SDCC 2017深圳站
大数据技术实战峰会

CSDN

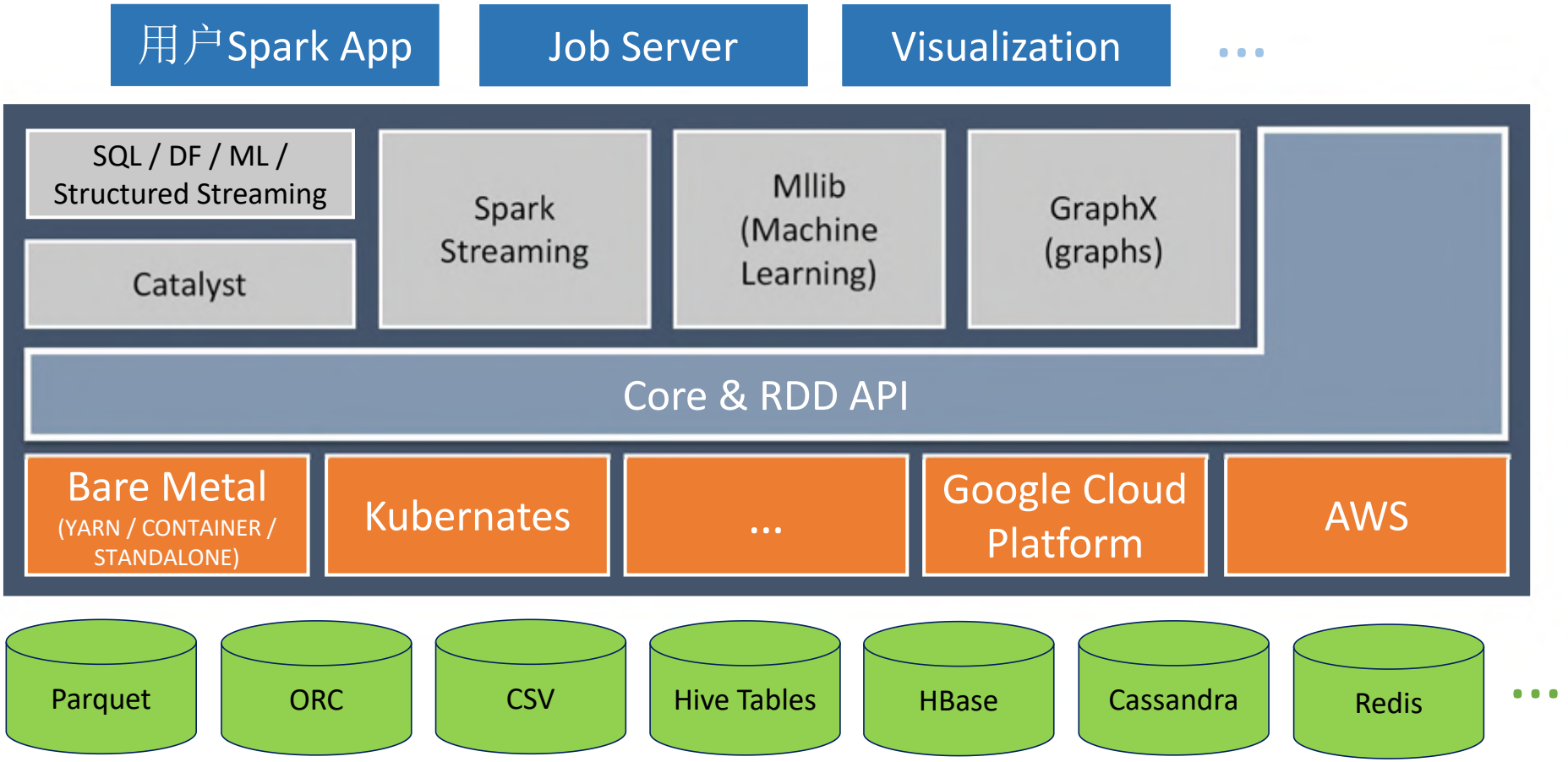
Apache Spark大数据计算性能分析与优化

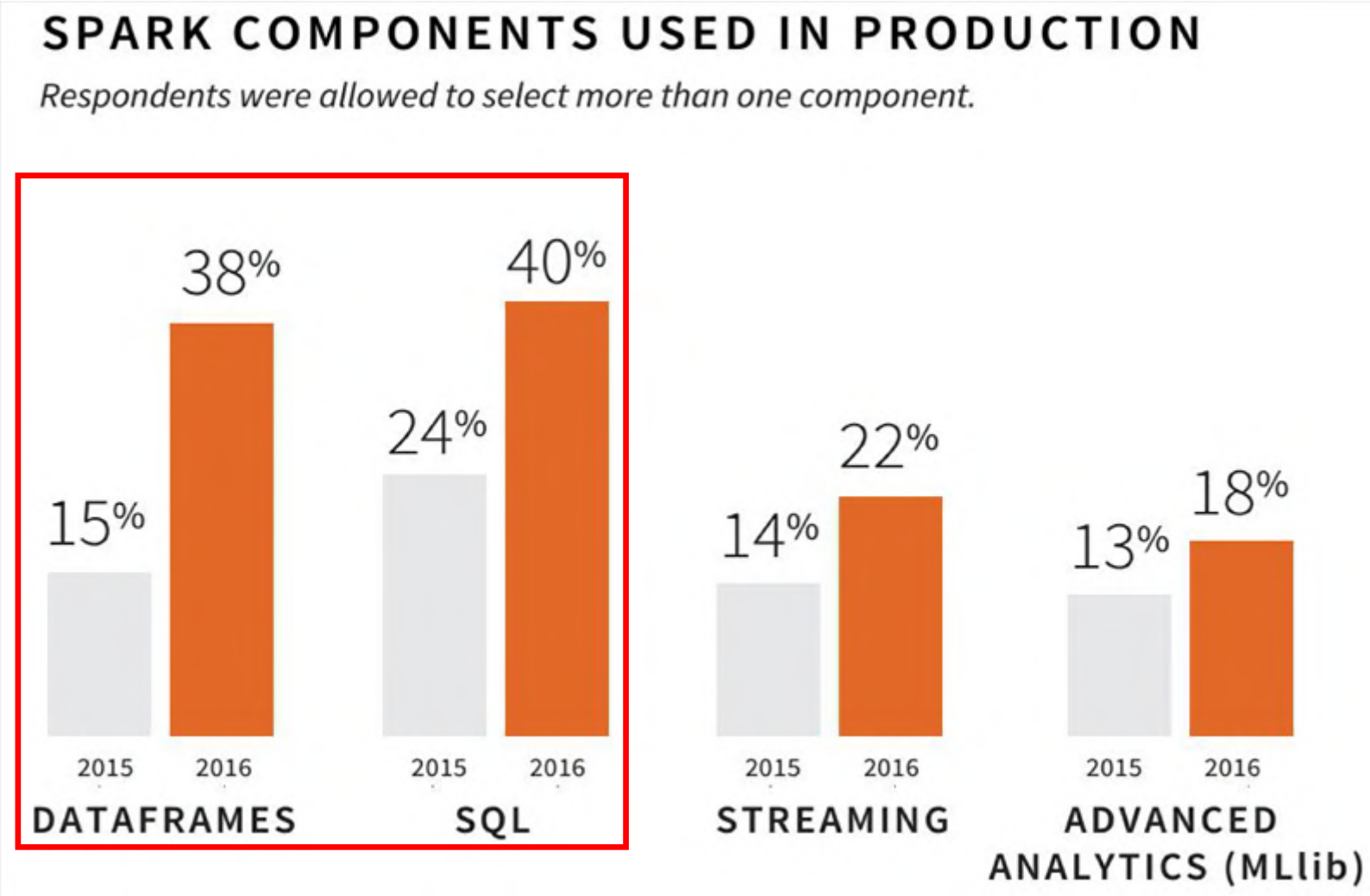
程浩 (hao.cheng@intel.com)

