

HiBench

the cross platforms micro-benchmark
suite for big data

Lv, Qi (qi.lv@intel.com)

July 22, 2015

About US

- Closely partnered with large web sites and ISVs on better user experiences
 - Key contributions for better customer adoption. E.g.,
 - Usability, Scalability and Performance
- More utilities to improve the stability & scalability
 - HiMeter: the light-weight workflow based big data performance analysis tool

Agenda

WHY

- Why we need big data benchmarking systems?

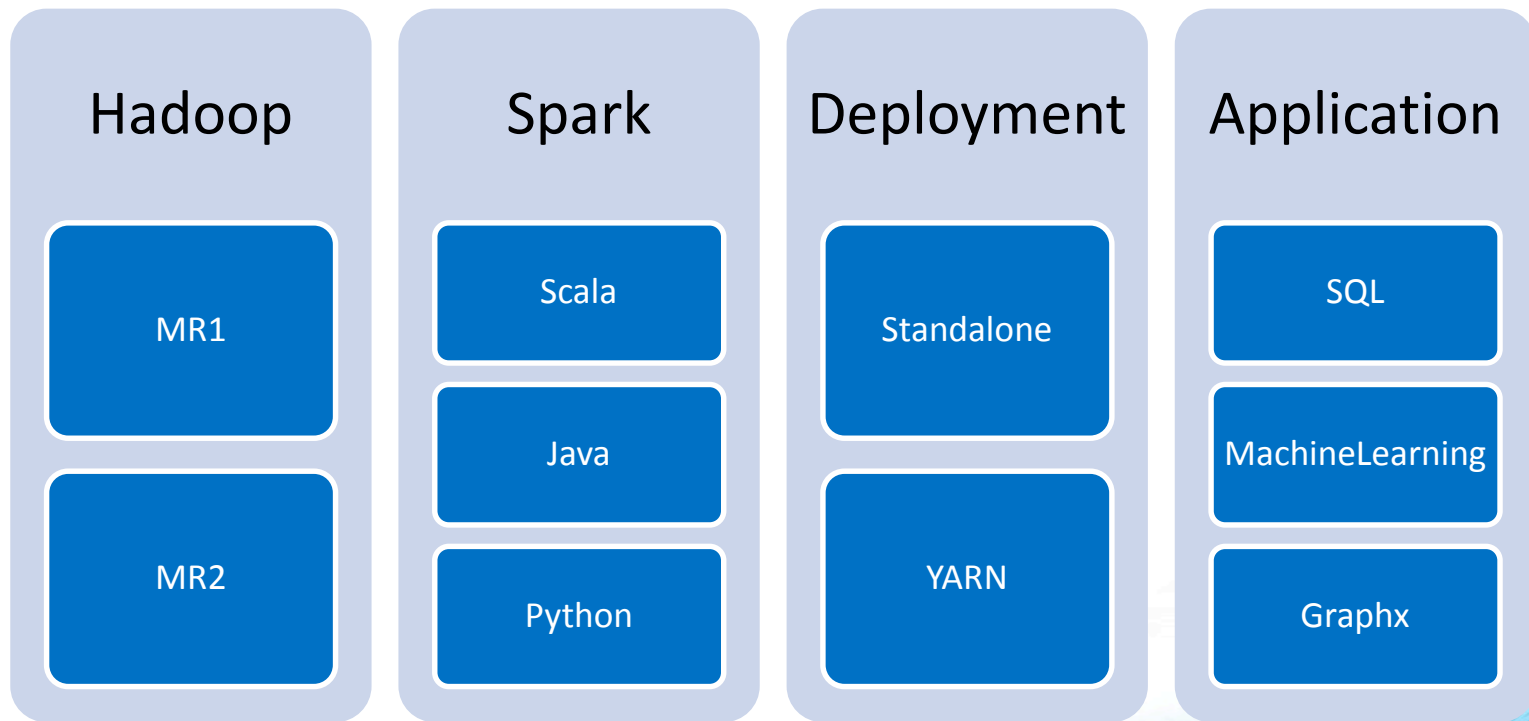
WHAT

- What is HiBench?

HOW

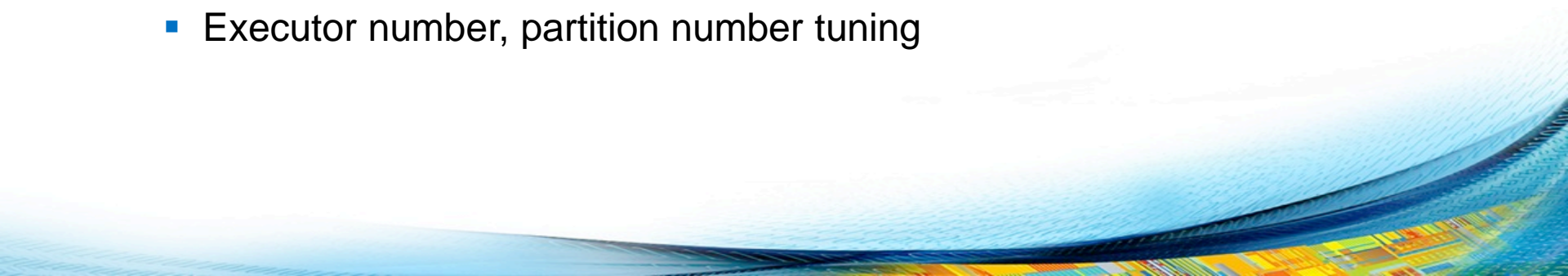
- How to use HiBench?

