How to Boost 100x Performance for Real World Application w/ Apache Spark

Jie.huang@intel.com
Jiangang.duan@Intel.com

June 2015

Agenda

- Self introduction
- Problem statement
- What we did?
- Case study
- Summary

This is team work!

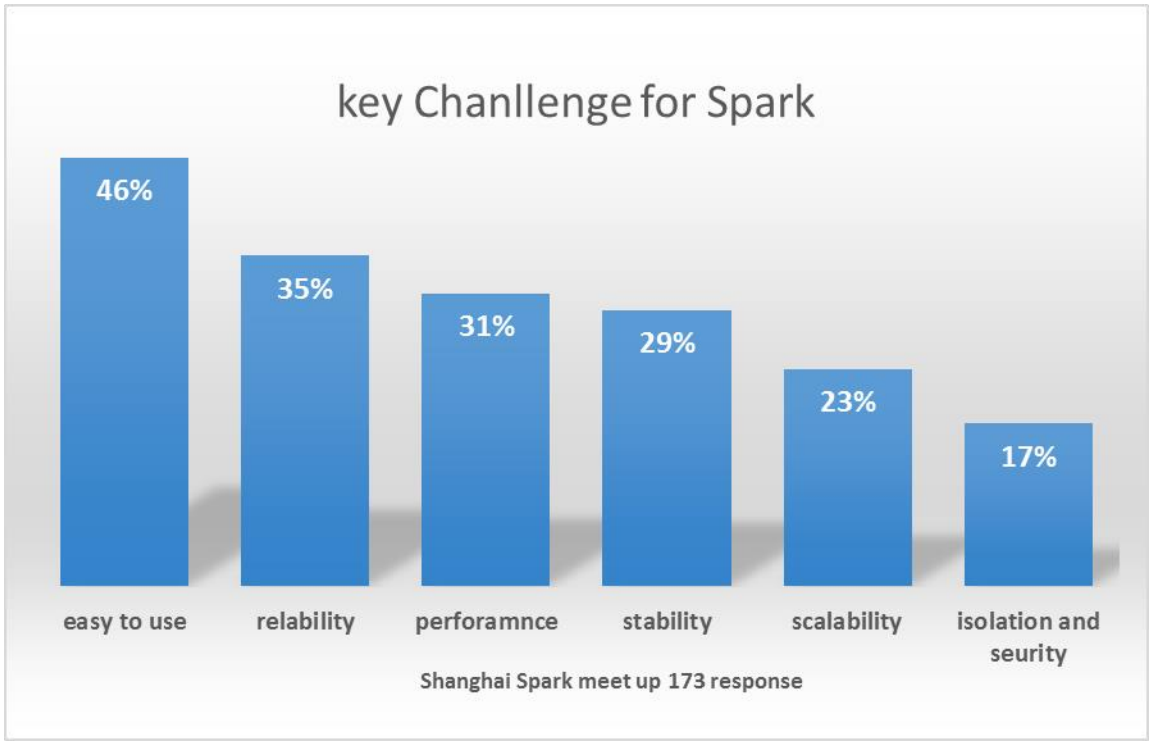
Thank Hao, Daoyuan, Saisai, Mingfei, Jiayin, Liye, Carson, Alex, Lex, Rui, Qi…

Self introduction

- Intel software team @China Shanghai
- Open source focus
- Start to work on Spark from early UCB days
- Working closely with end customers
  - Baidu, iqiyi, Tecent, Qihoo, JD, Sina, paypal…
- Technology and innovation oriented
  - Real-time, in-memory, complex analytics
  - Structure and unstructured data
  - Agility, Multitenancy, Scalability and elasticity
  - Bridging advanced research and real-world applications

Problem statement

Easy to use, reliability/stability and performance/ scalability are the common pain point



OOM
Slow
Variation
Concurrency issue
Memory control
Resource monitor
…

Intel Software

What we did?

