SPARK SUMMIT 2017

HDFS on Kubernetes -- Lessons Learned

Kimoon Kim (kimoon@pepperdata.com)

pepperdata

Outline

1. Kubernetes intro
2. Big Data on Kubernetes
3. Demo
4. Problems we fixed -- HDFS data locality

Kubernetes

New open-source cluster manager.

Runs programs in Linux containers.

1000+ contributors and 40,000+ commits.

"My app was running fine until someone installed their software"

More isolation is good

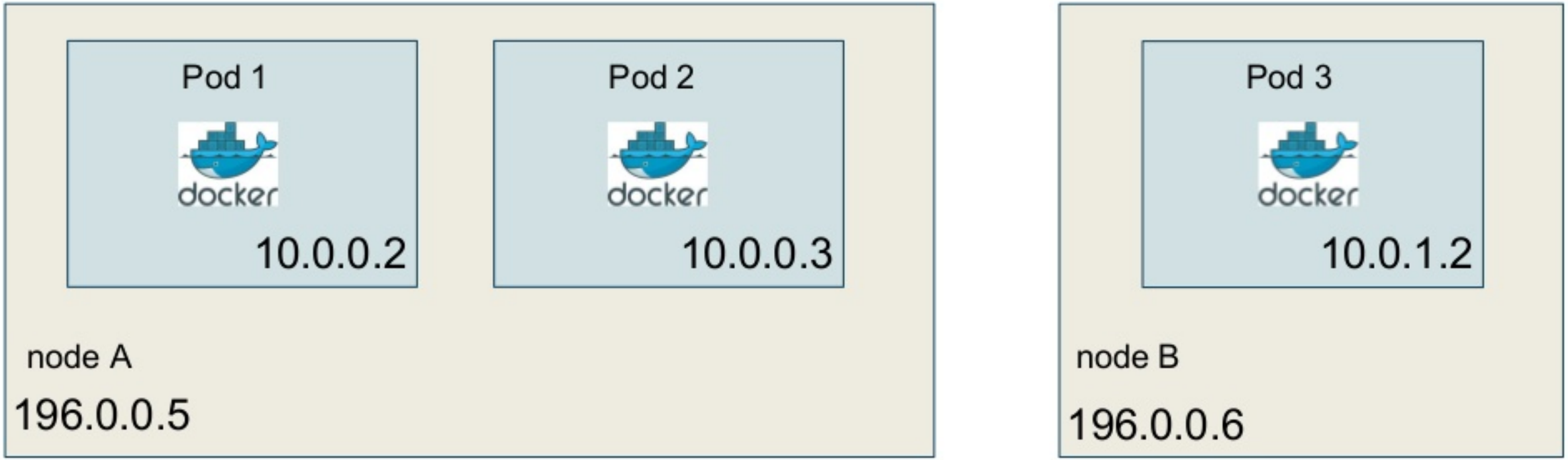
Kubernetes provides each program with:

- a lightweight virtual file system
  - an independent set of S/W packages
- a virtual network interface
  - a unique virtual IP address
  - an entire range of ports
- etc

Kubernetes architecture

Pod, a unit of scheduling and isolation.

- runs a user program in a primary container
- holds isolation layers like an virtual IP in an infra container



