Important Information

**ESMON**

用户手册

**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 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

目录

[目录 4](#_Toc503451148)

[图片列表 6](#_Toc503451149)

[DDN EXAScaler监控系统简介 9](#_Toc503451150)

[名词解释 9](#_Toc503451151)

[DDN的Collectd插件 9](#_Toc503451152)

[安装要求 11](#_Toc503451153)

[部署服务器 11](#_Toc503451154)

[监控服务器 11](#_Toc503451155)

[监控客户端 11](#_Toc503451156)

[安装过程 12](#_Toc503451157)

[在部署服务器上安装ESMON RPM 12](#_Toc503451158)

[在部署服务器上更新配置文件 12](#_Toc503451159)

[在集群上运行安装程序 13](#_Toc503451160)

[访问监控网络页面 14](#_Toc503451161)

[仪表盘 15](#_Toc503451162)

[集群状态仪表盘 15](#_Toc503451163)

[Lustre仪表盘 16](#_Toc503451164)

[服务器仪表盘 37](#_Toc503451165)

[SFA 物理磁盘仪表盘 42](#_Toc503451166)

[SFA虚拟磁盘表盘 45](#_Toc503451167)

[压力测试 48](#_Toc503451168)

[在部署客户端安装插件 48](#_Toc503451169)

[在部署客户端更新配置文件 48](#_Toc503451170)

[进行压力测试 49](#_Toc503451171)

[故障排查 52](#_Toc503451172)

图片列表

[图1: Grafana登陆界面 14](#_Toc503795796)

[图2：主仪表盘 15](#_Toc503795797)

[图3：集群状态仪表盘 16](#_Toc503795798)

[图4：Lustre仪表盘 16](#_Toc503795799)

[图5: Lustre系统总剩余容量面板 17](#_Toc503795800)

[图6: Lustre文件系统已使用总容量面板 17](#_Toc503795801)

[图7: Lustre文件系统每个OST剩余容量面板 18](#_Toc503795802)

[图8: 每个OST已使用容量面板 18](#_Toc503795803)

[图9: 用户已使用容量面板 19](#_Toc503795804)

[图10: 用户组已使用容量面板 19](#_Toc503795805)

[图11: 总的剩余索引节点数目面板 20](#_Toc503795806)

[图12: 总的已使用索引节点数目面板 20](#_Toc503795807)

[图13: 每个MDT剩余索引节点数目面板 21](#_Toc503795808)

[图14: 每个用户已使用索引节点数目面板 21](#_Toc503795809)

[图15: 每个用户组已使用索引节点数目面板 22](#_Toc503795810)

[图16: 每个MDT已使用索引节点数目面板 22](#_Toc503795811)

[图17: Lustre文件系统总的I/O吞吐量面板 23](#_Toc503795812)

[图18: 每个OST的I/O吞吐量面板 23](#_Toc503795813)

[图19: 每个OST写吞吐量面板 24](#_Toc503795814)

[图20: 每个OST读吞吐量面板 24](#_Toc503795815)

[图21: 总的元数据操作速率面板 25](#_Toc503795816)

[图22: 每个MDT的元数据操作速率面板 25](#_Toc503795817)

[图23: 每个客户端的元数据操作速率面板 26](#_Toc503795818)

[图24: 每种类型元数据操作速率面板 26](#_Toc503795819)

[图25: 不同块大小的写RPC速率面板 27](#_Toc503795820)

[图26: 写bulk RPC大小分布面板 27](#_Toc503795821)

[图27: 不同大小的读bulk RPC速率面板 28](#_Toc503795822)

[图28: 读bulk RPC大小分布面板 28](#_Toc503795823)

[图29: 写I/O不连续页面分布面板 29](#_Toc503795824)

[图30: 读I/O不连续页面分布面板 29](#_Toc503795825)

[图31: 写I/O不连续块分布面板 30](#_Toc503795826)

[图32: 读I/O不连续块分布面板 30](#_Toc503795827)

[图33: 每个写I/O碎片分布面板 31](#_Toc503795828)

[图34: 读I/O碎片分布面板 31](#_Toc503795829)

[图35: 已提交等待结束的磁盘写I/O分布面板 32](#_Toc503795830)

[图36: 已提交等待结束的磁盘读I/O分布面板 32](#_Toc503795831)

[图37: 写I/O时间分布面板 32](#_Toc503795832)

[图38: 读I/O时间分布面板 33](#_Toc503795833)

[图39: 磁盘写I/O大小分布面板 33](#_Toc503795834)

[图40: 磁盘读I/O大小分布面板 34](#_Toc503795835)

[图41: 每个客户端写吞吐量面板 34](#_Toc503795836)

[图42: 每个客户端读吞吐量面板 35](#_Toc503795837)

[图43: 每个作业I/O吞吐量面板 35](#_Toc503795838)

[图44: 每个作业写吞吐量面板 36](#_Toc503795839)

[图45: 每个作业读吞吐量面板 36](#_Toc503795840)

[图46: 每个作业元数据性能面板 37](#_Toc503795841)

[图47：服务器仪表盘 37](#_Toc503795842)

[图48：CPU 使用率面板 38](#_Toc503795843)

[图49：内存使用率面板 38](#_Toc503795844)

[图50：磁盘写速率面板 39](#_Toc503795845)

[图51：磁盘读速率面板 39](#_Toc503795846)

[图52：根目录磁盘使用率 40](#_Toc503795847)

[图53：负载面板 40](#_Toc503795848)

[图54：启动时间面板 41](#_Toc503795849)

[图55：用户数面板 41](#_Toc503795850)

[图56：温度面板 41](#_Toc503795851)

[图57: SFA物理磁盘仪表盘 42](#_Toc503795852)

[图58：I/O 吞吐率面板 42](#_Toc503795853)

[图59：IOPS面板 43](#_Toc503795854)

[图60：I/O 大小面板 43](#_Toc503795855)

[图61：写数据吞吐率面板 44](#_Toc503795856)

[图62：写数据的I/O大小样品采样数目面板 44](#_Toc503795857)

[图63：数据写入延迟样品采样数目面板 44](#_Toc503795858)

[图64：SFA虚拟磁盘表盘 45](#_Toc503795859)

[图65：I/O吞吐率面板 45](#_Toc503795860)

[图66：IOPS面板 46](#_Toc503795861)

[图67：Bytes per I/O on Virtual DiskPanel 46](#_Toc503795862)

[图68：Write Performance on Virtual DiskPanel 46](#_Toc503795863)

[图69：数据写入的I/O大小样品采集数目面板 47](#_Toc503795864)

[图70：数据写入的延迟样品采集数目面板 47](#_Toc503795865)

[图71：Influxdb Query Time 变化曲线图 50](#_Toc503795866)

[图72：每个任务读吞吐压测面板 51](#_Toc503795867)

# DDN EXAScaler监控系统简介

ESMON 是一款监控系统，通过采集DDN Exascaler的系统状态信息以达到对其进行性能监控及分析的目的。ESMON基于多种开源软件的监控系统，DDN同时还开发了一些外部插件以作功能扩展。

ESMON 的主要组件之一是 Collectd。Collectd是一个运行在监控对象上的守护进程，完成了系统性能的相关统计信息定期收集，并将这些数据以多种不同的机制进行存储。ESMON 是基于开源的Collectd而研发，同时也包含了许多其他插件，如Lustre、GPFS、Ganglia、Nagios、Stress、Zabbix等等。

## 名词解释

|  |  |
| --- | --- |
| **ESMON** | DDN Exascaler监控系统的缩写。 |
| **DDN EXAScaler** | DDN Exascaler是由DDN研发的基于Lustre的存储解决方案，它旨在解决极端数据密集型环境中最严苛的存储及数据管理问题。 |
| **部署服务器** | ESMON监控系统的安装进程将在此服务器上被触发。 |
| **监控服务器** | ESMON监控系统的数据库(Influxdb) 及网络服务 (Grafana) 将在此服务器上运行。 |
| **被监控客户端** | ESMON监控系统将从被监控客户端上收集各类指标数据，例如CPU, 内存、Lustre、SFA 存储等相关信息。守护进程Collectd 将在每个客户端后台运行。 |
| **DDN IME** | DDN 的无限内存引擎(Infinite Memory Engine) 是一款基于闪存的高速缓存系统，它简化了应用程序I/O路径，消除了系统瓶颈。 |
| **Lustre** | Lustre文件系统是一种开源的并行文件系统，它满足了许多高性能计算环境的海量数据存储需求。 |

## DDN的Collectd插件

为支持更多不同功能，DDN添加了一些附加的Collectd插件。

* **Filedata插件**：Filedata插件可通过读取及解析一组文件进行数据收集。用户需要在一个xml格式文件中对读取哪些文件、如何解析这些文件等进行定义。Filedata插件最常见的用途是通过正在运行的Lustre系统的/proc接口收集指标。
* **Ganglia插件**：Ganglia插件将collectd进程收集的指标信息发送给Ganglia服务器。
* **GPFS插件**：GPFS插件通过GPFS提供的*mmpmon*命令来收集性能信息。GPFS插件的定义文件格式和Filedata插件相同。同时，其*collectd.conf*的配置格式也与Filedata插件相似。
* **IME插件**：IME插件通过DDN IME收集性能信息。像GPFS插件一样，IME插件和Filedata插件共享相似的定义文件格式和配置格式。
* **SSH插件**：SSH插件能够通过使用SSH连接在远程主机上运行命令来收集来自DDN SFA存储的各项指标。像GPFS插件和IME插件一样，它的定义文件格式和配置格式和Filedata插件类似。
* **Stress插件：**Stress插件可以从Collectd客户端向服务器推送大量数据，来对监控系统性能进行高强度基准测试。
* **Stress2插件：**类似Stress 插件，但是能提供更灵活的配置, 功能更强大。
* **Zabbix插件**：Zabbix插件将指标数据从Collectd发送至Zabbix系统。

# 安装要求

## 部署服务器

* **操作系统版本：**CentOS7/RHEL7。
* **硬盘空闲空间：**>500MB。所有安装日志将被保存于部署服务器的*/var/log/esmon\_install*目录下，这需要占用一定空间。
* **网络：**部署服务器须能对监控服务器和被监控客户端发起无密码提示的SSH连接。
* **ESMON ISO镜像：**部署服务器上需有可用的ESMON ISO 镜像。

## 监控服务器

* **操作系统版本：** CentOS7/RHEL7。
* **硬盘空闲空间：**> 5GB。监控服务器上会运行Influxdb。如果Influxdb需容纳更多数据，则须预留更大空间。
* **网络：**监控服务器须运行SSHD。部署服务器必须可以通过无密码提示的SSH与监控服务器连接。

