Important Information

**ESMON**

User Manual

**Version 0.0.3| ref.notbd | Revision A0**



Information in this document is subject to change without notice and does not represent a commitment on the part of DataDirect Networks, Inc. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose other than the purchaser’s personal use without the written permission of DataDirect Networks, Inc.

**© 2017DataDirect Networks, Inc. All rights reserved.**

DataDirect Networks, the DataDirect Networks logo, DDN, DirectMon, Enterprise Fusion Architecture, EFA, ES7K, ES12K, ES14K, EXAScaler, GRIDScaler, GS7K, GS12K, GS14K, IME, IME14K, Infinite Memory Engine, Information in Motion, In-Storage Processing, MEDIAScaler, NAS Scaler, NoFS, ObjectAssure, ReACT, SFA, SFA 10000 Storage Fusion Architecture, SFA10K, SFA12K, SFA14K, SFA7700, SFX, Storage Fusion Architecture, Storage Fusion Fabric, Storage Fusion Xcelerator, SwiftCluster, WOS, the WOS logo are registered trademarks or trademarks of DataDirect Networks, Inc. All other brand and product names are trademarks of their respective holders.

DataDirect Networks makes no warranties, express or implied, including without limitation the implied warranties of merchantability and fitness for a particular purpose of any products or software. DataDirect Networks does not warrant, guarantee or make any representations regarding the use or the results of the use of any products or software in terms of correctness, accuracy, reliability, or otherwise. The entire risk as to the results and performance of the product and software are assumed by you. The exclusion of implied warranties is not permitted by some jurisdictions; this exclusion may not apply to you.

In no event will DataDirect Networks, their directors, officers, employees, or agents (collectively DataDirect Networks) be liable to you for any consequential, incidental, or indirect damages, including damages for loss of business profits, business interruption, loss of business information, and the like, arising out of the use or inability to use any DataDirect product or software even if DataDirect Networks has been advised of the possibility of such damages by you. Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, these limitations may not apply to you. DataDirect Networks liability to you for actual damages from any cause whatsoever, and regardless of the form of the action (whether in contract, tort including negligence, product liability or otherwise), is limited to the sum you paid for the DataDirect product or software.

All Products cited in this document are subject to DDN’s Limited Warranty Statement and, where applicable, the terms of DDN’s End User Software License Agreement (EULA).

The cited documents are available at: <http://www.ddn.com/support/policies/>

For archived versions of either document, please contact DDN.

October 2017

Table of Contents

[Terminology 8](#_Toc508494352)

[Collectd plugins of DDN 8](#_Toc508494353)

[Installation Server 10](#_Toc508494354)

[Monitoring Server 10](#_Toc508494355)

[Monitoring Client 10](#_Toc508494356)

[Preparing the Installation Server 11](#_Toc508494357)

[Updating the configuration file on the installation server 11](#_Toc508494358)

[Runninginstallation on the cluster 13](#_Toc508494359)

[Accessing the Monitoring Web Page 13](#_Toc508494360)

[Cluster Status Dashboard 14](#_Toc508494361)

[Lustre Status Dashboard 15](#_Toc508494362)

[Server Statistics 36](#_Toc508494363)

[SFA Physical Disk Dashboard 40](#_Toc508494364)

[SFA Virtual Disk Dashboard 43](#_Toc508494365)

[Installing stress2 RPM on Collectd Client 46](#_Toc508494366)

[Updating ConfigurationFile of Collectd Client 46](#_Toc508494367)

[Start Testing 47](#_Toc508494368)

List of Figures

[Figure 1: Grafana Login Web Page 13](#_Toc508494369)

[Figure 2: Home Dashboard 14](#_Toc508494370)

[Figure 3: Cluster Status Dashboard 15](#_Toc508494371)

[Figure 4: Lustre Statistics Dashboard 15](#_Toc508494372)

[Figure 5: Free Capacity in Total Panel 16](#_Toc508494373)

[Figure 6: Used Capacity in Total Panel 16](#_Toc508494374)

[Figure 7: Free Capacity per OST Panel 17](#_Toc508494375)

[Figure 8: Used Capacity per OST Panel 17](#_Toc508494376)

[Figure 9: Used Capacity per User Panel 18](#_Toc508494377)

[Figure 10: Used Capacity per Group Panel 18](#_Toc508494378)

[Figure 11: Free Inode Number in Total Panel 19](#_Toc508494379)

[Figure 12: Used Inode Number in Total Panel 19](#_Toc508494380)

[Figure 13: Free Inode Number per MDT Panel 20](#_Toc508494381)

[Figure 14: Used Inode Number per User Panel 20](#_Toc508494382)

[Figure 15: Used Inode Number Per Group Panel 21](#_Toc508494383)

[Figure 16: Used Inode Number per MDT Panel 21](#_Toc508494384)

[Figure 17: I/O Throughput in Total Panel 22](#_Toc508494385)

[Figure 18: I/O Throughput per OST Panel 22](#_Toc508494386)

[Figure 19: Write Throughput per OST Panel 23](#_Toc508494387)

[Figure 20: Read Throughput per OST Panel 23](#_Toc508494388)

[图21: Metadata Operation Rate in Total Panel 24](#_Toc508494389)

[Figure 22: Metadata Operation Rate Per MDT Panel 24](#_Toc508494390)

[Figure 23: Metadata Operation Rate per Client Panel 25](#_Toc508494391)

[Figure 24: Metadata Operation Rate per Type Panel 25](#_Toc508494392)

[Figure 25: Write Bulk RPC Rate per Size 26](#_Toc508494393)

[Figure 26: Size Distribution of Write Bulk RPC Panel 26](#_Toc508494394)

[Figure 27: Read Bulk RPC Rate per Size Panel 27](#_Toc508494395)

[Figure 28: Size Distribution of Read Bulk RPC Panel 27](#_Toc508494396)

[Figure 29: Distribution of Discontinuous Pages in Each Write I/O Panel 28](#_Toc508494397)

[Figure 30: Distribution of Discontinuous Pages in Each Read I/O Panel 28](#_Toc508494398)

[Figure 31: Distribution of Discontinuous Blocks in Each Write I/O Panel 29](#_Toc508494399)

[Figure 32: Distribution of Discontinuous Blocks in Each Read I/O Panel 29](#_Toc508494400)

[Figure 33: Distribution of Fragments in Each Write I/O Panel 30](#_Toc508494401)

[Figure 34: Distribution of Fragments in Each Read I/O Panel 30](#_Toc508494402)

[Figure 35: Distribution of in-flight Write I/O Number when Starting Each Write I/O Panel 30](#_Toc508494403)

[Figure 36: Distribution of in-flight Read I/O Number when Starting Each Read I/O Panel 31](#_Toc508494404)

[Figure 37: Distribution of Write I/O Time Panel 31](#_Toc508494405)

[Figure 38: Distribution of Read I/O Time Panel 32](#_Toc508494406)

[Figure 39: Distribution of Write I/O size on Disk Panel 32](#_Toc508494407)

[Figure 40: Distribution of Read I/O Size on Disk Panel 32](#_Toc508494408)

[Figure 41: Write Throughput per Client Panel 33](#_Toc508494409)

[Figure 42: Read Throughput per Client Panel 33](#_Toc508494410)

[Figure 43: I/O Throughput per Job Panel 34](#_Toc508494411)

[Figure 44: Write Throughput per Job Panel 34](#_Toc508494412)

[Figure 45: Read Throughput per Job Panel 35](#_Toc508494413)

[Figure 46: Metadata Performance per Job Panel 35](#_Toc508494414)

[Figure 47: Server Statistics Dashboard 36](#_Toc508494415)

[Figure 48: CPU Usage Panel 36](#_Toc508494416)

[Figure 49: Memory Usage Panel 37](#_Toc508494417)

[Figure 50: Disk Write Rate Panel 37](#_Toc508494418)

[Figure 51: Disk Read Rate Panel 38](#_Toc508494419)

[Figure 52: Disk Usage on Root Panel 38](#_Toc508494420)

[Figure 53: Load Panel 39](#_Toc508494421)

[Figure 54: Uptime Panel 39](#_Toc508494422)

[Figure 55: User Panel 39](#_Toc508494423)

[Figure 56: Temperature Panel 40](#_Toc508494424)

[Figure 57: SFA Physical Disk Dashboard 40](#_Toc508494425)

[Figure 58: I/O Performance on Physical DiskPanel 40](#_Toc508494426)

[Figure 59: IOPS on Physical Disk Panel 41](#_Toc508494427)

[Figure 60: Bytes per I/O on Physical Disk Panel 41](#_Toc508494428)

[Figure 61: Write Performance on Physical Disk Panel 42](#_Toc508494429)

[Figure 62: Write I/O Size Samples on Physical Disk Panel 42](#_Toc508494430)

[Figure 63: Write Latency Samples on Physical Disk Panel 42](#_Toc508494431)

[Figure 64: SFA Virtual Disk Dashboard 43](#_Toc508494432)

[Figure 65: I/O Performanceon Virtual DiskPanel 43](#_Toc508494433)

[Figure 66: I/O Operations per Secondon Virtual DiskPanel 44](#_Toc508494434)

[Figure 67: Bytes per I/O on Virtual DiskPanel 44](#_Toc508494435)

[Figure 68: Write Performance on Virtual DiskPanel 44](#_Toc508494436)

[Figure 69: Write I/O Size Samples on Virtual Disk Panel 45](#_Toc508494437)

[Figure 70: Write Latency Samples on Virtual Disk Panel 45](#_Toc508494438)

[Figure 71：Influxdb Query Time 48](#_Toc508494439)

[Figure 72：Read throughput per Job stress testing 49](#_Toc508494440)

# Introduction to DDN EXAScaler Monitoring System

*ESMON* is a monitoring system which can collect system statistics of DDN EXAScaler for performance monitoring and analysis. It is based on multiple widely used open-source software. Some extra plugins have been developed by DDN for enhancement.

One of the main components of *ESMON* is **Collectd**. **Collectd** is a daemon which collects system performance statistics periodically and provides mechanisms to store the values in a variety of ways. *ESMON* is based on the open-source **Collectd**, yet includes more plugins, such as Lustre, GPFS, Ganglia, Nagios, Stress, Zabbix and soon.

## Terminology

|  |  |
| --- | --- |
| **ESMON** | Abbreviation for DDN EXAScaler Monitoring System. |
| **DDN EXAScaler** | Software stack developed by DDN to overcome the toughest storage and data management challenges in extreme, data-intensive environments. |
| **Installation Server** | The server on which the installation process is triggered. |
| **Monitoring Server** | The server on which the database (Influxdb) and web server (Grafana) of the monitoring system will run. |
| **Monitoring Client(s)** | The servers from which the monitor system will collect metrics from. The metrics includes information about CPU, memory, Lustre, SFA storage, and so on. A collectd daemon will run on each monitoring client. |
| **DDN IME** | DDN’s Infinite Memory Engine (IME) is a flash-native, software-defined, storage cache that streamlines application IO, eliminating system bottlenecks. |
| **Lustre** | The Lustre file system is an open-source, parallel file system that supports many requirements of leadership class HPC simulation environments. |

## Collectd plugins of DDN

Several additional plugins are added to **Collectd** in ESMON to support various functions.

