HiBench

the cross platforms micro-benchmark suite for big data

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

July 22, 2015

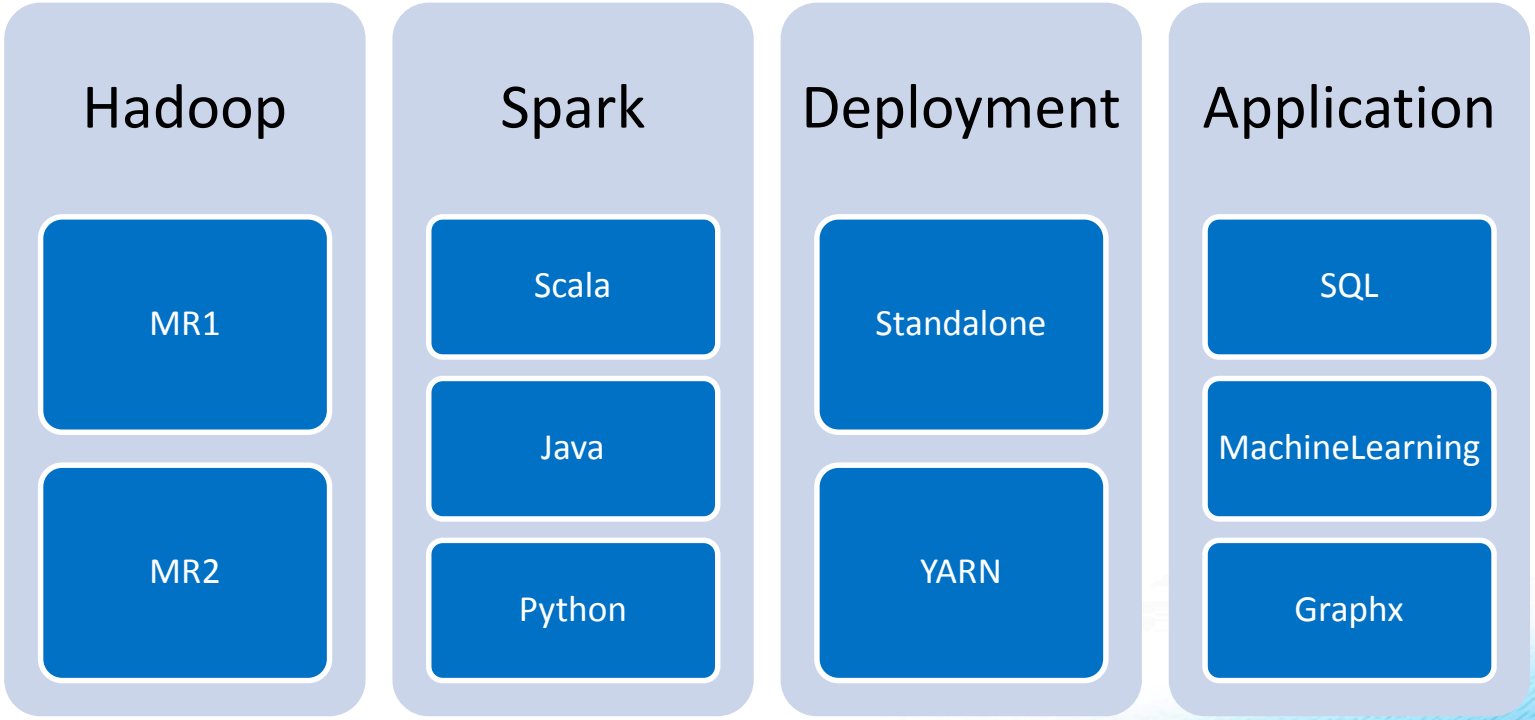
About US

- 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

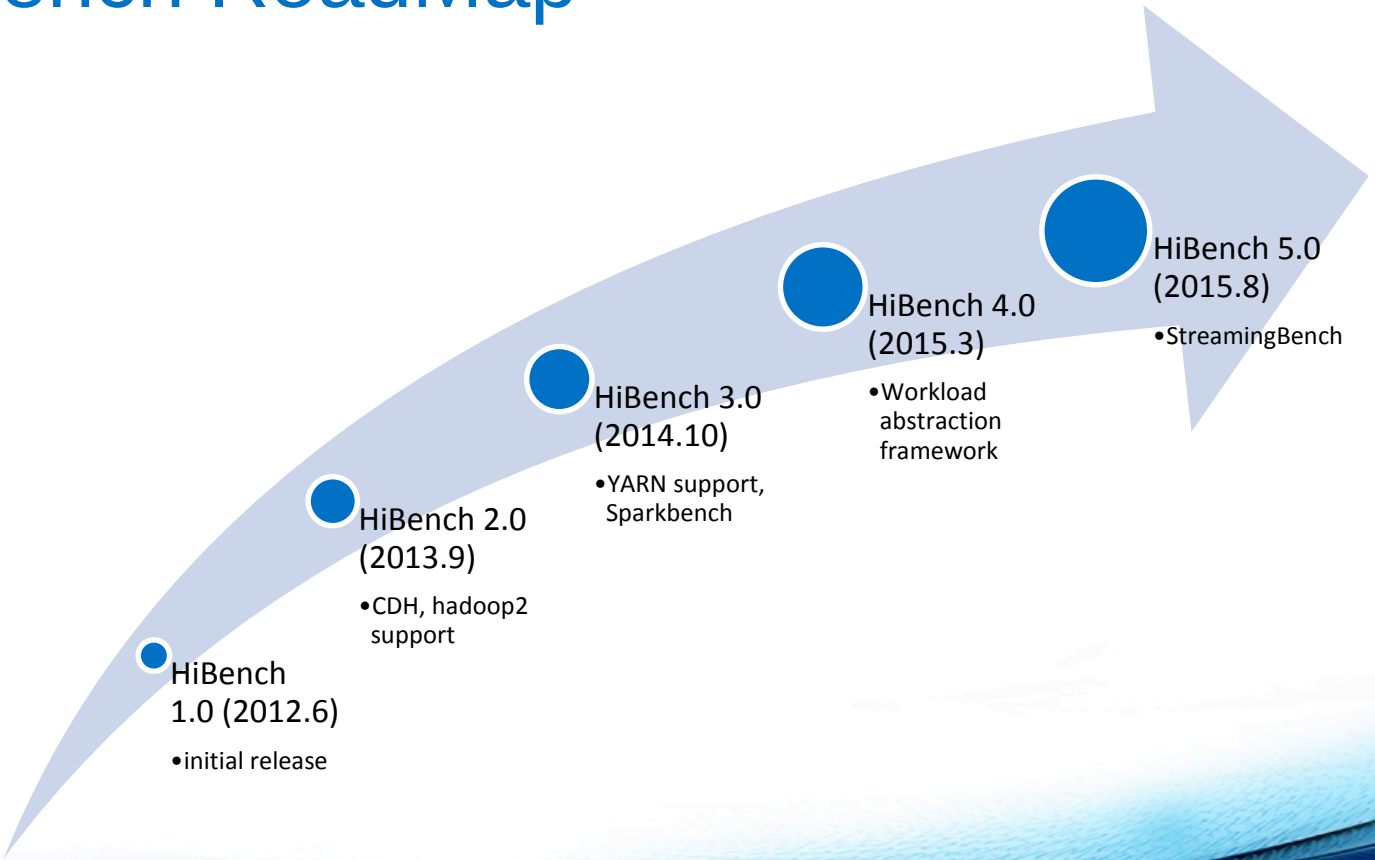
- Which framework is better?
  - Hadoop MR1/MR2
  - Spark scala/java/python
  - Standalone/YARN
- How many resources needed?
  - CPU cores, memory, network bandwidth
- Is the cluster configured properly?
  - Executor number, partition number tuning

