

**NetApp Solution Deployment Guidelines**

**Thomson Reuters**

**cDOT QoS and Performance Management**

**Synopsis:** This document details the approach to storage system performance management for NetApp clustered Data ONTAP (cDOT) systems at Thomson Reuters.

**Authors:** Michael Arndt (arndt@netapp.com)

**Contributors:** Mitchell Vallone, Ken Zola, Ian Daniel

**Document Version:** V2

**Date:** December 2013

**CONFIDENTIAL INFORMATION**

This document contains information proprietary to Thomson Reuters and may not be reproduced, disclosed or used in whole or part without express permission of Thomson Reuters.

© Thomson Reuters 2013

Contents

[1 Introduction 3](#_Toc373783230)

[1.1 Management Summary 3](#_Toc373783231)

[1.2 Assumptions 3](#_Toc373783232)

[1.3 Change History 3](#_Toc373783233)

[1.4 Distribution List 4](#_Toc373783234)

[1.5 Glossary 4](#_Toc373783235)

[2 Storage Performance Overview 5](#_Toc373783236)

[2.1 Storage Performance Concepts 5](#_Toc373783237)

[2.2 Storage Performance Tools 5](#_Toc373783238)

[2.2.1 PA with OCUM 5.2 for cDOT vs 7 Mode 6](#_Toc373783239)

[2.2.2 OPM 1.0 features and limitations 6](#_Toc373783240)

[2.2.3 Using the wlstats utility to bridge the OPM 1.0 gaps 6](#_Toc373783241)

[2.3 QoS Overview and Recommendations for TR 7](#_Toc373783242)

[2.4 QoS policy group limits at TR and thin provisioning considerations 8](#_Toc373783243)

[3 Using Performance Advisor (PA) 9](#_Toc373783244)

[3.1 Analyzing node level performance 9](#_Toc373783245)

[3.2 Analyzing Vserver level performance 10](#_Toc373783246)

[3.3 Analyzing volume level performance 11](#_Toc373783247)

[4 Using OnCommand Performance Manager (OPM) 12](#_Toc373783248)

[4.1 OPM dashboard 12](#_Toc373783249)

[4.2 Analyzing volume level performance 13](#_Toc373783250)

[4.3 Volume incident and configuration history 14](#_Toc373783251)

[4.4 Performance alerting 15](#_Toc373783252)

[5 Configuring and using the wlstats utility 16](#_Toc373783253)

[5.1 Running wlstats interactively 16](#_Toc373783254)

[5.2 Using wlstats in archive mode 17](#_Toc373783255)

[5.3 Locating the wlstats output for a node 17](#_Toc373783256)

[5.4 Graphing wlstats output with Excel 17](#_Toc373783257)

[6 Performance incident management 18](#_Toc373783258)

[6.1 Typical steps 18](#_Toc373783259)

[6.2 Opening cases with NetApp Support 18](#_Toc373783260)

[6.3 Using NetApp Perfstat 19](#_Toc373783261)

[6.3.1 Running Perfstat interactively 19](#_Toc373783262)

[6.3.2 Requirements for non-interactive Perfstat with cDOT 19](#_Toc373783263)

[6.3.3 Running Perfstat with cdot\_perfstat\_loop.sh 19](#_Toc373783264)

[7 Additional Resources 20](#_Toc373783265)

[7.1 NetApp peak performance by platform 20](#_Toc373783266)

[8 Clustered ONTAP CLI examples 21](#_Toc373783267)

[8.1 QoS policy group configuration 21](#_Toc373783268)

[8.1.1 Create a new QoS policy group with a MB/s limit 21](#_Toc373783269)

[8.1.2 Create a new QoS policy group with an IOPs limit 21](#_Toc373783270)

[8.1.3 Create a QoS policy group with no limit 21](#_Toc373783271)

[8.1.4 Modify a QoS policy group 21](#_Toc373783272)

[8.1.5 Setting a QoS policy group on a volume 21](#_Toc373783273)

[8.2 Gathering cDOT performance statistics from the CLI 21](#_Toc373783274)

[8.2.1 Using statistics show commands 21](#_Toc373783275)

[8.2.2 Using qos statistics commands 21](#_Toc373783276)

[8.3 Creating accounts for wlstats and NetApp Perfstat 22](#_Toc373783277)

[8.3.1 Create the wlstats account 22](#_Toc373783278)

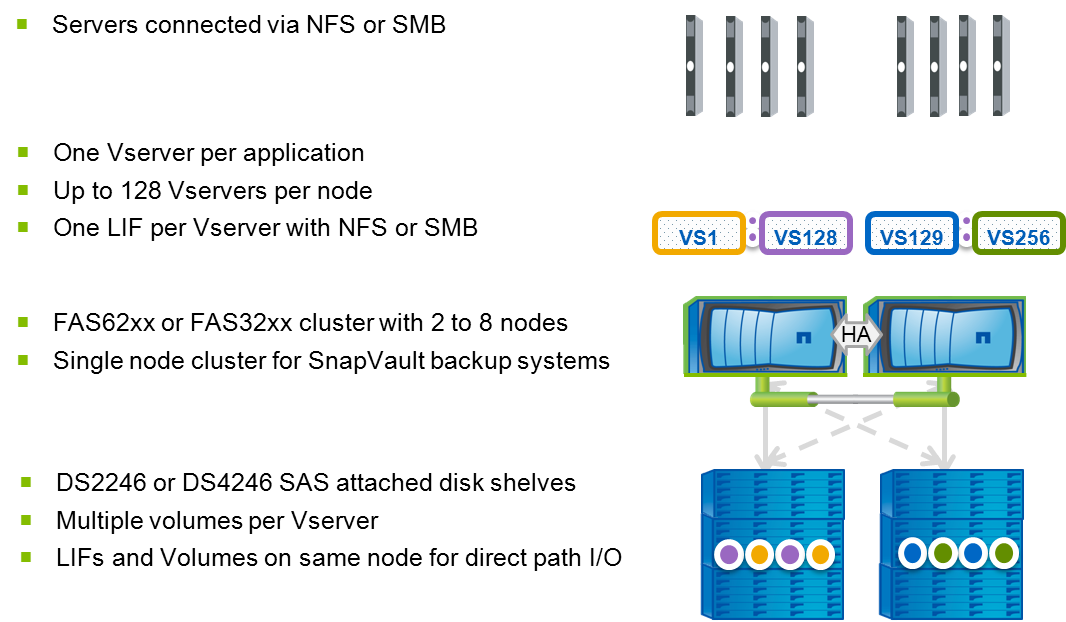
[8.3.2 Create perfstat account and setup publickey authentication 22](#_Toc373783279)

[8.3.3 Unlock diag account and set password for it 22](#_Toc373783280)

# Introduction

## Management Summary

This document details the performance management approach for NetApp clustered Data ONTAP (cDOT) storage solutions within Thomson Reuters. The majority of the shared storage and dedicated cDOT solution deployments will be configured identically for a number of components, and the performance management of those configurations will be described in this document. A typical shared storage cDOT architecture is depicted below.



## Assumptions

It is assumed the person(s) reading this document are conversant with NetApp hardware and software. They will also be conversant with the Linux and Windows operating systems, NFS, and CIFS protocols, as well as NetApp management tools.

## Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver** | **Date** | **Author** | **Key Changes** |
| 1 | November 2013 | Michael Arndt | Initial Version |
| 2 | December 2013 | Michael Arndt | Added QoS to document title, fixed paths for scripts to be in /filers/admin/scripts/support. |

## Distribution List

|  |  |
| --- | --- |
| **Name** | **Role** |
| Brett Truhler | Customer |
| Stewart Bird | Customer |
| Sridhar Chevendra | Customer |
| Mitchell Vallone | Reviewer |
| Ken Zola | Reviewer |
| Ian Daniel | Reviewer |

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| cDOT | clustered Data ONTAP |
| Node | One storage controller in a cDOT system. |
| Cluster | A collection of one or more nodes that form a cDOT storage system. |
| Vserver | A logical storage virtual server, also known as a Storage Virtual Machine (SVM), which contains LIFs, Volumes, and configuration information. |
| QoS | Quality of Service. A performance rate limiting feature of cDOT 8.2 and above. |
| QoS  Policy Group | A QoS policy group is used on a cDOT system to define a certain rate limit. This policy group is then assigned to volumes in order to enforce the limit. |
| Workload | A workload on a cDOT system is an object on the storage system, such as a volume, that has been assigned to a QoS policy group. If multiple volumes are assigned to the same QoS policy group, each volume is represented by a unique workload on the system. |
| OCUM | OnCommand Unified Manager. A storage system management product from NetApp that is used for reporting and alerting on storage resources. |
| OPM | OnCommand Performance Manager. A performance management product from NetApp that uses QoS performance statistics to perform graphing and alerting on a cDOT system. |
| PA | Performance Advisor. A performance graphing application that is part of the OCUM 5.2 and prior releases. |
| NetApp Perfstat | A NetApp performance data gathering tool used by NetApp support to troubleshoot performance issues. |

# Storage Performance Overview

## Storage Performance Concepts

Most storage performance analysis exercises will be driven by one of two scenarios:

1. A performance or stability problem has been reported somewhere within an application stack, and storage is being investigated to determine if it is performing as designed.
2. There is a question about how much additional workload can be added to a particular storage controller without exceeding the performance capabilities of the system.

In scenario #1, the main metric of interest will be the response time of the storage system, also known as latency. If the latency is within the boundaries of what is typically deemed to be acceptable, then it is unlikely that the storage system is a root cause for the performance or stability problem being investigated. If the latency is higher than the boundary of what is typically deemed to be acceptable, then the storage system may be contributing to the performance or stability problem being investigated.

The table below shows some commonly accepted values for latency metrics. If these values are not breached, this typically indicates a properly performing storage system. Some applications may require lower latency values in order to operate as designed, and some applications may tolerate much large latencies, but latencies below these values will typically be indicative of a well performing storage system.

|  |  |  |
| --- | --- | --- |
| Metric Type | Read/Other Latency | Write Latency |
| Metric Value | Under 20ms | Under 5ms |

Higher than expected values for other performance metrics such as the CPU utilization on the storage system, disk utilization, etc, are typically not indicators of a storage performance problem unless they are also accompanied by high latency.

In scenario #2, the first step is to determine the typical I/O profile of the workloads that are seen on the storage system. An I/O profile can be summarized as consisting of the following components:

* The number of IOPs seen during most of the I/O peaks over a given period of time.
* The average I/O size. This can be calculated by dividing the throughput (in units of kilobytes/second) by the number of IOPs.
* The ratio of reads versus writes.

Once the I/O profile is established, it can be compared against a known maximum desired IOPs value for the profile to determine how much additional workload can be added without exceeding the performance capabilities of the system. The maximum desired IOPs value for an I/O profile should be designed such that the system can handle the workload at an acceptable latency. The storage system may be able to handle higher IOPs values, but this will typically be done at a higher latency than desired.

## Storage Performance Tools

In Data OnTap 7 Mode, PA was used to show NetApp storage system performance. While this application does exist in OCUM 5.2 for cDOT, PA will not exist in any future version of OCUM (including 6.0 and above). OCUM 5.2 for cDOT may be used for a short period of time at TR, but the recommendation is to move to OCUM 6.1 for cDOT storage system management and OPM 1.0 for cDOT performance monitoring and alerting, in the timeframe of cDOT 8.2.1GA. This document will give a brief overview of how PA can be used for cDOT performance monitoring, but the majority of the document will focus on new tools for cDOT performance monitoring that leverage the QoS performance statistics available in cDOT 8.2 and above. A future version of this document may remove all references to usage of PA for performance monitoring in cDOT environments.

### PA with OCUM 5.2 for cDOT vs 7 Mode

Some of the features that were used in PA for 7 Mode environments are not available with PA for cDOT. The following list shows the key features of PA in 7 mode that TR used, which are either not available or operate differently with PA for cDOT.

* Performance data is in 5 minute increments for cDOT, as opposed to 1 minute granularity for 7 Mode.
* Custom performance views are not available with cDOT, and a limited number of performance counters are available with cDOT as compared to 7 Mode.
* Custom data retention is not available with cDOT, and most counters retain data for only 7 days.

### OPM 1.0 features and limitations

OPM 1.0 has a number of features that make it advantageous for cDOT environments. In addition, the OPM product is where future development for cDOT performance monitoring will take place. The following list shows some of the key features of OPM 1.0 that will benefit Thomson Reuters:

* Automatic threshold calculations based on workload history, in order to enable alerting when latency increases to a range that is higher than expected.
* Historical graphing of volume based performance counters with 5 minute granularity, with 90 days of performance data retention.
* Historical graphing of QoS latency counters in order to determine the source of latency when it occurs.
* Automatic alerting when QoS rate limits are causing significant increases in latency for a given volume.
* Automatic correlation of storage latency incidents to bully workloads on a system.
* Automatic tracking of performance related configuration changes, such a QoS policy group changes.

### Using the wlstats utility to bridge the OPM 1.0 gaps

While OPM 1.0 will provide advantages for cDOT performance monitoring at TR, there are some features that will come in the OPM 1.1 and 2.0 releases that must be accounted for while OPM 1.0 is in use. Those features, and the ways in which we will overcome them at TR, are listed below:

* Using 5 minute granularity for performance data may not be granular enough for some troubleshooting scenarios. At TR, we will use the data provided by the wlstats utility to obtain performance data that requires more granular analysis.
* A view of storage performance from the perspective of an entire node is not available in OPM 1.0, so that the headroom of a given node can be estimated, and high I/O workloads can be identified. At TR, we will use the data provided by the wlstats utility to summarize node level performance, and break down the workloads on a given node to determine what is driving the overall node utilization.

## QoS Overview and Recommendations for TR

The QoS infrastructure in cDOT 8.2 provides both a rate limiting feature and a set of advanced performance statistics that allow an administrator to easily determine which subsystem is responsible for the latency seen on a given workload. With this in mind, there is great value in using QoS policy groups even when you do not wish to set a hard limit on the amount of I/O being performed. The following describes the QoS latency counters that are available for analysis with OPM 1.0 and the wlstats utility:

* **Network Processing**: Latency observed per I/O operation in the Network subsystem (CPUs)
* **Cluster Interconnect**: Latency observed per I/O operation across the internally connected nodes in a Cluster (Indirect I/O)
* **Data Processing**: Latency observed per I/O operation in the Data management subsystem (CPUs)
* **Aggregate**: Latency observed per I/O operation in the Storage subsystem (Disk drives)
* **Policy Group**: Latency observed per I/O operation in the QoS subsystem (QoS rate limit being hit)

QoS in cDOT 8.2 allows rate limits to be set in terms of MB/s or IOPs values (but not both). Since high MB/s workloads are typically heavier on CPU utilization than high IOPs workloads, we will use QoS limits based on MB/s values at TR. Note that while we do not define specific rate limits for dedicated storage environments, it is still recommended that QoS policy groups be configured and applied at the volume level in order to utilize the QoS latency statistics that OPM and wlstats are based on.

