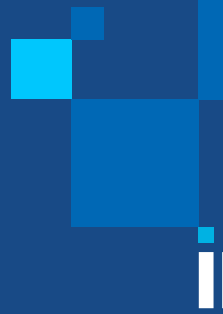
Accelerating Apache Spark with DAOS on Aurora

Carson Wang

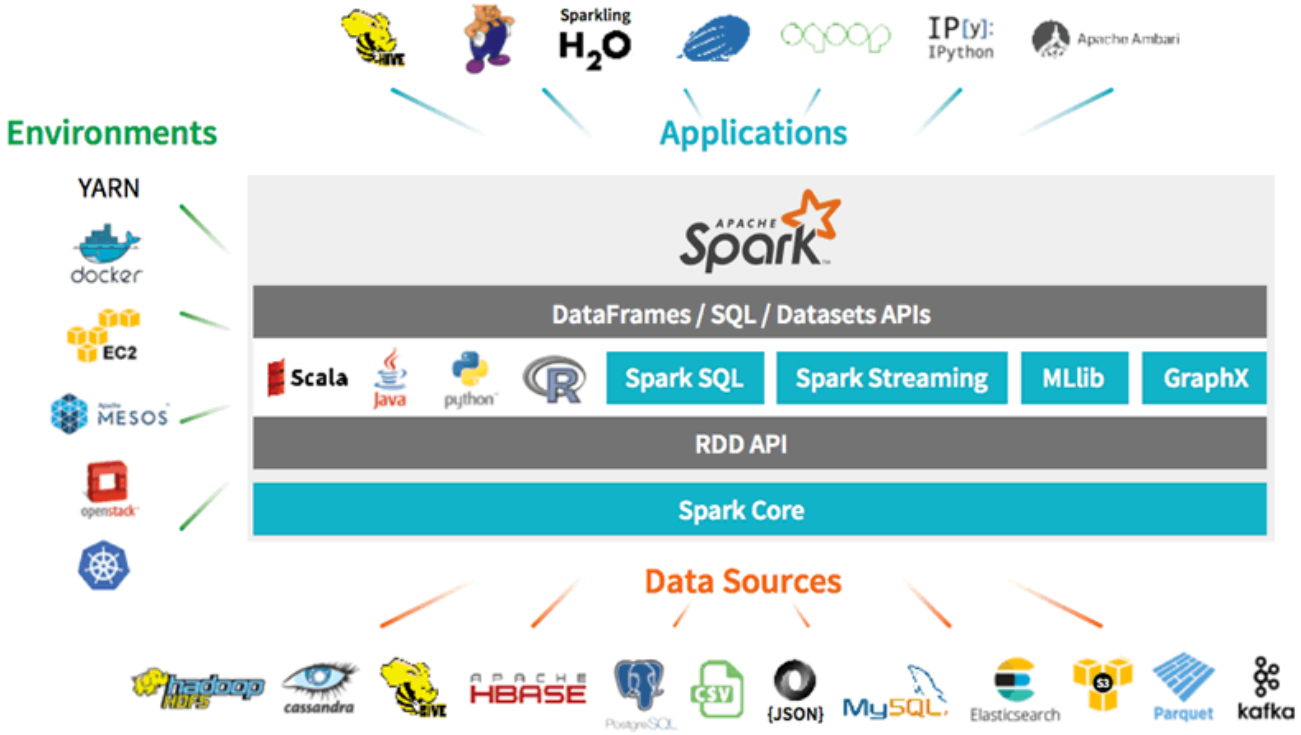
Nov. 2020



Agenda

- Apache Spark Overview
- DAOS Hadoop Filesystem
- DAOS Spark Shuffle Manager

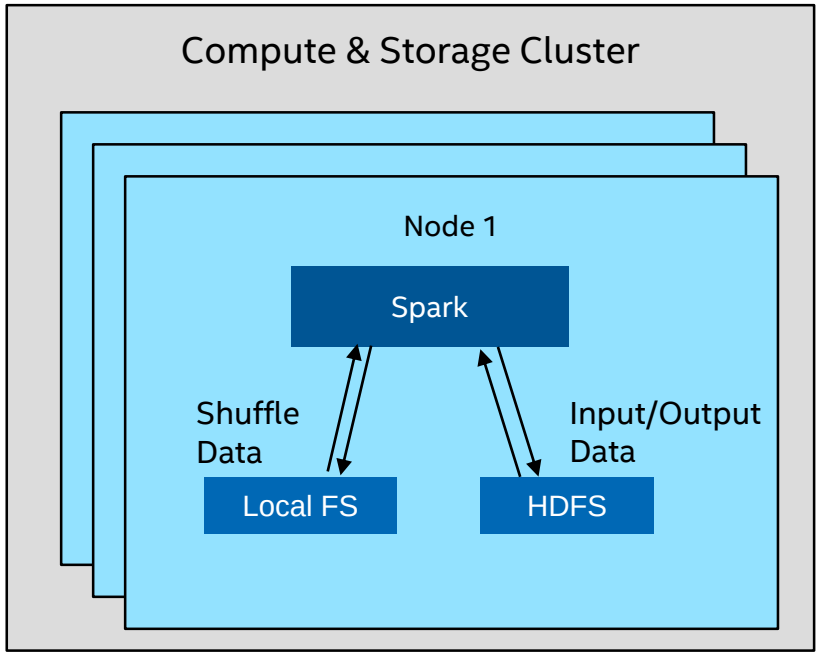
Apache Spark: A Unified Analytics Engine



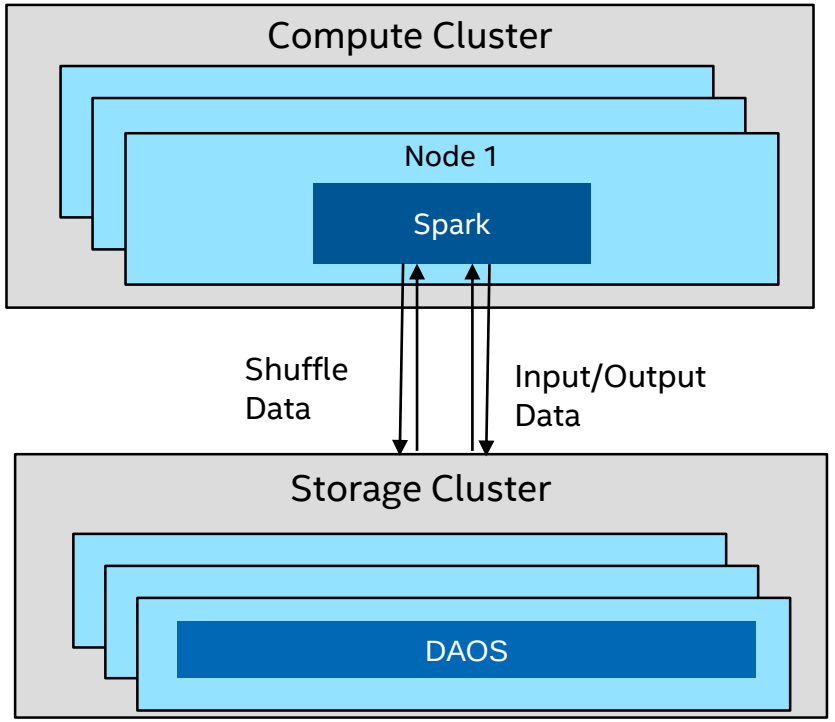
- A fast and general-purpose cluster computing system, supports a rich set of high level tools including Spark SQL for SQL processing, MLLib for machine learning, GraphX for graph processing and Spark Streaming.
- Provides high level APIs in Java, Scala, Python and R.
- Supports multiple resource manager(Apache Yarn, Apache Mesos, Kubernetes, standalone)
- Supports diverse data sources including HDFS, Apache Cassandra, Apache Hbase, etc.

Enable and Accelerate Spark with DAOS

- Spark Input/Output Storage: From HDFS to DAOS
- Spark Shuffle Data Storage: From Local FS to DAOS