## 监控客户端

* **操作系统版本：**CentOS7/RHEL7或CentOS6/RHEL6。
* **硬盘空闲空间：**> 200MB。必要的RPMs将被保存于 */var/log/esmon\_install*目录下，这将占用一定空间。
* **网络：**监控可虚幻须运行SSHD。部署服务器必须可以通过无密码提示的SSH与监控客户端连接。

# 安装过程

## 在部署服务器上安装ESMON RPM

1. 将 ESMON ISO 镜像文件拷贝至部署服务器上，如*/ISOs/esmon.iso*。
2. 挂载 ESMON ISO 镜像：

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

1. 安装 ESMON RPM：

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

## 在部署服务器上更新配置文件

在安装ESMON RPM之后，更新配置文件/etc/esmon\_install.conf。该文件包含了安装所需的所有必要信息。

* **iso\_path** —ESMON ISO镜像的保存路径。
* 在**ssh\_hosts**部分，详细定义了用如何从部署服务器上用SSH连接登录监控服务器、监控客户端的相关信息：
  + **host\_id**—该主机的唯一标识。两个不同的主机不能拥有相同的**host\_id**。
  + **hostname**—进行SSH连接时的主机名。
  + **ssh\_identity\_file**—进行SSH连接时的使用的SSH密钥文件。如果默认的SSH密钥文件可用，该参数可省略。

注意: 对同一个主机，**host\_id**和**hostname**可能不同，因为可以用多种方式连接至同一个主机。

* 在**client\_hosts**部分，定义了需要安装和配置ESMON客户端的所有主机的信息：
  + **host\_id**—该主机的唯一标识。两个不同的主机不能拥有相同的**host\_id**。
  + **lustre\_oss**—定义了是否启用 **Lustre OSS**上的数据收集功能。
  + **lustre\_mds**—定义了是否启用 **Lustre MDS**上的数据收集功能。
  + **ime**—定义了是否启用 **DDN IME**中指标收集功能。
* 在**server\_host**部分，定义了监控服务器的信息：
  + **host\_id**—该主机的唯一标识。
  + **drop\_database** —当此参数为真时(设置为**true**)，Influxdb中的ESMON数据库将被丢弃。当此参数为假时（设置为**false**）, Influxdb中的ESMON数据库将被保留。
  + **erase\_influxdb** —当此参数为真时(设置为**true**)，所有Influxdb中的数据和元数据都将被完全清除。通过启用**erase\_influxdb**可解决Influxdb的一些数据损坏问题。如果该参数不启用（设置为**false**）,则Influxdb中的数据和元数据将不会被完全清除。

**重要: 只有当Influxdb中数据和元数据无用时才可以启用erase\_influxdb**和**drop\_database**

下面是/etc/esmon\_install.conf的一个例子

**例子:**

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

client\_hosts:

- host\_id: Monitoring-Client1

lustre\_oss: true

lustre\_mds: true

ime: false

- host\_id: Monitoring-Client2

lustre\_oss: false

lustre\_mds: true

ime: false

server\_host:

- host\_id: Monitoring-Server

drop\_database: true

erase\_influxdb: true

## 在集群上运行安装程序

在部署服务器上的*/etc/esmon\_install.conf*更新完成后，运行以下命令在集群上启动安装程序：

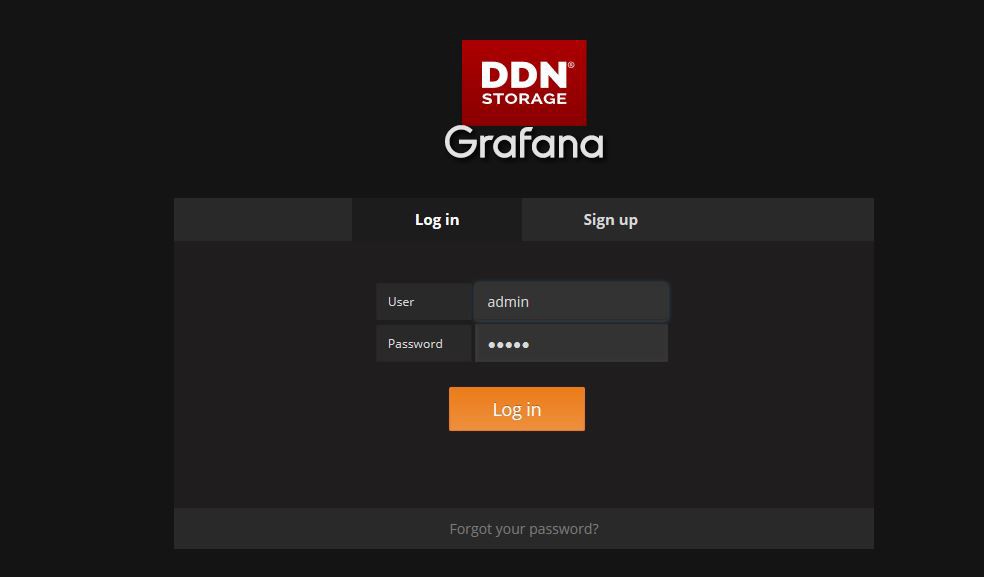
esmon\_install

所有可用于调试的相关日志将被保存在*/var/log/esmon\_install*目录下。

## 访问监控网络页面

Grafana服务将自动在监控服务器启动。默认HTTP 端口为3000。通过访问该端口可跳转至登录页面 ()，默认用户名密码皆为*admin*。

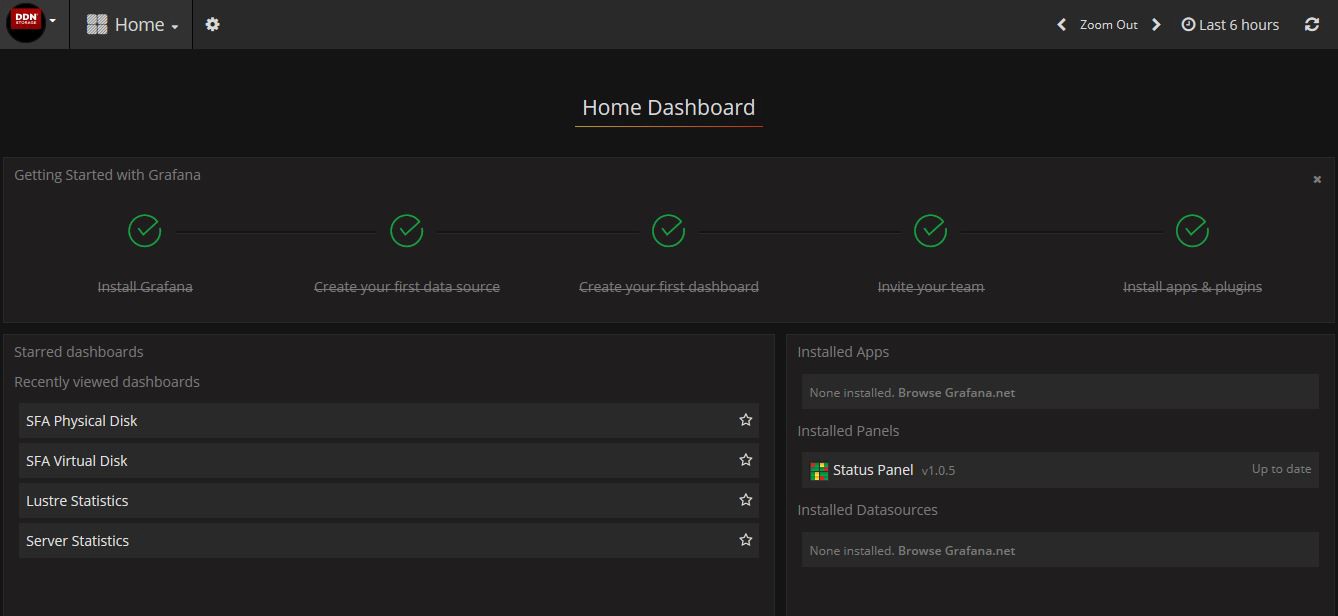
图1: Grafana登陆界面



# 仪表盘

在主仪表盘上（），可通过选择不同的仪表盘来浏览由ESMON收集的各种数据。

图2：主仪表盘



## 集群状态仪表盘

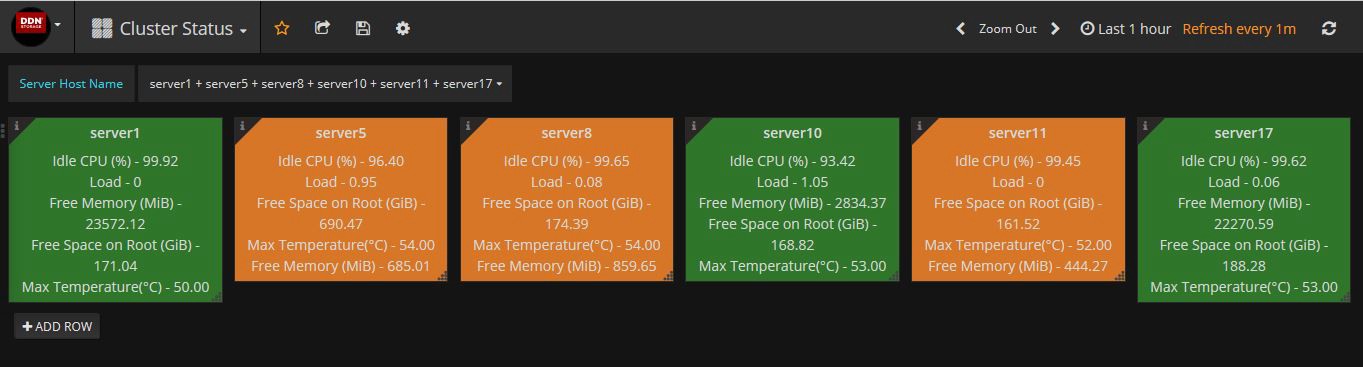
集群状态仪表盘（空闲 CPU 不足5%；

* + 负载数量高于10；
  + 空闲内存不足 100 MiB；
  + 根目录空闲空间不足1GiB。

图3）显示了集群中服务器的状态信息概要。其中，面板的背景颜色与服务器的运行状态相关

* 绿色：服务器正常运行。
* 黄色：警告信息，说明发生一个或多个下列情况：
  + 空闲 CPU 不足20%；
  + 负载数量高于5；
  + 空闲内存不足 1000 MiB；
  + 根目录空闲空间不足10 GiB。
* 红色：严重警告信息，说明发生一个或多个下列情况：
  + 空闲 CPU 不足5%；
  + 负载数量高于10；
  + 空闲内存不足 100 MiB；
  + 根目录空闲空间不足1 GiB。