- 关于我自己
- Spark概要简介
- Spark SQL基准测试
- 性能比较分析和启发
- 进行中的优化工作预告

程浩，任职于Intel亚太研发中心，大数据技术团队，Spark研发经理，Spark（活跃）开发者

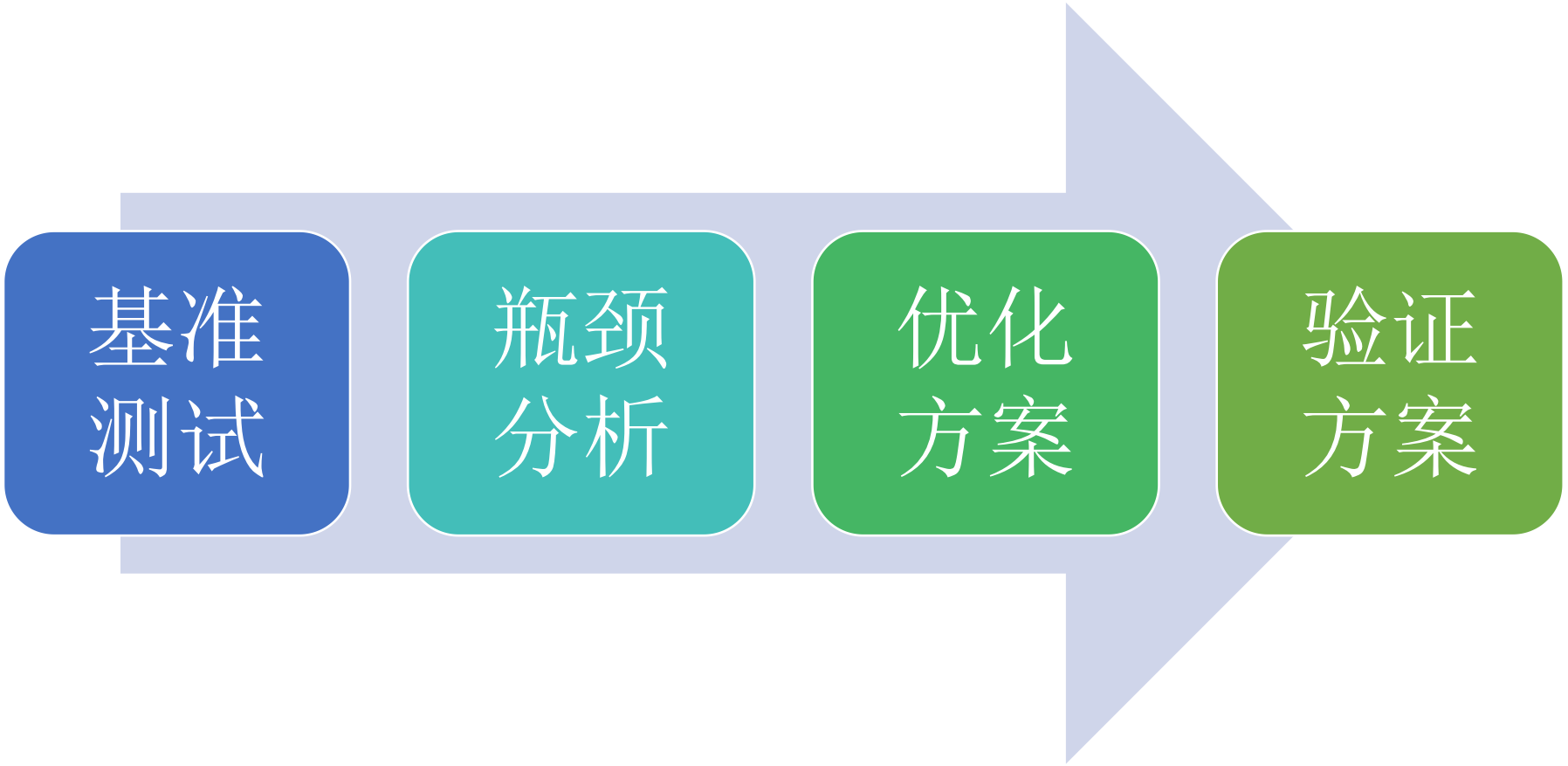
- 关于我自己
- Spark概要简介
- Spark SQL基准测试
- 性能比较分析和启发
- 进行中的优化工作预告


用户Spark App
Job Server
Visualization
...
SQL / DF / ML / Structured Streaming
Catalyst
Spark Streaming
Mllib (Machine Learning)
GraphX (graphs)
Core & RDD API
Bare Metal (YARN / CONTAINER / STANDALONE)
Kubernates
...
Google Cloud Platform
AWS
Parquet
ORC
CSV
Hive Tables
HBase
Cassandra
Redis
...



https://databricks.com/blog/2016/09/27/spark-survey-2016-released.html

- 关于我自己
- Spark概要简介
- **Spark SQL基准测试**
- 性能比较分析和启发
- 进行中的优化工作预告


基准测试
瓶颈分析
优化方案
验证方案

实验环境和测试集

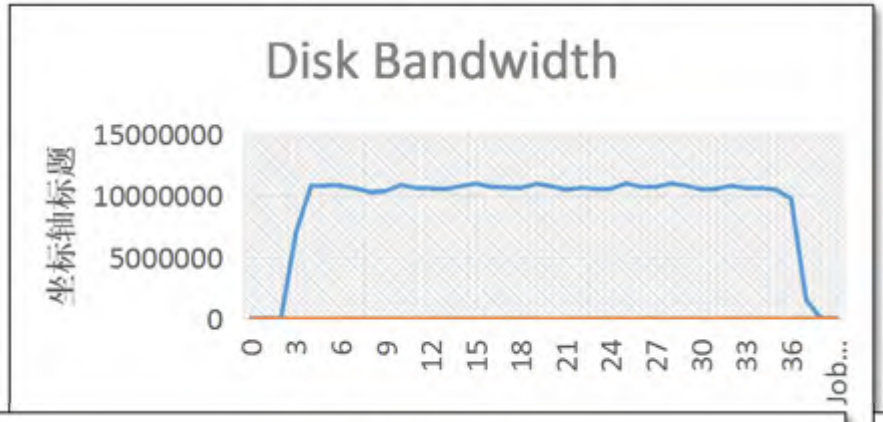
Nodes	Master	Slave
Roles	Hadoop Name Node, Spark Master	Hadoop Data Node, Spark Slaves
Services	Name Node, Resource Manager	Data Node, Node Manager
Numbers	1	7
Processer	Intel Xeon E5-2650 v3 (HSW) / Intel Xeon E5-2680 v4 (BDW) (Dual Socket / node)	
Memory	256GB	128 / 256GB
Storage	OS Disk: 480GB SSD	OS Disk: 480GB SSD Data Disk: 1TB SATA HDD x 8 / Data Disk: Intel S3520 SSD x 8 / Data Disk: Intel P3600 SSD x 3
Network	10Gb	10Gb

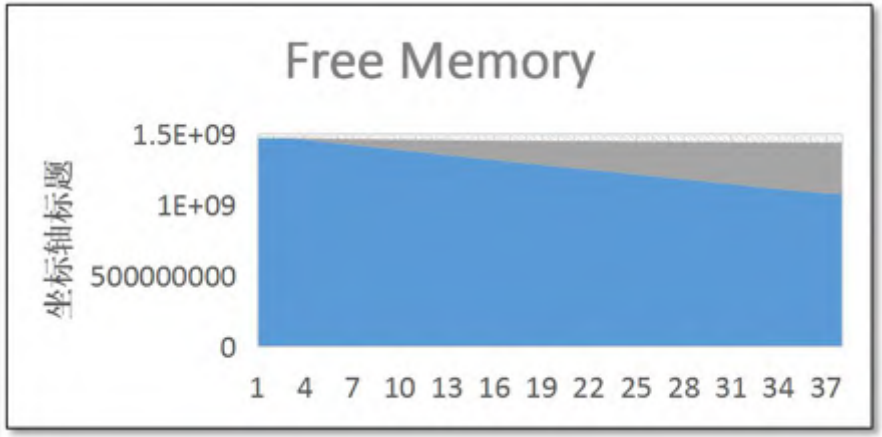
Workload (TPC-DS)	
Queries	19,42,43,52,55,63,68,72,98
Data Scale (Raw Data)	10 TB
Data Format	Parquet
Compression Codec	Snappy
Data Size	~3TB

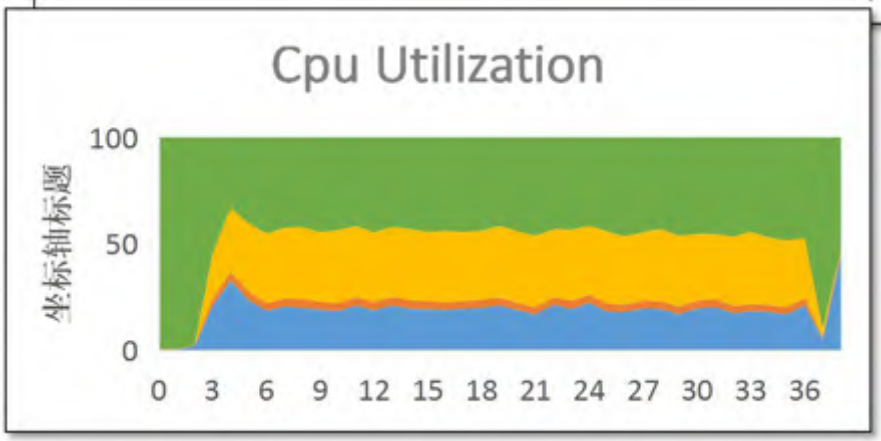
Hadoop/Spark Configuration	
Hadoop version	2.7.3
Spark version	2.1.0
Executor memory	25~40 GB
Executor Cores	8 – 10 / executors
Executor Number	5 / nodes
Spark Mode	yarn-client
JDK Version	1.8.0_112
memory.Overhead	10% Executor Memory
Shuffle Partition #	200
Broadcast threshold	30MB
broadcastTimeout	3600 sec
GC	Parallel GC