- Define a better workload
  - HiBench
- Provide a better profiling tool
  - HiMeter (Dew)
- Regression testing and share w/ community
  - SparkScore Web portal
- Work with customers to solve problems
  - User case study

HiBench

- The bigdata micro benchmark suite
  - Open source released https://github.com/Intel-hadoop/hibench
  - Consists of 10 workloads for different categories
  - Support Hadoop MR and Spark(scala, java, python)
  - MR1/standalone, Yarn
- Extensively used by internal and external users
  - 200+ star and 160+ forks on Github
- V5.0 will include streamBench (by end of 2015)

1.0 (2010)	2.2 (2012)	3.0 (14H2)	4.0 (15H1)
✓ Paper published in ICDE'10 workshops ✓ 3 Categories, 8 workloads ✓ Internal use and shared 3nd. Part under NDA	✓ 2012 open source under Apache 2.0 License ✓ Add Analytical Query Category, 2 hive workloads (join, aggregation)	✓ 2014 Yarn support ✓ Unify MR1 & MR2 to one branch ✓ Add Sleep job ✓ Add concurrent mode	✓ Spark support (scala, java, python) ✓ Unified configuration and reports ✓ Easy visualization report

HiMeter (Dew)

A light weight nonintrusive big data profiler

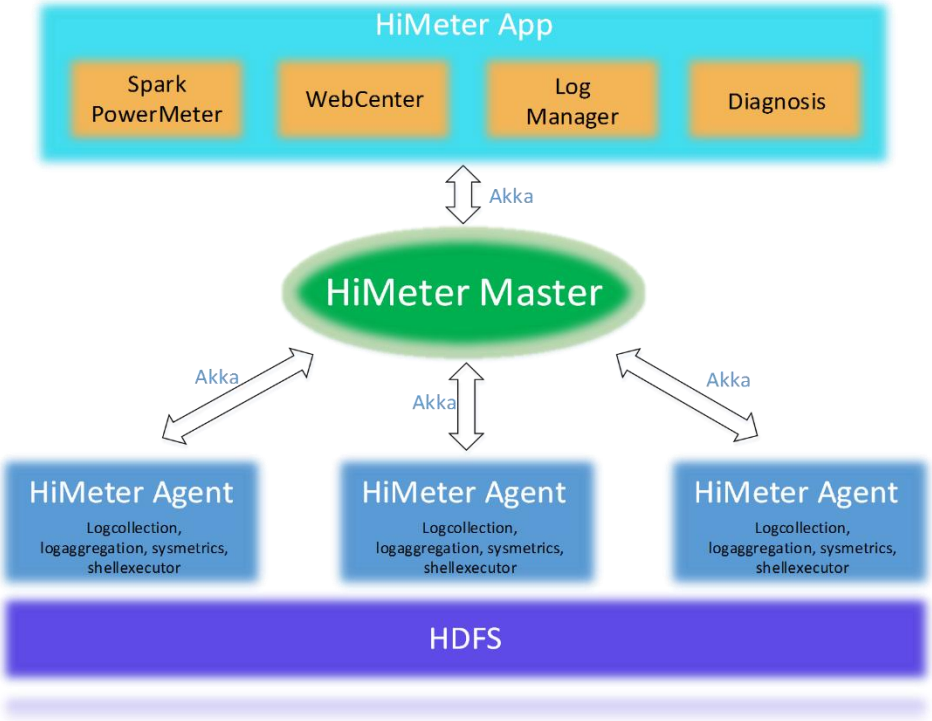
Motivation: provide a tool to profile and tune Spark base cluster performance

Approach

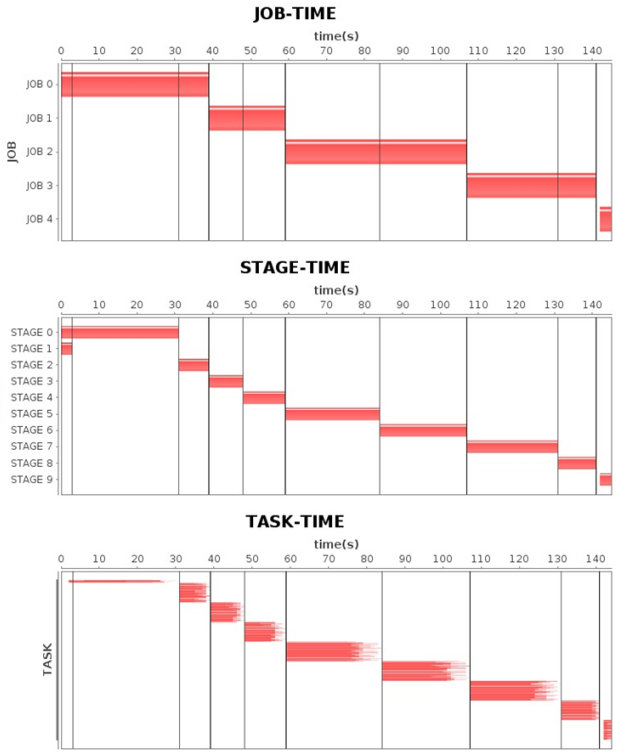
- **Dynamically monitoring big data computing cluster.**
- **Offline analyzing workload performance and giving out performance report and tuning guide**