* **Filedata plugin**: The Filedata plugin is able to collect data by reading and parsing a set of files. An XML-formatted definition file is required for the Filedata plugin to understand which files to read and how to parse these files. The most common usage of the Filedata plugin is to collect metrics through /proc interfaces of a running Lustre system.
* **Ganglia plugin**: The Ganglia plugin can send metrics collected by a сollectd client daemon to Ganglia server.
* **GPFS plugin**: The GPFS plugin can collect performance information through “mmpmon” commands provided by GPFS. The GPFS plugin shares the same definition file format with the Filedataplugin. The configuration format of GPFS in collectd.conf is also similar tothe Filedata plugin.
* **IME Plugin**: The IME plugin can collect performance information from DDN IME. Like the GPFS plugin, the IME plugin shares the similar definition file format and configuration format with the Filedata plugin.
* **SSH plugin**: The SSH plugin is able to collect metrics by running commands on remote hosts by using SSH connections. The SSH plugin is used to collect metrics from DDN SFA Storage. Like the GPFS plugin and the IME plugin, the IME plugin shares the similar definition file format and configuration format with the Filedata plugin.
* **Stress plugin**: The Stress plugin can push a large amount of metrics to server from collectd client in order to benchmark the performance of the collecting system under high pressure.
* **Zabbix plugin**: The Zabbixplugin is able to send metrics from collectdto Zabbix system.

# Installation Requirements

## Installation Server

* **OS distribution:**CentOS7/RHEL7
* **Free disk space:**>500MB.TheInstallation Server will save all installation logs to the /var/log/esmon\_install directory, which requires some free disk space.
* **Network:** The Installation Server must be able to start SSH connections to the Monitoring Server and Monitoring Clients without a password prompt.
* **ESMON ISO image:** The ESMON ISO image must be available on the Installation Server.

## Monitoring Server

* **OS distribution:** CentOS7/RHEL7
* **Free disk space:**> 5GB. Influxdb will be running on this server. More disk space is required to keep more data in Influxdb
* **Network:** SSHD should be running on the Monitoring Server. The Installation Server must be able to connect to the Monitoring Server without a password prompt.

## Monitoring Client

* **OS distribution:** CentOS7/RHEL7 or CentOS6/RHEL6
* **Free disk space:**> 200MB. The installation server will save necessary RPMs in directory /var/log/esmon\_install, which requires some free disk space.
* **Network:** SSHD should be running on the Monitoring Client. The Installation Server must be able to connect to the Monitoring Client without a password prompt.
* **EXAScaler version:** EXAScaler 2.x or EXAScaler 3.x

# Installation Process

## Preparing the Installation Server

1. Copy the ESMON ISO image file to the Installation Server, for example, to /ISOs/esmon.iso.
2. Mount the ESMON ISO image:

mount -o loop /ISOs/esmon.iso /media

1. Install the ESMON RPM on the installation server:

rpm -ivh /media/RPMS/rhel7/esmon\*.rpm

## Updating the configuration file on the installation server

After the ESMON RPM has been installed on the Installation Server, update the configuration file/etc/esmon\_install.conf, which includes all the necessary information for installation. Define the following parameters:

* **iso\_path** — The path where the ESMON ISO image is saved.
* **clients\_reinstall** — Define whether to reinstall (**true**) ESMON clients or not (**false**).
* **collect\_interval** — The interval seconds to collect data point on ESMON clients.
* **continuous\_query\_interval** — The interval of continuous query. The value of **continuous\_query\_interval \* collect\_interval** is the real interval seconds between two adjacent data points of each continuous query.
* **lustre\_exp\_ost** — Define whether to enable (**true**) or disable (**false**) metrics collection of export information of Lustre OST, usually disabled on Lustre file systems with large number of clients to avoid flood of metrics.
* **lustre\_exp\_mdt** — Define whether to enable (**true**) or disable (**false**) metrics collection of export information of Lustre MDT, usually disabled on Lustre file systems with large number of clients to avoid flood of metrics.
* **lustre\_default\_version** — The default Lustre version to use if the Lustre RPMs installed on the ESMON client is not with the supported version. The current supported values of the parameter are **es2**, **es3**, **es4** and **error**. If the parameter **error** is configured, failure will be raised when a ESMON client is using a unsupported Lustre version.
* In the section **ssh\_hosts**, specify details necessary to log in to the Monitoring Server and to each Monitoring Client using SSH connection:
  + **host\_id** — The unique ID of the host. Two hosts should not share the same **host\_id**.
  + **Hostname** — The host name to use when connecting to the host using SSH.
  + **ssh\_identity\_file** — The SSH key file used for connecting to the host. The parameter **ssh\_identity\_file** can be omitted if the default SSH identity file works.

Note: **host\_id** and **hostname** can be different for a host, because there can be multiple ways to connect to the same host.

* In the section **client\_hosts**, specify information about all of the hosts where ESMON client packages should be installed and configured:
  + **host\_id** — The unique ID of the host. Two hosts should not share the same **host\_id**.
  + **lustre\_oss** —Define whether to enable (**true**) or disable (**false**) metrics collection of Lustre OSS.
  + **lustre\_mds** —Define whether to enable (**true**) or disable (**false**) metrics collection of Lustre MDS.
  + **ime**—Define whether to enable (**true**) or disable (**false**) metrics collection of DDN IME.
* In the section **server\_hosts**, specify information about all of the hosts where ESMON server packages should be installed and configured:
  + **host\_id** — The unique ID of the host.
  + **reinstall** — Define whether to reinstall (**true**) ESMON server or not (**false**). If the parameter is set to **false**, both **drop\_database** and **rase\_influxdb** will be ignored.
  + **drop\_database** — If the parameter is set to **true**, the ESMON database in Influxdb will be dropped. If the parameter is set to **false**, the ESMON database in Influxdb will be kept as it is.
  + **erase\_influxdb** — If the parameter is enabled (set to **true**), all the data and metadata of Influxdb will be completely erased. By enabling **erase\_influxdb**, some corruption problems of Influxdb could be fixed. If the parameter is disabled (set to **false**), the data and metadata of Influxdb will not be completely erased.

**Important: erase\_influxdb** and **drop\_database**should only be enabled when the data or metadata in Influxdb is not needed anymore.

Below is an example of /etc/esmon\_install.conf:

**Example:**

iso\_path: /work/ISOs/esmon.iso

ssh\_hosts:

- host\_id: Monitoring-Server

hostname: Monitoring-Server

ssh\_identity\_file: /root/.ssh/id\_rsa

- host\_id: Monitoring-Client1

hostname: Monitoring-Client1

ssh\_identity\_file: /root/.ssh/id\_rsa

- host\_id: Monitoring-Client2

hostname: Monitoring-Client2

ssh\_identity\_file: /root/.ssh/id\_rsa

clients\_reinstall: true

collect\_interval: 60

continuous\_query\_interval: 4

lustre\_default\_version: es3

lustre\_exp\_ost: false

lustre\_exp\_mdt: false

client\_hosts:

- host\_id: Monitoring-Client1

lustre\_oss: true

lustre\_mds: true

ime: false

sfas:

- name: OST

controller0\_host: 10.0.0.1

controller1\_host: 10.0.0.2

infiniband: false

- host\_id: Monitoring-Client2

lustre\_oss: false

lustre\_mds: true

ime: false

server\_host:

host\_id: Monitoring-Server

reinstall: true

drop\_database: true

erase\_influxdb: true

## Runninginstallation on the cluster

After the */etc/esmon\_install.conf*file has been updated correctly on the Installation Server, run the following command to start the installation on the cluster:

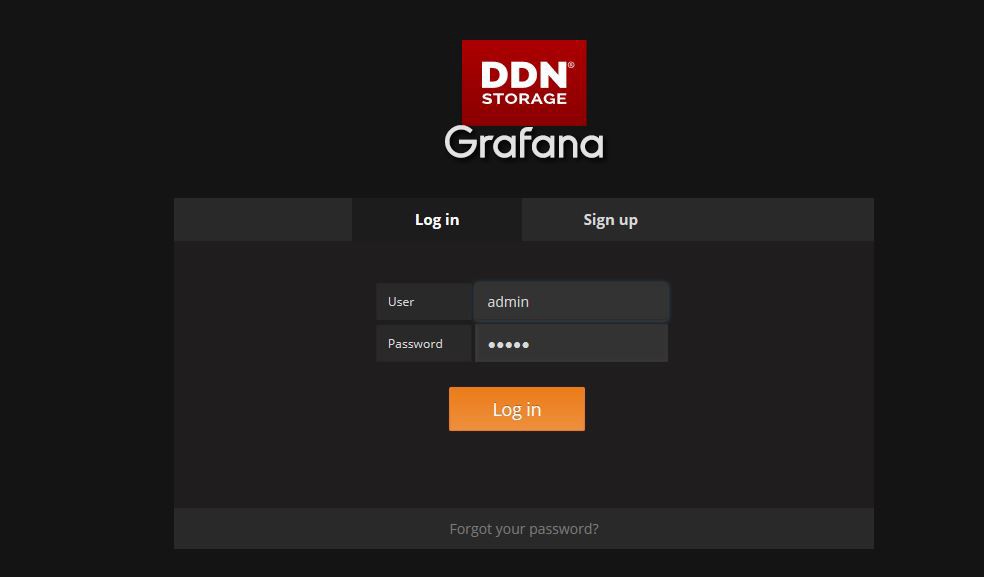
esmon\_install

All the logs which are useful for debugging are saved under /var/log/esmon\_install directory of the Installation Server.

## Accessing the Monitoring Web Page

The Grafana service is started on the Monitoring Server automatically. The default HTTP port is 3000. A login web page will beshown through that port (see Figure 1 below). The default user and password are both “admin”.

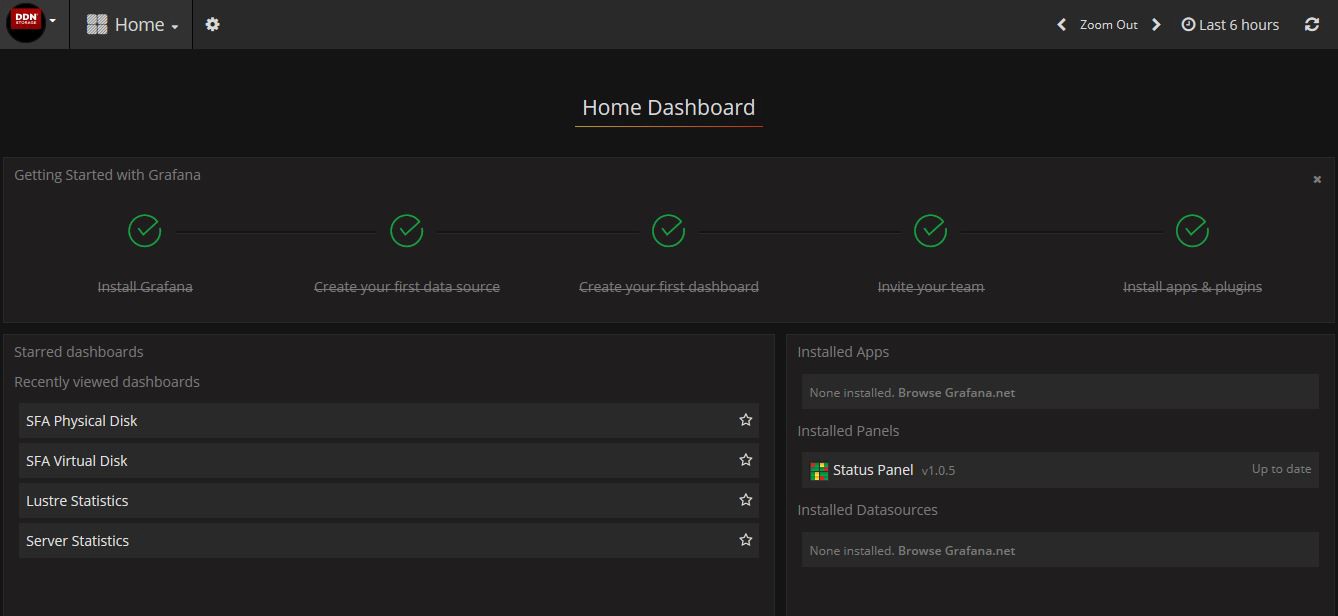
Figure 1: Grafana Login Web Page



# Dashboards

From the Home dashboard (see Figure 2) different dashboards can be chosen from to view different metrics collected by ESMON.

