRIDGE     OVER
igb0        phys      1500   up       --         --
igb1        phys      1500   unknown  --         --
ixgbe0      phys      1500   up       --         --
ixgbe1      phys      1500   up       --         --
ixgbe2      phys      1500   up       --         --
ixgbe3      phys      1500   down     --         --
nx0         aggr      1500   up       --         ixgbe2 ixgbe3
private0    aggr      1500   up       --         ixgbe0 ixgbe1
nx99        aggr      1500   up       --         igb0
ES200100-001-02:P:/>
```

To detect a faulty physical network link between the switches and Unity:

- Run this command:

  `nic show-link -s`

  Under the column **IERRORS**, you will see a value bigger than **0**.

```
ES200100-001-02:P:/> nic show-link -s
LINK            IPACKETS   RBYTES   IERRORS    OPACKETS    OBYTES       OERRORS
igb0            131689745 117511754969 0       399300      38884428     0
igb1            0          0         0         0           0            0
ixgbe0          37109753   7918539878 0       43152056    19987036004  0
ixgbe1          37590215   6889103519 0       44060756    19761741407  0
ixgbe2          31251553   1945248265 5283    512069      29422744     0
ixgbe3          31043547   1938512832 0       420787      26774212     0
nx0             62295100   3883761097 5283    932856      56196956     0
private0        74699968   14807643397 0      87212812    39748777411  0
nx99            131689745 117511754969 0       399300      38884428     0
```

Chapter 5

Host considerations

We recommend following host operating system and application best practices as published by host operating system and application vendors. However, this section provides some settings on the client-side that are known to improve performance in some cases.

This section covers these topics:

Windows

- Windows clients should use CIFS for higher performance
- For large block streaming, raise the TCP window size to 2 MB

For Windows clients, it is recommended to use the CIFS protocol rather than the NFS protocol as the performance achieved is typically higher. Also, for customers doing large block streaming, we recommend raising the TCP Window Size in Windows to be 2 MB.

Please refer to this article for details:

http://technet.microsoft.com/en-us/library/cc938219.aspx

Linux

- For NAS storage, use NFSv3 or NFSv4
- For iSCSI and NAS, set the TCP window sizes to 4 MB
- For Fibre Channel, use `dm multipath 0.5.0` for enhanced failover

On Linux platforms, a few best practices apply. First, for NAS use cases, NFS typically outperforms CIFS and it is recommended to use NFSv3 or NFSv4 for NAS access.

For iSCSI and NAS, performance can be optimized by setting the TCP window sizes to 4 MB (see below).

Finally, for Fibre Channel access, it is *highly* recommended to use version `0.5.0` or greater of the `dm multipath` package. This version has better performance than version `0.4.9` which is currently standard on most distributions. Moreover, `dm multipath 0.5.0` provides enhanced failover and fail back in the event of Unity controller failover.

To set the TCP window size in Linux:

- Edit the `/etc/sysctl.conf` file and add/edit the following lines.

```
net.ipv4.tcp_moderate_rcvbuf=1
net.core.wmem_max=4194304
net.core.rmem_max=4194304
net.ipv4.tcp_wmem="4096 2097152 4194304"
net.ipv4.tcp_rmem="4096 2097152 4194304
```

VMware Virtual machine recommendations

- Use VMware Workstation 8 or higher
- Use raw device mapping (RDM) for raw devices and LUNs with more than 2 TB
- If you are using VSS, use RDM in physical compatibility mode
- Install VMware Client tools
- With Windows VMs, install the latest service pack
- Work with a central swap datastore for all VMs
- Work with a multi-pool design with multiple datastore repositories

We recommend these best practices for VMware virtual machines:

- Work with VMware version 8 minimum.
- For raw devices and for LUNs with more than 2 TB, use raw device mapping (RDM).
- If you are using Unity with Microsoft Volume Shadow Copy, use RDM in physical compatibility mode.
- On Unity, it is highly recommended to host all storage pools on the same controller. The VAAI plugin does not work across controllers so operations like cloning or vMotion will not benefit from VAAI if the pools are located on different controllers.
- Install the VMware Client Tools. For more information on these tools and to install them, refer to the VMware documentation.
