HiMeter - Performance Analysis Framework for Big Data

Presenter: Zhang, Liye

Contributor: Hu, Jiayin

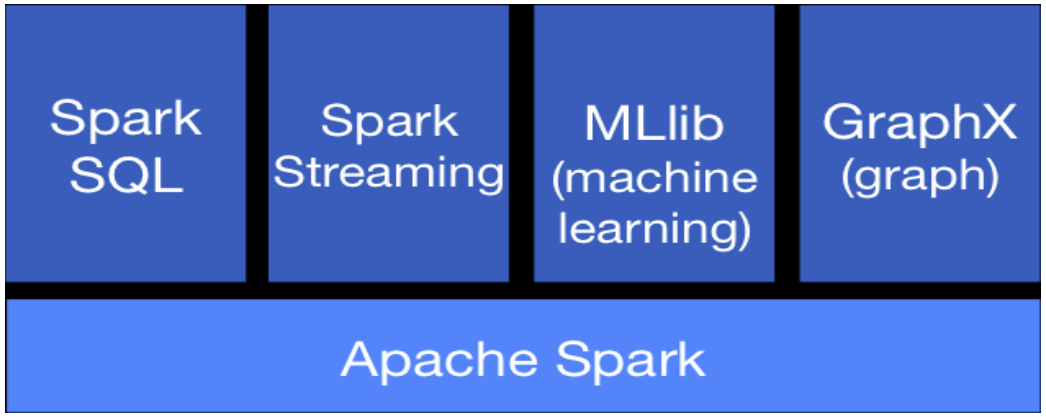
07/18/2015

Agenda

- BackGround
  - Apache Spark
  - Spark Configuration & Tuning
  - Spark WebUI
- What is HiMeter
- How to use HiMeter
- Case Study
- Conclusion

Apache Spark

- **Apache Spark™ Lightning-fast cluster computing (https://github.com/apache/spark)**
- **Significantly outperforms Hadoop MR**
  - **Iterative, Interactive, Incremental and In-memory computing**
  - **Up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.**
- **Generality usage**
  - **Rich modules**
  - **Easy integration**



**From https://spark.apache.org/*

Spark Configuration & Tuning

- **Official Document**

  http://spark.apache.org/docs/latest/

Spark Configuration

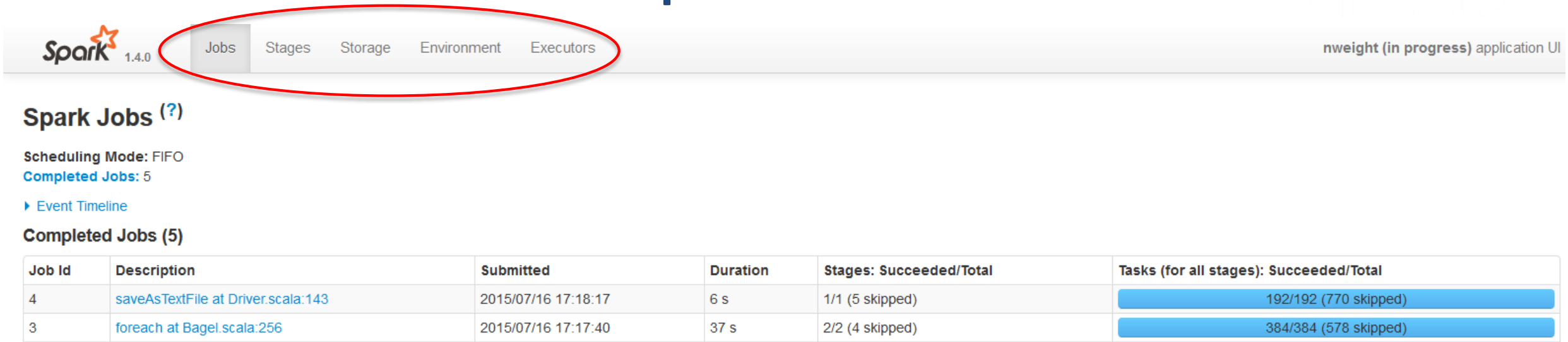
- Spark Properties
  - Dynamically Loading Spark Properties
  - Viewing Spark Properties
  - Available Properties
    - Application Properties
    - Runtime Environment
    - Shuffle Behavior
    - Spark UI
    - Compression and Serialization
    - Execution Behavior
    - Networking
    - Scheduling
    - Dynamic Allocation
    - Security
    - Encryption
    - Spark Streaming
    - SparkR
    - Cluster Managers
      - YARN
      - Mesos
      - Standalone Mode
- Environment Variables
- Configuring Logging
- Overriding configuration directory