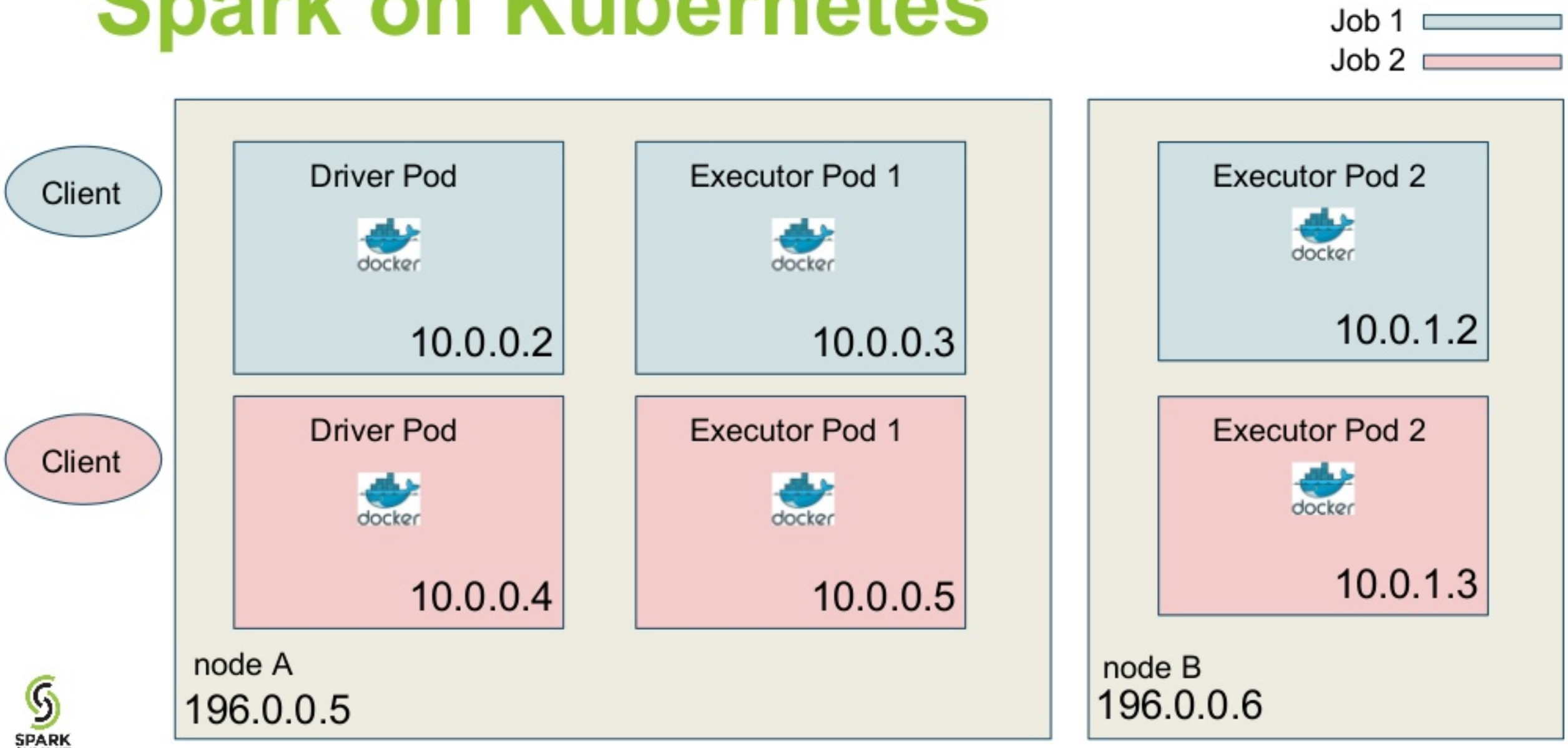
Big Data on Kubernetes

github.com/apache-spark-on-k8s

- Google, Haiwen, Hyperpilot, Intel, Palantir, Pepperdata, Red Hat, and growing.
- patching up Spark Driver and Executor code to work on Kubernetes.

Yesterday's talk:
spark-summit.org/2017/events/apache-spark-on-kubernetes/

Spark on Kubernetes



What about storage?

Spark often stores data on HDFS.

How can Spark on Kubernetes access HDFS data?

