Apache Spark at Apple

Spark + AI Summit 2018

Sam Maclennan and Vishwanath Lakkundi

Traditional Apache Hadoop at Apple

We schedule with YARN on HDFS

A lot of batch processing including ingest

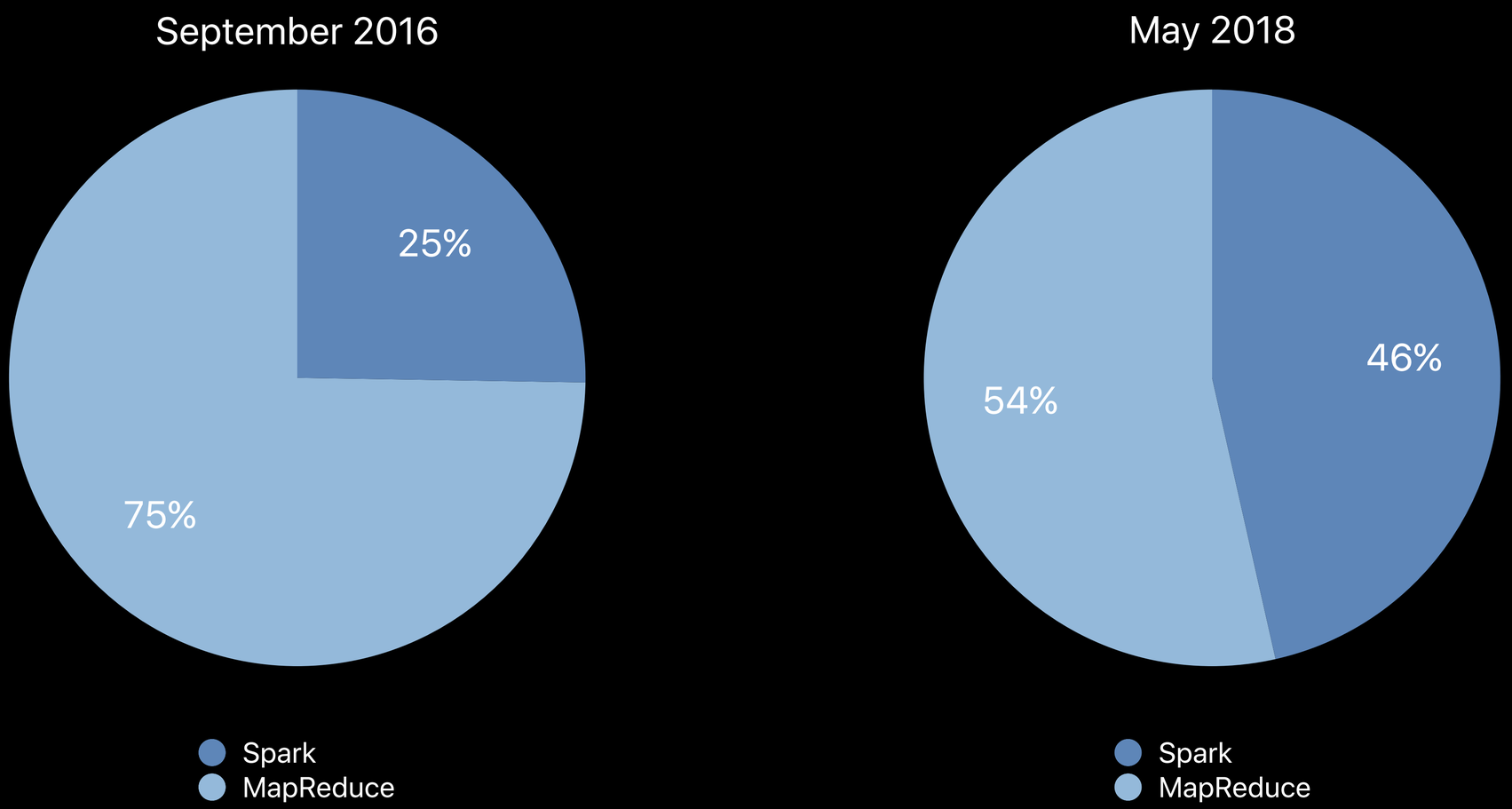
Along Comes Apache Spark

Users started testing pre 1.0

New set of users vs. MapReduce

Largely adhoc

Spark Growth



Challenges Scaling to Production