Tuning Spark

- Data Serialization
- Memory Tuning
  - Determining Memory Consumption
  - Tuning Data Structures
  - Serialized RDD Storage
  - Garbage Collection Tuning
- Other Considerations
  - Level of Parallelism
  - Memory Usage of Reduce Tasks
  - Broadcasting Large Variables
  - Data Locality
- Summary

So many parameters, so many tuning aspects…

Spark WebUI



- Provide spark metrics
- Job,Stage,Task running time
- No system metrics
- Need other monitoring tools

Agenda

- BackGround
- What is HiMeter
  - Brief Introduction
  - Architecture
  - Work Flow
- How to use HiMeter
- Case Study
- Conclusion

Brief Introduction

- HiMeter

Realtime cluster monitoring

- Each node system metrics
- Whole cluster average status

A light-weight distributed performance analysis framework

- Distributed log collection and query
- Spark performance diagnosis
- Spark application management and report

A big data application management system

- Application registration
- Application execution
- Dew registered services monitor

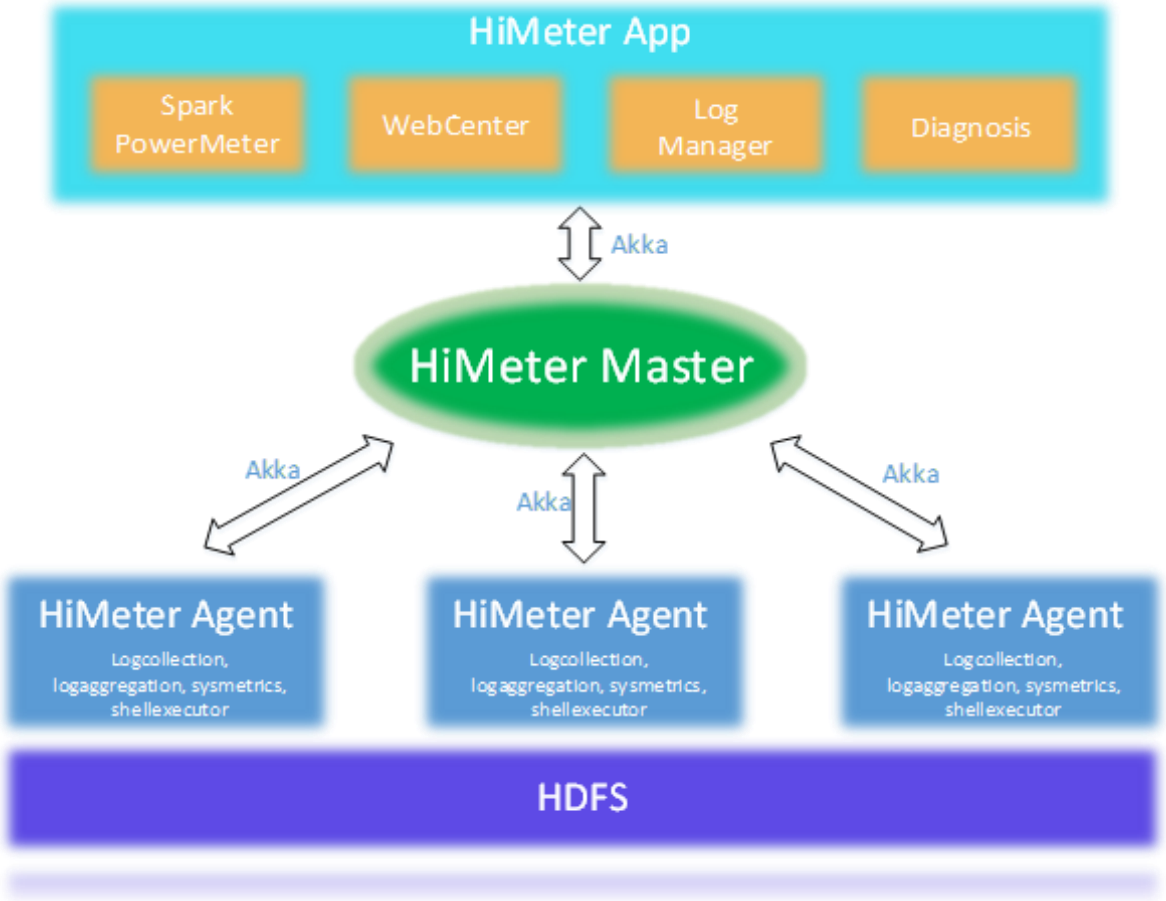
Platform Supported:

- Apache Hadoop (HDFS) 1.x & 2.x
- Spark 0.9+

Environment recommended:

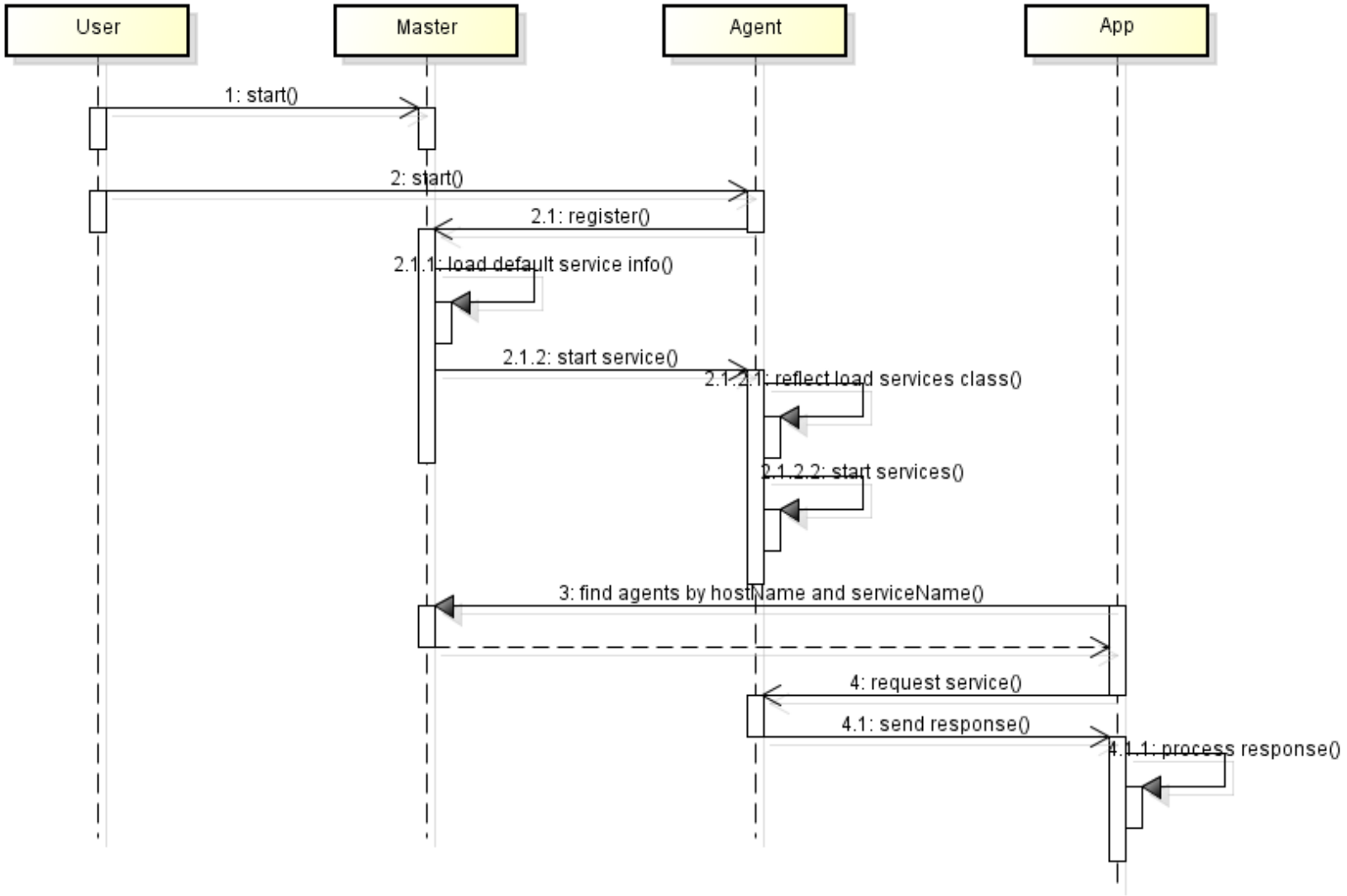
- JDK8 for compile
- dstat installed on all cluster nodes
- ssh passphraseless

Architecture



- Light-weight distributed
- Akka for communication
- HDFS for data storage
- Long run Agent
  - LogCollector
  - LogAggregation
  - SysMetrics
  - ShellExecutor

Work Flow



- Master, Agent, App are JVM processes.
- Services are either threads or processes.
- One Master in cluster
- At least one Agent on each node
- App can run on any node.

Agenda

- BackGround
- What is HiMeter
- How to use HiMeter
  - Quick start
  - Configuration
  - Web Components
- Case Study
- Conclusion

Quick Start

- Download the source code and build
  - mvn clean install -Dhadoop-version=your_deployed_hadoop_version -DskipTests
- Configurations
  - Edit conf/slaves, include all cluster nodes
  - Edit conf/dew.conf, set:
    - hdfs=hdfs://hostname:port (e.g. hdfs=hdfs://sr100:8020)
    - master:=hostname:port (e.g. master=sr100:6766)
- Deploy
  - Copy Dew dir to all cluster nodes
- Start/Stop Dew
  - sbin/start-all.sh & sbin/stop-all.sh