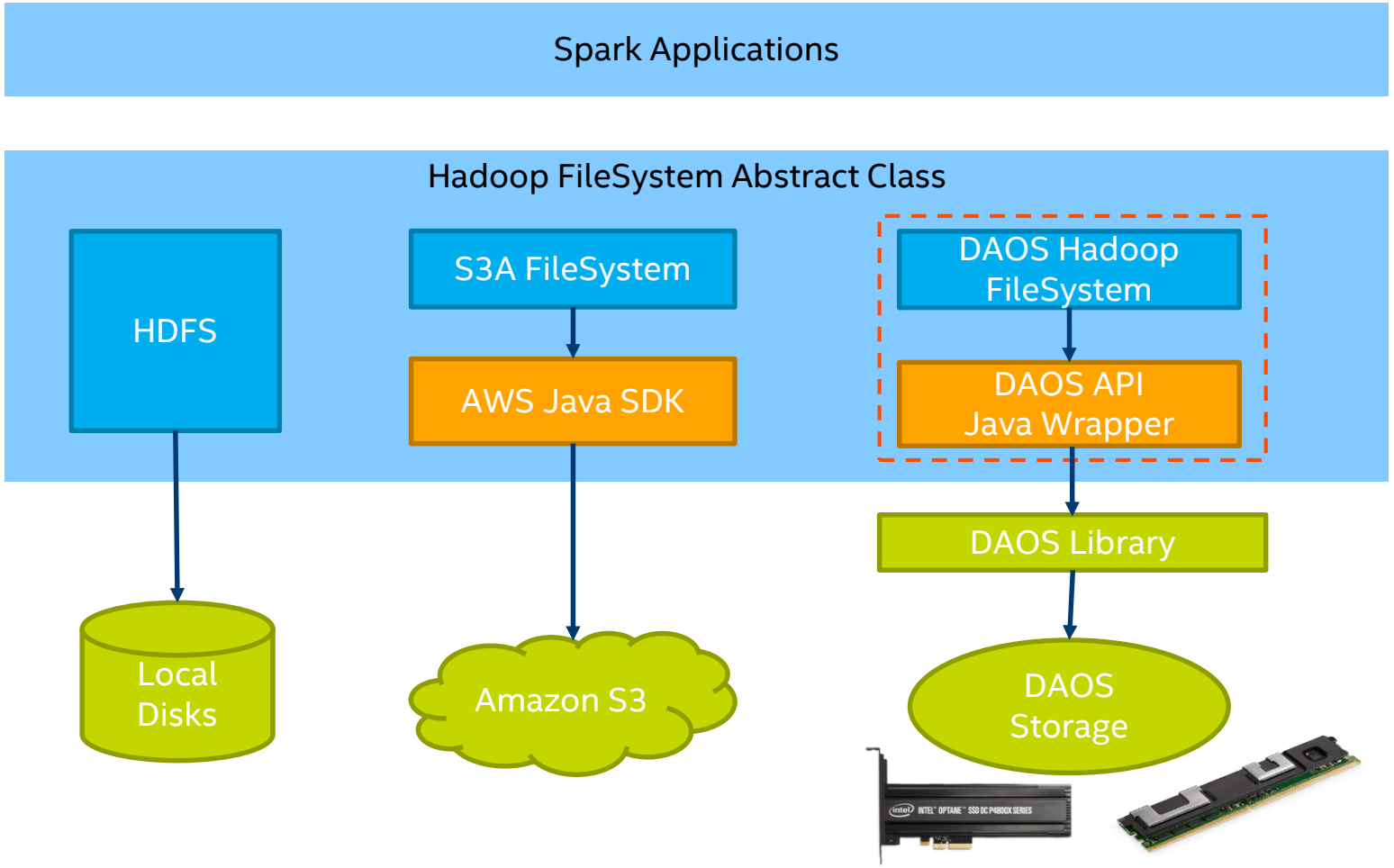


Colocated Cluster



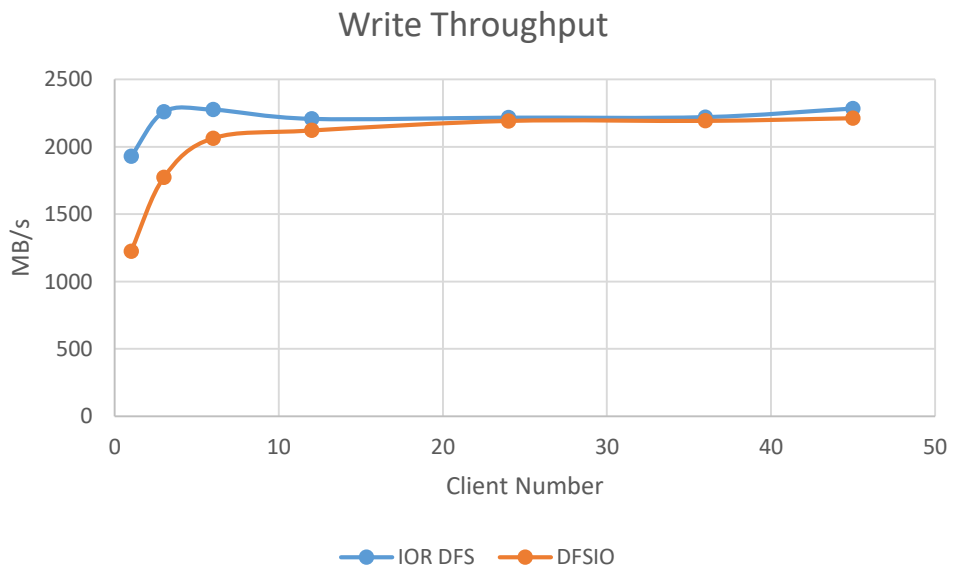
Disaggregated Cluster

DAOS Hadoop Filesystem Interface



DAOS Hadoop Filesystem Throughput

- DAOS Hadoop Filesystem delivers the same read/write throughput as DAOS DFS API with enough parallelism.