Philosophy

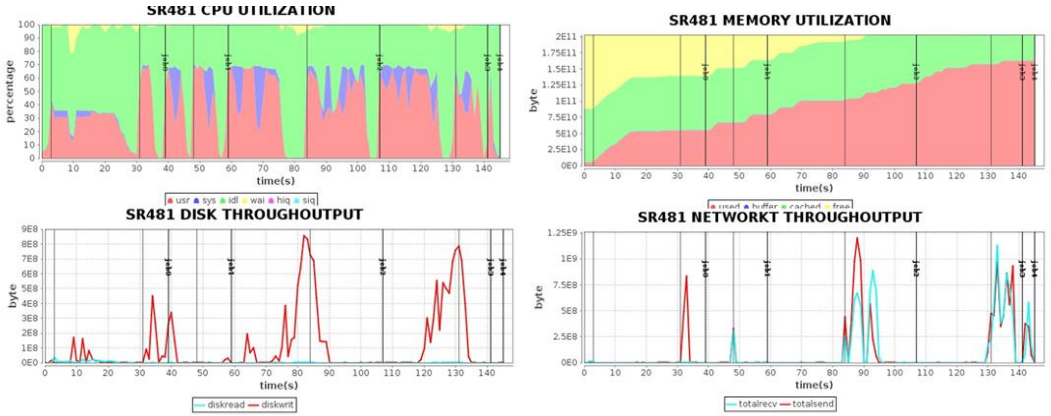
- **Scalable : scale from small to huge cluster.**
- **Light-weight : little impact to the computing cluster**
- **Extensible : pluggable for big data apps**



HiMeter (Dew)



- Spark work flow (Job, Stage, Task)
- System metrics (CPU, Mem, Disk, Network)

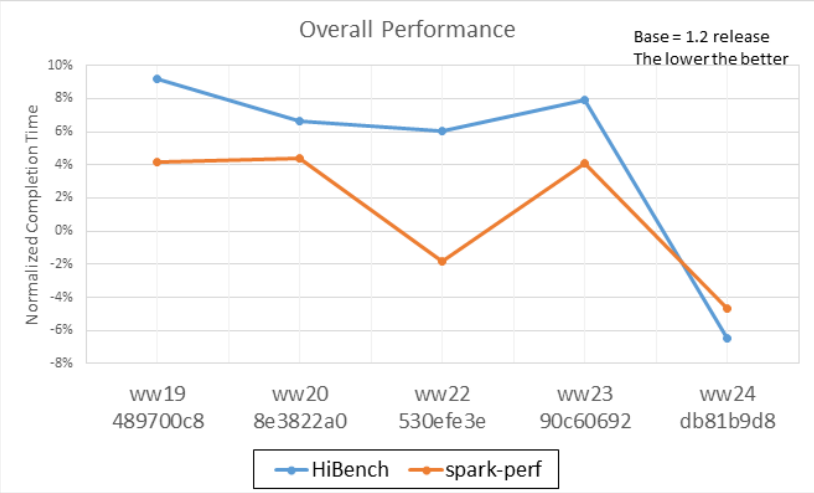


- Smart to provide diagnosis suggestions

hostName	diagnosisName	level	describe	advice
sr481	load-Disk-Read	middle	load-Disk-Read is lower than cluster average by 40.64%	Check the node or your application algorism.
sr479	load-Disk-Read	low	load-Disk-Read is lower than cluster average by 22.43%	Check the node or your application algorism.
sr480	waste-CPU	middle	Cpu resources waste percent is 51.85%. More time on non-computation task.	Improve node's disk and network performance.
sr482	waste-CPU	middle	Cpu resources waste percent is 53.29%. More time on non-computation task.	Improve node's disk and network performance.
sr481	waste-CPU	middle	Cpu resources waste percent is 52.78%. More time on non-computation task.	Improve node's disk and network performance.
sr479	waste-CPU	middle	Cpu resources waste percent is 51.3%. More time on non-computation task.	Improve node's disk and network performance.