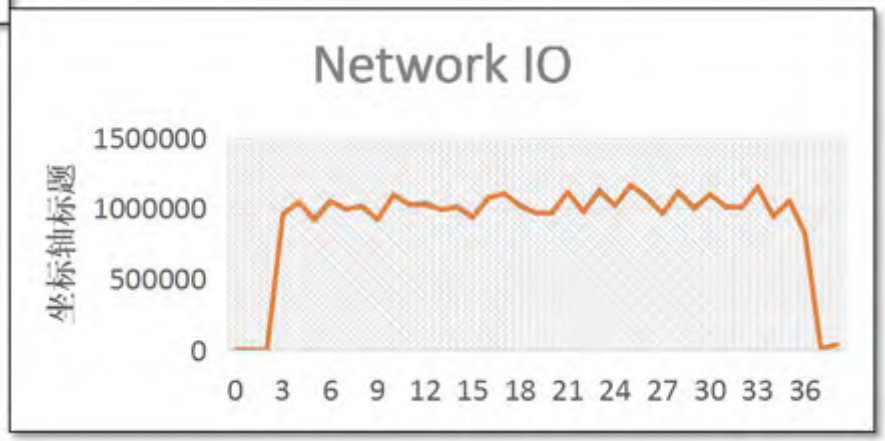
基准测试的性能数字和结论均在相关测试环境下获得，只有相对参考意义，请谨慎使用。

Performance Analysis Tool(PAT) 适用于与在分布式环境下收集系统资源信息，包括CPU、磁盘、网络、内存等，并以图形化的形式展现出来。





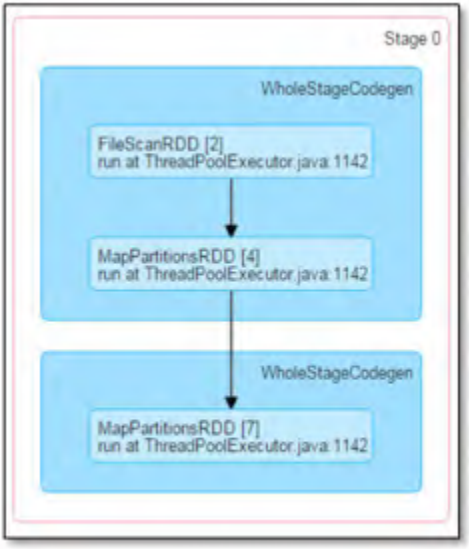




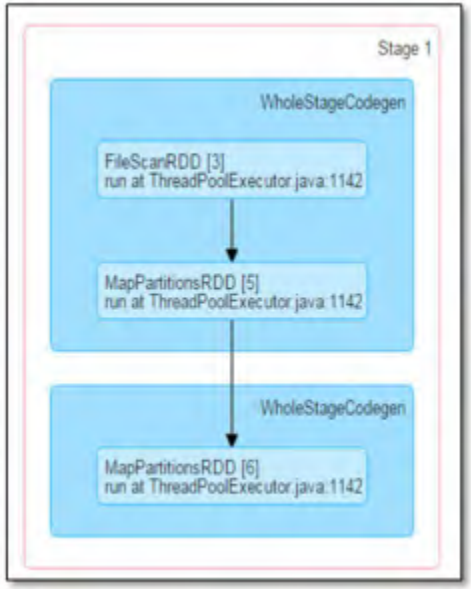
PAT: https://github.com/intel-hadoop/PAT

```
SELECT dt.d_year, item.i_category_id, item.i_category, sum(ss_ext_sales_price)
FROM date_dim dt, store_sales, item
WHERE dt.d_date_sk = store_sales.ss_sold_date_sk
      AND store_sales.ss_item_sk = item.i_item_sk
      AND item.i_manager_id = 1
      AND dt.d_moy=11
      AND dt.d_year=2000
GROUP BY      dt.d_year
            ,item.i_category_id
            ,item.i_category
ORDER BY   sum(ss_ext_sales_price) DESC , dt.d_year
            ,item.i_category_id
            ,item.i_category
```

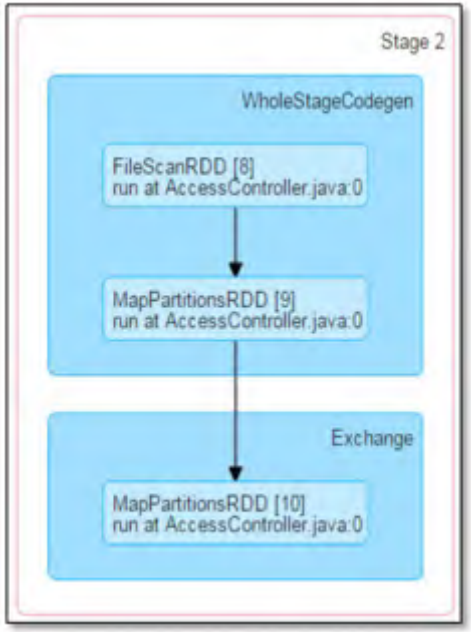
Spark SQL DAG for Q42



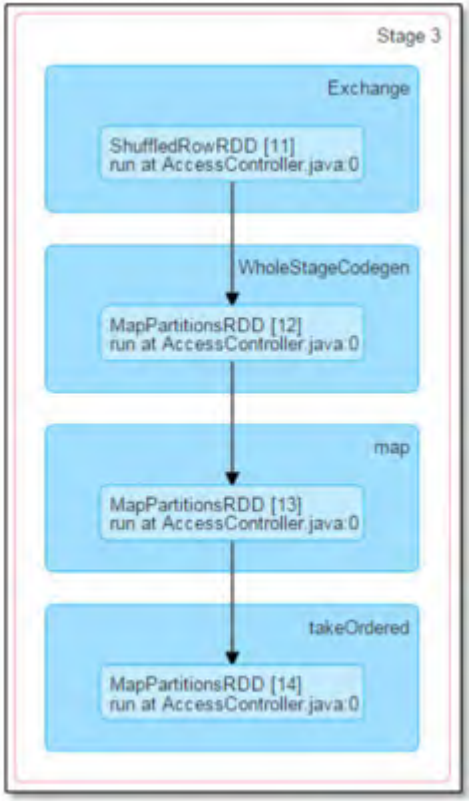
Input Data Size: 291.5KB
Duration: 0.2sec



Input Data Size: 2.3MB
Duration: 0.2sec



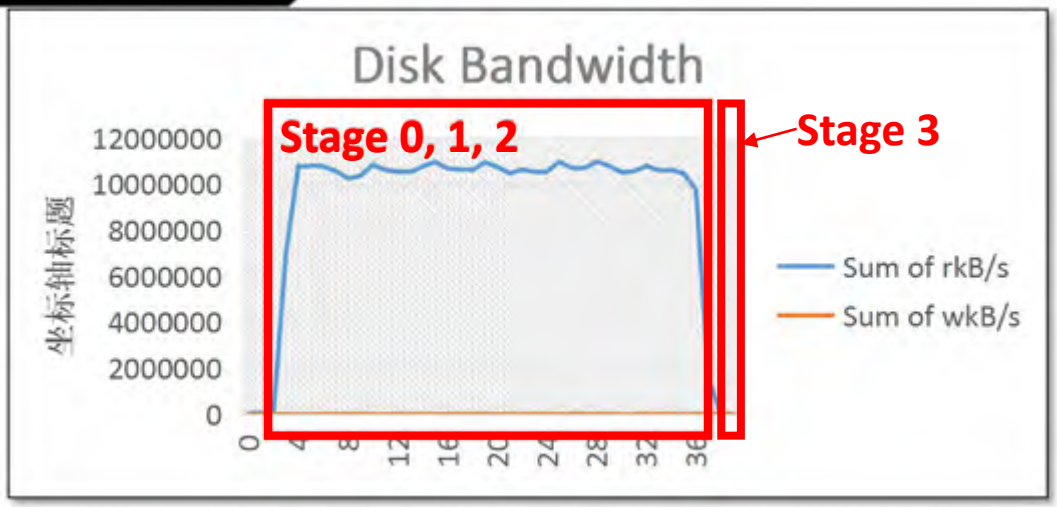
Input Data Size: 221.4GB
Duration: 35s
Shuffle Write Size: 8.8MB

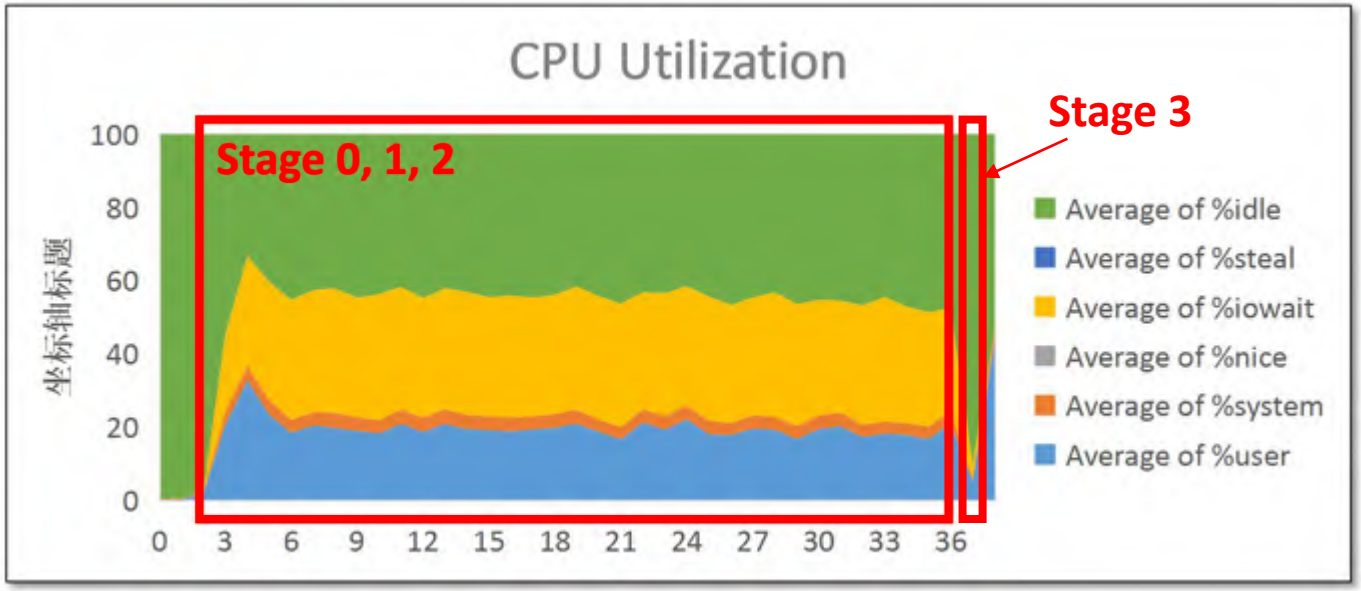


Input Data Size: n/a
Duration: 1s
Shuffle Read Size: 8.8MB

Q42的CPU & Disk使用情况

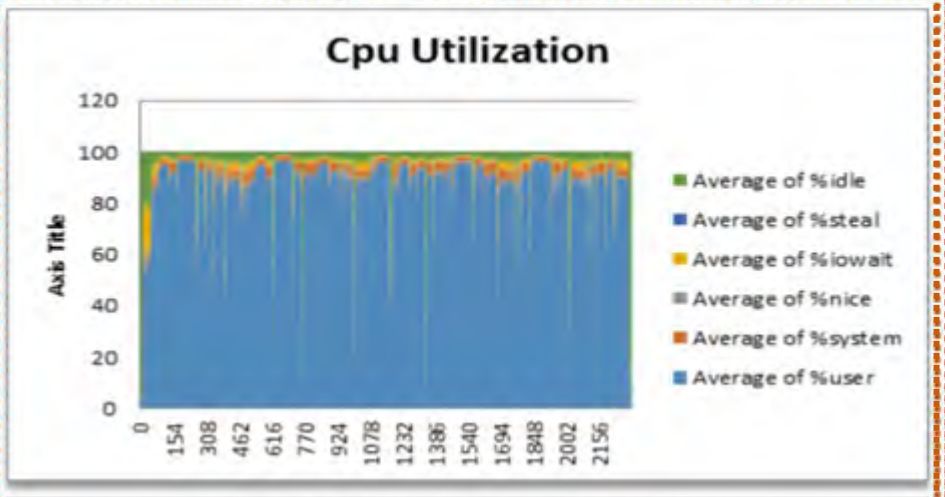
- 1 Master + 7 Workers
- Spark 2.1 on YARN
- Total Data Size = 10TB
- Use Intel S3520 1.6 TB SATA SSD * 8
- CPU: Intel HSW E5 2650 v3 (20 vcore)
- Spark Cores: 40 per node (dual sockets)
- Execution Time: 38sec

