Sun ZFS Storage 7000 Analytics Guide

Part No: 821–0782–11
July 2010

100927@24661

Contents

Preface

The *Sun ZFS Storage Analytics Guide* contains conceptual and procedural analytics documentation for Oracle's Sun ZFS Storage 7000 series of NAS appliances.

This documentation is also available while using the appliance Browser User Interface, accessible via the "HELP" button. The appliance documentation may be updated using the System Upgrade procedure documented in this book.

Who Should Use This Book

These notes are for users and system administrators who service and use the Sun ZFS Storage 7000 Appliances.

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party Web sites mentioned in this document. Oracle does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Oracle will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.education.oracle.com`)

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Use the `help` command to show available actions. `Last login: Mon Oct 13 15:43:05 2008 from kiowa`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`caji console login:` **`root`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	To view an individual property, use `get` *propertyname*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

CLI Prompts in Command Examples

The following table shows the default Command Line Interface prompts for the appliance.

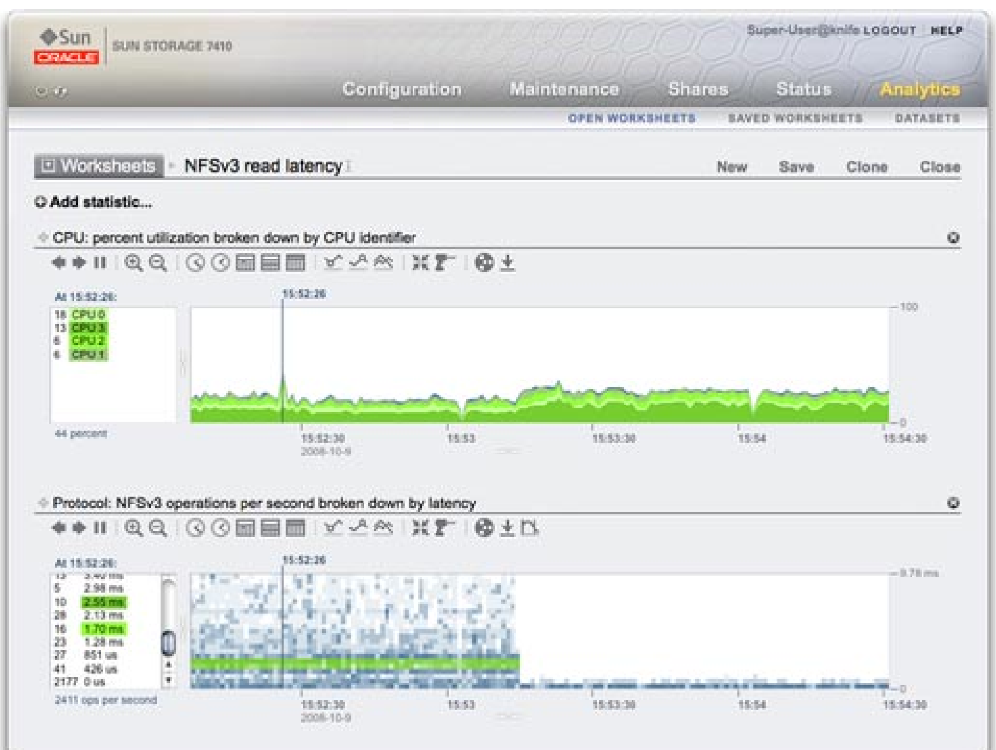
TABLE P–2 CLI Prompts

Type	Prompt
Appliance CLI	`machine_name:>`

CHAPTER 1

Introduction

Analytics



Using analytics to examine CPU utilization and NFSv3 operation latency

Introduction

This appliance is equipped with an advanced DTrace based facility for server analytics. Analytics provides real time graphs of various statistics, which can be saved for later viewing. About a dozen high level statistics are provided, such as NFSv3 operations/sec, which can then be customized to provide lower level details. Groups of viewed statistics can be saved as worksheets for future reference. To learn about the interface for Analytics, see Open Worksheets.

- Concepts - analytics overview
- Statistics - about the available statistics
- Overhead - performance overhead of statistics
- Open Worksheets - the main page for viewing analytics
- Saved Worksheets - saved analytics worksheets
- Datasets - manage analytics statistics

Concepts

Analytics

Analytics is an advanced facility to graph a variety of statistics in real-time and record this data for later viewing. It has been designed for both long term monitoring and short term analysis. When needed, it makes use of DTrace to dynamically create custom statistics, which allows different layers of the operating system stack to be analyzed in detail.

The following topics provide an overview of how Analytics operates, and links to sections with more details.

Drilldown Analysis

Analytics has been designed around an effective performance analysis technique called *drill-down analysis*. This involves checking high level statistics first, and to focus on finer details based on findings so far. This quickly narrows the focus to the most likely areas.

For example, a performance issue may be experienced and the following high level statistics are checked first:

- Network bytes/sec
- NFSv3 operations/sec
- Disk operations/sec
- CPU utilization

Network bytes/sec is found to be at normal levels, and the same for disk operations and CPU utilization. NFSv3 operations/sec is somewhat high, and the type of NFS operation is then checked and found to be of type "read". So far we have drilled down to a statistic which could be named "NFS operations/sec of type read", which we know is higher than usual.

Some systems may have exhausted available statistics at this point, however Analytics can drill down much further. "NFSv3 operations/sec of type read" can then be viewed *by* client - which means, rather than examining a single graph - we can now see separate graphs for each NFS client. (These separate graphs sum to the original statistic that we had.)

Let's say we find that the host "kiowa" is responsible for a majority of the NFS reads. We can use Analytics to drill down further, to see what files this client is reading. Our statistic becomes "NFSv3 operations/sec of type read for client kiowa broken down by filename". From this, we can see that kiowa is reading through every file on the NFS server. Armed with this information, we can ask the owner of kiowa to explain.

The above example is possible in Analytics, which can keep drilling down further if needed. To summarize, the statistics we examined were:

- "NFSv3 operations/sec"
- "NFSv3 operations/sec by type"
- "NFSv3 operations/sec of type read by client"
- "NFSv3 operations/sec of type read for client kiowa broken down by filename"

These match the statistic names as created and viewed in Analytics.

Statistics

In Analytics, the user picks statistics of interest to display on custom worksheets. Statistics available from Analytics include:

- Network device bytes by device and direction
- NFS operations by filename, client, share, type, offset, size and latency
- SMB operations by filename, client, share, type, offset, size and latency
- Disk operations by type, disk, offset, size and latency
- CPU utilization by CPU-id, mode and application

See the Open Workshetes view for listing statistics, and the Preferences view for enabling advanced Analytics - which will make many more statistics available. The Statistics page discusses available statistics in more detail.

Datasets

A *dataset* refers to all existing data for a particular statistic. Datasets contain:

- Statistic data cached in memory due to the statistic being opened or archived.
- Archived statistic data on disk.

Datasets can be managed in the Datasets view.

Actions

The following actions may be performed on statistics/datasets:

Action	Description
Open	Begin reading from the statistic (every second) and cache values in memory as a dataset. In Open Worksheets, statistics are opened when they are added to the view, allowing them to be graphed in real-time. The data is kept in memory while the statistic is being viewed.
Close	Closes the statistic view, discarding the in memory cached dataset.
Archive	Sets the statistic to be permanently opened and archived to disk. If the statistic had already been opened, then all cached data in memory is also archived to disk. Archiving statistics creates permanent datasets, visible in the Datasets view (those with a non-zero "on disk" value). This is how statistics may be recorded 24x7, so that activity from days, weeks and months in the past can be viewed after the fact.
Destroy	Close the statistic, destroy the dataset and delete all archived data from disk.
Suspend	Pause an archived statistic. New data will not be read, but the existing disk archive will be left intact.
Resume	Resumes a previously suspended statistic, so that it will continue reading data and writing to the archive.

Worksheets

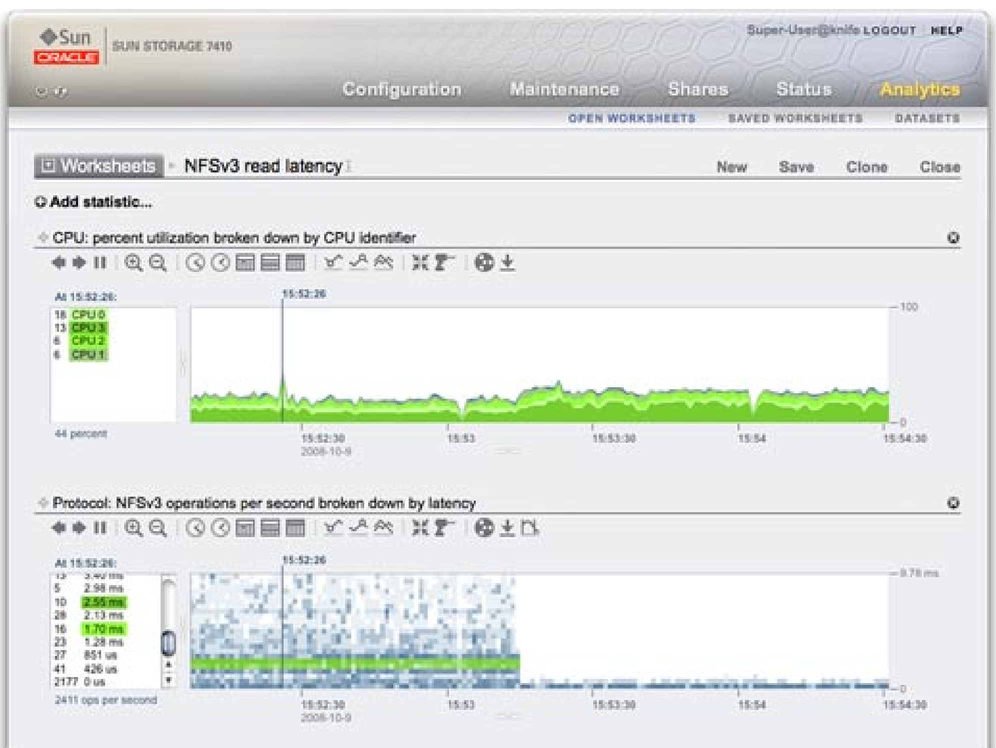
A worksheet is the BUI screen on which statistics are graphed. Multiple statistics can be plotted at the same time, and worksheets may be assigned a title and saved for future viewing. The act of saving a worksheet will automatically execute the archive action on all open statistics - meaning whatever statistics were open, will continue to be read and archived forever.

See the Open Worksheets section for how to drive worksheets, and the Saved Worksheets section for managing previously saved worksheets.

CHAPTER 2

Analytics Interface

Open Worksheets



Using Analytics to examine CPU utilization and NFSv3 operation latency

Worksheets

This is the main interface for Analytics. See Concepts for an overview of Analytics.

A worksheet is a view where multiple statistics may be graphed. The screenshot at the top of this page shows two statistics:

- CPU: percent utilization broken down by CPU identifier - as a *graph*
- Protocol: NFSv3 operations per second broken down by latency - as a *quantize plot*

Click the screenshot for a larger view. The following sections introduce Analytics features based on that screenshot.

Graph

The CPU utilization statistic in the screenshot is rendered as a graph. Graphs provide the following features:

- The left panel lists components of the graph, if available. Since this graph was "... broken down by CPU identifier", the left panel lists CPU identifiers. Only components which had activity in the visible window (or selected time) will be listed on the left.
- Left panel components can be clicked to highlight their data in the main plot window.
- Left panel components can be shift clicked to highlight multiple components at a time (such as in this example, with all four CPU identifiers highlighted).
- Left panel components can be right clicked to show available drilldowns.
- Only ten left panel components will be shown to begin with, followed by "...". You can click the "..." to reveal more. Keep clicking to expand the list completely.
- The graph window on the right can be clicked to highlight a point in time. In the example screenshot, 15:52:26 was selected. Click the pause button followed by the zoom icon to zoom into the selected time. Click the time text to remove the vertical time bar.
- If a point in time is highlighted, the left panel of components will list details for that point in time only. Note that the text above the left box reads "At 15:52:26:", to indicate what the component details are for. If a time wasn't selected, the text would read "Range average:".
- Y-axis auto scales to keep the highest point in the graph (except for utilization statistics, where are fixed at 100%).
- The line graph button will change this graph to plot just lines without the flood-fill. This may be useful for a couple of reasons: some of the finer detail in line plots can be lost in the flood fill, and so selecting line graphs can improve resolution. This feature can also be used to vertical zoom into component graphs: first, select one or more components on the left, then switch to the line graph.