Big data ecosystem is complex



Frequent Questions from our Partners

- Which framework is better?
 - Hadoop MR1/MR2
 - Spark scala/java/python
 - Standalone/YARN
- How many resources needed?
 - CPU cores, memory, network bandwidth
- Is the cluster configured properly?
 - Executor number, partition number tuning



Meet HiBench

- Micro-bench oriented
 - Summarized from real application
 - Regression test
- Reputation
 - AMP lab
 - Yahoo
 - IBM
 - Pivotal




First Glance of HiBench



Core

- Sort
- wordcount
- terasort
- Sleep




MLLib

- KMeans
- Bayes




GraphX

- Pagerank



SQL

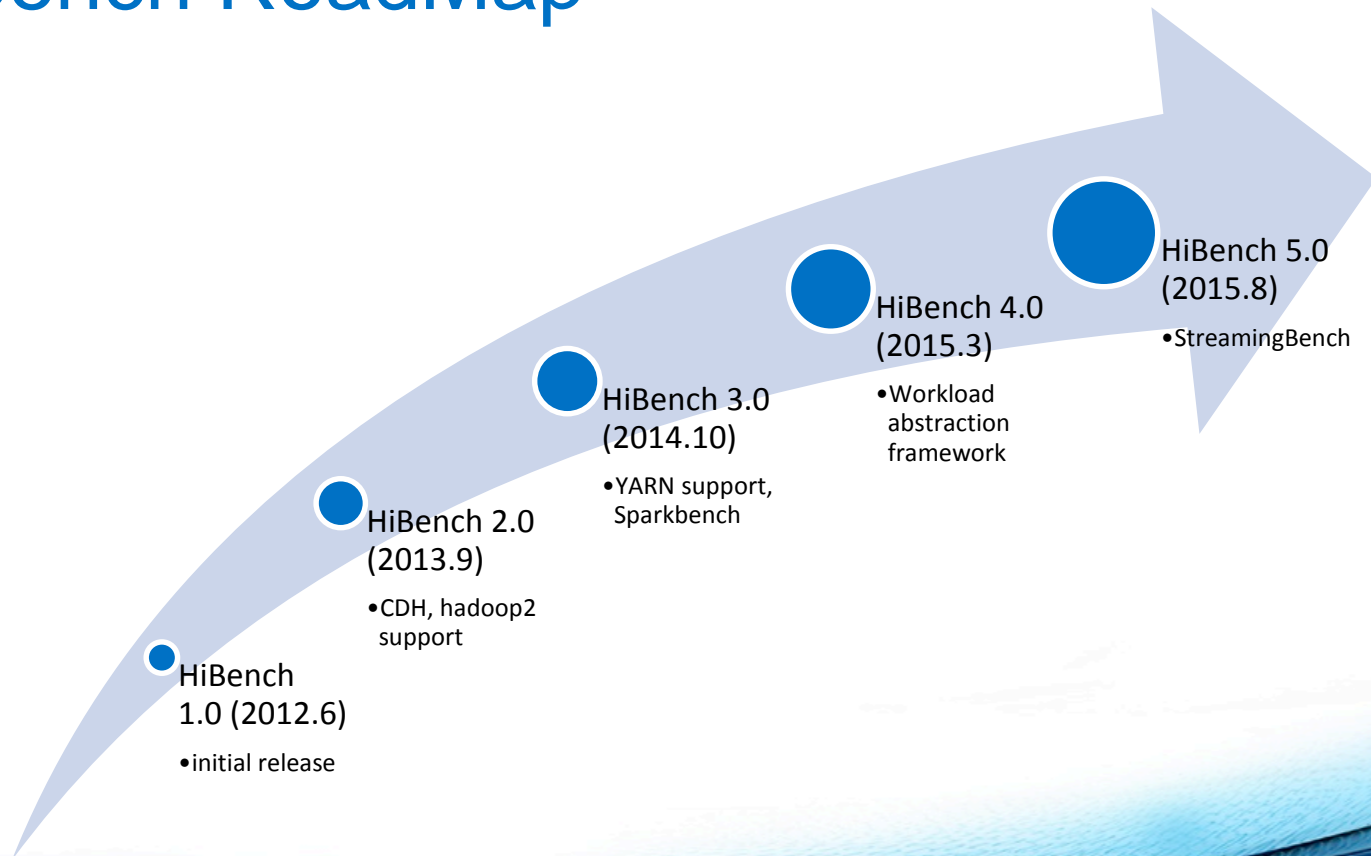
- Aggregation
- Join
- Scan



Streaming

- Identify
- grep
- wordcount
- project
- ::