Performance Portal for Apache Spark

- Publish Spark performance regularly (weekly)
- W/ workload HiBench & Sparkperf now
- State of art Hardware
  - 1N master, 10N slave cluster w/ 10gb network, each with
    - Intel® Xeon® CPU E5-2697 v2 @ 2.70GHz
    - 128GB RAM + 8 x 1TB SATA HDD



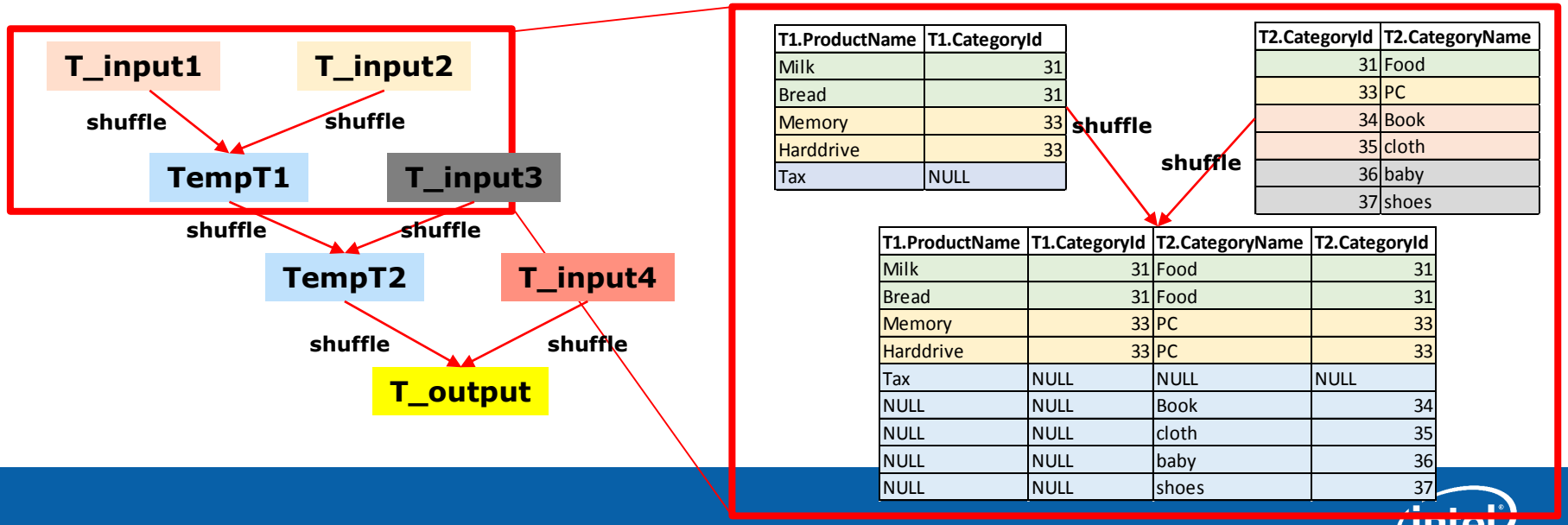
JOB	ww19	ww20	ww22	ww23	ww24
commit	489700c8	8e3822a0	530efe3e	90c60692	db81b9d8
sleep	%	%	-2%	-3%	-4%
wordcount	18%	11%	8%	8%	-19%
kmeans	92%	62%	72%	93%	87%
scan	-5%	-7%	%	-1%	-26%
bayes	-24%	-20%	-18%	-11%	-30%
aggregation	6%	11%	%	9%	-15%
join	5%	1%	%	1%	-13%
sort	-3%	-1%	-12%	-13%	-18%
pagerank	2%	3%	4%	3%	-11%
terasort	-7%	0%	-10%	-7%	-17%