Quantize Plot

The NFS latency statistic in the screenshot is rendered as a quantize plot. The name refers to the how the data is collected and displayed. For each statistic update, data is quantized into buckets, which are drawn as blocks on the plot. The more events in that bucket for that second, the darker the block will be drawn.

The example screenshot shows NFSv3 operations were spread out to 9 ms and beyond - with latency on the y-axis - until an event kicked in about half way and the latency dropped to less than 1 ms. Other statistics can be plotted to explain the drop in latency (the filesystem cache hit rate showed steady misses go to zero at this point - a workload had been randomly reading from disk (0 to 9+ ms latency), and switched to reading files that were cached in DRAM.)

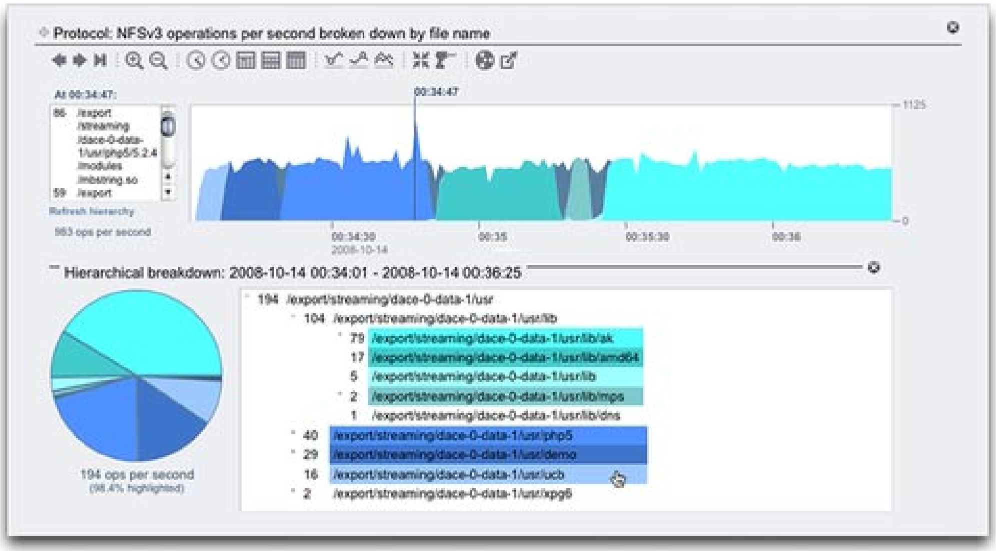
Quantize plots are used for I/O latency, I/O offset and I/O size, and provide the following features:

- Detailed understanding of data profile (not just the average, maximum or minimum) these visualize all events and promote pattern identification.
- Vertical outlier elimination. Without this, the y-axis would always be compressed to include the highest event. Click the crop outliers icon  to toggle between different percentages of outlier elimination. Mouse over this icon to see the current value.
- Vertical zoom: click a low point from the list in the left box, then shift-click a high point. Now click the crop outliers icon to zoom to this range.

Show Hierarchy

Graphs by filename have a special feature - "Show hierarchy" text will be visible on the left. When clicked, a pie-chart and tree view for the traced filenames will be made available.

The following screenshot shows the hierarchy view:



As with graphs, the left panel will show components based on the statistic break down, which in this example was by filename. Filenames can get a little too long for that left panel - try expanding it by clicking and dragging the divider between it and the graph; or use the hierarchy view.

The hierarchy view provides the following features:

- The filesystem may be browsed, by clicking "+" and "-" next to file and directory names.
- File and directory names can be clicked, and their component will shown in the main graph.
- Shift click pathnames to display multiple components at once, as shown in this screenshot.
- The pie chart on the left shows the ratio of each component to the total.
- Slices of the pie may be clicked to perform highlighting.
- If the graph isn't paused, the data will continue to scroll. The hierarchy view can be refreshed to reflect the data visible in the graph by clicking "Refresh hierarchy".

There is a close button on the right to close the hierarchy view.

Common

The following features are common to graphs and quantize plots:

- The height may be expanded. Look for a white line beneath in the middle of the graph, click and drag downwards.
- The width will expand to match the size of your browser.
- Click and drag the move icon ✥ to switch vertical location of the statistics.

Background Patterns

Normally graphs are displayed with various colors against a white background. If data is unavailable for any reason the graph will be filled with a pattern to indicate the specific reason for data unavailability:

- The gray pattern indicates that the given statistic was not being recorded for the time period indicated. This is either because the user had not yet specified the statistic or because data gathering had been explicitly suspended.
- The red pattern indicates that data gathering was unavailable during that period. This is most commonly seen because the system was down during the time period indicated.
- The orange pattern indicates an unexpected failure while gathering the given statistic. This can be caused by a number of aberrant conditions. If it is seen persistently or in critical situations, contact your authorized support resource and/or submit a support bundle.

Saving a Worksheet

Worksheets can be saved for later viewing. As a side effect, all visible statistics will be archived - meaning that they will continue to save new data after the saved worksheet has been closed.

To save a worksheet, click the "Untitled worksheet" text to name it first, then click "Save" from the local navigation bar. Saved worksheets can be opened and managed from the Saved Worksheets section.

Toolbar Reference

A toolbar of buttons is shown above graphed statistics. The following is a reference for their function:

Icon	Click	Shift-Click
	move backwards in time (moves left)	move backwards in time (moves left)
	move forwards in time (moves right)	move forwards in time (moves right)
	forward to now	forward to now
	pause	pause
	zoom out	zoom out

Icon	Click	Shift-Click
	zoom in	zoom in
	show one minute	show two minutes, three, four, ...
	show one hour	show two hours, three, four, ...
	show one day	show two days, three, four, ...
	show one week	show two weeks, three, four, ...
	show one month	show two months, three, four, ...
	show minimum	show next minimum, next next minimum, ...
	show maximum	show next maximum, next next maximum, ...
	show line graph	show line graph
	show mountain graph	show mountain graph
	crop outliers	crop outliers
	sync worksheet to this statistic	sync worksheet to this statistic
	unsync worksheet statistics	unsync worksheet statistics
	drilldown	rainbow highlight
	save statistical data	save statistical data
	export statistical data	export statistical data

Mouse over each button to see a tooltip to describe the click behavior.

CLI

Viewing analytics statistics is possible from the CLI. See:

- Reading Datasets - for listing recent statistics from available datasets.
- Saved Worksheets:CLI - for how to dump worksheets in CSV, which may be suitable for automated scripting.

Tips

- If you'd like to save a worksheet that displays an interesting event, make sure the statistics are paused first (sync all statistics, then hit pause). Otherwise the graphs will continue to scroll, and when you open the worksheet later the event may no longer be on the screen.
- If you are analyzing issues after the fact, you will be restricted to the datasets that were already being archived. Visual correlations can be made between them when the time axis is synchronized. If the same pattern is visible in different statistics - there is a good chance that it is related activity.
- Be patient when zooming out to the month view and longer. Analytics is clever about managing long period data, however there can still be delays when zooming out to long periods.

Tasks

BUI

▼ Monitoring NFSv3 or SMB by operation type

1. **Click the add icon.**
2. **Click the "NFSv3 operations" or "SMB operations" line.**
3. **Click "Broken down by type of operation".**

▼ Monitoring NFSv3 or SMB by latency

1. **Click the add icon.**
2. **Click the "NFSv3 operations" or "SMB operations" line.**
3. **Click "Broken down by latency".**

▼ Monitoring NFSv3 or SMB by filename

1. **Click the add icon.**
2. **Click the "NFSv3 operations" or "SMB operations" line.**
3. **Click "Broken down by filename".**
4. **When enough data is visible, click the "Show hierarchy" text on the left to display a pie-chart and tree-view for the path names that were traced in the graph.**
5. **Click "Refresh hierarchy" when the pie-chart and tree-view become out of date with the scrolling data in the graph.**

▼ Saving a worksheet

1. **Click the "Untitled worksheet" text and type in a custom name**
2. **Click "Save" from the local navigation bar.**

Saved Worksheets

Introduction

Open Worksheets may be saved for at least these reasons:

- To create custom performance views which display statistics of interest.
- To investigate performance events for later analysis. A worksheet may be paused on a particular event and then saved, so that others can open the worksheet later and study the event.

Properties

The following properties are stored for saved worksheets:

Field	Description
Name	Configurable name of the saved worksheet. This will be displayed at the top of the Open Worksheets view

Field	Description
Comment	Optional comment (only visible in the BUI)
Owner	User who owns the worksheet
Created	Time the worksheet was created
Modified	Time the worksheet was last modified (only visible in the CLI)

BUI

Mouse over saved worksheet entries to expose the following controls:

icon	description
	This will upload a support bundle that includes this worksheet, allowing for off-line analysis of your system by your support provider. You should only do this if you have been explicitly asked to upload such a bundle by support personnel.
	Append the datasets saved in this worksheet to the current worksheet in Open Worksheets
	Edit the worksheet to change the name and comment
	Destroy this worksheet

Single click an entry to open that worksheet. This may take several seconds if the worksheet was paused on a time in the distant past, or if it spanned many days, as the appliance must read the statistic data from disk back into memory.

CLI

Worksheet maintenance actions are available under the `analytics worksheets` context. Use the `show` to view the current saved worksheets:

```
walu:> analytics worksheets
walu:analytics worksheets> show
Worksheets:

WORKSHEET       OWNER   NAME
worksheet-000   root    830 MB/s NFSv3 disk
worksheet-001   root    8:27 event
```

Worksheets may be selected so that more details may be viewed. Here one of the statistics is dumped and retrieved in CSV format from the saved worksheet:

```
walu:analytics worksheets> select worksheet-000
walu:analytics worksheet-000> show
Properties:
                          uuid = e268333b-c1f0-401b-97e9-ff7f8ee8dc9b
                          name = 830 MB/s NFSv3 disk
                         owner = root
                         ctime = 2009-9-4 20:04:28
                         mtime = 2009-9-4 20:07:24

Datasets:

DATASET      DATE        SECONDS NAME
dataset-000  2009-9-4         60 nic.kilobytes[device]
dataset-001  2009-9-4         60 io.bytes[op]

walu:analytics worksheet-000> select dataset-000 csv
Time (UTC),KB per second
2009-09-04 20:05:38,840377
2009-09-04 20:05:39,890918
2009-09-04 20:05:40,848037
2009-09-04 20:05:41,851416
2009-09-04 20:05:42,870218
2009-09-04 20:05:43,856288
2009-09-04 20:05:44,872292
2009-09-04 20:05:45,758496
2009-09-04 20:05:46,865732
2009-09-04 20:05:47,881704
[...]
```

If there was a need to gather Analytics statistics using an automated CLI script over SSH, it would be possible to create a saved worksheet containing the desired statistics which could then be read in this fashion. This is one way to view analytics from the CLI; also see Reading datasets.

CHAPTER 3

Statistics and Datasets

Statistics

Introduction

Analytics statistics provide incredible appliance observability, showing how the appliance is behaving and how clients on the network are using it.