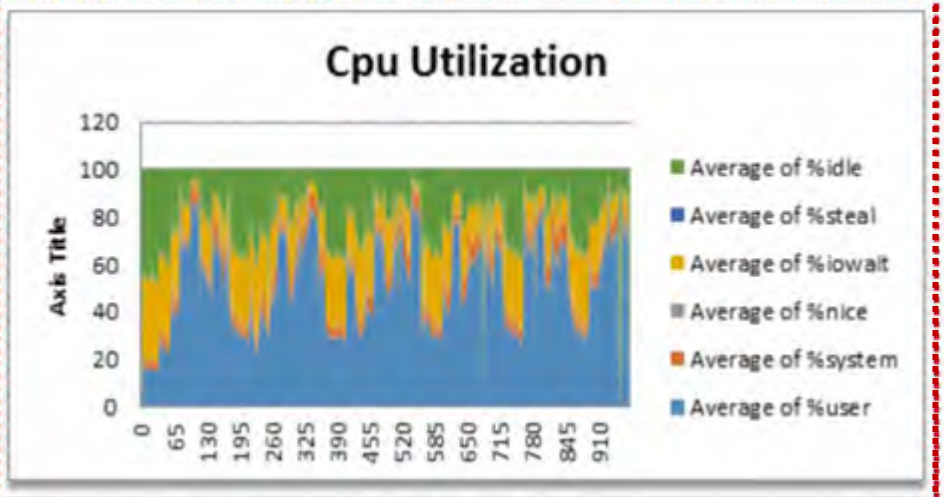


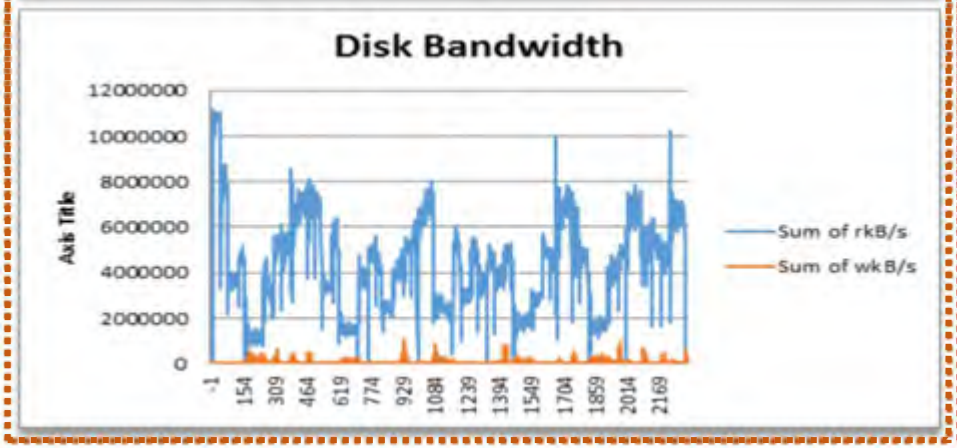


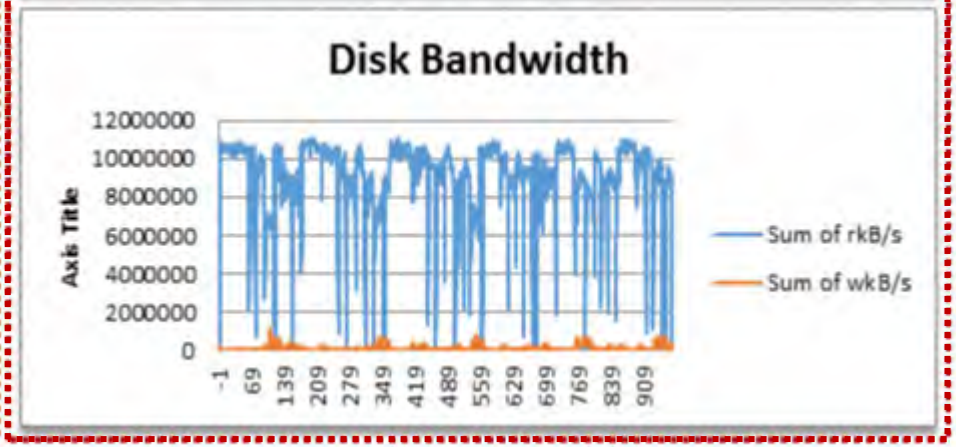
Is an IO Intensive workload w/ lots of Disk Read I/O in Stage 0, 1, 2.