Subscribe @https://lists.01.org/mailman/listinfo/sparkscore
Home page @http://01org.github.io/sparkscore

intel Software

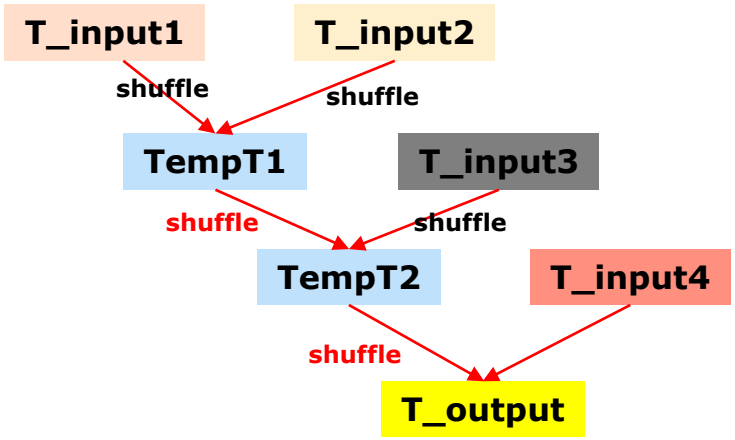
User Case one
Handle multiple tables join better

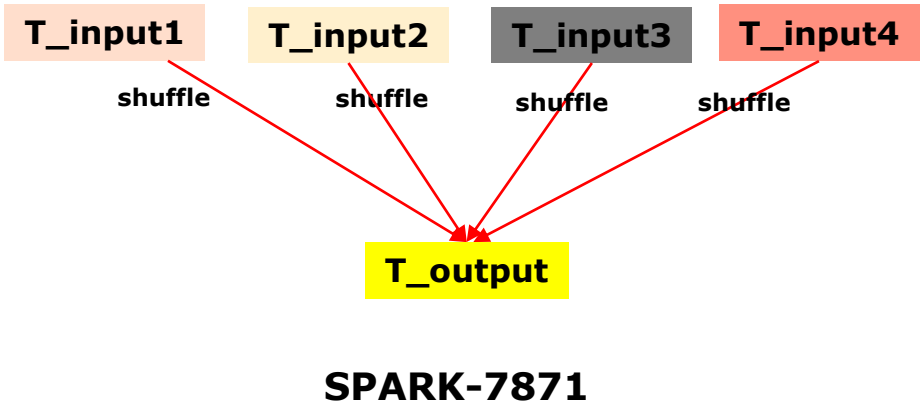
- It is quite common to see complex query in real work cases
  - E.g., Multiple tables join(full outer) on the same key
- Problem statement:
  - It causes large intermediate data with noteworthy skew
  - Low efficiency while involving multiple shuffle phases