Listenerbus

Fault tolerance

History Server

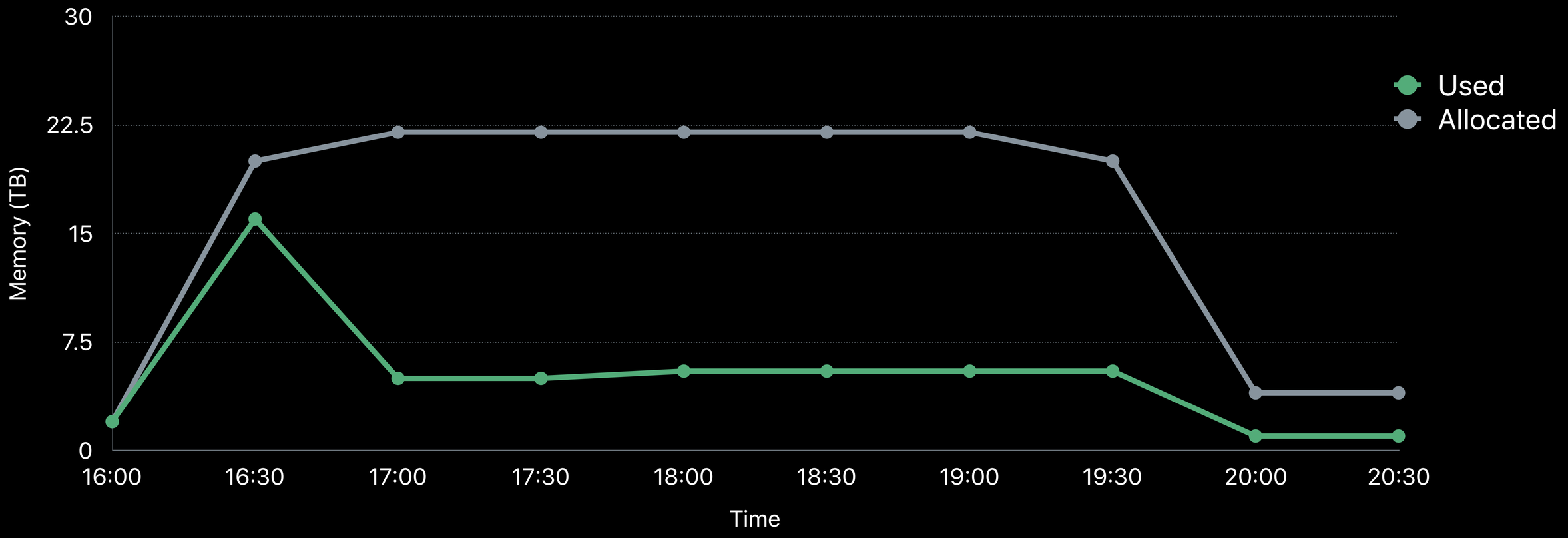
Streaming jobs

More Challenges Optimizing Resource Use

Estimating Spark resource usage is harder than MapReduce

Visualizing it is harder too

Visualizing Resource Use



Hadoop/Spark Footprint

Over an exabyte of storage

Half a million cores

Over 5PB of RAM

Hadoop Growth