- 关于我自己
- Spark概要简介
- Spark SQL基准测试
- **性能比较分析和启发**
- 进行中的优化工作预告



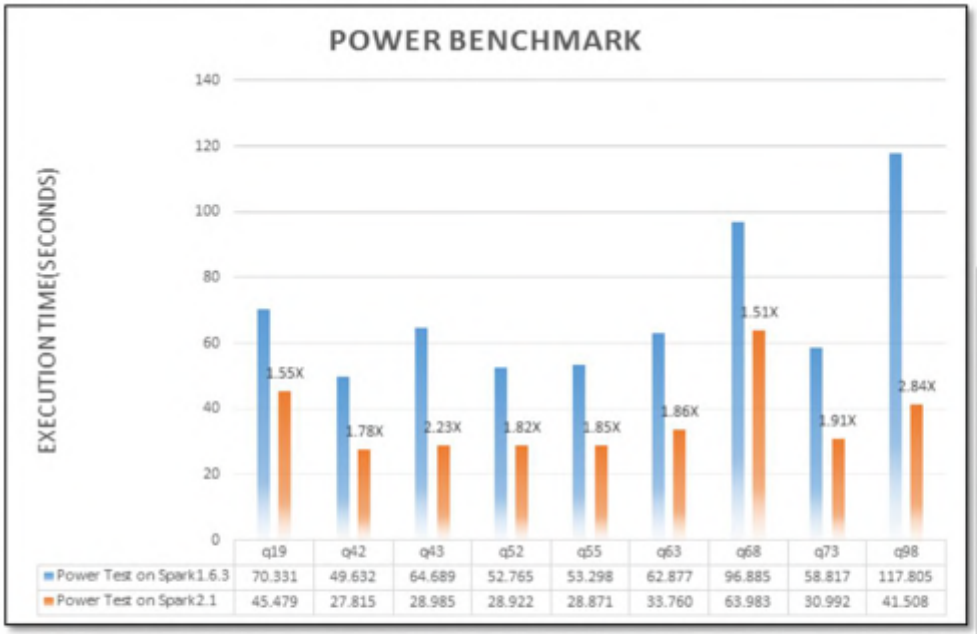


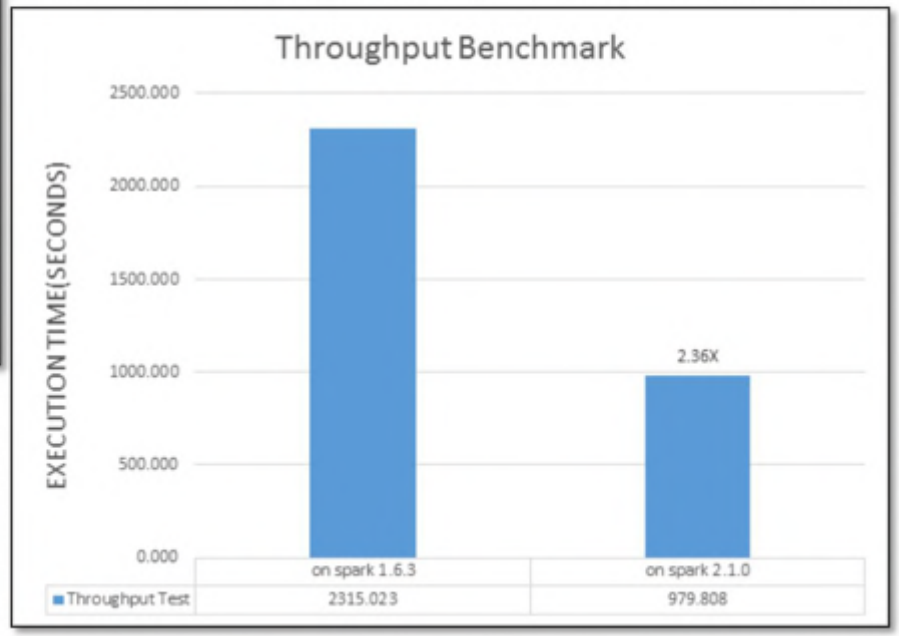




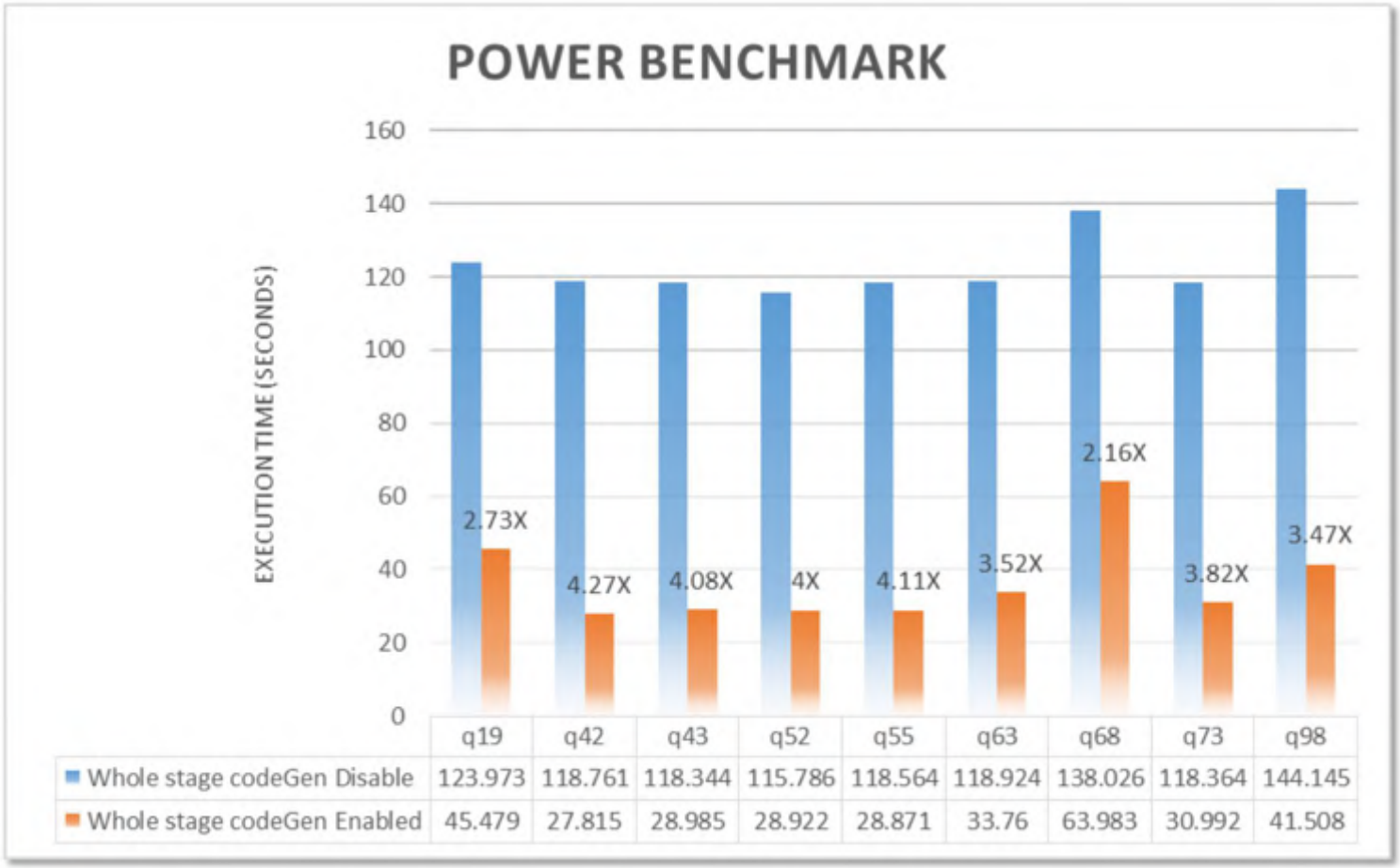
SPARK 1.6

SPARK 2.1





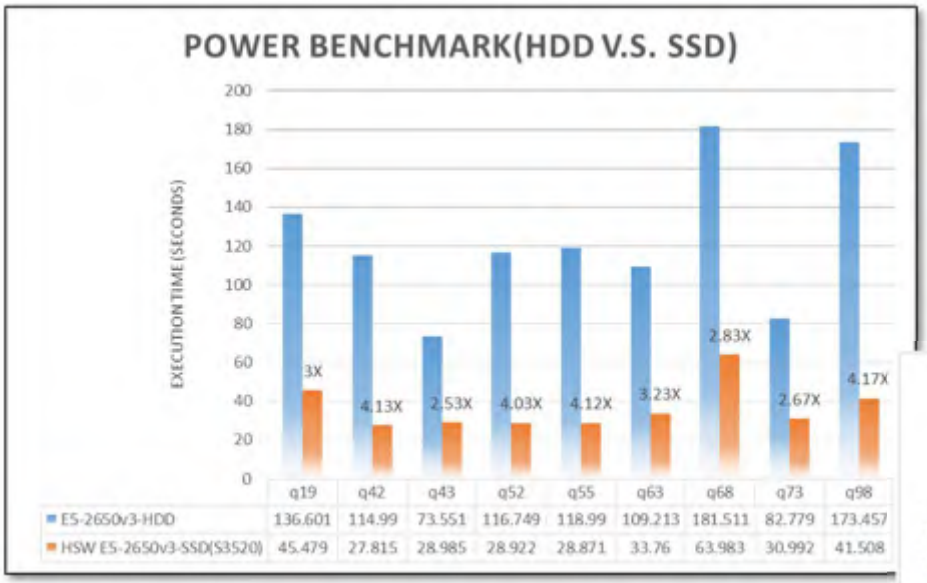
Spark 2.1 boost 1.5X~2.8X performance!

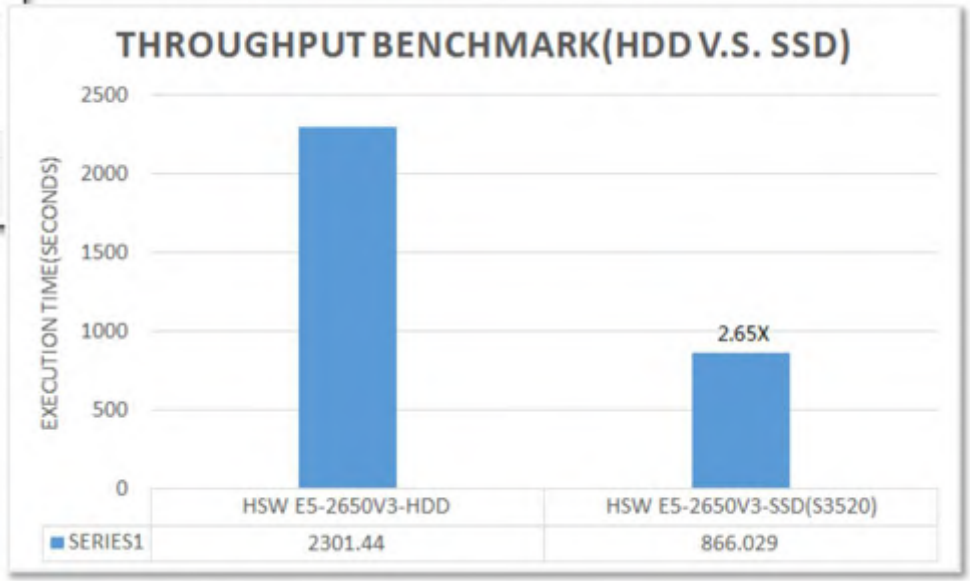


	q19	q42	q43	q52	q55	q63	q68	q73	q98
Whole stage codeGen Disable	123.973	118.761	118.344	115.786	118.564	118.924	138.026	118.364	144.145
Whole stage codeGen Enabled	45.479	27.815	28.985	28.922	28.871	33.76	63.983	30.992	41.508

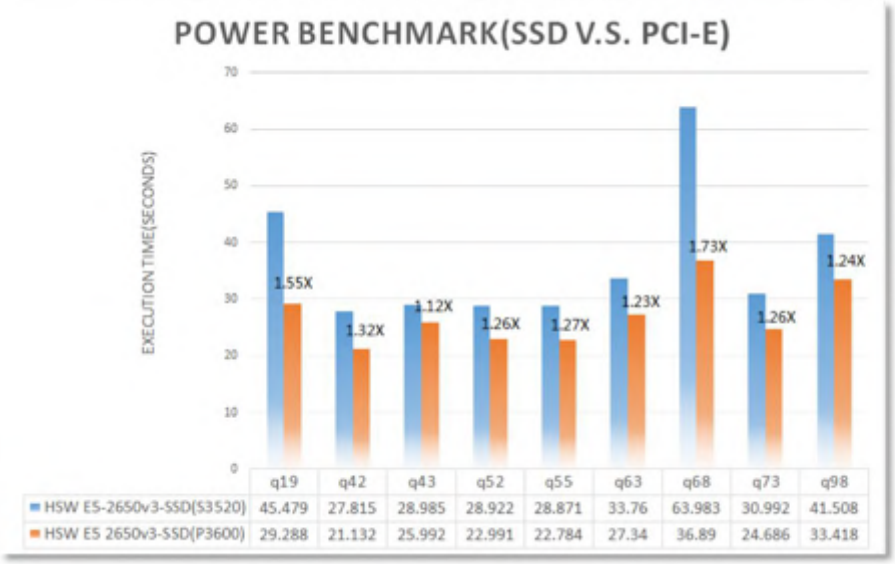
"Whole Stage Codegen" provides 3X performance boost!

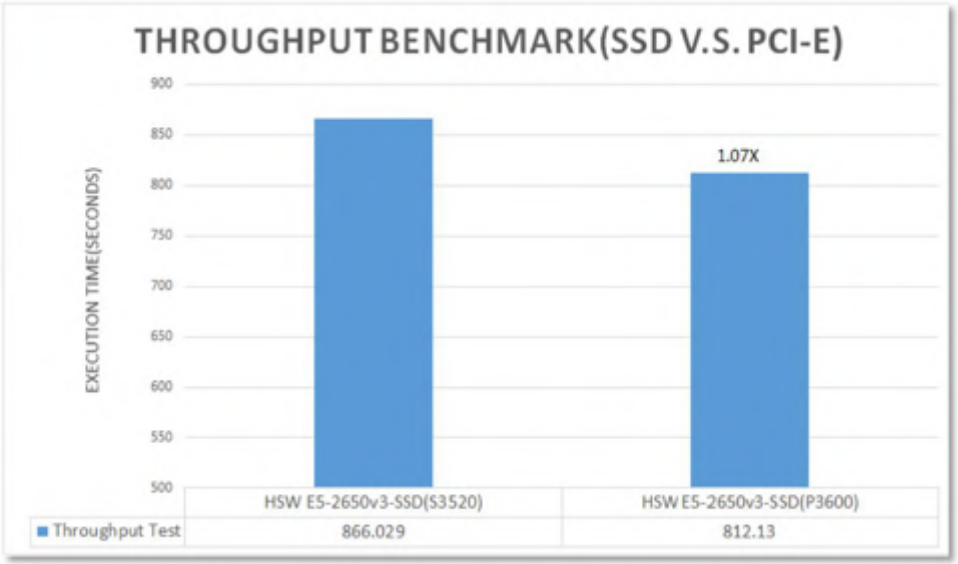
https://databricks.com/blog/2016/05/23/apache-spark-as-a-compiler-joining-a-billion-rows-per-second-on-a-laptop.html