User Case one
Handle multiple tables join better

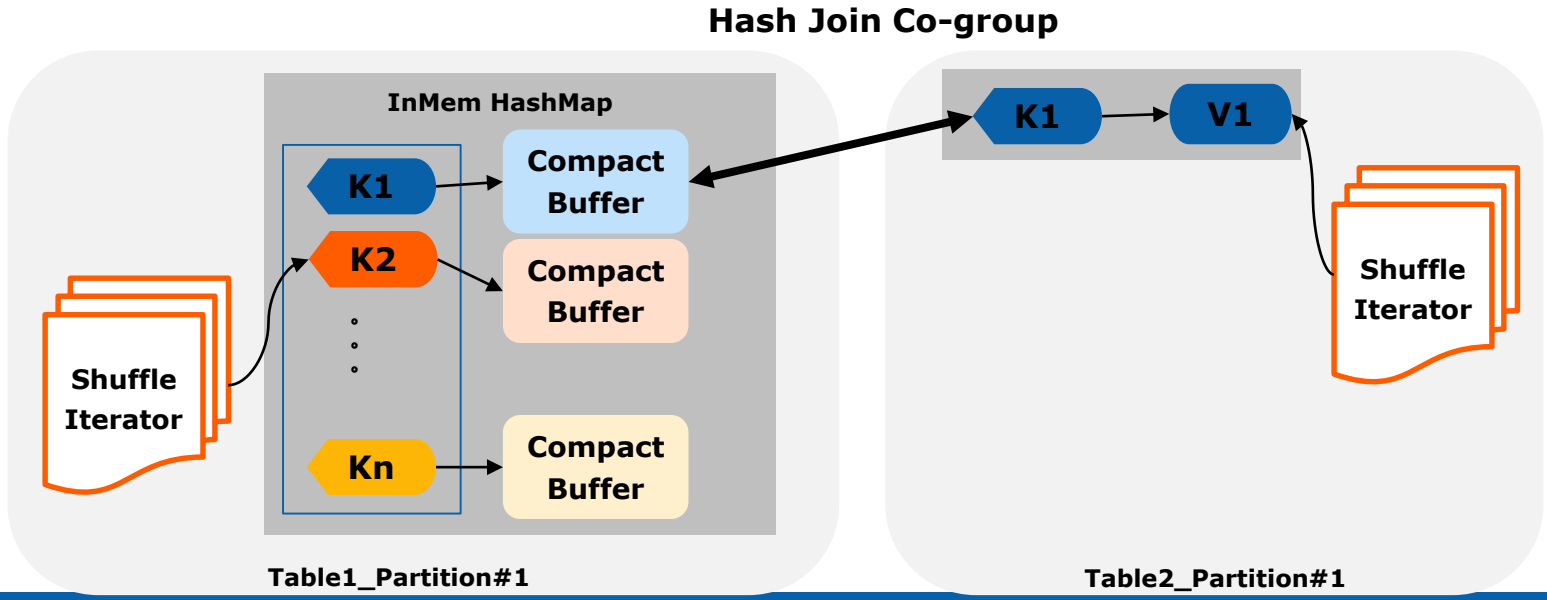
- To combine multiple shuffle outputs into one single stage [SPARK-7871]
  - Avoid data skew which may be accumulated by previous full outer join outputs
  - Save unnecessary shuffle costs
- Make job done and with 2x speedup vs. Hive.