In 18 months

- Storage—Up 2x
- Cores—Up 2x
- Memory—Up 4x
- Network—Now non-blocking

Future Challenges for Spark on YARN

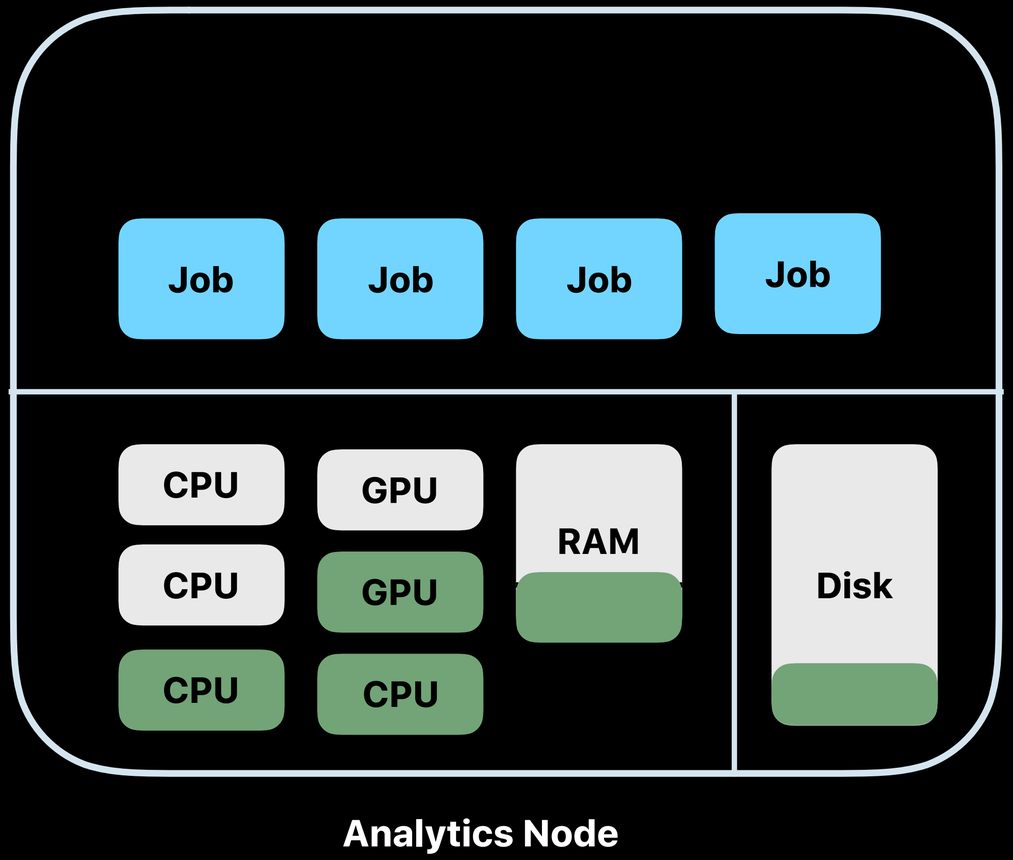
Tougher time with streaming jobs and having less reliance on data locality

Dimensions of hardware changing for Spark

Elastic Self Service Spark

Elastic Self Service Spark

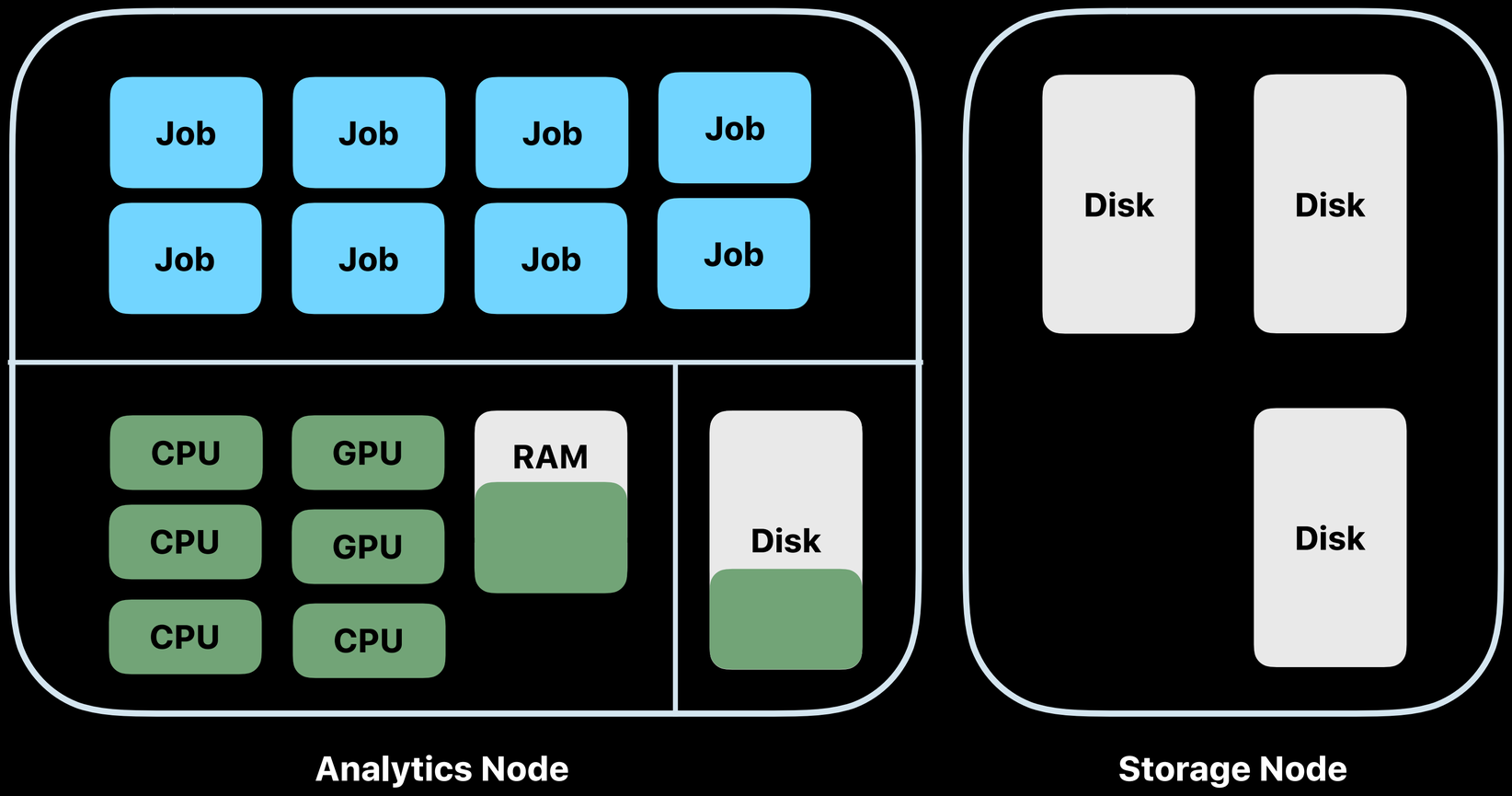
Why?