Descriptions

While the statistics presented by Analytics may appear straight forward, there may be additional details to be aware of when interpreting their meaning. This is especially true for the purposes of performance analysis, where precise understanding of the statistics is often necessary. The following pages document each of the available statistics and breakdowns:

Analytics

- CPU: Percent utilization *
- Cache: ARC accesses *
- Cache: L2ARC I/O bytes
- Cache: L2ARC accesses
- Data Movement: NDMP bytes transferred to/from disk
- Data Movement: NDMP bytes transferred to/from tape
- Data Movement: Shadow migration bytes
- Data Movement: Shadow migration ops
- Data Movement: Shadow migration requests
- Disk: Disks *
- Disk: I/O bytes *
- Disk: I/O operations *

- Network: Device bytes
- Network: Interface bytes
- Protocol: SMB operations
- Protocol: Fibre Channel bytes
- Protocol: Fibre Channel operations
- Protocol: FTP bytes
- Protocol: HTTP/WebDAV requests
- Protocol: iSCSI bytes
- Protocol: iSCSI operations
- Protocol: NFSv2 operations
- Protocol: NFSv3 operations
- Protocol: NFSv4 operations
- Protocol: SFTP bytes
- Protocol: SRP bytes
- Protocol: SRP operations

** recommended reading*

Advanced Analytics

These statistics are only visible if Advanced Analytics is enabled in Preferences. These are statistics of lesser interest and are not typically needed for system observability. They are often dynamic which can induce higher overhead, and expose more complex areas of the system which require additional expertise to understand properly:

- CPU: CPUs
- CPU: Kernel spins
- Cache: ARC adaptive parameter
- Cache: ARC evicted bytes
- Cache: ARC size
- Cache: ARC target size
- Cache: DNLC accesses
- Cache: DNLC entries
- Cache: L2ARC errors
- Cache: L2ARC size
- Data Movement: NDMP file system operations
- Data Movement: NDMP jobs
- Disk: Percent utilization
- Disk: ZFS DMU operations
- Disk: ZFS logical I/O bytes
- Disk: ZFS logical I/O operations
- Memory: Dynamic memory usage
- Memory: Kernel memory
- Memory: Kernel memory in use
- Memory: Kernel memory lost to fragmentation

- Network: IP bytes
- Network: IP packets
- Network: TCP bytes
- Network: TCP packets
- System: NSCD backend requests
- System: NSCD operations

Default Statistics

For reference, the following are the statistics that are enabled and archived by default on a factory installed appliance. These are the thirty or so statistics you see in the Datasets view when you first configure and login to the appliance:

Category	Statistic
CPU	percent utilization
CPU	percent utilization broken down by CPU mode
Cache	ARC accesses per second broken down by hit/miss
Cache	ARC size
Cache	ARC size broken down by component
Cache	DNLC accesses per second broken down by hit/miss
Cache	L2ARC accesses per second broken down by hit/miss
Cache	L2ARC size
Data Movement	NDMP bytes transferred to/from disk per second
Disk	Disks with utilization of at least 95 percent broken down by disk
Disk	I/O bytes per second
Disk	I/O bytes per second broken down by type of operation
Disk	I/O operations per second
Disk	I/O operations per second broken down by disk
Disk	I/O operations per second broken down by type of operation
Network	device bytes per second
Network	device bytes per second broken down by device
Network	device bytes per second broken down by direction

Category	Statistic
Protocol	SMB operations per second
Protocol	SMB operations per second broken down by type of operation
Protocol	FTP bytes per second
Protocol	Fibre Channel bytes per second
Protocol	Fibre Channel operations per second
Protocol	HTTP/WebDAV requests per second
Protocol	NFSv2 operations per second
Protocol	NFSv2 operations per second broken down by type of operation
Protocol	NFSv3 operations per second
Protocol	NFSv3 operations per second broken down by type of operation
Protocol	NFSv4 operations per second
Protocol	NFSv4 operations per second broken down by type of operation
Protocol	SFTP bytes per second
Protocol	iSCSI operations per second
Protocol	iSCSI bytes per second

These have been chosen to give broad observability across protocols with minimal statistic collection overhead, and are usually left enabled even when benchmarking. For more discussion on statistic overhead, see Overhead.

Tasks

Statistics Tasks

▼ Determining the impact of a dynamic statistic

For this example task we will determine the impact of "Protocol: NFSv3 operations per second broken down by file name":

1. **Go to Open Worksheets.**

2 **Add the statistic: "Protocol: NFSv3 operations per second as a raw statistic". This is a static statistic and will have negligible performance impact.**

3 **Create steady NFSv3 load; or wait for a period of steady load.**

4 **Add the statistic: "Protocol: NFSv3 operations per second broken down by filename". As this statistic is being created, you may see a temporary sharp dip in performance.**

5 **Wait at least 60 seconds.**

6 **Close the by-filename statistic by clicking on the close icon.**

7 **Wait another 60 seconds.**

8 **Now examine the "Protocol: NFSv3 operations per second as a raw statistic" graph by pausing and zooming out to cover the previous few minutes. Was there a drop in performance when the by-filename statistic was enabled? If the graph looks erratic, try this process again - or try this with a workload that is more steady.**

9 **Click on the graph to see the values at various points, and calculate the percentage impact of that statistic.**

CPU Percent utilization

CPU: Percent Utilization

This shows the average utilization of the appliance CPUs. A CPU may be a core on a socket or a hardware thread; the number and type can be seen under Hardware. For example, a system may have four sockets of quad-core CPUs, meaning there are 16 CPUs available to the appliance. The utilization shown by this statistic is the average across all CPUs.

The appliance CPUs can reach 100% utilization, which may or may not be a problem. For some performance tests the appliance is deliberately driven to 100% CPU utilization to measure it at peak performance.

Example

This example shows CPU: Percent utilization broken down by CPU mode, while the appliance served over 2 Gbytes/sec of cached data over NFSv3:

image

An average of 82% utilization suggests that there could be more headroom available, and that appliance may be able to serve more than 2 Gbytes/sec (it can). (The breakdowns only add to 81%; the extra 1% is due to rounding.)

The high level of CPU utilization does mean that overall latency of NFS operations may increase, which can be measured by Protocol: NFSv3 operations broken down by latency, as operations may be waiting for CPU resources more often.

When to check

When searching for system bottlenecks. This may also be checked when enabling features that consume CPU, such as compression, to gauge the CPU cost of that feature.

Breakdowns

Available breakdowns of CPU Percent utilization:

Breakdown	Description
CPU mode	Either user or kernel. See the CPU modes table below.
CPU identifier	Numeric operating system identifier of the CPU.
application name	Name of the application which is on-CPU.
process identifier	Operating system process ID (PID).
user name	Name of the user who owns the process or thread which is consuming CPU.

The CPU modes are:

CPU mode	Description
user	This is a user-land process. The most common user-land process consuming CPU is akd, the appliance kit daemon, which provides administrative control of the appliance.
kernel	This is a kernel-based thread which is consuming CPU. Many of the appliance services are kernel-based, such as NFS and SMB.

Further Analysis

A problem with this CPU utilization average is that it can hide issues when a single CPU is at 100% utilization, which may happen if a single software thread is saturated with work. Use the Advanced Analytic CPU: CPUs broken down by percent utilization, which represents utilization as a heat map of CPUs, allowing a single CPU at 100% to be easily identified.

Details

CPU utilization represents the time spent processing CPU instructions in user and kernel code, that are not part of the idle thread. Instruction time includes stall cycles on the memory bus, so high utilization can be caused by the I/O movement of data.

Cache ARC accesses

Cache: ARC accesses

The ARC is the Adaptive Replacement Cache, and is an in-DRAM cache for filesystem and volume data. This statistic shows accesses to the ARC, and allows its usage and performance to be observed.

When to check

When investigating performance issues, to check how well the current workload is caching in the ARC.

Breakdowns

Available breakdowns of Cache ARC accesses are:

Breakdown	Description
hit/miss	The result of the ARC lookup. hit/miss states are described in the table below.
file name	The file name that was requested from the ARC. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
L2ARC eligibility	This is the eligibility of L2ARC caching, as measured at the time of ARC access. A high level of ARC misses which are L2ARC eligible would suggest that the workload would benefit from 2nd level cache devices.
project	This shows the project which is accessing the ARC.
share	This shows the share which is accessing the ARC.

As described in Overhead, breakdown such as by file name would be the most expensive to leave enabled.

The hit/miss states are:

hit/miss breakdown	Description
data hits	A data block was in the ARC DRAM cache and returned.
data misses	A data block was not in the ARC DRAM cache. It will be read from the L2ARC cache devices (if available and the data is cached on them) or the pool disks.
metadata hits	A metadata block was in the ARC DRAM cache and returned. Metadata includes the on-disk filesystem framework which refers to the data blocks. Other examples are listed below.
metadata misses	A metadata block was not in the ARC DRAM cache. It will be read from the L2ARC cache devices (if available and the data is cached on them) or the pool disks.
prefetched data/metadata hits/misses	ARC accesses triggered by the prefetch mechanism, not directly from an application request. More details on prefetch follow.

Details

Metadata

Examples of metadata:

- Filesystem block pointers
- Directory information
- Data deduplication tables
- ZFS uberblock

Prefetch

Prefetch is a mechanism to improve the performance of streaming read workloads. It examines I/O activity to identify sequential reads, and can issue extra reads ahead of time so that the data can be in cache before the application requests it. Prefetch occurs *before the ARC* by performing accesses to the ARC - bear this in mind when trying to understand prefetch ARC activity. For example, if you see:

Type	Description
prefetched data missess	prefetch identified a sequential workload, and requested that the data be cached in the ARC ahead of time by performing ARC accesses for that data. The data was not in the cache already, and so this is a "miss" and the data is read from disk. This is normal, and is how prefetch populates the ARC from disk.
prefeteched data hits	prefetch identified a sequential workload, and requested that the data be cached in the ARC ahead of time by performing ARC accesses for that data. As it turned out, the data was already in the ARC - so these accesses returned as "hits" (and so the prefetch ARC access wasn't actually needed). This happens if cached data is repeatedly read in a sequential manner.

After data has been prefetched, the application may then request it with its own ARC accesses. Note that the sizes may be different: prefetch may occur with a 128 Kbyte I/O size, while the application may be reading with an 8 Kbyte I/O size. For example, the following doesn't appear directly related:

- data hits: 368
- prefetch data misses: 23

However it may be: if prefetch was requesting with a 128 KByte I/O size, 23 x 128 = 2944 Kbytes. And if the application was requesting with an 8 Kbyte I/O size, 368 x 8 = 2944 Kbytes.

Further Analysis

To investigate ARC misses, check that the ARC has grown to use available DRAM using Cache: ARC size.

Cache L2ARC IO bytes

Cache: L2ARC I/O bytes

The L2ARC is the 2nd Level Adaptive Replacement Cache, and is an SSD based cache that is accessed before reading from the much slower pool disks. The L2ARC is currently intended for random read workloads. This statistic shows the read and write byte rates to the L2ARC cache devices, if cache devices are present.

When to check

This can be useful to check during warmup. The write bytes will show the rate of L2ARC warmup of time.

Breakdowns

Breakdown	Description
type of operation	read or write. Read bytes are hits on the cache devices. Write bytes show the cache devices populating with data.

Further Analysis

Also see Cache: L2ARC accesses.

Cache L2ARC accesses

Cache: L2ARC accesses

The L2ARC is the 2nd Level Adaptive Replacement Cache, and is an SSD based cache that is accessed before reading from the much slower pool disks. The L2ARC is currently intended for random read workloads. This statistic shows L2ARC accesses if L2ARC cache devices are present, allowing its usage and performance to be observed.

When to check

When investigating performance issues, to check how well the current workload is caching in the L2ARC.

Breakdowns

Breakdown	Description
hit/miss	The result of the L2ARC lookup. hit/miss states are described in the table below.
file name	The file name that was requested from the L2ARC. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
L2ARC eligibility	This is the eligibility of L2ARC caching, as measured at the time of L2ARC access.
project	This shows the project which is accessing the L2ARC.
share	This shows the share which is accessing the L2ARC.

As described in Overhead, breakdown such as by file name would be the most expensive to leave enabled.

Further Analysis

To investigate L2ARC misses, check that the L2ARC has grown enough in size using the Advanced Analytic Cache: L2ARC size. The L2ARC typically takes hours, if not days, to warm up hundreds of Gbytes when feeding from small random reads. The rate can also be checked by examining writes from Cache: L2ARC I/O bytes. Also check the Advanced Analytic Cache: L2ARC errors to see if there are any errors preventing the L2ARC from warming up.

Cache: ARC accesses by L2ARC eligibility can also be checked to see if the data is eligible for L2ARC caching in the first place. Since the L2ARC is intended for random read workloads, it will ignore sequential or streaming read workloads, allowing them to be returned from the pool disks instead.

Data Movement NDMP bytes transferred tofrom disk

Data Movement: NDMP bytes transferred to/from disk

This statistic shows total NDMP bytes transferred per second to or from the local pool disks. It will indicate how much data is being read or written for NDMP backups. This statistic will be zero unless NDMP is configured and active.

When to check

When investigating NDMP backup performance. This can also be checked when trying to identify an unknown disk load, some of which may be caused by NDMP.

Breakdowns

Breakdown	Description
type of operation	read or write.

Further Analysis

Also see Data Movement: NDMP bytes transferred to/from tape.