HiBench RoadMap



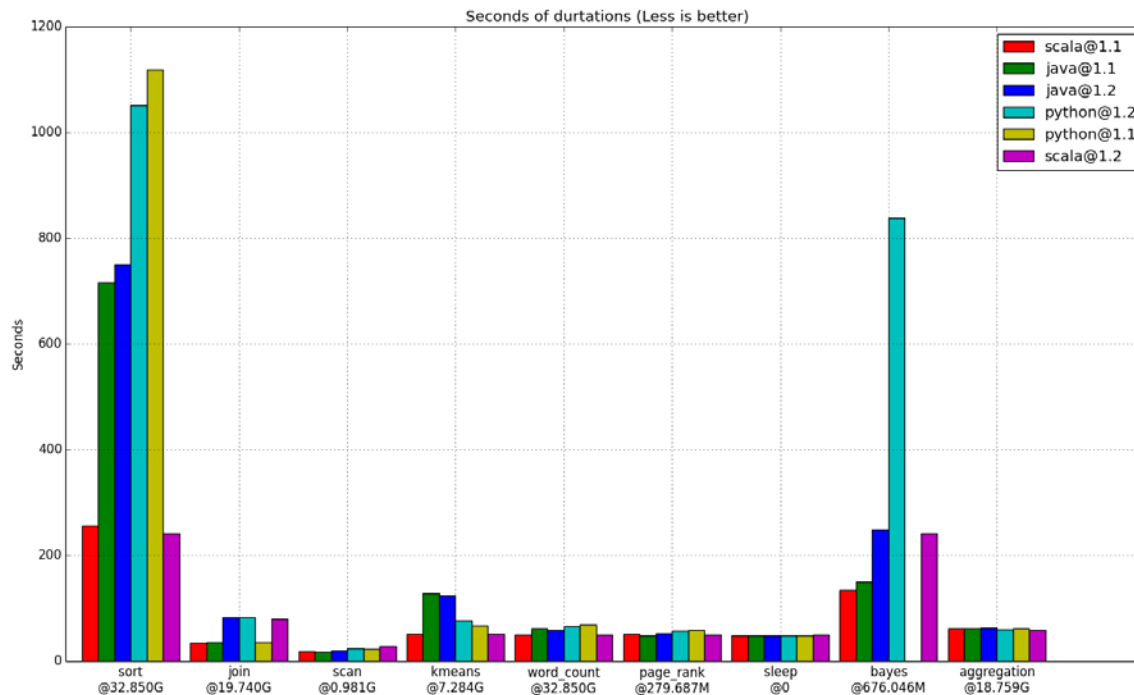
Key Features

- Workload abstraction
 - Typical workloads in classic application domains
 - Micro-bench workloads oriented
- Comparison between frameworks & configurations
 - MR1 / MR2, standalone / YARN
 - sequence / text, compression options / disable
- Scalable configuration
 - Global configuration for different scales
 - Dedicated configuration for individual workloads
- Metrics
 - Durations
 - Throughputs, Throughput per nodes

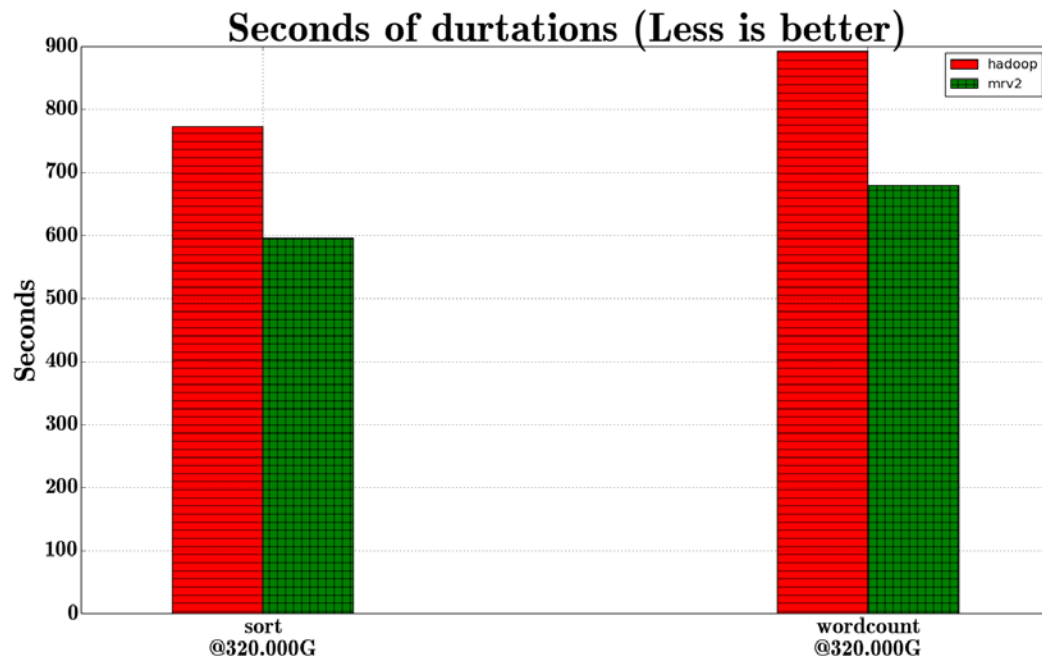
Showcasing how to explore the answer

- Cluster configuration
 - E5-2697 @ 2.7G 24C48T
 - Memory: 192 GB
 - Disks: 8 SSDs
 - Network: 10 GbE
 - Node size: 4
- Software stack
 - Spark: master (1.3.0-SNAPSHOT)
 - Hadoop1.0.4(MR1) / CDH5.3 (MR2)
 - JDK: oracle-1.8.0_25

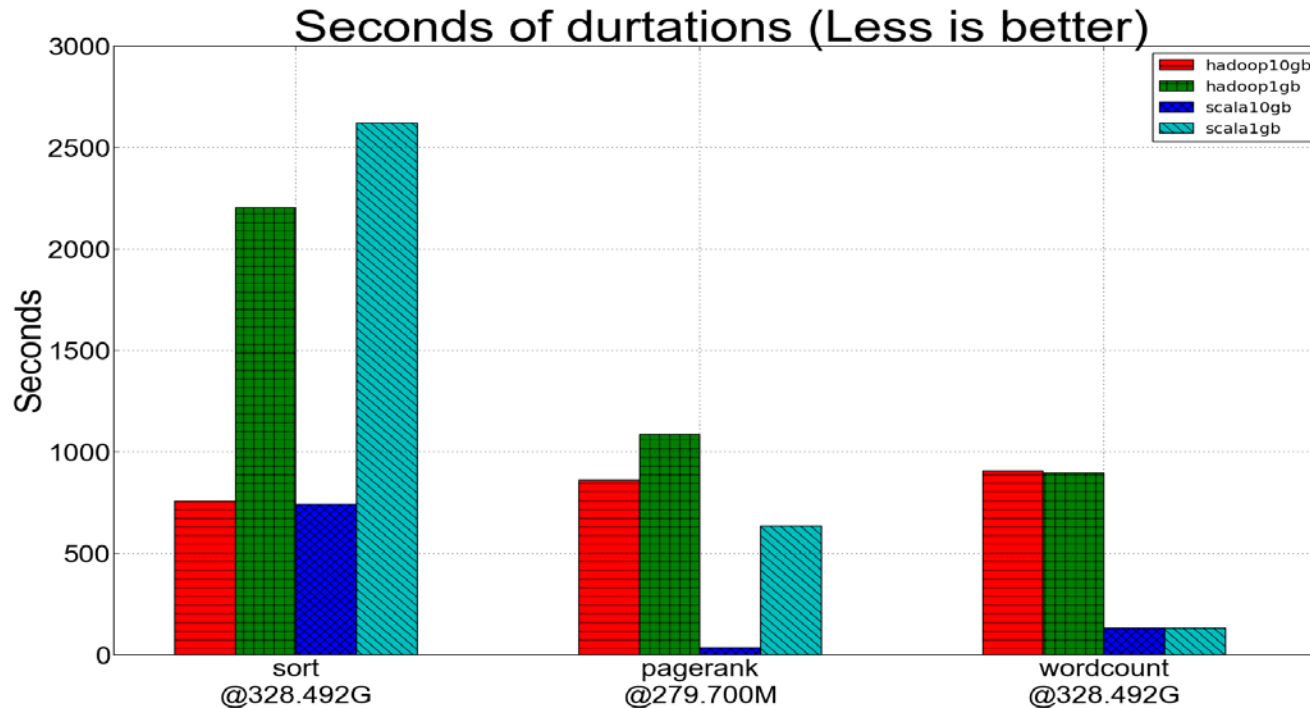
Comparison of language APIs (spark)