Analytics Node

Elastic Self Service Spark

Why?



Elastic Self Service Spark

Why?

Scale compute and storage independently

Maximize utilization of resources

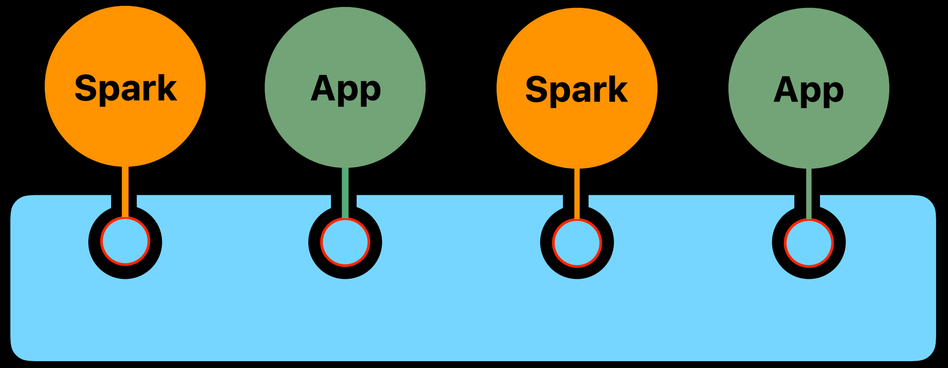
Elastic Self Service Spark

Goals

- Scalable, Multi-tenant, and on-demand Spark
- Security-first design
- Developer and Data Scientist productivity
- Cost efficiency
- Uptime
- Connectivity and Metrics