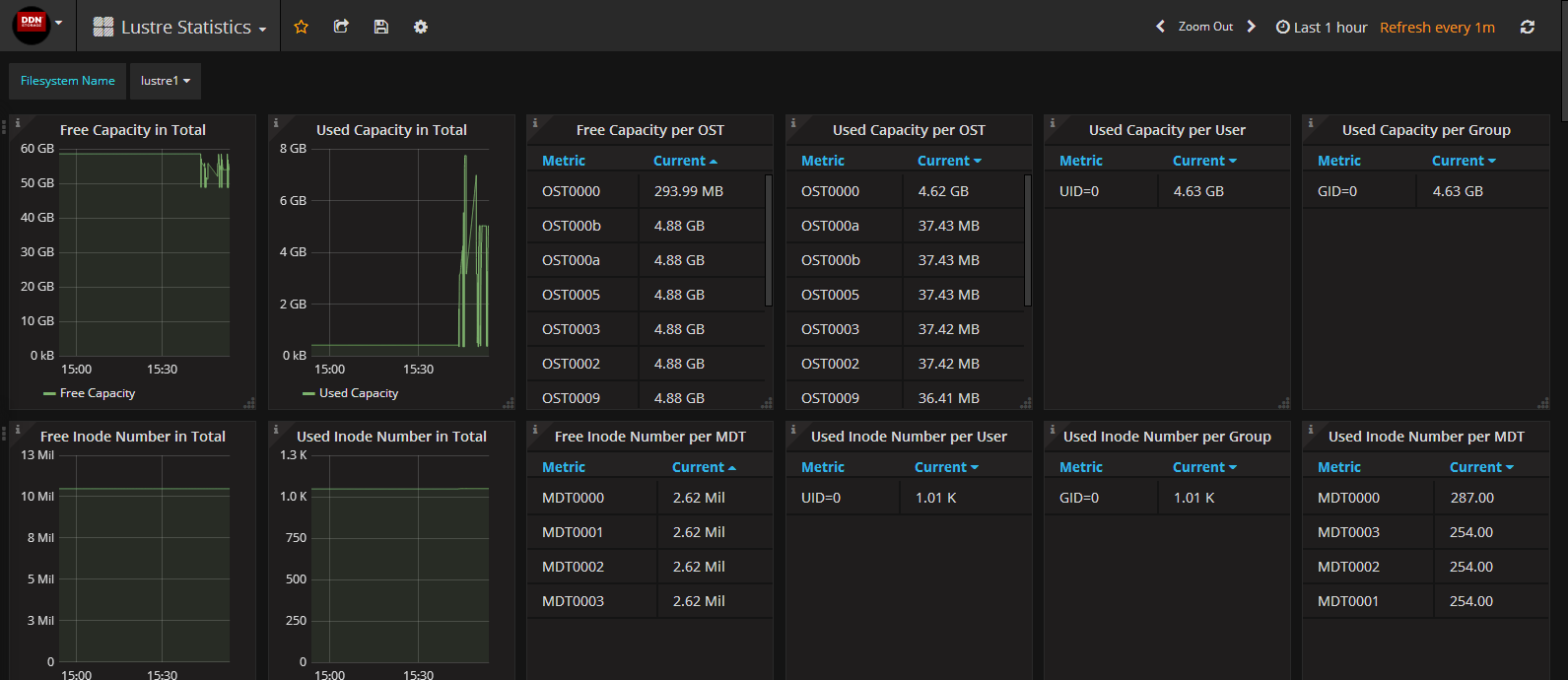
图3：集群状态仪表盘



## Lustre仪表盘

Lustre仪表盘（）显示了 Lustre 文件系统统计数据。

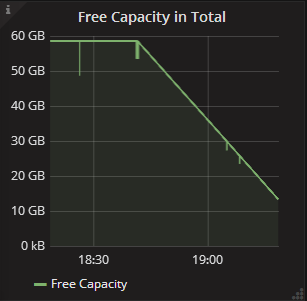
图4：Lustre仪表盘



以下是一些主要指标的面板视图：

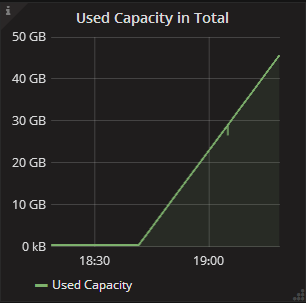
* 系统总剩余容量面板() 显示了Lustre文件系统剩余容量随时间的变化。从大约18:40开始运行测试用例”dd if=/dev/zero of=/mnt/lustre/file bs=1M”，图中显示出剩余容量正在以大约20MB/s的速度不断消耗。

图5: Lustre系统总剩余容量面板



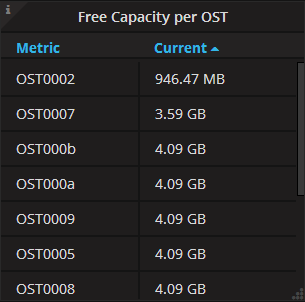
* 系统已使用总容量面板() 显示了Lustre文件系统已使用总容量随时间的变化。该图的测试用例为“dd if=/dev/zero of=/mnt/lustre/file bs=1M”,从图可以看出已使用容量从18:40开始以大约20MB/s的速度递增。

图6: Lustre文件系统已使用总容量面板



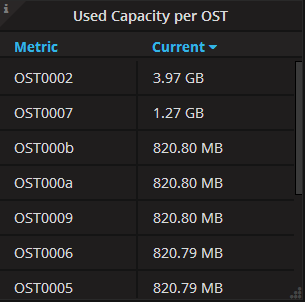
OST剩余容量面板() 显示了Lustre文件系统每个OST剩余容量大小。如图所示，OST0002剩余容量为946.47MB， OST0007剩余容量为3.59GB，其他OST剩余容量都为4.09GB。点击Current，可以对根据当前容量从小到大（或从大到小）进行排序。

图7: Lustre文件系统每个OST剩余容量面板



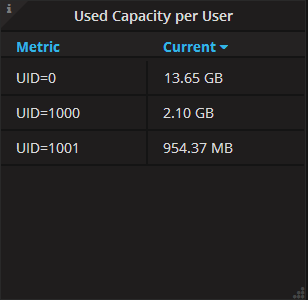
OST已使用容量面板() 显示了Lustre文件系统每个OST已使用的容量大小。如图所示，OST0002已使用容量为3.97GB， OST0007已使用容量为1.27GB，其余OST已使用容量约为820.8MB。点击Current，可以对根据当前已使用容量从小到大（或从大到小）进行排序。

图8: 每个OST已使用容量面板



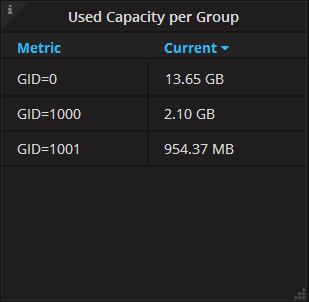
用户已使用容量面板() 显示了Lustre文件系统每个用户已使用的容量大小。如图所示，UID为0的用户当前使用的容量为13.65GB；UID为1000的用户当前使用的容量为2.10GB；UID=1001的用户当前使用的容量为954.37MB。

图9: 用户已使用容量面板



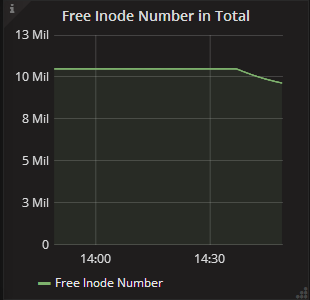
用户组已使用容量面板() 显示了Lustre文件系统每个用户组已使用的容量大小。如图所示，GID为0的用户当前使用的容量为13.65GB；GID为1000的用户当前使用的容量为2.10GB；GID=1001的用户当前使用的容量为954.37MB。

图10: 用户组已使用容量面板



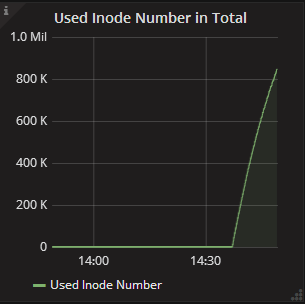
总的剩余索引节点数目面板() 显示了Lustre文件系统总的剩余索引节点数目随时间的变化。该图所使用的测试用例为“mdtest–C –n 900000 –d /mnt/lustre/mdtest/”，从图可以看出从大约14::35开始运行测试用例后，剩余的索引节点数目于大约1100Ops的速度减少。

图11: 总的剩余索引节点数目面板



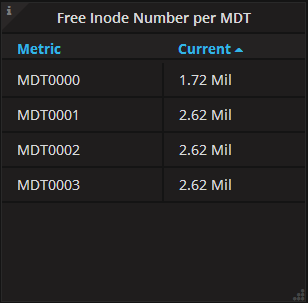
总的已使用索引节点数目面板() 显示了Lustre文件系统总的已使用索引节点数目随时间的变化。该图所使用的测试用例为“mdtest–C –n 900000 –d /mnt/lustre/mdtest/”，从图可以看出从大约14:35开始运行测试用例后，已使用的索引节点数目于大约1100Ops的速度增加。

图12: 总的已使用索引节点数目面板



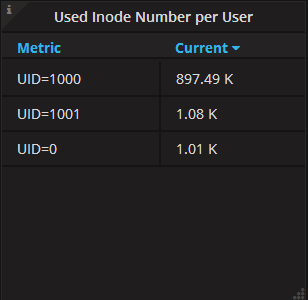
每个MDT剩余索引节点数目面板() 显示了Lustre文件系统每个MDT剩余索引节点数目。如图所示，MDT0000剩余索引节点的数目为1.72Mil；系统其他各个MDT空闲的索引节点数目为2.62 Mil。点击Current，可以对根据当前索引节点数目从小到大（或从大到小）进行排序。

图13: 每个MDT剩余索引节点数目面板



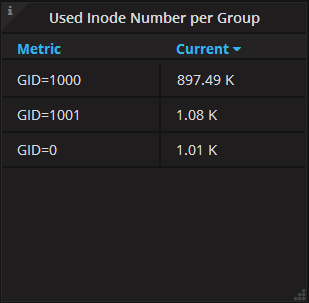
每个用户已使用索引节点数目面板() 显示了Lustre文件系统每个用户已使用索引节点数目。如图所示，UID=1000的用户已使用的索引节点数目为897.49K； UID=1001的用户已使用的索引节点数目为1.08K；UID=0的用户已使用的索引节点数目为1.01K。点击Current，可以对根据当前索引节点数目从小到大（或从大到小）进行排序。