Data Movement NDMP bytes transferred tofrom tape

Data Movement: NDMP bytes transferred to/from tape

This statistic shows total NDMP bytes per second transferred to or from attached tape devices. This statistic will be zero unless NDMP is configured and active.

When to check

When investigating NDMP backup performance.

Breakdowns

Breakdown	Description
type of operation	read or write.

Further Analysis

Also see Data Movement: NDMP bytes transferred to/from disk.

Data Movement Shadow migration bytes

Data Movement: Shadow migration bytes

This statistic tracks total Shadow Migration bytes per second transferred as part of migrating file or directory contents. This does not apply to metadata (extended attributes, ACLs, etc). It gives a rough approximation of the data transferred, but source datasets with a large amount of metadata will show a disproportionally small bandwidth. The complete bandwidth can be observed by looking at network analytics.

When to check

When investigating Shadow Migration activity.

Breakdowns

Breakdown	Description
file name	The file name that was migrated. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
project	This shows the project which contains a shadow migration.
share	This shows the share which is being migrated.

Further Analysis

Also see Data Movement: Shadow migration ops and Data Movement: Shadow migration requests.

Data Movement Shadow migration ops

Data Movement: Shadow migration ops

This statistic tracks Shadow Migration operations that require going to the source filesystem.

When to check

When investigating Shadow Migration activity.

Breakdowns

Breakdown	Description
file name	The file name that was migrated. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
project	This shows the project which contains a shadow migration.
share	This shows the share which is being migrated.
latency	Measure the latency of requests from the shadow migration source.

Further Analysis

Also see Data Movement: Shadow migration bytes and Data Movement: Shadow migration requests.

Data Movement Shadow migration requests

Data Movement: Shadow migration requests

This statistic tracks Shadow Migration requests for files or directories that are not cached and known to be local to the filesystem. It does account for both migrated and unmigrated files and directories, and can be used to track the latency incurred as part of shadow migration, as well as track the progress of background migration. It currently encompasses both synchronous and asynchronous (background) migration, so it's not possible to view only latency visible to clients.

When to check

When investigating Shadow Migration activity.

Breakdowns

Breakdown	Description
file name	The file name that was migrated. Using this breakdown allows hierarchy mode to be used, so that filesystem directories can be navigated.
project	This shows the project which contains a shadow migration.
share	This shows the share which is being migrated.
latency	Measure the latency incurred as part of shadow migration.

Further Analysis

Also see Data Movement: Shadow migration ops and Data Movement: Shadow migration bytes.

Disk Disks

Disk: Disks

The Disks statistic is used to display the heat map for disks broken down by percent utilization. This is the best way to identify when pool disks are under heavy load. It may also identify problem disks that are beginning to perform poorly, before their behavior triggers a fault and automatic removal from the pool.

When to check

Any investigation into disk performance.

Breakdowns

Breakdown	Description
percent utilization	A heat map with utilization on the Y-axis and each level on the Y-axis colored by the number of disks at that utilization: from light (none) to dark (many).

Interpretation

Utilization is a better measure of disk load than IOPS or throughput. Utilization is measured as the time during which that disk was busy performing requests (see Details below). At 100% utilization the disk may not be able to accept more requests, and additional I/O may wait on a queue. This I/O wait time will cause latency to increase and overall performance to decrease.

In practise, disks with a consistant Utilization of 75% or higher are an indication of heavy disk load.

The heat map allows a particular pathology to be easily identified: a single disk misperforming and reaching 100% utilization (a bad disk). Disks can exhibit this symptom before they fail. Once disks fail, they are automatically removed from the pool with a corresponding alert. This particular problem is during the time *before* they fail, when their I/O latency is increasing and slowing down overall appliance performance, but their status is considered healthy - they have yet to identify any error state. This situation will be seen as a feint line at the top of the heat map, showing that a single disk has stayed at 100% utilization for some time.

Suggested interpretation summary:

Observed	Suggested Interpretation
Most disks consistently over 75%	Available disk resources are being exhausted.
Single disk at 100% for several seconds	This can indicate a bad disk that is about to fail.

Further Analysis

To understand the effect of busy disks on I/O, see Disk: I/O operations broken down by latency. For understanding the nature of the I/O, such as IOPS, throughput, I/O sizes and offsets, use Disk: I/O operations and Disk: I/O bytes.

Details

This statistic is actually a measure of percent busy, which serves as a reasonable approximation of percent utilization since the appliance manages the disks directly. Technically this isn't a direct measure of disk utilization: at 100% busy, a disk may be able to accept more requests which it serves concurrently by inserting into and reordering its command queue, or serves from its on-disk cache.

Disk IO bytes

Disk: I/O bytes

This statistic shows the back-end throughput to the disks. This is after the appliance has processed logical I/O into physical I/O based on share settings, and after software RAID as configured by Storage.

For example, an 8 Kbyte write over NFSv3 may became a 128 Kbyte write after the record size is applied from the share settings, which may then become a 256 Kbyte write to the disks after mirroring is applied, plus additional bytes for filesystem metadata. On the same mirrored environment, an 8 Kbyte NFSv3 read may become a 128 Kbyte disk read after the record size is applied, however this doesn't get doubled by mirroring (the data only needs to be read from one half.) It can help to monitor throughput at all layers at the same time to examine this behavior, for example by viewing:

- Network: device bytes - data rate on the network (logical)
- Disk: ZFS logical I/O bytes - data rate to the share (logical)
- Disk: I/O bytes - data rate to the disks (physical)

When to check

To understand the nature of back-end disk I/O, after an issue has already been determined based on disk utilization or latency. It is difficult to identify an issue from disk I/O throughput alone: a single disk may be performing well at 50 Mbytes/sec (sequential I/O), yet poorly at 5 Mbytes/sec (random I/O.)

Using the disk breakdown and the hierarchy view can be used to determine if the JBODs are balanced with disk I/O throughput. Note that cache and log devices will usually have a different throughput profile to the pool disks, and can often stand out as the highest throughput disks when examining by-disk throughput.

Breakdowns

Breakdown	Description
type of operation	read or write.
disk	pool or system disk. This breakdown can identify system disk I/O vs pool disk I/O, and I/O to cache and log devices.

Further Analysis

See Disk: Disks broken down by percent utilization for the best measure of disk utilization. Disk: I/O operations can also be used to examine operations/sec instead of bytes/sec.

Disk IO operations

Disk: I/O operations

This statistic shows the back-end I/O to the disks (disk IOPS). This is after the appliance has processed logical I/O into physical I/O based on share settings, and after software RAID as configured by Storage.

For example, 16 sequential 8 Kbyte NFSv3 writes may become a single 128 Kbyte write sometime later after the data has been buffered in the ARC DRAM cache, which may then become multiple disk writes due to RAID - such as two writes to each half of a mirror. It can help to monitor I/O at all layers at the same time to examine this behavior, for example by viewing:

- Protocol: NFSv3 operations - NFSv3 writes (logical)
- Disk: ZFS logical I/O operations - share I/O (logical)
- Disk: I/O operations - I/O to the disks (physical)

This statistic includes a breakdown of disk I/O latency, which is a direct measure of performance for synchronous I/O, and also useful as a measure of the magnitude of back-end disk load. It is difficult to identify issues from disk IOPS alone without considering latency: a single disk may be performing well at 400 IOPS (sequential and small I/O hitting mostly from the disk's on-board DRAM cache), yet poorly at 110 IOPS (random I/O causing head seek and waiting on disk rotation.)

When to check

Whenever disk performance is investigated, using:

- Disk: I/O operations broken down by latency

This is presented as a heat map allowing the pattern of I/O latency to be observed, and outliers to be easily identified (click the outlier elimination button to view more). Disk I/O latency is often related to the performance of the delivered logical I/O, such as with synchronous reads (non-prefetch), and synchronous writes. There are situations where the latency is not directly related to logical I/O performance, such as asynchronous writes being flushed sometime later to disk, and for prefetch reads.

After an issue has already been determined based on disk I/O latency or utilization, the nature of the disk I/O can be investigated using the other breakdowns, which show disk I/O counts (IOPS). There are no useful IOPS limits per-disk that can be discussed, as such a limit depends on the type of IOPS (random or sequential) and I/O size (large or small). Both of these attributes can be observed using the breakdowns:

- Disk: I/O operations broken down by offset
- Disk: I/O operations broken down by size

Using the disk breakdown and the hierarchy view can also be used to determine if the JBODs are balanced with disk IOPS. Note that cache and log devices will usually have a different I/O profile to the pool disks, and can often stand out as the highest IOPS disks when examining by-disk I/O.

Breakdowns

Breakdown	Description
type of operation	read or write.
disk	pool or system disk. This can be useful to identify system disk I/O vs pool disk I/O, and I/O to cache and log devices.
size	a heat map showing the distribution of I/O sizes.
latency	a heat map showing the latency of disk I/O, as measured from when the I/O was requested to the disk to when the disk returned the completion.
offset	a heat map showing the disk location offset of disk I/O. This can be used to identify random or sequential disk IOPS (often best by vertically zooming the heat map to make out details.)

Further Analysis

See Disk: Disks broken down by percent utilization for the best measure of disk utilization. Disk: I/O bytes can also be used to examine bytes/sec instead of operations/sec.

Network Device bytes

Network: Device bytes

This statistic measures network device activity in bytes/sec. Network devices are the physical network ports, and are shown in the Device column of Network. The measured bytes by this stastistic includes all network payload headers (Ethernet, IP, TCP, NFS/SMB/etc.)

When to check

Network bytes can be used a rough measure of appliance load. It should also be checked whenever performance issues are investigated, especially for 1 Gbit/sec interfaces, in case the bottleneck is the network device. The maximum practical throughput for network devices in each direction (in or out) based on speed:

- 1 Gbit/sec Ethernet: ~120 Mbytes/sec device bytes
- 10 Gbit/sec Ethernet: ~1.16 Gbytes/sec device bytes

If a network device shows a higher rate than these, use the direction breakdown to see the inbound and outbound components.

Breakdowns

Breakdown	Description
direction	in or out, relative to the appliance. For example, NFS reads to the appliance would be show as out(bound) network bytes.
device	network device (see Devices in Network).

Further Analysis

Also see Network: Interface bytes for network throughput at the interface level, instead of the device level.

Network Interface bytes

Network: Interface bytes

This statistic measures network interface activity in bytes/sec. Network interfaces are the logical network interfaces, and are shown in the Interface column of Network. The measured bytes by this stastistic includes all network payload headers (Ethernet, IP, TCP, NFS/SMB/etc.)

Example

See Network: Device bytes for an example of a similar statistic with similar breakdowns.

When to check

Network bytes can be used a rough measure of appliance load. This statistic can be used to see the rate of network bytes through different interfaces. To examine network devices that make up an interface, especially to identify if there are balancing problems with LACP aggregations, use the Network Device bytes statistic.

Breakdowns

Breakdown	Description
direction	in or out, relative to the appliance. For example, NFS reads to the appliance would be show as out(bound) network bytes.
interface	network interface (see Interfaces in Network).

Further Analysis

Also see Network: Device bytes for network throughput at the device level, instead of the interface level.

Protocol SMB operations

Protocol: SMB operations

This statistic shows SMB operations/sec (SMB IOPS) requested by clients to the appliance. Various useful breakdowns are available: to show the client, filename and latency of the SMB I/O.

Example

See Protocol: NFSv3 operations for an example of a similar statistic with similar breakdowns.

When to check

SMB operations/sec can be used as an indication of SMB load, and can be viewed on the dashboard.

Use the latency breakdown when investigating SMB performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the SMB latency is high, drill down further on latency to identify the type of operation and filename for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client and filename breakdowns, and the filename hierarchy view. It's best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on a busy production server.

Breakdowns

Breakdown	Description
type of operation	SMB operation type (read/write/readX/writeX/...)
client	remote hostname or IP address of the SMB client.
filename	filename for the SMB I/O, if known and cached by the appliance. If the filename is not known it is reported as "<unknown>".
share	the share for this SMB I/O.
project	the project for this SMB I/O.
latency	a heat map showing the latency of SMB I/O, as measured from when the SMB request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the SMB request, and to perform any disk I/O.
size	a heat map showing the distribution of SMB I/O sizes.
offset	a heat map showing the file offset of SMB I/O. This can be used to identify random or sequential SMB IOPS. Use the Disk I/O operations statistic to check whether random SMB IOPS maps to random Disk IOPS after the filesystem and RAID configuration has been applied.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: SMB operations per second of type read broken down by latency" (to examine latency for reads only)
- "Protocol: SMB operations per second for file '/export/fs4/10ga' broken down by offset" (to examine file access pattern for a particular file)
- "Protocol: SMB operations per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by the SMB activity; Cache: ARC accesses broken down by hit/miss to see how well an SMB read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol Fibre Channel bytes

Protocol: Fibre Channel bytes

This statistic shows Fibre Channel bytes/sec requested by initiators to the appliance.