Elastic Self Service Spark

Shared, Multitenant Compute Infrastructure

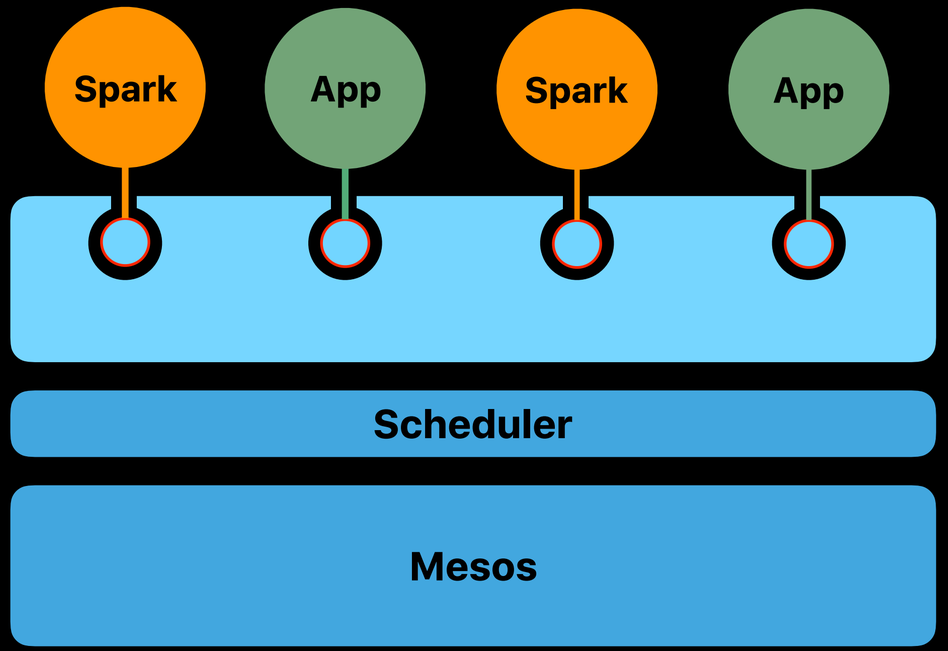


Elastic Self Service Spark

Shared, Multitenant Compute Infrastructure

Built on Apache Mesos and a custom scheduler

