The Practice of Presto & Alluxio in E-Commerce Big Data Platform

Tao Huang, JD.com
Big Data Platfrom Engineer

2019-06-20

Contents

1 JD BDP

JD BDP

cluster scale

Tens of thousands of nodes

Thousands of users

Computing ability

Tens of PB offline data daily

Millions of jobs daily

Storage capacity

Hundreds of PB data

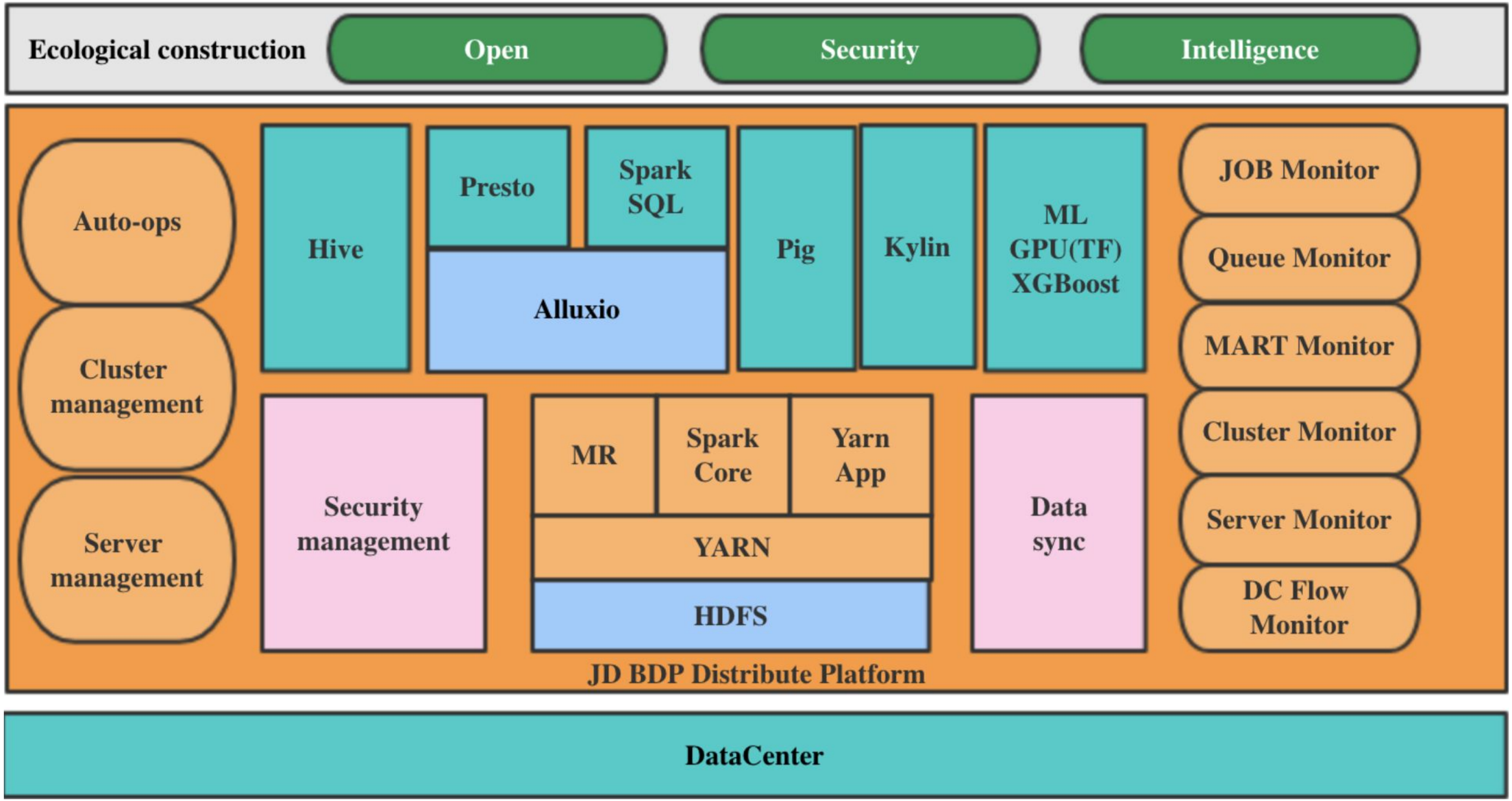
Tens of PB daily increase

Business scale

Tens of business units