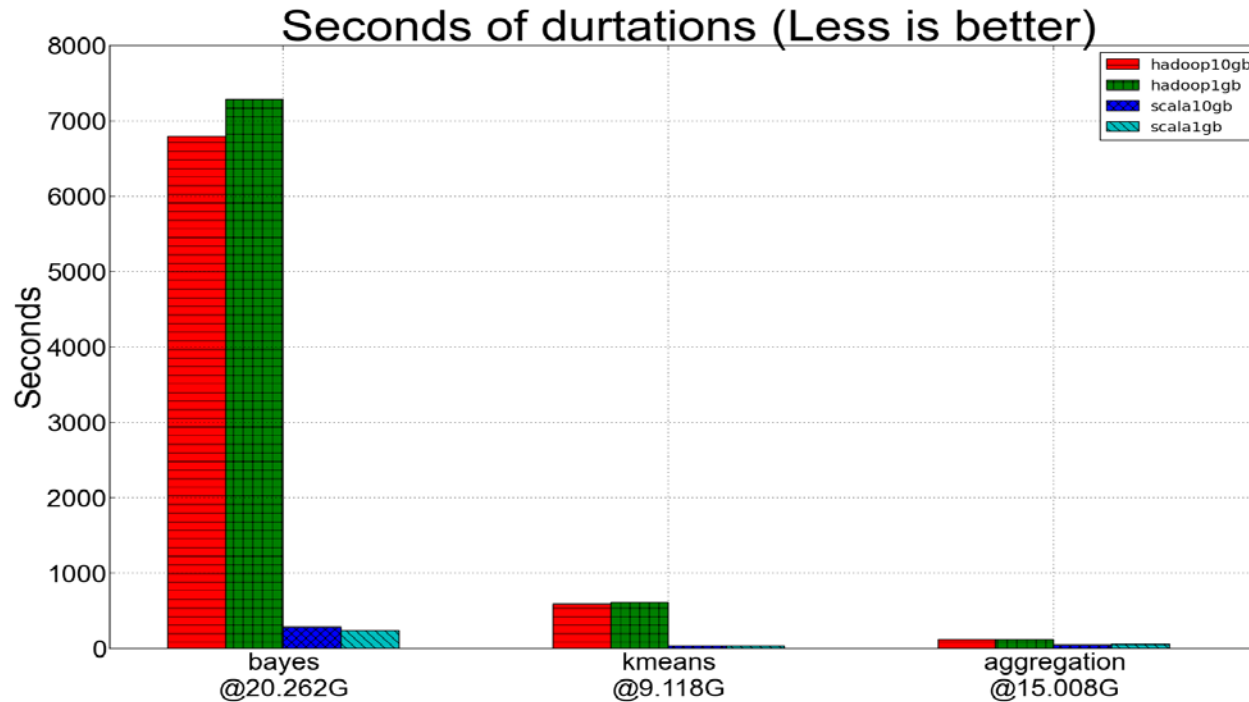
MR1 vs MR2(CDH5.3)