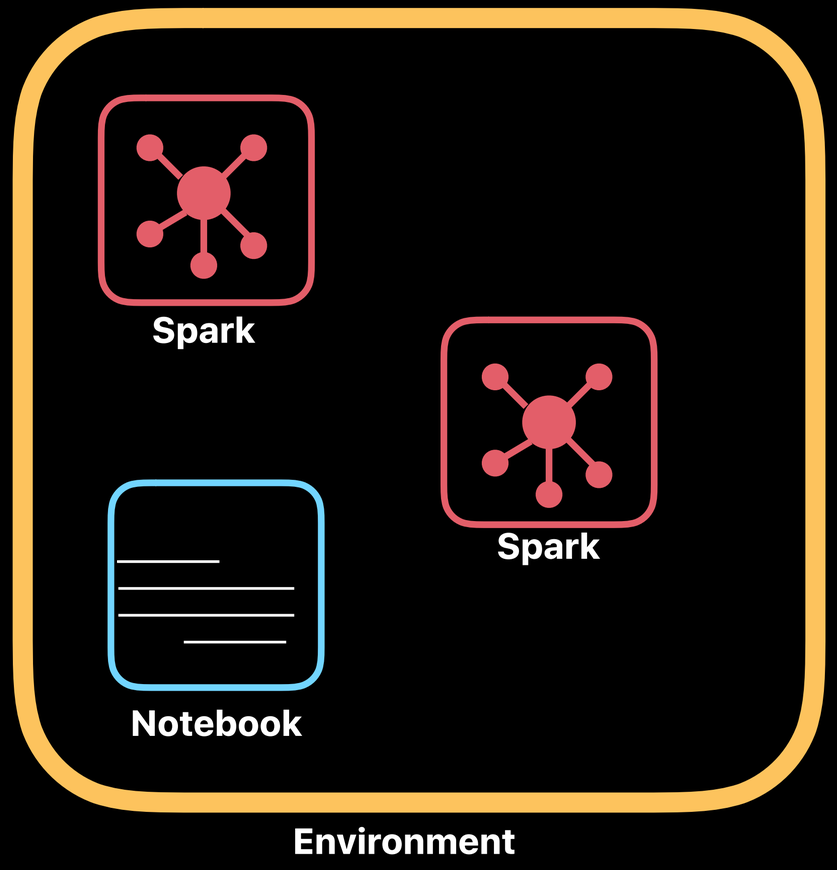
Spark 1.6.2, 2.1.1, 2.2.1 and 2.3



Concepts

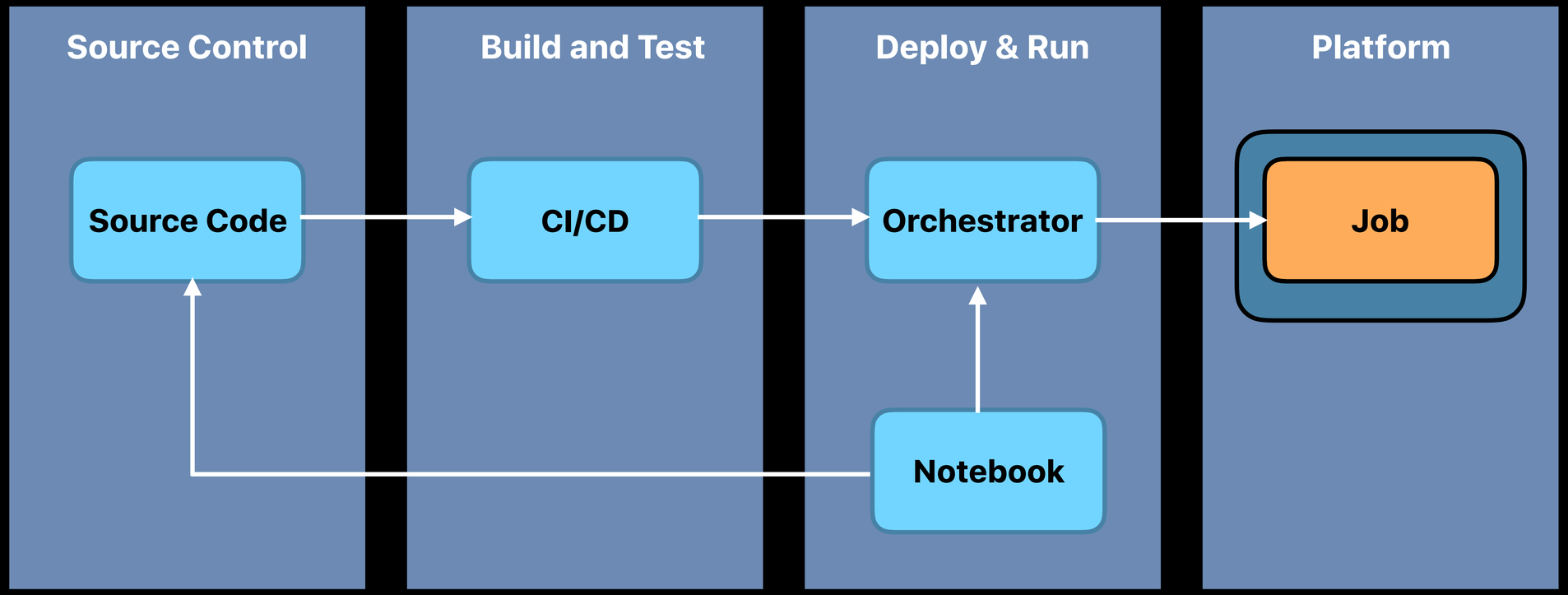
Logical Environment

- For network isolation
- For Automatic scale up/down boundaries
- For CI/CD
- One place to see them all