WebCenter

WebCenter — the Web UI for big data application management

$cd app.webCenter

- **Configuration**
  - Copy conf.properties.template to conf.properties
  - Change the configuration as you wish, also can keep the default
- **Start WebCenter**
  - ./start-web.sh
- **Log in WebCenter**
  - Web link: hostname:6077
  - User name: admin
  - Password: admin

SparkPowermeter

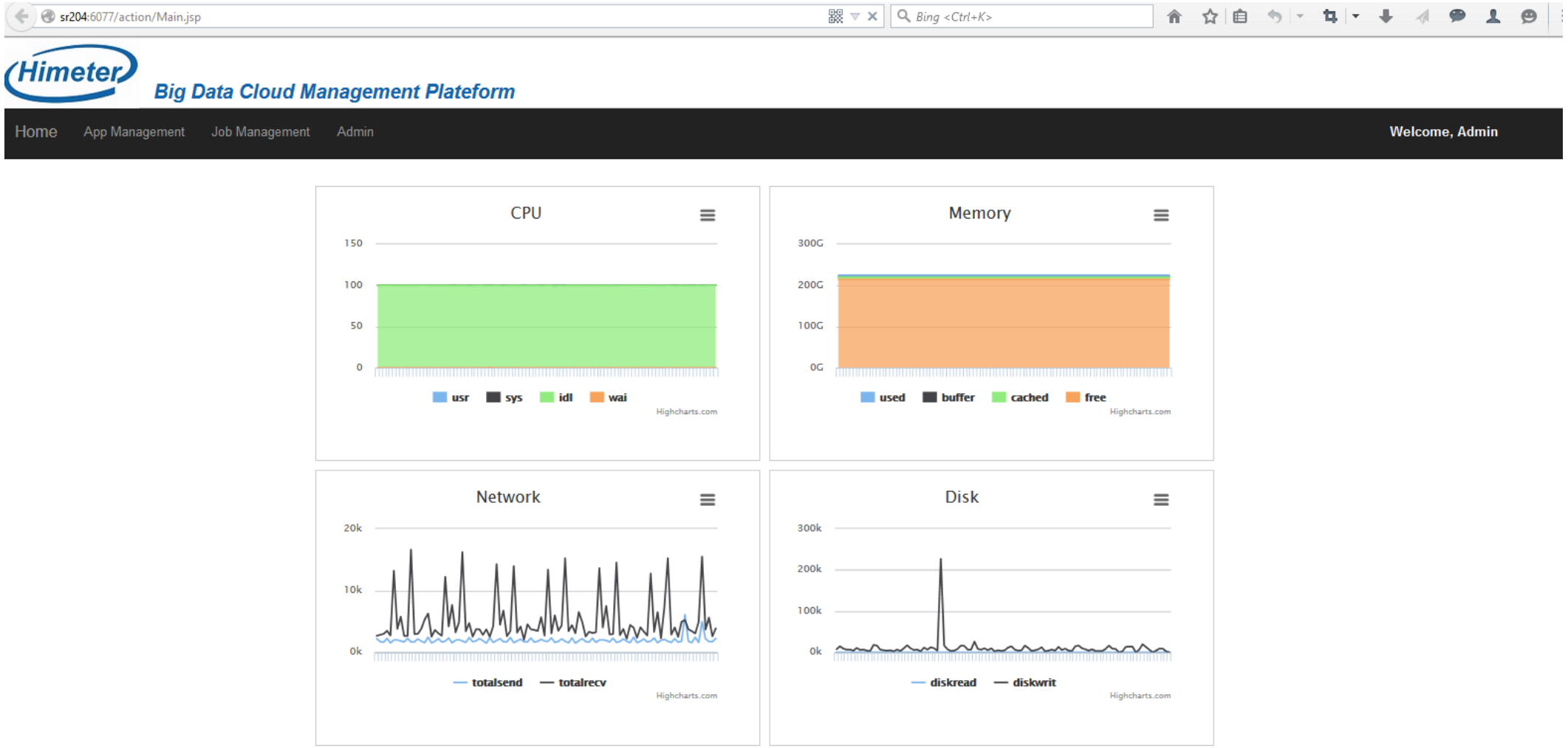
SparkPowermeter— A tool which analyze spark application performance base on spark data flow.

$cd app.sparkpowermeter

- Configuration
  - Copy conf.properties.template to conf.properties
  - Keep the configuration default or change it as you wish
- Run SparkPowerMeter (in two ways)
  - ./analyze.sh [spark driver log file path]
  - ./analyze.sh startTime(yy/MM/dd HH:mm:ss) endTime(yy/MM/dd HH:mm:ss)

Note: this functionality has been integrated into webCenter, you can either use webUI or command line to generate the system metrics report.