图14: 每个用户已使用索引节点数目面板



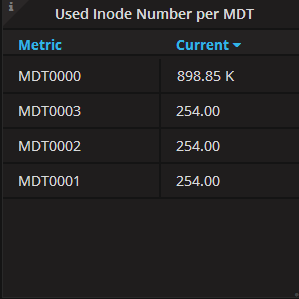
每个用户组已使用的索引节点数目面板() 显示了Lustre文件系统每个用户组已使用的索引节点数目。如图所示，GID=1000的用户组已使用的索引节点数目为897.49K； GID=1001的用户组已使用的索引节点数目为1.08K；GID=0的用户组已使用的索引节点数目为1.01K。点击Current，可以对根据当前索引节点数目从小到大（或从大到小）进行排序。

图15: 每个用户组已使用索引节点数目面板



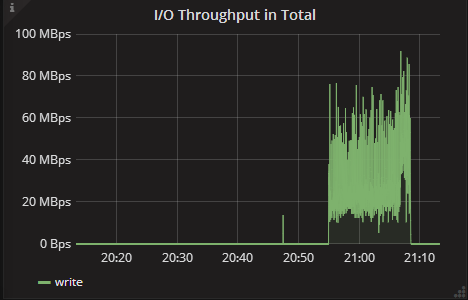
每个MDT已使用的索引节点数目面板() 显示了Lustre文件系统每个MDT已使用的索引节点数目。如图所示，MDT0000使用索引节点数目为898.85K，MDT0001为254。

图16: 每个MDT已使用索引节点数目面板



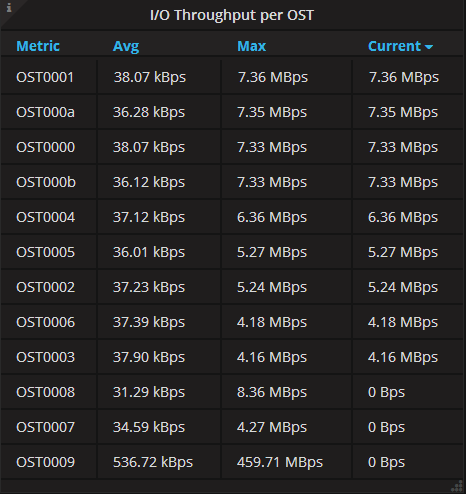
总的I/O吞吐量面板() 显示了Lustre文件系统总的I/O吞吐量随时间的变化。

图17: Lustre文件系统总的I/O吞吐量面板



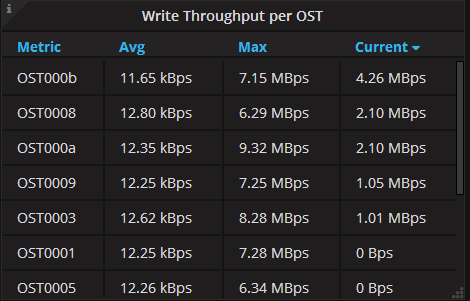
每个OST的I/O吞吐量面板() 显示了Lustre文件系统每个OST的I/O吞吐量信息，包括平均值、最大值和当前值。

图18: 每个OST的I/O吞吐量面板



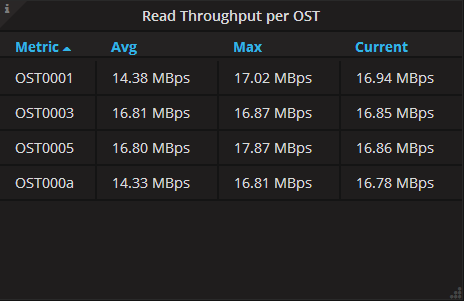
每个OST的写吞吐量面板() 显示了Lustre文件系统每个OST的写吞吐量信息，包括平均值、最大值和当前值。

图19: 每个OST写吞吐量面板



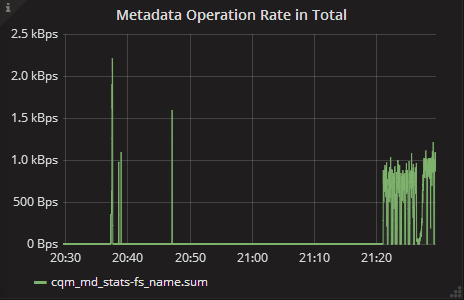
每个OST的读吞吐量面板() 显示了Lustre文件系统每个OST的读吞吐量信息，包括平均值、最大值和当前值。

图20: 每个OST读吞吐量面板



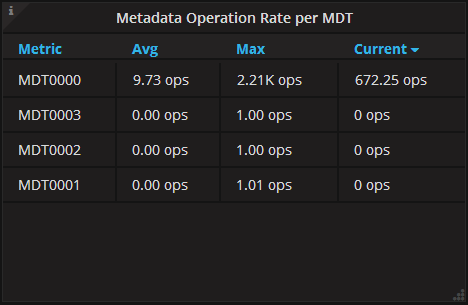
总的元数据操作速率面板() 显示了Lustre文件系统总的元数据操作速率随时间的变化，其单位为Ops， Operation Per Second。

图21: 总的元数据操作速率面板



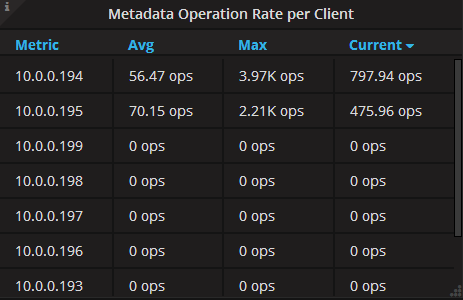
每个MDT的元数据操作速率面板() 显示了Lustre文件系统每个MDT的元数据操作速率信息，其单位为Ops， Operation Per Second。其包括平均值、最大值和当前值。

图22: 每个MDT的元数据操作速率面板



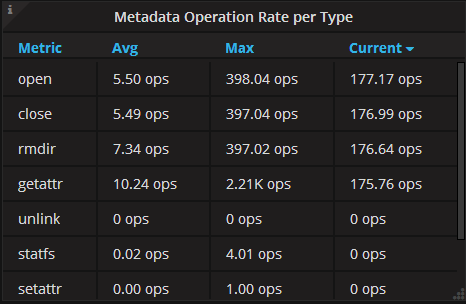
每个客户端的元数据操作速率面板() 显示了Lustre文件系统每个客户端（client）的元数据操作速率信息，其单位为Ops， Operation Per Second。其包括平均值、最大值和当前值。

图23: 每个客户端的元数据操作速率面板



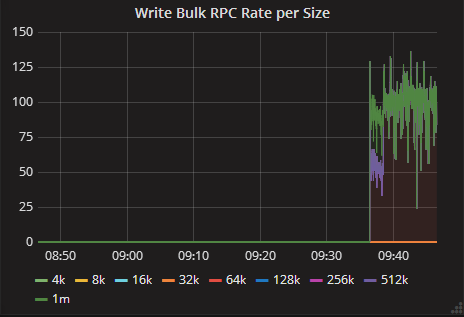
每种类型的元数据操作速率面板() 显示了Lustre文件系统每种类型的元数据操作速率信息，其单位为Ops， Operation Per Second。其包括平均值、最大值和当前值。当前测试用例为删除一个打目录下的所有文件的元数据操作速率统计。

图24: 每种类型元数据操作速率面板



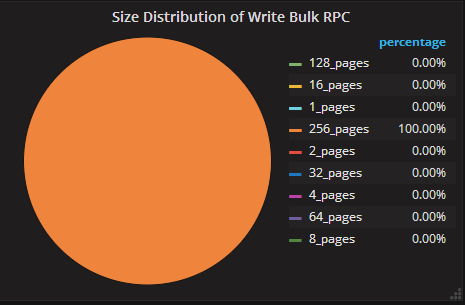
不同块大小的写RPC速率面板() 显示了Lustre文件系统不同块大小的写RPC速率随时间的变化。Lustre文件系统统计了不同bulk RPC大小信息，从4K到16M，下图显示了不同bulk 大小的RPC的速率信息。图中所示的测试用例为：在两个客户端分别运行”dd if=/dev/zero of=/mnt/lustre/test1 bs=1M oflag=direct”和“dd if=/dev/zero of=/mnt/lustre/test2 bs=64k oflag=direct”测试，收集到的信息。

图25: 不同块大小的写RPC速率面板



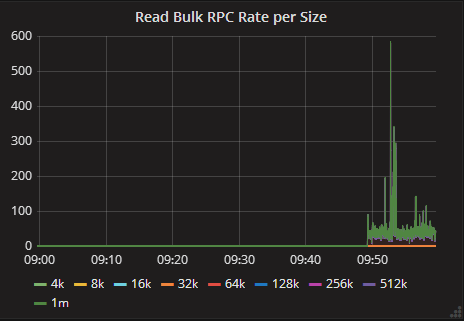
写bulk RPC大小分布面板() 显示了Lustre文件系统不同bulk大小的写RPC数目的比例信息。如图所示，大小为256页的写bulk RPC数目占总的写RPC数目的百分比为100%。

图26: 写bulk RPC大小分布面板



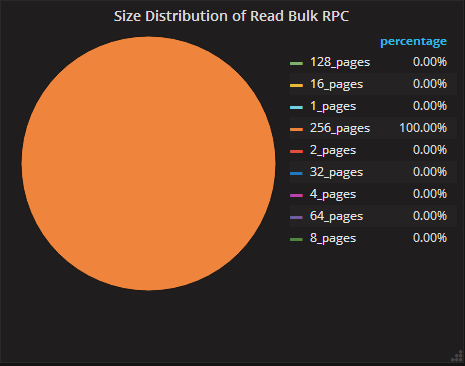
不同大小的读bulk RPC速率面板(如) 显示了Lustre文件系统不同bulk大小的读RPC速率随时间的变化。Lustre文件系统统计了不同bulk RPC大小信息，从4K到16M（Lustre中最大RPC bulk I/O大小为16MB），下图显示了不同bulk 大小的读RPC的速率信息。图中所示的测试用例为：在两个客户端分别运行”dd if=/mnt/lustre/test1 of=/dev/zero bs=1M iflag=direct”和“dd if=/mnt/lustre/test2 of=/dev/zero bs=64k iflag=direct”测试，收集到的信息。

图27: 不同大小的读bulk RPC速率面板



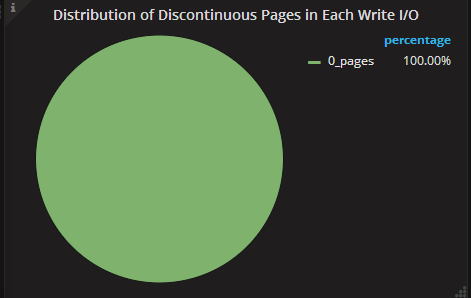
读bulk RPC大小分布面板(如) 显示了Lustre文件系统不同bulk大小的读RPC数目比例信息。如图所示，该图正在进行的测试用例为”dd if=/mnt/lustre/file of=/dev/zero bs=1M”，大小为256页的读bulk RPC数目占总的RPC数目的百分比为100%。

