Flash Memory Summit

Using an In-Memory Data Accelerator to Improve Cloud Analytics

Jian Zhang, jian.zhang@intel.com

August, 2019

Agenda

- **Background and motivation**
- Bigdata analytics on the cloud: the challenges & optimizations
- Accelerate bigdata analytics on cloud with in memory data accelerator (IMDA)
  - IMDA as Cache
  - IMDA as shuffle
- Summary

Challenges of scaling Hadoop* Storage

BOUNDED Storage and Compute resources on Hadoop Nodes brings challenges

Data Capacity

Silos

Costs

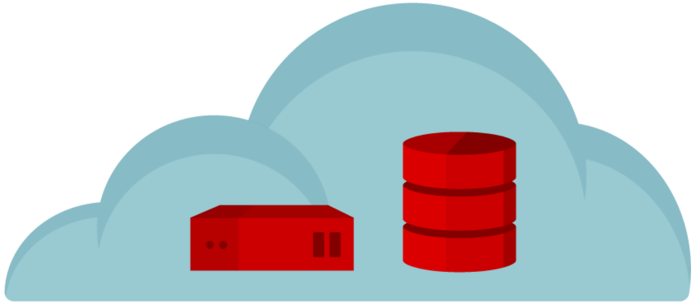
Performance
& efficiency

Typical Challenges

Data/Capacity	Multiple Storage Silos
Space, Power, Utilization	Inadequate Performance
Upgrade Cost	Provisioning and Configuration

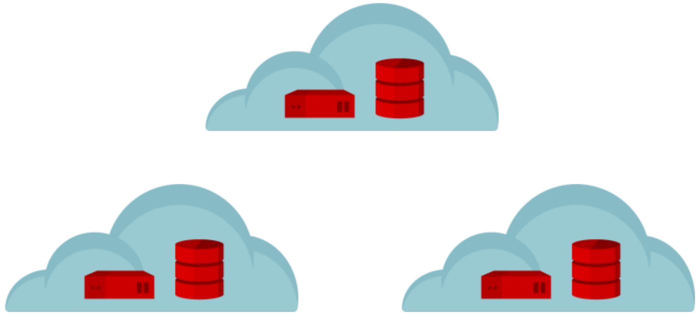
*Other names and brands may be claimed as the property of others.

Discontinuity in bigdata infrastructure makes different solution



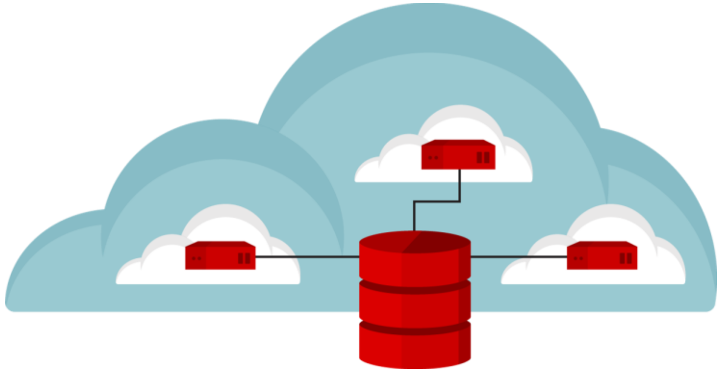
SINGLE LARGE CLUSTER

Get a bigger cluster
for many teams to share.



MULTIPLE SMALL CLUSTERS

Give each team their own dedicated cluster, each with a copy of PBs of data.

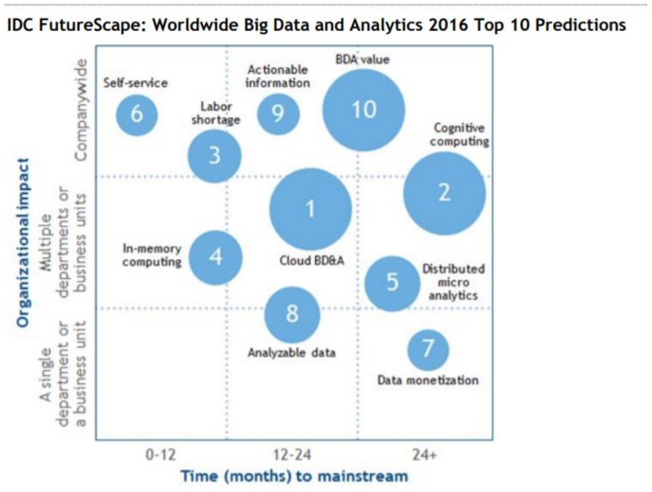


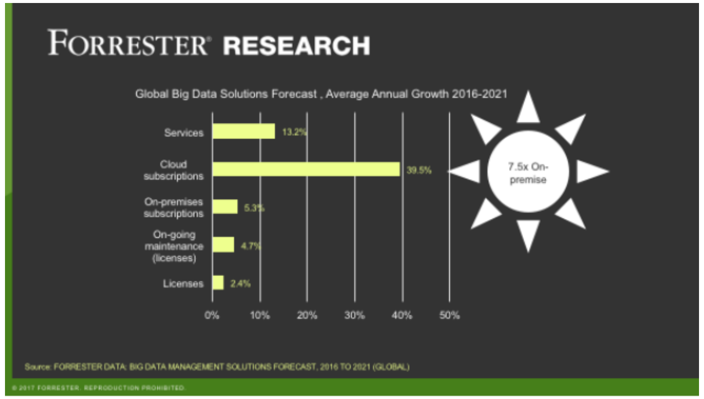
ON DEMAND ANALYTIC CLUSTERS

Give teams ability to spin-up/spin-down clusters which can share data sets.

Cloud based Bigdata Analytics Market Trend

- IDC No.1 Big Data and analytics predictions
  - Through 2020, spending on cloud-based BDA technology will grow 4.5x faster than spending for on-premises solutions [1]
- FORRESTER: Public cloud adoption is the No. 1 priority for technology decision makers investing in big data.[2]
- Cloud-based big data services offer all the same benefits associated with other public cloud services.





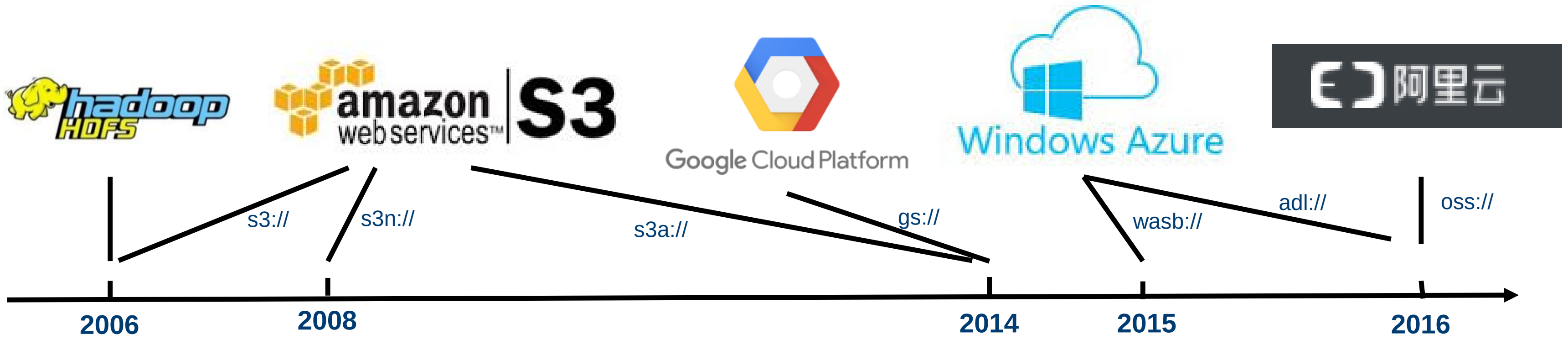
Source: **IDC FutureScape: Worldwide Big Data and Analytics 2016 Predictions**
Source: https://www.oracle.com/webfolder/s/delivery_production/docs/FY16h1/do