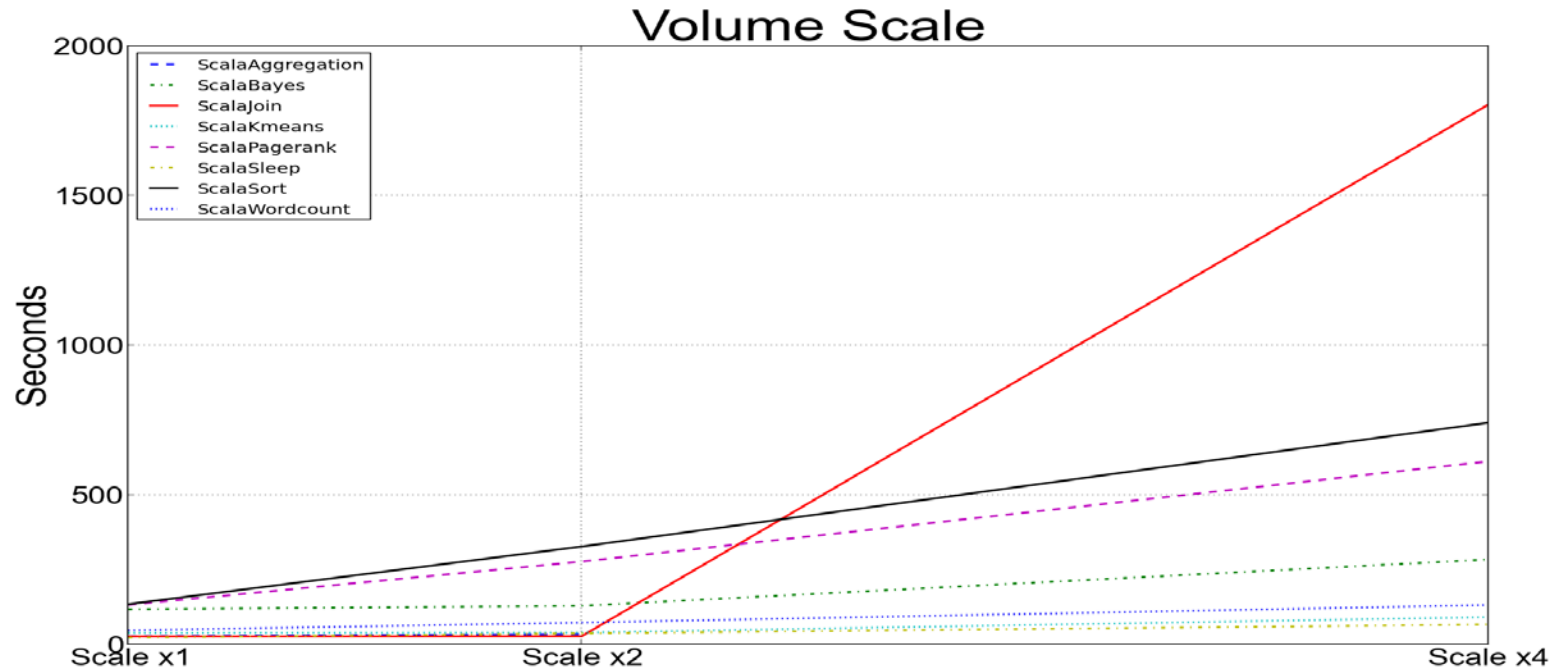
Impact of Network bandwidth



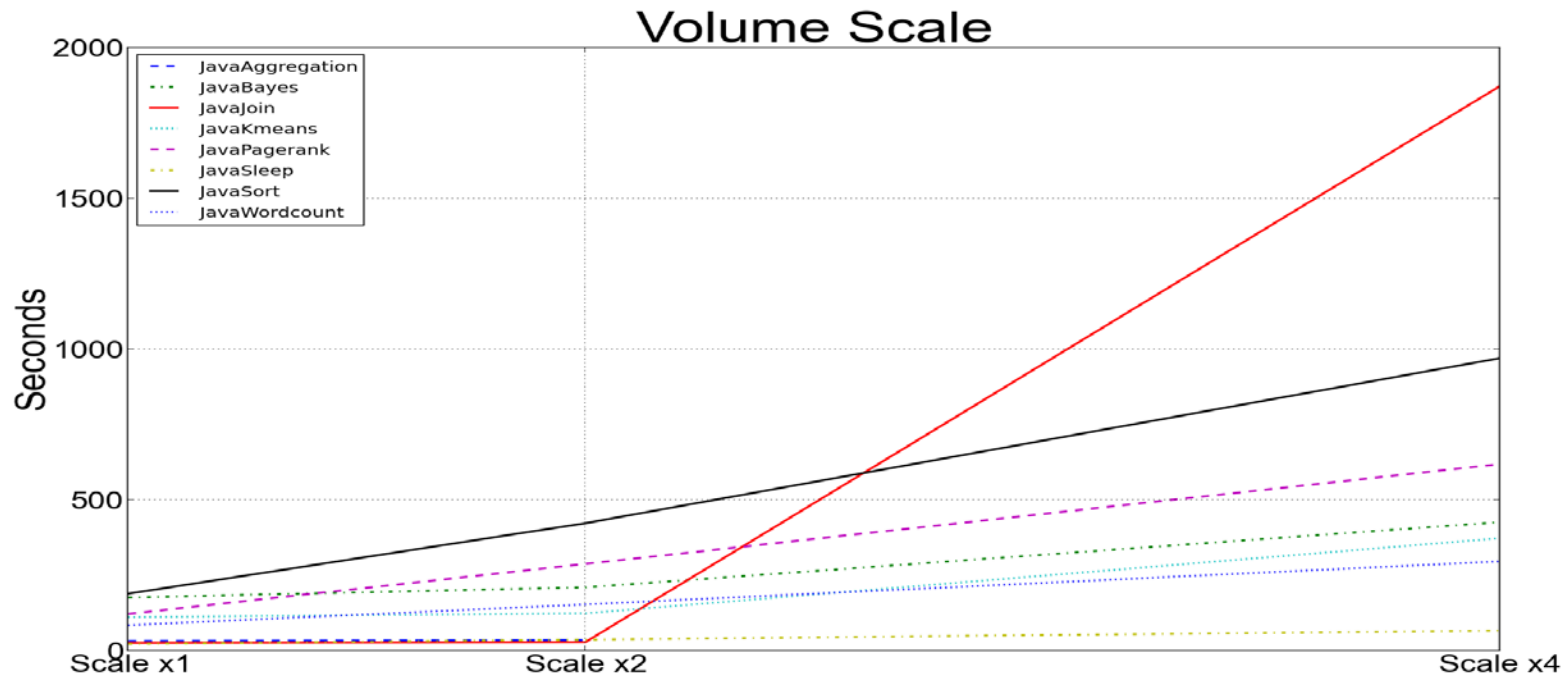
Impact of Network bandwidth



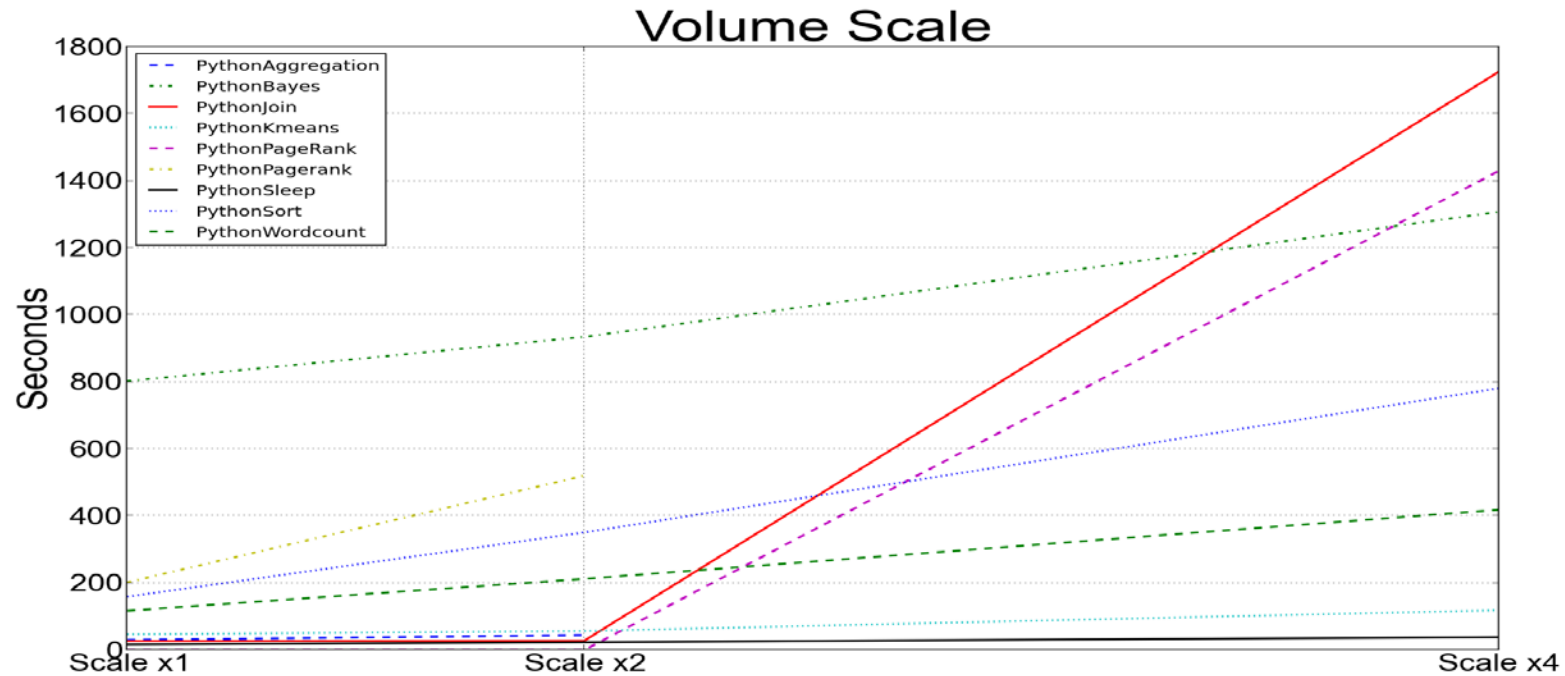
Data volume scalability Spark/scala



Data volume scalability Spark/java



Data volume scalability Spark/python



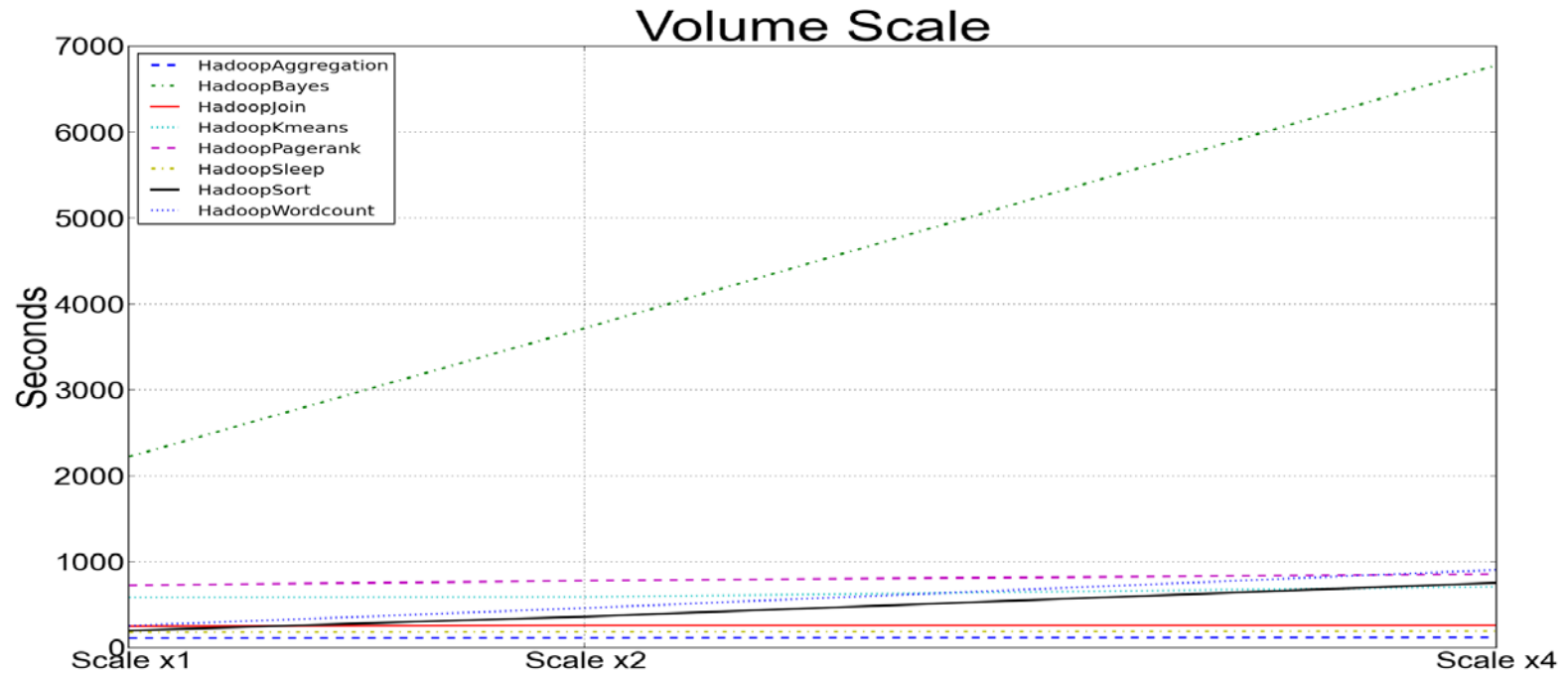
Q & A

Available at:
<https://github.com/intel-hadoop/HiBench>

Backup

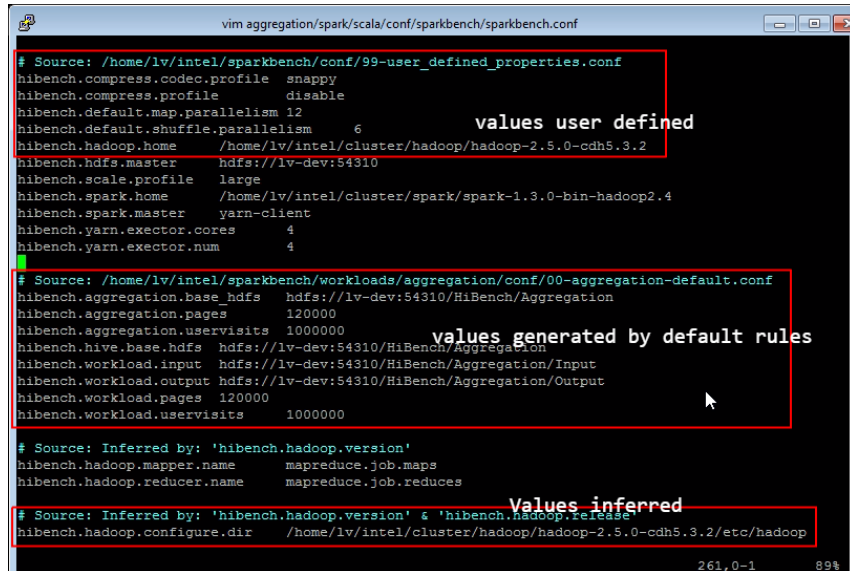


Data volume scalability – hadoop1



Report configuration example

- All configurations are classified accordingly
- Some configurations are auto probe & generated



```
vim aggregation/spark/scala/conf/sparkbench/sparkbench.conf

# Source: /home/lv/intel/sparkbench/conf/99-user_defined_properties.conf
hibench.compress.codec.profile snappy
hibench.compress.profile disable
hibench.default.map.parallelism 12
hibench.default.shuffle.parallelism 6
hibench.hadoop.home /home/lv/intel/cluster/hadoop/hadoop-2.5.0-cdh5.3.2
hibench.hdfs.master hdfs://lv-dev:54310
hibench.scale.profile large
hibench.spark.home /home/lv/intel/cluster/spark/spark-1.3.0-bin-hadoop2.4
hibench.spark.master yarn-client
hibench.yarn.executor.cores 4
hibench.yarn.executor.num 4

# Source: /home/lv/intel/sparkbench/workloads/aggregation/conf/00-aggregation-default.conf
hibench.aggregation.base_hdfs hdfs://lv-dev:54310/HiBench/Aggregation