图28: 读bulk RPC大小分布面板



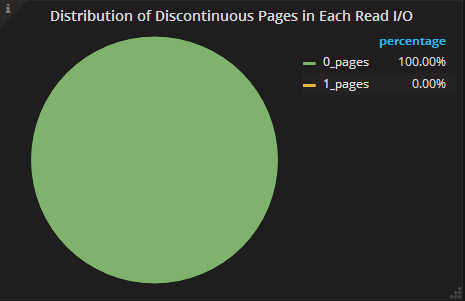
在Lustre 一次读写I/O中，如果写或者读的下一个页面与上一个读写页面的结尾（也就是下一个偏移量）不连续的话，那么就认为这个页面是不连续页面。在一次I/O RPC中可能有多个不连续页面。不连续的页面越少，底层的磁盘系统就能获取更好的性能。写I/O不连续页面分布面板() 显示了Lustre文件系统写I/O不连续页面数目比例信息。如图所示，不连续页面为“0\_pages”数目占的百分比为100%，说明所有的页面都是连续的。

图29: 写I/O不连续页面分布面板



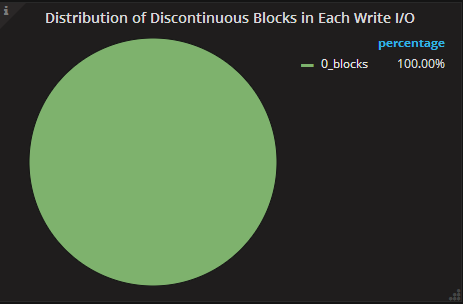
读I/O不连续页面分布面板() 显示了Lustre文件系统读I/O不连续页面数目比例信息。如图所示，不连续页面为“0\_pages”数目占的百分比为100%，说明几乎所有的读页面都是连续的。

图30: 读I/O不连续页面分布面板



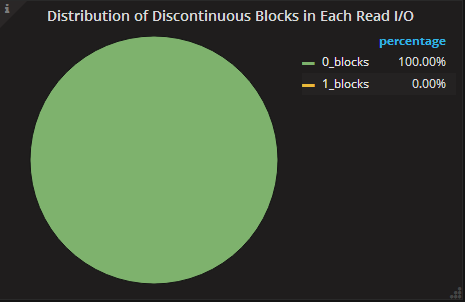
写I/O不连续块分布面板() 显示了Lustre文件系统写I/O不连续块I/O的数目比例信息。在Lustre 一次读写I/O中，不连续块的含义与不连续页的含义类似。一个块中含有的页面数量由底层文件系统（Ldiskfs）决定。I/O中存在不连续块，则意味着这里一定存在不连续页；但反之不一定成立。如图所示，写I/O不连续块为“0\_blocks”数目占的块I/O百分比为100%，说明几乎所有的写I/O都都是连续的。

图31: 写I/O不连续块分布面板



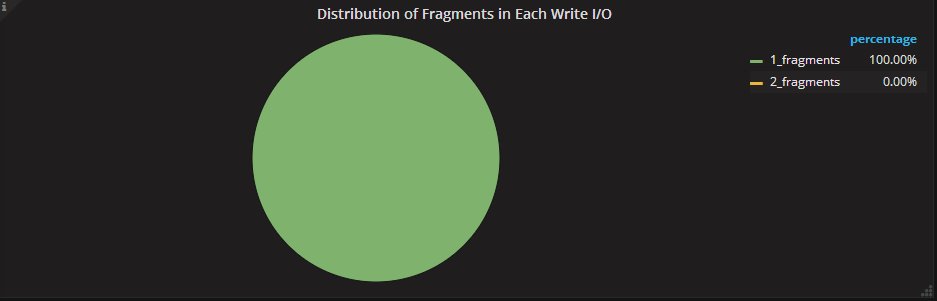
读I/O不连续块分布面板() 显示了Lustre文件系统读I/O不连续块I/O的数目比例信息。如图所示，不连续块为“0\_blocks”数目占的块I/O百分比为100%，说明几乎所有的读I/O都没有被分裂，都是连续的。

图32: 读I/O不连续块分布面板



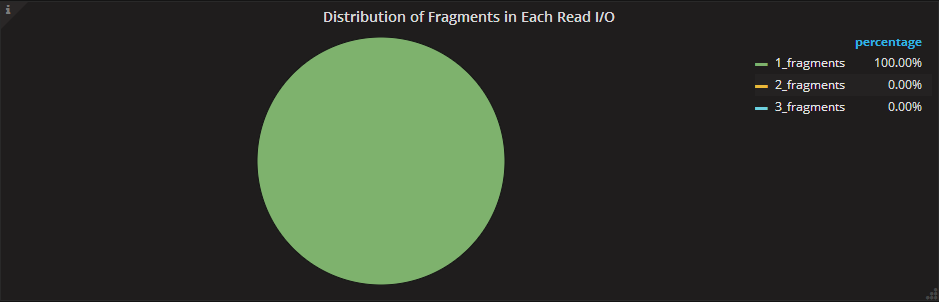
在将读写I/O中提交到磁盘上过程中，由于各种原因（例如单次I/O页面过大），Lustre OSD发起的单个I/O可能被分裂为多个的磁盘I/O。写I/O碎片分布面板() 显示了Lustre文件系统每个写I/O分裂成多个磁盘 I/O的数目比例信息。如图所示，碎片为1表示I/O没有被分裂，“1\_fragments”数目占的块I/O百分比为100%，说明几乎所有的写I/O都没有被分裂，都是整块连续的；“2\_fragments”表示Lustre写I/O被分裂为两个磁盘块I/O，其占用的百分比为0%。

图33: 每个写I/O碎片分布面板



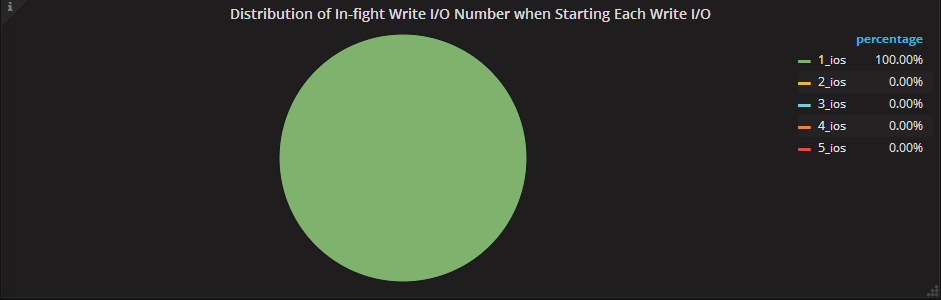
读I/O碎片分布面板(如) 显示了Lustre文件系统每个读I/O分裂成碎片化磁盘 I/O的数目比例信息。如图所示，碎片为1表示I/O没有被分裂，“1\_fragments”数目占的块I/O百分比为100%，说明几乎所有的读I/O都没有被分裂碎片化，都是连续的；“2\_fragments”表示Lustre读I/O被分裂为两个磁盘块I/O，其占用的百分比为0%。

图34: 读I/O碎片分布面板



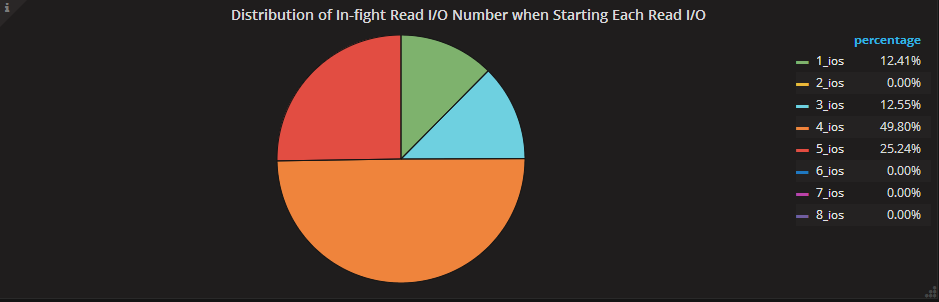
已提交等待结束的磁盘写I/O分布面板() 显示了当前OSD已提交等待结束的磁盘写I/O比例信息。如图所示，”1\_ios”表示当前正在运行的磁盘I/O为1个，其所占比例为100%。

图35: 已提交等待结束的磁盘写I/O分布面板



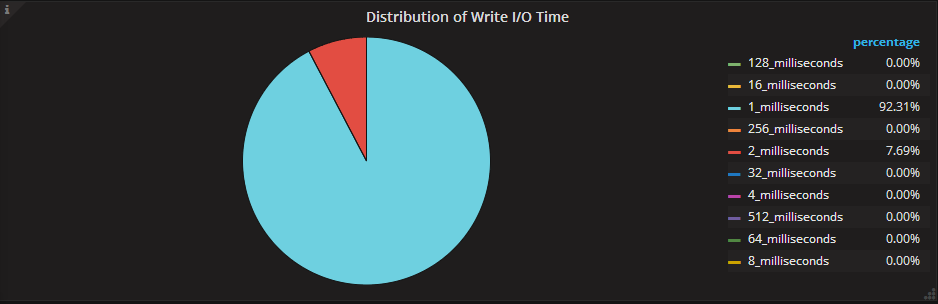
已提交等待结束的磁盘读I/O分布面板() 显示了当前OSD已提交等待结束的（正在运行过程中）磁盘读I/O比例信息。如图所示，”1\_ios”表示当前正在运行的磁盘I/O为1个，其所占比例为12.41%；“3\_ios”表示当前挂起的磁盘I/O数目为3，其占的百分比为12.55%；”4\_ios”表示当前挂起的磁盘I/O数目为4，其占的百分位49.80%。

图36: 已提交等待结束的磁盘读I/O分布面板



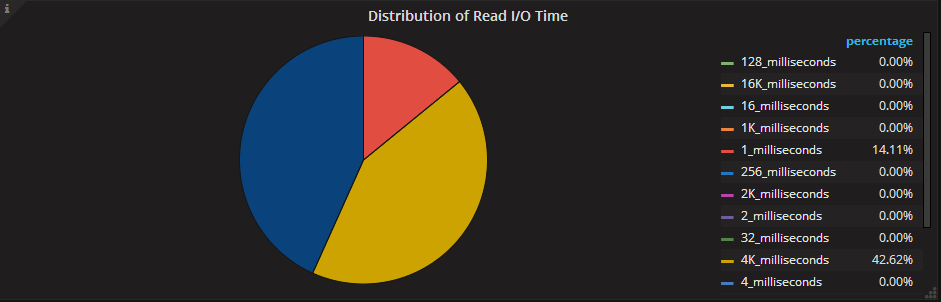
写I/O时间分布面板() 显示了当前OSD写I/O时间分布比例信息。如图所示，”1\_milliseconds”表示写I/O时间小于1毫秒的I/O次数所占I/O次数的百分比，其所占比例为92.31%；“2\_milliseconds”表示写I/O时间在1毫秒和2毫秒之间的I/O次数所占百分比，其值为7.69%。