Benefits of bigdata analytics on the cloud

Independent scale of compute and storage	Single copy of data	Agile application development	Hybrid cloud deployment	Simple and flexible software management
• Rightsized HW for each layer • Reduce resource wastage • Cost saving	• Multiple compute cluster share common data repo/lake • Simplified data management • Reduced provisioning overhead • Improve security	• In-memory cloning • Snapshot service • Quick & efficient copies	• Mix and match resources depending on workload nature and life cycle	• Avoid software version management • Upgrade compute software only

Bigdata analytics on the cloud ecosystem

Hadoop **C**ompatible **F**ile **S**ystem abstraction layer: Unified storage API interface Hadoop fs –ls s3a://job/

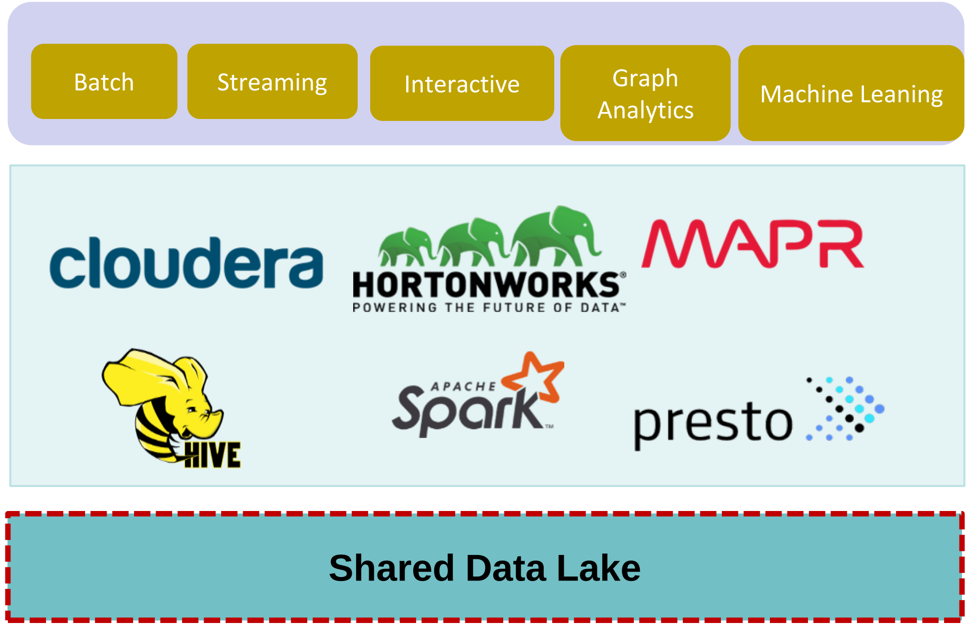


Agenda

- Background and motivation
- **Bigdata analytics on the cloud: the challenges & optimizations**
- Accelerate bigdata analytics on cloud with in memory data accelerator (IMDA)
  - IMDA as Cache
  - IMDA as shuffle
- Summary

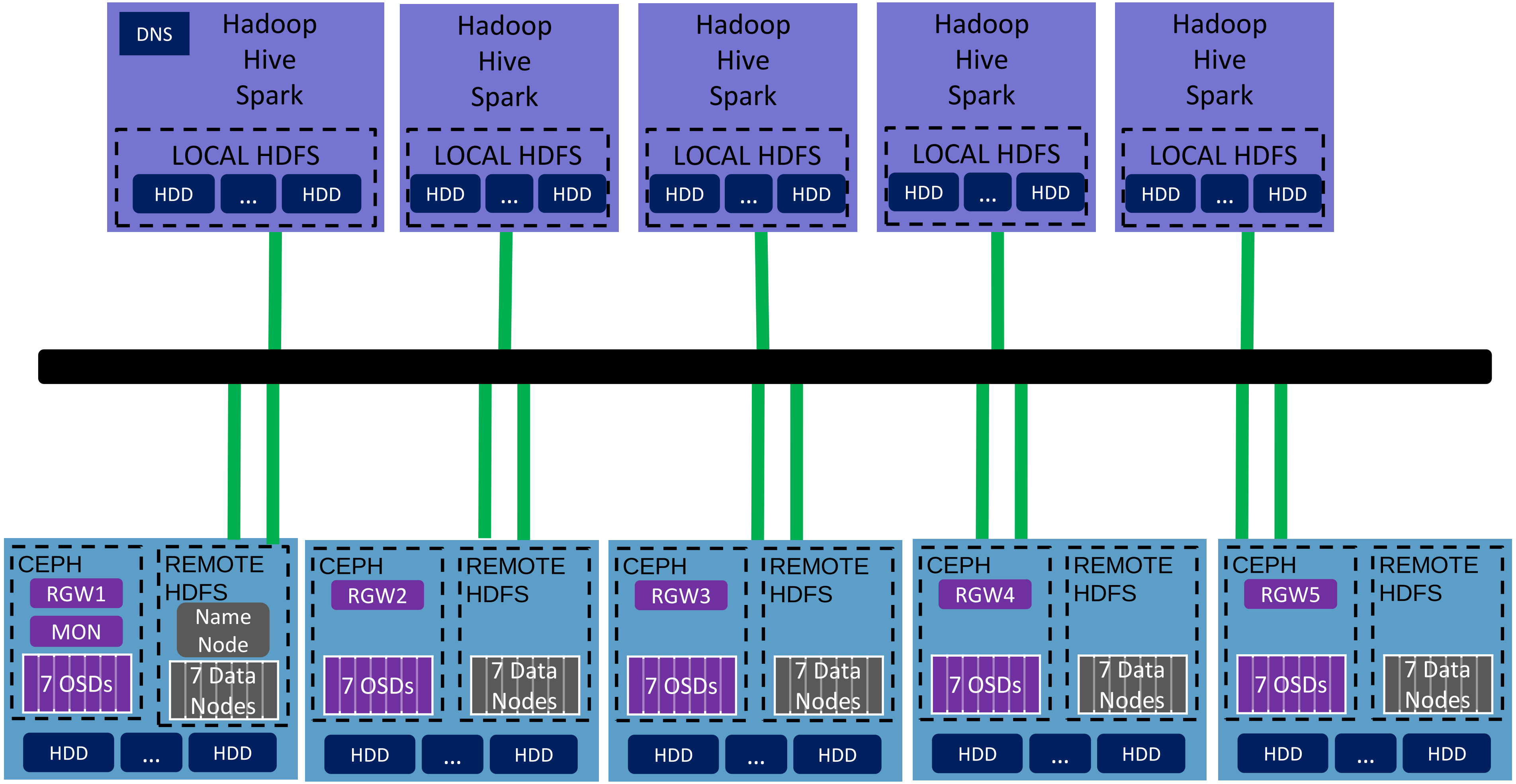
Performance Gap

Architectures – Storage Disaggregation



Replace HDFS with Shared data lake

Performance gaps: System configurations



5x Compute Node

Hardware:

- ntel® Xeon™ processor Gold 6140 @ 2.3GHz, 384GB Memory
- 1x 82599 10Gb NIC
- 5x P4500 SSD (2 for spark-shuffle)

Software:

- Hadoop 2.8.1
- Spark 2.2.0
- Hive 2.2.1
- RHEL7.3

5x Storage Node

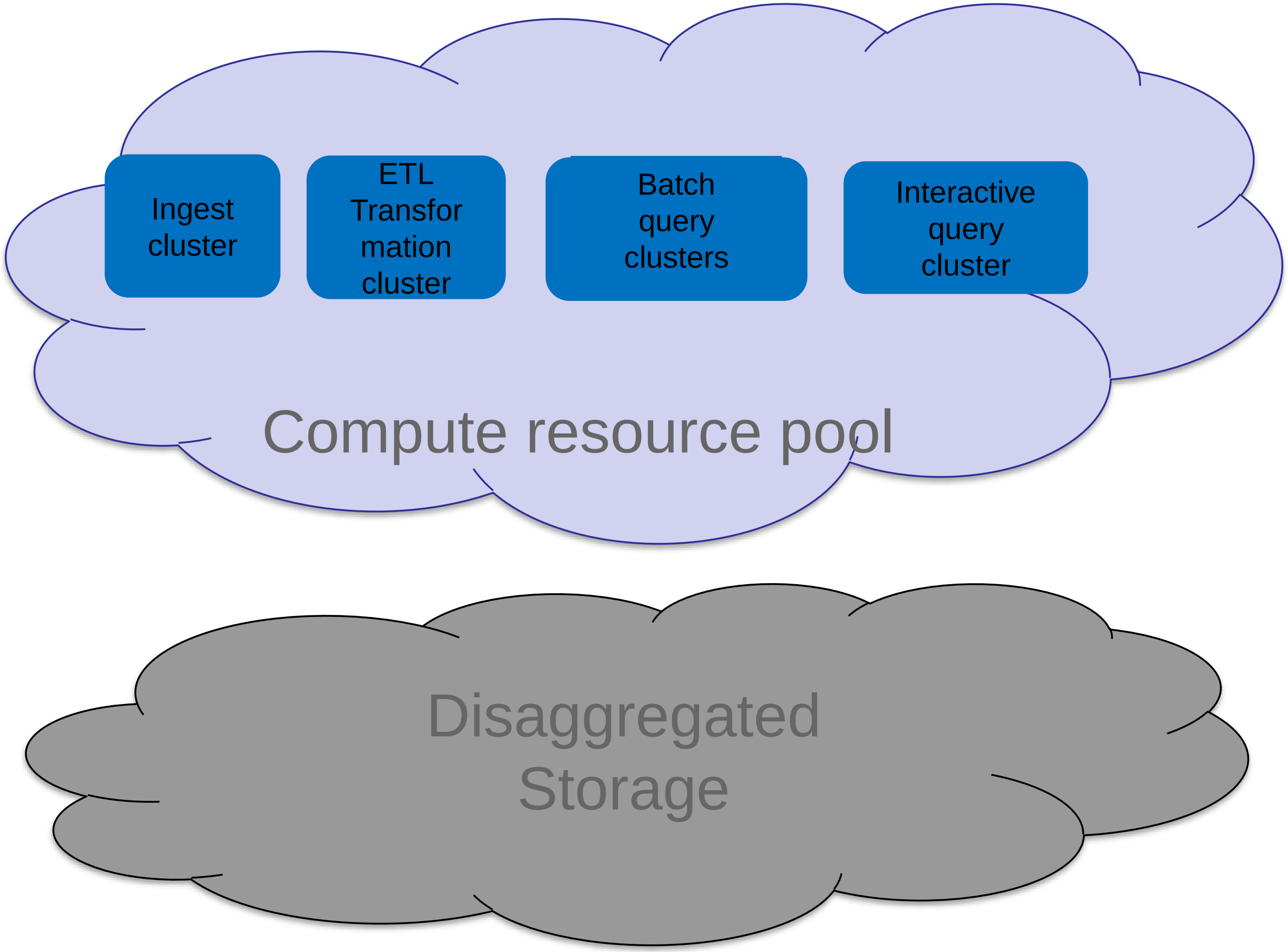
- Intel(R) Xeon(R) CPU Gold 6140 @ 2.30GHz, 192GB Memory
- 2x 82599 10Gb NIC
- 7x 1TB HDD for Ceph bluestore or HDFS namenode and datanode

Software:

- Hadoop 2.8.1
- Ceph 12.2.7
- RHEL7.3

*Other names and brands may be claimed as the property of others.

Performance gaps: usage cases



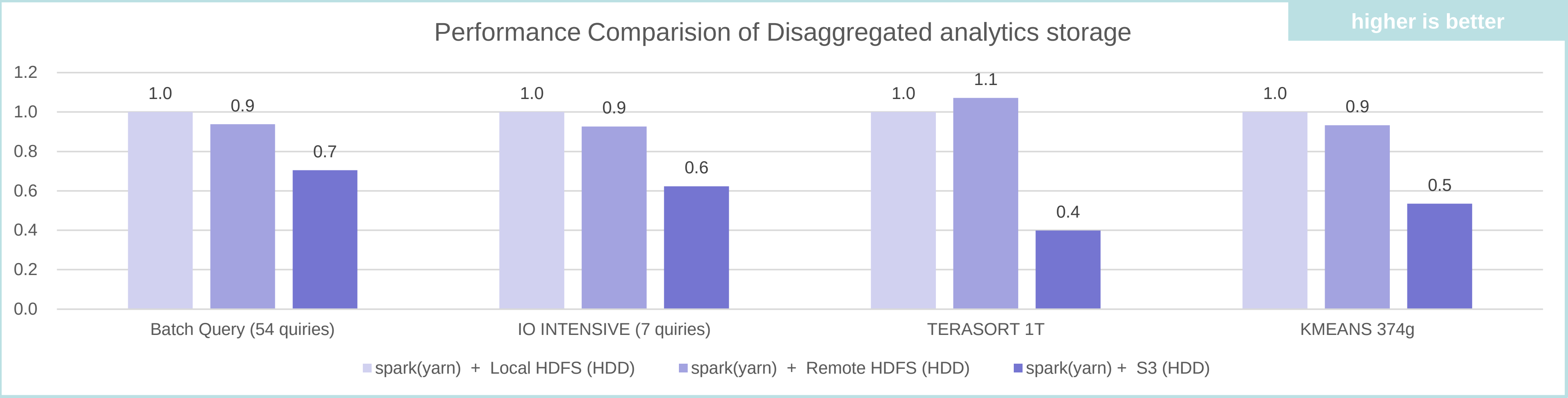
- **Simple Read/Write**
  - **Terasort**: a popular benchmark that measures the amount of time to sort one terabyte of randomly distributed data on a given computer system.
- **TPC-DS derived tests:**
- **Batch Analytics**
  - To consistently executing analytical process to process large set of data.
  - **UC11:** Leveraging 54 derived from TPC-DS * queries with intensive reads across objects in different buckets
  - **I/O intensive queries:** selected 9 I/O intensive queries from TPC-DS

Kmeans

- K-means is one of the most commonly used clustering algorithms that clusters the data points into a predefined number of clusters.