Meet HiBench

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

First Glance of HiBench

HiBench RoadMap



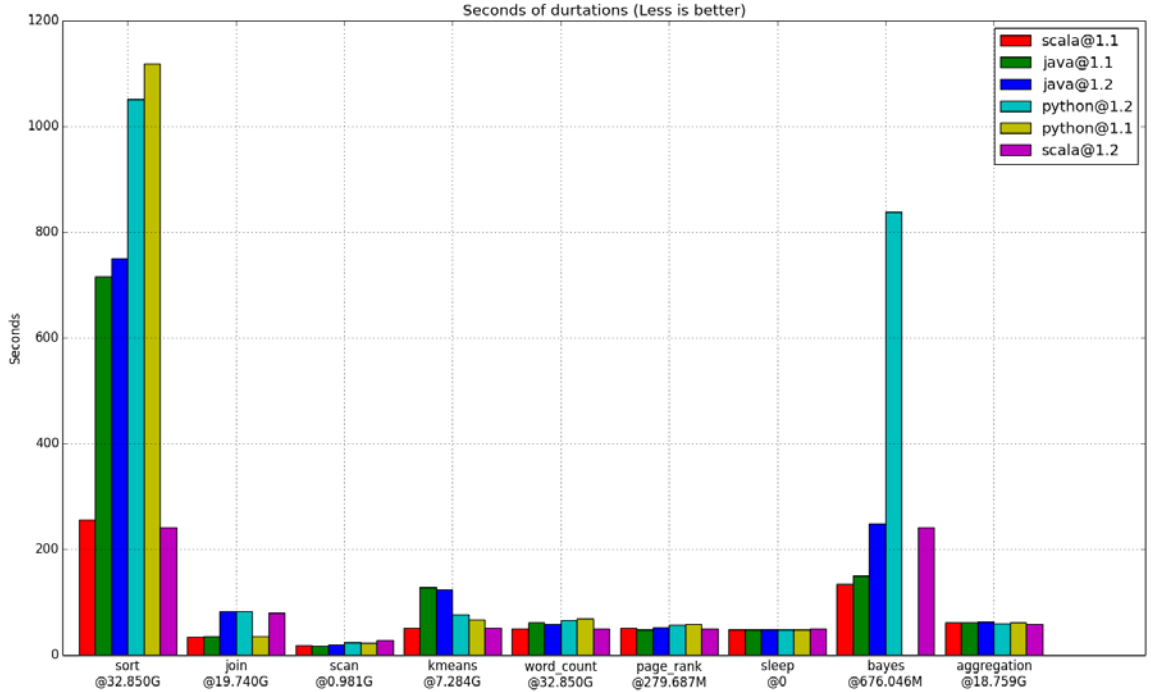
Key Features

- 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