图37: 写I/O时间分布面板



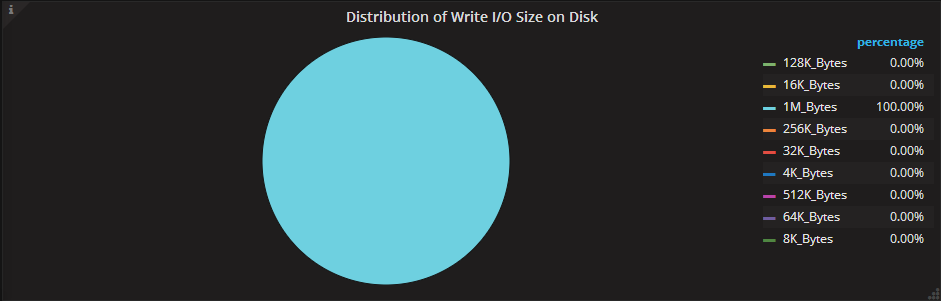
读I/O时间分布面板() 显示了OSD读I/O时间分布比例信息。如图所示，”1\_milliseconds”表示读I/O时间小于1毫秒的I/O次数所占I/O次数的百分比，其所占比例为14.11%；“4K\_milliseconds”表示写I/O时间在2K毫秒和4K毫秒之间的I/O次数所占百分比，其值为42.62%。

图38: 读I/O时间分布面板



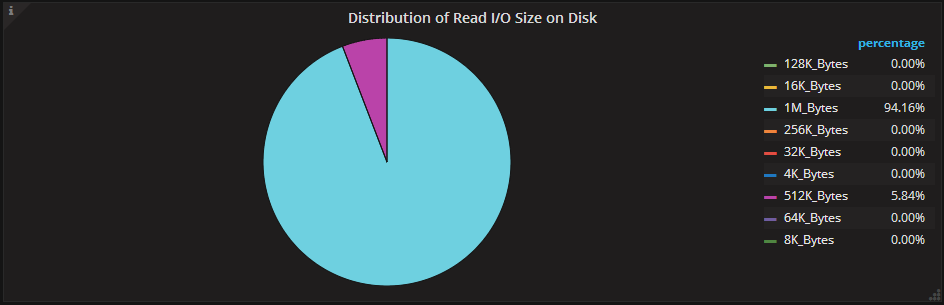
磁盘写I/O大小分布面板() 显示了OSD写I/O大小分布比例信息。如图所示，”1M\_Bytes”表示磁盘写I/O大小为512K字节到1M字节之间的I/O次数的所占百分比，其值为100%。

图39: 磁盘写I/O大小分布面板



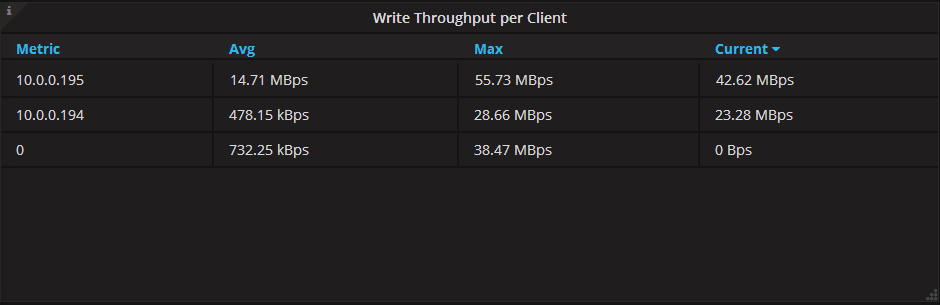
磁盘读I/O大小分布面板() 显示了OSD读I/O大小分布比例信息。如图所示，”1M\_Bytes”表示磁盘读I/O大小为512K字节到1M字节之间的I/O次数的所占百分比，其值为94.16%；“512K\_Bytes”表示磁盘读I/O大小在256K字节到512K字节之间的I/O次数所占百分比，其值为5.84%。

图40: 磁盘读I/O大小分布面板



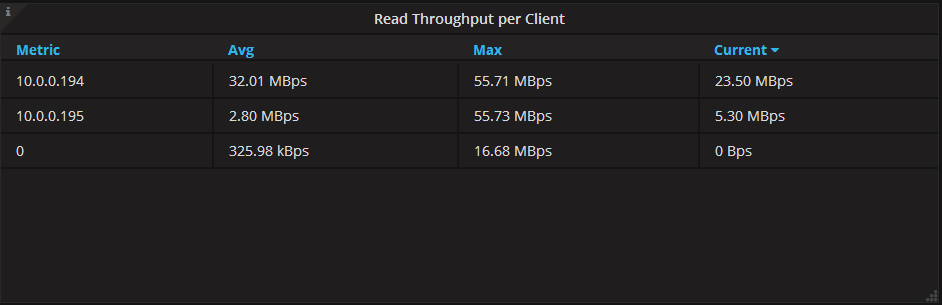
每个客户端写吞吐量面板（）显示了每个客户端写吞吐量信息，包括平均值，最大值和当前值。图中表示IP地址为“10.0.0.195”的客户端的平均I/O吞吐量为14.71MBps，最大值为55.73MBps，当前吞吐量为42.62MBps。

图41: 每个客户端写吞吐量面板



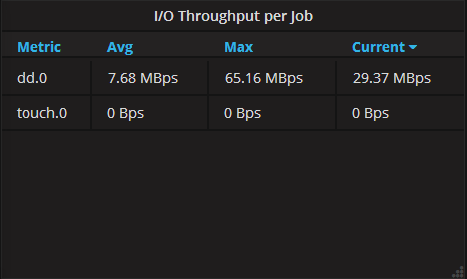
每个客户端读吞吐量面板（）显示了每个客户端读吞吐量信息，包括平均值，最大值和当前值。图中表示IP地址为“10.0.0.194”的客户端的平均读吞吐量为32.01MBps，最大值为55.71MBps，当前吞吐量为23.50MBps。

图42: 每个客户端读吞吐量面板



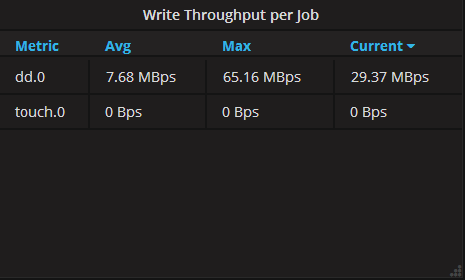
每个作业I/O吞吐量面板（）显示了每个作业I/O吞吐量信息，包括平均值，最大值和当前值。图中JOBID为“dd.0”的作业的平均I/O吞吐量为7.68MBps，最大值为65.16MBps，当前吞吐量为29.37MBps。

图43: 每个作业I/O吞吐量面板



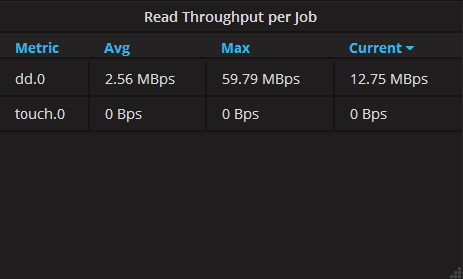
每个作业写吞吐量面板（）显示了每个作业写吞吐量信息，包括平均值，最大值和当前值。图中JOBID为“dd.0”的作业的平均写吞吐量为7.68MBps，最大值为65.16MBps，当前吞吐量为29.37MBps。

图44: 每个作业写吞吐量面板



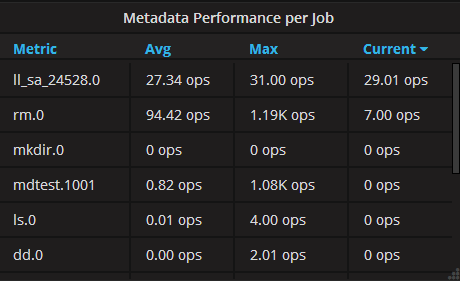
每个作业读吞吐量面板（）显示了每个作业读吞吐量信息，包括平均值，最大值和当前值。图中JOBID为“dd.0”的作业的平均读吞吐量为2.56MBps，最大值为59.79MBps，当前吞吐量为12.75MBps。

图45: 每个作业读吞吐量面板



每个作业元数据性能面板（）显示了每个作业元数据性能信息，包括平均值，最大值和当前值，其单位为Ops （Operations per Second）。图中JOBID为“rm.0”的作业的平均元数据性能为94.42ops，最大值为1.19K ops，当前性能为7.00 ops。

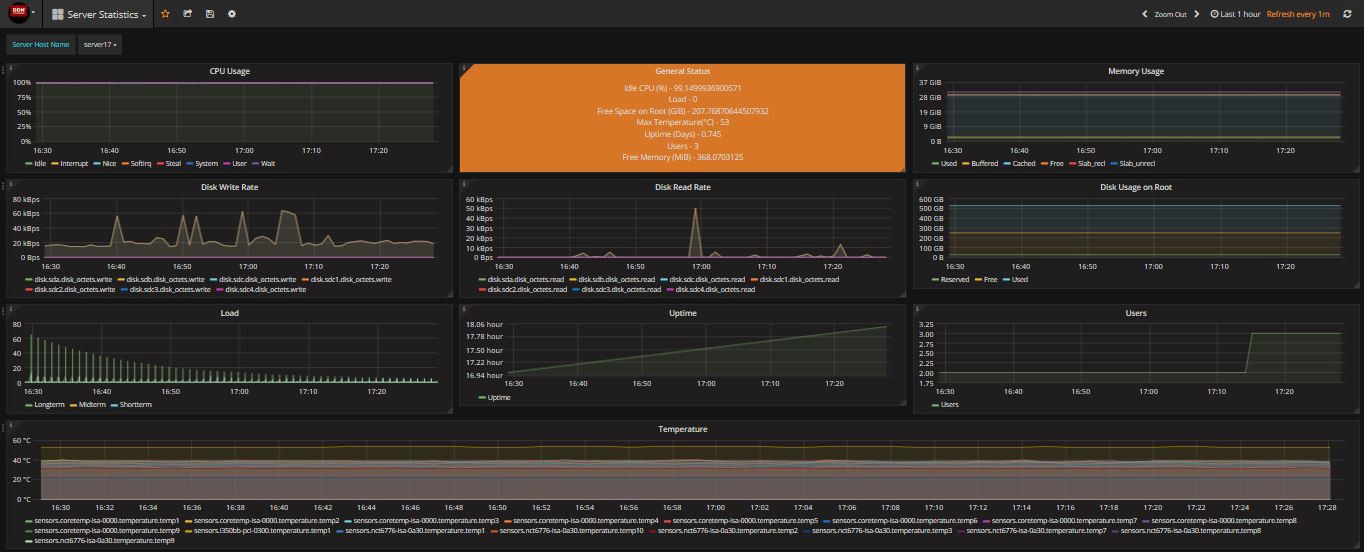
图46: 每个作业元数据性能面板



## 服务器仪表盘

服务器仪表盘 () 显示了服务器的统计信息。

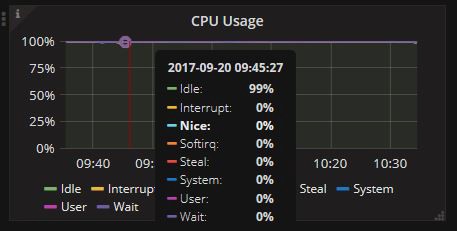
图47：服务器仪表盘



以下是在服务器仪表盘中可以找到的面板：

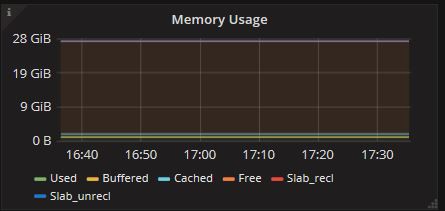
* CPU使用率面板（）显示了CPU在不同状态下的使用率，即CPU在执行用户代码、执行系统代码、等待数据读写操作、空闲时所占用的时间。