2 DAOS Servers, each with 1 P4500 NVMe; 3 Client Nodes; Network: 100 Gb Omni-Path

Using DAOS Hadoop Filesystem in Spark

- Add the DAOS Hadoop FS jar file to the classpath of the Spark executor and driver

```
spark.executor.extraClassPath    /path/to/DAOS Hadoop FS jar
spark.driver.extraClassPath      /path/to/DAOS Hadoop FS jar
```

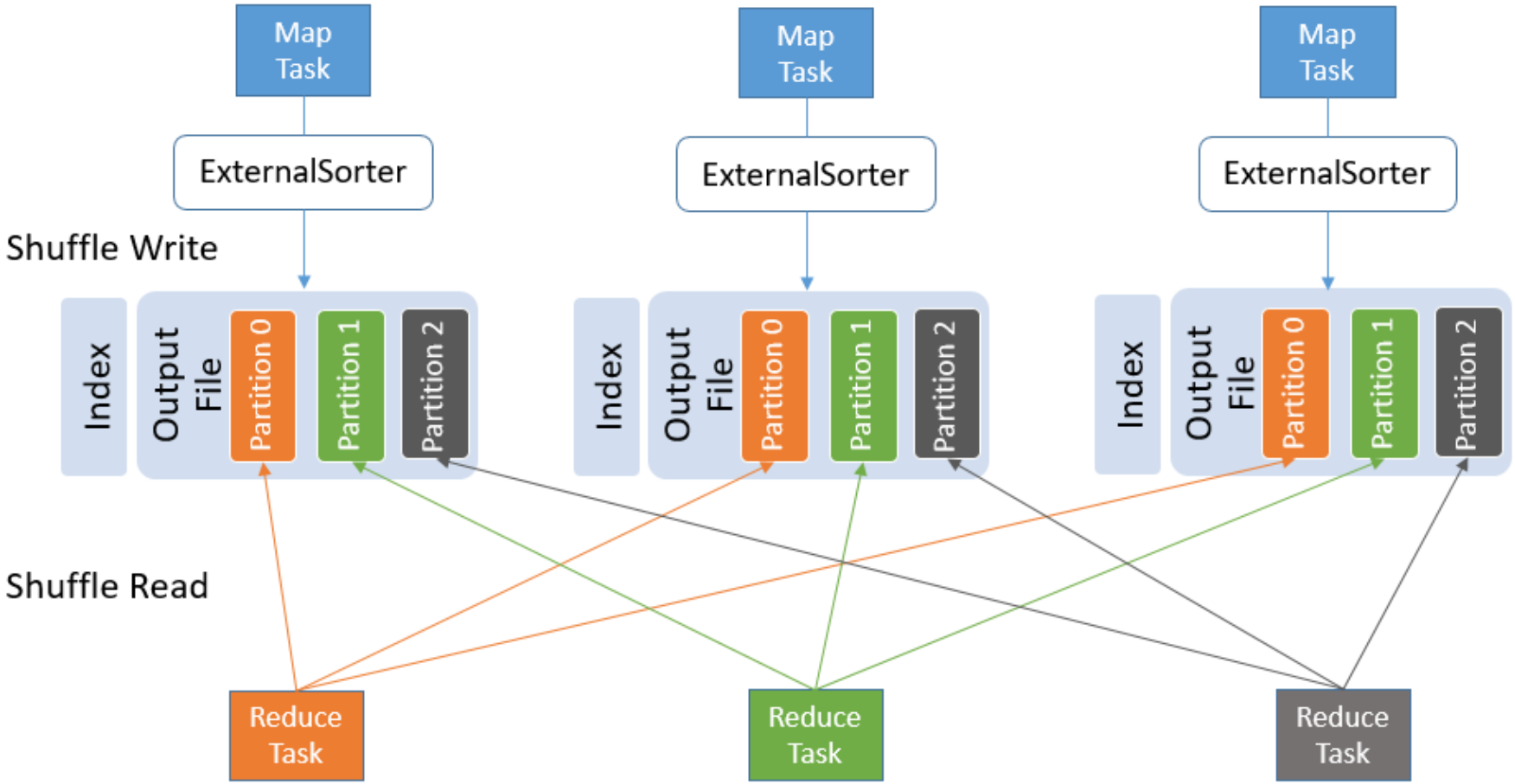
- For DAOS UNS path, read data using the following URI in Spark

```
df = spark.read.json("daos:///UNS/path/root/dir/file")
```

