
# Experiences with Lustre* and Hadoop*

**Gabriele Paciucci (Intel)**

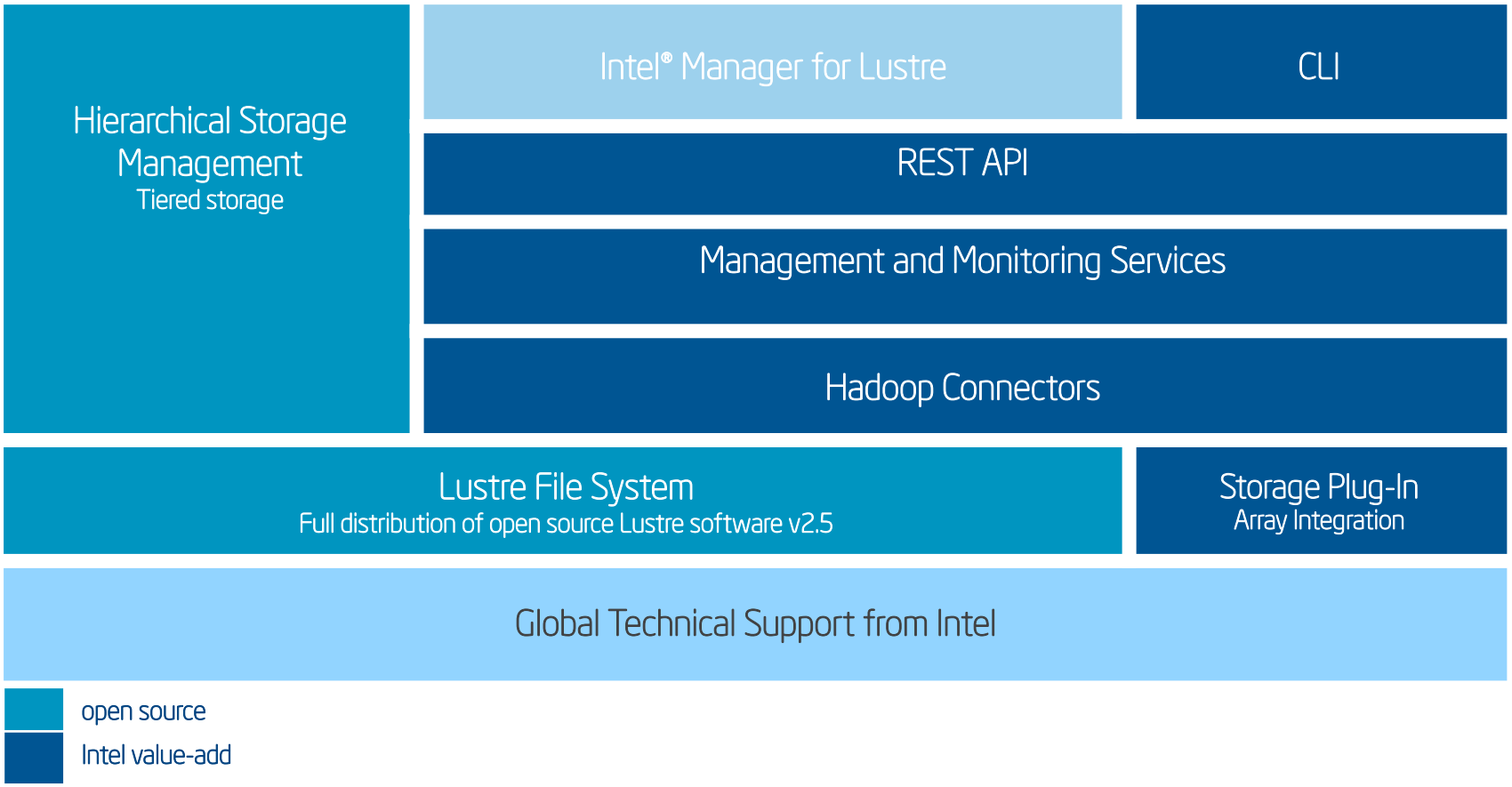
**June, 2014**

** Some name and brands may be  claimed as the property of others.