Hundreds of data models

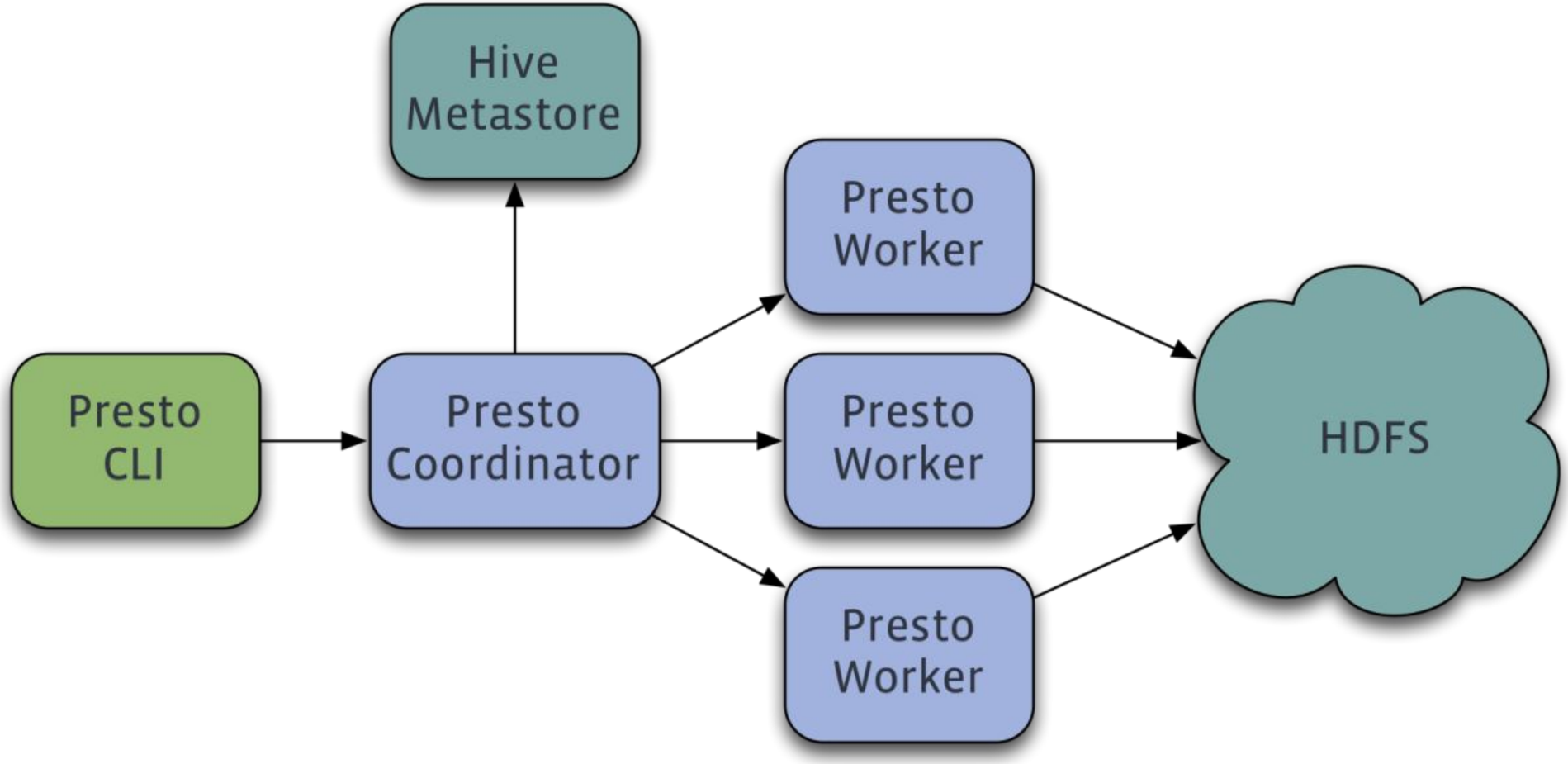
BDP architecture



2

Practice with Presto in BDP

Presto Architecture



Our Works on Presto

01 Cluster Scaling

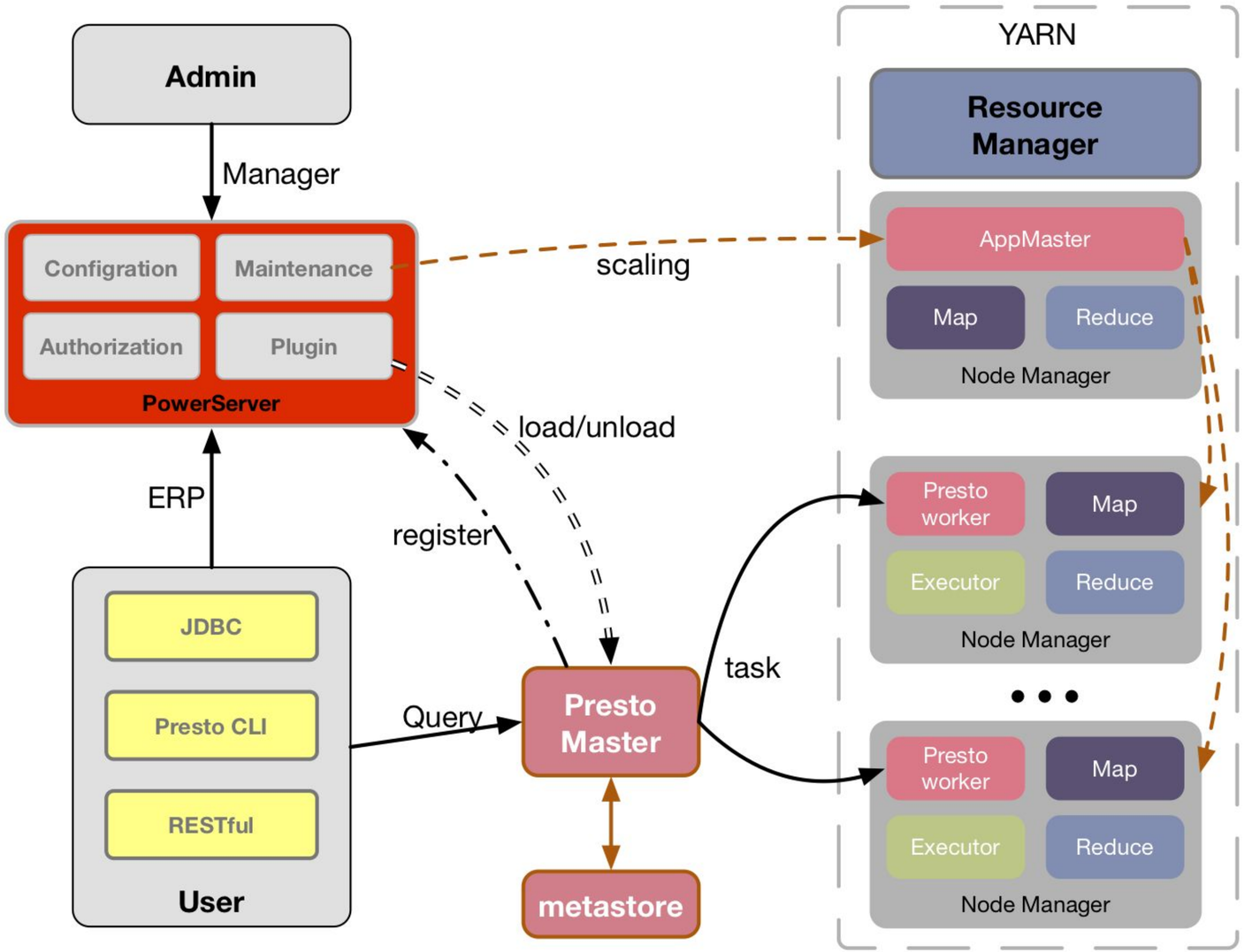
02 Job Isolation

03 ERP Authorization

04 Operation & Maintenance

Presto on YARN

YARN	Unified Resource Management
Dynamic Resource	Presto worker scaling
Configuration	Configure Presto in Web
Plugin	load/unload plugins