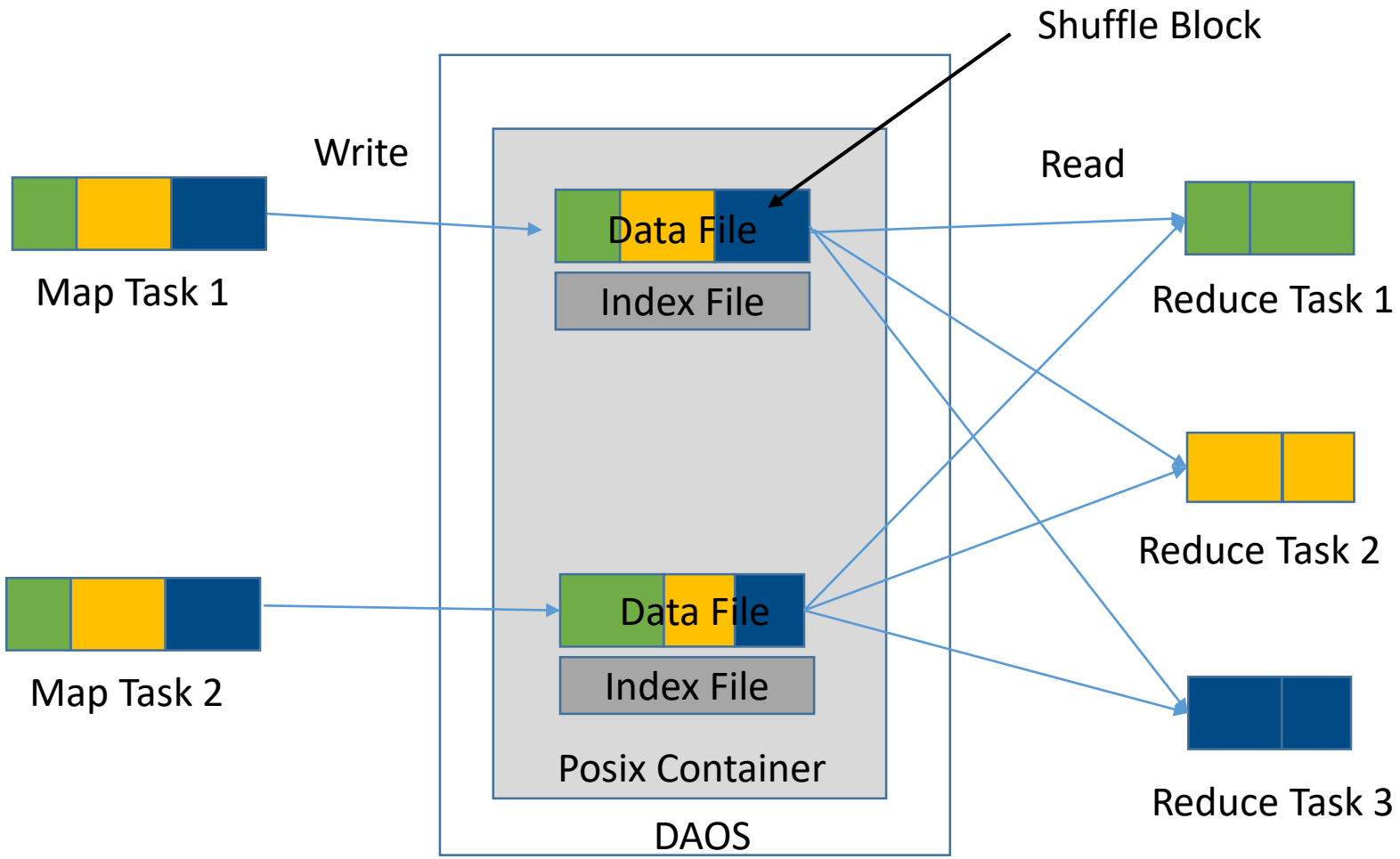
- For non-UNS path, add pool and container UUID in configuration file daos-site.xml and use the following URI in Spark.

```
df = spark.read.json("daos:///root/dir/file")
```