*

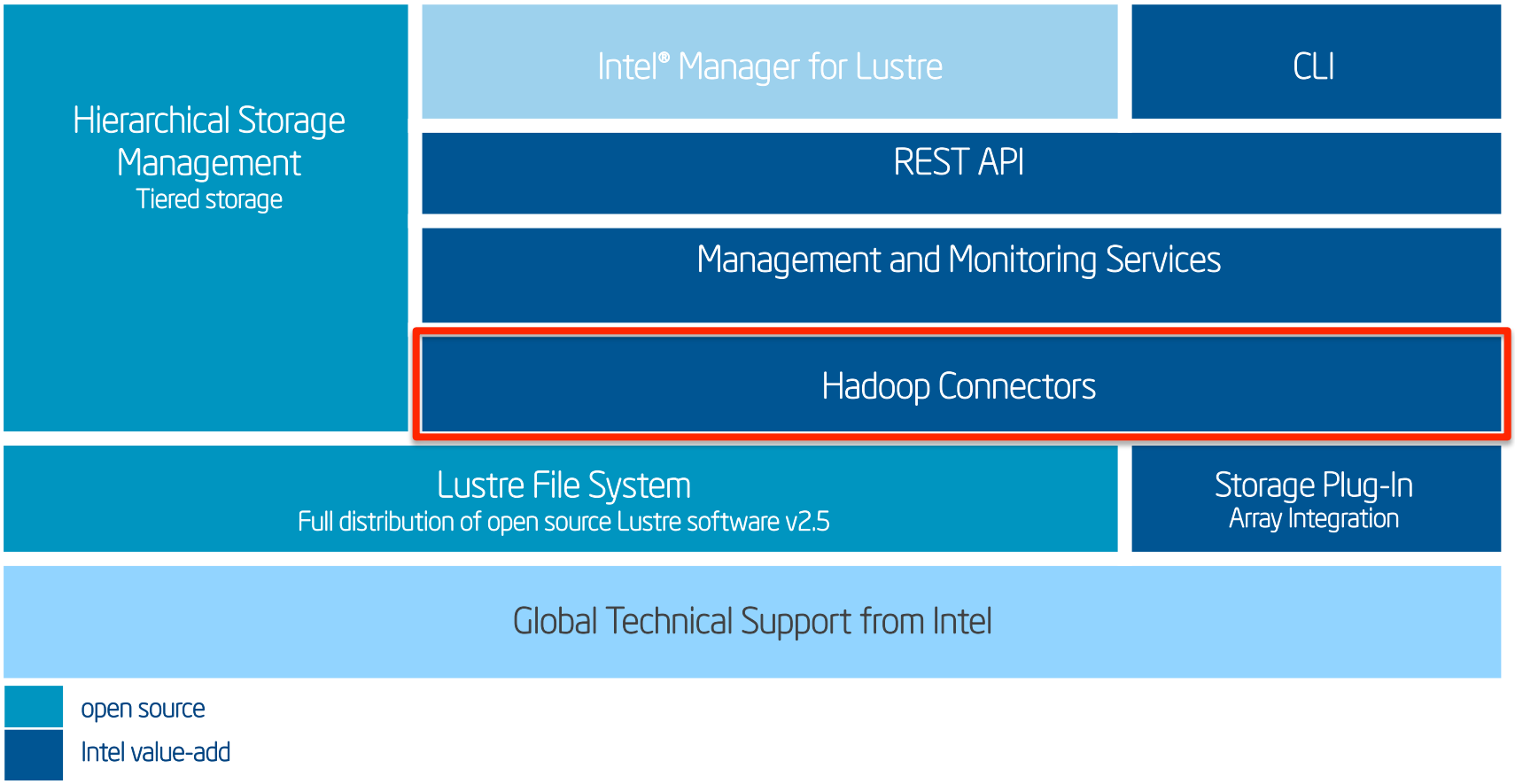
# Agenda

- Overview
- Intel Enterprise Edition for Lustre 2.0
- How to configure Cloudera Hadoop with Lustre using HAL
- Benchmarks results
- Conclusion