The following considerations and recommendations should be noted with respect to QoS policy group configuration at TR:

* QoS policy group should be applied at the volume level, as this is what OPM 1.0 supports.
* When a single application is hosted per Vserver in the shared storage environment, a single QoS policy group should be configured per Vserver. The QoS policy group name should be the same as the Vserver name. This should be the most common configuration in the shared storage environment.
* If multiple applications are being served from a single Vserver, then one QoS policy group per application should be configured. The QoS policy group name should be the same as the volume name if there is a single volume for the application, or as the common volume prefix if multiple volumes are used by the application. The Oracle and SQL log backup environment on cDOT at TR is an example of this configuration.
* Using volume level QoS policy group configuration still allows for a single QoS policy group per Vserver in the shared storage environment. If a Vserver has multiple volumes assigned to the same QoS policy group, the sum of the I/O on all the volumes is allowed to go as high as that one QoS policy group limit.
* QoS policy groups should be based on MB/s throughput values when rate limiting is desired.
* QoS policy groups should be configured with a rate limit value of *INF* when using QoS policy groups to track performance metrics without limiting performance. This would be common in dedicated storage solutions, as well as in the Oracle and SQL log backup environment at TR.

## QoS policy group limits at TR and thin provisioning considerations

The following tables show the recommended values to use for QoS rate limits at TR in the cDOT shared storage environment. These values were determined based on the performance capabilities of the storage systems used for these storage tiers, as well as the overall capacity planned per node for the tier.

The first table shows the capacity and performance capabilities of the cDOT shared storage tiers. The values shown are defined on a per node basis:

|  |  |  |
| --- | --- | --- |
| Shared Storage Tier | Maximum Capacity  (Per design plan) | Maximum Throughput  (32k I/O size with 50/50 R/W) |
| FAS6220 High Tier | 56TB | 1065MB/sec |
| FAS6220 Low Tier | 78TB | 640MB/sec |

Based on the data in the above table, we can calculate the proper QoS policy group limits for each of the cDOT shared storage tiers. We also document a minimum QoS policy group limit that would be applied for applications with low capacity requirements, and the value to use when configuring a QoS policy group for tracking purposes (no rate limit defined).

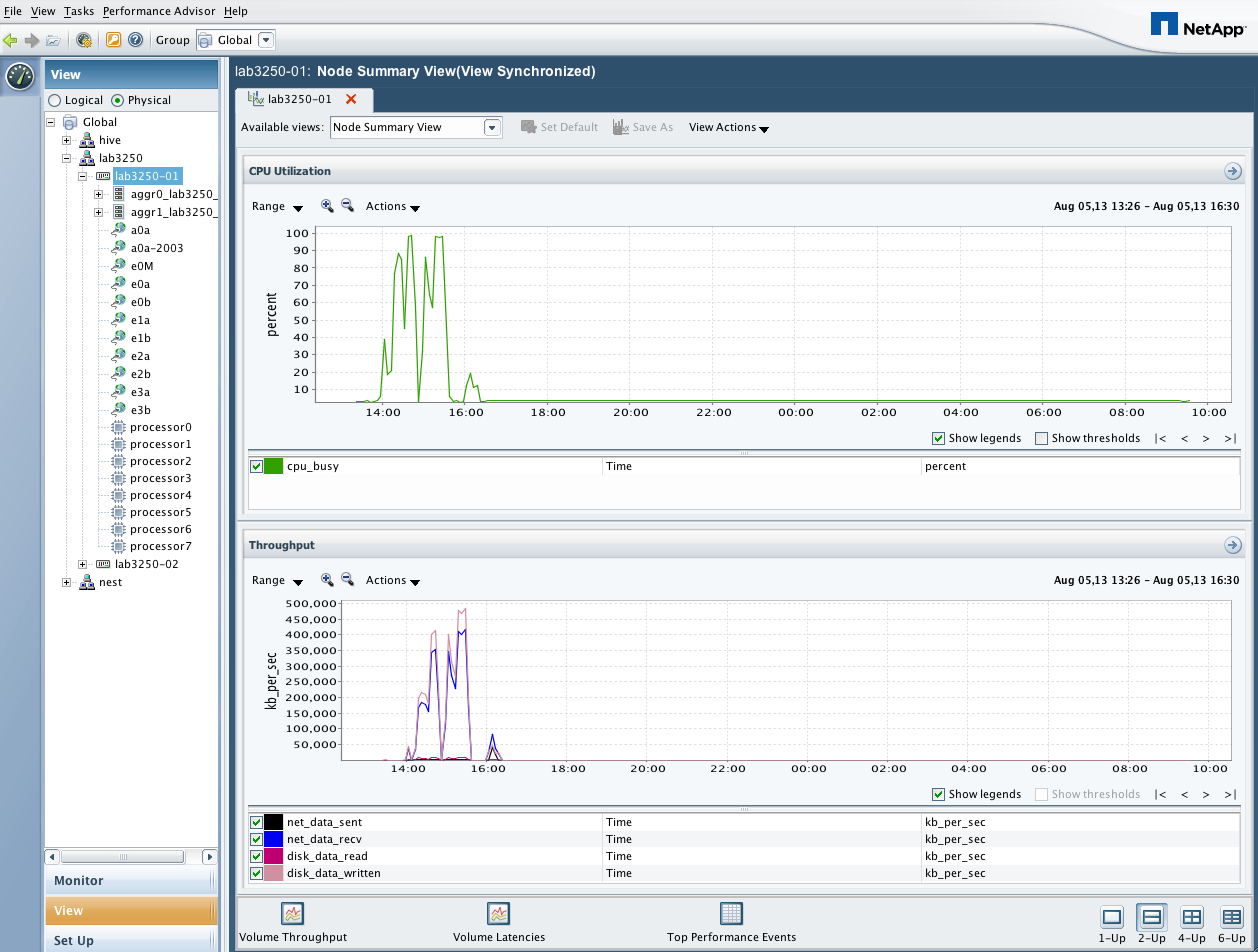
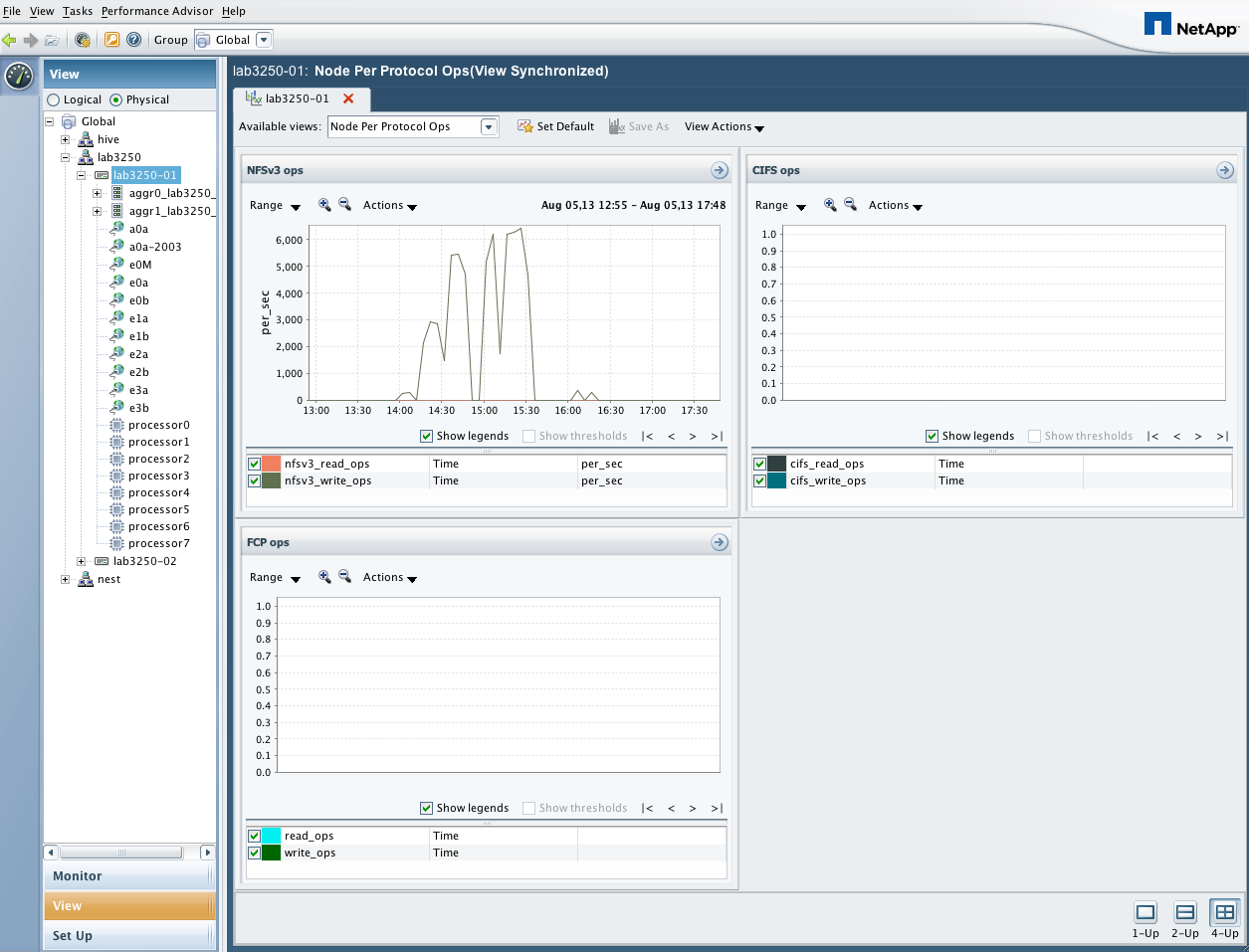
|  |  |
| --- | --- |
| Shared Storage Tier | QoS limit |
| Minimum QoS Limit  (For low capacity applications) | 10MB/s |
| FAS6220 High Tier | 19KB/s per GB |
| FAS6220 Low Tier | 8KB/s per GB |
| Dedicated | INF |