Cluster Status



Agents Status

Big Data Cloud Management Platform

Home | App Management | Job Management | Admin

Himeter Agents Status

IP	HostName	URL	Type	Services
10.1.0.104	sr204	akka.tcp://Agent@sr204:54996/user/dew/agent	branch	[shell, logcollection, dstatweb, dstat]
10.1.2.104	sr504	akka.tcp://Agent@sr504:43493/user/dew/agent	branch	[shell, logcollection, dstatweb, dstat]
10.1.2.104	sr504	akka.tcp://Agent@sr504:40902/user/dew/agent	leaf	[logaggregation]
10.1.2.105	sr505	akka.tcp://Agent@sr505:35233/user/dew/agent	branch	[shell, logcollection, dstatweb, dstat]
10.1.2.106	sr506	akka.tcp://Agent@sr506:48287/user/dew/agent	branch	[shell, logcollection, dstatweb, dstat]
10.1.2.106	sr506	akka.tcp://Agent@sr506:53945/user/dew/agent	leaf	[logaggregation]
10.1.2.107	sr507	akka.tcp://Agent@sr507:55037/user/dew/agent	branch	[shell, logcollection, dstatweb, dstat]

Application & Job Registration

Add New Application

Name
kmeans

Host
sr145

Path
/home/username/workload/kmeans

Executable
./run.sh

Strategy
reExecute

Type
spark

Submit

Add New Job

Name
daily

Defination
nweight,wordcount

Cycle
0 0 2

Submit

Crontab syntax (e.g. 0 0 2 * * ?), keep blank for a single run

Execution Result Report

Application Record List

AppName	StartTime	EndTime	Result	Operation
test1	3/5/15 12:56:00 PM.512	3/5/15 12:57:09 PM.565	success	Analysis LogQuery Diagnosis DriverLog
test1	3/4/15 12:56:00 PM.077	3/4/15 12:57:06 PM.458	success	Analysis LogQuery Di...osis DriverLog
test1	3/3/15 12:56:00 PM.122	3/3/15 12:57:06 PM.241	success	Analysis LogQuery Diagnosi... DriverLog

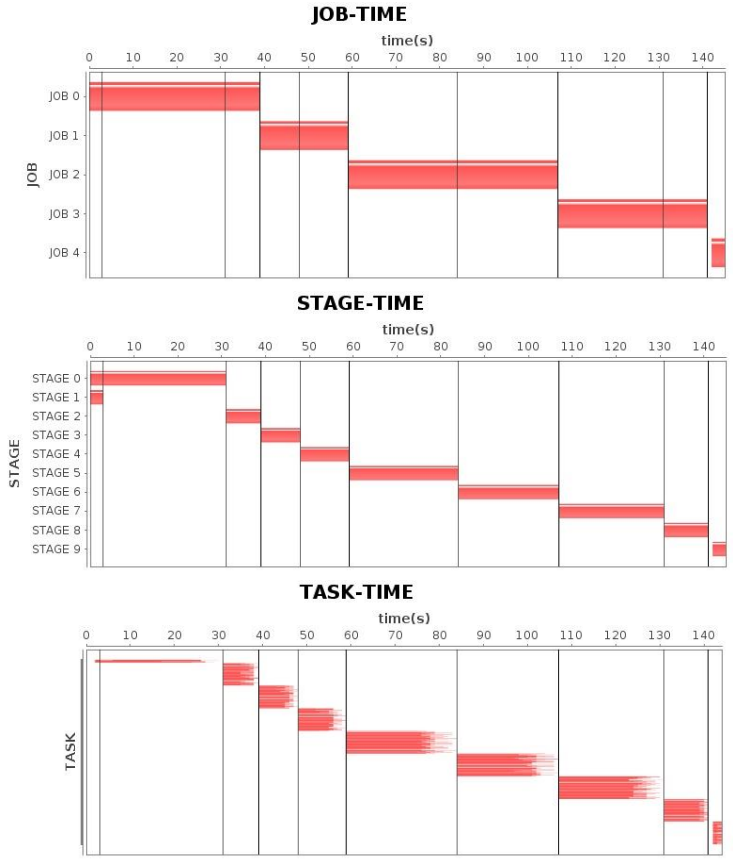
4 usefull links to analyze workload and cluster performance