# Intel Enterprise Edition for Lustre v 2.0

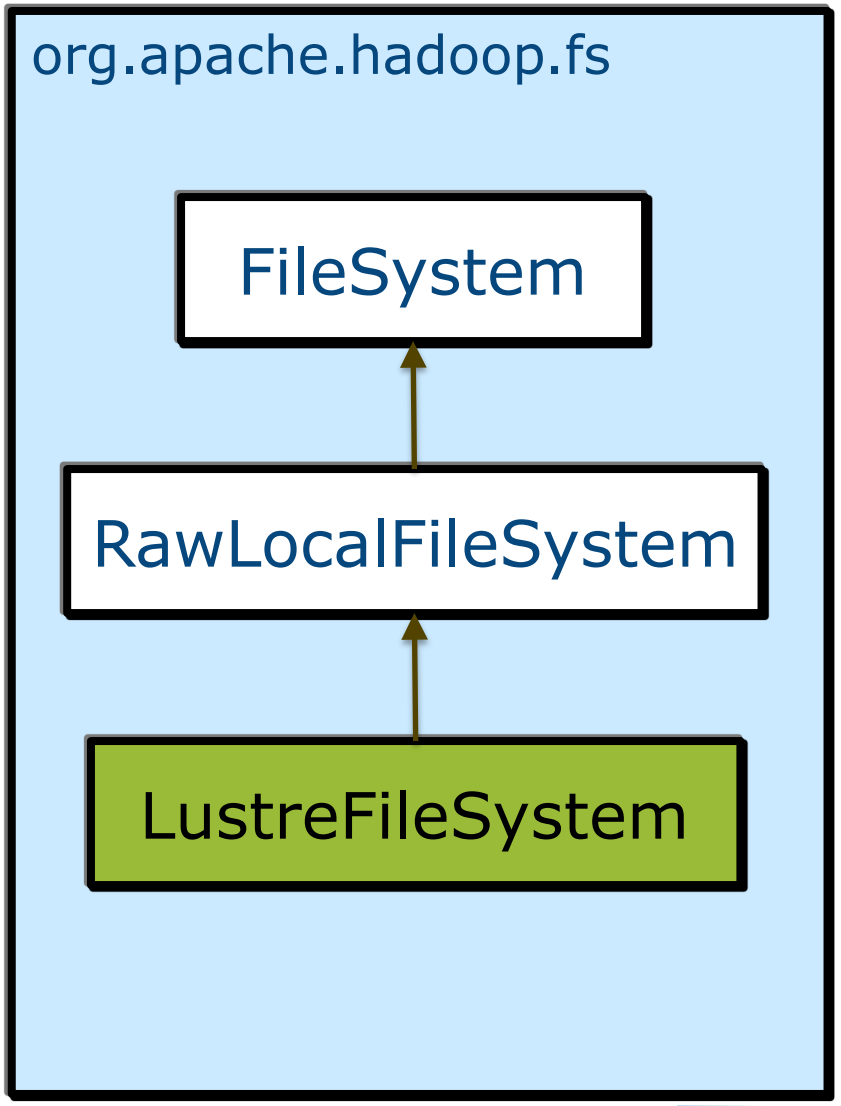


# Hadoop Connectors for Lustre (HAL and HAM)



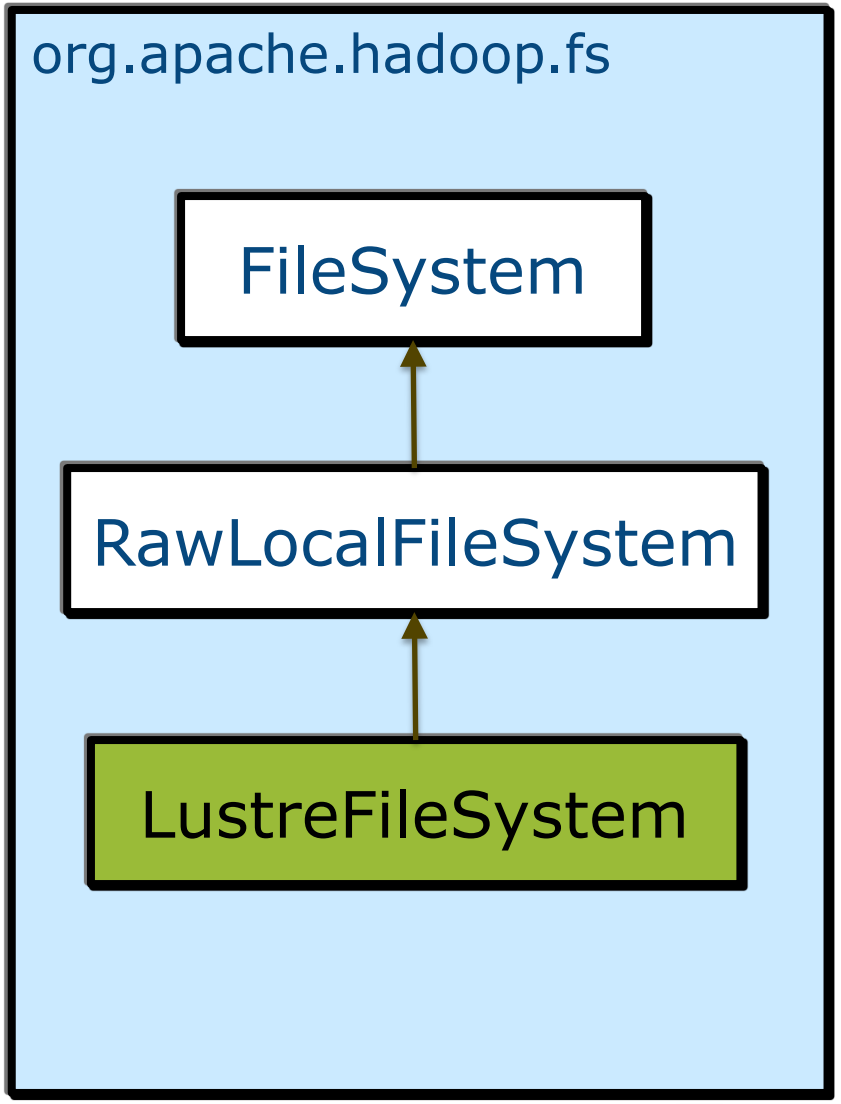
# Hadoop Adapter for Lustre (HAL)

- Replace HDFS with Lustre
- Plugin for Apache Hadoop 2.3 and CDH 5.0.1
- No changes to Lustre needed
- Enable HPC environments to use MapReduce v2 with existing data in place
- Allow Hadoop environments to migrate to a general purpose file system



# Hpc Adapter for Mapreduce (HAM)

- Replace YARN with Slurm
- Plugin for Apache Hadoop 2.3 and CDH 5.0.1
- No changes to Lustre needed
- Enable HPC environments to use Slurm as scheduler for MapReduce v2 jobs
- Allow Hadoop environments to migrate to a more sophisticated scheduler