Figure 2: Home Dashboard

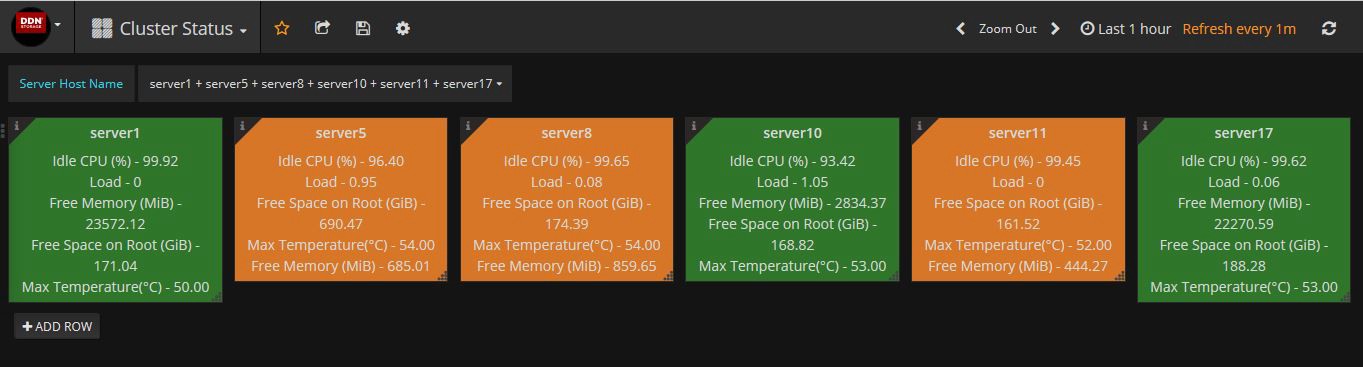


## Cluster Status Dashboard

The **Cluster Status**dashboard (see Figure 3below) shows a summarized status of the servers in the cluster. The background color of panels show the servers’ working status:

* If the color of the panel is green, it means the server is under normal condition.
* If the color of the panel is yellow, it means the server is under warning status due to one or more of the following conditions:
  + Idle CPU is less than 20%
  + Load is higher than 5
  + Free memory is less than 1000 MiB
  + Free space of “/” is less than 10 GiB
* If the color of the panel is red, it means the server is under critical status due to one or more of the following conditions:
  + Idle CPU is less than 5%
  + Load is higher than 10
  + Free space of “/” is less than 1 GiB
  + Free memory is less than 100 MiB

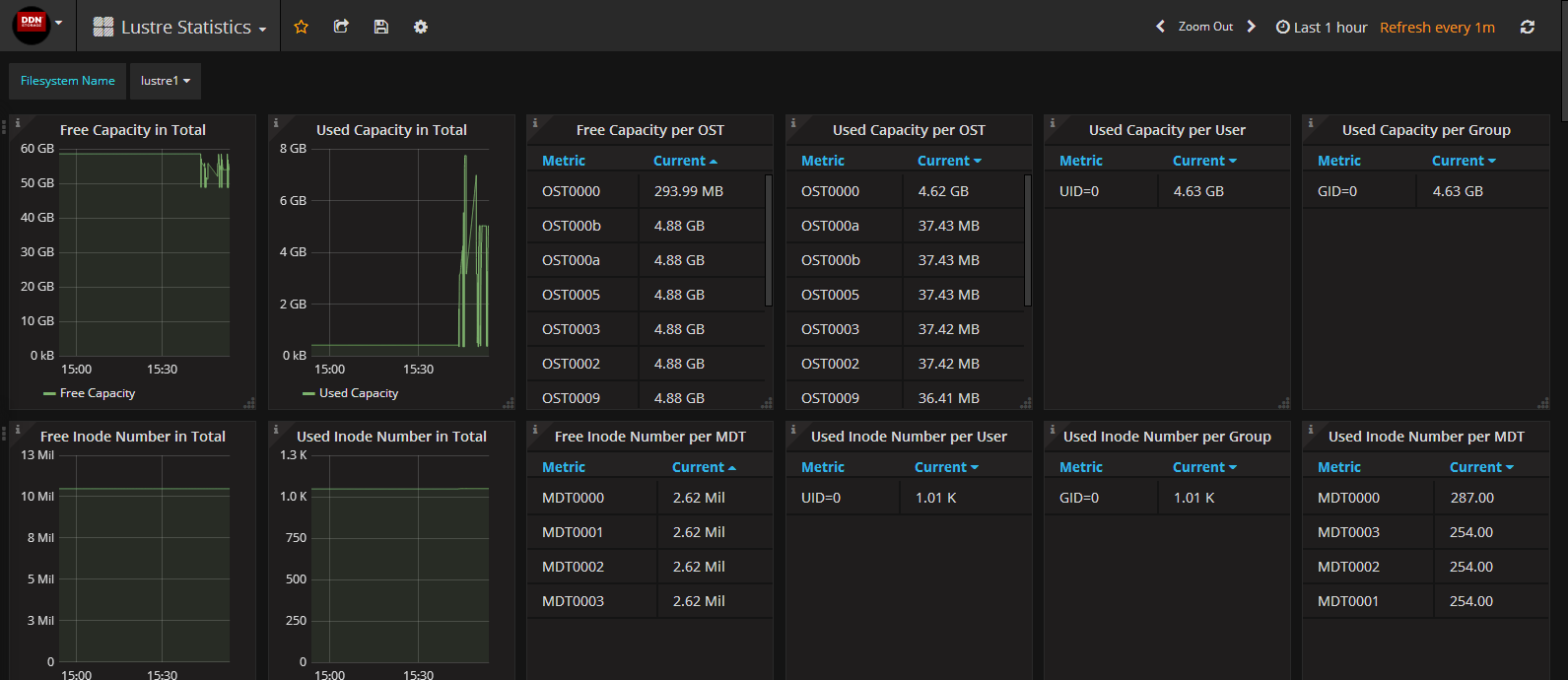
Figure 3: Cluster Status Dashboard



## Lustre Status Dashboard

The Lustre Statistics dashboard (Figure 4) show metrics of Lustre file systems.

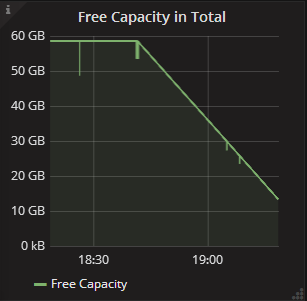
Figure 4: Lustre Statistics Dashboard



Following pictures are some of the panels in *Lustre Statistics* Dashboard

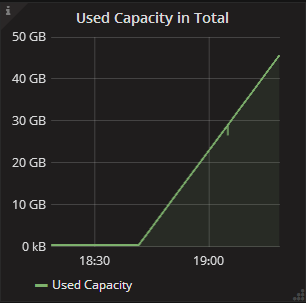
* The Free Capacity in Total (Figure 5) shows how much free capacity remained in the Lustre Filesystem. The test case used in the figure is running “dd if=/dev/zero of=/mnt/lustre/file bs=1M” from about 18:40, and it shows that the free capacity is being consumed at a speed of about 20MB/s.

Figure 5: Free Capacity in Total Panel



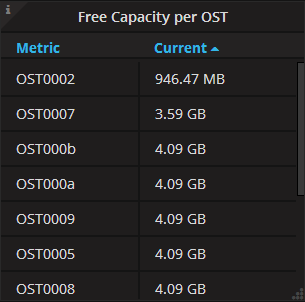
* The Used Capacity in Total panel (Figure 6) shows how much capacity used in total in the Lustre Filesystem. The test case used in the figure is running “dd if=/dev/zero of=/mnt/lustre/file bs=1M” from about 18:40, and it can be seen from the figure that the used capacity has increased at about 20 MB/s.

Figure 6: Used Capacity in Total Panel



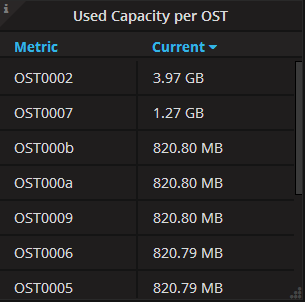
The Free Capacity per OST panel (Figure 7) shows how much free capacity per OST remained in the Lustre Filesystem. As shown in the figure, OST0002 free capacity is 946.47MB， OST0007 free capacity is 3.59GB，the free capacity of the left OSTs is all 4.09G. By clicking on the “Current”, the current free capacity can be sorted in the ascending of descending order.

Figure 7: Free Capacity per OST Panel



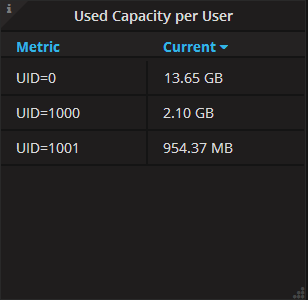
The Used Capacity per OST panel (Figure 8) shows how much capacity per OST used in the Lustre Filesystem. As shown in the figure, OST0002 used capacity is 3.97GB， OST0007 used capacity is 1.27GB，the used capacity of the left OSTs is 820.8MB。By clicking on the “Current”, the current used capacity can be sorted in the ascending of descending order.

Figure 8: Used Capacity per OST Panel



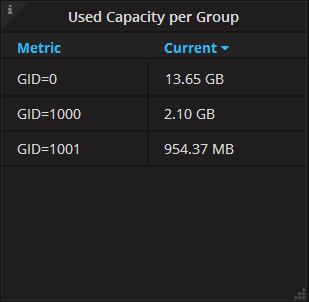
The Used Capacity per User panel (Figure 9) shows how much capacity per user used in the Lustre Filesystem. As shown in the figure, the current used capacity of the user (UID=0) is 13.65GB; The current used capacity of the user (UID=1000) is 2.10GB; The current used capacity of the user (UID=1001) is 954.37MB。

Figure 9: Used Capacity per User Panel



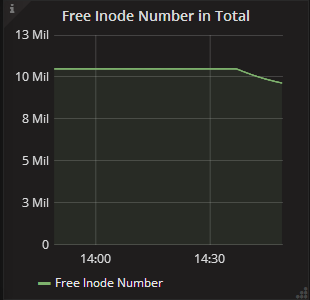
The Used Capacity per Group panel (Figure 10) shows how much capacity per group used in the Lustre Filesystem. As shown in the figure, the current used capacity of the group (GID=0) is 13.65GB; The current used capacity of the group (GID=1000) is 2.10GB; The current used capacity of the group (GID=1001) is 954.37MB。

Figure 10: Used Capacity per Group Panel



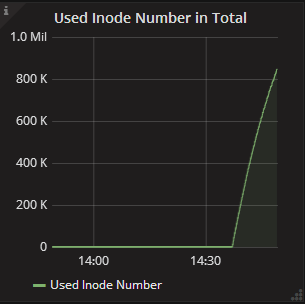
The Free Inode Number in Total panel (Figure 11) shows the variation on time of free inode number in total in the Lustre Filesystem. The test case used in the figure is running“mdtest–C –n 900000 –d /mnt/lustre/mdtest/” from about 14:35, from the figure it can be seen that from that time on the free inode number is decreased and exhausted at a speed of about 1100 Ops (Operation per Second).

Figure 11: Free Inode Number in Total Panel



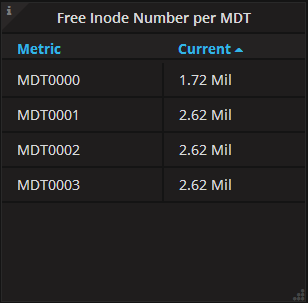
The Used Inode Number in Total panel (Figure 12) show the variation on time of used inode number in total in the Lustre Filesystem. The test case used in the figure is running “mdtest–C –n 900000 –d /mnt/lustre/mdtest/” from about 14:35, from the figure it can be seen that the used inode number is increased in a speed of about 1100 Ops (Operation per Second).