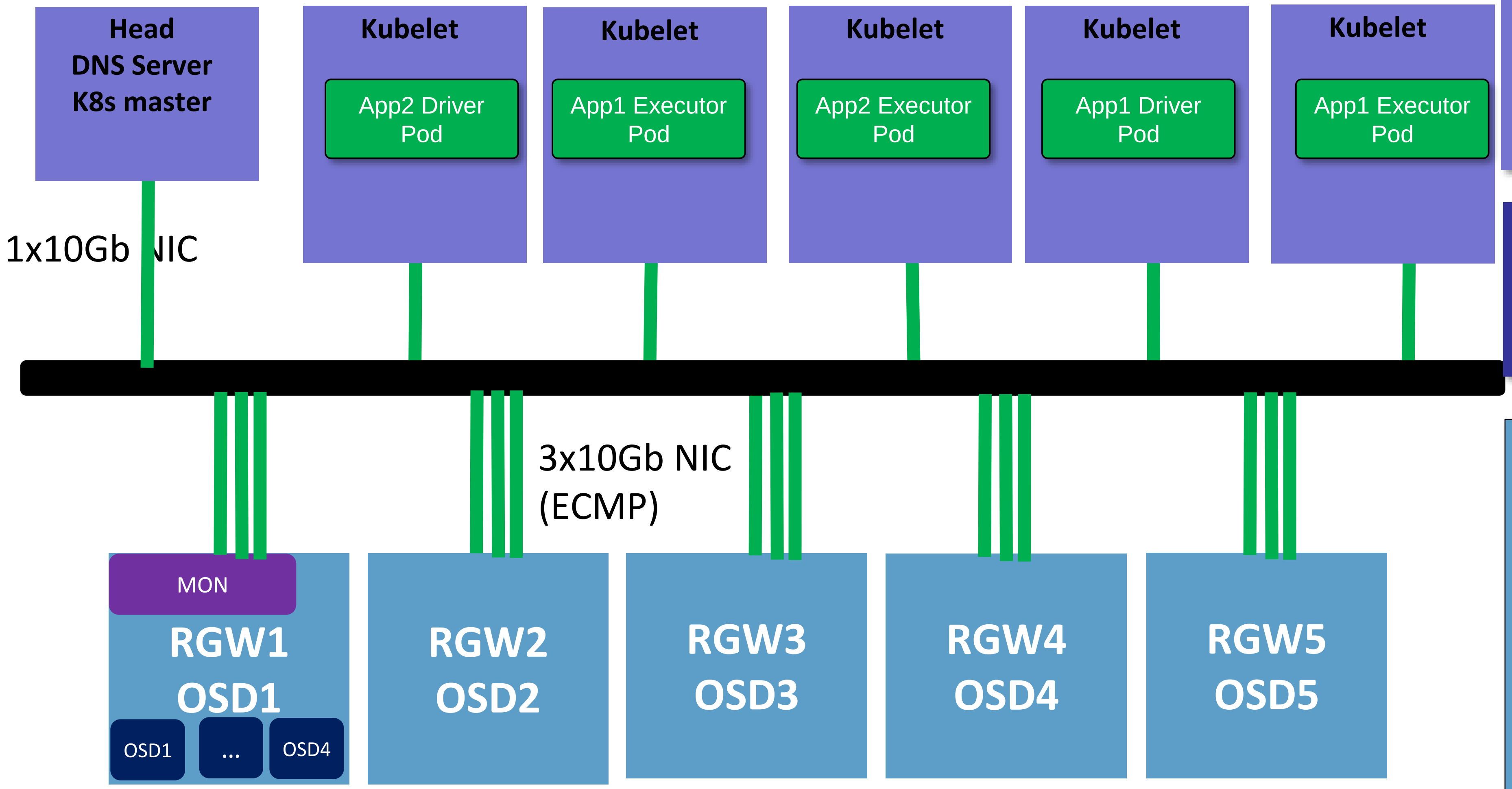
Flash Memory Summit

Performance gaps



- Storage disaggregation leads to performance regression
  - Up to **10%** for remote HDFS, Terasort performance is higher as usable memory increased
  - Up to **60%** for S3 object storage (optimized results, up to 11.5x perf. boost through tunings compared with default parameters)
- One important cause for the performance gap: s3a does not support Transactional Writes
  - Most of bigdata software (Spark, Hive) relies on HDFS's atomic rename feature to support atomic writes
  - During job submit, commit protocol is used to specify how results should be written at the end of job
  - First stage task output into temporary locations, and only moving (renaming) data to final location upon task or job completion
  - S3a implements this with: COPY+DELETE+HEAD+POST

Serverless architecture: configuration



5x Compute Node

- Intel® Xeon™ processor E5-2699 v4 @ 2.2GHz, 128GB mem
- 2x10G 82599 10Gb NIC
- 2x SSDs
- 3x Data storage (can be emliminated)

Software:

- Hadoop 2.8.1
- Spark 2.2.0
- Hive 2.2.1
- Presto 0.177
- CentOS 7.5

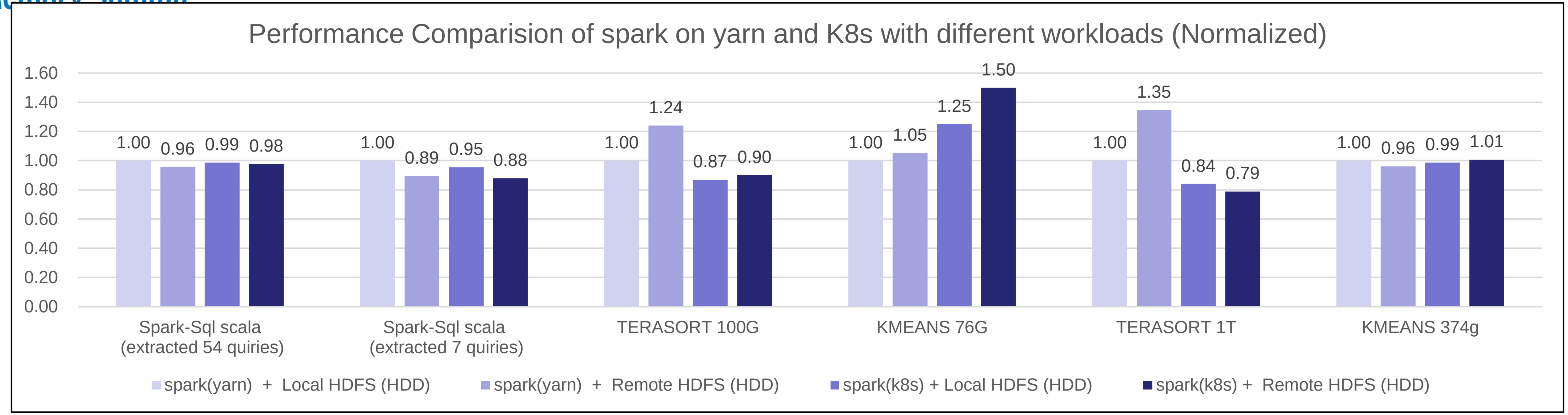
Compute Orchestration

- K8s 1.11

5x Storage Node, 5 RGW nodes(co-located)

- Intel(R) Xeon(R) CPU E5-2699v4 2.20GHz
- 128GB Memory
- 3x 82599 10Gb NIC
- 1x Intel® P3700 1.0TB SSD as joural
- 4x 1.6TB Intel® SSD DC P3520 as data drive
- 1 OSD instances one each P3520 SSD
- CentOS 7.5
- Ceph Jewel