图48：CPU 使用率面板



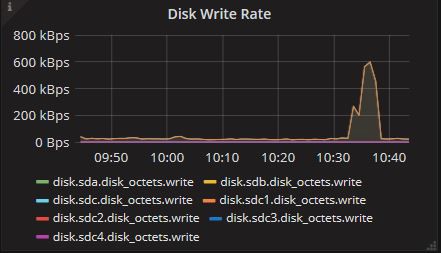
* 内存使用率面板（）显示了内存的使用情况，包括：Used, Buffered, Cached, Free, Slab\_recl, Slab\_unrecl。

图49：内存使用率面板



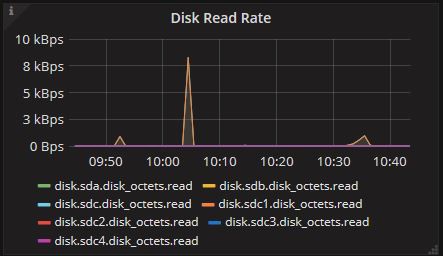
* 磁盘写速率面板（）显示出服务器磁盘写入速率随时间的变化。

图50：磁盘写速率面板



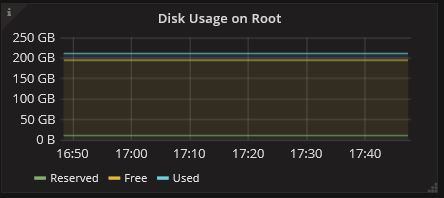
* 磁盘读速率面板（）显示出服务器磁盘写入速率随时间的变化。

图51：磁盘读速率面板



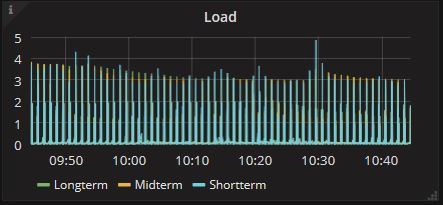
* 根目录磁盘使用率面板（图52）显示了根目录的空闲空间、使用空间、预留空间。

图52：根目录磁盘使用率



* 负载面板（）显示了服务器负载情况，即服务器运行队列中可运行的任务数量。相关信息由操作系统提供，分为三类：
  + **Shortterm（短期）**— 一分钟内的平均负载。
  + **Midterm（中期）**—五分钟内的平均负载。
  + **Longterm（长期）**—十五分钟内的平均负载。

图53：负载面板



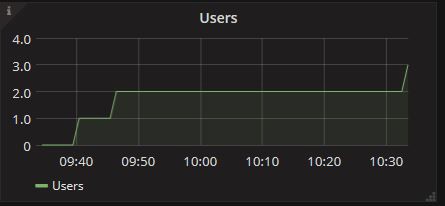
* 启动时间面板（图54）显示出服务器已经启动了多长时间。

图54：启动时间面板



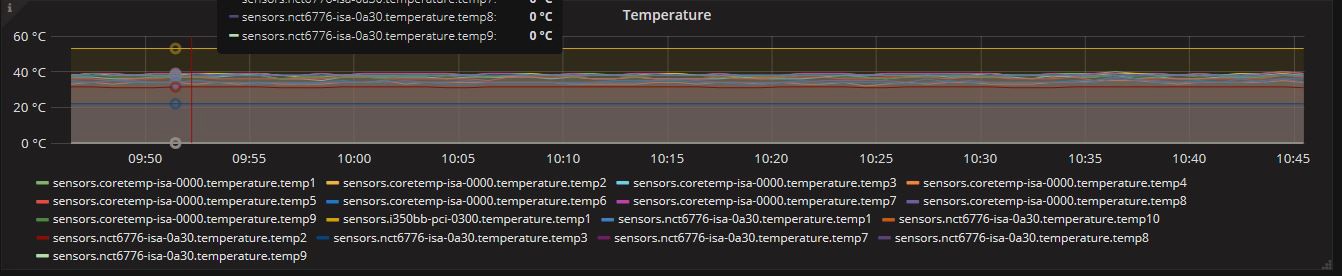
* 用户数面板（图55）显示出登陆系统的用户数量。

图55：用户数面板



* 温度面板（图56）显示出从各温度传感器中收集的温度信息。

图56：温度面板



## SFA 物理磁盘仪表盘

SFA物理磁盘仪表盘**（**）显示了SFA物理磁盘的相关信息。

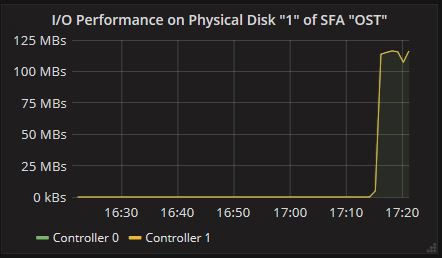
图57: SFA物理磁盘仪表盘



下面是可以在SFA物理磁盘表盘中可以找到的面板：

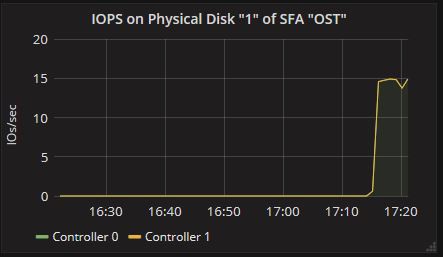
* I/O 吞吐率面板（图58）显示出该物理磁盘的I/O吞吐率。

图58：I/O 吞吐率面板



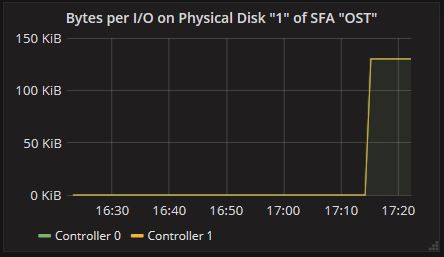
* IOPS面板（图59）显示出改物理磁盘每秒I/O操作数目。

图59：IOPS面板



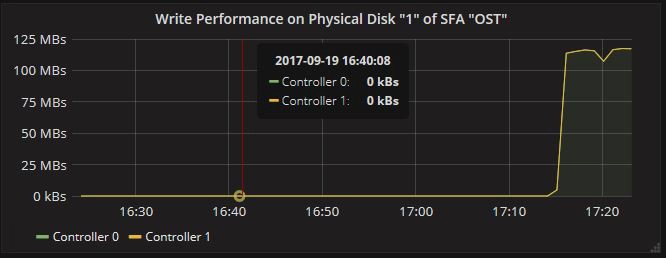
* I/O大小面板(图60) 显示出各控制器中该物理硬盘的每次I/O的平均大小。

图60：I/O 大小面板



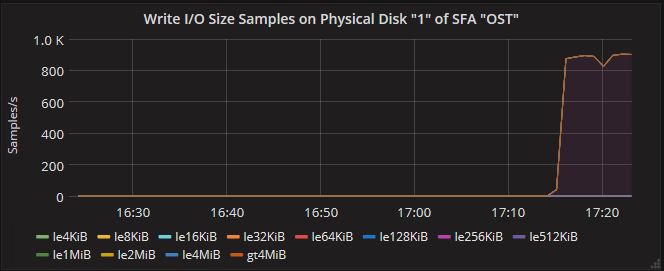
* 写数据吞吐率面板（图61）显示出各控制器中，该物理硬盘的写入吞吐率。

图61：写数据吞吐率面板



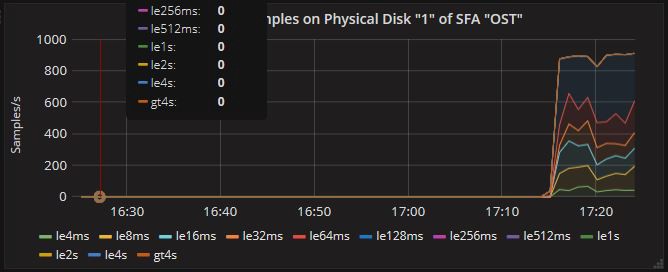
* 写数据的I/O大小样品采样数目面板（图62）显示出该物理磁盘上，每种I/O大小的采样数目。