Figure 12: Used Inode Number in Total Panel



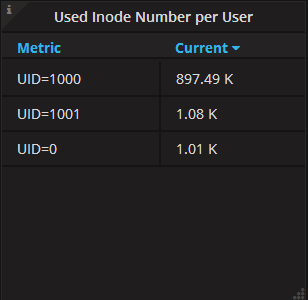
The Free Inode Number per MDT panel (Figure 13) shows how much inode number per MDT remained in the Lustre Filesystem. As shown in the figure, MDT0000 free inode number is 1.72Mil; the free inode number of all other MDTs is 2.62 Mil.By clicking on the “Current”, the current free inode number per MDT in the system can be sorted in the ascending of descending order.

Figure 13: Free Inode Number per MDT Panel



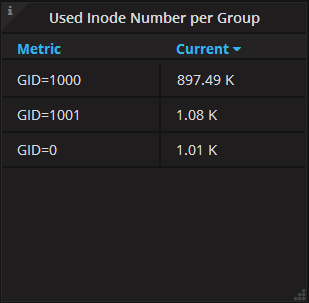
The Used Inode Number per User panel (Figure 14) shows how much inode number per user used in the Lustre Filesystem. As shown in the figure, the used inode number of the user (UID=1000)is 897.49K;the used inode number of the user (UID=1001) is 1.08K；the used inode number of the user (UID=0) is 1.01K. By clicking on the “Current”, the current used inode number per user in the system can be sorted in the ascending of descending order.

Figure 14: Used Inode Number per User Panel



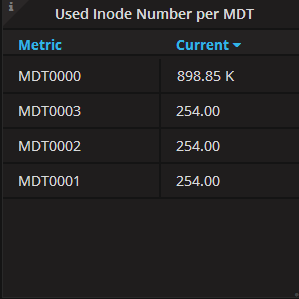
The Used Inode Number per Group panel (Figure 15) shows how much inode number per group used in the Lustre Filesystem. As shown in the figure, the used inode number of the group (GID=1000) is 897.49K; the used inode number of the group (GID=1001) is 1.08K；the used inode number of the group (GID=0) is 1.01K. By clicking on the “Current”, the current used inode number per group in the system can be sorted in the ascending of descending order.

Figure 15: Used Inode Number Per Group Panel



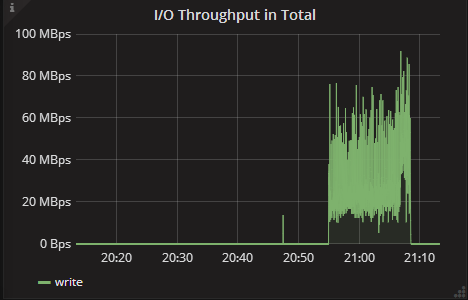
The Used Inode Number per MDT (Figure 16) shows the inode number per MDT used in the Lustre Filesystem.As shown in the figure, MDT0000 used inode number is 898.85K，MDT0001 is 254.

Figure 16: Used Inode Number per MDT Panel



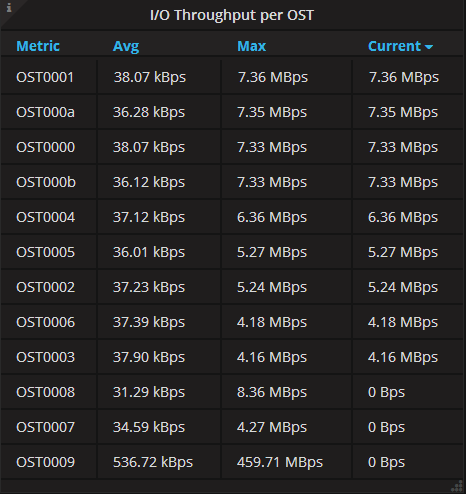
I/O Throughput in Total panel (Figure 17) shows the variation on time of I/O throughput in total in the Lustre Filesystem.

Figure 17: I/O Throughput in Total Panel



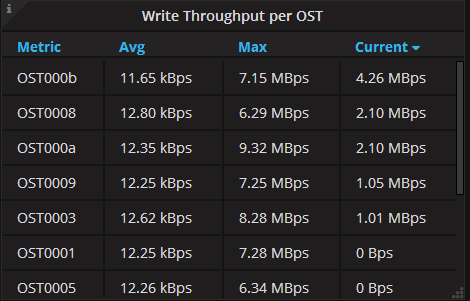
The I/O Throughput per OST panel (Figure 18)show the information of I/O throughput per OST in the LustreFilesystem. It includes the average/max/current values.

Figure 18: I/O Throughput per OST Panel



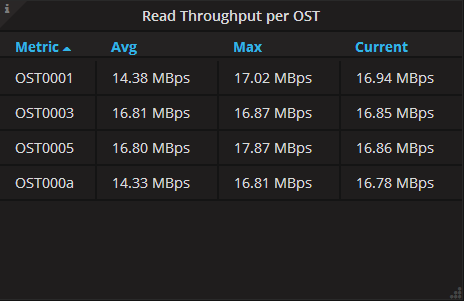
The Write Throughput per OST panel (Figure 19) shows the information of the write throughput per OST in the Lustre Filesystem. It includes the avgerage/max/current values.

Figure 19: Write Throughput per OST Panel



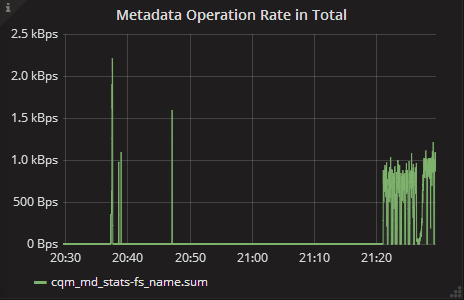
The Read Throughput per OST Panel (Figure 20) shows the information of the read throughput per OST in the Lustre Filesystem. It includes average/max/current values.

Figure 20: Read Throughput per OST Panel



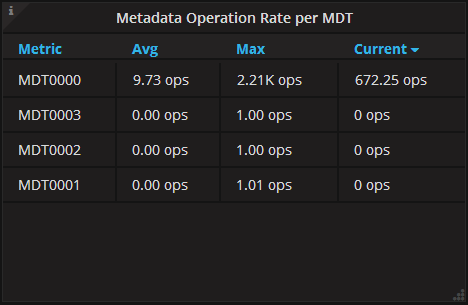
The Metadata Operation Rate in Total panel (Figure 21) shows the variation on time of metadata operation rate in total in the Lustre Filesystem, the unit is Ops, Operation Per Second.

图21: Metadata Operation Rate in Total Panel



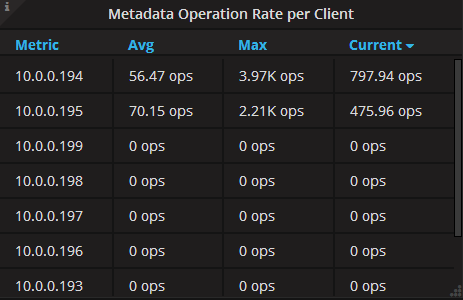
The Metadata Operation Rate per MDT panel (Figure 22) shows the metric information of the metadata operation rate per MDT in the Lustre Filesystem，the unit is Ops, Operation Per Second. It includes the average/max/current values.

Figure 22: Metadata Operation Rate Per MDT Panel



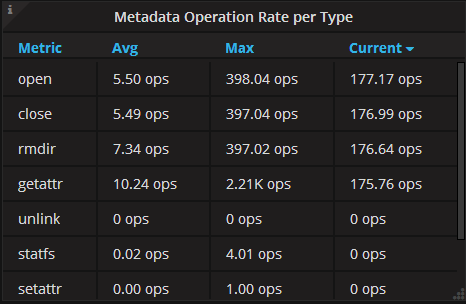
The Metadata Operation Rate per Client panel (Figure 23) shows the metric information of the metadata operation rate per client in the LustreFilesystem, the unit is Ops, Operation Per Second. It includes average/max/current values.

Figure 23: Metadata Operation Rate per Client Panel



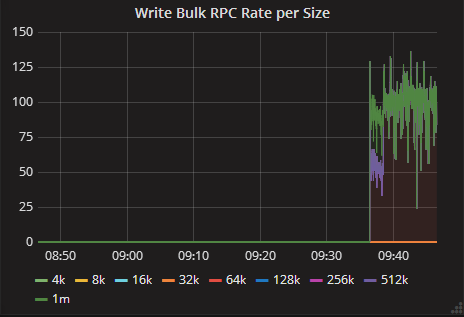
The Metadata Operation Rate per Type Panel (Figure 24) shows the metric information of the metadata operation rate per type in the LustreFilesystem, the unit is Ops, Operation Per Second. It includes average/max/current values. The current test case used is the operations that remove all files under a directory.

Figure 24: Metadata Operation Rate per Type Panel



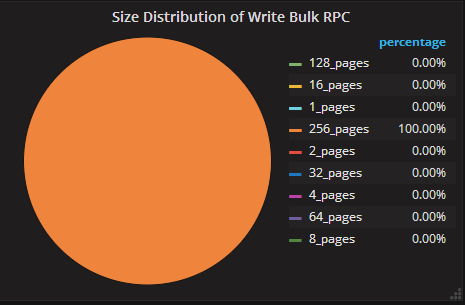
The Write Bulk RPC Rate per Size Panel (Figure 25) shows the variation on time of the write bulk RPC rate with different size in the Lustre Filesystem. Lustre statistics the different bulk RPC size varying from 4k to 16M. The figure blew shows the information of write RPC Rate with different bulk size. The used test case generated the collected information is that two client run ”dd if=/dev/zero of=/mnt/lustre/test1 bs=1M oflag=direct”, “dd if=/dev/zero of=/mnt/lustre/test2 bs=64k oflag=direct”, respectively.

Figure 25: Write Bulk RPC Rate per Size



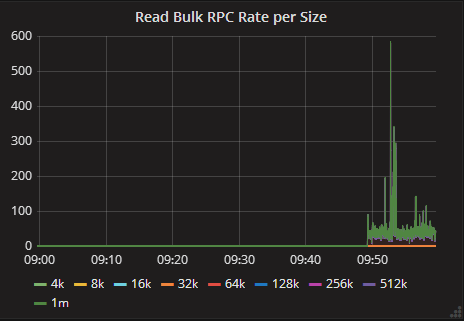
The Size Distribution of Write Bulk RPC panel (Figure 26) shows the ratio information of the write bulk RPC with different bulk size in the Lustre Filesystem. As shown in the figure, the percentage of total for the number of the write bulk RPC number with 256 pages is 100%.

Figure 26: Size Distribution of Write Bulk RPC Panel



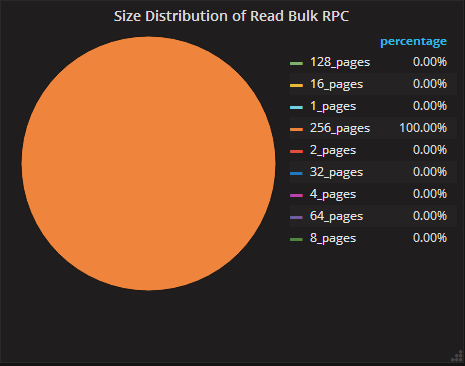
The Read Bulk RPC Rate per Size panel (Figure 27) shows the variation on time of the read bulk RPC rate per size in the Lustre Filesystem. Lustre statistics the different bulk RPC size information varying from 4K to 16M (the max. RPC bulk I/O size in Lustre is 16MB). The figure blew shows the read RPC rate with different bulk I/O size. The used test case to generate the collected information is that two clients run “dd if=/mnt/lustre/test1 of=/dev/zero bs=1M iflag=direct” and “dd if=/mnt/lustre/test2 of=/dev/zero bs=64k iflag=direct”, respectively.