Note that while these values are based on physical capacity planned for a given node, the same values can be used when thin provisioning is in use to overprovision that physical capacity. In this model, the QoS performance capabilities would be thin provisioned just like physical storage capacity is. This will continue to provide significant benefit in the shared storage environment, due to the fact that only a small number of workloads will typically hit their QoS limits. The majority of workloads in the shared storage environment at TR will use less performance resources than planned for, just as they use less physical storage capacity than they request.

# Using Performance Advisor (PA)

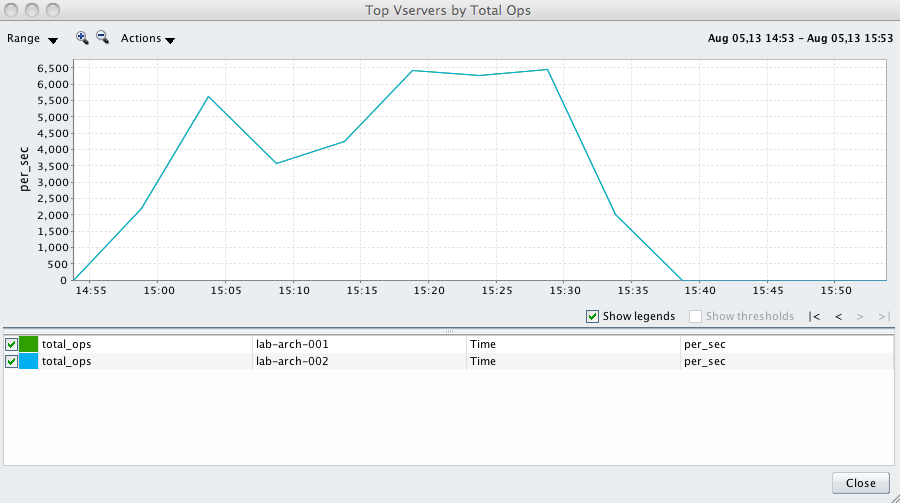
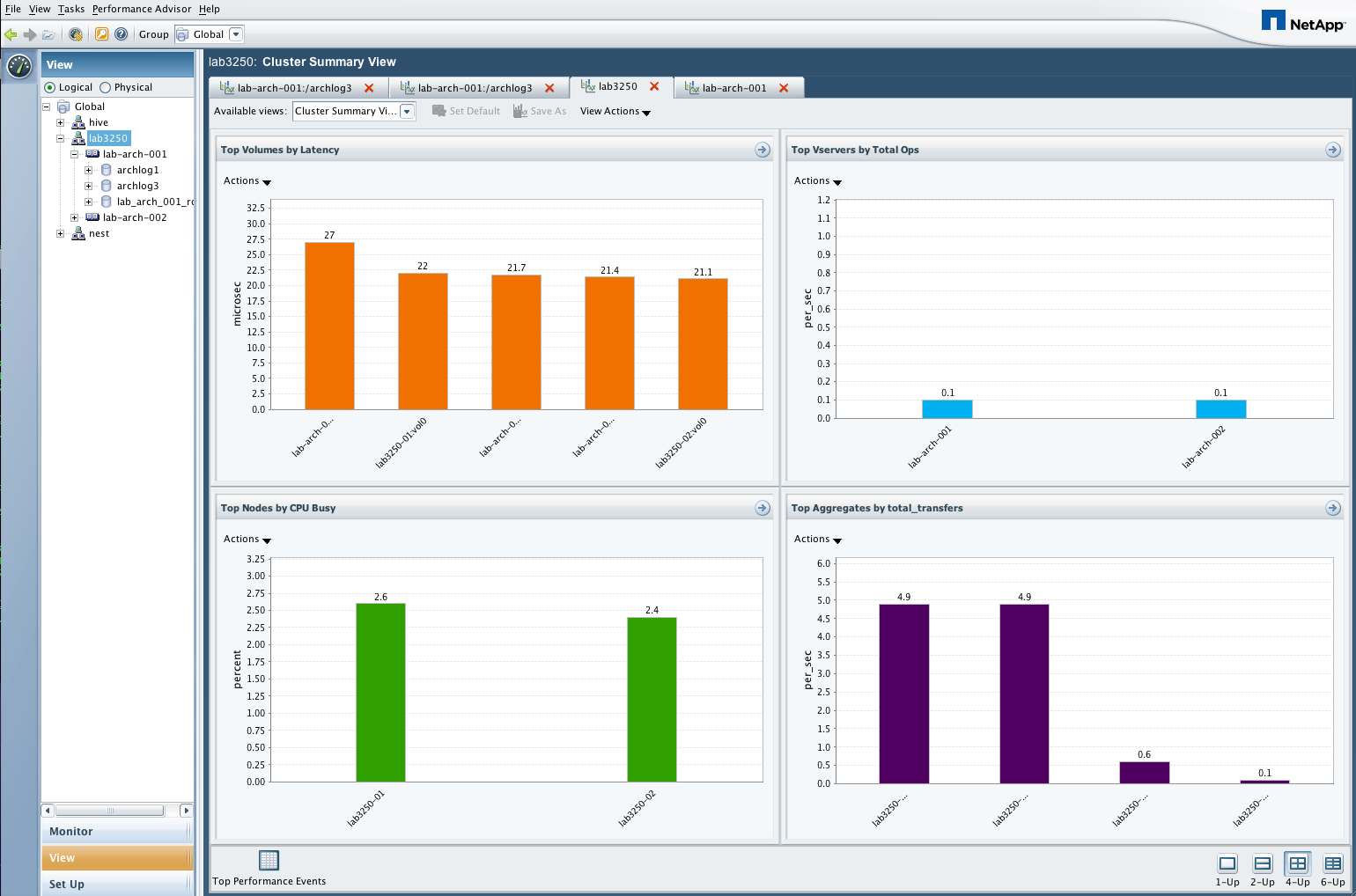
## Analyzing node level performance

Node level performance in PA can be seen in the *Physical* view of PA. The *Node Per Protocol Ops* view and *Node Summary View* can be used to see the performance metrics required to calculate the IO Profile on the node. Examples of these views are shown below.