Job Record List

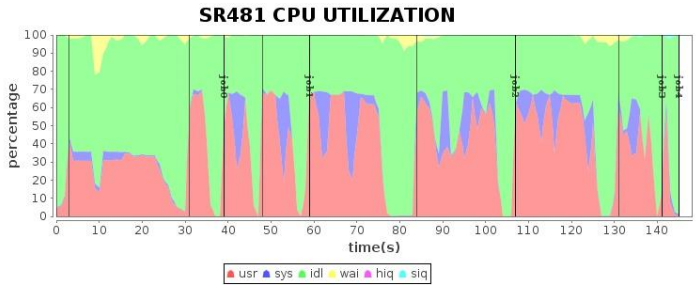
JobName	StartTime	EndTime	Result
app1	3/5/15 12:56:00 PM.004	3/5/15 12:56:00 PM.004	success
app1	3/4/15 12:56:00 PM.020	3/4/15 12:57:06 PM.458	success
app1	3/3/15 12:56:00 PM.042	3/3/15 12:57:06 PM.241	success
app1	2/11/15 11:14:18 AM.839	2/11/15 11:15:26 AM.452	success
app1	2/11/15 9:28:59 AM.513	2/11/15 9:30:10 AM.583	success
app1	2/6/15 3:06:55 PM.724	2/6/15 3:08:01 PM.985	failure

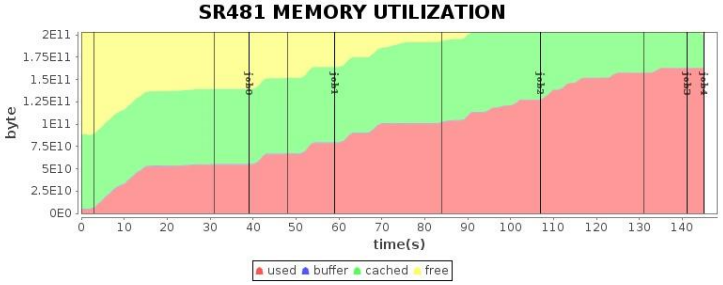
Analysis

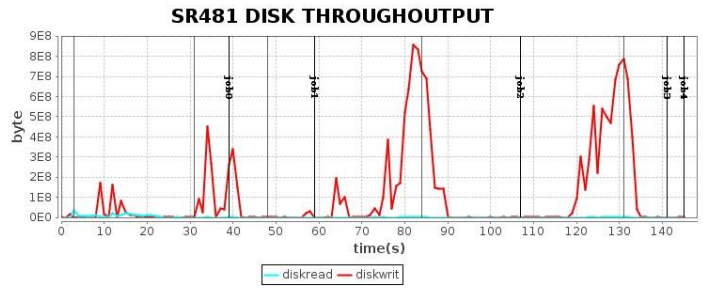
Spark work flow (Job, Stage, Task)

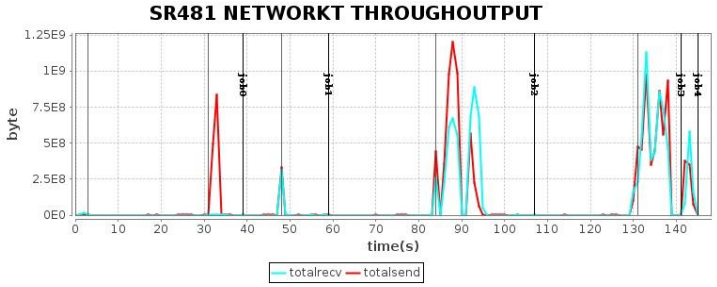


System metrics (CPU, Mem, Disk, Network)









Log query

All App List

New App

Search App

App Record

Search App Instance

WARN

Search

Query Result

driver.log 15/05/13 10:30:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
driver.log 15/05/13 10:30:56 WARN spark.SparkConf: In Spark 1.0 and later spark.local.dir will be overridden by the value set by the cluster manager (via SPARK_LOCAL_DIRS in mesos/standalone and LOCAL_DIRS in YARN).
driver.log 15/05/13 10:30:56 WARN spark.SparkConf:
driver.log 15/05/13 10:30:56 WARN spark.SparkConf: Setting 'spark.executor.extraJavaOptions' to '-Dspark.kryoserializer.buffer.mb=10 -XX:+UseParallelGC -XX:+UseParallelOldGC -XX:ParallelGCThreads=8 -XX:+UseTLAB -verbose:gc -XX:-PrintGCDetails -XX:+PrintGCTimeStamps -Dspark.storage.memoryFraction=0.6 ' as a work-around.
driver.log 15/05/13 10:30:56 WARN spark.SparkConf: Setting 'spark.driver.extraJavaOptions' to '-Dspark.kryoserializer.buffer.mb=10 -XX:+UseParallelGC -XX:+UseParallelOldGC -XX:ParallelGCThreads=8 -XX:+UseTLAB -verbose:gc -XX:-PrintGCDetails -XX:+PrintGCTimeStamps -Dspark.storage.memoryFraction=0.6 ' as a work-around.
driver.log 15/05/13 10:30:56 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Diagnosis

All App List

New App

Search App

App Record

Search App Instance

Show DiagnosisResult

	hostName	diagnosisName	level	describe	advice
	sr486	load-Disk-Read	high	load-Disk-Read is lower than cluster average by 56.53%	Check the node or your application algorism.
	sr485	load-Disk-Read	high	load-Disk-Read is lower than cluster average by 64.43%	Check the node or your application algorism.
	sr484	waste-CPU	middle	Cpu resources waste percent is 68.15%. More time on non-computation task.	Improve node's disk and network performance.
	sr483	waste-CPU	middle	Cpu resources waste percent is 66.58%. More time on non-computation task.	Improve node's disk and network performance.
	sr486	waste-CPU	middle	Cpu resources waste percent is 67.91%. More time on non-computation task.	Improve node's disk and network performance.
	sr485	waste-CPU	middle	Cpu resources waste percent is 69.75%. More time on non-computation task.	Improve node's disk and network performance.