Figure 27: Read Bulk RPC Rate per Size Panel



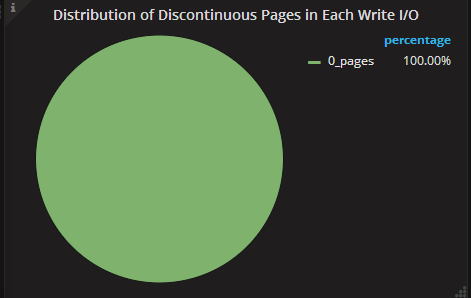
The Size Distribution of Read Bulk RPC Panel (Figure 28) shows the ratio information of read bulk RPC with different bulk I/O size in the Lustre Filesystem. As shown in the figure, the percentage of total for the number of the read bulk RPC number with 256 pages is 100% where the current used test case is running”dd if=/mnt/lustre/file of=/dev/zero bs=1M”.

Figure 28: Size Distribution of Read Bulk RPC Panel



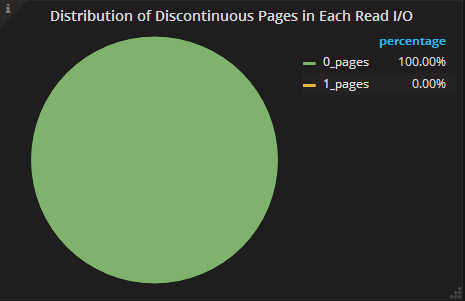
In each Lustre I/O, If the next page to be written or read in the I/O is not with the next offset, that page is a discontinuous page. There could be multiple discontinuous pages in an I/O. I/Os with less discontinuous pages are more friendly to OSDs and underlying disk system will obtain much better performance. The Distribution of Discontinuous Pages in Each Write I/O (Figure 29) shows the ratio information of the discontinuous pages in each write I/O in the Lustre Filesystem. As shown in the figure, the percentage of total for discontinuous pages “0\_pages” is 100%, which means all pages are continuous.

Figure 29: Distribution of Discontinuous Pages in Each Write I/O Panel



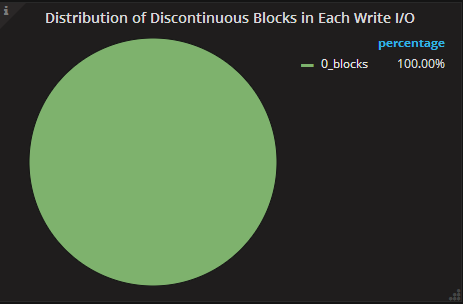
The Distribution of Discontinuous Pages in Each Read I/O panel (Figure 30) shows the ratio information of discontinuous pages in each read I/O in the Lustre Filesystem. As shown in the figure, the percentage of total for discontinuous pages “0\_pages” is 100%, which means all pages are continuous.

Figure 30: Distribution of Discontinuous Pages in Each Read I/O Panel



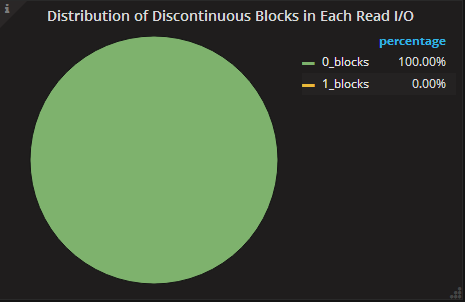
The Distribution of Discontinuous Blocks (Figure 31) shows the ratio information of the discontinuous blocks in each write I/O in the Lustre Filesystem. In each Lustre read/write I/O, the meaning of discontinuous blocks is similar to discontinuous pages. How many pages a block contains is decided by underlying filesystem (ldiskfs).If an I/O has discontinuous blocks, it must exist discontinuous pages; But the converse is not necessarily true. As shown in the figure, the percentage of total for write discontinuous blocks “0\_blocks” is 100%, which means nearly all write I/O are continuous.

Figure 31: Distribution of Discontinuous Blocks in Each Write I/O Panel



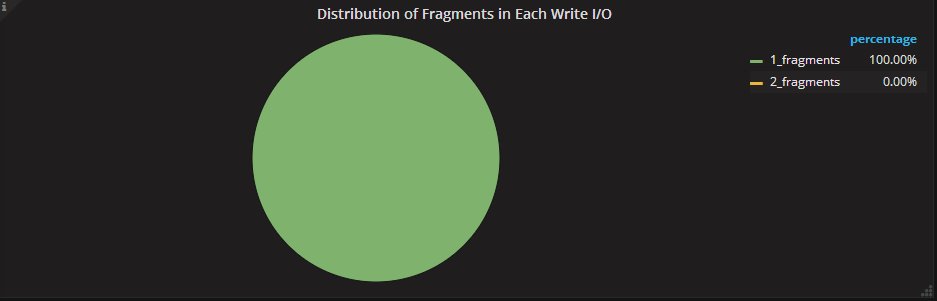
The Distribution of Discontinuous Blocks in Each Read I/O panel (Figure 32) shows the ratio information of discontinuous blocks in Each Read I/O in the Lustre Filesystem. As shown in the figure, the percentage of total for discontinuous blocks “0\_blocks” number is 100%, and it means that nearly all read I/Os are not split and continuous.

Figure 32: Distribution of Discontinuous Blocks in Each Read I/O Panel



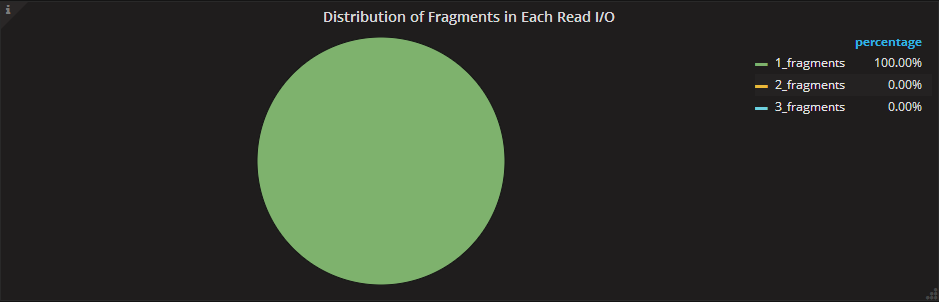
When submit read/write I/O to the underlying disk system, due to various reasons (i.e. the page number of single I/O is too big), a I/O sent by Lustre OSD may be split into multiple disk I/Os. The Distribution of Fragments in Each Write I/O panel (Figure 33) shows the ratio information of the number that a write I/O split into disk I/Os in the Lustre Filesystem. As shown in the figure, Fragment “1” means the I/O is not split. The percentage of total for the number of “1\_fragments” is 100%, which means that nearly all write I/Os are not split and all of them are continuous; “2\_fragments” represents that Lustre write I/O is split into two disk block I/Os, and the percentage of total is 0%.

Figure 33: Distribution of Fragments in Each Write I/O Panel



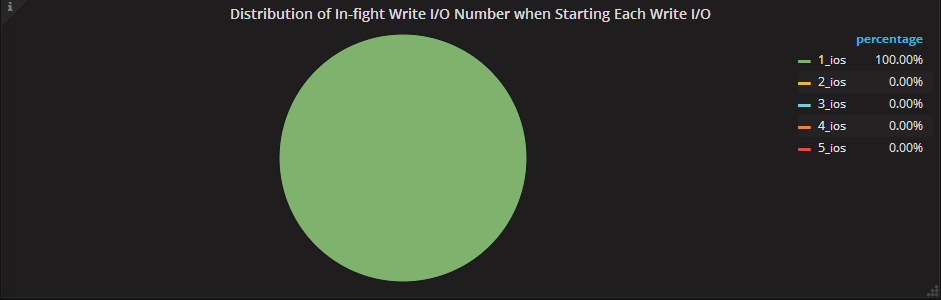
The Distribution of Fragments in Each Read I/O panel (Figure 34) shows the ratio information of the number that a read I/O split into disk I/O in the Lustre Filesystem. As show in the figure, Fragment “1” means the I/O is not split. The percentage of total for the number of “1\_fragments” is 100%, which means that nearly all read I/Os are not split and all of them are continuous; “2\_fragments” represents that Lustre read I/O is split into two disk block I/Os, and the percentage of total is 0%.

Figure 34: Distribution of Fragments in Each Read I/O Panel



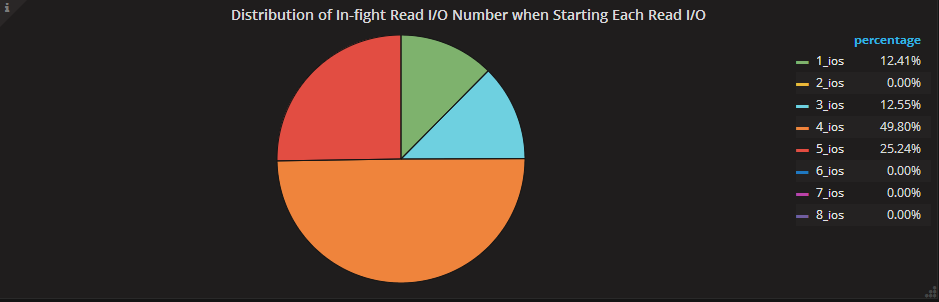
The Distribution of in-flight Write I/O Number when Starting Each Write I/O panel (Figure 35) shows the ratio information of current disk write I/O number that OSD has already submitted when starting each write I/O in the Lustre Filesystem. As shown in the figure, ”1\_ios” represents the number of current running disk I/O is 1, and the percentage of total is 100%.

Figure 35: Distribution of in-flight Write I/O Number when Starting Each Write I/O Panel



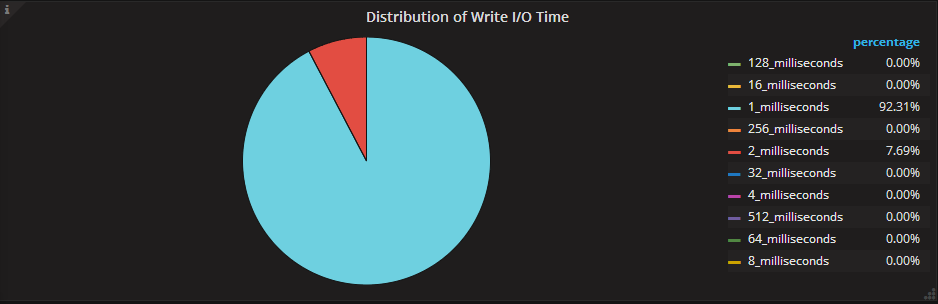
The Distribution of in-flight Read I/O Number when Starting Each Read I/O panel (Figure 36) shows the ratio information of current disk read I/O number that OSD has already submitted when starting each read I/O in the Lustre Filesystem. As shown in the figure, “1\_ios” represents the number of current running disk I/Os is 1, and the percentage of total is 12.42%; “3\_ios” represents the current number of pending disk I/Os is 3, and the percentage of total is 12.55%; “4\_ios” represents the current number of pending disk I/Os is 4, the percentage of total in 49.80%.