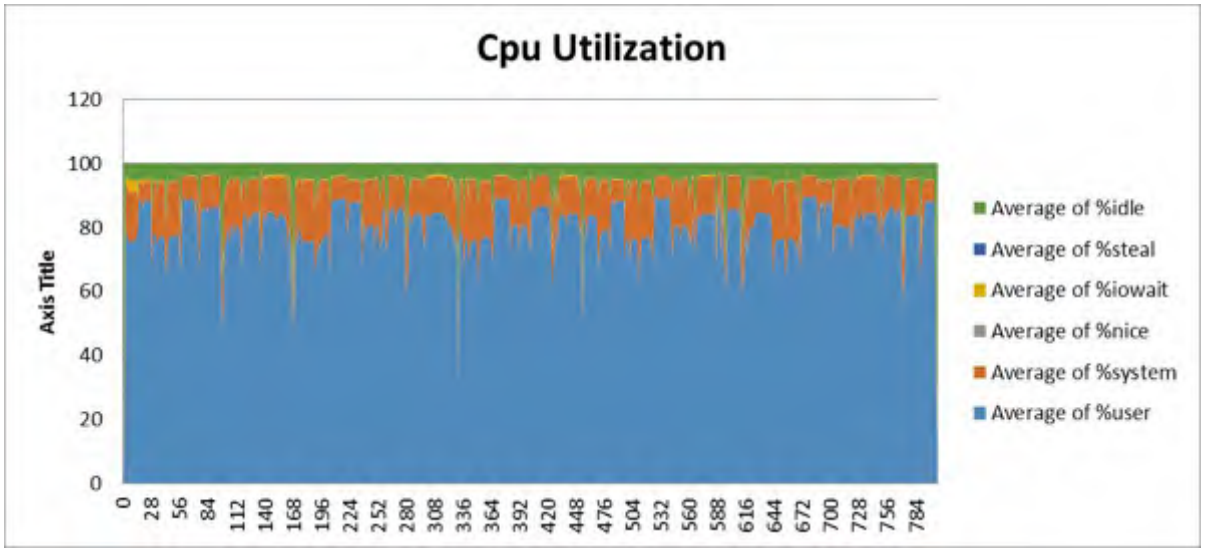


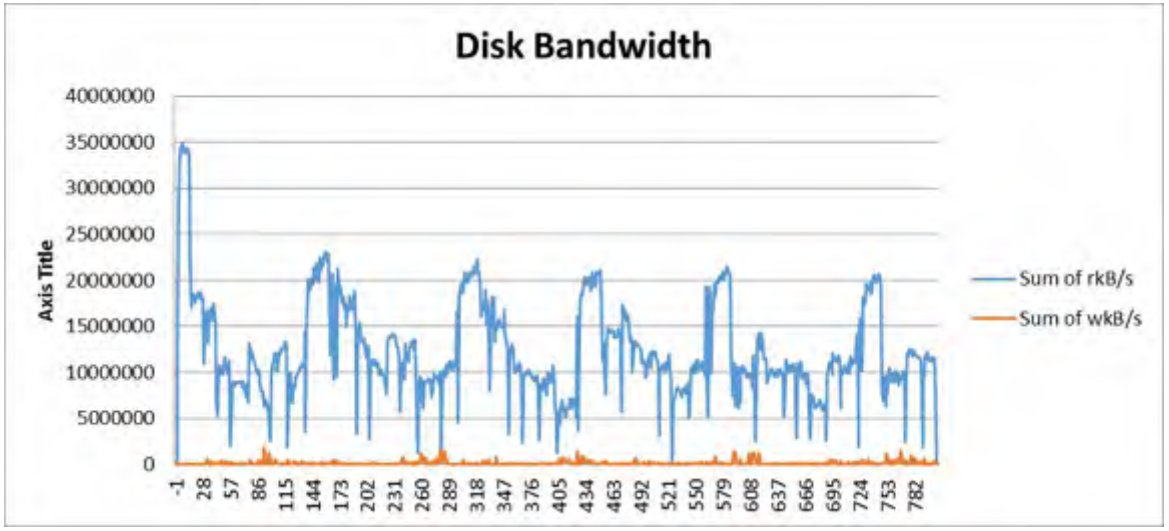
Use SSD brings avg. 2.82X performance improvement!

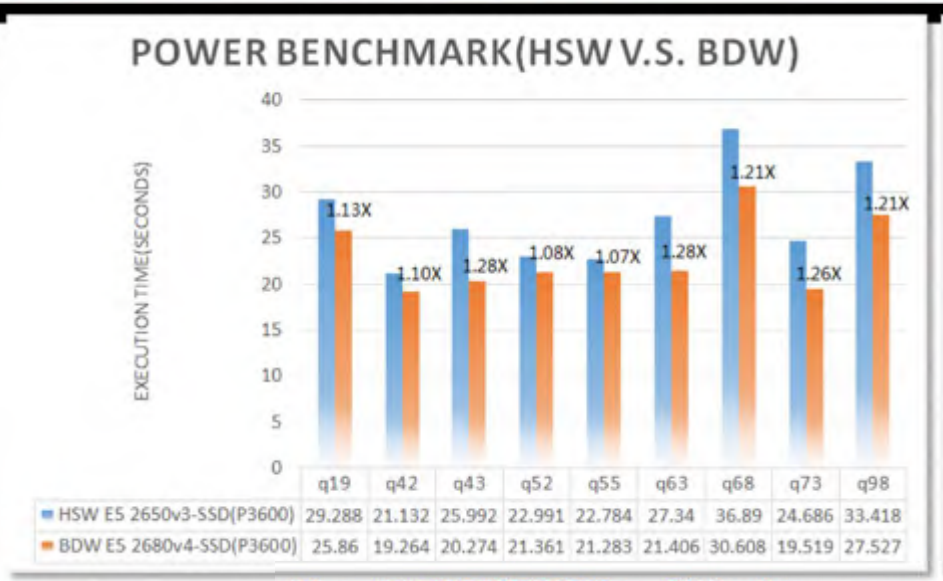


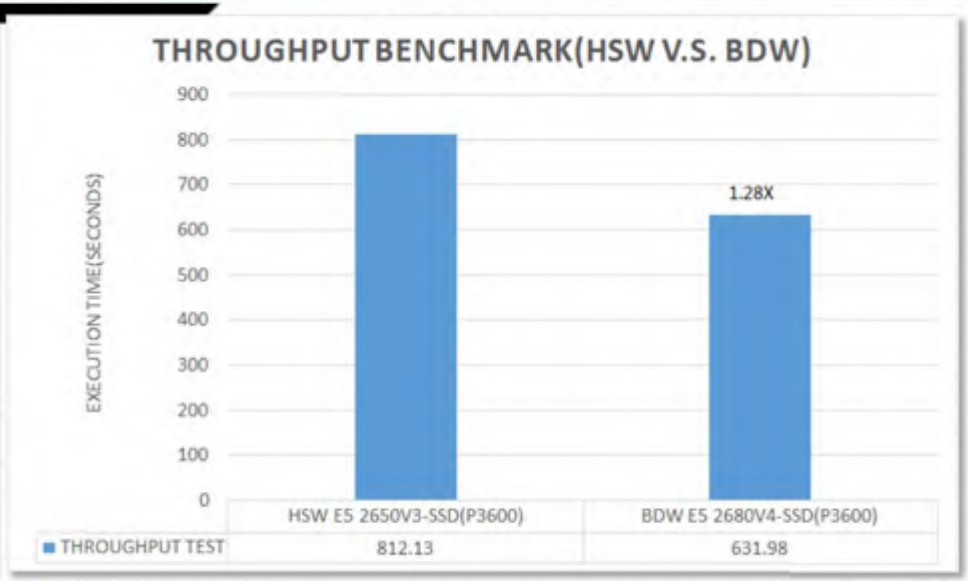


Use PCI-E SSD brings avg. 1.11X performance improvement!

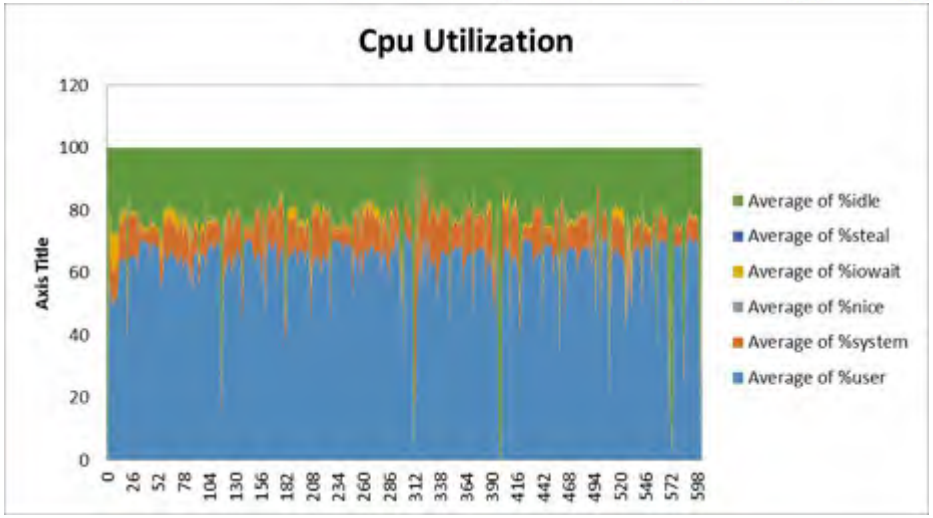

Cpu Utilization
Axis Title
120
100
80
60
40
20
0
0
28
56
84
112
140
168
196
224
252
280
308
336
364
392
420
448
476
504
532
560
588
616
644
672
700
728
756
784
Average of %idle
Average of %steal
Average of %iowait
Average of %nice
Average of %system
Average of %user