Serverless analytics Performance



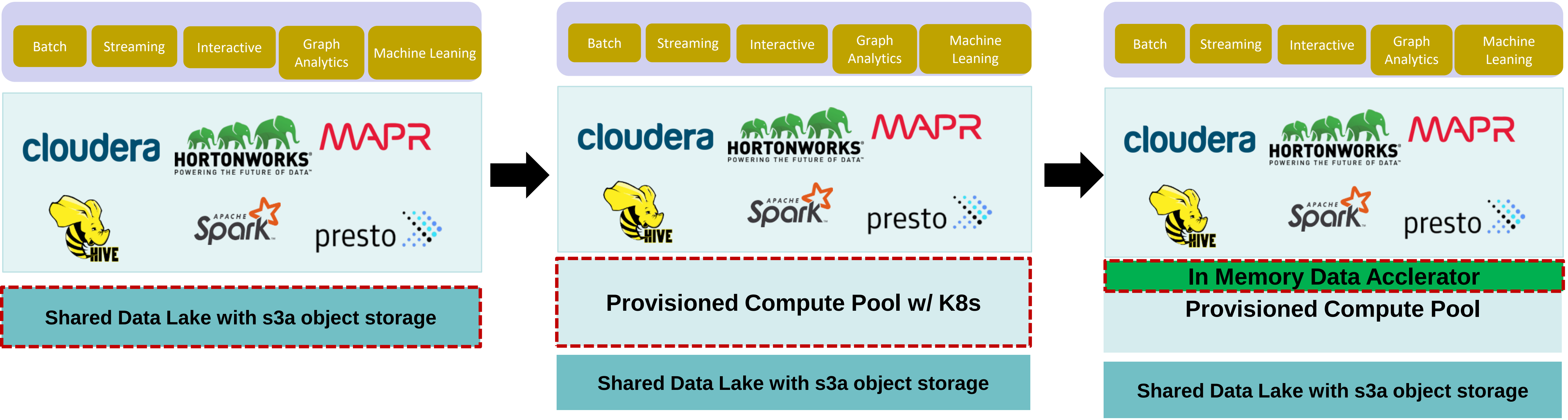
- Spark on kubernetes delivers similar performance compared with spark on yarn

Running Compute Services in K8s brings littlie performance impact for typical SQL workloads

Agenda

- Background and motivation
- Bigdata analytics on the cloud: the challenges & optimizations
- **Accelerate bigdata analytics on cloud with in memory data accelerator (IMDA)**
  - IMDA as Cache
  - IMDA as shuffle
- Summary

Architecture – IN Memory data accelerator

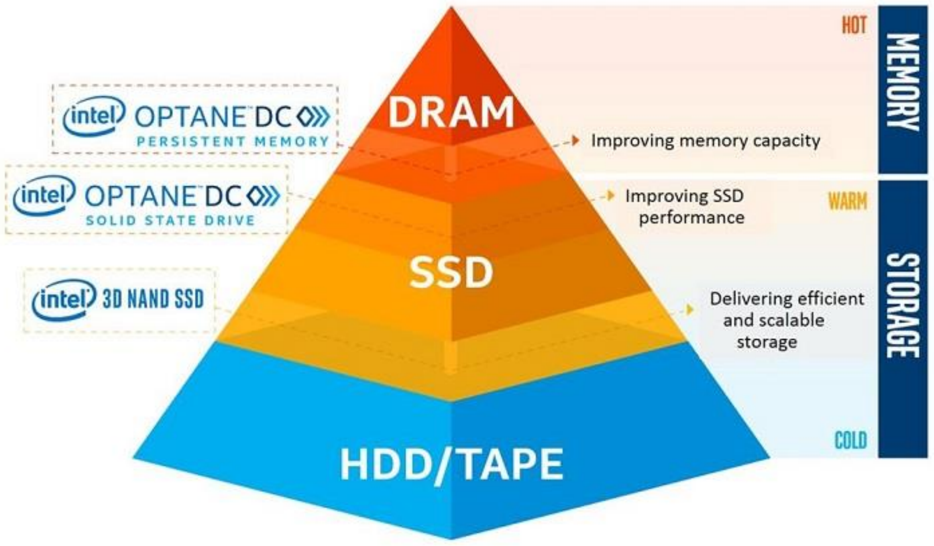


Replace HDFS with disaggregated s3 object storage

Compute services in Kurbernetes

In Memory Data Acclerator

Persistent Memory and RDMA



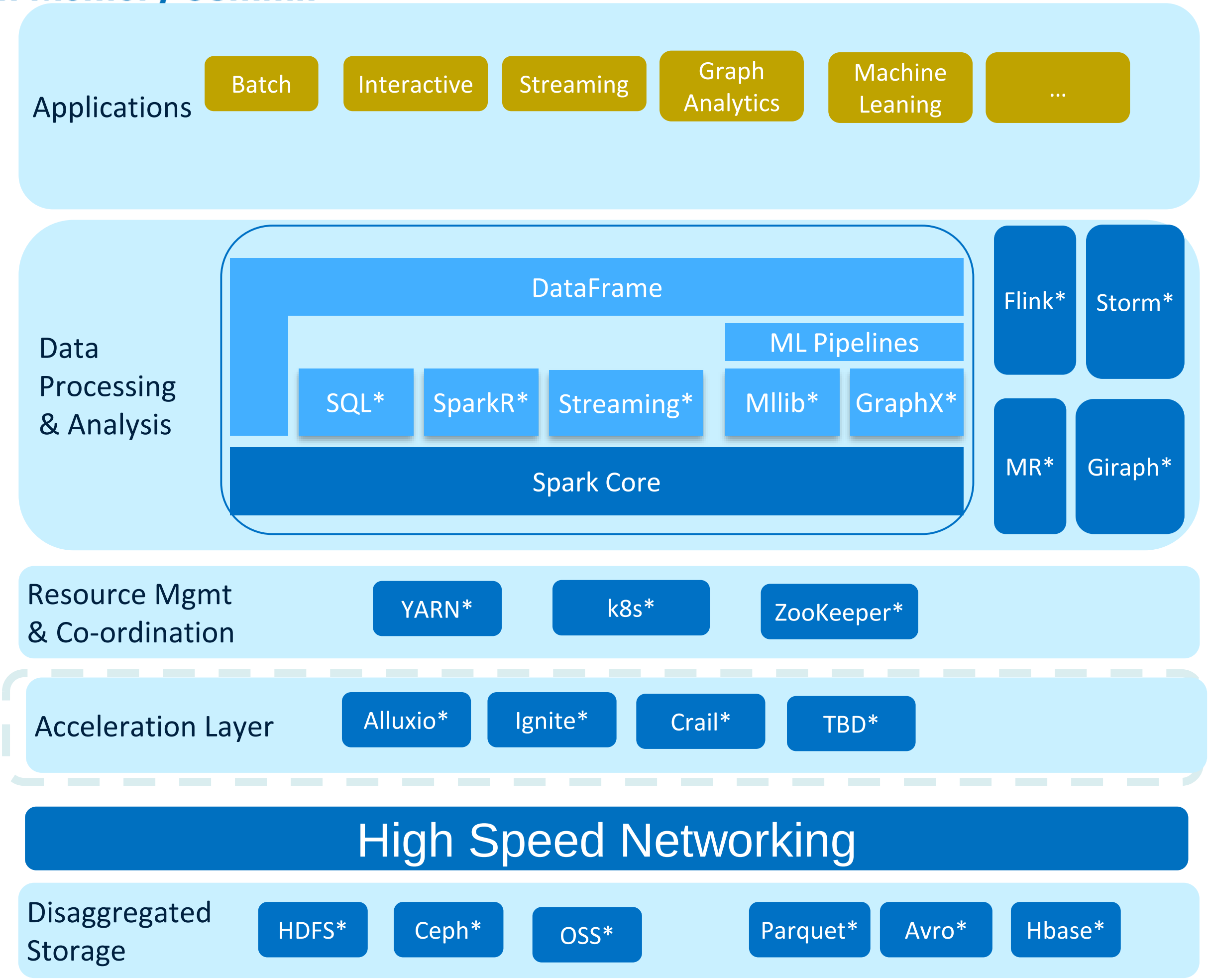
- **Persistent Memory:**
- PMEM represents a new class of memory and storage technology architected specifically for data center usage
- Combination of high-capacity, affordability and persistence.