Figure 36: Distribution of in-flight Read I/O Number when Starting Each Read I/O Panel



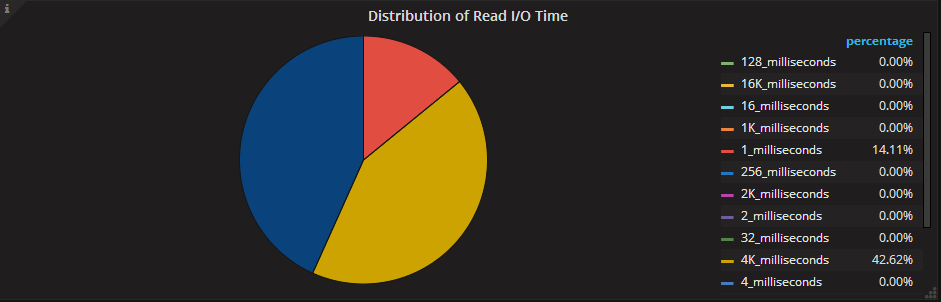
The Distribution of Write I/O Time panel (Figure 37) shows the ratio information of current OSD write I/O time in the Lustre Filesystem. As shown in the figure, “1\_milliseconds” represents the percentage of total for the I/O number that I/O time is less than 1 millisecond, and the value is 92.31%;“2\_milliseconds” represents the percentage of total for the I/O number that I/O time is between 1 millisecond and 2 milliseconds, and its value is 7.69%.

Figure 37: Distribution of Write I/O Time Panel



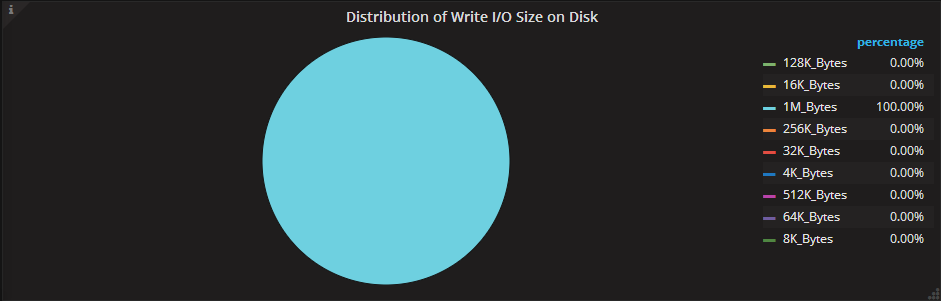
The Distribution of Read I/O Time Panel (Figure 38) shows the ratio information of current OSD read I/O time in the Lustre Filesystem. As shown in the figure, “1\_milliseconds” represents the percentage of total for the I/O number that I/O time is less than 1 milliseconds, and its value is 14.11%; “4K\_milliseconds” represents the percentage of total for the I/O number which I/O time is between 2K milliseconds and 4K milliseconds, and its value is 42.62%.

Figure 38: Distribution of Read I/O Time Panel



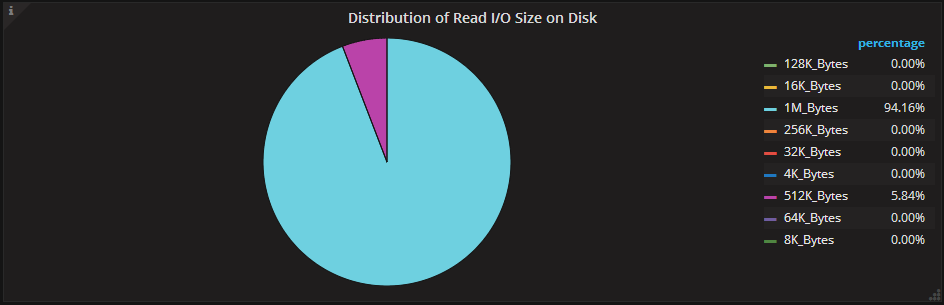
The distribution of Write I/O size on Disk Panel (Figure 39) shows the ratio information of OSD write I/O size distribution in the Lustre Filesystem. As shown in the figure, “1M\_Bytes” represents the percentage of total for I/O count with disk I/O size between 512K to 1M bytes, and its value is 100%.

Figure 39: Distribution of Write I/O size on Disk Panel



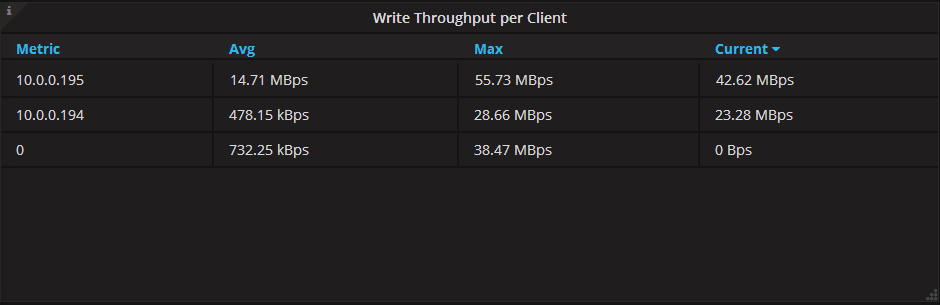
The Distribution of Read I/O Size on Disk panel (Figure 40) shows the ratio information of OSD read I/O size显示了distribution in the Lustre Filesystem. As shown in the figure, “1M\_Bytes” represents the percentage of total for I/O count with disk I/O size between 512K to 1M bytes, and its value is 100%; “512K\_Bytes” represents the percentage of total for I/O count with disk I/O size between 256K and 512K bytes, and its value is 5.84%.

Figure 40: Distribution of Read I/O Size on Disk Panel



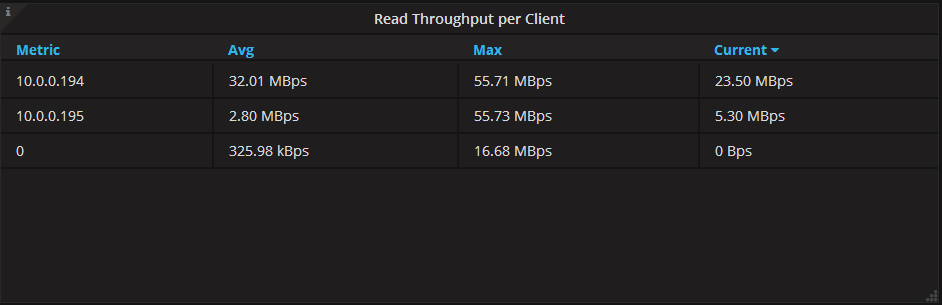
The Write Throughput per Client panel（Figure 41）shows the metric information of the write throughput显示了per client in the Lustre Filesystem. It includes average/max/current values. As shown in the figure, the average/max/current values of the write throughput for the client with IP address “10.0.0.195” are 14.71MBps/55.73MBps/42.62MBps, respectively.

Figure 41: Write Throughput per Client Panel



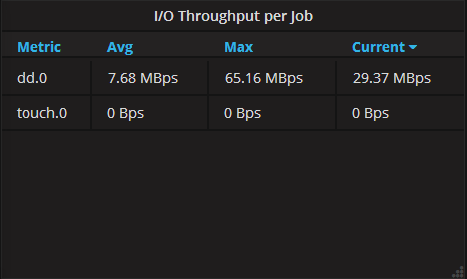
The Read Throughput per Client panel (Figure 42) shows the metric information of the read throughput per client in the Lustre Filesystem. It includes average/max/current values. As shown in the figure, the average/max/current values of the read throughput for the client with IP address “10.0.0.194” are 32.01MBps/55.71MBps/23.50MBps.

Figure 42: Read Throughput per Client Panel



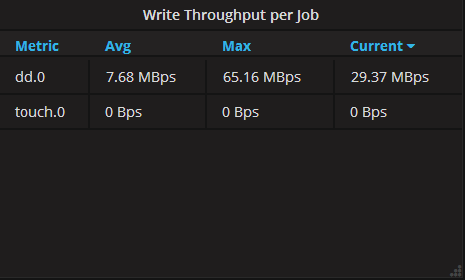
The I/O Throughput per Job panel (Figure 43) shows the metric information of the I/O throughput per jobin the Lustre Filesystem. It includes average/max/current values. As shown in the figure, for the job with JOBID “dd.0”, the average I/O throughput is 7.68MBps, the max value is 65.16MBps, and the current I/O throughput is 29.37MBps.

Figure 43: I/O Throughput per Job Panel



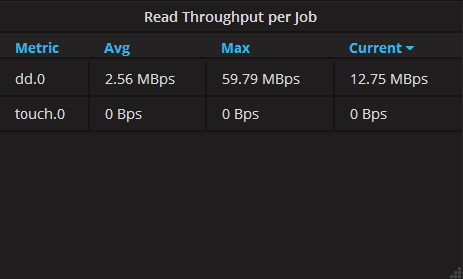
The Write Throughput per Job panel（Figure 44）shows the metric information of the write throughput per job in the Lustre Filesystem. It includes average/max/current values. As shown in the figure, for the job with JOBID “dd.0”, the average I/O throughput is 7.68MBps, the max value is 64.16MBps, and the current I/O throughput is 29.37MBps.

Figure 44: Write Throughput per Job Panel



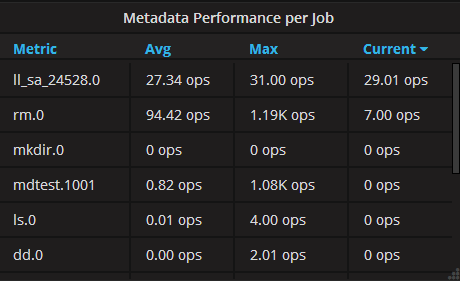
The Read Throughput per Job panel (Figure 45) shows the metric information of the read throughput per job in the Lustre Filesystem. It includes average/max/current values. As shown in the figure, for the job with JOBID “dd.0”, the average I/O throughput is 2.56MBps, the max value is 59.79MBps, and the current I/O throughput is 12.75MBps.

Figure 45: Read Throughput per Job Panel



The Metadata Performance per Job panel (Figure 46) shows the metric information of the metadata performance per job in the Lustre Filesystem. It includes average/max/current values, and the unit is Ops (Operations per Second). As shown in the figure, for the job with JOBID “rm.0”, the average metadata performance is 94.42 ops, max value is 1.19K ops, and the current performance is 7.00 Ops.

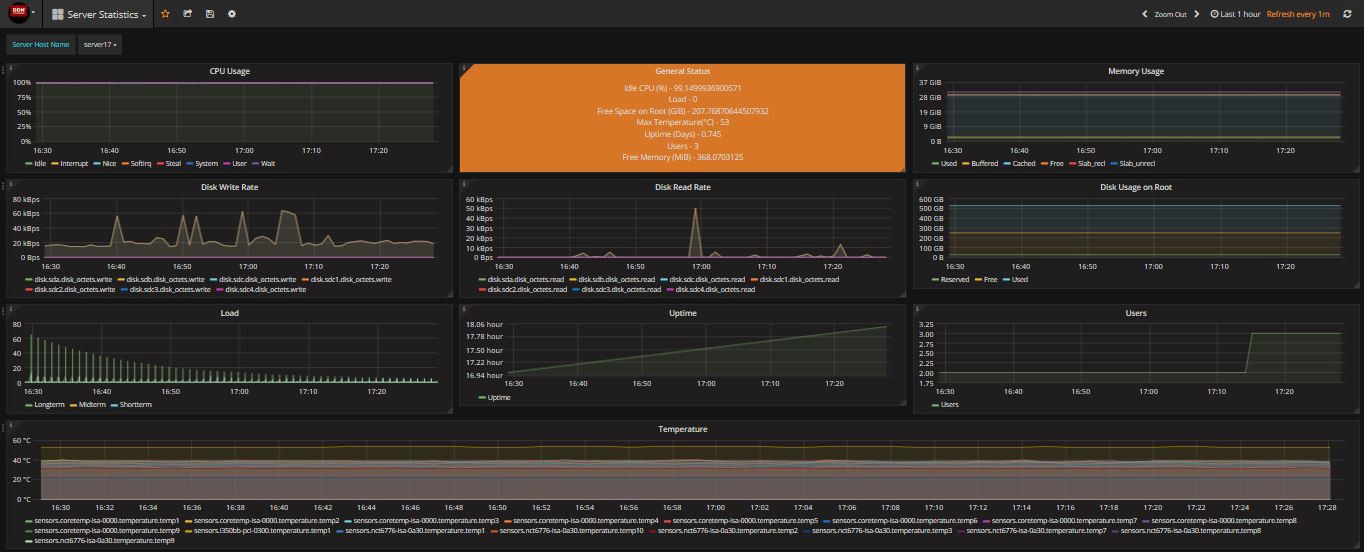
Figure 46: Metadata Performance per Job Panel



## Server Statistics

The Server Statisticsdashboard (Figure 47) shows detailed information about a server.