# Why Use Lustre for Hadoop?

## Convergence of HPC and data analytics

## Desire for HPC systems to run Hadoop workloads

- Hadoop is the most popular software stack for big data analytics
- Lustre is the file system of choice for HPC clusters

## But, HDFS expects nodes with locally attached disks

## Most HPC clusters are diskless compute nodes

## Benefits of using Lustre for Hadoop applications

- Improved application performance – without changes to app
- Management simplicity lowers costs
- More efficient and productive storage resources
- No data transfer overhead for staging inputs and extracting results
- No need to partition storage into HPC (Lustre) and Analytics (HDFS)

# Our Setup Overview

1. Preparation
2. Install and setup Intel Enterprise Edition Lustre 2.0
3. Mount Lustre on Hadoop's compute nodes
4. Install and setup Cloudera Hadoop Distribution 5.0.1 and HAL (included in IEEL 2.0)
5. Direct Hadoop IOs to Lustre instead of HDFS

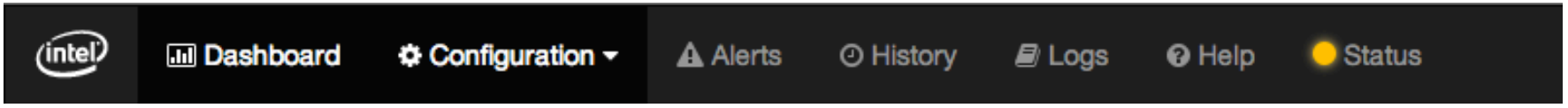
# Preparation

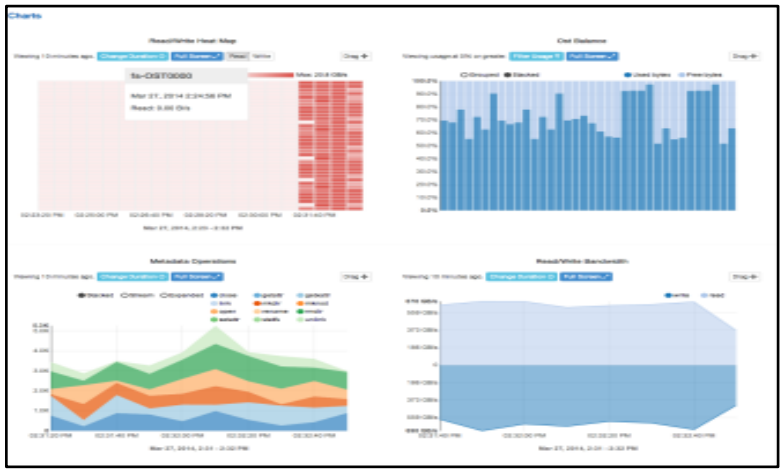
- Consistent UID and GID, especially for the Hadoop users
  - The best way is to setup the global naming server and connect Lustre server and Hadoop server there.
  - For our lab, we used these commands:

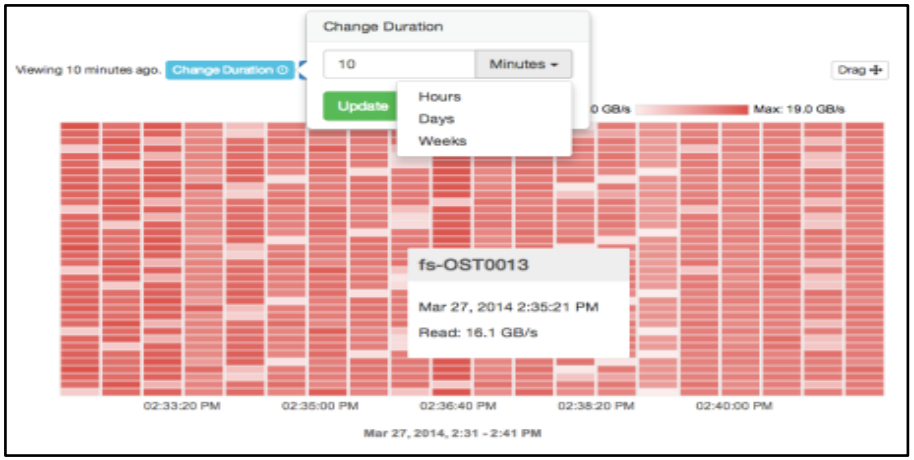
```
#groupadd -g 499 zookeeper
#groupadd -g 495 mapred
#groupadd -g 497 yarn
#groupadd -g 498 hadoop
#groupadd -g 496 hdfs
#adduser -u 496 -g 496 hdfs
#adduser -u 495 -g 495 mapred
#adduser -u 497 -g 497 yarn
#adduser -u 498 -g 499 zookeeper
#groupmems -g hadoop -a yarn;
#groupmems -g hadoop -a mapred;
#groupmems -g hadoop -a hdfs;
#usermod -d /var/lib/hadoop-yarn -s /bin/bash yarn;
#usermod -d /var/lib/hadoop-mapreduce -s /bin/bash mapred;
#usermod -d /var/lib/hadoop-hdfs -s /bin/bash hdfs
#usermod -d /var/run/zookeeper -s /sbin/nologin zookeeper
```