Driver Log

All App List

New App

Search App

App Record

Search App Instance

Driver Log

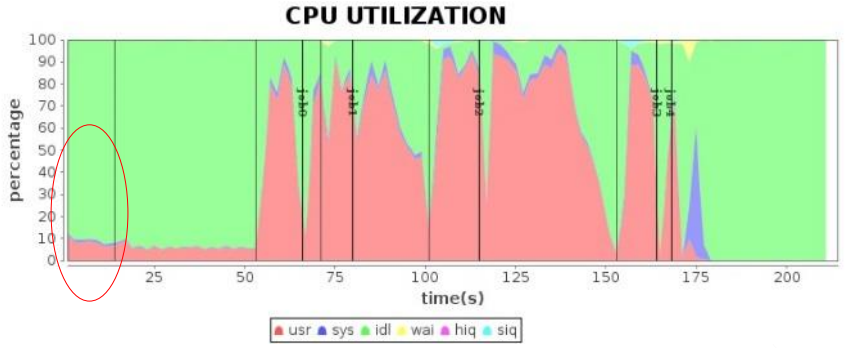
```
memory:
input: hdfs://sr409:8020/user/yucai/1ginput/yk_20131104
output: hdfs://sr409:8020/user/yucai/test.graph.output
degree: 3
maxOutEdges: 30
partitions: 160
storageLevel: 3
memFraction: 0.6
disableKryo: true
model: bagel
15/05/13 10:30:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
where applicable
15/05/13 10:30:54 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted hdfs://sr409:8020/user/yucai/test.graph.output
+ /home/yucai/work/spark/spark-1.3.0//bin/spark-submit --class com.intel.youku.graph.NWeight --name NWeight --master yarn-client
--num-executors 16 --executor-memory 45G --driver-memory 10G --executor-cores 10 --jars lib/fastutil-6.5.7.jar target/scala-2.10/graph-
n-degree-_2.10-1.0.jar hdfs://sr409:8020/user/yucai/1ginput/yk_20131104 hdfs://sr409:8020/user/yucai/test.graph.output 3 30 160 3 0.6
true bagel
tput: No value for $TERM and no -T specified
15/05/13 10:30:56 INFO spark.SparkContext: Running Spark version 1.3.0
```

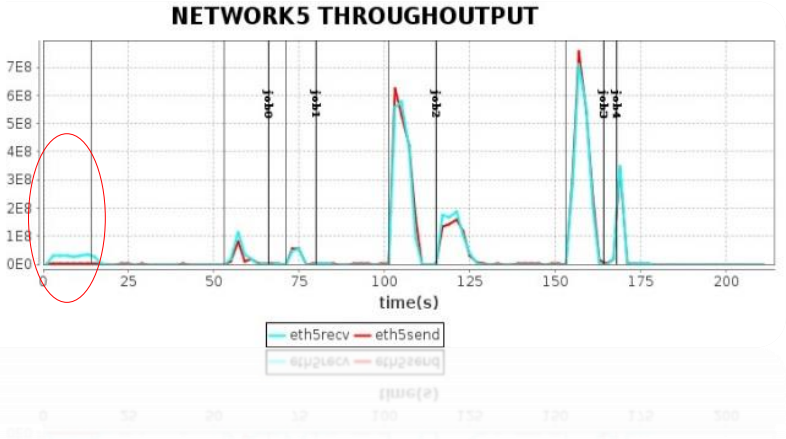
Agenda

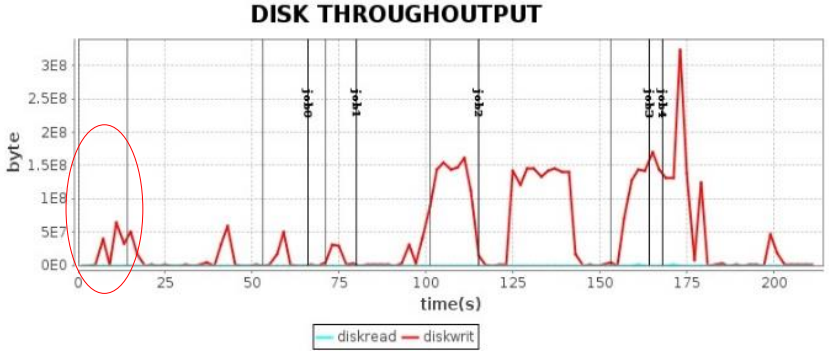
- BackGround
- What is HiMeter
- How to use HiMeter
- Case Study
- Conclusion