PowerServer for operation and maintenance

Plugin manager

- export query result
- update plugin

ERP Authorization

- track users’query
- security

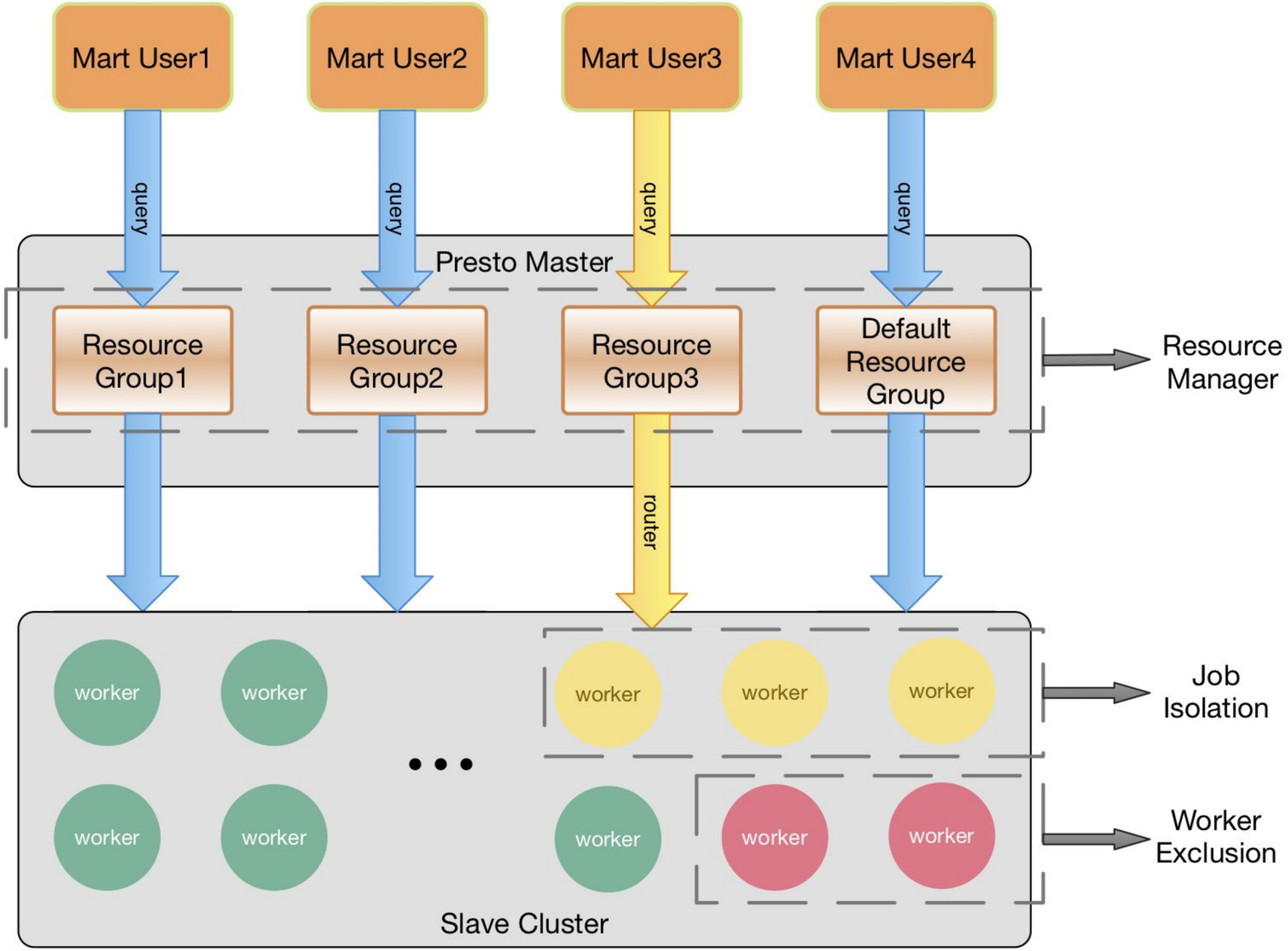
Dynamic Congfiguation

- route query to cluster
- adjust resource group

Auto Maintenance

- dynamic auto-scale
- start/stop cluster

Intelligent Scheduler



Application Scenario

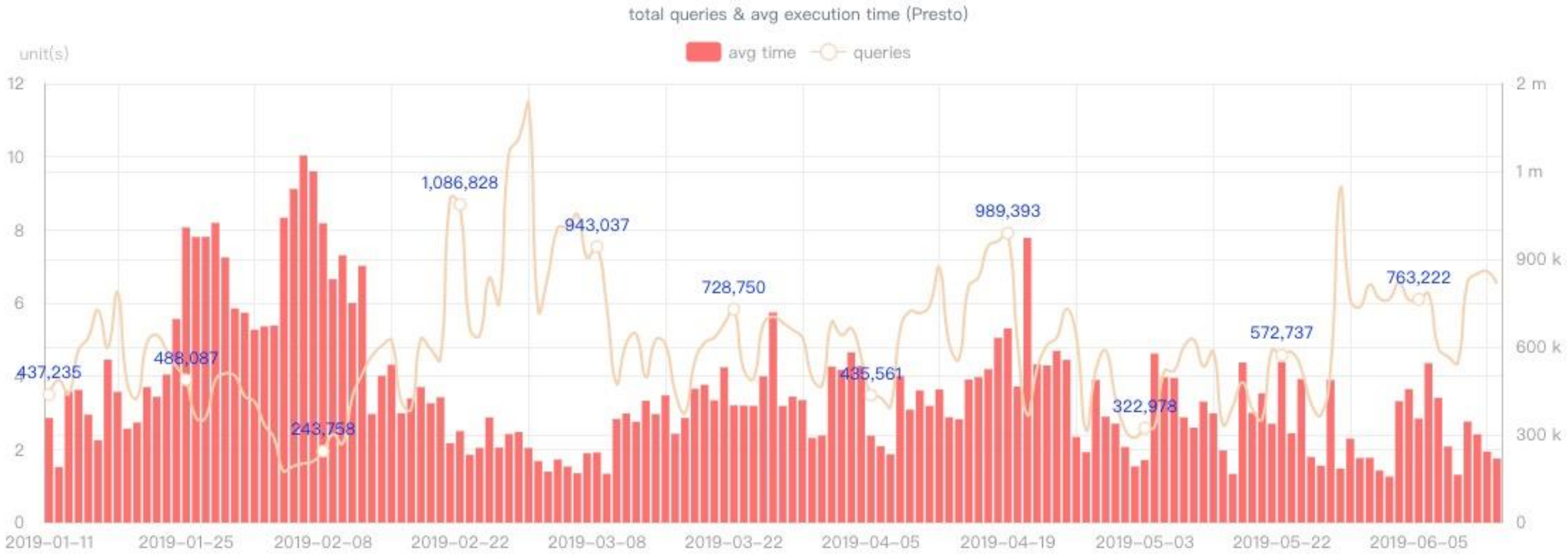
Periodical Queries

- controllable data range
- high query frequency
- high data reuse rate
- high proportion

Unpredicatible Queries

- controllable data range
- low query frequency
- low data resuse rate
- low proportion