Hadoop Distributed File System

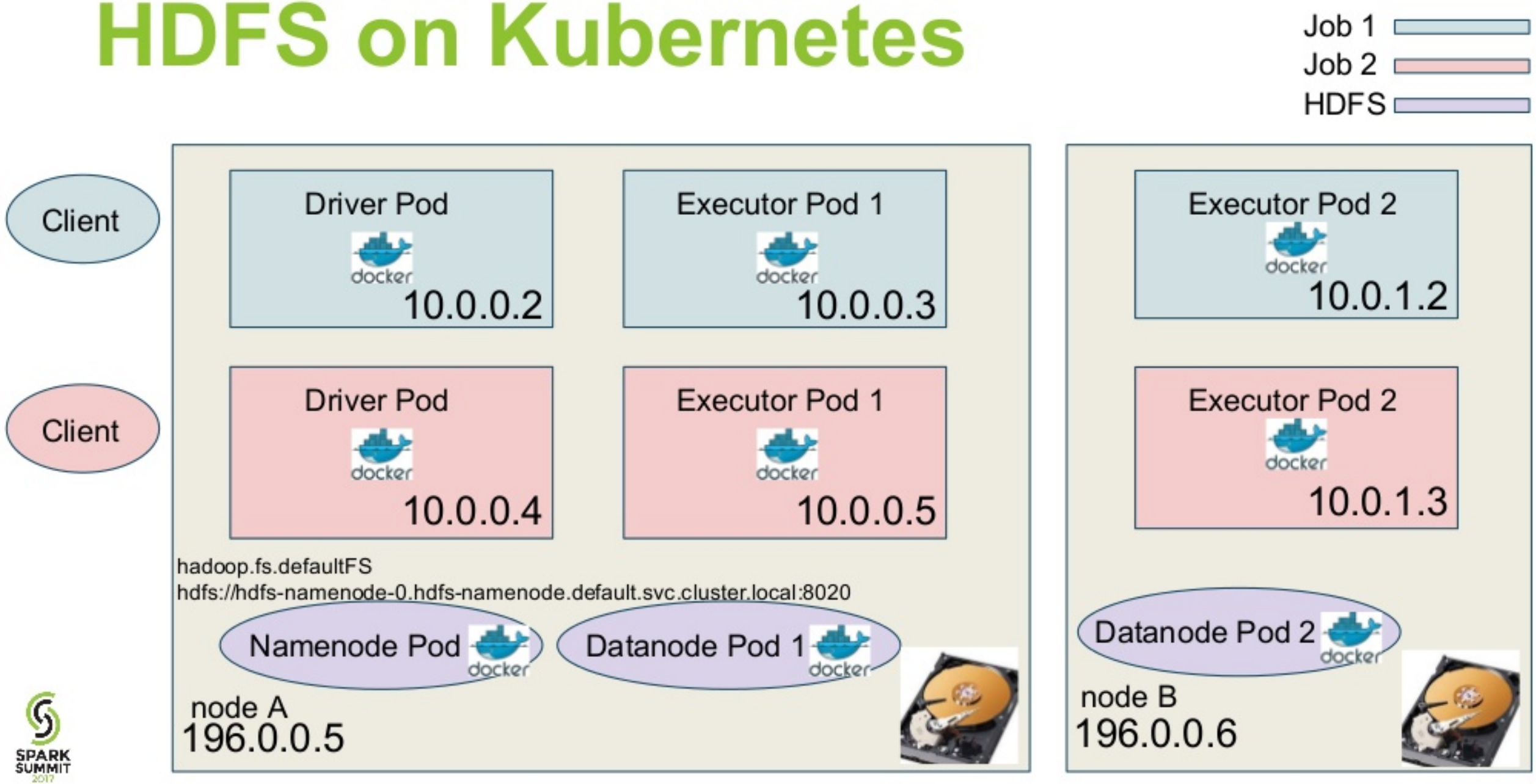
Namenode

- runs on a central cluster node.
- maintains file system metadata.

Datanodes

- on every cluster node.
- read and write file data on local disks.