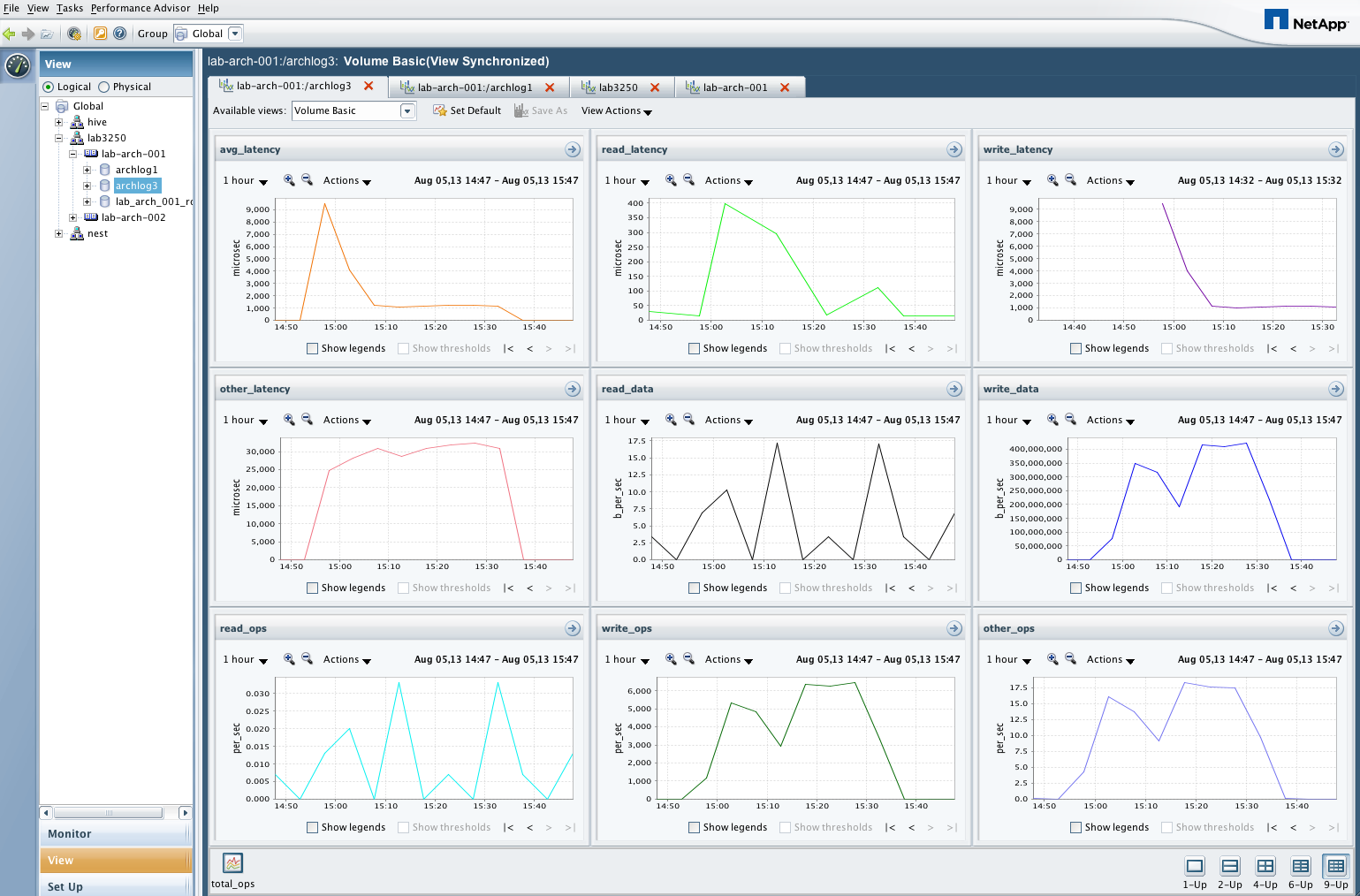
## Analyzing Vserver level performance

Vserver level performance can be seen in the *Logical* view in PA. The *Cluster Summary View* provides a bar chart with the most heavily utilized Vservers listed. These bar charts can then be right clicked, and the *show as line chart* option can be selected to see a historical view of the Vservers from a given bar chart. Examples of these graphs are shown below.



## Analyzing volume level performance

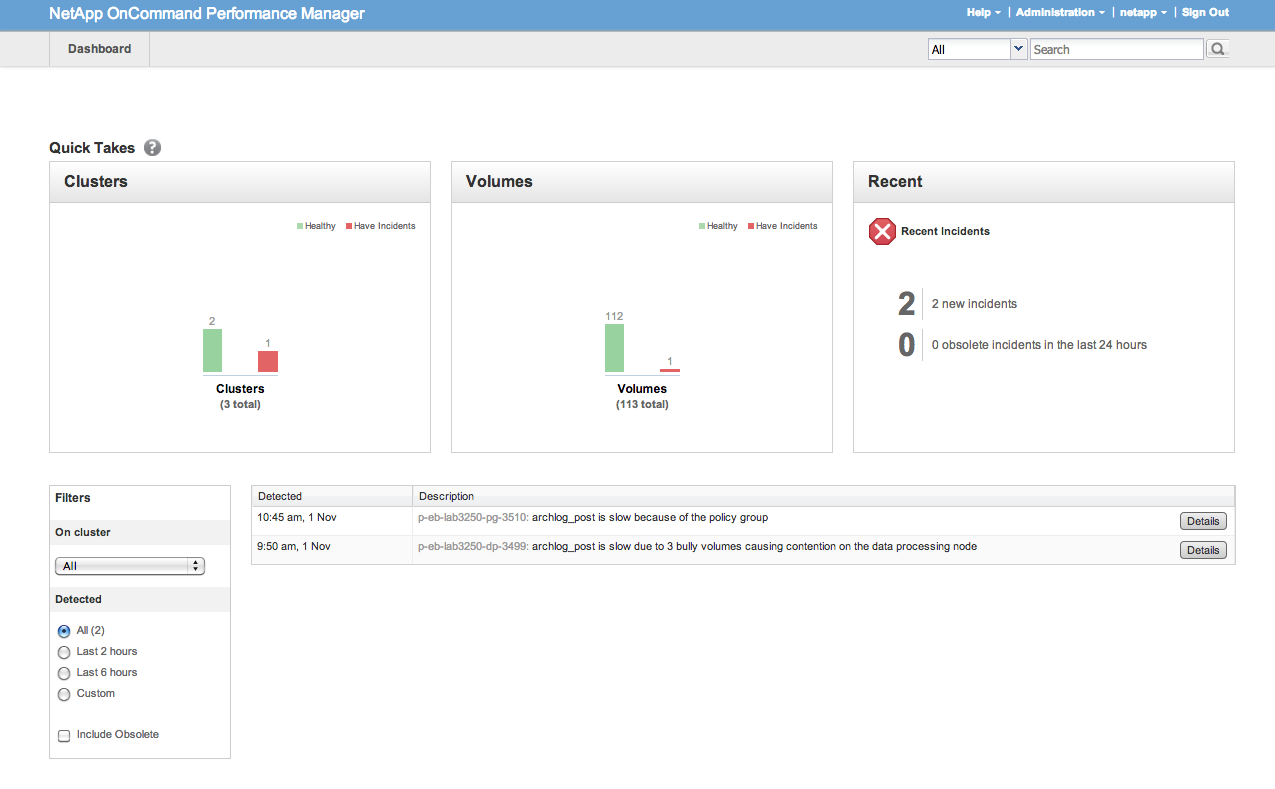
Volume level performance can be seen in the *Logical* view in PA. The *Cluster Basic View* provides graphs that show IOPs, throughput, and latency on a per volume basis. Examples of these graphs are shown below.