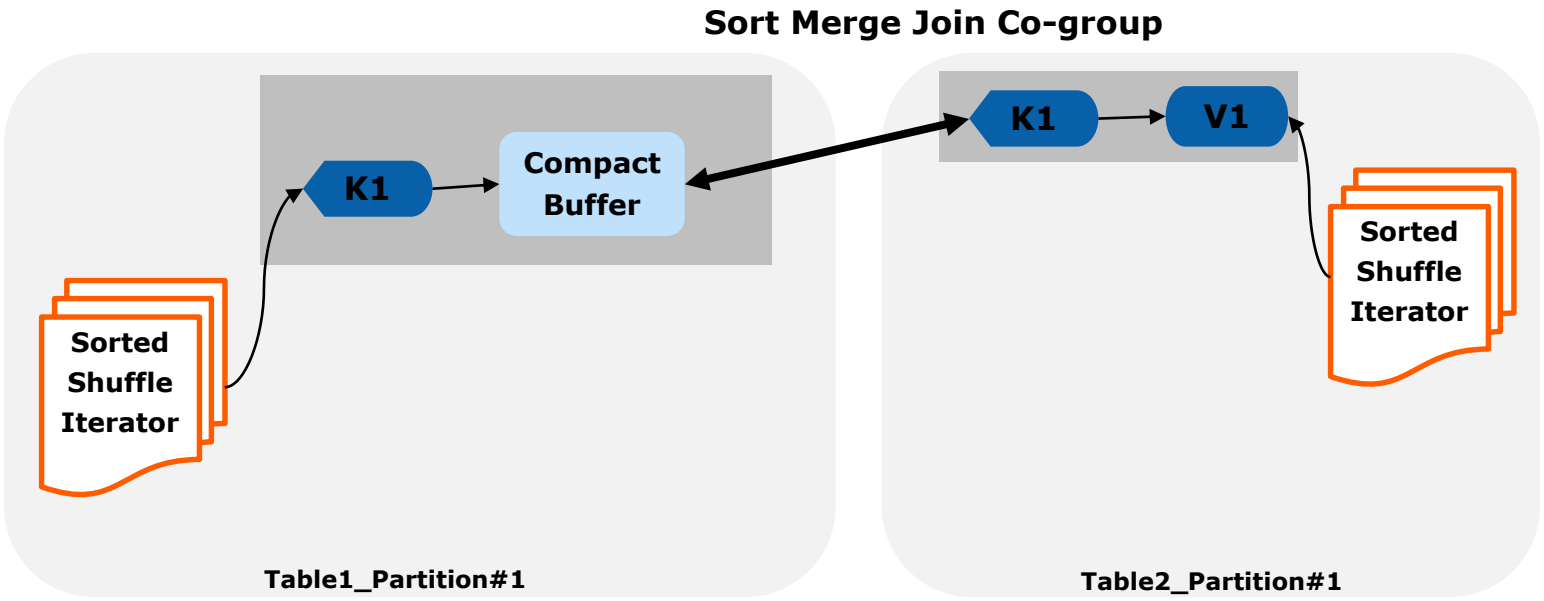
User Case two
SMJ to save more memory

- Problem statement
  - Join with large tables takes really a long time(GC or OOM) & quite difficult for end user to guess the partition number
  - Increasing partition number doesn't solve that due to data skew in most of real world cases.



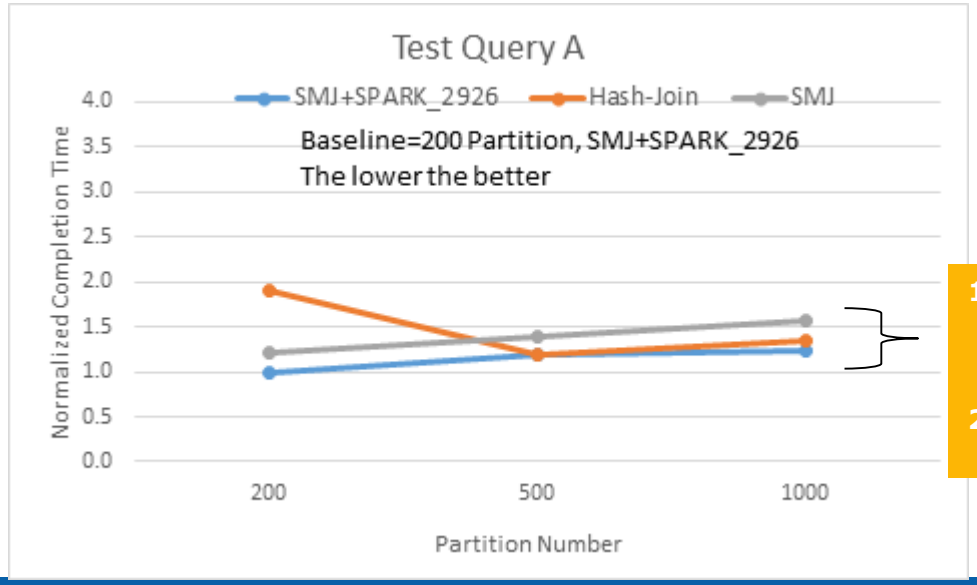
User Case two
SMJ to save more memory

- Sort Merge Join (SPARK-2213, SPARK-7165)
- Much lower memory pressure



User Case two
SMJ to save more memory

- By using reduce size sort based shuffle, it improves the SMJ performance by 20%
- Significantly reduced GC time according to the customers' observation



1. **SMJ W/ SPARK_2926 performs quite close to Hash Join (in-mem)**
2. **W/ SPARK_2926 is 20% lower than SMJ W/O it.**

User Case Three
Manage the memory in a smart way

- Commonly use Bagel/GraphX for graph analytics
  - The present iteration only depends on its previous step, I.e., RDD[n] is only used in RDD[n+1] computation
- Problem statement:
  - Memory space is continuously increased in Bagel app

Iteration	Cache Size/iteration	Total Cached Size (before optimize)
Initial	4.3G	4.3G
1	8.2G	12.5G
2	98.8G	111.3G
3	90.8G	202.1G

User Case Two
Manage the memory in a smart way

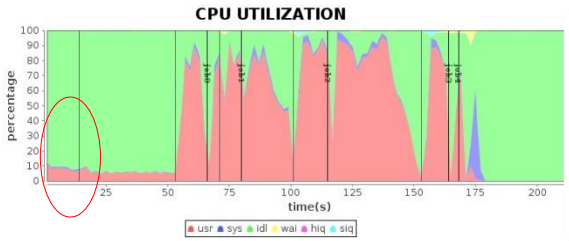
- Free those obsolete RDDs not be used anymore
  - I.e., To un-persist RDD[n-1] after RDD[n] is done SPARK-2661
- The total memory usage is > 50% off