# Install and Setup Intel Enterprise Edition Lustre 2.0

- Intel Manager for Lustre is the value added software tool to manage and monitor Intel Enterprise Edition for Lustre 2.0
- IML simplify the complexity of Lustre
- IML installed and configured in managed mode to collect throughput data
- We used Lustre version 2.5.15 shipped with IEEL 2.0 RC2





# Mount Lustre on Hadoop's compute nodes

- On all of Hadoop nodes, we mounted the same Lustre file system before installing any Hadoop software.

```
# mount -t lustre 192.168.1.39@o2ib:/lustre /scratch/

# df -h
Filesystem                  Size  Used Avail Use% Mounted on
/dev/sda2                   444G   22G  400G   6% /
192.168.1.39@o2ib0:/lustrefs  1.6T  160G  1.4T  11% /scratch
```

# Install and Setup Cloudera Hadoop Distribution 5.0.1 and HAL (included in IEEL 2.0)

- On all Hadoop's compute nodes:
  - Installed Oracle JDK version 1.7.0_45 and set the JAVA_HOME
  - Installed CDH repository in rpm format

- Based on the specific role of each Hadoop's compute node, we installed the following packages:

| Role | Command used to install |
|---|---|
| Resource Manager | yum install hadoop-yarn-resourcemanager |
| History Server | yum install hadoop-mapreduce-historyserver hadoop-yarn-proxyserver |
| Node Manager | yum install hadoop-yarn-nodemanager hadoop-mapreduce |

# Install and Setup Cloudera Hadoop Distribution 5.0.1 and HAL (included in IEEL 2.0)

- On one **single** Hadoop's compute node, we set permissions on the hadoop root directory:

  *# mkdir /scratch/hadoop*
  *# chmod 0777 /scratch/hadoop*
  *# setfacl -R -m group:hadoop:rwx /scratch/hadoop*
  *# setfacl -R -d -m group:hadoop:rwx /scratch/hadoop*

- On all Hadoop's compute nodes:

  *# cp -r /etc/hadoop/conf.hdfs /etc/hadoop/conf.lustre*
  *# alternatives --install /etc/hadoop/conf hadoop-conf /etc/hadoop/conf.lustre 60*
  *# alternatives --set hadoop-conf /etc/hadoop/conf.lustre*

- On all Hadoop's compute nodes:

  *# cp ./ieel-2.0.0/hadoop/hadoop-lustre-plugin-2.3.0.jar /usr/lib/hadoop/*

# CDH 5.0.1 configuration for Lustre

- CORE-SITE.XML

| Property name | Value | Description |
|---|---|---|
| fs.defaultFS | lustre:// | Configure Hadoop to use Lustre as the default file system. |
| fs.lustre.impl | org.apache.hadoop.fs.LustreFileSystem | Configure Hadoop to use Lustre Filesystem |
| fs.AbstractFileSystem.lustre.impl | org.apache.hadoop.fs.LustreFileSystem$LustreFs | Configure Hadoop to use Lustre class |
| fs.root.dir | /scratch/hadoop | Hadoop root directory on Lustre mount point. |

# CDH 5.0.1 configuration for Lustre

- MAPRED-SITE.XML

| Property name | Value | Description |
|---|---|---|
| mapreduce.map.speculative | *false* | Turn off map tasks speculative execution (this is incompatible with Lustre currently) |
| mapreduce.reduce.speculative | *false* | Turn off reduce tasks speculative execution (this is incompatible with Lustre currently) |
| mapreduce.job.map.output.collector.class | org.apache.hadoop.mapred.SharedFsPlugins$MapOutputBuffer | Defines the MapOutputCollector implementation to use, specifically for Lustre, for shuffle phase |
| mapreduce.job.reduce.shuffle.consumer.plugin.class | org.apache.hadoop.mapred.SharedFsPlugins$Shuffle | Name of the class whose instance will be used to send shuffle requests by reducetasks of this job |

# Test HAL and run Hadoop's services

- From one node:

```
# sudo -u hdfs hadoop fs -mkdir -p /test1/test2
# sudo -u hdfs hadoop fs -ls /test*
Found 1 items
drwxrwxrwx   - hdfs hdfs       4096 2014-06-09 18:05 /test1/test2
# ls -l /scratch/hadoop/test*
total 4
drwxrwxrwx+ 2 hdfs hdfs 4096 Jun  9 18:05 test2
```

- Based on the specific role of each Hadoop's compute node, we started these services:

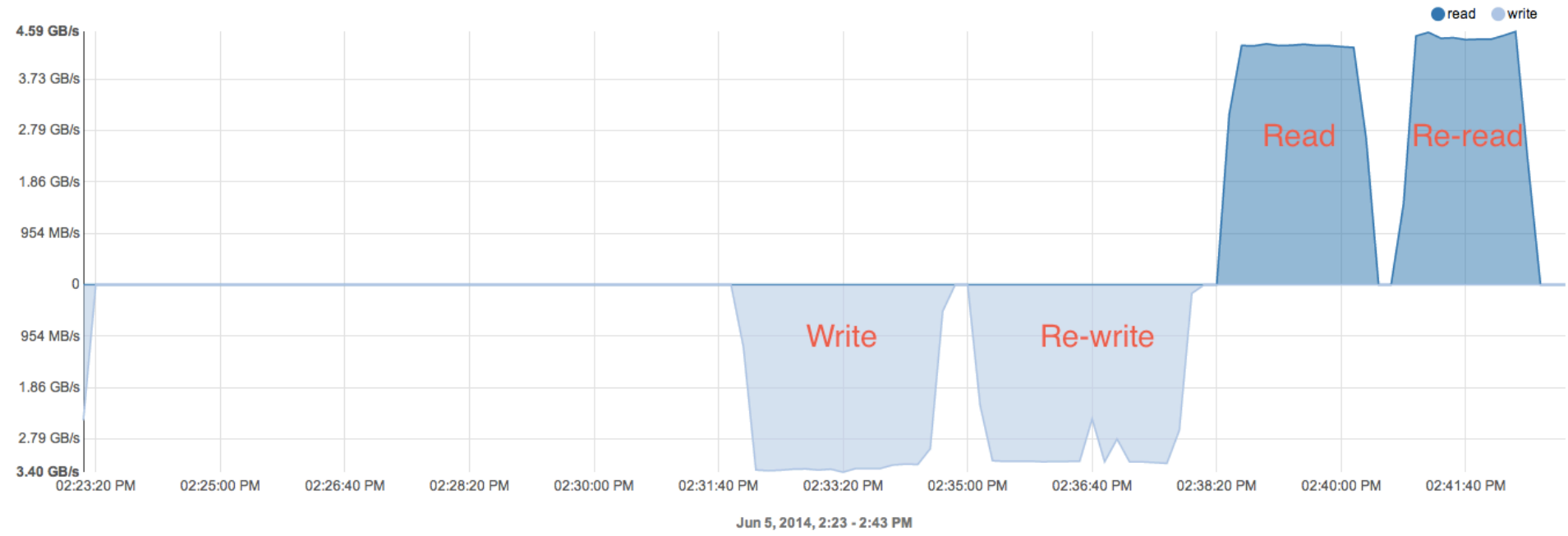
| Role | Services to start |
|---|---|
| Resource Manager | service hadoop-yarn-resourcemanager start |
| History Server | service hadoop-mapreduce-historyserver  start |
| Node Manager | service hadoop-yarn-nodemanager start |

# Benchmark

- We used iozone as baseline. Iozone is a widely adopted benchmark tools for traditional HPC
- The DFSIO benchmark is a read and write test for Hadoop's file-system. It is helpful for tasks such as stress testing Hadoop's file-system, to discover performance bottlenecks in your network, to shake out the hardware, OS and Hadoop setup of your cluster machines and to give you a first impression of how fast your cluster is in terms of IO.
- The TeraSort benchmark is probably the most well-known Hadoop benchmark. Basically, the goal of TeraSort is to sort 1TB of data (or any other amount of data you want) as fast as possible. It is a benchmark that combines testing the Hadoop's fs and MapReduce layers of an Hadoop cluster. As such it is not surprising that the TeraSort benchmark suite is often used in practice, which has the added benefit that it allows us – among other things – to compare the results of our own cluster with the clusters of other people.