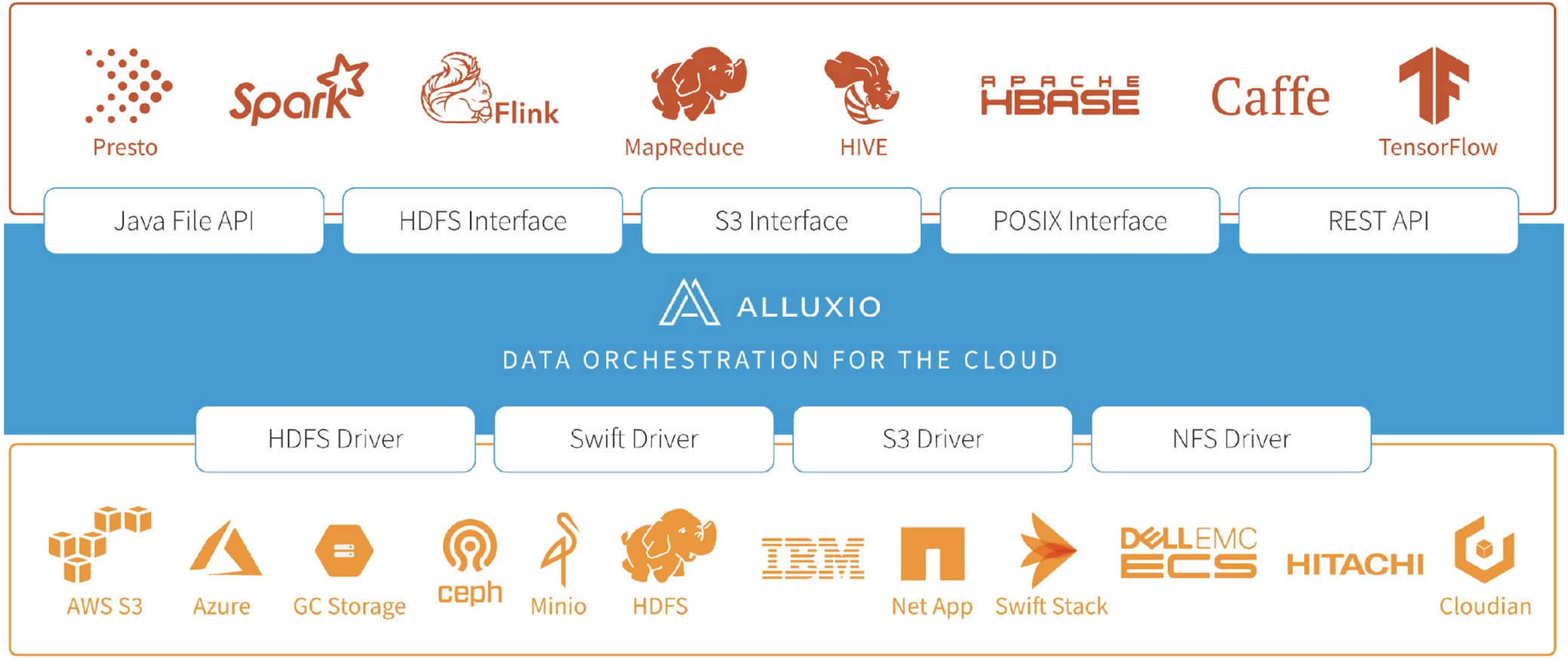
Presto Jobs in BDP



3

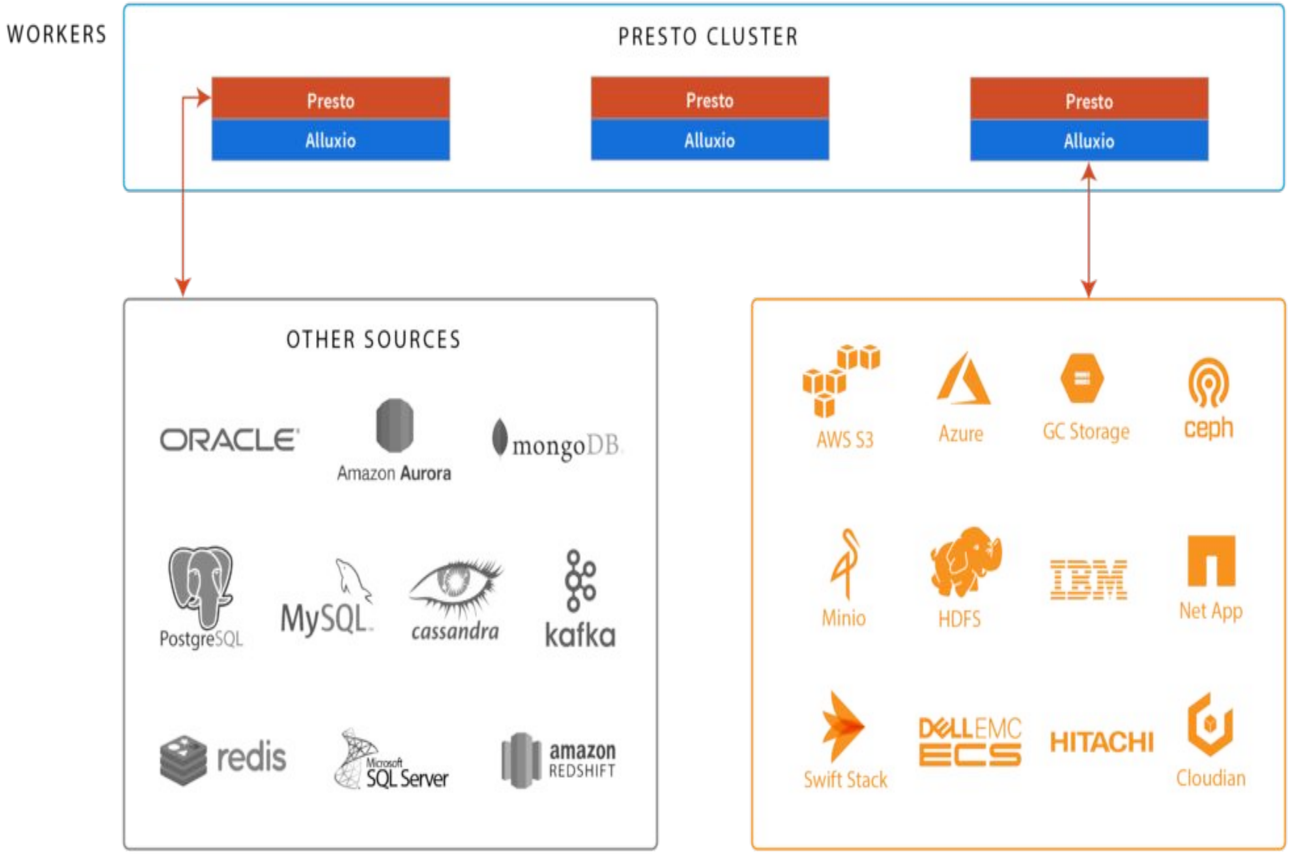
Presto & Alluxio Stack

Data Ecosystem with Alluxio



- **Apps only talk to Alluxio**
- **Simple Add/Remove**
- **No App Changes**
- **Highest performance in Memory**