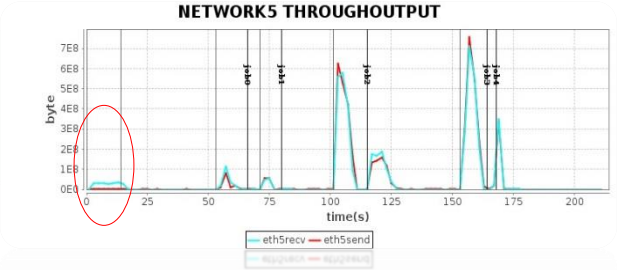
Iteration	Cache Size/iteration	Total Cached Size (before optimize)	Total Cached Size (after optimize)
Initial	4.3G	4.3G	4.3G
1	8.2G	12.5G	8.2G
2	98.8G	111.3G	98.8G
3	90.8G	202.1G	90.8G

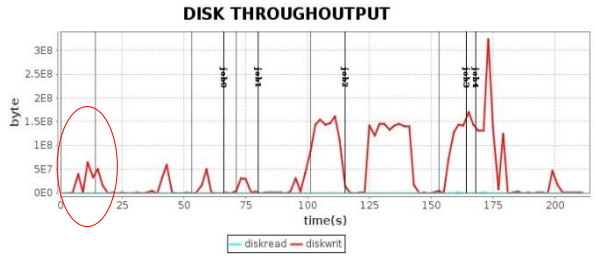
See more tuning @https://databricks.com/blog/2015/05/28/tuning-java-garbage-collection-for-spark-applications.html

User Case Three
Save the IO bandwidth

- Mostly to run Spark on Yarn in data center
- 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

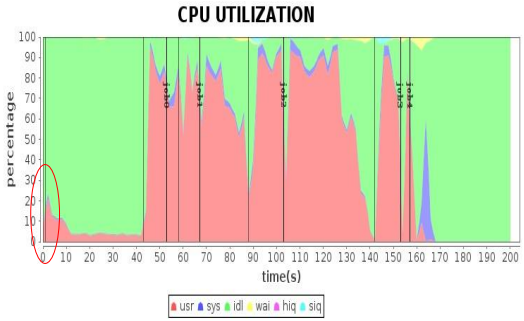


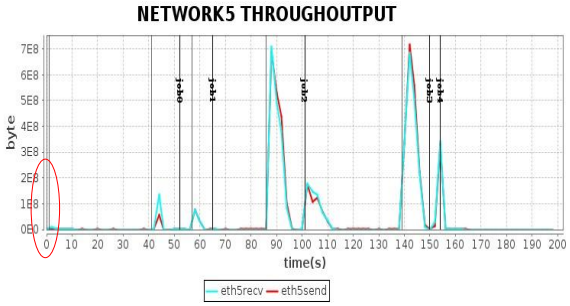


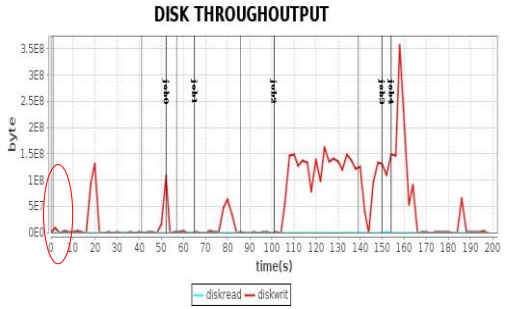


User Case Three
Save the IO bandwidth

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







Summary

- Spark team inside Intel
- Scalability, reliability and stability @ spark
- projects we did to solve the problem
- Working with partners together to improve spark performance
- Intel wants to work with industry and community to make Spark better

Checking our demo booth for more details

Notices and Disclaimers

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.
  A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.
- Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.
- The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.
- Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm
- Intel, the Intel logo, Intel Xeon, and Xeon logos are trademarks of Intel Corporation in the U.S. and/or other countries.
- Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: Go to: Learn About Intel® Processor Numbers http://www.intel.com/products/processor_number
- All the performance data are collected from our internal testing. Some results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.
- *Other names and brands may be claimed as the property of others.
-