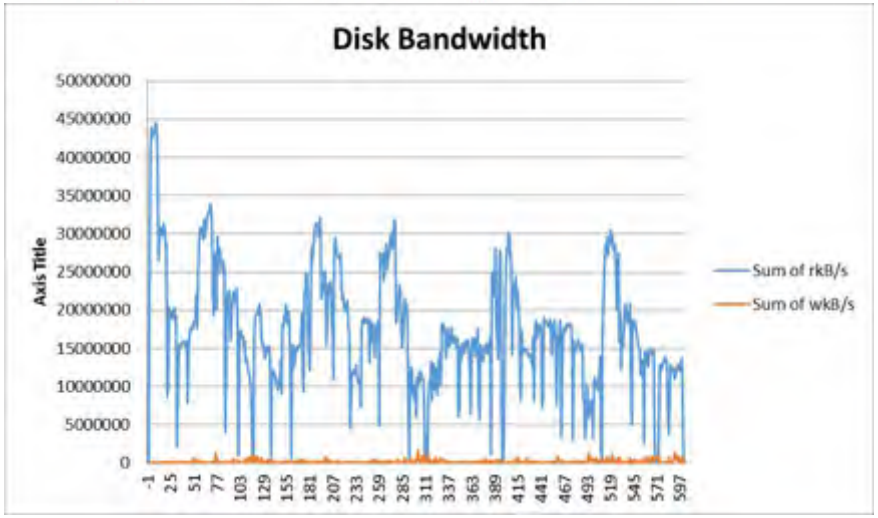

Disk Bandwidth
Axis Title
40000000
35000000
30000000
25000000
20000000
15000000
10000000
5000000
0
-1
28
57
86
115
144
173
202
231
260
289
318
347
376
405
434
463
492
521
550
579
608
637
666
695
724
753
782
Sum of rkB/s
Sum of wkB/s



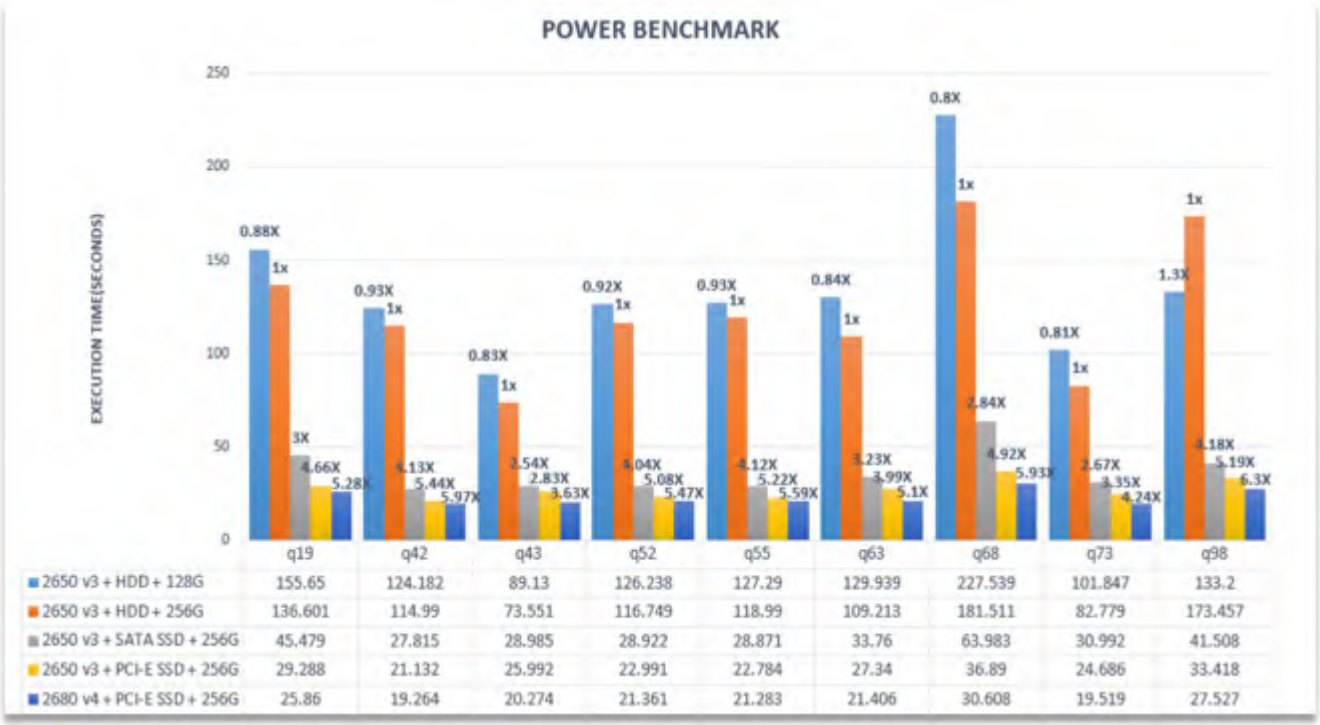


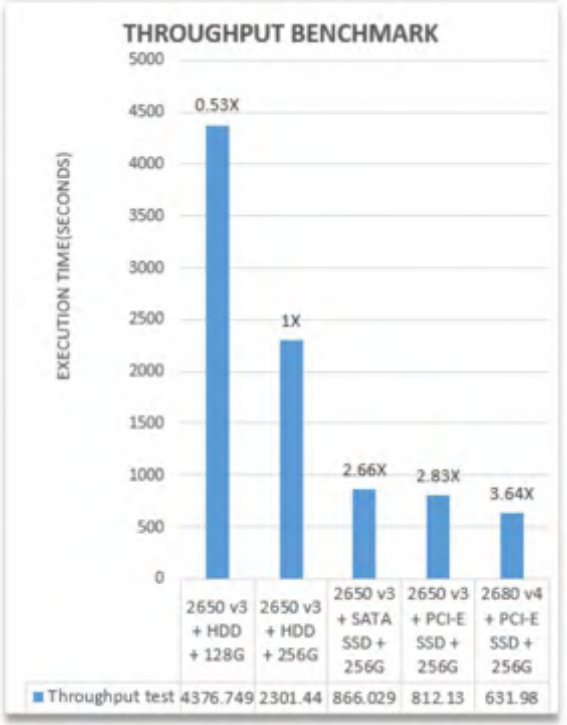
Use BDW(2680 v4) brings avg. 1.23X performance improvement!





完整性能分析对比





- SPARK 2.1有明显的性能提升
- 128 GB 内存可能会极大制约性能，因为分配给单核的内存有限可能会导致GC加剧
- SATA SSD (Intel S3520) 相对HDD性能提升非常明显(2.6x – 4.9x)
- PCI-E SSD (Intel P3600) 相对HDD性能也会有明显提升，但是只有配合更高端的CPU(如:E5 2680v4)或者网络才能充分发挥其优势

磁盘TCO 模型(顺序读写)

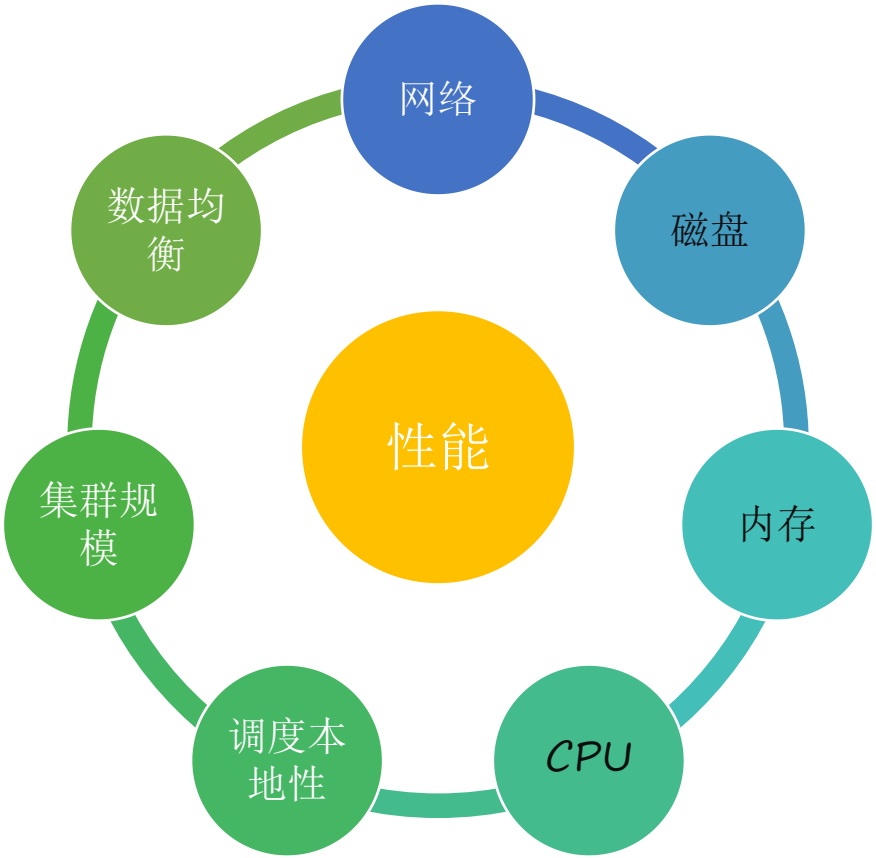
	Intel HSW(2650v3)	
Disk Types	HDD(1TB SATA)	SATA SSD(s3520)
Numbers of Drives	8	8
Total Capacity	1TB x 8	1.2TB x 8
Performance Gain	1x	~2.65x
Drive Cost	$800	$4224
Power Cost	$421	$68
Cooling Cost	$505	$82
Enclosure Cost	$3943	$3943
Reliability	$1008	$339
Total Cost	$6677	$8656
Cost (per GB)	1x	1.08x
Perf (per Dollar)	1.0x	~2.4x