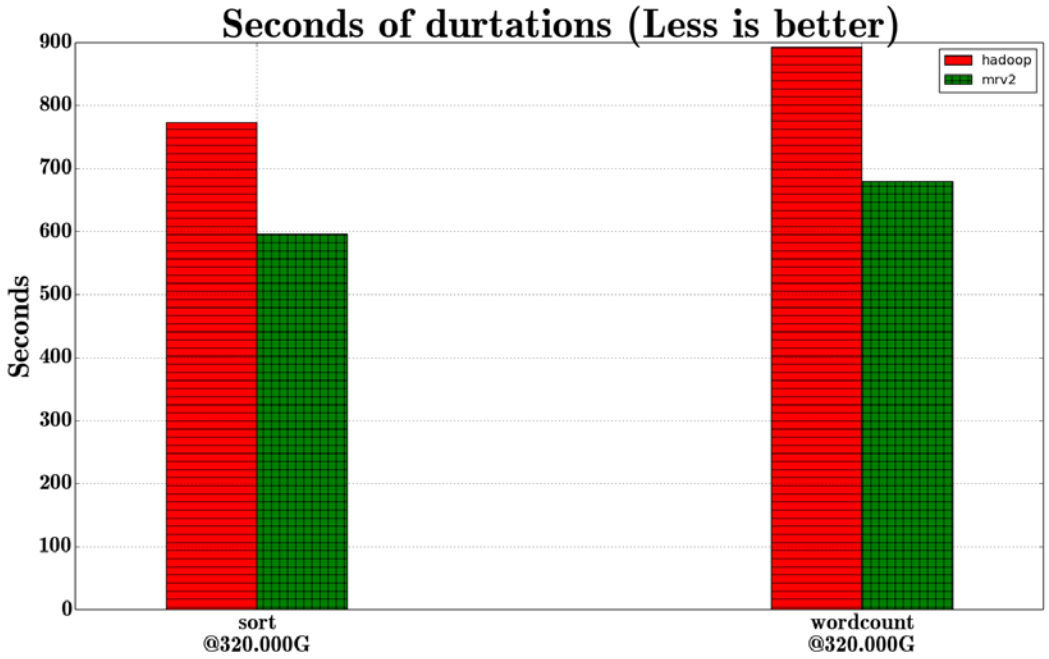
Showcasing how to explore the answer

- 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

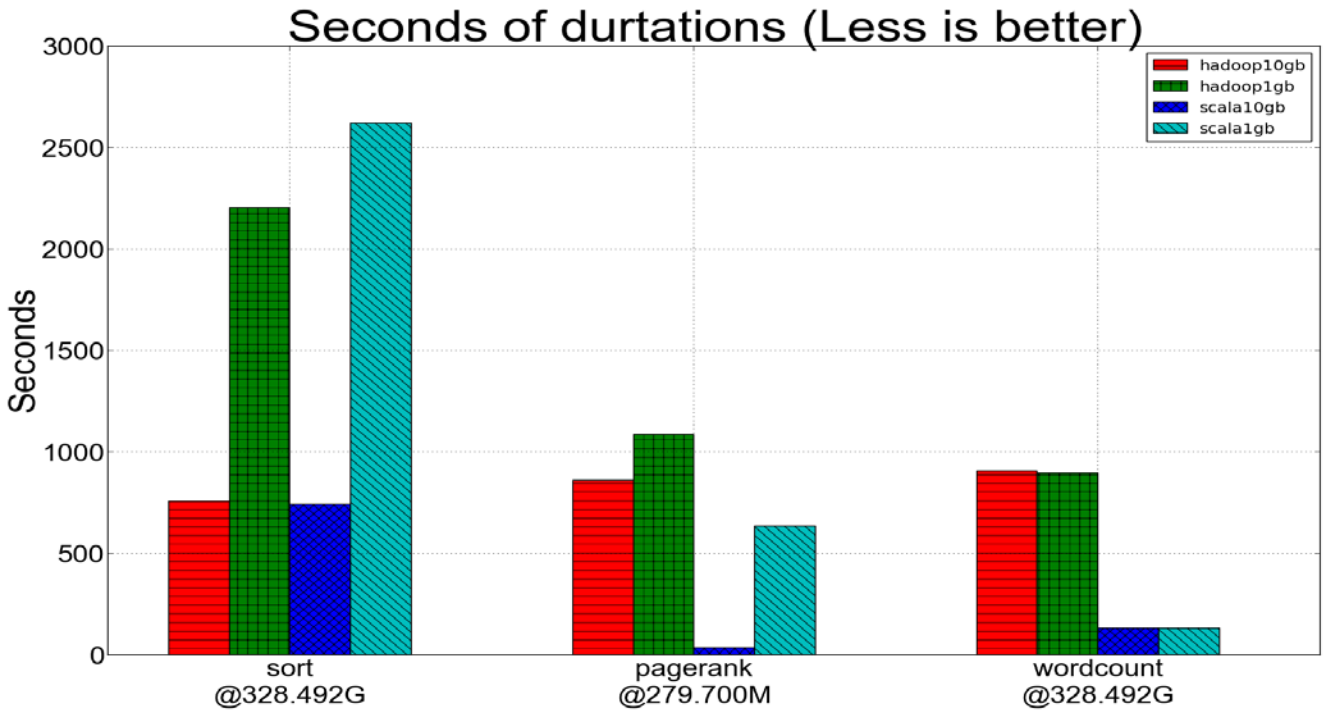
Comparison of language APIs (spark)