Presto + Alluxio = Better Together



Higher query throughput

Consistent low query latency

Eliminates network traffic

JD Contribution to Alluxio

New Web UI

ui-grid based
sort/pagination/filter add an input field

Cache Consistency

check startsup
check every time

Watermark Evict Strategy

high watermark start evict
low watermark stop evict

JVM Pause Monitor

monitor JVM pause periodically
log message and metrics

Shell Command

cp/ls/load/rm/format

Change Log Level

shell
RESTful API

Bugfix

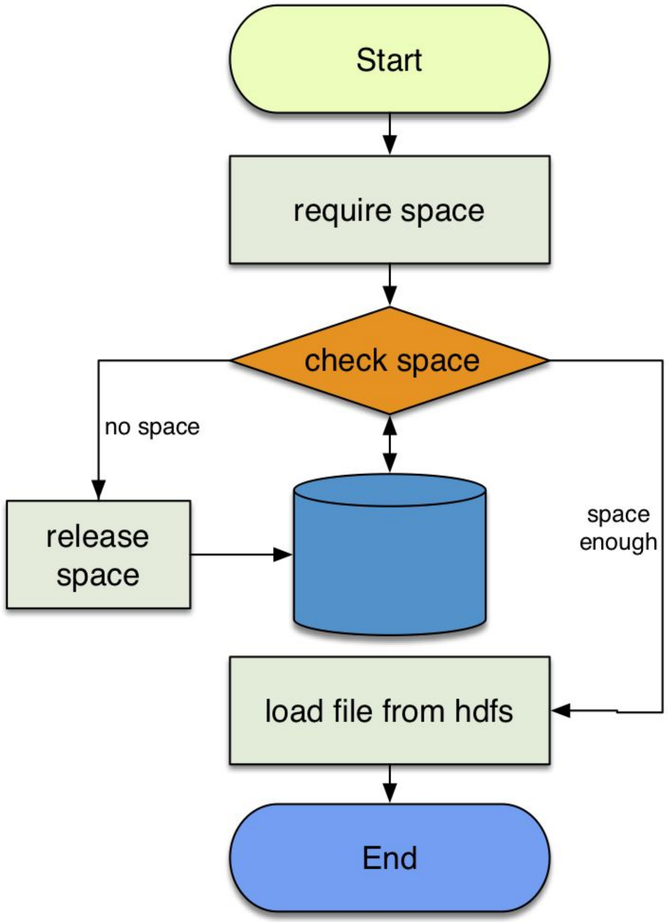
Deadlock
thrift add timeout time
…

Test

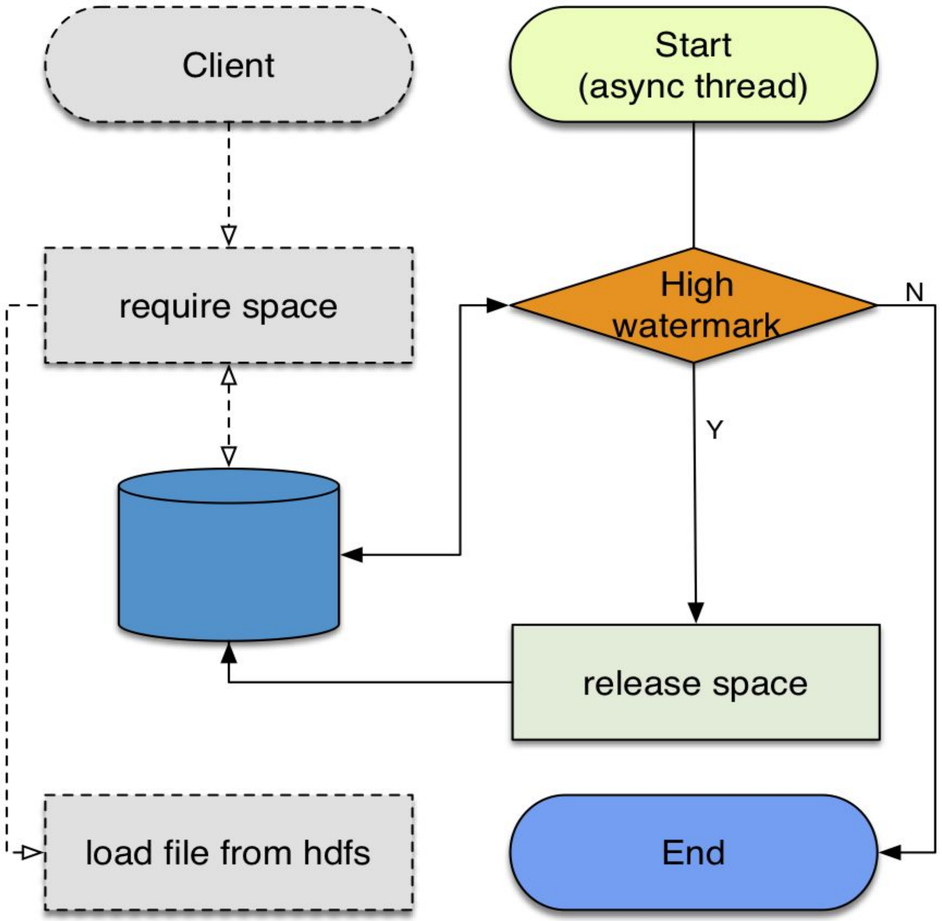
SyncQuery
AlluxioTools
…

Watermark Evict Strategy

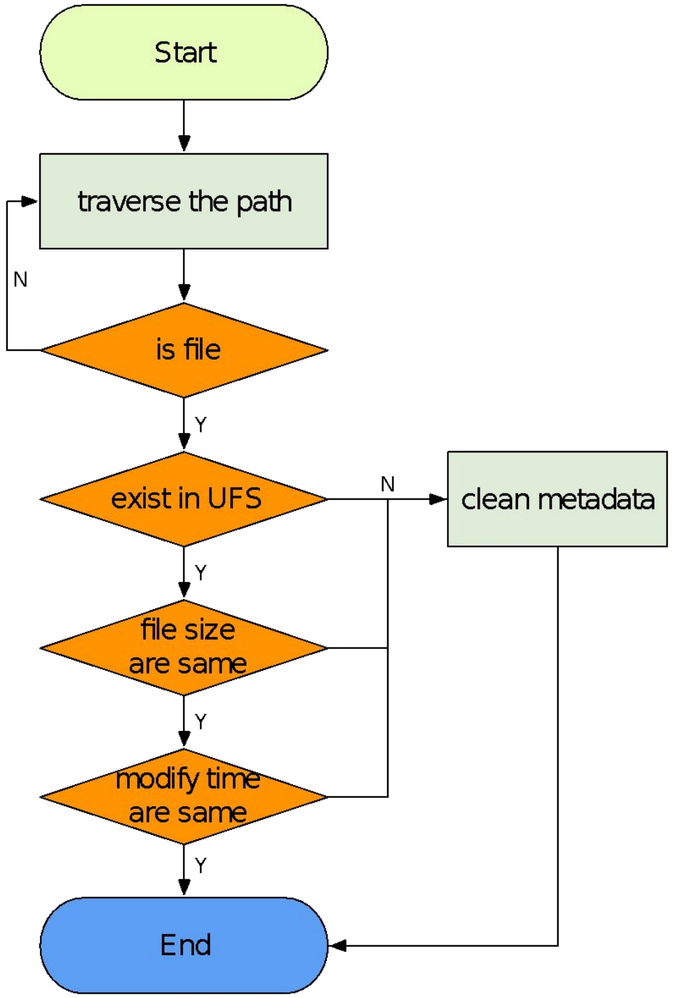
Sync Evit Strategy



Async Evit Strategy



Cache Consistency



Keep Alluxio & HDFS Consistency

To ensure that dirty data is not read. There are three ways to trigger file consistency check.

RPC API

Client request metadata by getFileId, getFileInfo, listStatus, etc

Alluxio master will check file cache consistency

RESTful API

calling reloadMetaData to trigger Alluxio to reload all metadata

Alluxio Master startup

check file cache consistency while master start up

Presto on Alluxio

- Alluxio led to **10x** performance improvement
- Hundreds of nodes
- More than 2 years in production enviroment.

Why Presto on Alluxio?

When we use Alluxio for Presto, we make some changes and bring some good features