hibench.aggregation.pages 120000
hibench.aggregation.uservisits 1000000
hibench.hive.base_hdfs hdfs://lv-dev:54310/HiBench/Aggregation
hibench.workload.input hdfs://lv-dev:54310/HiBench/Aggregation/Input
hibench.workload.output hdfs://lv-dev:54310/HiBench/Aggregation/Output
hibench.workload.pages 120000
hibench.workload.uservisits 1000000

# Source: Inferred by: 'hibench.hadoop.version'
hibench.hadoop.mapper.name mapreduce.job.maps
hibench.hadoop.reducer.name mapreduce.job.reducees

# Source: Inferred by: 'hibench.hadoop.version' & 'hibench.hadoop.release'
hibench.hadoop.configure.dir /home/lv/intel/cluster/hadoop/hadoop-2.5.0-cdh5.3.2/etc/hadoop
```

values user defined

values generated by default rules

Values inferred

261,0-1 89%

Troubleshooting

- Configuration issue
 - Check configuration parsing sequence to confirm your configuration is parsed properly

```
lv@lv-dev: ~sparkbench
$ mv 99-user_defined_properties.conf ..
lv at lv-dev in ~sparkbench/conf (v4.0-branch*)
$ cd ..
lv at lv-dev in ~sparkbench (v4.0-branch*)
$ ls
99-user_defined_properties.conf  bin  memo  README.md  src  workloads
Base_SCSReportforHiBench40.html  conf  memo-  report  WHATSNEW.md
lv at lv-dev in ~sparkbench (v4.0-branch*)