Case Study

- Only send jar file once for those co-located executors in Yarn SPARK-2713
- >10x speedup in bootstrap

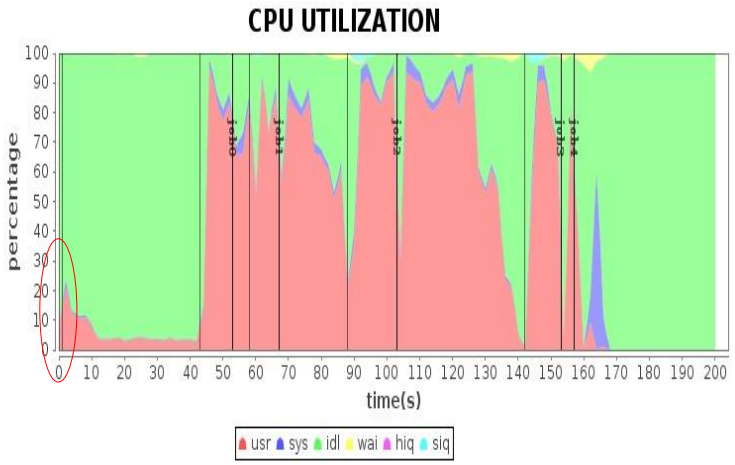


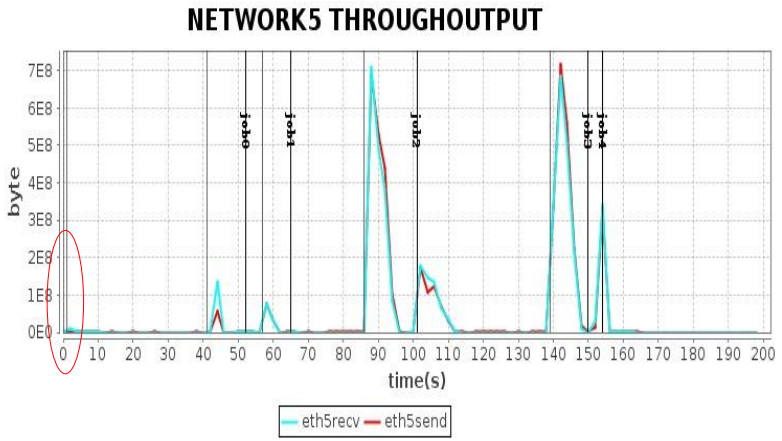


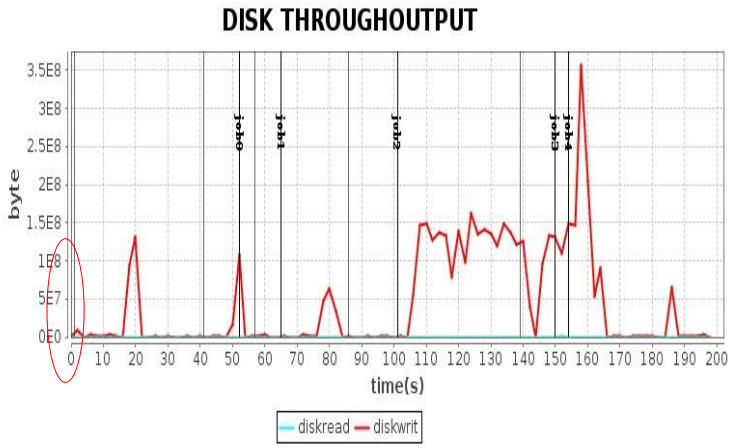


Case Study

- Each executor copies one job jar in Yarn
- Problem statement:
  - Co-located executors(containers) on the same NM have redundant copies
  - Leads to network/disk IO bandwidth consumption with big files
  - Causes long time dispatching period in bootstrap







Agenda

- BackGround
- What is HiMeter
- How to use HiMeter
- Conclusion
  - Advantages
  - TODOs

Conclusion

- Advantages
  - ✓ Friendly user interface
    - Easy to build, easy to use
    - Do anything with web console
  - ✓ Flexible architecture
    - Easy to build large scale distributed computation cluster
    - Easy to implement new distributed service and application
  - ✓ No couple but tightly integrate big data engine(Spark, Hadoop)
    - With plugin distributed service and application
- TODOs
  - Separate system metrics for multiple applications
  - High available when some servers or application crashes

Intern Hiring

Email to :

jie.huang@intel.com

Q & A

Thanks