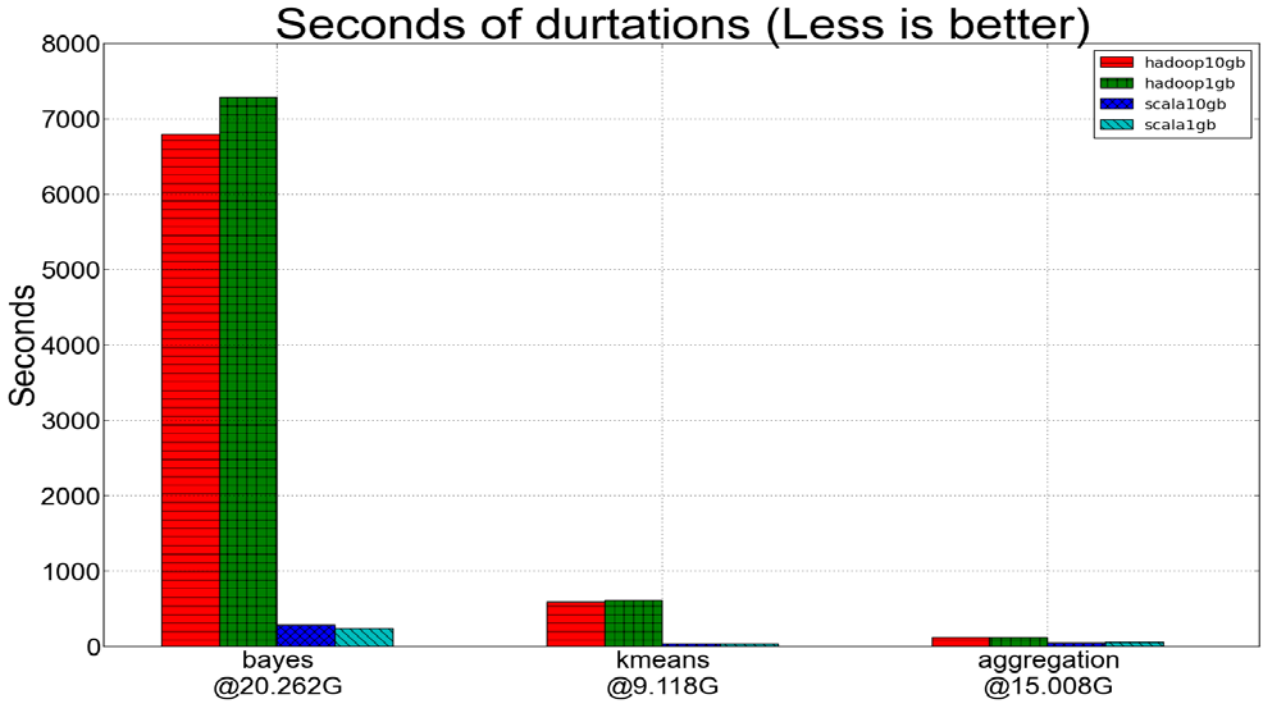
MR1 vs MR2(CDH5.3)