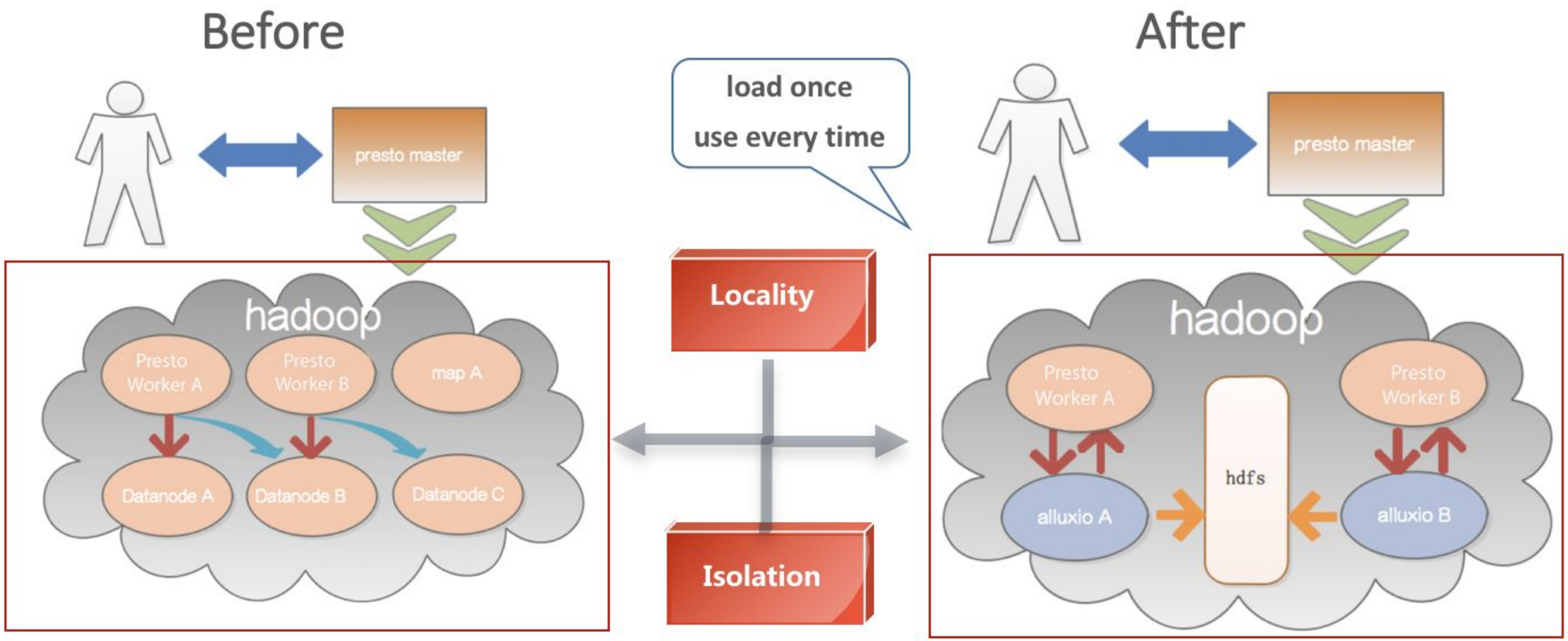
High Performance

Consistent Low Query Latency

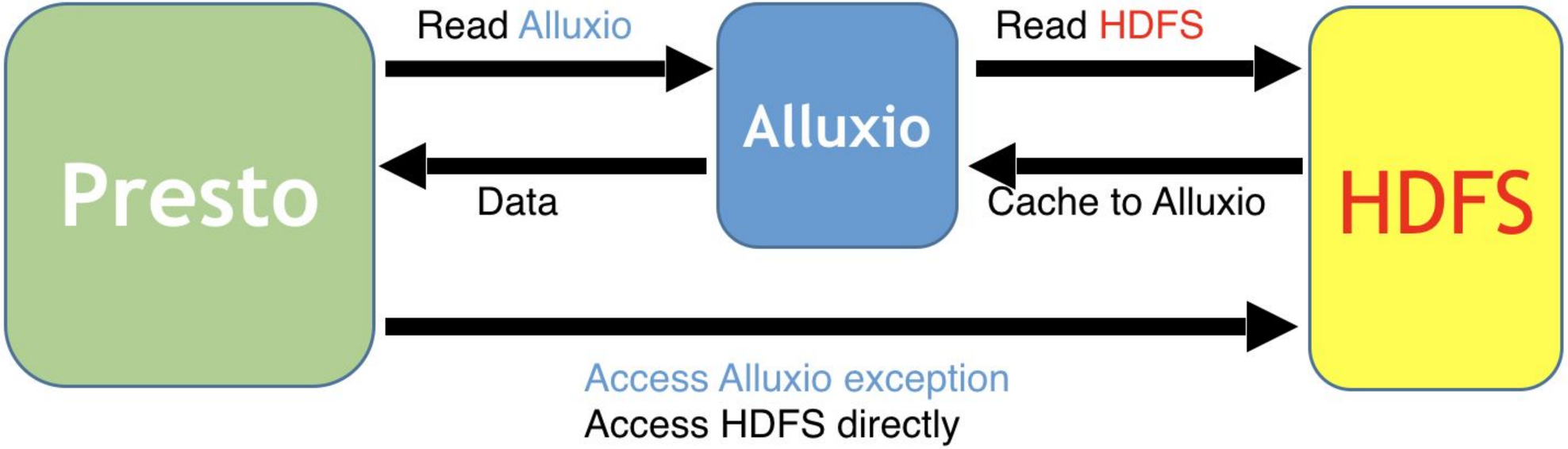
Eliminate Network Traffic

Others: Fault-tolerant & Pluggable

Presto on Alluxio



Presto on Alluxio



Presto on Alluxio

HDFS

9s –> 9s –> 12s –> 18s –> 10s –> 14s

```
presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_004552_00030_6ta4h, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:09 [43.3M rows, 5.29GB] [5.02M rows/s, 629MB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_004603_00031_6ta4h, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:09 [43.3M rows, 5.29GB] [4.87M rows/s, 610MB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_004614_00032_6ta4h, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:12 [43.3M rows, 5.29GB] [3.76M rows/s, 471MB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_004628_00033_6ta4h, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:18 [43.3M rows, 5.29GB] [2.38M rows/s, 299MB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_004653_00034_6ta4h, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:10 [43.3M rows, 5.29GB] [4.31M rows/s, 539MB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_004706_00035_6ta4h, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:14 [43.3M rows, 5.29GB] [3.08M rows/s, 385MB/s]
```

ALLUXIO

20s –> 0.9s –> 0.6s –> 1s –> 0.5s –> 0.6s –>0.6s –> 0.3s

```
presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_001529_00034_5r5i4, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:20 [43.3M rows, 5.29GB] [2.15M rows/s, 269MB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_001637_00035_5r5i4, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:01 [43.3M rows, 5.29GB] [44.3M rows/s, 5.41GB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_001641_00036_5r5i4, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:01 [43.3M rows, 5.29GB] [70.4M rows/s, 8.61GB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_001643_00037_5r5i4, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:01 [43.3M rows, 5.29GB] [36.5M rows/s, 4.47GB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_001646_00038_5r5i4, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:01 [43.3M rows, 5.29GB] [77.1M rows/s, 9.43GB/s]

presto:adw> select count(1) from [XXX] where day='2017-10-13';
  _col0
----------
 43281455
(1 row)

Query 20171014_001648_00039_5r5i4, FINISHED, 8 nodes
Splits: 59 total, 59 done (100.00%)
0:01 [43.3M rows, 5.29GB] [67.4M rows/s, 8.24GB/s]
```