Example

See Protocol: iSCSI bytes for an example of a similar statistic with similar breakdowns.

When to check

Fibre Channel bytes/sec can be used as an indication of FC load, in terms of throughput. For a deeper analysis of FC activity, see Protocol: Fibre Channel operations.

Breakdowns

Breakdown	Description
initiator	Fibre Channel client initiator
target	local SCSI target
project	the project for this FC request.
lun	the LUN for this FC request.

See the SAN page for terminology definitions.

Further Analysis

See Protocol: Fibre Channel operations for numerous other breakdowns on FC operations; also see Cache: ARC accesses broken down by hit/miss to see how well an FC read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol Fibre Channel operations

Protocol: Fibre Channel operations

This statistic shows Fibre Channel operations/sec (FC IOPS) requested by initiators to the appliance. Various useful breakdowns are available: to show the initiator, target, type and latency of the FC I/O.

Example

See Protocol: iSCSI operations for an example of a similar statistic with similar breakdowns.

When to check

Fibre Channel operations/sec can be used as an indication of FC load, and can also be viewed on the dashboard.

Use the latency breakdown when investigating FC performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the FC latency is high, drill down further on latency to identify the client initiator, the type of operation and LUN for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client initiator are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client initiator, lun and command breakdowns.

Breakdowns

Breakdown	Description
initiator	Fibre Channel client initiator
target	local SCSI target
project	the project for this FC request.
lun	the LUN for this FC request.
type of operation	FC operation type. This shows how the SCSI command is transported by the FC protocol, which can give an idea to the nature of the I/O.
command	SCSI command sent by the FC protocol. This can show the real nature of the requested I/O (read/write/sync-cache/...).
latency	a heat map showing the latency of FC I/O, as measured from when the FC request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the FC request, and to perform any disk I/O.
offset	a heat map showing the file offset of FC I/O. This can be used to identify random or sequential FC IOPS. Use the Disk I/O operations statistic to check whether random FC IOPS maps to random Disk IOPS after the LUN and RAID configuration has been applied.
size	a heat map showing the distribution of FC I/O sizes.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: Fibre Channel operations per second of command read broken down by latency" (to examine latency for SCSI reads only)

Further Analysis

See Protocol: Fibre Channel bytes for the throughput of this FC I/O; also see Cache: ARC accesses broken down by hit/miss to see how well an FC read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol FTP bytes

Protocol: FTP bytes

This statistic shows FTP bytes/sec requested by clients to the appliance. Various useful breakdowns are available: to show the client, user and filename of the FTP requests.

Example

FTP

When to check

FTP bytes/sec can be used as an indication of FTP load, and can be viewed on the dashboard.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client, user and filename breakdowns, and the filename hierarchy view. It may be best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on appliances with high rates of FTP activity.

Breakdowns

Breakdown	Description
type of operation	FTP operation type (get/put/...)
user	username of the client
filename	filename for the FTP operation, if known and cached by the appliance. If the filename is not known it is reported as "<unknown>".
share	the share for this FTP request.

Breakdown	Description
project	the project for this FTP request.
client	remote hostname or IP address of the FTP client.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: FTP bytes per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Cache: ARC accesses broken down by hit/miss to see how well an FTP read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol HTTPWebDAV requests

Protocol: HTTP/WebDAV requests

This statistic shows HTTP/WebDAV requests/sec requested by HTTP clients. Various useful breakdowns are available: to show the client, filename and latency of the HTTP request.

When to check

HTTP/WebDAV requests/sec can be used as an indication of HTTP load, and can also be viewed on the dashboard.

Use the latency breakdown when investigating HTTP performance issues, especially to quantify the magnitude of the issue. This measures the latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the HTTP latency is high, drill down further on latency to identify the file, size and response code for the high latency HTTP requests, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client initiator are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client, response code and requested filename breakdowns.

Breakdowns

Breakdown	Description
type of operation	HTTP request type (get/post)
response code	HTTP response (200/404/...)
client	client hostname or IP address
filename	filename requested by HTTP
latency	a heat map showing the latency of HTTP requests, as measured from when the HTTP request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the HTTP request, and to perform any disk I/O.
size	a heat map showing the distribution of HTTP request sizes.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: HTTP/WebDAV operations per second of type get broken down by latency" (to examine latency for HTTP GETs only)
- "Protocol: HTTP/WebDAV requests per second for response code '404' broken down by file name (to see which non-existant files were requested)
- "Protocol: HTTP/WebDAV requests per second for client 'deimos.sf.fishpong.com' broken down by file name" (to examine files requested by a particular client)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by HTTP activity; also see Cache: ARC accesses broken down by hit/miss to see how well an HTTP read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol iSCSI bytes

Protocol: iSCSI bytes

This statistic shows iSCSI bytes/sec requested by initiators to the appliance.

When to check

iSCSI bytes/sec can be used as an indication of iSCSI load, in terms of throughput. For a deeper analysis of iSCSI activity, see Protocol: iSCSI operations.

Breakdowns

Breakdown	Description
initiator	iSCSI client initiator
target	local SCSI target
project	the project for this iSCSI request.
lun	the LUN for this iSCSI request.
client	the remote iSCSI client hostname or IP address

See the SAN page for terminology definitions.

Further Analysis

See Protocol: iSCSI operations for numerous other breakdowns on iSCSI operations; also see Cache: ARC accesses broken down by hit/miss to see how well an iSCSI read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol iSCSI operations

Protocol: iSCSI operations

This statistic shows iSCSI operations/sec (iSCSI IOPS) requested by initiators to the appliance. Various useful breakdowns are available: to show the initiator, target, type and latency of the iSCSI I/O.

When to check

iSCSI operations/sec can be used as an indication of iSCSI load, and can also be viewed on the dashboard.

Use the latency breakdown when investigating iSCSI performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the iSCSI latency is high, drill down further on latency to identify the client initiator, the type of operation and LUN for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client initiator are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client initiator, lun and command breakdowns.

Breakdowns

Breakdown	Description
initiator	iSCSI client initiator
target	local SCSI target
project	the project for this iSCSI request.
lun	the LUN for this iSCSI request.
type of operation	iSCSI operation type. This shows how the SCSI command is transported by the iSCSI protocol, which can give an idea to the nature of the I/O.
command	SCSI command sent by the iSCSI protocol. This can show the real nature of the requested I/O (read/write/sync-cache/...).
latency	a heat map showing the latency of iSCSI I/O, as measured from when the iSCSI request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the iSCSI request, and to perform any disk I/O.
offset	a heat map showing the file offset of iSCSI I/O. This can be used to identify random or sequential iSCSI IOPS. Use the Disk I/O operations statistic to check whether random iSCSI IOPS maps to random Disk IOPS after the LUN and RAID configuration has been applied.
size	a heat map showing the distribution of iSCSI I/O sizes.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: iSCSI operations per second of command read broken down by latency" (to examine latency for SCSI reads only)

Further Analysis

See Protocol: iSCSI bytes for the throughput of this iSCSI I/O; also see Cache: ARC accesses broken down by hit/miss to see how well an iSCSI read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol NFSv2 operations

Protocol: NFSv2 operations

This statistic shows NFSv2 operations/sec (NFS IOPS) requested by clients to the appliance. Various useful breakdowns are available: to show the client, filename and latency of the NFS I/O.

Example

See Protocol: NFSv3 operations for an example of a similar statistic with similar breakdowns.

When to check

NFS operations/sec can be used as an indication of NFS load, and can be viewed on the dashboard.

Use the latency breakdown when investigating NFS performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the NFS latency is high, drill down further on latency to identify the type of operation and filename for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client and filename breakdowns, and the filename hierarchy view. It's best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on a busy production server.

Breakdowns

Breakdown	Description
type of operation	NFS operation type (read/write/getattr/setattr/lookup/...)
client	remote hostname or IP address of the NFS client.
filename	filename for the NFS I/O, if known and cached by the appliance. There are some circumstances where the filename is not known, such as after a cluster failover and when clients continue to operate on NFS filehandles without issuing an open to identify the filename; in these situations the filename reported is "<unknown>".

Breakdown	Description
share	the share for this NFS I/O.
project	the project for this NFS I/O.
latency	a heat map showing the latency of NFS I/O, as measured from when the NFS request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the NFS request, and to perform any disk I/O.
size	a heat map showing the distribution of NFS I/O sizes.
offset	a heat map showing the file offset of NFS I/O. This can be used to identify random or sequential NFS IOPS. Use the Disk I/O operations statistic to check whether random NFS IOPS maps to random Disk IOPS after the filesystem and RAID configuration has been applied.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: NFSv2 operations per second of type read broken down by latency" (to examine latency for reads only)
- "Protocol: NFSv2 operations per second for file '/export/fs4/10ga' broken down by offset" (to examine file access pattern for a particular file)
- "Protocol: NFSv2 operations per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by the NFS activity; Cache: ARC accesses broken down by hit/miss to see how well an NFS read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol NFSv3 operations

Protocol: NFSv3 operations

This statistic shows NFSv3 operations/sec (NFS IOPS) requested by clients to the appliance. Various useful breakdowns are available: to show the client, filename and latency of the NFS I/O.

When to check

NFS operations/sec can be used as an indication of NFS load, and can be viewed on the dashboard.

Use the latency breakdown when investigating NFS performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the NFS latency is high, drill down further on latency to identify the type of operation and filename for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client and filename breakdowns, and the filename hierarchy view. It's best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on a busy production server.

Breakdowns

Breakdown	Description
type of operation	NFS operation type (read/write/getattr/setattr/lookup/...)
client	remote hostname or IP address of the NFS client.
filename	filename for the NFS I/O, if known and cached by the appliance. There are some circumstances where the filename is not known, such as after a cluster failover and when clients continue to operate on NFS filehandles without issuing an open to identify the filename; in these situations the filename reported is "<unknown>".
share	the share for this NFS I/O.
project	the project for this NFS I/O.
latency	a heat map showing the latency of NFS I/O, as measured from when the NFS request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the NFS request, and to perform any disk I/O.
size	a heat map showing the distribution of NFS I/O sizes.
offset	a heat map showing the file offset of NFS I/O. This can be used to identify random or sequential NFS IOPS. Use the Disk I/O operations statistic to check whether random NFS IOPS maps to random Disk IOPS after the filesystem and RAID configuration has been applied.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: NFSv3 operations per second of type read broken down by latency" (to examine latency for reads only)

- "Protocol: NFSv3 operations per second for file '/export/fs4/10ga' broken down by offset" (to examine file access pattern for a particular file)
- "Protocol: NFSv3 operations per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by the NFS activity; Cache: ARC accesses broken down by hit/miss to see how well an NFS read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol NFSv4 operations

Protocol: NFSv4 operations

This statistic shows NFSv4 operations/sec (NFS IOPS) requested by clients to the appliance. Various useful breakdowns are available: to show the client, filename and latency of the NFS I/O.

Example

See Protocol: NFSv3 operations for an example of a similar statistic with similar breakdowns.

When to check

NFS operations/sec can be used as an indication of NFS load, and can be viewed on the dashboard.

Use the latency breakdown when investigating NFS performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the NFS latency is high, drill down further on latency to identify the type of operation and filename for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client and filename breakdowns, and the filename hierarchy view. It's best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on a busy production server.