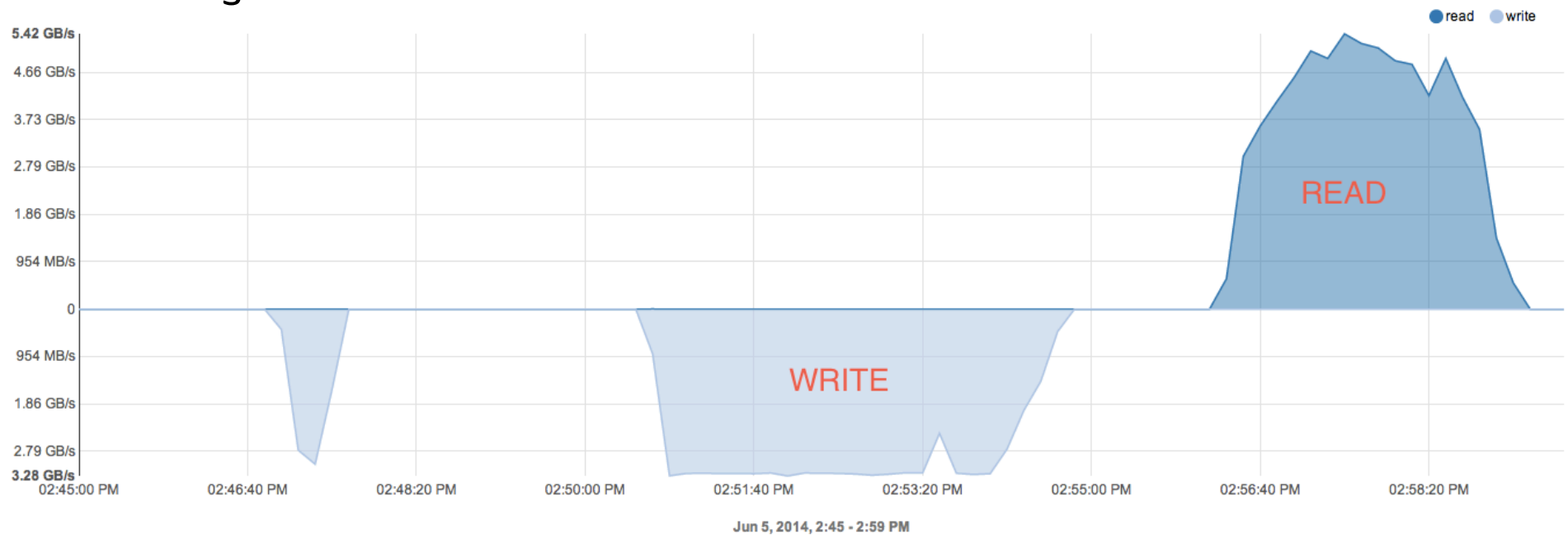
# Baseline using iozone

- In the chart a screenshot from IML of the aggregate throughput of Lustre during the iozone benchmark
- We achieved a peak performance of 3.4GB/sec writing and 4.59GB/sec reading
- Our goal is to achieve the same performance using Hadoop and HAL



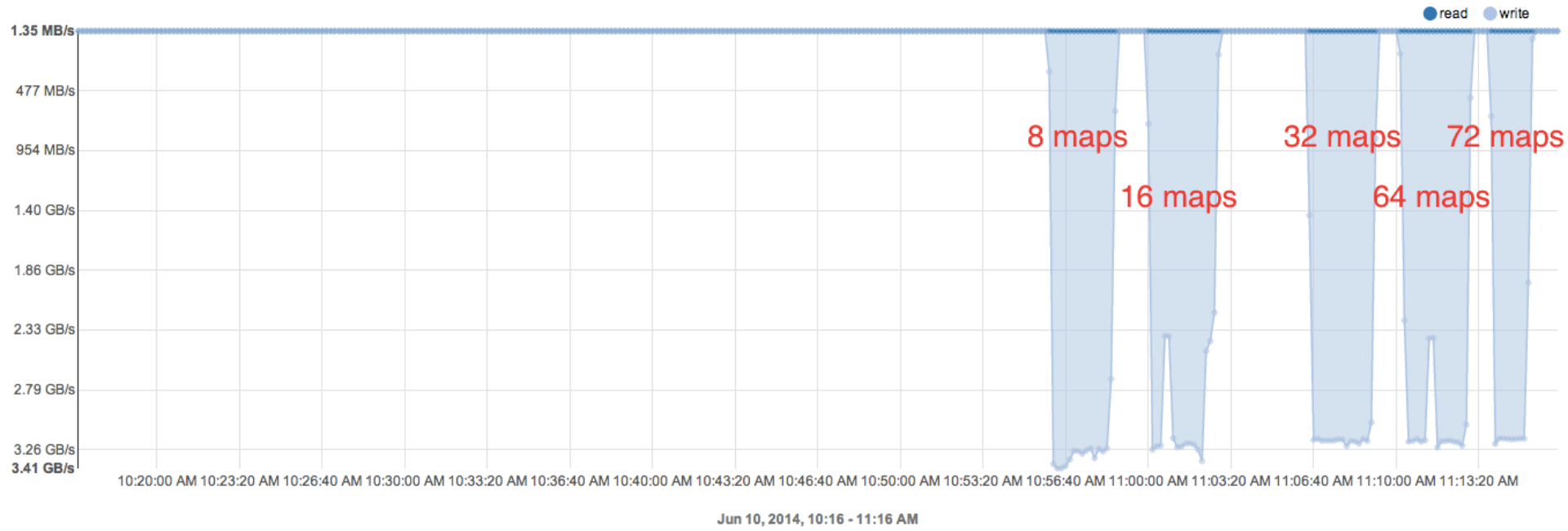
# Pure Hadoop's fs benchmark : DFSIO

- In the chart a screenshot from IML of the aggregate throughput of Lustre during the DFSIO benchmark
- We used 72 map tasks (8 maps each compute nodes) and 10GB data each map task
- We achieved a peak performance of 3.28GB/sec writing and 5.42GB/sec reading