HDFS on Kubernetes



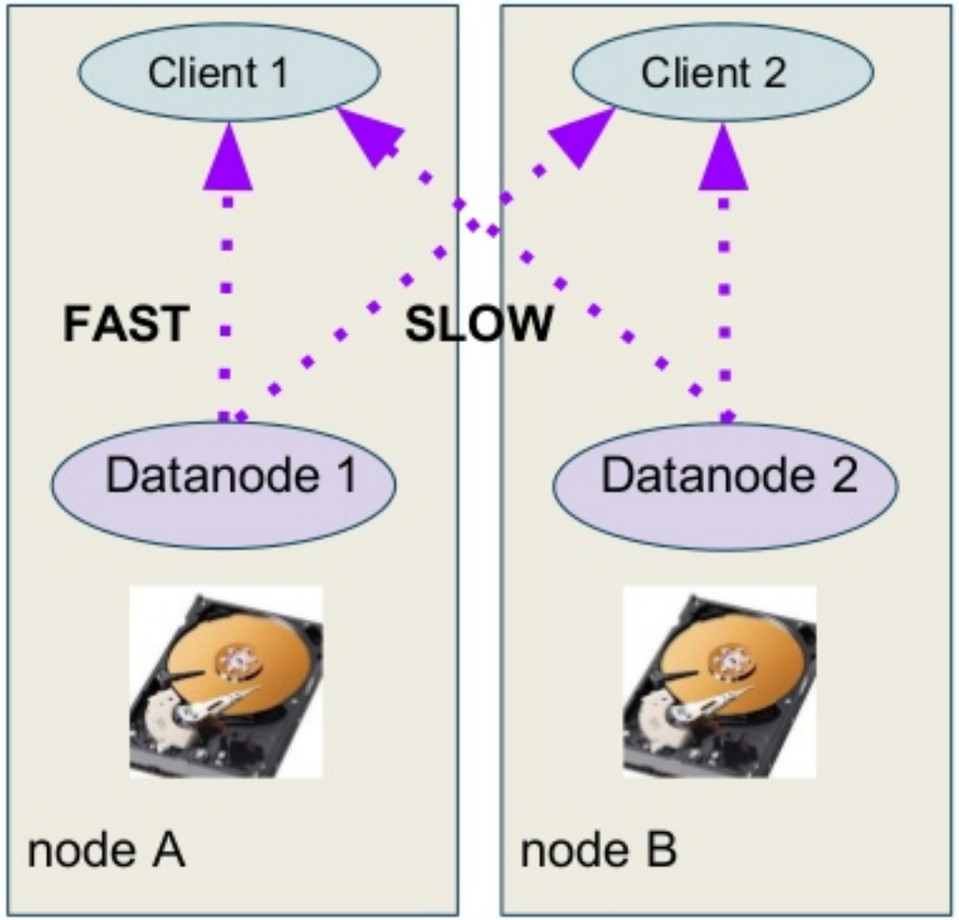
Demo

1. Label cluster nodes
2. Stand up HDFS
3. Launch a Spark job
4. Check Spark job output

What about data locality?

- Read data from local disks when possible
- Remote reads can be slow if network is slow