Breakdowns

Breakdown	Description
type of operation	NFS operation type (read/write/getattr/setattr/lookup/...)
client	remote hostname or IP address of the NFS client.
filename	filename for the NFS I/O, if known and cached by the appliance. There are some circumstances where the filename is not known, such as after a cluster failover and when clients continue to operate on NFS filehandles without issuing an open to identify the filename; in these situations the filename reported is "<unknown>".
share	the share for this NFS I/O.
project	the project for this NFS I/O.
latency	a heat map showing the latency of NFS I/O, as measured from when the NFS request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the NFS request, and to perform any disk I/O.
size	a heat map showing the distribution of NFS I/O sizes.
offset	a heat map showing the file offset of NFS I/O. This can be used to identify random or sequential NFS IOPS. Use the Disk I/O operations statistic to check whether random NFS IOPS maps to random Disk IOPS after the filesystem and RAID configuration has been applied.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: NFSv4 operations per second of type read broken down by latency" (to examine latency for reads only)
- "Protocol: NFSv4 operations per second for file '/export/fs4/10ga' broken down by offset" (to examine file access pattern for a particular file)
- "Protocol: NFSv4 operations per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Network: Device bytes for a measure of network throughput caused by the NFS activity; Cache: ARC accesses broken down by hit/miss to see how well an NFS read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Protocol SFTP bytes

Protocol: SFTP bytes

This statistic shows SFTP bytes/sec requested by clients to the appliance. Various useful breakdowns are available: to show the client, user and filename of the SFTP requests.

Example

See Protocol: FTP bytes for an example of a similar statistic with similar breakdowns.

When to check

SFTP bytes/sec can be used as an indication of SFTP load, and can be viewed on the dashboard.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client, user and filename breakdowns, and the filename hierarchy view. It may be best to enable these breakdowns for short periods only: the by-filename breakdown can be one of the most expensive in terms of storage and execution overhead, and may not be suitable to leave enabled permanently on appliances with high rates of SFTP activity.

Breakdowns

Breakdown	Description
type of operation	SFTP operation type (get/put/...)
user	username of the client
filename	filename for the SFTP operation, if known and cached by the appliance. If the filename is not known it is reported as "<unknown>".
share	the share for this SFTP request.
project	the project for this SFTP request.
client	remote hostname or IP address of the SFTP client.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: SFTP bytes per second for client 'phobos.sf.fishpong.com' broken down by file name" (to view what files a particular client is accessing)

Further Analysis

See Cache: ARC accesses broken down by hit/miss to see how well an SFTP read workload is returning from cache; and Disk: I/O operations for the back-end disk I/O caused.

Since SFTP uses SSH to encrypt FTP, there will be additional CPU overhead for this protocol. To check overall CPU utilization of the appliance, see CPU: Percent utilization.

Protocol SRP bytes

Protocol: SRP bytes

This statistic shows SRP bytes/sec requested by initiators to the appliance.

Example

See Protocol: iSCSI bytes for an example of a similar statistic with similar breakdowns.

When to check

SRP bytes/sec can be used as an indication of SRP load, in terms of throughput. For a deeper analysis of SRP activity, see Protocol: SRP operations.

Breakdowns

Breakdown	Description
initiator	SRP client initiator
target	local SCSI target
project	the project for this SRP request.
lun	the LUN for this SRP request.

See the SAN page for terminology definitions.

Further Analysis

See Protocol: SRP operations for numerous other breakdowns on SRP operations; also see Cache: ARC accesses broken down by hit/miss to see how well an SRP read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

Protocol SRP operations

Protocol: SRP operations

This statistic shows SRP operations/sec (SRP IOPS) requested by initiators to the appliance. Various useful breakdowns are available: to show the initiator, target, type and latency of the SRP I/O.

Example

See Protocol: iSCSI operations for an example of a similar statistic with similar breakdowns.

When to check

SRP operations/sec can be used as an indication of SRP load.

Use the latency breakdown when investigating SRP performance issues, especially to quantify the magnitude of the issue. This measures the I/O latency component for which the appliance is responsible for, and displays it as a heat map so that the overall latency pattern can be seen, along with outliers. If the SRP latency is high, drill down further on latency to identify the client initiator, the type of operation and LUN for the high latency, and, check other statistics for both CPU and Disk load to investigate why the appliance is slow to respond; if latency is low, the appliance is performing quickly, and any performance issues experienced on the client initiator are more likely to be caused by other factors in the environment: such as the network infrastructure, and CPU load on the client itself.

The best way to improve performance is to eliminate unnecessary work, which may be identified through the client initiator, lun and command breakdowns.

Breakdowns

Breakdown	Description
initiator	SRP client initiator
target	local SCSI target
project	the project for this SRP request.
lun	the LUN for this SRP request.
type of operation	SRP operation type. This shows how the SCSI command is transported by the SRP protocol, which can give an idea to the nature of the I/O.
command	SCSI command sent by the SRP protocol. This can show the real nature of the requested I/O (read/write/sync-cache/...).

Breakdown	Description
latency	a heat map showing the latency of SRP I/O, as measured from when the SRP request arrived on the appliance from the network, to when the response is sent; this latency includes the time to process the SRP request, and to perform any disk I/O.
offset	a heat map showing the file offset of SRP I/O. This can be used to identify random or sequential SRP IOPS. Use the Disk I/O operations statistic to check whether random SRP IOPS maps to random Disk IOPS after the LUN and RAID configuration has been applied.
size	a heat map showing the distribution of SRP I/O sizes.

These breakdowns can be combined to produce powerful statistics. For example:

- "Protocol: SRP operations per second of command read broken down by latency" (to examine latency for SCSI reads only)

Further Analysis

See Protocol: SRP bytes for the throughput of this SRP I/O; also see Cache: ARC accesses broken down by hit/miss to see how well an SRP read workload is returning from cache, and Disk: I/O operations for the back-end disk I/O caused.

CPU CPUs

CPU: CPUs

The CPUs statistic is used to display the heat map for CPUs broken down by percent utilization. This is the most accurate way to examine how CPUs are utilized.

When to check

When investigating CPU load, after checking the utilization average from CPU: Percent utilization.

This statistic is particularly useful for identifying if a single CPU is fully utilized, which can happen if a single thread is saturated with load. If the work performed by this thread cannot be offloaded to other threads so that it can be run concurrently across multiple CPUs, then that single CPU can become the bottleneck. This will be seen as a single CPU stuck at 100% utilization for several seconds or more, while the other CPUs are idle.

Breakdowns

Breakdown	Description
percent utilization	A heat map with utilization on the Y-axis and each level on the Y-axis colored by the number of CPU at that utilization: from light (none) to dark (many).

Details

CPU utilization includes the time to process instructions (that are not part of the idle thread); which includes memory stall cycles. CPU utilization can be caused by:

- executing code (including spinning on locks)
- memory load

Since the appliance primarily exists to move data, memory load often dominates. So a system with high CPU utilization may actually be high as it is moving data.

CPU Kernel spins

CPU: Kernel spins

This statistic counts the number of spin cycles on kernel locks, which consume CPU.

An understanding of operating system internals is required to properly interpret this statistic.

When to check

When investigating CPU load, after checking CPU: Percent utilization and CPU: CPUs broken down by percent utilization.

Some degree of kernel spins is normal for processing any workload, due to the nature of multi-threaded programming. Compare the behavior of kernel spins over time, and for different workloads, to develop an expectation for what is normal.

Breakdowns

Breakdown	Description
type of synchronization primitive	type of lock (mutex/...)
CPU identifier	CPU identifier number (0/1/2/3/...)

Cache ARC adaptive parameter

Cache: ARC adaptive parameter

This is arc_p from the ZFS ARC. This shows how the ARC is adapting its MRU and MFU list size depending on the workload.

An understanding of ZFS ARC internals may be required to properly interpret this statistic.

When to check

Rarely; this may be useful for identifying internal behavior of the ARC, however there are other statistics to check before this one.

If there are caching issues on the appliance, check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytics Cache: ARC size and Cache: ARC evicted bytes for further details on the ARC behavior.

Breakdowns

none.

Cache ARC evicted bytes

Cache: ARC evicted bytes

This statistic shows bytes that were evicted from the ZFS ARC, as part of its usual housekeeping. The breakdown allows L2ARC eligibility to be examined.

An understanding of ZFS ARC internals may be required to properly interpret this statistic.

When to check

This could be checked if you were considering to install cache devices (L2ARC), as this statistic can be broken down by L2ARC state. If L2ARC eligible data was frequently being evicted from the ARC, then the presence of cache devices could improve performance.

This may also be useful to check if you have issues with cache device warmup. The reason may be that your workload is not L2ARC eligible.

If there are ARC caching issues on the appliance, also check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytics Cache: ARC size for further details on the ARC behavior.

Breakdowns

Breakdown	Description
L2ARC state	shows L2ARC cached or uncached, L2ARC eligible or ineligible.

Cache ARC size

Cache: ARC size

This statistic shows the size of the primary filesystem cache, the DRAM based ZFS ARC.

An understanding of ZFS ARC internals may be required to properly interpret this statistic.

When to check

When examining the effectiveness of the ARC on the current workload. The ARC should automatically increase in size to fill most of available DRAM, when enough data be accessed by the current workload to be placed in the cache. The breakdown allows the contents of the ARC to be identified by type.

This may also be checked when using cache devices (L2ARC) on systems with limited DRAM, as the ARC can become consumed with L2ARC headers.

If there are ARC caching issues on the appliance, also check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload.

Breakdowns

Available breakdowns:

Breakdown	Description
component	type of data in the ARC. See table below

ARC component types:

Component	Description
ARC data	cached contents, including filesystem data and filesystem metadata.
ARC headers	space consumed by metadata of the ARC itself. The ratio of headers to data is relative to the ZFS record size used; a small record size may mean more ARC headers to refer to the same volume.
ARC other	other kernel consumers of the ARC
L2ARC headers	space consumed by tracking buffers stored on L2ARC devices. If the buffer is on the L2ARC and yet still in ARC DRAM, it is considered "ARC headers" instead.

Cache ARC target size

Cache: ARC target size

This is arc_c from the ZFS ARC. This shows how the target size which the ARC is attempting to maintain. For the actual size, see the Advanced Analytic Cache: ARC size.

An understanding of ZFS ARC internals may be required to properly interpret this statistic.

When to check

Rarely; this may be useful for identifying internal behavior of the ARC, however there are other statistics to check before this one.

If there are caching issues on the appliance, check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytics Cache: ARC size and Cache: ARC evicted bytes for further details on the ARC behavior.

Breakdowns

none.

Cache DNLC accesses

Cache: DNLC accesses

This statistic shows accesses to the DNLC (Directory Name Lookup Cache). The DNLC caches pathname to inode lookups.

An understanding of operating system internals may be required to properly interpret this statistic.

When to check

This may be useful to check if a workload accesses millions of small files, for which the DNLC can help.

If there are generic caching issues on the appliance, first check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytic Cache: ARC size for the size of the ARC.

Breakdowns

Breakdown	Description
hit/miss	shows counts for hits/misses, allowing the effectiveness of the DNLC to be checked.

Cache DNLC entries

Cache: DNLC entries

This shows the number of entries in the DNLC (Directory Name Lookup Cache). The DNLC caches pathname to inode lookups.

An understanding of operating system internals may be required to properly interpret this statistic.

When to check

This may be useful to check if a workload accesses millions of small files, for which the DNLC can help.

If there are generic caching issues on the appliance, first check the Cache: ARC accesses statistic to see how well the ARC is performing, and the Protocol statistics to understand the requested workload. Then, check the Advanced Analytic Cache: ARC size for the size of the ARC.

Breakdowns

none.

Cache L2ARC errors

Cache: L2ARC errors

This statistic shows L2ARC error statistics.

When to check

This may be useful to leave enabled when using cache devices, for when troubleshooting L2ARC issues beyond the standard statistics.

Breakdowns

Available breakdowns:

Breakdown	Description
error	L2ARC error type. See table below.

L2ARC error types:

Error	Description
memory abort	The L2ARC choose not to populate for a one second interval due to a shortage of system memory (DRAM) which holds the L2ARC metadata. Continual memory aborts will prevent the L2ARC from warming up.
bad checksum	A read from a cache device failed the ZFS ARC checksum. This may be an indicator that a cache device is beginning to fail.
io error	A cache device returned an error. This may be an indicator that a cache device is beginning to fail.

Cache L2ARC size

Cache: L2ARC size

This shows the size of data stored on the L2ARC cache devices. This is expected to increase in size over a period of hours or days, until the amount of amount of constant L2ARC eligible data is cached, or the cache devices are full.

When to check

When troubleshooting L2ARC warmup. If the size is small, check that the workload applied should be populating the L2ARC using the statistic Cache: ARC evicted bytes broken down by L2ARC state, and use the Protocol breakdowns such as by size and by offset to confirm that the workload is of random I/O. Sequential I/O does not populate the L2ARC. Another statistic to check is Cache: L2ARC errors.