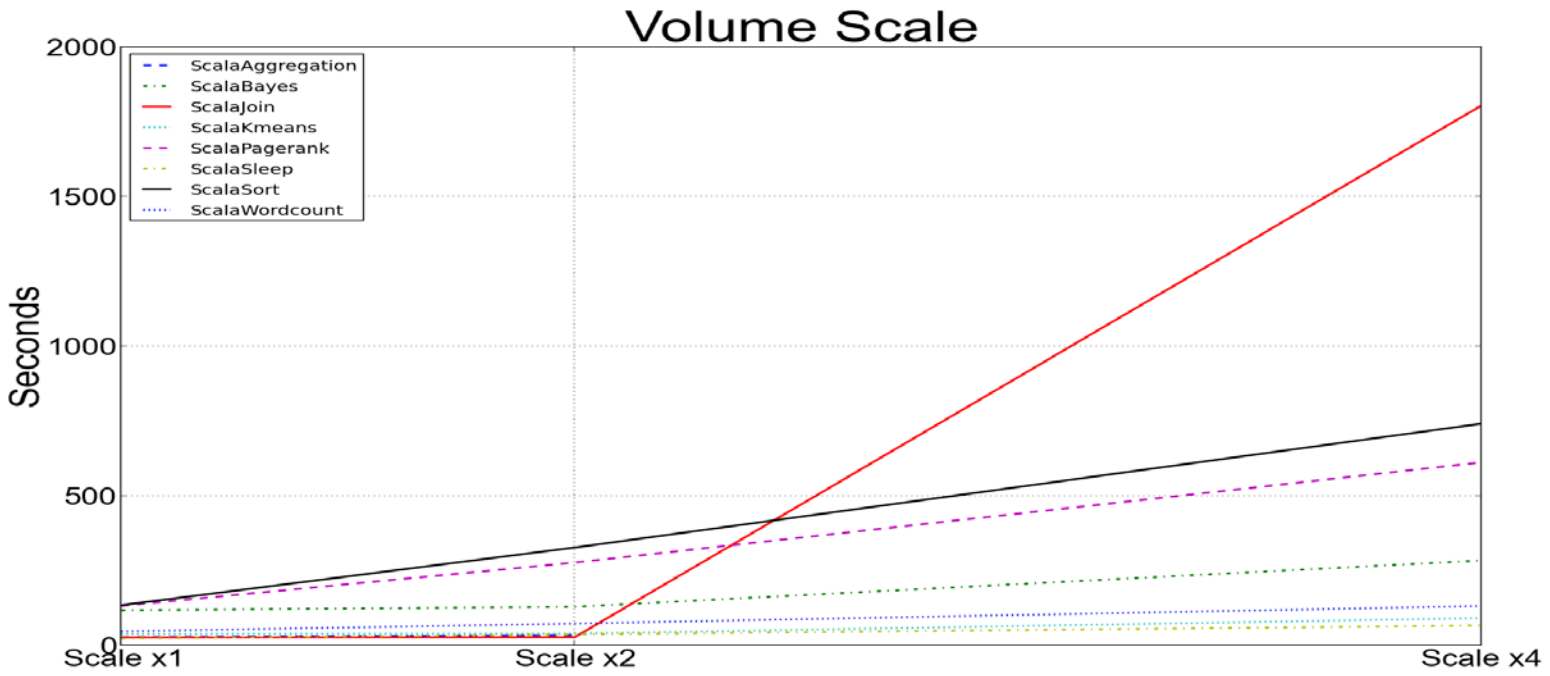
Impact of Network bandwidth



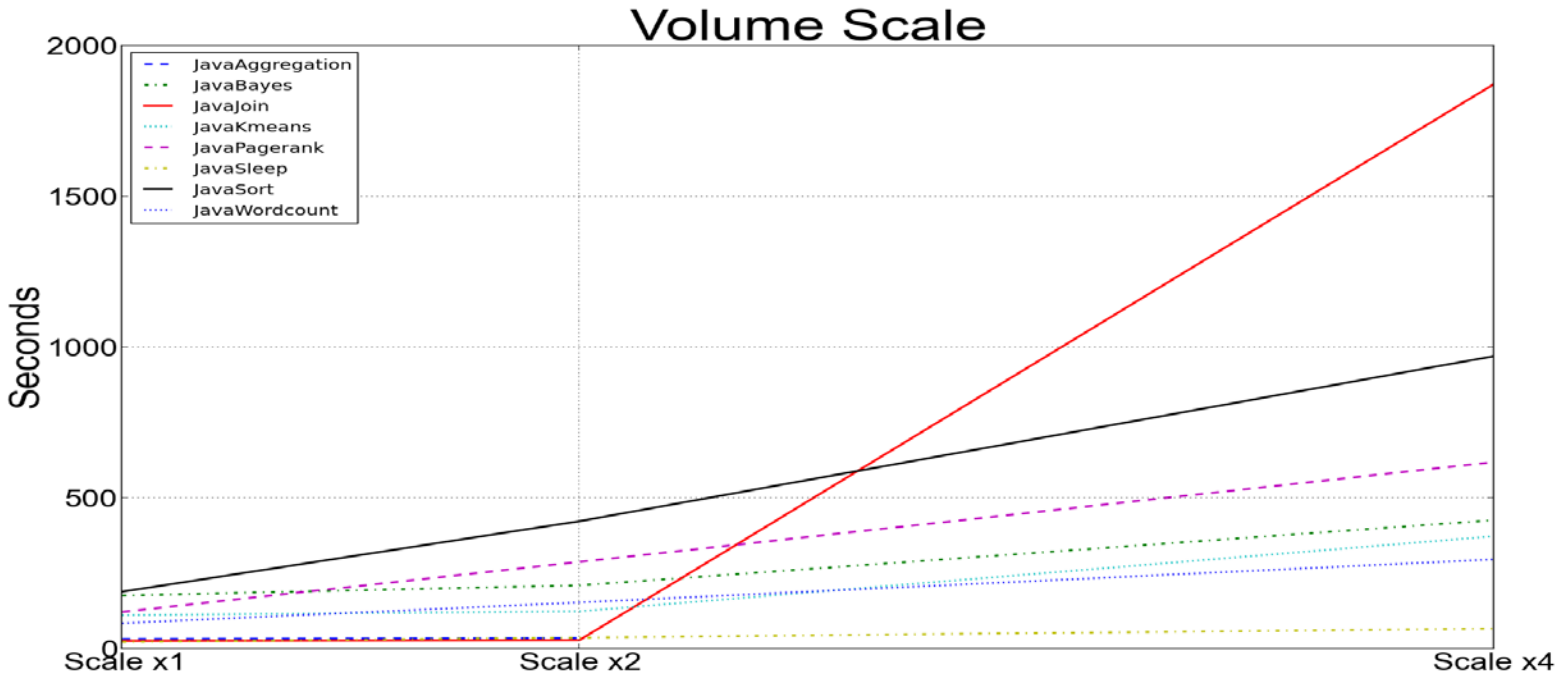
Impact of Network bandwidth



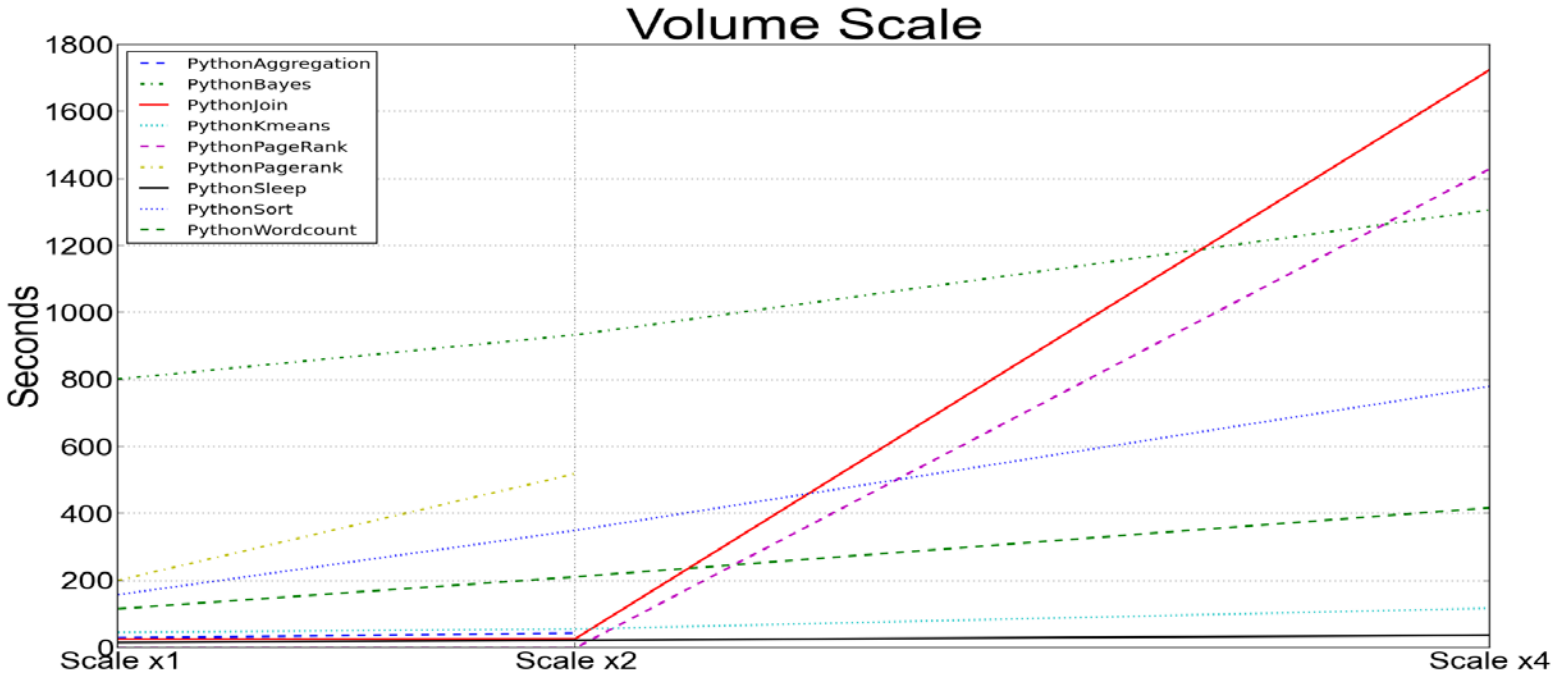
Data volume scalability Spark/scala