RDMA: Remote Direct Memory Access

- Accessing (i.e. reading from or writing to) memory on a remote machine without interrupting the processing of the CPU(s) on that system.
  - Zero-copy - applications perform data transfer without the network software stack involvement, data is being send received directly to the buffers without being copied between the network layers.
  - Kernel bypass - applications perform data transfer directly from userspace, no context switches.
  - No CPU involvement - applications can access remote memory without consuming any CPU in the remote machine.

Picture source: https://software.intel.com/en-us/blogs/2018/10/30/intel-optane-dc-persistent-memory-a-major-advance-in-memory-and-storage-architecture

Leveraging In memory data accelerator to accelerate intermediate data access

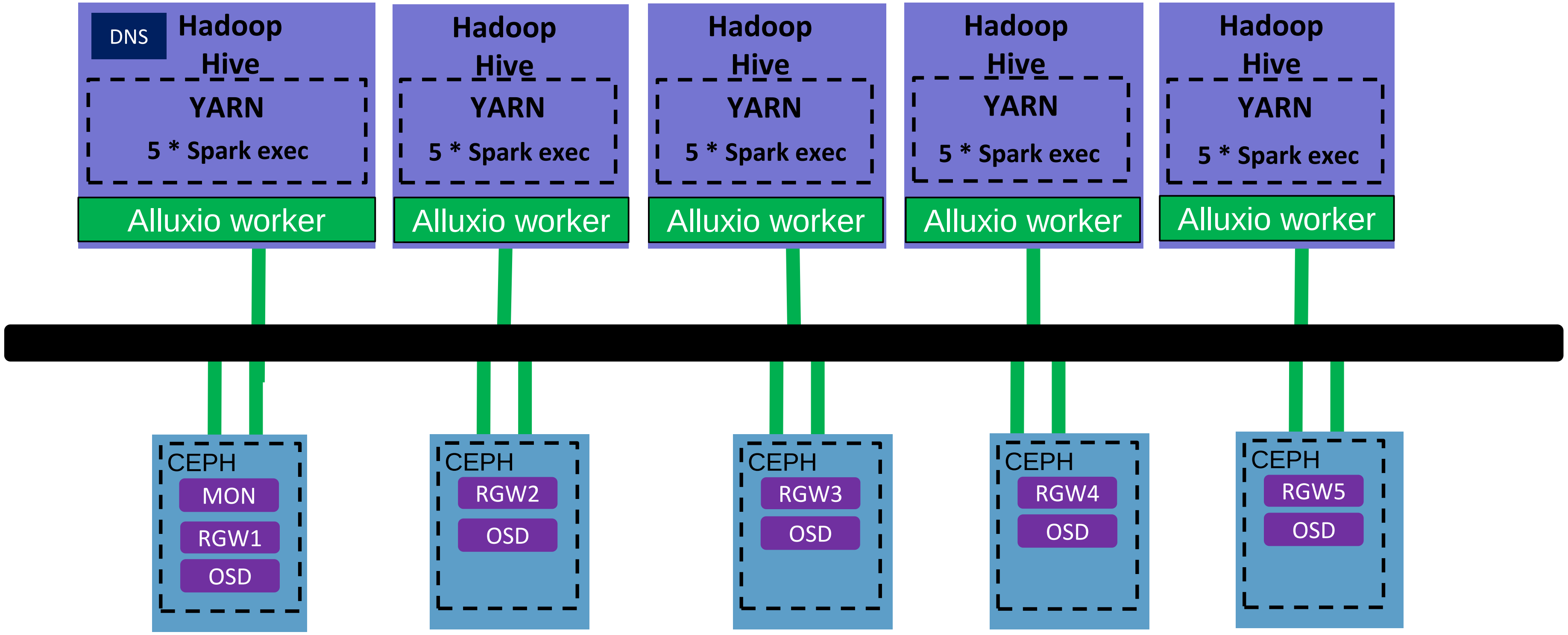


- Leverage new HW technologies & products that delivers significant performance improvement
  - Persistent memory, RDMA, GPU
- Using in memory data accelerator layer to accelerate ephemeral data access
  - Caching hot data in to shorten I/O stack
  - Unifies underlying Filesystem
  - Shuffle/spill to AEP improves latency, reduced GC
  - Columnar format storage optimized for GPU
- It requires a storage and network co-design to fully leverage those technologies or HWs address the bottlenecks
  - Optimized libraries to bypass filesystem, avoid user space/kernel space context switch

Agenda

- Background and motivation
- Bigdata analytics on the cloud: the challenges & optimizations
- Accelerate bigdata analytics on cloud with in memory data accelerator (IMDA)
  - **IMDA as Cache**
  - IMDA as shuffle
- Summary

System configurations



5x Compute Node

Hardware:

- ntel® Xeon™ processor Gold 6140 @ 2.3GHz, 384GB Memory
- 1x 82599 10Gb NIC
- 5x P4500 SSD (2 for spark-shuffle)

Software:

- Hadoop 2.8.1
- Spark 2.2.0
- Hive 2.2.1
- RHEL7.3
- Alluxio: 2.0.0, 200GB DRAM Cache

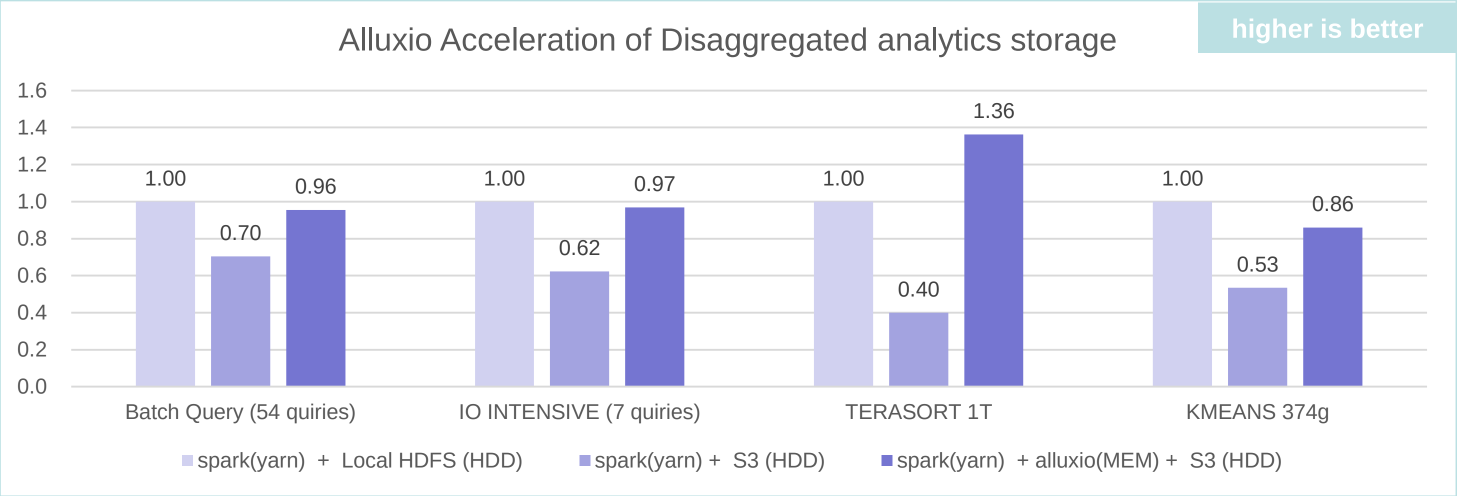
5x Storage Node

- Intel(R) Xeon(R) CPU Gold 6140 @ 2.30GHz, 192GB Memory
- 2x 82599 10Gb NIC
- 7x 1TB HDD for Ceph bluestore or HDFS namenode and datanode