$ workloads/aggregation/prepare/prepare.sh
Parsing conf: /home/lv/intel/sparkbench/conf/00-default-properties.conf
Parsing conf: /home/lv/intel/sparkbench/conf/10-data-scale-profile.conf
Parsing conf: /home/lv/intel/sparkbench/workloads/aggregation/conf/00-aggregation-default.conf
Parsing conf: /home/lv/intel/sparkbench/workloads/aggregation/conf/10-aggregation-userdefine.conf
Traceback (most recent call last):
  File "/home/lv/intel/sparkbench/bin/functions/load-config.py", line 440, in <module>
    load_config(conf_root, load_conf, load_conf_order)
  File "/home/lv/intel/sparkbench/bin/functions/load-config.py", line 209, in load_config
    generate_optional_val(hibench.hadoop.home, hibench.spark.home, ...)
  File "/home/lv/intel/sparkbench/bin/functions/load-config.py", line 209, in generate_optional_val
    if hadoop_version[0] != '1': # hadoop2? or CDH's MR1?
IndexError: string index out of range
/home/lv/intel/sparkbench/bin/functions/workload-functions.sh: 第 33 行.: 需要文件名参数
.: 用法:. 文件名 [参数]
start HadoopPrepareAggregation bench
workloads/aggregation/prepare/prepare.sh:行25: INPUT_HDFS: 为绑定变量

lv at lv-dev in ~sparkbench (v4.0-branch*)
$
```

Forget to create 99-user_defined_properties.conf
Missing user defined properties like
hibench.hadoop.home, hibench.spark.home, ...