图62：写数据的I/O大小样品采样数目面板



* 数据写入延迟样品采样数目面板（图63）显示出显示出该物理磁盘上，每种I/O延迟的采样数目。

图63：数据写入延迟样品采样数目面板



## SFA虚拟磁盘表盘

SFA虚拟磁盘表盘（）显示出SFA虚拟磁盘表盘的信息。

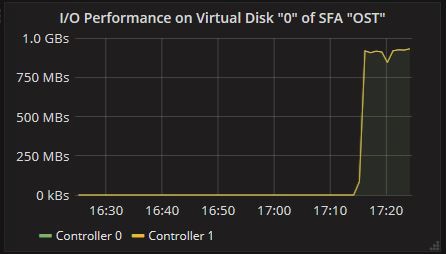
图64：SFA虚拟磁盘表盘



下面是可以在SFA虚拟磁盘表盘中找到的面板：

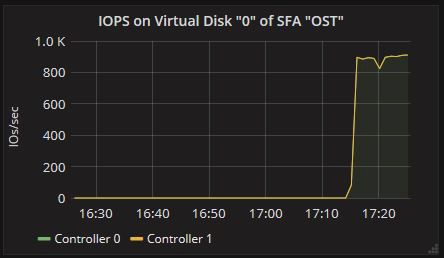
* I/O吞吐率面板（图65）显示出该虚拟磁盘上的I/O吞吐率。

图65：I/O吞吐率面板



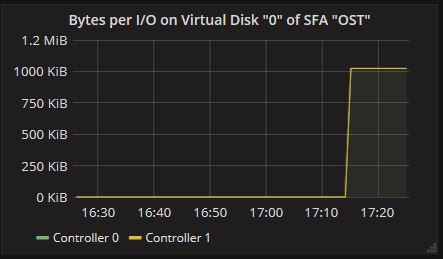
* IOPS面板（图66）显示出在不同的控制器上，该虚拟磁盘上的IOPS。

图66：IOPS面板



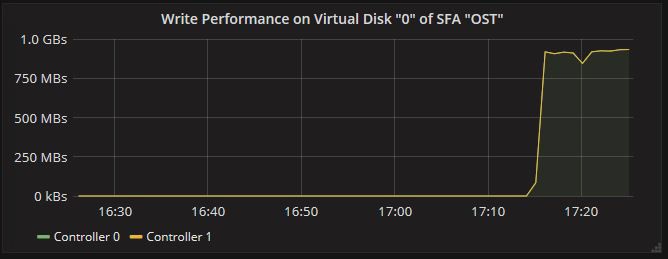
* I/O大小面板（图67）显示出各控制器上，该虚拟磁盘每次I/O的平均大小。

图67：Bytes per I/O on Virtual DiskPanel



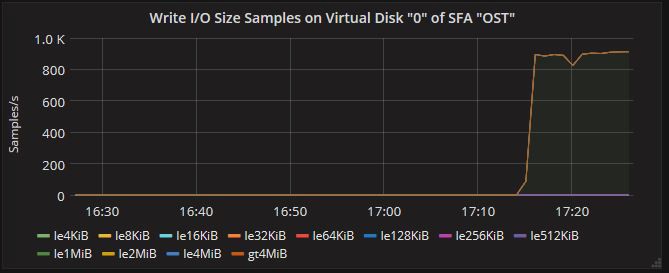
* 数据写入速率面板（图68）显示出每个控制器上，该虚拟磁盘的数据写入速率。

图68：Write Performance on Virtual DiskPanel



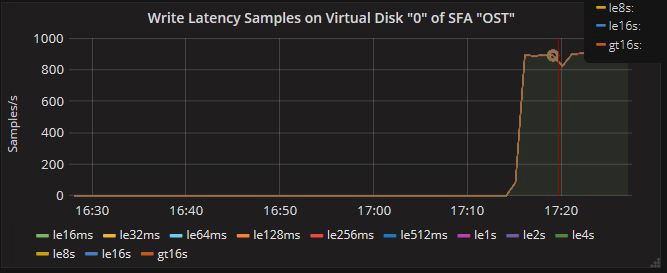
* 数据写入的I/O大小样品采集数目面板（图69）显示出该SFA虚拟磁盘上，数据写入的I/O大小的分布。

图69：数据写入的I/O大小样品采集数目面板



* 数据写入的延迟样品采集数目面板（）显示出该SFA虚拟磁盘上，数据写入延迟的分布。

图70：数据写入的延迟样品采集数目面板



# 压力测试

为了验证监控在高负载的集群环境下是否运行良好，我们设计了collectd-stress2 插件来进行压力测试。它是collectd-stress 插件的升级版，它可以通过几个collectd客户端来模拟几百个服务器收集的数以万计的监控数据。

## 在部署客户端安装插件

由于stress2插件会产生大量的模拟的监控数据，污染数据库，所以该插件默认不会在所有的客户端安装。当使用esmon\_install 部署完监控系统之后，用户可以选择某个客户端作为压测客户端。可以在ISO目录下找到collectd-stress2\*.rpm并通过以下命令进行安装：

rpm --ivh collectd-stress2\*.rpm

## 在部署客户端更新配置文件

安装stress2 RPM之后，更新配置文件/etc/collectd.conf, 添加下面的配置：

* **Thread**—定义了测试线程数目**。**
* **Metric**—定义了一个监控目标的所有属性,它可以指定多次，用来同时模拟不同的监控目标**，**它包含以下属性:
  + **Variable**—定义了监控目标变化的范围以及变化的速度，可以指定多次。
    - **Name**—定义了变量名。
    - **Number**—定义了变量变化的范围最大值。
    - **UpdateIterval**—定义了变量变更的间隔时间。
  + **Host**—定义了客户端的主机名，通常定义为"${key:hostname}"，由程序自动设置当前主机名，它和下面的**Plugin**，**PluginInstance**，**Type**，**TypeInstance**共同组成collectd 描述收集数据对象的区分符，详情请参见[Collectd 的描述符](https://collectd.org/wiki/index.php/Naming_schema)**。**
  + **Plugin**—定义了collectd 识别符中的plugin 成员**。**
  + **PluginInstance**—定义了collectd 识别符中的plugininstance 成员。
  + **Type**—定义了collectd 识别符中的type 成员,详细可参见[数据类型](https://collectd.org/wiki/index.php/Derive)**。**
  + **TypeInstance**—定义了collectd 识别符中的type instance成员。
  + **TsdbName**—定义了提交到数据库格式中的名字。
  + **TsdbTags**—定义了提交到数据库格式的标签，方便后期分类查找。

**下面是一个**/etc/collectd.conf**的例子。**

**例子:**

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>

## 进行压力测试

在更改了配置文件之后，重启Collectd:

service collectd restart

通过查看/var/log/messages 日志，可以看到如下消息：

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

上面信息显示stress2 插件已经成功加载运行，并且产生了大量的监控数据。

下面将以上面的配置文件为例，分析在如下指定的硬件环境下，如何分析对应的监控瓶颈。

* **操作系统版本：**CentOS7。
* **内存：**128GB。
* **处理器:** Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz。
* **硬盘：**Samsung SSD 850 2B6Q。

监控客户端和服务器端运行在同一台服务器上，Influxdb 数据存在SSD对应的Ext4文件系统上。

**前提条件：**

* **Collectd 收集数据间隔：**60秒。
* **Grafana 展示历史时间长度：**1小时。
* **Grafana 自动刷新间隔：**60 秒。
* **Collectd 运行时间：大于1个小时。**

**结论：**

* **Grafana 页面一直处于刷新状态：**监控过载。
* **Grafana 页面有空闲：**监控运行良好。

理论上Grafana 的刷新时间，等于数据库查询时间加上页面布局的加载时间。

我们可以通过查询数据库，来衡量数据库的查询性能，比如针对ESMON Grafana 默认的“Read Throughput per Job”查询，数据库端相应的查询语句：

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""

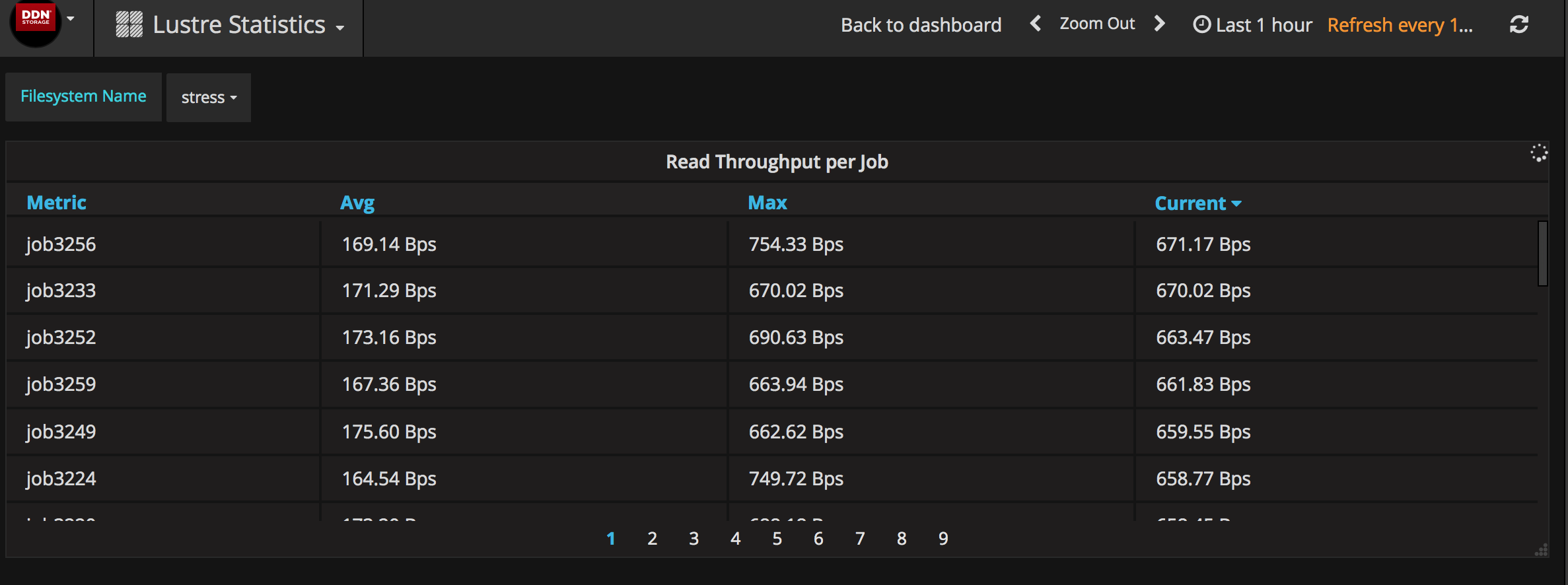
随着监控软件的运行，可以在数据库后端的host 上执行该条语句，验证数据库端查询的时间。

如图71所示，Influxdb 在一个小时以内，数据查询时间呈线性增长，因为查询获得的数据量会随着时间增长而增长。但是一个小时以后，查询时间稳定在二十多秒。

图71：Influxdb Query Time 变化曲线图

验证完了数据库端的负载，我们还需要验证Grafana界面的加载情况，如图72 所示，通过Grafana登陆查看“Read Throughput per Job”。

图72：每个任务读吞吐压测面板



如果页面一直处于刷新状态，页面在60秒以内无法加载出来，则说明当前配置下监控处于过载状态，说明当前需要升级监控的硬件配置，或者增大收集间隔。否则则表明监控运行良好。通过不断调整*/etc/collectd.conf中的*job\_id的Number，观察页面加载延迟，可以得到当前硬件配置下可支持的最大压力。测试表面，该设置条件下，监控可以支持10 个OST上并行运行的7000个作业。

# 故障排查

部署服务器收集了所有可用于调试的日志，并将其保存至的*/var/log/esmon\_install/[installing\_date]*目录下。一旦操作失败，相关错误信息将被输出至*/var/log/esmon\_install/[installing\_date]/error.log*文件中。通常，第一条错误信息包含了导致该操作失败的原因。