# Using OnCommand Performance Manager (OPM)

## OPM dashboard

After logging in to OPM, the main OPM dashboard is displayed. This dashboard shows an overview of the clusters and volumes being monitored, as well as the current performance incidents that are in progress. In order to view historical performance graphs for a given volume, use the search input box on the upper right hand corner of the page to find the volume of interest.

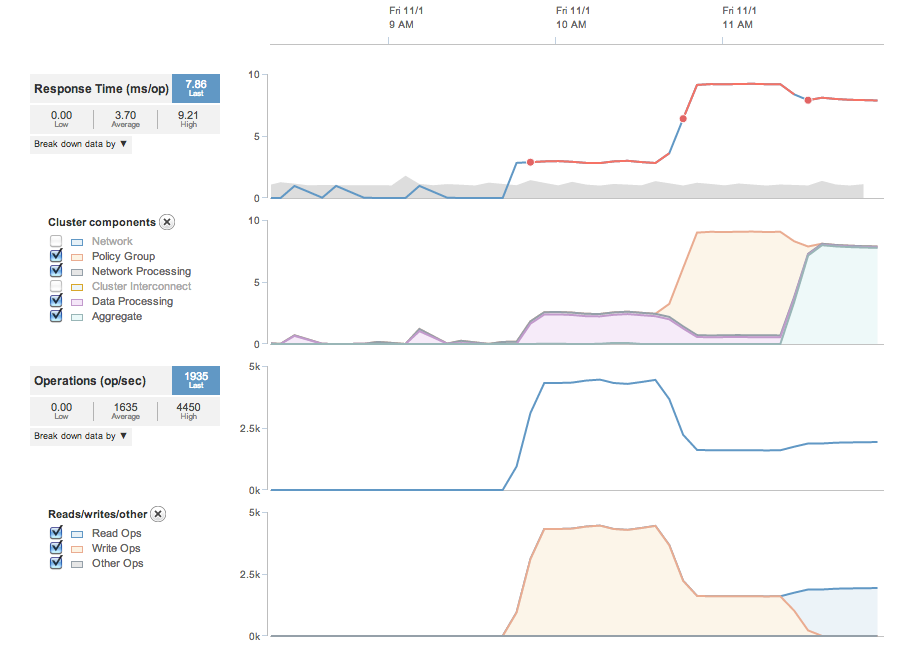


## Analyzing volume level performance

Once a volume has been selected from the search input on the main OPM dashboard, the volume performance page is displayed. The time frame slider at the top of the page allows you to scroll back in the graph history, and the time frame selectors can be dragged closer together to zoom in to a shorter range of time on a graph.



Once the desired time frame has been selected, use *the Break down data by* dropdown to show the QoS latency statistics, as denoted by the *Cluster components* list in the *Response time* graph area. The Operations graph can also be broken down into *Read/Write/Other* components by selecting the *Break down data by* dropdown.

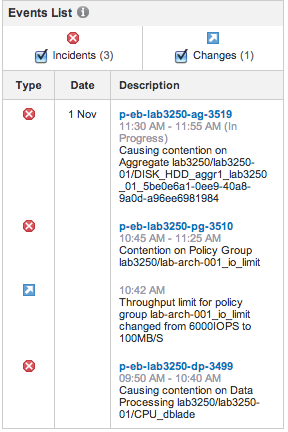


The above graphs show an example of a workload that changed over a short period of time. In this example, a heave write workload is applied to a previously idle volume, which causes write latency to increase due to *Data Processing* (CPU). Next, a QoS policy group limit is applied to the volume, and the latency increase is obviously attributed to the *Policy Group*. Finally, the workload changes to a read workload, and the main source of latency changes to the *Aggregate* (disk).

In this example, each change in workload type caused a new incident to be created (as denoted by the red sections of line and red dots on the main latency graph) due to the fact that this volume was previously idle, and this change in latency was determined to be out of the expected range. If these workloads continued occur on a regular basis, the expected range would adjust so that future latencies of this range would not trigger a performance incident.

## Volume incident and configuration history

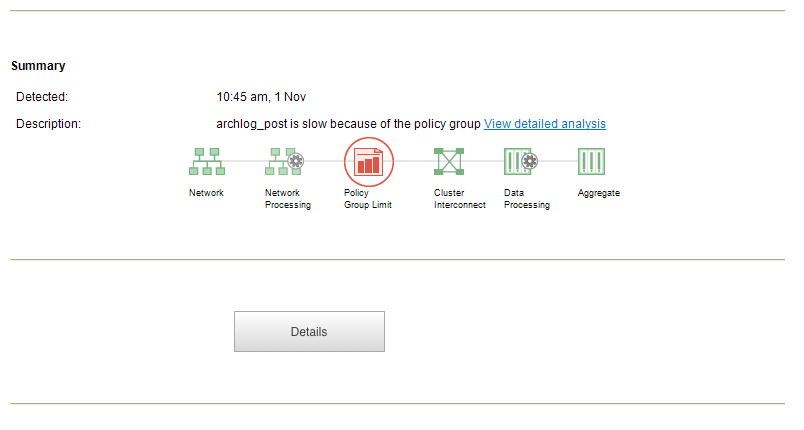
Also shown on the volume performance page will be the historical event list of incidents and configuration changes. In the following example, we see the performance incidents and configuration changes that are relative to the volume performance graph example from the previous page.



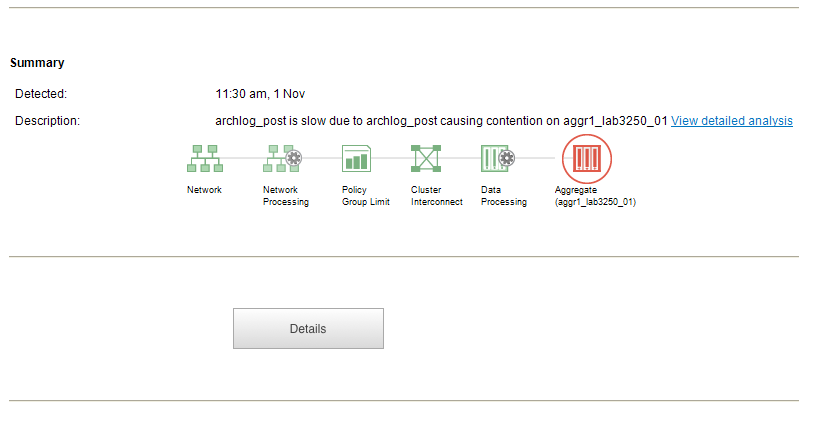
## Performance alerting

Performance alerting in OPM is significantly simplified as compared to PA. Threshold limits are automatically calculated by OPM for each workload, based on historical performance monitoring. Configuring email alerts in OPM is performed by simply adding an email address or list of email addresses (comma or semicolon separated) to OPM via the *Administration -> Configure Email Alerts* dialog from the top menu bar in the OPM web interface. The following screenshots show examples of email alerts that were created for the incidents shown in the previous example of volume performance graphs.