Software:

- Hadoop 2.8.1
- Ceph 12.2.7
- RHEL7.3

Performance overview



Using Alluxio IMDA as cache:

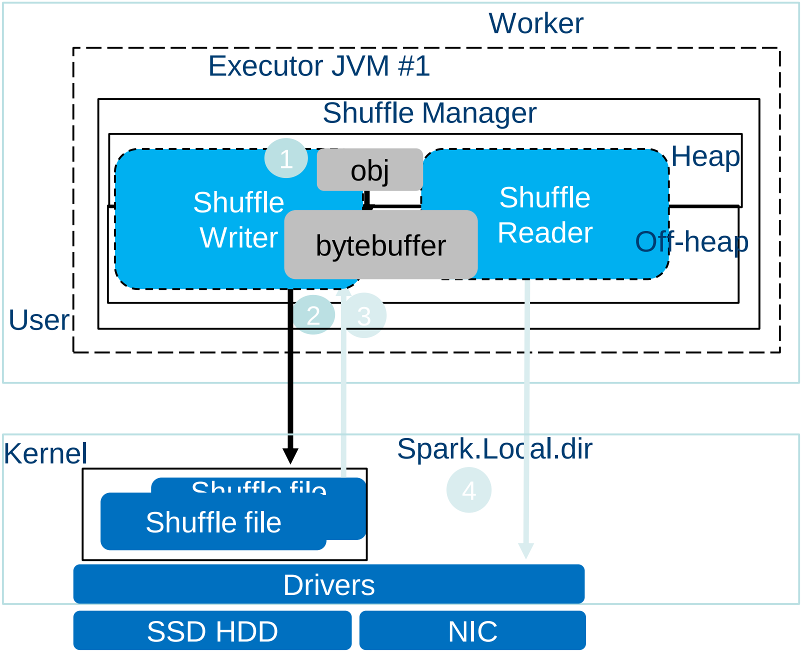
- For terasort, **3.4x** speedup over S3 object storage, **1.36x** speedup over local HDFS.
- For TPCDS test, up to **1.56x** performance speedup for IO intensive queries, slightly lower than local HDFS.
- For KMeans test, **1.62x** speedup over S3 object storage, 14% lower compared with local HDFS.
  - KMeans is a CPU intensive workload

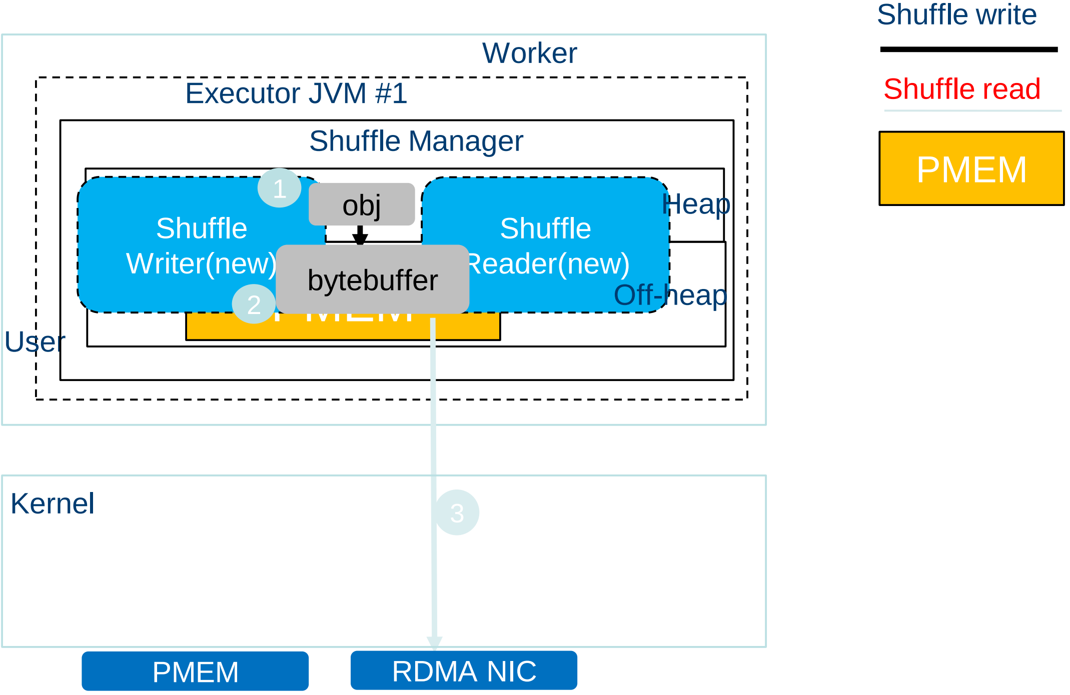
Using Alluxio IMDA cache improved in IO intensive workloads but remains headroom in other cases.

Agenda

- Background and motivation
- Bigdata analytics on the cloud: the challenges & optimizations
- Accelerate bigdata analytics on cloud with in memory data accelerator (IMDA)
  - IMDA as Cache
  - **IMDA as shuffle**
- Summary

Spark-PMoF Design





- 1. Serialize obj to off-heap memory
- 2. Write to local shuffle dir
- 3. Read from local shuffle dir
- 4. Send to remote reader through TCP-IP

➢ Lots of context switch
➢ POSIX buffered read/write on shuffle disk
➢ TCP/IP based socket send for remote shuffle read

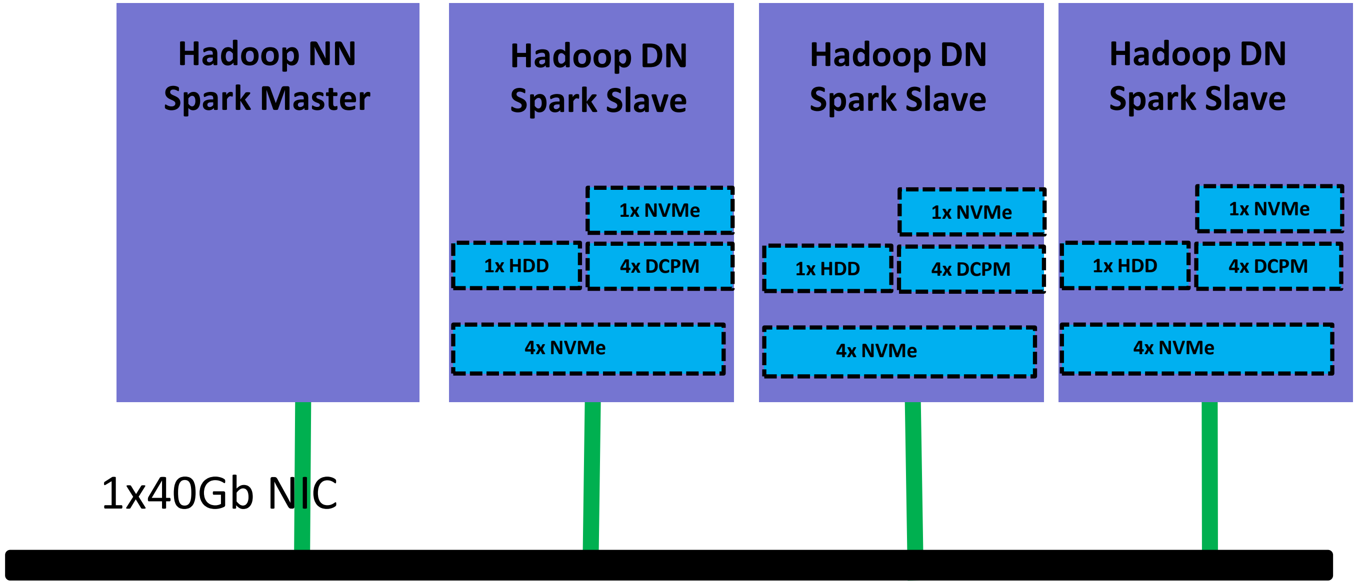
1. Serialize obj to off-heap memory
2. Persistent to PMEM
3. Read from remote PMEM through RDMA, PMEM is used as RDMA memory buffer