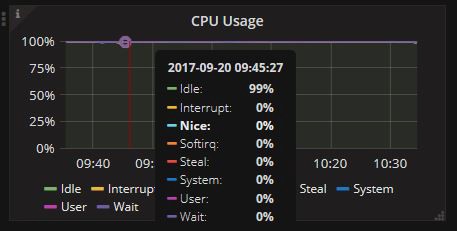
Figure 47: Server Statistics Dashboard



Below you will find description of some of the panels in the **Server Statistics** dashboard:

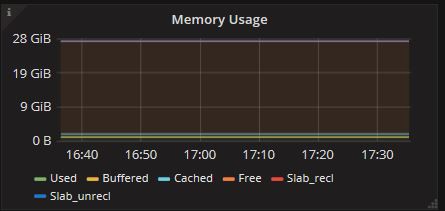
* The CPU Usage panel (Figure 48) shows amount of time spent by the CPU in various states, most notably executing user code, executing system code, waiting for IO-operations and being idle.

Figure 48: CPU Usage Panel



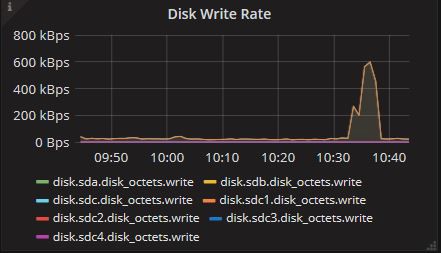
* The Memory Usage panel (Figure 49) shows how much memory has been used. The values are reported by the operating system. The categories are: **Used**, **Buffered**, **Cached**, **Free**, **Slab\_recl**, **Slab\_unrecl**.

Figure 49: Memory Usage Panel



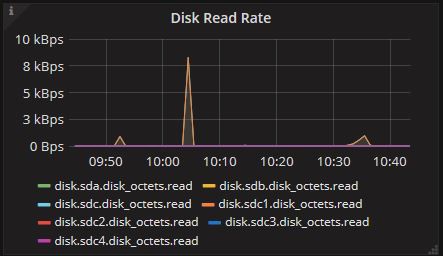
* The Disk Write panel(Figure 50) shows the disk write rate of theserver.

Figure 50: Disk Write Rate Panel



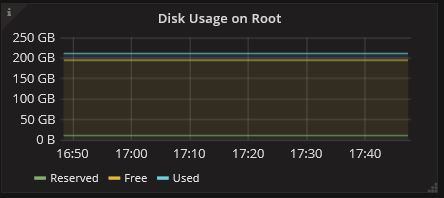
* The Disk Read panel (Figure 51) shows the disk read rate of the server.

Figure 51: Disk Read Rate Panel



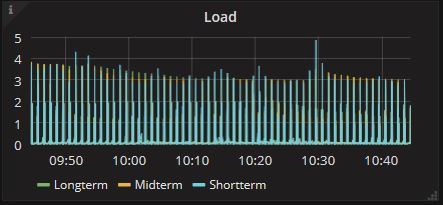
* The Disk Usage on Root panel(Figure 52) showsfreespace,usedspaceandreservedspaceonthe disk that is mounted as Root.A warning message will be generated when there’s little free space left.

Figure 52: Disk Usage on Root Panel



* The Load panel (Figure 53) shows the load on the server. The system load is defined as the number of runnable tasks in the run-queue and is provided by many operatingsystemsas follows:
  + **Shortterm**— one minute average
  + **Midterm**— five minutes average
  + **Longterm**— fifteen minutes average

Figure 53: Load Panel



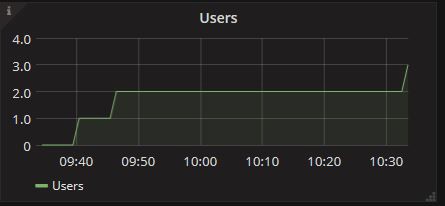
* The Uptime panel (Figure 54) shows how longtheserverisworking.Itkeepstrackofthesystemuptime.

Figure 54: Uptime Panel



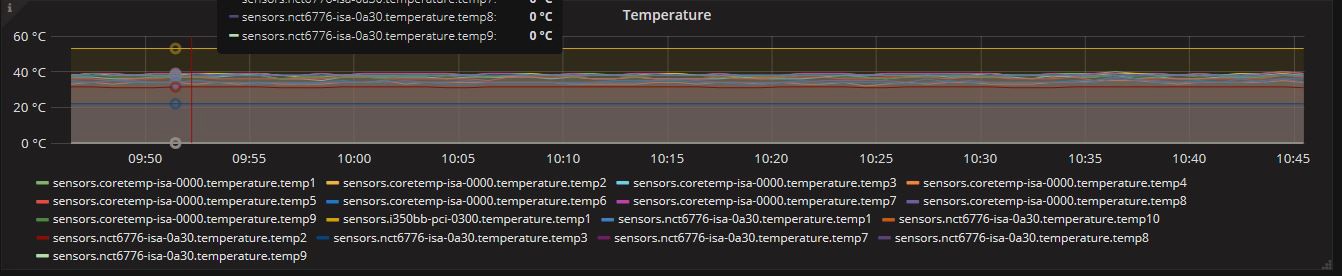
* The User panel (Figure 55) shows the number of users currently logged into the system.

Figure 55: User Panel



* The Temperature panel (Figure 56) shows the temperature collected from sensors.

Figure 56: Temperature Panel



## SFA Physical Disk Dashboard

The **SFA Physical Disk** dashboard shown in Figure 57 displays information about DDN SFAphysical disks.

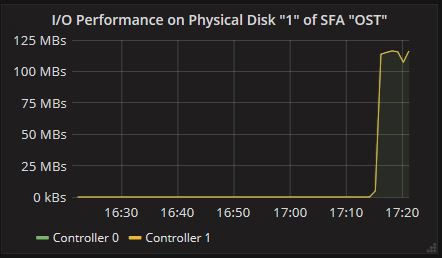
Figure 57: SFA Physical Disk Dashboard



Below you will find description of some of the panels in the **SFA Physical Disk** dashboard:

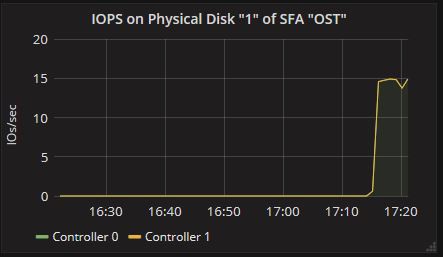
* The I/O Performance panel (Figure 58) shows I/O speed at a specific time.

Figure 58: I/O Performance on Physical DiskPanel



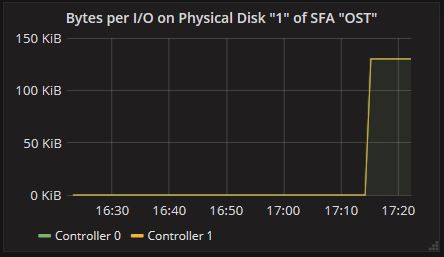
* The IOPS panel (Figure 59) shows I/O operations per second on Physical Disk.

Figure 59: IOPS on Physical Disk Panel



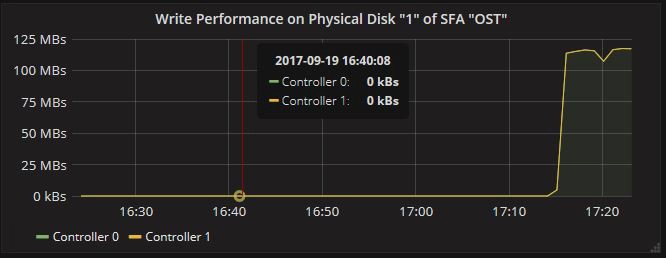
* The Bytes per I/O panel (Figure 60) shows the I/O bytes per second on each controller.

Figure 60: Bytes per I/O on Physical Disk Panel



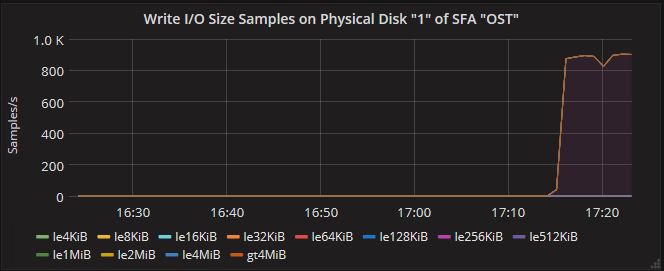
* The Write Performance panel (Figure 61) shows the write performance on each controller.

Figure 61: Write Performance on Physical Disk Panel



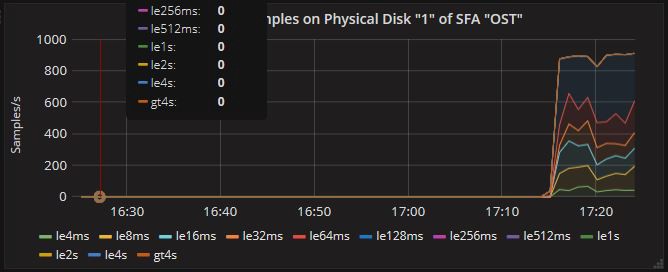
* The Write I/O Size Samples panel (Figure 62) shows the account of writing operation on each size.

Figure 62: Write I/O Size Samples on Physical Disk Panel



* The Write Latency Samples panel (Figure 63) showstheaccountofwritingoperationon each latency.

Figure 63: Write Latency Samples on Physical Disk Panel



## SFA Virtual Disk Dashboard

The **SFA Virtual Disk** dashboard (Figure 64 ) shows information aboutDDN SFAvirtual disks:

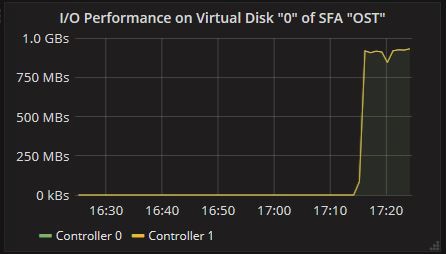
Figure 64: SFA Virtual Disk Dashboard



Below you will find description of some of the panels in the **SFA Virtual Disk** dashboard:

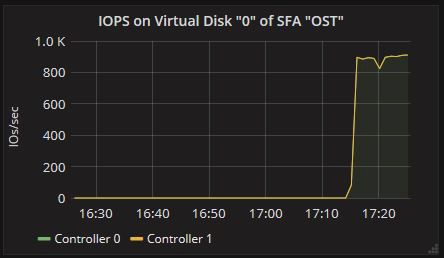
* The I/O Performance panel (Figure 65) in shows the I/O speed at a specific time.

Figure 65: I/O Performanceon Virtual DiskPanel



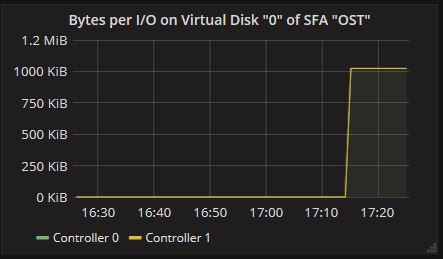
* The IOPS panel (Figure 66) shows I/O operations per second on Virtual Disk.