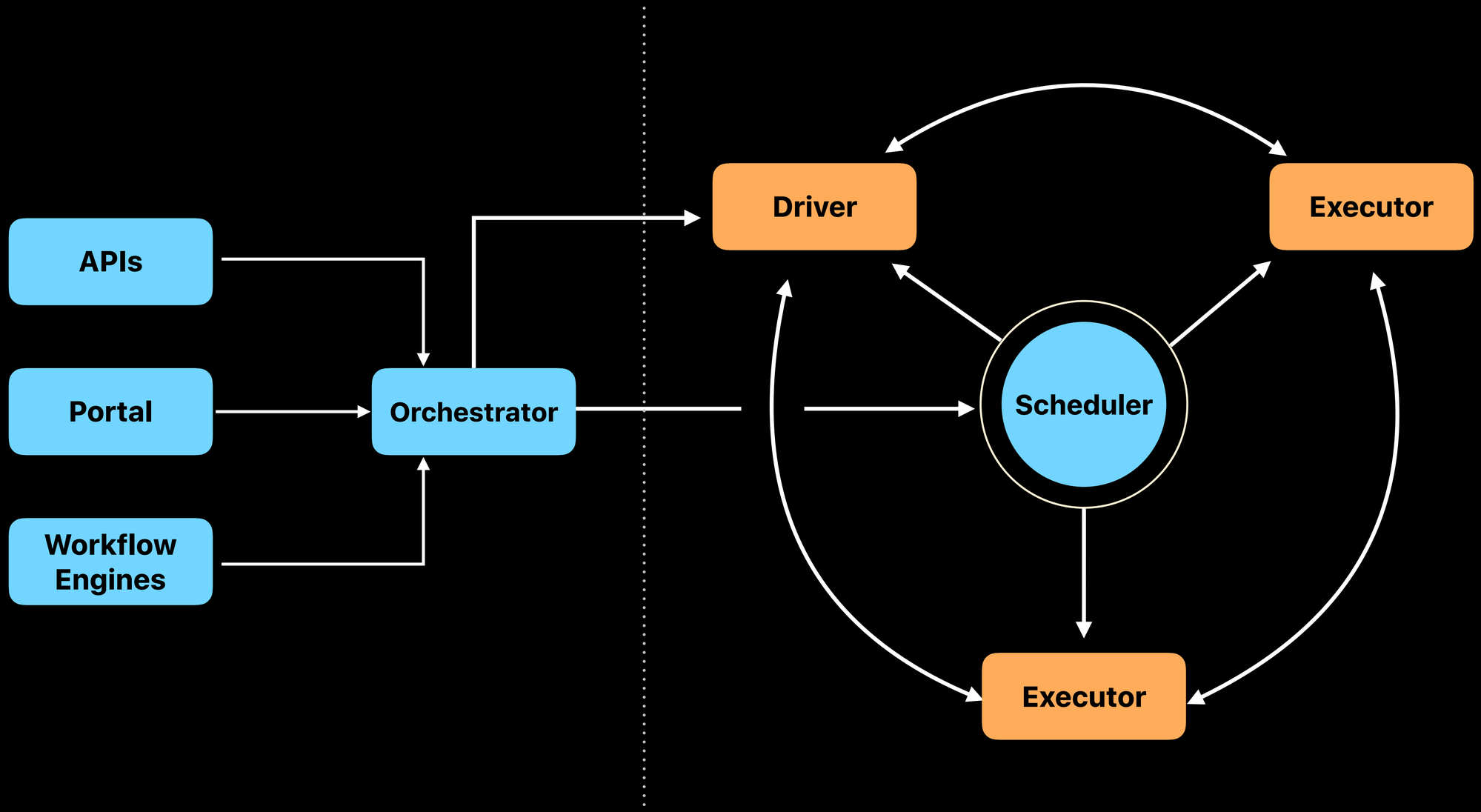


Productionizing Spark Jobs

Develop, build, deploy, run



How Does it Work?



Configuring Your Job

Yaml based Job spec

Define jobs and properties

```
jobs:
  # Name of the job
  - name: word-search


   jobClass: 'spark.examples.WordSearch'
   sparkVersion: 2.2.0


    # The properties for the spark job.
    properties:
       spark.executor.instances: 1000
       spark.executor.cores: 8
```

Security

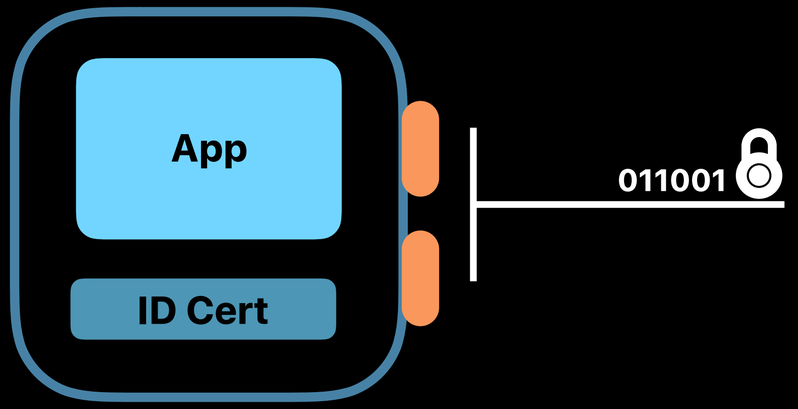
Application certificates for each driver and executor

Predefined network policy for connectivity

Authorized allocation of ports

Encryption on wire for all data

Secrets



Telemetry and Forensics

Driver and Executor Logs available through log management system

Automatic integration with Telemetry system

User defined metrics

Ability to alert on Key metrics

- Driver/Executor Memory
- Task Metrics (Failed Tasks, Input Records, etc)
- BlockManager, Scheduler
- System, Load Avg