The first email alert example shows alerting for a Policy Group Limit incident.



The next example shows alerting for an incident in which the Aggregate was the main source of latency:



# Configuring and using the wlstats utility

The *wlstat* script gathers and displays cDOT workload statistics for real time analysis or archival purposes. The statistics are grouped and displayed on a per QoS policy group basis, in order to allow for easy per-application performance analysis when we group applications into their own QoS policy group. The data provided by the *wlstats* utility will be used for granular performance analysis at the QoS policy group level as well as at the physical node level. The *wlstats* utility is available from the NetApp communities set at the following URL:

<https://communities.netapp.com/docs/DOC-29904>

## ****Running wlstats interactively****

The *wlstats* utility can be used to view performance data in real time, and will output data in tab delimited format by default. In the TR environment, the *wlstats* utility is installed at */filers/admin/scripts/support/wlstats* on all OCUM servers. The following command usage shows the arguments that the wlstats command accepts:

Usage: wlstats -s systemname [-n nodename] [-w workload]

[-u username ] [-p password] [-f pwfile]

[-i interval] [-c count] [-a] [-l]

-s The hostname or IP of the clustered ONTAP system.

-n Optional nodename to limit the scope of collection.

-w Optional QoS policy group workload output filter.

-u The username to connect with (default is admin).

-p The password to use (will prompt if not provided).

-f A file to read the password from.

-i The interval in seconds between stats (default is 3).

-c The count of intervals (default is 1200).

-a Archive mode, which produces CSV output.

-l Limit total script runtime to interval \* count.

The output fields of the *wlstats* command are as follows:

Output Fields:

IOPS: Number of I/Os per second.

R Ops: Number of Read I/Os per second.

W Ops: Number of Write I/Os per second.

Tput: Throughput in KB/sec.

R Tput: Read throughput in KB/sec.

W TPut: Write throughput in KB/sec.

Lat: Latency.

R Lat: Read Latency.

W Lat: Write Latency.

NetLat: Latency due to network and protocol processing.

CluLat: Latency due to cluster network traversal.

DatLat: Latency due to filesystem processing.

DskLat: Latency due to disk I/O.

QoSLat: Latency due to QoS throttling.

The following is an example of running wlstats interactively:

# ./wlstats.pl -s <cluster>

Enter password for admin@<cluster>:

Policy Groups IOPS R Ops W Ops Tput R Tput W Tput Lat R Lat W Lat NetLat CluLat DatLat DskLat QoSLat

(<cluster>:all) KB/s KB/s KB/s ms ms ms ms ms ms ms ms

-total- 1804 808 996 53175 25864 27311 8.1 10.4 6.2 7.2 0.0 0.2 0.5 7.2

viking 1647 808 839 52708 25864 26845 8.9 10.4 7.3 0.0 0.0 0.2 0.5 7.9

User-Best-Effort 157 0 157 466 0 466 0.2 0 0.2 0.0 0.0 0.1 0.1 0.0

svm11pg1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

----- Tue Sep 3 16:00:20 2013 -----

Policy Groups IOPS R Ops W Ops Tput R Tput W Tput Lat R Lat W Lat NetLat CluLat DatLat DskLat QoSLat

(<cluster>:all) KB/s KB/s KB/s ms ms ms ms ms ms ms ms

-total- 1966 786 1179 52379 24954 27425 7.2 11.5 4.3 5.2 0.0 0.4 1.4 5.2

viking 1581 777 803 50581 24872 25709 8.9 11.6 6.2 0.0 0.0 0.4 1.7 6.5

User-Best-Effort 385 9 376 1798 82 1716 0.2 2.3 0.1 0.0 0.0 0.1 0.1 0.0

svm11pg1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

----- Tue Sep 3 16:00:23 2013 -----

## ****Using wlstats in archive mode****

In addition to viewing statistics in real time, the wlstats utility can archive data in CSV format. The *cdot\_stats\_harvester.sh* wrapper script is configured on TR OCUM servers to gather these statistics and save the output in CSV files that represent one hour of performance data per node. The cron job entries for this wrapper, and a command to limit retention of the output files to the most recent 30 days, are shown below:

15 \* \* \* \* svcstg\_scriptuser /filers/admin/scripts/support/cdot\_stats\_harvester.sh >> /dfm/netapp/scripts/cdot\_stats\_harvester.log 2>&1

5 5 \* \* 0 svcstg\_scriptuser find /dfm/netapp/scripts/cdotstats -mtime +30 -exec rm {} \; >/dev/null 2>&1

At this time, the *wlstats* utility is not meant to be used for full troubleshooting of performance cases with the NetApp global support center. Currently, NetApp Perfstat will continue to be the data gathering tool of choice when engaging the NetApp global support center to assist with troubleshooting performance issues.

## ****Locating the wlstats output for a node****

The *cdot\_stats\_harvester.sh* wrapper uses the *wlstats* utility and places output files in the */dfm/netapp/scripts/cdotstats* directory on each of the OCUM servers. Within this directory are subdirectories corresponding to each of the clusters, and the nodes of each cluster, being managed by the OCUM server. For example, the directory layout would be as follows for a cluster named *hive* with *nodes hive-01, hive-02, hive-03, and hive-04*:

/dfm/netapp/scripts/cdotstats/hive/hive-01

/dfm/netapp/scripts/cdotstats/hive/hive-02

/dfm/netapp/scripts/cdotstats/hive/hive-03

/dfm/netapp/scripts/cdotstats/hive/hive-04

The files in each of the above directories would be time/date stamped based on the hour in which they started recording data.

## ****Graphing wlstats output with Excel****

The *wlstats* output will be graphed by TR for historical analysis. The URL to access these graphs will be added to this document once that work has been completed by TR.

In addition, data in the *wlstats* CSV files can be manually graphed using Excel, by using the following steps:

1. Open the CSV file in Excel and select all the data.
2. In the "Insert" tab, select "Pivot Table" and insert it into a new sheet.
3. From the Pivot Table Field List, drag the "Date" field to "Row Labels".
4. From the Pivot Table Field List, drag the "Policy Groups" field to "Column Labels".
5. Drag one or more counters of interest to "Values".
6. Click on the counter in "Values", select "Value Field Settings" and choose to "Summarize value field" by "Sum" (for throughput or ops) or "Average" (for latency).
7. In the "Pivot Table Tools" tab, selct "Pivot Chart".
8. Select or unselect policy groups from the pivot table as desired.

# Performance incident management

## ****Typical steps****

The following steps should be used as a standard practice when analysing performance incidents. First, use the OPM graphs or the *wlstats* data to determine if the latency for a given volume or QoS policy group is at an acceptable level. Based on this investigation, perform the following steps:

1. If the latency is at acceptable levels, and is not higher during the incident timeframe than it was at other times, then storage is most likely not the root cause of the incident.
2. If the latency is not at acceptable levels during the incident timeframe, use the OPM graphs or *wlstats* data to determine the source of the latency.
   1. If there is latency due to cluster interconnect traffic, then indirect I/O is being performed, and this should be corrected.
   2. If the latency is due to a QoS Policy Group limit, then you must decide if the limit should be increased or if the application needs to reduce the amount of I/O being performed.
   3. If the latency is due to Aggregate utilization, then the disk drives on the system may be overloaded. In this case, one or more workloads may need to be moved to another aggregate in the cluster in order to more evenly distribute the I/O, or more disk drives may be required in the aggregate.
   4. If the latency is due to one of the CPU subsystems, then determine if the node is at or above the recommended peak values for the platform, based on the IOPs and I/O profile. If the node is above the recommended amount of I/O given the I/O Profile, then one or more workloads may need to be moved to another node.
   5. If the above steps do not provide a conclusive answer, NetApp Perfstat data should be collected and a NetApp support should be opened.