	Intel HSW(2650v3)	Intel BDW(2680v4)
Disk Types	PCIe SSD(p3600)	PCIe SSD(p3600)
Numbers of Drives	3	3
Total Capacity	1.6TB x 3	1.6TB x 3
Performance Gain	~2.83x	~3.64x
Drive Cost	$4782	$4782
Power Cost	$35	$35
Cooling Cost	$42	$42
Enclosure Cost	$0	$0
Reliability	$287	$287
Total Cost	$5146	$5146
Cost (per GB)	-	-
Perf (per Dollar)	1.0x	~1.28x

http://estimator.intel.com/ssddc/ *请以实际网站工具查询信息为准

- 关于我自己
- Spark概要简介
- Spark SQL基准测试
- 性能比较分析和启发
- 进行中的优化工作预告

影响Spark(SQL)性能的主要因素



Adaptive Execution带来的几个好处

- 在作业运行期间，自动根据数据大小将Shuffle Join转换成Broadcast Join；
- 在作业运行期间，自动调整Aggregation、Join的Partition Number，避免手动设置该值导致在不同数据大小规模下的不适用性，或者OOM；
- 在作业运行期间，根据数据分布，自动调节平均并行调度任务数据量；
- 在作业运行期间，自动根据数据量大小调整JOIN的顺序(Runtime CBO)；
- ……

让作业在集群中运行得更加均衡！作业调度粒度更合理！

https://issues.apache.org/jira/browse/SPARK-9850

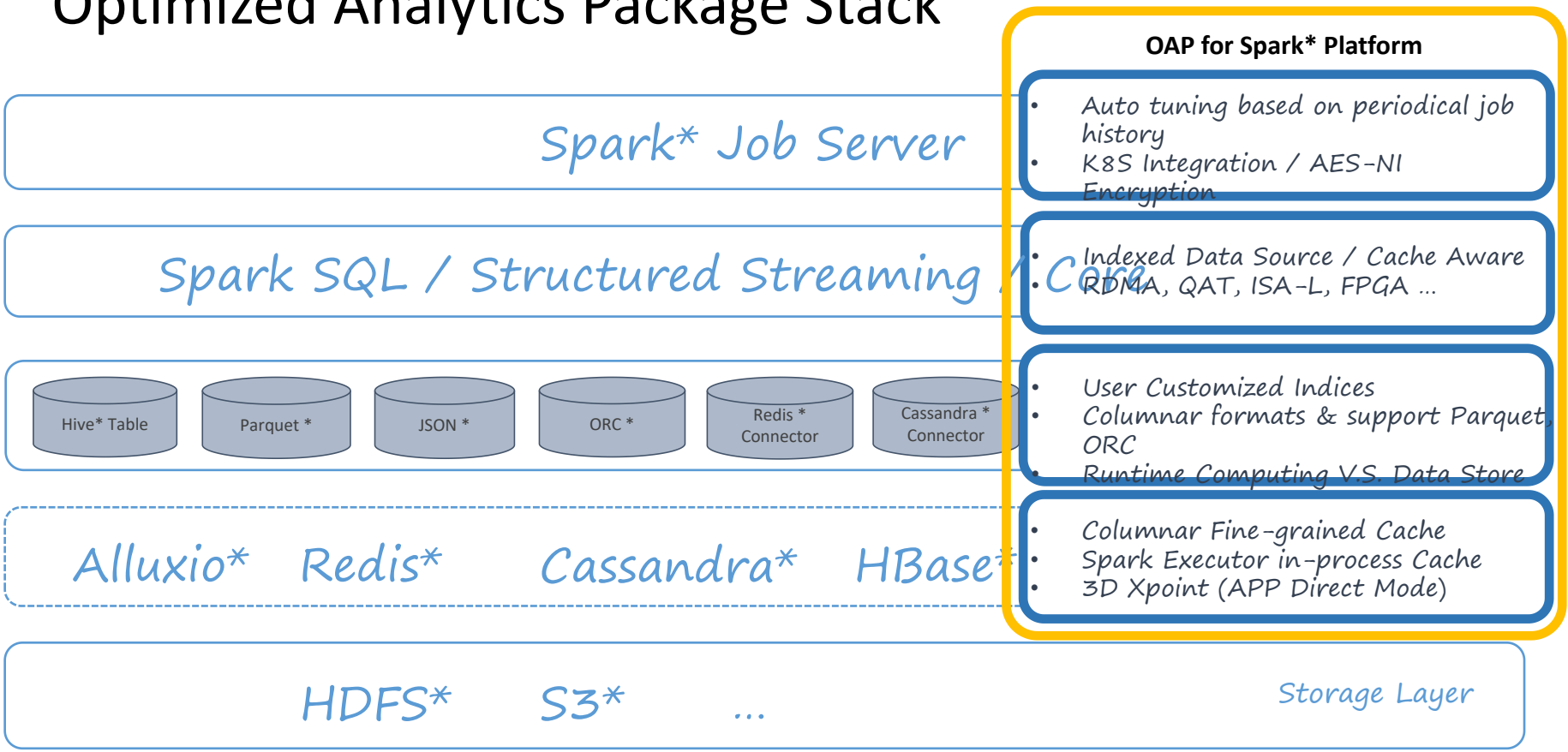
压缩解压性能

Compressor	Ratio	Compression	Decompression
memcpy	1.000	7300 MB/s	7300 MB/s
LZ4 fast 8 (v1.7.3)	1.799	**911 MB/s**	**3360 MB/s**
LZ4 default (v1.7.3)	**2.101**	**625 MB/s**	**3220 MB/s**
LZO 2.09	2.108	620 MB/s	845 MB/s
QuickLZ 1.5.0	2.238	510 MB/s	600 MB/s
Snappy 1.1.3	2.091	450 MB/s	1550 MB/s
LZF v3.6	2.073	365 MB/s	820 MB/s
Zstandard 1.1.1 -1	2.876	330 MB/s	930 MB/s
Zstandard 1.1.1 -3	3.164	200 MB/s	810 MB/s
zlib deflate 1.2.8 -1	2.730	100 MB/s	370 MB/s
LZ4 HC -9 (v1.7.3)	**2.720**	34 MB/s	**3240 MB/s**
zlib deflate 1.2.8 -6	3.099	33 MB/s	390 MB/s

https://github.com/lz4/lz4#benchmarks

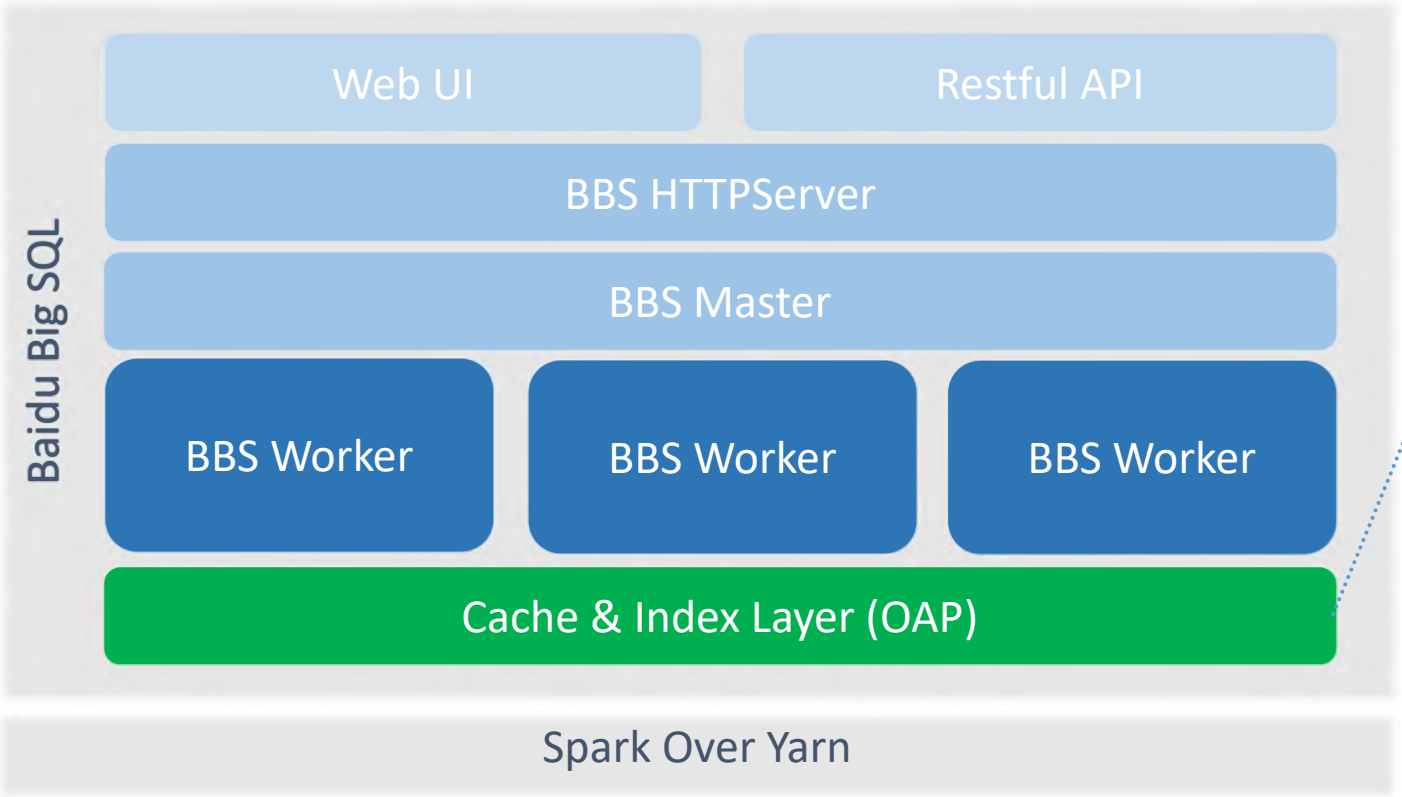
https://software.intel.com/en-us/intel-ipp
https://github.com/01org/isa-l
https://www.intel.com/content/www/us/en/embedded/technology/quickassist/overview.html

Optimized Analytics Package Stack



https://github.com/Intel-bigdata/OAP

Use Case From Baidu



- Big SQL is a widely used ad-hoc query service over spark-sql in Baidu.
- OAP as a core module for boosting the query with index and cache plays an important role in the whole stack.

Q & A