History Server

Multi-tenant history server

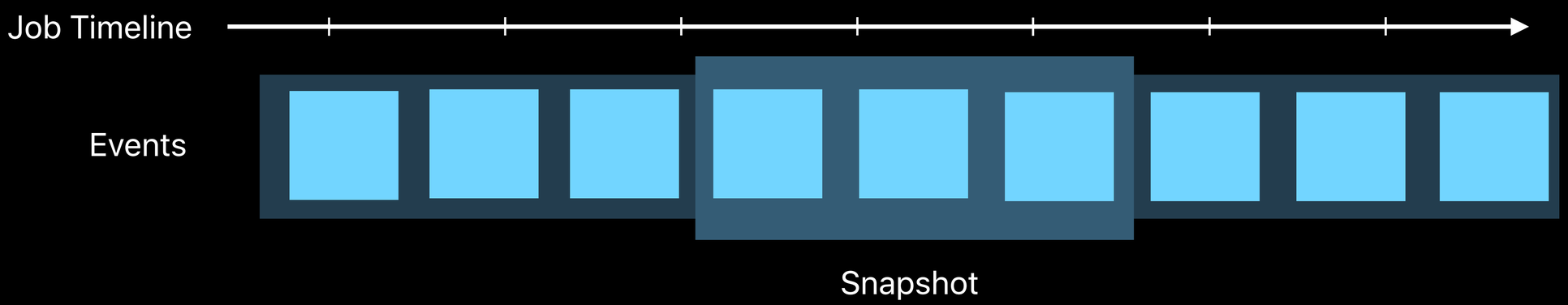
History server based off of Spark 2.3

History Server

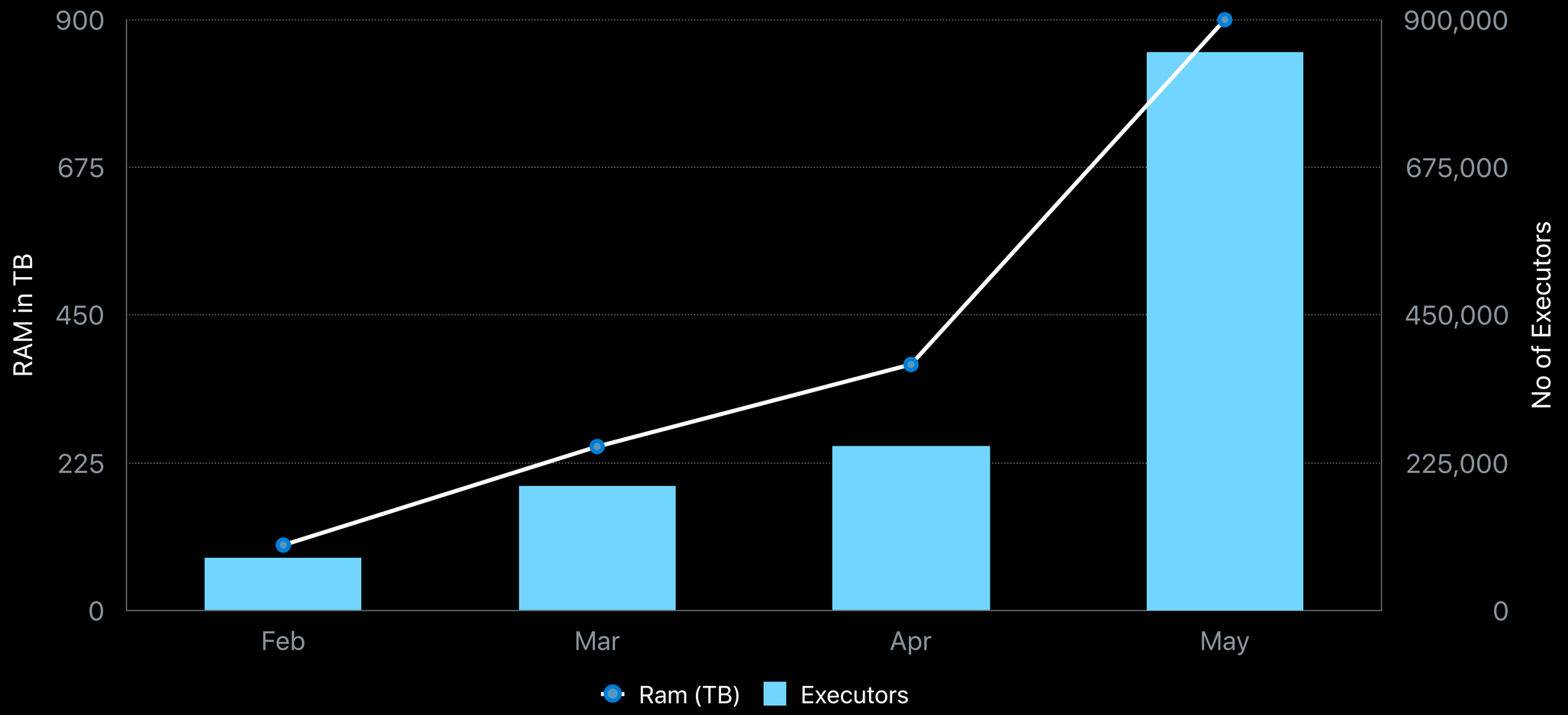
Multi-tenant history server

History server based off of Spark 2.3

Stores aggregated view of (most recent) 10000 jobs



Elastic Self Service Spark



Open Source Is In Our Culture

Open Source Is in Our Culture

Apache
ZooKeeper™

We are hiring!