## ****Opening cases with NetApp Support****

In order to ensure a timely response from the NetApp global support center, it is recommended to only open a support case when you have the following information ready to share:

1. The storage cluster hostname, the storage node name, and storage node serial number.
2. A concise description of the problem, including the metric of interest. Some good examples of the metric of interest are as follows:
   1. NFS write latency spiked to 50ms during a certain timeframe, and the I/O workload on the system was at a level that this type of storage controller should be able to handle.
   2. A certain batch processing job used to run in 10 minutes, and last night it took 60 minutes to complete. We can see elevated NFS read latency during last night’s batch job timeframe, as compared to previous nights.
3. When did the issue start happening? Were there any other known changes in the environment that coincide with the start of this issue?
4. Which volumes on the storage controller are affected? All volumes, or only certain volumes?
5. What are the clients doing that are accessing the storage controller? For example, clients may be using the storage for storing Oracle data over NFS, storing VMware ESX data over NFS, or for storing user home directory data over CIFS.
6. What OS is being used by the clients of this storage controller? If there are any client side OS messages that are relevant, please include that information.
7. NetApp Perfstat output from the storage controller being investigated, which was gathered **during the time which this incident occurred**. See section the following section for the possible methods by which Perfstat data can be gathered. Opening a performance case with NetApp global support without having valid Perfstat output **during the timeframe of the incident** will significantly slow case resolution, and you will likely be asked to gather Perfstat data in order to continue.

## ****Using NetApp Perfstat****

### Running Perfstat interactively

If you are gathering Perfstat data at a known timeframe, it can be run in interactive mode. In this mode, the Perfstat utility will prompt you for the admin and diag passwords to use when connecting to the storage system.

### Requirements for non-interactive Perfstat with cDOT

Unlike Perfstat for 7 Mode, the Perfstat utility for cDOT requires the following configuration in order to be run non-interactively with cDOT:

1. An account named perfstat must be configured on the storage system, and SSH publickey authentication must be configured for this account.
2. The diag user must be unlocked and have a password configured.

Examples of performing both of these steps are provided in the *CLI Examples* section of this document.

### Running Perfstat with cdot\_perfstat\_loop.sh

In order to catch performance incidents that have a short duration, are intermittent, and unpredictable in terms of their timing, we have a script that will continuously gather Perfstat data in 1 minute intervals. This script is located at */filers/admin/scripts/support/cdot\_perfstat\_loop.sh* on all OCUM servers. Edit this script and set the appropriate values for *the PERFSTAT\_PATH*, *DIAGPASS*, and *SSHPRIKEY* variables as required.

In order to run this script, simply supply the cluster name as the first argument, an optional node name (to gather data from a single node only) or *all* (to gather data from all nodes in the cluster) as the second argument, and the path to an output directory as the third argument. Since this script runs until it is killed, it is recommend to run this via a *screen* session. For example:

*ocum-cli# screen bash*

*ocum-cli# /filers/admin/scripts/support/cdot\_perfstat\_loop.sh hive hive-01 /dfm/netapp/perfstat/hive*

*<CTRL-A> <CTRL-D>*

The *<CTRL-A> <CTRL-D>* key sequence will disconnect your screen session and the script will continue to run. In order to view the list of running screen sessions, use the *screen –list* command. Use *screen –r <socketid>* to reconnect to your screen session when you want to kill the script with the *kill* <PIDS> command, and exit the screen session with the *exit* command. The output file(s) to upload to the NetApp global support center will have a date/time stamp and there will be one file per hour.

Note that while this script gives us a high probability of catching short lived performance issues, there is a chance that it will miss them due to the startup/shutdown time required for each Perfstat iteration.

# Additional Resources

## NetApp peak performance by platform

The following tables list the maximum desirable IOPs values for the commonly used platforms in TR, depending on the I/O profile.

**FAS6220 controllers with 600GB SAS drives**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Max IOPs** | Read/Write Mix | 100/0 | 75/25 | 50/50 | 25/75 | 0/100 |
| Avg IO Size |  |  |  |  |  |  |
| 32k |  | 62,266 | 44,117 | 34,090 | 27,778 | 23,437 |
| 24k |  | 57,537 | 44,776 | 36,585 | 30,821 | 26,785 |
| 16k |  | 41,324 | 45,455 | 40,541 | 36,955 | 33,422 |
| 8k |  | 28,121 | 37,022 | 45,455 | 44,563 | 42,857 |
| 4k |  | 23,074 | 30,658 | 47,104 | 48,875 | 50,504 |

**FAS6220 controllers with 2TB SATA drives**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Max IOPs** | Read/Write Mix | 100/0 | 75/25 | 50/50 | 25/75 | 0/100 |
| Avg IO Size |  |  |  |  |  |  |
| 32k |  | 58,330 | 41,721 | 20,502 | 13,414 | 10,352 |
| 24k |  | 52,702 | 40,518 | 29,205 | 19,371 | 14,061 |
| 16k |  | 29,431 | 30,152 | 36,733 | 32,814 | 23,990 |
| 8k |  | 19,565 | 20,038 | 24,064 | 38,961 | 34,090 |
| 4k |  | 16,355 | 17,389 | 20,503 | 32,639 | 37,037 |

**Notes on peak performance max IOPs values:**

* Peak IOP values are per controller.
* Use this only as a guide when determining controller headroom on existing systems.
* The values assume adequate spindles are in place for workloads that are CPU limited in these configurations.
* Higher values are possible for the small random read workloads that are disk limited in these configurations.
* The values do not account for additional tasks such as snapvault, deduplication, disk scrubs, etc.

# Clustered ONTAP CLI examples

## QoS policy group configuration

### Create a new QoS policy group with a MB/s limit

qos policy-group create -policy-group <pgname> -vserver <vsname> -max-throughput <XX>MB/s

qos policy-group show

### Create a new QoS policy group with an IOPs limit

qos policy-group create -policy-group <pgname> -vserver <vsname> -max-throughput <XX>iops

qos policy-group show

### Create a QoS policy group with no limit

qos policy-group create -policy-group <pgname> -vserver <vsname> -max-throughput INF

qos policy-group show

### Modify a QoS policy group

qos policy-group modify -policy-group <pgname> -max-throughput <XX>MB/s

qos policy-group show

### Setting a QoS policy group on a volume

volume modify -vserver <vsname> -volume <volname> -qos-policy-group <pgname>

volume show -vserver <vsname> -fields qos-policy-group

## Gathering cDOT performance statistics from the CLI

### Using statistics show commands

statistic show-periodic -interval 1

statistics show-periodic -node <node> -instance cluster -interval 1

### Using qos statistics commands

qos statistics performance show

qos statistics characteristics show

qos statistics latency show

## Creating accounts for wlstats and NetApp Perfstat

### Create the wlstats account

security login role create -role wlstats -cmddirname statistics -access all -vserver <cluster\_vserver>

security login role create -role wlstats -cmddirname version -access all -vserver <cluster\_vserver>

security login create -username wlstats -application ontapi -authmethod password -role wlstats -vserver <cluster\_vserver>

security login show

security login role show

security login role show-ontapi

### Create perfstat account and setup publickey authentication

security login create -username perfstat –vserver <admin\_vserver> -application ssh -authmethod publickey –role admin

security login publickey create -username perfstat -vserver <admin\_vserver> -publickey “<ssh\_publickey>”

security login show

security login publickey show

### Unlock diag account and set password for it

security login unlock -username diag

security login password -username diag

security login show