The L2ARC size does shrink, if data that was cached is deleted from the filesystem.

Breakdowns

none.

Data Movement NDMP file system operations

Data Movement: NDMP file system operations

This statistic shows accesses to the NDMP file system operations/sec.

When to check

This could be useful to check when investigating the source of ZFS load. This would be after checking all other sources of file system activity, via the Protocol statistics. Also see the standard Analytics statistic Data Movement: NDMP bytes transferred to/from disk and Data Movement: NDMP bytes transferred to/from tape.

Breakdowns

Breakdown	Description
type of operation	read/write/...

Data Movement NDMP jobs

Data Movement: NDMP jobs

This statistic shows active NDMP job counts.

When to check

When monitoring NDMP progress, and troubleshooting NDMP. Also see the standard Analytics statistic Data Movement: NDMP bytes transferred to/from disk and Data Movement: NDMP bytes transferred to/from tape.

Breakdowns

Breakdown	Description
type of operation	type of job: backup/restore.

Disk Percent utilization

Disk: Percent utilization

This statistic shows average utilization across all disks. The per-disk breakdown shows the utilization that that disk contributed to the total average, not the utilization of that disk.

When to check

This statistic may be useful to trigger an alert based on the average for all disks.

Investigating disk utilization is usually much more effective using the standard Analytics statistic Disk: Disks broken down by percent utilization - which instead of averaging utilization, presents it as a heat map. This allows individual disk utilization to be examined.

Breakdowns

Breakdown	Description
disk	disks, including system and pool disks.

The disk breakdown shows the contribution to the average percent which each disk made.

Notes

A system with 100 disks would never show more than 1 for any disk breakdown, unless that disk was selected and displayed separately as a raw statistic. Such a system would also show 0 percent utilization for disks less than 50% busy, due to rounding. Since this may be a source of confusion, and that there is a better statistic available for most situations (Disk: Disks), this statistic has been placed in the Advanced category.

See Disk: Disks broken down by percent utilization for a different and usually more effective way to display this data.

Disk ZFS DMU operations

Disk: ZFS DMU operations

This statistic shows ZFS DMU (Data Management Unit) operations/sec.

An understanding of ZFS internals is required to properly interpret this statistic.

When to check

Troubleshooting performance issues, after all relevant standard Analytics have been examined.

The DMU object type breakdown can identify if there is excessive DDT (Data Deduplication Table) activtity. See Data Deduplication.

Breakdowns

Breakdown	Description
type of operation	read/write/...
DMU object level	integer
DMU object type	ZFS plain file/ZFS directory/DMU dnode/SPA space map/...

Disk ZFS logical IO bytes

Disk: ZFS logical I/O bytes

This statistic shows logical access to the ZFS file system as bytes/sec. Logical I/O refers to the type of operations as those that are requested to the file system, such as by NFS; as opposed to physical I/O, which are the requests by the file system to the back-end pool disks.

When to check

This could be useful while investigating how I/O is processed between the Protocol layer and pool disks.

Breakdowns

Breakdown	Description
type of operation	read/write/...
pool name	Name of the disk pool.

Disk ZFS logical IO operations

Disk: ZFS logical I/O operations

This statistic shows logical access to the ZFS file system as operations/sec. Logical I/O refers to the type of operations as those that are requested to the file system, such as by NFS; as opposed to physical I/O, which are the requests by the file system to the back-end pool disks.

When to check

This could be useful while investigating how I/O is processed between the Protocol layer and pool disks.

Breakdowns

Breakdown	Description
type of operation	read/write/...
pool name	Name of the disk pool.

Memory Dynamic memory usage

Memory: Dynamic memory usage

This statistic gives a high level view of memory (DRAM) consumers, updated every second.

When to check

This can be used to check that the filesystem cache has grown to consume available memory.

Breakdowns

Available breakdowns:

Breakdown	Description
application name	See table below.

Application names:

Application Name	Description
cache	The ZFS filesystem cache (ARC). This will grow to consume as much of available memory as possible, as it caches frequently accessed data.
kernel	The operating system kernel.
mgmt	The appliance management software.
unused	Unused space.

Memory Kernel memory

Memory: Kernel memory

This statistic shows kernel memory which is allocated, and can be broken down by kernel cache (kmem cache).

An understanding of operating system internals is required to understand this statistic.

When to check

Rarely. If the dashboard were to show kernel memory as a large consumer of available DRAM (in the Usage: Memory section), then this may be used when troubleshooting the cause. Also see Memory: Kernel memory in use and Memory: Kernel memory lost to fragmentation.

Breakdowns

Breakdown	Description
kmem cache	Kernel memory cache name.

Memory Kernel memory in use

Memory: Kernel memory in use

This statistic shows kernel memory which is in use (populated), and can be broken down by kernel cache (kmem cache).

An understanding of operating system internals is required to understand this statistic.

When to check

Rarely. If the dashboard were to show kernel memory as a large consumer of available DRAM (in the Usage: Memory section), then this may be used when troubleshooting the cause. Also see Memory: Kernel memory lost to fragmentation.

Breakdowns

Breakdown	Description
kmem cache	Kernel memory cache name.

Memory Kernel memory lost to fragmentation

Memory: Kernel memory lost to fragmentation

This statistic shows kernel memory which is currently lost to fragmentation, and can be broken down by kernel cache (kmem cache). Such a state can occur when memory is freed (for example, when cached file system data is deleted), and the kernel has yet to recover the memory buffers.

An understanding of operating system internals is required to understand this statistic.

When to check

Rarely. If the dashboard were to show kernel memory as a large consumer of available DRAM (in the Usage: Memory section), then this may be used when troubleshooting the cause. Also see Memory: Kernel memory in use.

Breakdowns

Breakdown	Description
kmem cache	Kernel memory cache name.

Network IP bytes

Network: IP bytes

This statistic shows IP payload bytes/second, excluding the Ethernet/IB and IP headers.

When to check

Rarely. Network throughput monitoring can be achieved using the standard Analytics statistic Network: Device bytes, which is enabled and achived by default. Examining by-client throughput can usually be achieved through the Protocol statistic (for example, Protocol: iSCSI bytes, which allows other useful breakdowns based on the protocol). This statistic is most useful if the previous two were not appropriate for some reason.

Breakdowns

Breakdown	Description
hostname	remote client, either as a hostname or IP address.
protocol	IP protocol: tcp/udp
direction	relative to the appliance. in/out

Network IP packets

Network: IP packets

This statistic shows IP packets/second.

When to check

Rarely. Since packets usually map to protocol operations, it is often more useful to examine these using the Protocol statistics (for example, Protocol: iSCSI operations, which allows other useful breakdowns based on the protocol).

Breakdowns

Breakdown	Description
hostname	remote client, either as a hostname or IP address.
protocol	IP protocol: tcp/udp
direction	relative to the appliance. in/out

Network TCP bytes

Network: TCP bytes

This statistic shows TCP payload bytes/second, excluding the Ethernet/IB, IP and TCP headers.

When to check

Rarely. Network throughput monitoring can be achieved using the standard Analytics statistic Network: Device bytes, which is enabled and achived by default. Examining by-client throughput can usually be achieved through the Protocol statistic (for example, Protocol: iSCSI bytes, which allows other useful breakdowns based on the protocol). This statistic is most useful if the previous two were not appropriate for some reason.

Breakdowns

Breakdown	Description
client	remote client, either as a hostname or IP address.
local service	TCP port: http/ssh/215(administration)/...
direction	relative to the appliance. in/out

Network TCP packets

Network: TCP packets

This statistic shows TCP packets/second.

When to check

Rarely. Since packets usually map to protocol operations, it is often more useful to examine these using the Protocol statistics (for example, Protocol: iSCSI operations, which allows other useful breakdowns based on the protocol).

Breakdowns

Breakdown	Description
client	remote client, either as a hostname or IP address.
local service	TCP port: http/ssh/215(administration)/...
direction	relative to the appliance. in/out

System NSCD backend requests

System: NSCD backend requests

This statistic shows requests made by NSCD (Name Service Cache Daemon) to back-end sources, such as DNS, NIS, etc.

An understanding of operating system internals may be required to properly interpret this statistic.

When to check

It may be useful to check the latency breakdown if long latencies were experienced on the appliance, especially during administrative logins. The breakdowns for the database name and source will show what the latency is for, and which remote server is responsible.

Breakdowns

Breakdown	Description
type of operation	request type
result	success/fail
database name	NSCD database (DNS/NIS/...)
source	hostname or IP address of this request

Breakdown	Description
latency	time for this request to complete

System NSCD operations

System: NSCD operations

This statistic shows requests made to NSCD (Name Service Cache Daemon).

An understanding of operating system internals may be required to properly interpret this statistic.

When to check

This can be used to check the effectiveness of the NSCD cache, by using the hit/miss breakdown. Misses become backend requests to remote sources, which can be examined using System: NSCD backend requests.

Breakdowns

Breakdown	Description
type of operation	request type
result	success/fail
database name	NSCD database (DNS/NIS/...)
latency	time for this request to complete
hit/miss	result from the cache lookup: hit/miss

Datasets

Introduction

The term *dataset* refers to the in memory cached and on disk saved data for a statistic, and is presented as an entity in Analytics with administration controls.

Datasets are automatically created when statistics are viewed in Open Worksheets, but are not saved to disk for future viewing unless they are *archived*. See the Actions section of Concepts.

BUI

The Analytics->Datasets page in the BUI lists all datasets. These include open statistics that are being viewed in a worksheet (and as such are temporary datasets - they will disappear when the worksheet is closed), and statistics that are being archived to disk.

The following fields are displayed in the Dataset view for all datasets:

Field	Description
Status icon	See below table
Name	Name of the statistic/dataset
Since	First timestamp in dataset. For open statistics, this is the time the statistic was opened - which may be minutes earlier. For archived statistics, this is the first time in the archived dataset which indicates how far back in the past this dataset goes - which may be days, weeks, months. Sorting this column will show the oldest datasets available.
On Disk	Space this dataset consumes on disk
In Core	Space this dataset consumers in main memory

The following icons are visible in the BUI view; some of these will only be visible during mouse over of a dataset entry:

icon	description
	Dataset is actively collecting data
	Dataset is currently suspended from collecting data
	Suspend/Resume archived datasets
	Enable archiving of this dataset to disk
	Destroy this dataset

See Actions for descriptions for these dataset actions.

CLI

The `analytics datasets` context allows management of datasets.

Viewing available datasets

Use the show command to list datasets:

```
caji:analytics datasets> show
Datasets:

DATASET     STATE   INCORE ONDISK NAME
dataset-000 active    674K  35.7K arc.accesses[hit/miss]
dataset-001 active    227K  31.1K arc.l2_accesses[hit/miss]
dataset-002 active    227K  31.1K arc.l2_size
dataset-003 active    227K  31.1K arc.size
dataset-004 active    806K  35.7K arc.size[component]
dataset-005 active    227K  31.1K cpu.utilization
dataset-006 active    451K  35.6K cpu.utilization[mode]
dataset-007 active   57.7K      0 dnlc.accesses
dataset-008 active    490K  35.6K dnlc.accesses[hit/miss]
dataset-009 active    227K  31.1K http.reqs
dataset-010 active    227K  31.1K io.bytes
dataset-011 active    268K  31.1K io.bytes[op]
dataset-012 active    227K  31.1K io.ops
...
```

Many of the above datasets are archived by default, there is only one that is additional: "dataset-007", which has no ONDISK size, indicating that it is a temporary statistic that isn't archived. The names of the statistics are abbreviated versions of what is visible in the BUI: "dnlc.accesses" is short for "Cache: DNLC accesses per second".