Data volume scalability Spark/java



Data volume scalability Spark/python

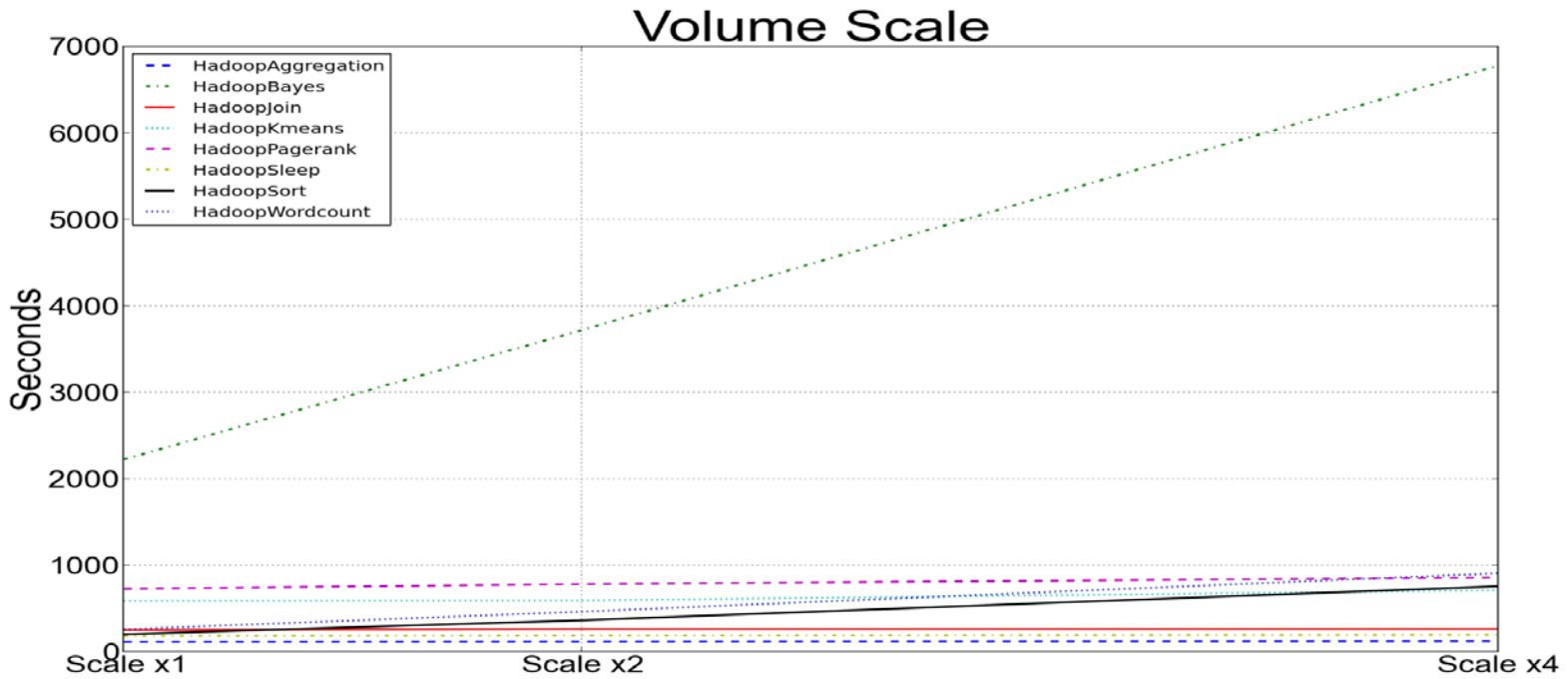


Q & A

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

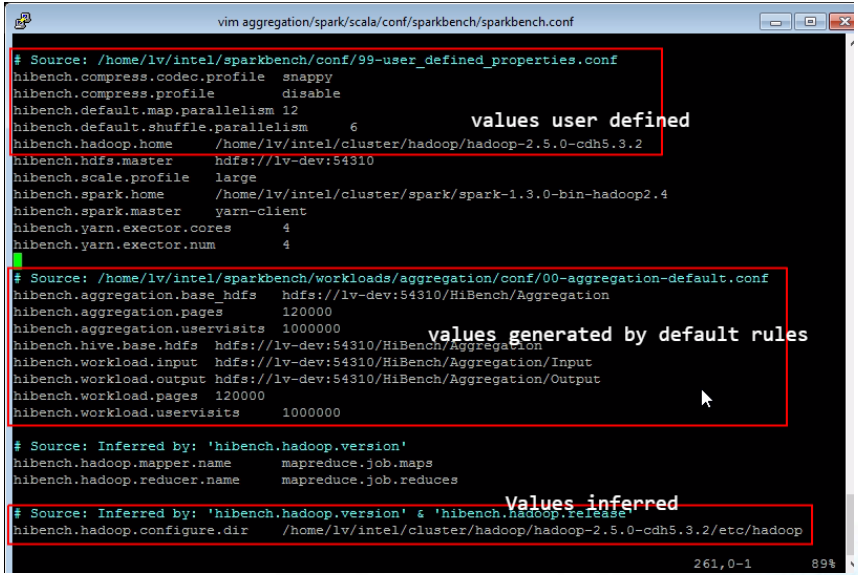
Backup