- When working with the Microsoft Windows platform, ensure that you have the latest service pack as well as all recommended patches installed.
- Make sure that your virtual machine is working with the right partition alignment.
- Work with a central swap datastore for all virtual machines. By default VMware creates a virtual swap file that usually is equal to the amount of memory allocated to each virtual machine. Reallocate the virtual machine swap file to a central VMware datastore.
- To achieve better performance for virtualized applications as well as management of your virtual environment, work with a multi-pool design with multiple datastore repositories in VMware vSphere 5.x.

For further details on setting up VMware, refer to the *Nexsan Unity VMware Best Practices Guide*.

NFS

- Set the block size to optimize transfer speeds: `rsize=32768` and `wsize=32768`
- Allow the same file system to be mounted repeatedly using `noac`
- Reduce the number of metadata operations using `noatime`

NFS is a protocol rich in options that are appropriate in different circumstances. This section is not meant to be exhaustive but rather just to help the reader avoid common pitfalls and implement common best practices.

We recommend the following options to always be set:

- `rsize=32768`
- `wsize=32768`
- `noac`
- `noatime`

The `noac` option can be taken off in order to improve performance, but this is known to cause high latency on certain file system metadata operations. This latency is due to caching of data on the NFS client and the fact that for certain metadata operations (`fsstat`, for example). The client needs to first flush its cache before it can issue the metadata operation.

Finally, where appropriate, we recommend using the `noatime` option in order to reduce the number of metadata operations and improve overall performance.

Chapter 6

Performance FAQs

This section provides questions to help you identify and solve performance issues. If you need to contact Nexsan Technical Support, please provide details about the performance issues you are experiencing on your Unity, especially the methodology or benchmark being used to report your performance issues (for example, slow responsiveness of VMs, low data transfer speeds, etc.).

Hardware configuration

- How many disks do you have?
- What type of disks do you have? (SAS/SATA)
- How many FASTier devices?

Unity configuration

- What is your RAID configuration?
- How many RAID sets do you have?
- How many storage pools do you have? If more than one, are they on different controllers?
- If you have FASTier devices, how have they been assigned (read/write and to which pool)?
- Is LACP and/or jumbo frames enabled on the Unity?
- What is Unity model?
- Are you using any external storage?

Usage

- What do you use the storage for? (CIFS, NFS or block devices)

Network performance

- Does the client have any driver settings that can be changed?
- Is the client operating at the correct link speed?
- Is the client CPU being maxed out?
- Is the client CPU properly load balancing among its cores?
- Is traffic being routed through a slower link before getting to Unity?

  For example, if a 10-G network is on the same subnet as 1-G routing should be examined.

- Are there at least as many incoming client connections as ports in Unity?

  Typically 1 client makes a single connection and is thus limited to a single port's throughput.
- Are there any MTU mismatches?

VMware environment

These questions only apply if you are using storage hosted on VMware.

- Have you enabled VAAI?
- How many VMs are running on the VMware server? Are they all using the storage of a single pool or are they spread across pools?
- What is your expected IOPS?
- What type of applications do the VMs run?

Index

H

I

J

L

M

N

O

P

R

S

T

U

V

W

Nexsan Headquarters

1287 Anvilwood Ave,
Sunnyvale
CA 94089
United States

Nexsan Shipping

302 Enterprise Street , Suite A
Escondido, CA 92029
United States of America

Nexsan Canada

1405 Trans Canada Highway, Suite 300 Dorval, QC
H9P 2V9
Canada

Nexsan UK

Units 33–35, Parker Centre, Mansfield Road Derby,
DE21 4SZ
United Kingdom

Nexsan Unity Documentation & Online Help page:
https://www.nexsan.com/support/support-unity/

Nexsan Unity support:
https://www.nexsan.com/support

Worldwide Web
www.nexsan.com

Document Reference: 20190820PM021111

This product is protected by one or more of the following patents, and other pending patent applications worldwide:

United States patents US8,191,841, US8,120,922;

United Kingdom patents GB2466535B, GB2467622B, GB2467404B, GB2296798B, GB2297636B