Troubleshooting(2)

- Pay attention to highlighted yellow and red message:
 - Yellow: warning
 - Red: Error
- If you doubt it's a configuration issue, please check report/<workload>/<language api>/conf/sparkbench/sparkbench.conf to double confirm that.

```
lv@lv-dev: ~-sparkbench
lv at lv-dev in ~-sparkbench (v4.0-branch)
$ workloads/aggregation/spark/scala/bin/run.sh
Parsing conf: /home/lv/intel/sparkbench/conf/00-default-properties.conf
Parsing conf: /home/lv/intel/sparkbench/conf/10-data-scale-profile.conf
Parsing conf: /home/lv/intel/sparkbench/conf/99-user_defined_properties.conf
Parsing conf: /home/lv/intel/sparkbench/workloads/aggregation/conf/00-aggregation-default.conf
Parsing conf: /home/lv/intel/sparkbench/workloads/aggregation/conf/10-aggregation-userdefine.conf
Parsing conf: /home/lv/intel/sparkbench/workloads/aggregation/spark/scala/scala.conf
start ScalaSparkAggregation bench
hdfs rm -r: /home/lv/intel/cluster/hadoop/hadoop-2.5.0-cdh5.3.2/bin/hadoop --config /home/lv/intel/cluster/hadoop/hadoop-2.5.0-cdh5.3.2/etc/hadoop fs -rm -r -skipTrash hdfs://lv-dev:54310/HiBench/Aggregation/Output
rm: `hdfs://lv-dev:54310/HiBench/Aggregation/Output': No such file or directory
Export env: SPARKBENCH_PROPERTIES_FILES=/home/lv/intel/sparkbench/report/aggregation/spark/scala/conf/sparkbench/sparkbench.conf
Submit Spark job: /home/lv/intel/cluster/spark/spark-1.3.0-bin-hadoop2.4/bin/spark-submit --properties-file /home/lv/intel/sparkbench/report/aggregation/spark/scala/conf/sparkbench/spark.conf --class com.intel.sparkbench.sql.ScalaSparkSQLBench --master yarn-client --num-executors 4 --executor-cores 4 --executor-memory 4G --driver-memory 4G /home/lv/intel/sparkbench/src/sparkbench/target/sparkbench-4.0-SNAPSHOT.jar
15/05/05 16:02:09 ERROR cluster.YarnScheduler: Lost executor 1 on lv-dev.sh.intel.com: remote Akka client disassociated
15/05/05 16:02:09 WARN remote.ReliableDeliverySupervisor: Association with remote system [akka.tcp://sparkExecutor@lv-dev.sh.intel.com:37644] has failed, address is now gated for [5000] ms. Reason is: [Disassociated].
15/05/05 16:02:09 ERROR cluster.YarnClientSchedulerBackend: Asked to remove non-existent executor
15/05/05 16:02:11 INFO metastore.MetaStoreDirectSql: MySQL check failed, assuming we are not on mysql. Lexical error at line 1, column 5: Encountered: "8" (64), after : ""
15/05/05 16:02:12 WARN metastore.ObjectStore: Version information not found in metastore. hive.metastore.schema.verification is not enabled so recording the schema version 0.13.1aa
15/05/05 16:02:15 INFO spark.SparkContext: Starting job: runJob at InsertIntoHiveTable.scala:83
```


System utilization chart

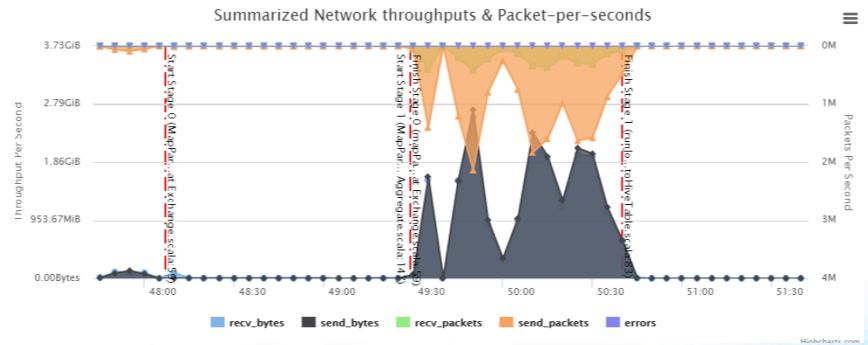
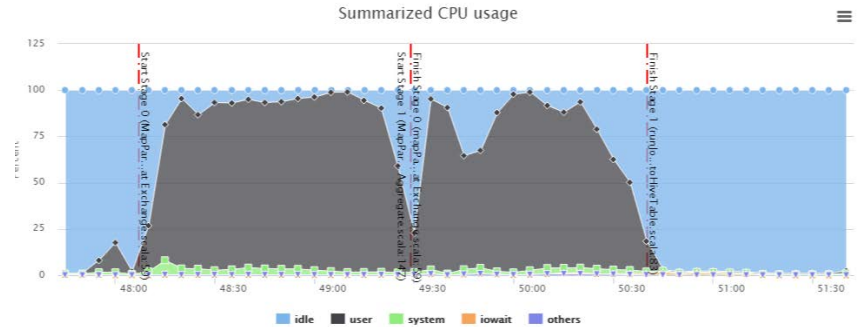
■ Chart

■ CPU chart

- Sys/User/IOWait/
- Others=nice+irq+softirq

■ Network chart

- Recv, send bytes
- Recv, send packets
- Errors=send_err+recv_err+send_drop+recv_drop



System utilization chart(2)

- Chart
 - Disk chart
 - Read, write bytes
 - Read, write IOPS
 - Memory chart
 - Used, buffer/cache, free
 - System load chart
 - Load5/10/15
 - Running processes
 - All process numbers(with threads)