Data volume scalability – hadoop1



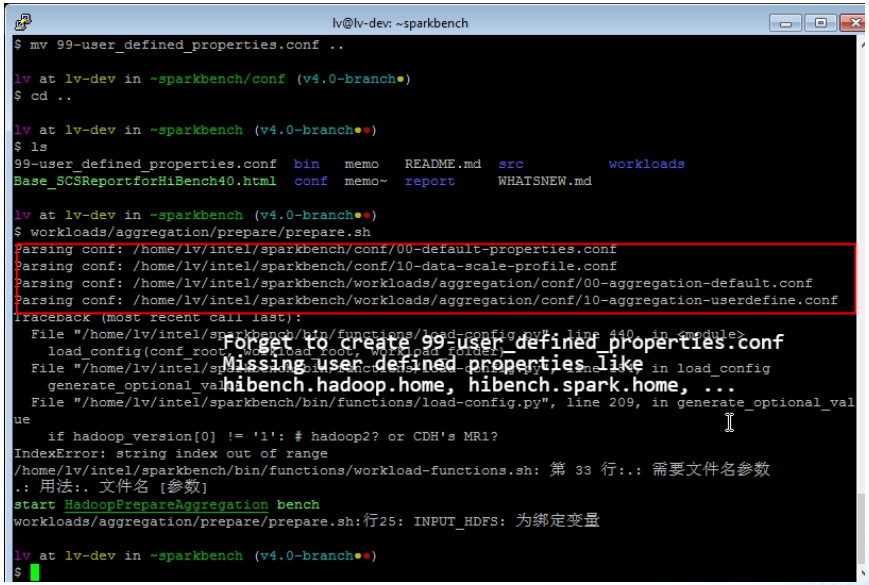
Report configuration example

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



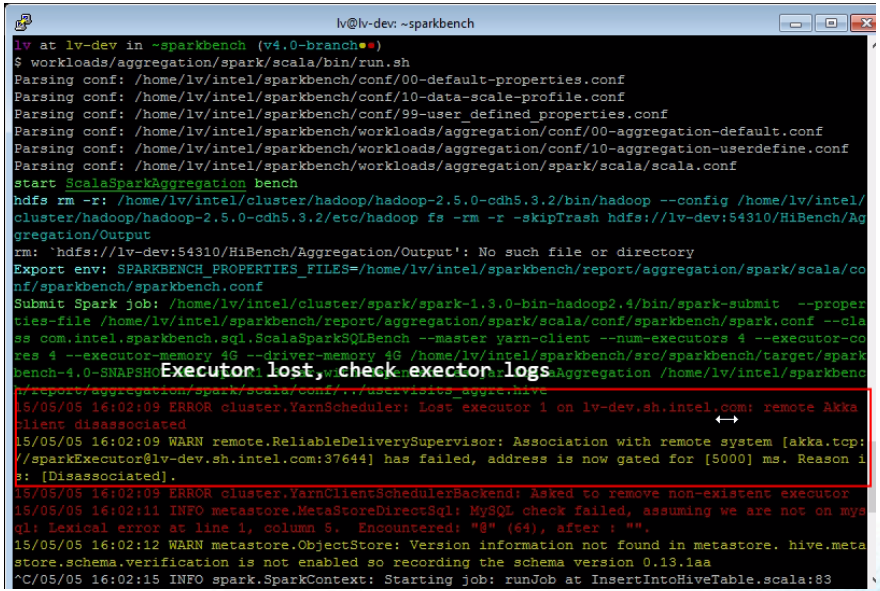
Troubleshooting