Figure 66: I/O Operations per Secondon Virtual DiskPanel



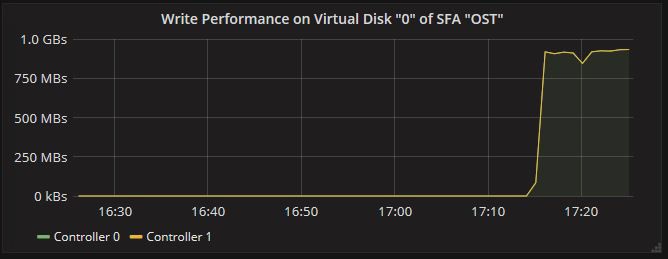
* The Bytes per I/O panel (Figure 67) shows I/O bytes per second on each controller.

Figure 67: Bytes per I/O on Virtual DiskPanel



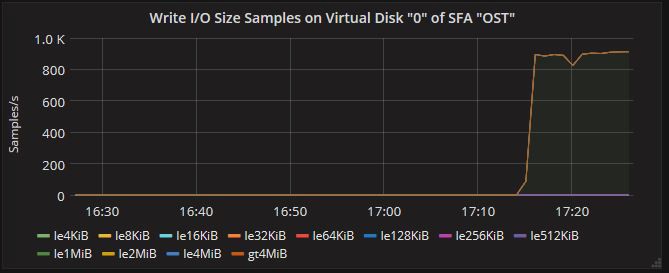
* The Write Performance panel (Figure 68) shows write performance on each controller.

Figure 68: Write Performance on Virtual DiskPanel



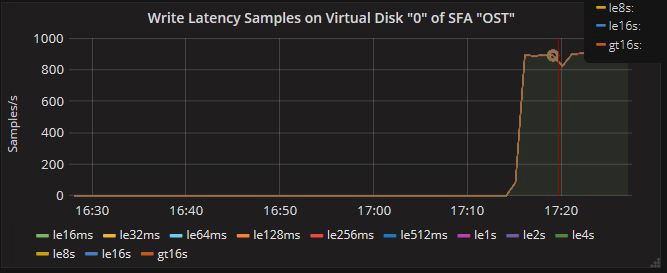
* The Write I/O Size Samples panel (Figure 69)shows the size distributions of write I/Os.

Figure 69: Write I/O Size Samples on Virtual Disk Panel



* The Write Latency Samples panel (Figure 70) shows the latency distributions of write I/Os.

Figure 70: Write Latency Samples on Virtual Disk Panel



# Stress Testing

In order to check whether the monitoring system works well under high pressure, we designed the collectd-stress2 plug-in for stress testing. It is an upgraded version of the stress plugin which can use a couple of Collectd clients to simulate tens of thousands of metrics collected from hundreds of servers.

## Installing stress2 RPM on Collectd Client

The stress2 plug-in will generate a large amount of simulated monitoring data and contaminate the database, so the plug-in shall not be installed on all clients by default. After deploying the monitoring system using esmon\_install, a couple of Collectdclient can be used as the testing clients. You can find collectd-stress2 \* .rpm under the ISO directory and install it with the following command:

rpm --ivh collectd-stress2\*.rpm

## Updating ConfigurationFile of Collectd Client

After installing stress2 RPM, update the configuration file /etc/collectd.conf and add the following configuration:

* **Thread**—Defines the number of test threads.
* **Metric**—Defines all the attributes of a monitoring target. It can be specified multiple times to simulate different monitoring targets at the same time. It contains the following attributes:
  + **Variable**—Define the scope of the monitoring target changes and the speed of change, it can be specified multiple times.
    - **Name**—Definethe variable name.
    - **Number**—Defines the maximum range of variable changes.
    - **UpdateIterval**—Defines the time interval between variable changes.
  + **Host**—Define the host name of the client, usually defined as "$ {key: hostname}", the program automatically sets the current host name. It describe the discriminator of the collection data object together with the following **Plugin**, **PluginInstance**, **Type**, **TypeInstance**, See [Naming Schema](https://collectd.org/wiki/index.php/Naming_schema) for detailed.
  + **Plugin**—Defines the plugin member in the collectd identifier.
  + **PluginInstance**—Defines the plugininstance member in the collectd identifier.
  + **Type**—The type member of the collectd identifier is defined. For details, see [Data Types](https://collectd.org/wiki/index.php/Derive).
  + **TypeInstance**—Defines the type instance member in the collectd identifier.
  + **TsdbName**—Defines the name submitted to the database format.
  + **TsdbTags**—Defined the tags submitted to the database format to facilitate the late classification search.

Below is an example of /etc/collectd.conf.

**Example:**

LoadPlugin stress2

<Plugin "stress2">

  Thread 32

  <Metric>

<Variable>

    Name "ost\_index"

    Number 10

    UpdateIterval 0

</Variable>

<Variable>

    Name "job\_id"

    Number 7000

    UpdateIterval 10

</Variable>

  Host "${key:hostname}"

  Plugin "stress-${variable:ost\_index:OST%04x}"

  PluginInstance "jobstat\_${variable:job\_id:job%d}"

  Type "derive"

  TypeInstance "sum\_read\_bytes"

  TsdbName "ost\_jobstats\_samples"

  TsdbTags "optype=sum\_read\_bytes fs\_name=stress ost\_index=${variable:ost\_index:OST%04x} job\_id=${variable:job\_id:job%d}"

   </Metric>

  <Metric>

<Variable>

    Name "mdt\_index"

    Number 10

    UpdateIterval 0

</Variable>

<Variable>

    Name "md\_stats"

    Number 10

    UpdateIterval 10

</Variable>

  Host "${key:hostname}"

  Plugin "stress-${variable:mdt\_index:MDT%04x}"

  PluginInstance "md\_stats"

  Type "derive"

  TypeInstance "open"

  TsdbName "md\_stats"

  TsdbTags "optype=open fs\_name=stress mdt\_index=${variable:mdt\_index:MDT%04x} mdt\_stats\_open=${variable:mdt\_stats\_open:%d}"

   </Metric>

</Plugin>

## Start Testing

After modifyingthe configuration file, restart Collectd:

service collectd restart

By viewing  */var/log/messages l*og, you can see the following message:

server11 collectd[20830]: stress2: time: 1.79244 for 70100 commits with 32 threads, 39108.70099 commits/second

The above information shows that stress2 plug-in has been successfully loaded , and generated a lot of monitoring data. With the above configuration fileand following specified hardware environment, the corresponding monitoring bottlenecks were checked.

* **OS：**CentOS7.
* **Memory：**128GB.
* **CPU: I**ntel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz.
* **Disk：**Samsung SSD 850 2B6Q.

The monitoring client and database server are running on the same host, Influxdb data is stored on SSD with Ext4 file system.

**Preconditions:**

* **CollectdInterval:** 60 seconds.
* **GrafanaHistory:** 1 hour.
* **Grafana Refresh Interval:** 60 seconds.
* **Collectd Running Time:** more than 1 hour.

**Conclusion:**

* **Grafanakeeps on refreshing:** monitor overload.
* **Grafanahas idle time:** monitor running well.

In theory, Grafana's refresh time equals the database query time plus the web page load time.

We can query the database to measure the performance of the database query. For example the following is the default query command for ESMON Grafana "Read Throughput per Job":

influx -database esmon\_database –execute \

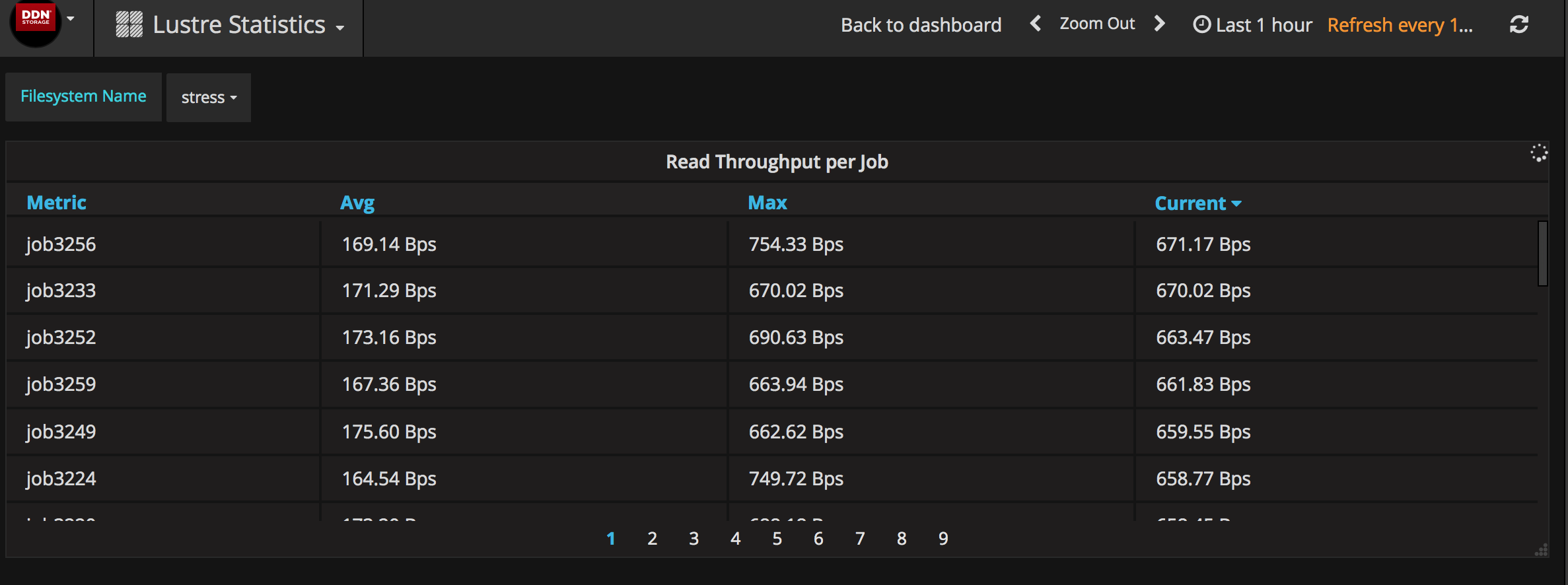
"SELECT "value" FROM "ost\_jobstats\_samples" WHERE ("optype" = 'sum\_read\_bytes' AND "fs\_name" = 'stress') AND time >= now() - 1h GROUP BY "job\_id""

With the monitoring software running, the above command on the database host can be executed to verify the query time.As shown in Figure 71, the query time of the Influxdb grew linearly during the first one hour, because the data points kept on accumulating . But after an hour the query time became steady, which is also expected behavior.

Figure 71：Influxdb Query Time

After verifying the load on the database side, we also need to verify the loading status of the Grafana. As shown in Figure 72, Login Grafana to see "Read Throughput per Job".

Figure 72：Read throughput per Job stress testing



If the page is always refreshing and the page can be loaded within 60 seconds, that means, under the current configuration, the monitoring system can handle the current pressure. Otherwise, the monitoring system can be considered overloaded. In that case, either hardware need to be upgraded to higher specification or the data collecting/refreshing intervals need to be increased. By continuously adjusting the number of **job\_id** in */etc/collectd.conf* and check the page refreshing latency, the maximum supported metrics can be known under the current hardware configuration. Tests show that if Lustre has 10 OSTs, with above hardware, the monitoring system can support up to 7000 running jobs at the same time without any problem.

# Troubleshooting

The directory*/var/log/esmon\_install/[installing\_date]* on the Installation Servergathersallthelogsthatisusefulfordebugging.Ifafailurehappens,someerror messageswillbeprintedtofile*/var/log/esmon\_install/[installing\_date]/error.log*. The first error message usually contains the informationabout the cause of failure.