Spark Shuffle Overview

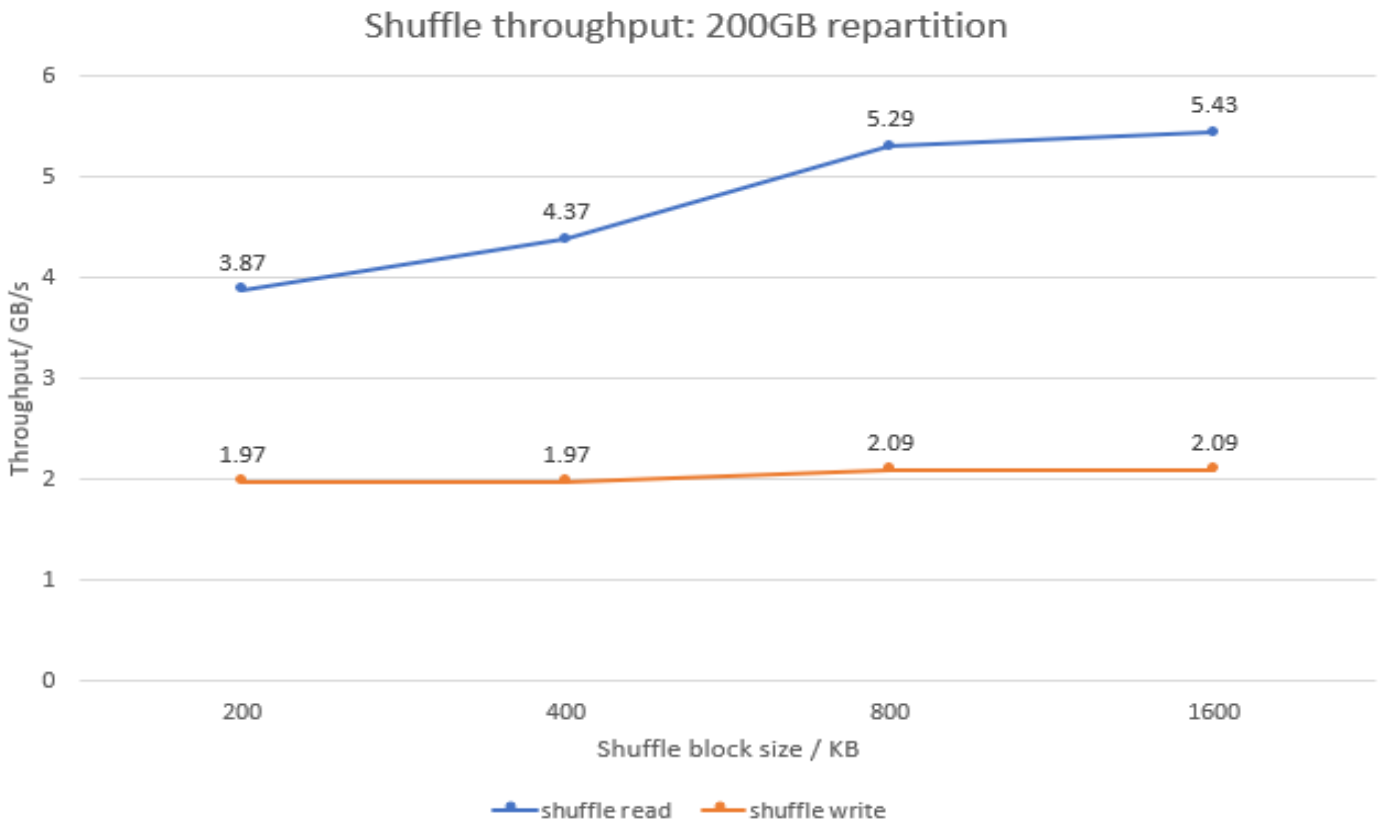


DAOS Hadoop FS based Shuffle



1. Map task writes the shuffle output file to DAOS using DAOS Hadoop filesystem API. The shuffle data file is sorted by partition ID.
2. Reduce task reads the shuffle partitions from every shuffle data file.
3. Every reduce task needs to open all shuffle data files. File handles can only be cached at Spark executor level.