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



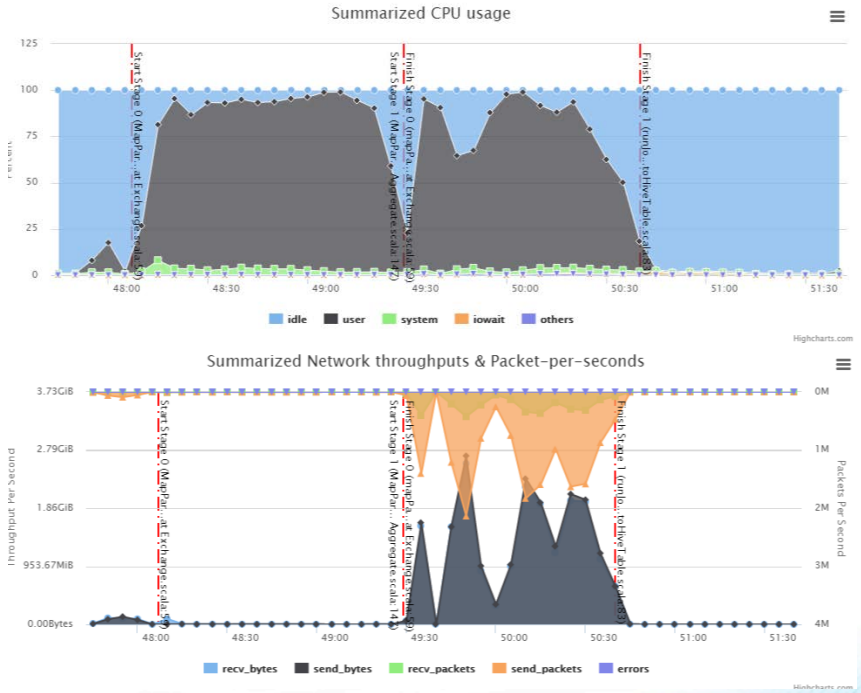
Troubeshooting(2)

- 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.



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)