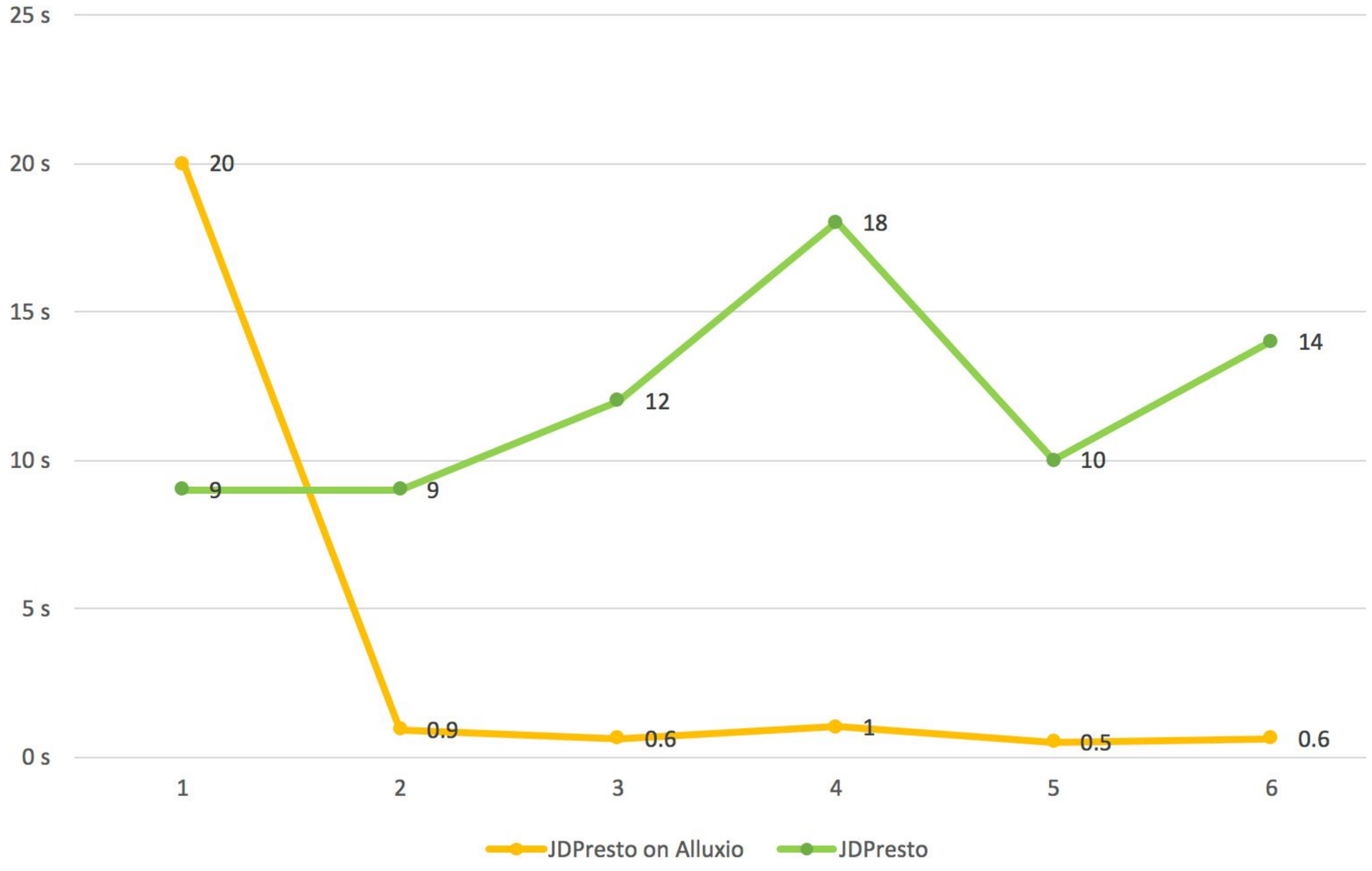
Presto on Alluxio

auto-refresh

20171014_001652_00041_5r5i4 root presto-cli	399.65ms (100%) FINISHED Completed Splits: 59	select count(1) from	where day='2017-10-13'
20171014_001650_00040_5r5i4 root presto-cli	664.44ms (100%) FINISHED Completed Splits: 59	select count(1) from	where day='2017-10-13'
20171014_001648_00039_5r5i4 root presto-cli	640.74ms (100%) FINISHED Completed Splits: 59	select count(1) from	where day='2017-10-13'
20171014_001646_00038_5r5i4 root presto-cli	557.10ms (100%) FINISHED Completed Splits: 59	select count(1) from	where day='2017-10-13'
20171014_001643_00037_5r5i4 root presto-cli	1.18s (100%) FINISHED Completed Splits: 59	select count(1) from	where day='2017-10-13'
20171014_001641_00036_5r5i4 root presto-cli	613.02ms (100%) FINISHED Completed Splits: 59	select count(1) from	where day='2017-10-13'
20171014_001637_00035_5r5i4 root presto-cli	976.23ms (100%) FINISHED Completed Splits: 59	select count(1) from	where day='2017-10-13'
20171014_001529_00034_5r5i4 root presto-cli	20.14s (100%) FINISHED Completed Splits: 59	select count(1) from	where day='2017-10-13'

first query

Presto on Alluxio



4

Ongoing Exploration

Presto Exploration

Presto Master Load Balancing

As the amount of data grows, the cluster size becomes larger, and query tasks become more and more, Master will become a performance bottleneck. To achieve load balancing, how to improve Presto will be a challenge.

Thread Level Resource Isolation

The execution tasks running on the workers compete for resources, especially the jobs in the test phase. If we can restrict the execution tasks with CGroups, it will reduce the mutual impact among queries.

Unify Larger Clusters

Large-scale cluster help improving resource utilization. In the past year, we have reduced the number of clusters from more than 100 to 20. Within ensuring query efficiency, we will further increase the cluster size to reduce the number of clusters.

Alluxio Exploration

Exploring more application scenarios

Stores MapReduce/Spark shuffle data, to reduce disk storage pressure and speed up access to shuffle data

Porting HDFS Authentication to Alluxio

We are going to port custom authentication on our HDFS to Alluxio.

HDFS RBF or Alluxio

We have tried to use HDFS router-based fedration, but its performance does not meet our online requirements. We find that Alluxio also has forwarding capabilities and hopes that Alluxio will perform better, That is what we are doing.

Thank You!

huangtao6@jd.com