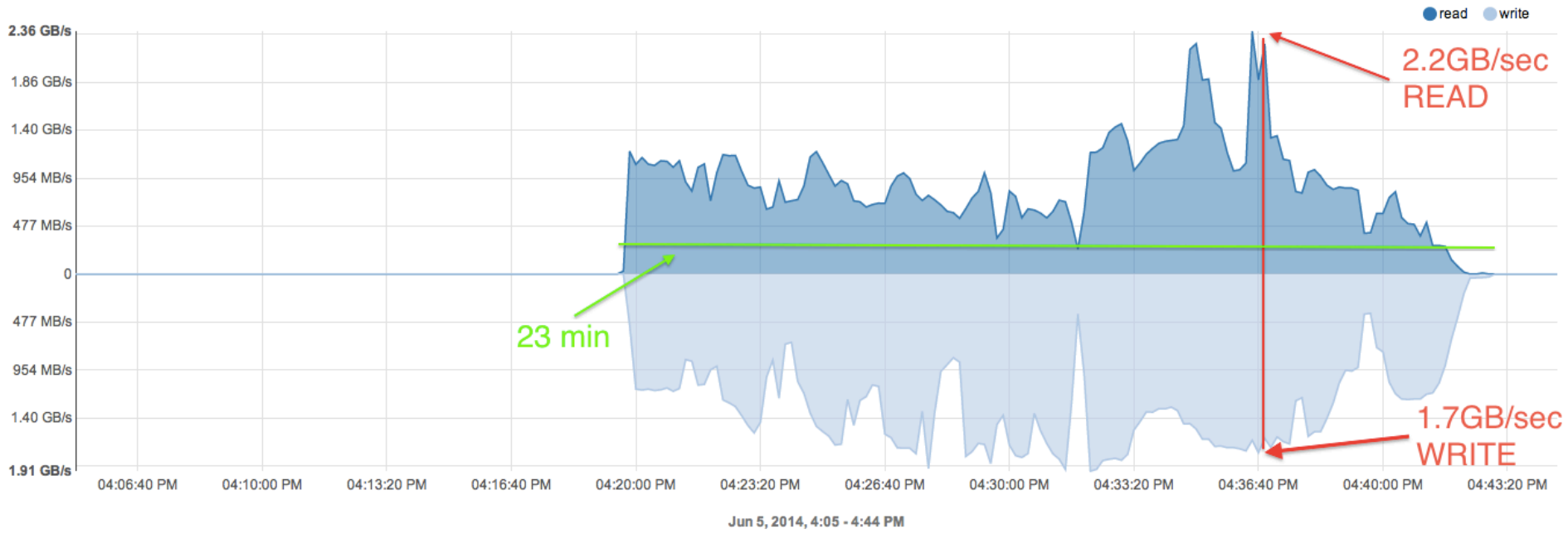
# Fill the I/O throughput with few clients

- Hadoop clusters are not so large compared with traditional HPC platforms
- Intel Enterprise Edition for Lustre 2.0 includes the single thread I/O performance improvement patch
- Single thread workloads are improved, but also the general single client scalability.
- In this experiment we can achieve the full throughput (3.41GB/sec) of our cluster using DFSIO and only 8 maps tasks.



# Benchmark results from Terasort

- In the chart a screenshot from IML of the aggregate throughput of Lustre during the Terasort benchmark
- We used 72 map tasks,144 reduce tasks and 500GB data size. **No Lustre or Hadoop optimizations.**
- The benchmark took 23 minutes to complete.
- The workload is challenging Lustre (50% READ and 50% WRITE)
- We achieved a peak performance of 3.9GB/sec (R: 2.2GB/sec W: 1.7GB/sec)



# Local metadata caching benefit in IEEL 2.0

- I/O patterns for Hadoop applications are very different from traditional HPC – often MANY metadata operations
- Metadata servers can become overloaded when run Hadoop jobs at scale

- New extended attributes caching improves metadata server performance

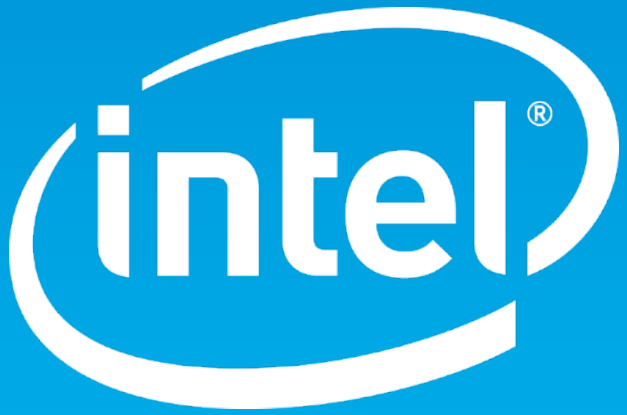
- During the previous terasort benchmark we collected the llite.*.stats from a client and the metadata activity received from the metadata server (mdt.*.md_stat), we can notice the huge benefit of the metadata local caching:

| | Single Client requests | 9x Clients metadata activity expected | Real metadata activity on the metadata server |
|---|---|---|---|
| open | 411954 | 3707586 | 4620535 |
| close | 411954 | 3707586 | 4620535 |
| getattr | 1851157 | 16660413 | 3239834 |
| getxattr | 52723133 | 474508197 | 1154874 |
| unlink | 128231 | 1154079 | 1154726 |

# Conclusion

- We achieved the goal to clean install Cloudera Distribution for Hadoop and HAL
- We proved the ability to use with Hadoop all the throughput available from Lustre
- The new version of Intel Enteprise Edition for Lustre is optimized for Hadoop
- SSD and Lustre can sustain the atypical Hadoop workload

*Acknowledgements*
*Servers and storage infrastructure, networking and hardware support was supplied by CSCS*


intel