➢ No context switch
➢ Efficient read/write on PMEM
➢ RDMA read for remote shuffle read based on HPNL

Spark PMoF: https://github.com/intel-bigdata/spark-pmof

HPNL: https://github.com/intel-bigdata/hpnl

Benchmark configuration



3 Node cluster

Hardware:

- Intel® Xeon™ processor Gold 6140 CPU @ 2.30GHz, 384GB Memory
- 1x Mellanox ConnectX-4 40Gb NIC
- Shuffle Devices :
  - 1x 1T HDD/NVMe for shuffle
  - 4x 256GB DCPM for shuffle
- 4x 1T NVMe for HDFS

Software:

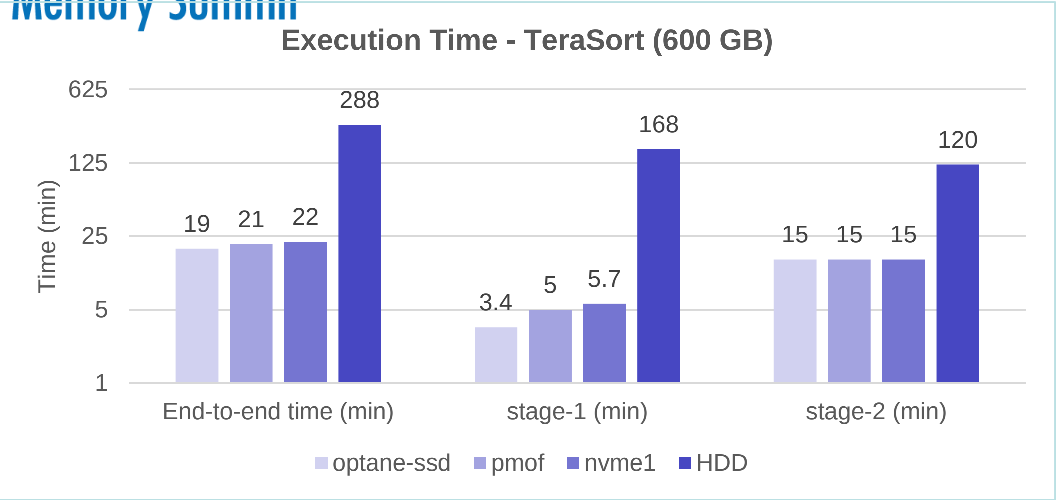
- Hadoop 2.7
- Spark 2.3
- Fedora 27 with WW26 BKC

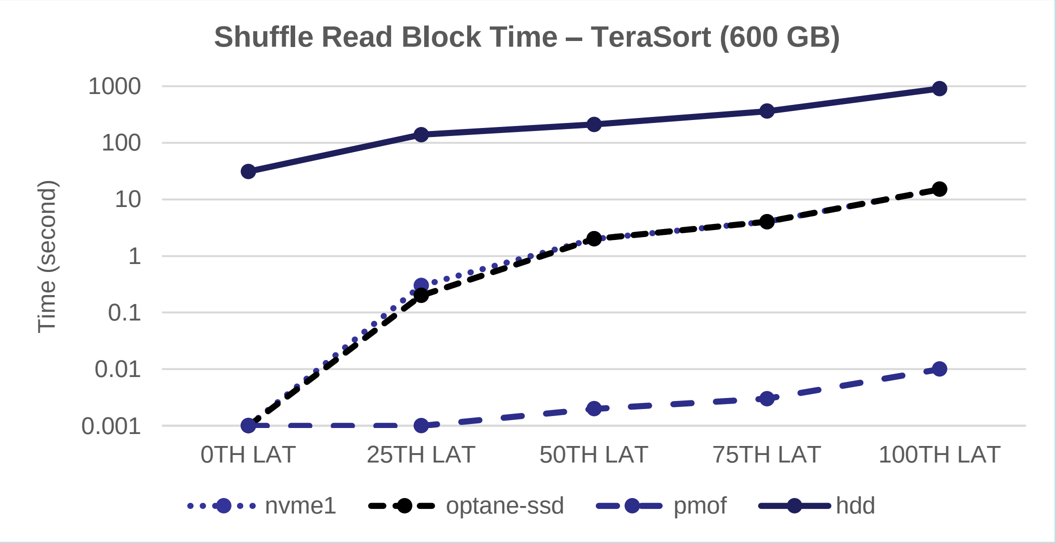
Workloads

Terasort 1TB:

- hibench.spark.master yarn-client
- hibench.yarn.executor.num 12
- yarn.executor.num 12
- hibench.yarn.executor.cores 8
- yarn.executor.cores 8
- spark.shuffle.compress false
- spark.shuffle.spill.compress false
- spark.executor.memory 60g
- spark.executor.memoryoverhead 10G
- spark.driver.memory 80g
- spark.eventLog.compress = false
- spark.executor.extraJavaOptions=-XX:+UseG1GC
- spark.hadoop.yarn.timeline-service.enabled false
- spark.serializer org.apache.spark.serializer.KryoSerializer
- hibench.default.map.parallelism 200
- hibench.default.shuffle.parallelism 1000

Spark PMoF Performance





- Spark-PMoF shows great end-to-end execution time in TeraSort.
  - ~13.7x performance benefit over HDD.
  - ~5% performance benefit over NVMe (P4500).
  - ~10.5% slower than Optane-SSD (P4800), since Optane-SSD has higher write bandwidth than DCPM.
- Spark-PMoF shows ultra low shuffle remote read latency.
  - Median latency reduces by ~1000x than NVMe and Optane-SSD, reduces ~105000x than HDD.
  - Tail latency reduces by ~1500x than NVMe and Optane-SSD, reduces ~90000x than HDD.

Flash Memory Summit

Agenda

- Background and motivation
- Bigdata analytics on the cloud: the challenges & optimizations
- Accelerate bigdata analytics on cloud with in memory data accelerator (IMDA)
  - IMDA as Cache
  - IMDA as shuffle
- **Summary**

Flash Memory Summit

Summary

- Bigdata analytics is the key cloud workload, customer is adopting
- Lots of challenges running Bigdata analytics on public cloud, including functionality, simplicity, performance gaps
- With bigdata analytics on public cloud, a new high performance, low latency in memory data accelerator leveraging state-of-art HW technologies can help to address the performance gaps
- POC with Alluxio IMDA as Cache and Spark PMoF as shuffle demonstrated significant performance and latency improvement

Notices and Disclaimers

- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.
- This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.
- The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.
- Intel, the Intel logo, Xeon, Optane, Optane DC Persistent Memory are trademarks of Intel Corporation in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others
-

Legal Information: Benchmark and Performance Disclaimers

- Performance results are based on testing as of Feb. 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information, see Performance Benchmark Test Disclosure.
- Configurations: see performance benchmark test configurations.