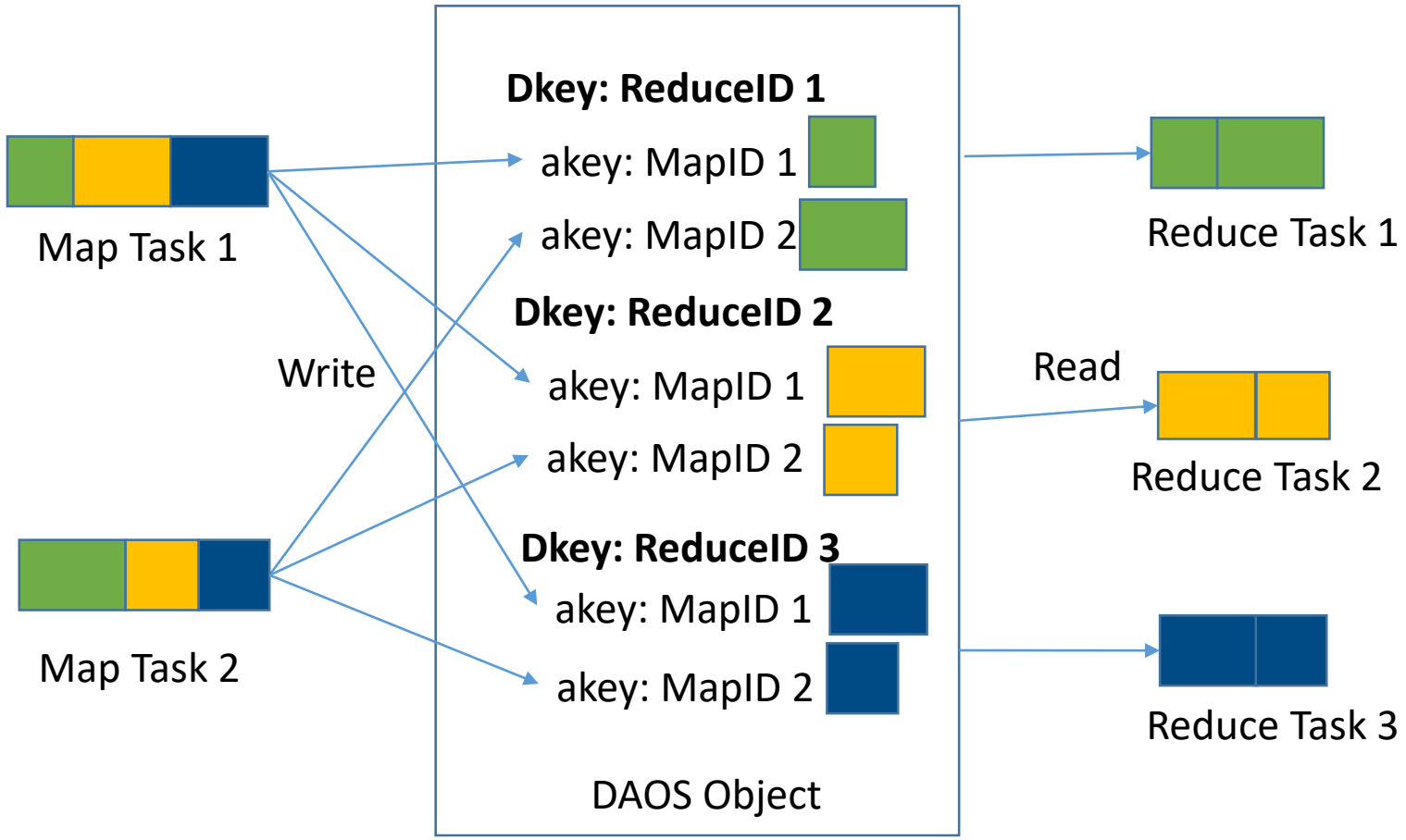
DAOS Hadoop FS based Shuffle Performance



Challenges:

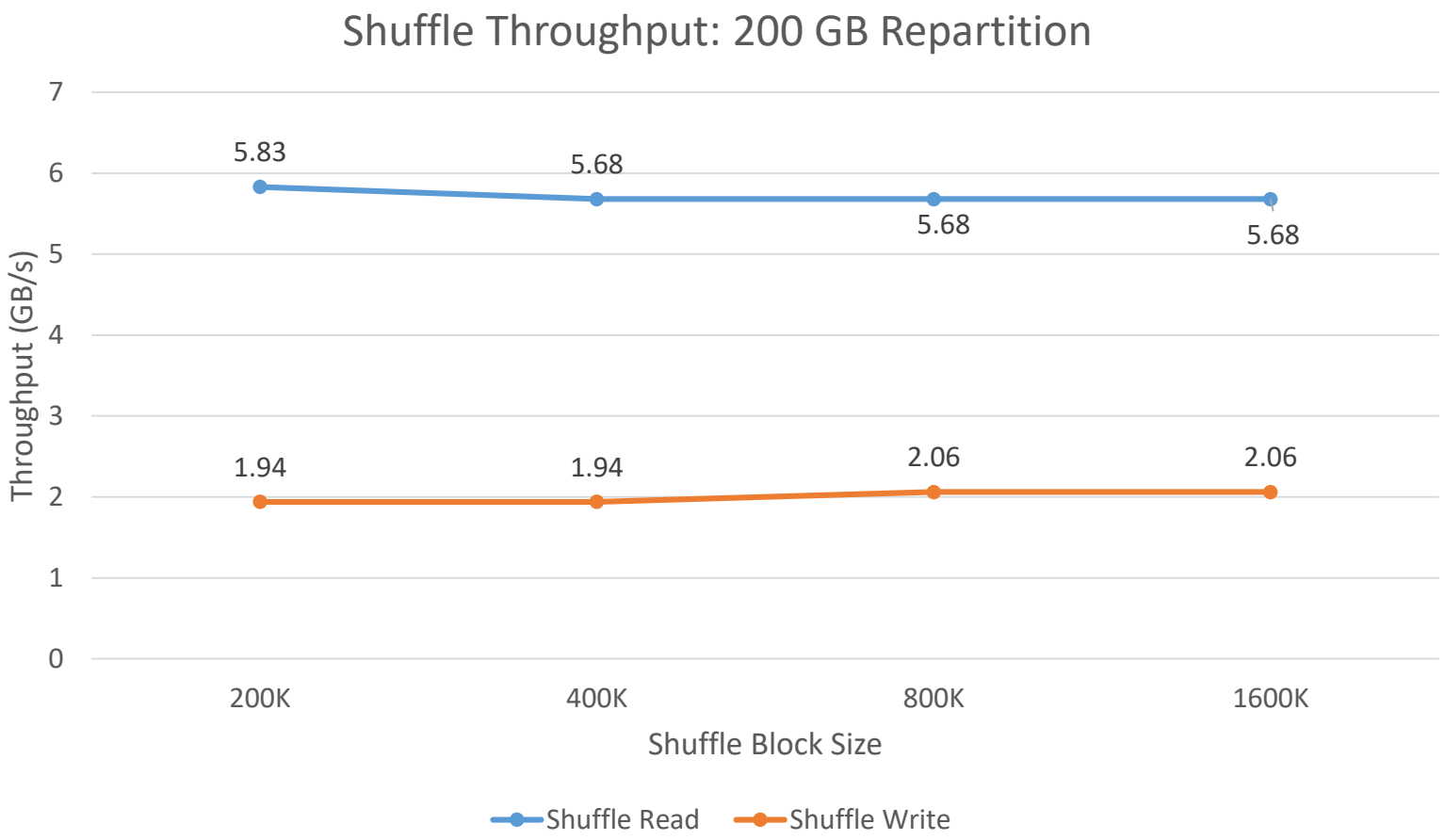
Frequent file lookup/open overhead during Shuffle Read, especially when shuffle block size is small.

DAOS Object based Spark Shuffle



1. Map task divides the data into R partitions where R equals to the number of reducers, and every small block will be stored into a DAOS object with a specific Dkey and akey where Dkey equals to ReduceID and akey equals to Map attempt ID.
2. For every reduce task, its shuffle data will be stored under the same Dkey. During shuffle read, every reduce task just reads all successful akeys' value under the Dkey.

DAOS Object based Spark Shuffle Performance



- Shuffle Read Throughput has been improved, 1.5x for 200KB shuffle block.
- Shuffle Write Throughput doesn't change as it already hits the NVMe limit.

Using DAOS Object based Shuffle in Spark

- Add the remote shuffle jar file to the classpath of the Spark executor and driver

spark.executor.extraClassPath	/path/to/DAOS remote shuffle jar
spark.driver.extraClassPath	/path/to/DAOS remote shuffle jar

- Enable DAOS remote shuffle

spark.shuffle.manager	org.apache.spark.shuffle.daos.DaosShuffleManager
spark.shuffle.daos.pool.uuid	<Pool UUID>
spark.shuffle.daos.container.uuid	<Container UUID>

Future Plan

- Performance:
  - Switch to DAOS async API in the DAOS Spark shuffle manager.
- User Experience:
  - Simplify user configuration and set better default values.
  - Test and benchmark to cover more use cases.


intel