Specific dataset properties can be viewed after selecting it:

```
caji:analytics datasets> select dataset-007
caji:analytics dataset-007> show
Properties:
                          name = dnlc.accesses
                      grouping = Cache
                   explanation = DNLC accesses per second
                        incore = 65.5K
                          size = 0
                     suspended = false
```

Reading datasets

Datasets statistics can be read using the read command, followed by the number of previous seconds to display:

```
caji:analytics datasets> select dataset-007
caji:analytics dataset-007> read 10
DATE/TIME                  /SEC       /SEC BREAKDOWN
2009-10-14 21:25:19         137          - -
2009-10-14 21:25:20         215          - -
2009-10-14 21:25:21         156          - -
2009-10-14 21:25:22         171          - -
2009-10-14 21:25:23        2722          - -
2009-10-14 21:25:24         190          - -
```

```
2009-10-14 21:25:25         156          - -
2009-10-14 21:25:26         166          - -
2009-10-14 21:25:27         118          - -
2009-10-14 21:25:28        1354          - -
```

Breakdowns will also be listed if available. The following shows CPU utilization broken down CPU mode (user/kernel), which was available as `dataset-006`:

```
caji:analytics datasets> select dataset-006
caji:analytics dataset-006> read 5
DATE/TIME                %UTIL      %UTIL BREAKDOWN
2009-10-14 21:30:07          7          6 kernel
                                        0 user
2009-10-14 21:30:08          7          7 kernel
                                        0 user
2009-10-14 21:30:09          0          - -
2009-10-14 21:30:10         15         14 kernel
                                        1 user
2009-10-14 21:30:11         25         24 kernel
                                        1 user
```

The summary is shown in "%UTIL", and contributing elements in "%UTIL BREAKDOWN". At 21:30:10, there 14% kernel time and 1% user time. The 21:30:09 line shows 0% in the "%UTIL" summary, and so does not list breakdowns ("--").

Suspending and Resuming all datasets

The CLI has a feature that is not yet available in the BUI: the ability to suspend and resume all datasets. This may be useful when benchmarking the appliance to determine its absolute maximum performance. Since some statistics can consume significant CPU and disk resources to archive, benchmarks performed with these statistics enabled are invalid.

To suspend all datasets use `suspend`:

```
caji:analytics datasets> suspend
This will suspend all datasets. Are you sure? (Y/N) y
caji:analytics datasets> show
Datasets:

DATASET     STATE   INCORE ONDISK NAME
dataset-000 suspend   638K   584K arc.accesses[hit/miss]
dataset-001 suspend   211K   172K arc.l2_accesses[hit/miss]
dataset-002 suspend   211K   133K arc.l2_size
dataset-003 suspend   211K   133K arc.size
...
```

To resume all datasets use `resume`:

```
caji:analytics datasets> resume
caji:analytics datasets> show
Datasets:

DATASET     STATE   INCORE ONDISK NAME
```

```
dataset-000 active    642K   588K arc.accesses[hit/miss]
dataset-001 active    215K   174K arc.l2_accesses[hit/miss]
dataset-002 active    215K   134K arc.l2_size
dataset-003 active    215K   134K arc.size
...
```

CHAPTER 4

Performance Impact

Overhead

Performance Impact

Analytics statistic collection comes at some cost to overall performance. This should not be an issue if you understand what that cost will be, and how to minimize or avoid it. Types of performance impact are discussed in the storage and execution sections.

Storage

Analytics statistics can be archived, meaning they will be a dataset that is continually read and saved to the system disks in one second summaries. This allows statistics to be viewed month by month, day by day, right down to second by second. Data is not discarded - if an appliance has been running for two years, you can zoom down to by-second views for any time in the previous two years for your archived datasets. Depending on the type of statistic, this could present an issue with system disk usage.

You can monitor the growing sizes of the datasets in the Datasets view, and destroy datasets that are growing too large. The system disks have compression enabled, so the sizes visible in the datasets view will be larger than the space consumed on disk after compression. See the System view for system disk usage and available space.

The following are example sizes taken from an appliance that has been running for over 4 months:

Category	Statistic	Span	Dataset Size*	Disk Consumed*
CPU	percent utilization	130 days	127 MB	36 MB

Category	Statistic	Span	Dataset Size*	Disk Consumed*
Protocol	NFSv3 operations per second	130 days	127 MB	36 MB
Protocol	NFSv3 operations per second broken down by type of operation	130 days	209 MB	63 MB
CPU	percent utilization broken down by CPU mode	130 days	431 MB	91 MB
Network	device bytes per second broken down by device	130 days	402 MB	119 MB
Disk	I/O bytes per second broken down by disk	130 days	2.18 GB	833 MB
Disk	I/O operations per second broken down by latency	31 days	1.46 GB	515 MB

** These sizes will vary depending on your workload; they have been provided as a rough guide.*

It is worth noting that the appliance has been intended to have 500 Gbyte mirrored system disks, most of which will be available to store datasets.

The factors that affect consumed disk space are:

- Type of statistic: raw vs breakdowns
- For breakdowns: number of breakdowns, and breakdown name length
- Activity rate

Keep an eye on the size in the Datasets view. If a dataset is growing too large, and you want to stop it from growing but keep the historic data - use the suspend action.

Raw statistics

Statistics that are a single value (sometimes written "as a raw statistic") will not consume much disk space for these reasons:

- Integer values consume a fixed and small amount of space.
- The archives are compressed when saved - which will significantly reduce the size for statistics that are mostly zero.

Examples:

- CPU: percent utilization
- Protocol: NFSv3 operations per second

Breakdowns

Statistics that have breakdowns can consume much more data, as shown in the previous table, since:

- Each breakdown is saved per second. For by-file and by-hostname breakdowns, the number of breakdowns per second may reach into the hundreds (how many different files or hosts had activity in a one second summary) - all of which must be saved to disk.
- Breakdowns have dynamic names, which themselves can be long. You may only have ten active files in your breakdown by-file statistics, but each pathname could be dozens of characters in size. This doesn't sound like much, but the dataset will grow steadily when this data is saved every second.

Examples:

- CPU: percent utilization broken down by CPU mode
- Protocol: NFSv3 operations per second broken down by type of operation
- Disk: I/O bytes per second broken down by disk
- Disk: I/O bytes per second broken down by latency

Exporting Statistics

There may come a time where you'd like to archive statistics on a different server, either to free up disk space on the appliance or for other purposes. See Open Worksheets for the export button, or Saved Worksheets for the CLI section, both of which provide a way to download the statistic data in CSV format.

Execution

Enabling statistics will incur some CPU cost for data collection and aggregation. In many situations, this overhead will not make a noticeable difference on system performance. However for systems under maximum load, including benchmark loads, the small overhead of statistic collection can begin to be noticeable.

Here are some tips for handling execution overheads:

- For dynamic statistics, only archive those that are important to record 24x7.
- Statistics can be suspended, eliminating data collection and the collection overhead. This may be useful if gathering a short interval of a statistic is sufficient for your needs (such as troubleshooting performance). Enable the statistic, wait some minutes, then click the power icon in the Datasets view to suspend it. Suspended datasets keep their data for later viewing.
- Keep an eye on overall performance via the static statistics when enabling and disabling dynamic statistics.

- Be aware that drilldowns will incur overhead for all events. For example, you may trace "NFSv3 operations per second for client deimos", when there is currently no NFSv3 activity from deimos. *This doesn't mean that there is no execution overhead for this statistic*. The appliance must still trace every NFSv3 event, then compare the host with "deimos" to see if the data should be recorded in this dataset - however we have already paid most of the execution cost at this point.

Static Statistics

Some statistics are sourced from operating system counters are always maintained, which may be called *static statistics*. Gathering these statistics has negligible effect on the performance of the system, since to an extent the system is already maintaining them (they are usually gathered by an operating system feature called *Kstat*). Examples of these statistics are:

Category	Statistic
CPU	percent utilization
CPU	percent utilization broken down by CPU mode
Cache	ARC accesses per second broken down by hit/miss
Cache	ARC size
Disk	I/O bytes per second
Disk	I/O bytes per second broken down by type of operation
Disk	I/O operations per second
Disk	I/O operations per second broken down by disk
Disk	I/O operations per second broken down by type of operation
Network	device bytes per second
Network	device bytes per second broken down by device
Network	device bytes per second broken down by direction
Protocol	NFSv3/NFSv4 operations per second
Protocol	NFSv3/NFSv4 operations per second broken down by type of operation

When seen in the BUI, those from the above list without "broken down by" text may have "as a raw statistic".

Since these statistics have negligible execution cost and provide a broad view of system behaviour, many are archived by default. See the default statistics list.

Dynamic Statistics

These statistics are created dynamically, and are not usually maintained by the system (they are gathered by an operating system feature called *DTrace*). Each event is *traced*, and each second this trace data is aggregated into the statistic. And so the cost of this statistic is proportional to the number of events.

Tracing disk details when the activity is 1000 ops/sec is unlikely to have a noticeable affect on performance, however measuring network details when pushing 100,000 packets/sec *is* likely to have a negative effect. The type of information gathered is also a factor: tracing file names and client names will increase the performance impact.

Examples of dynamic statistics include:

Category	Statistic
Protocol	SMB operations per second
Protocol	SMB operations per second broken down by type of operation
Protocol	HTTP/WebDAV requests per second
Protocol	... operations per second broken down by client
Protocol	... operations per second broken down by file name
Protocol	... operations per second broken down by share
Protocol	... operations per second broken down by project
Protocol	... operations per second broken down by latency
Protocol	... operations per second broken down by size
Protocol	... operations per second broken down by offset

"..." denotes any of the protocols.

The best way to determine the impact of these statistics is to enable and disable them while running under steady load. Benchmark software may be used to apply that steady load. See Tasks for the steps to calculate performance impact in this way.

Glossary

7110	Sun Storage 7110 Unified Storage System
7120	Sun ZFS Storage 7120
7210	Sun Storage 7210 Unified Storage System
7310	Sun Storage 7310 Unified Storage System
7320	Sun ZFS Storage 7320
7410	Sun Storage 7410 Unified Storage System
7420	Sun ZFS Storage 7420
7720	Sun ZFS Storage 7720
Active Directory	Microsoft Active Directory server
Alerts	Configurable log, email or SNMP trap events
Analytics	appliance feature for graphing real-time and historic performance statistics
ARC	Adaptive Replacement Cache
BUI	Browser User Interface
CLI	Command Line Interface
Cluster	Multiple heads connected to shared storage
Controller	See "Storage Controller"
CPU	Central Prcessing Unit
CRU	Customer Replaceable Component
Dashboard	appliance summary display of system health and activity
Dataset	the in-memory and on-disk data for a statistic from Analytics
DIMM	dual in-line memory module
Disk Shelf	the expansion storage shelf that is connected to the head node or storage controller
DNS	Domain Name Service

DTrace	a comprehensive dynamic tracing framework for troubleshooting kernel and application problems on production systems in real-time
FC	Fibre Channel
FRU	Field Replaceable Component
FTP	File Transfer Protocol
GigE	Gigabit Ethernet
HBA	Host Bus Adapter
HCA	Host Channel Adapter
HDD	Hard Disk Drive
HTTP	HyperText Transfer Protocol
Hybrid Storage Pool	combines disk, flash, and DRAM into a single coherent and seamless data store.
Icons	icons visible in the BUI
iSCSI	Internet Small Computer System Interface
Kiosk	a restricted BUI mode where a user may only view one specific screen
L2ARC	Level 2 Adaptive Replacement Cache
LDAP	Lightweight Directory Access Protocol
LED	light-emitting diode
Logzilla	write IOPS accelerator
LUN	Logical Unit
Masthead	top section of BUI screen
Modal Dialog	a new screen element for a specific function
NFS	Network File System
NIC	Network Interface Card
NIS	Network Information Service
PCIe	Peripheral Component Interconnect Express
Pool	provide storage space that is shared across all filesystems and LUNs
Project	a collection of shares
PSU	Power Supply Unit

QDR	quad data rate
Readzilla	read-optimized flash SSD for the L2ARC
Remote Replication	replicating shares to another appliance
Rollback	reverts all of the system software and all of the metadata settings of the system back to their state just prior to applying an update
SAS	Serial Attached SCSI
SAS-2	Serial Attached SCSI 2.0
SATA	Serial ATA
Schema	configurable properties for shares
Scripting	automating CLI tasks
Service	appliance service software
Share	ZFS filesystem shared using data protocols
SIM	SAS Interface Module
Snapshot	an image of a share
SSD	Solid State Drive
SSH	Secure Shell
Statistic	a metric visible from Analytics
Storage Controller	the head node of the appliance
Support Bundle	auto-generated files containing system configuration information and core files for use by remote support in debugging system failures
Title Bar	local navigation and function section of BUI screen
Updates	software or firmware updates
WebDAV	Web based Distributed Authoring and Versioning
ZFS	on-disk data storage subsystem

Index