- Implemented in HDFS daemons, integrated with app frameworks like Spark



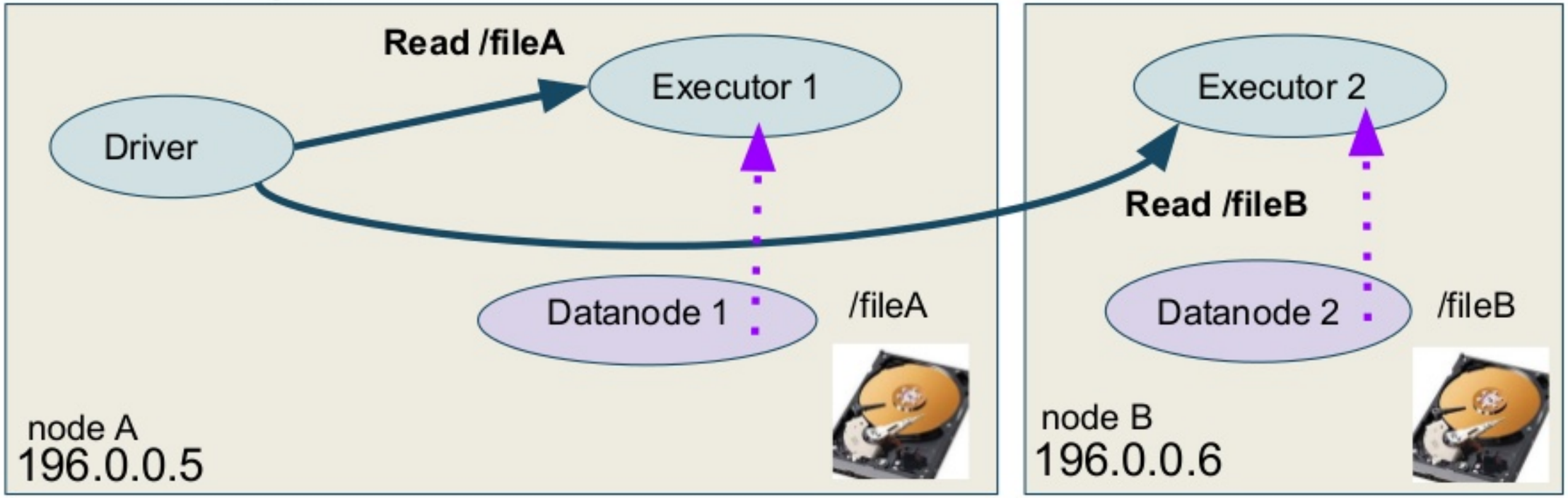
HDFS data locality in YARN

Job 1
HDFS

Spark driver sends tasks to right executors with tasks' HDFS data.

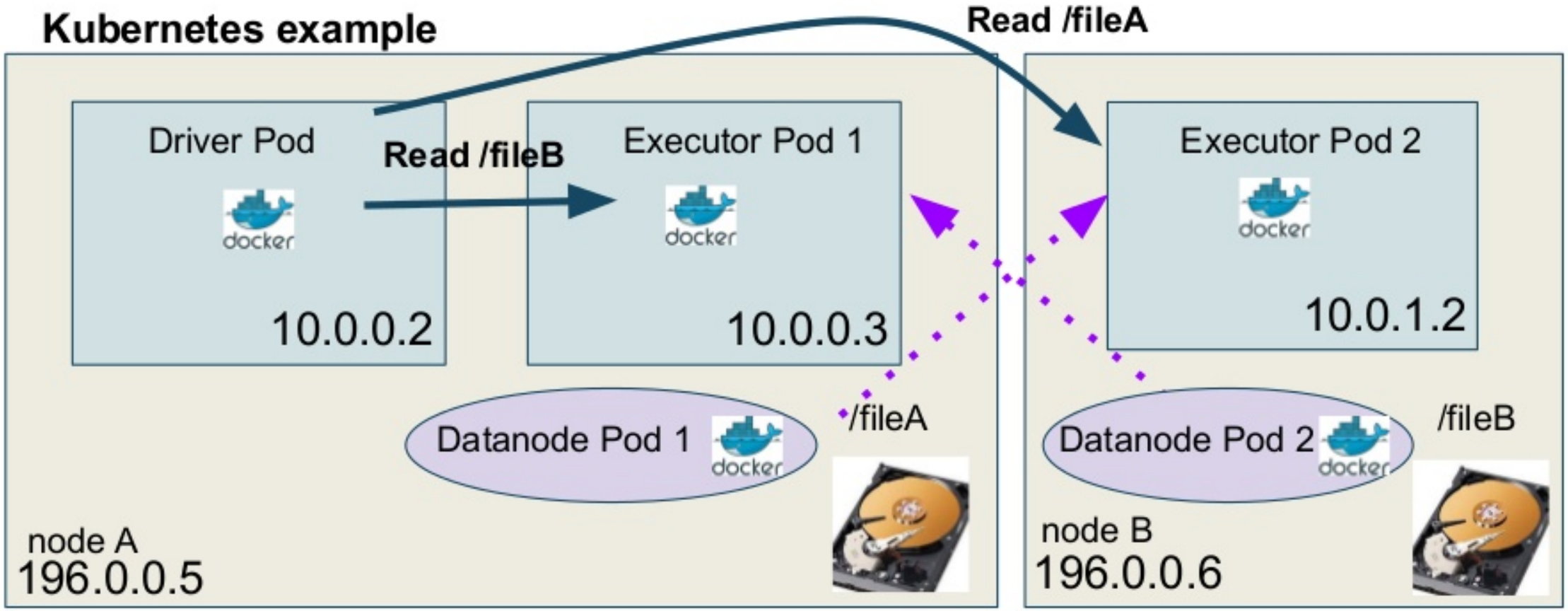
(/fileA → Datanode 1 → 196.0.0.5) == (Executor 1 →196.0.0.5)

YARN example



Hmm, how do I find right executors in Kubernetes...

(/fileA → Datanode 1 → 196.0.0.5) != (Executor 1 → **10.0.0.3**)

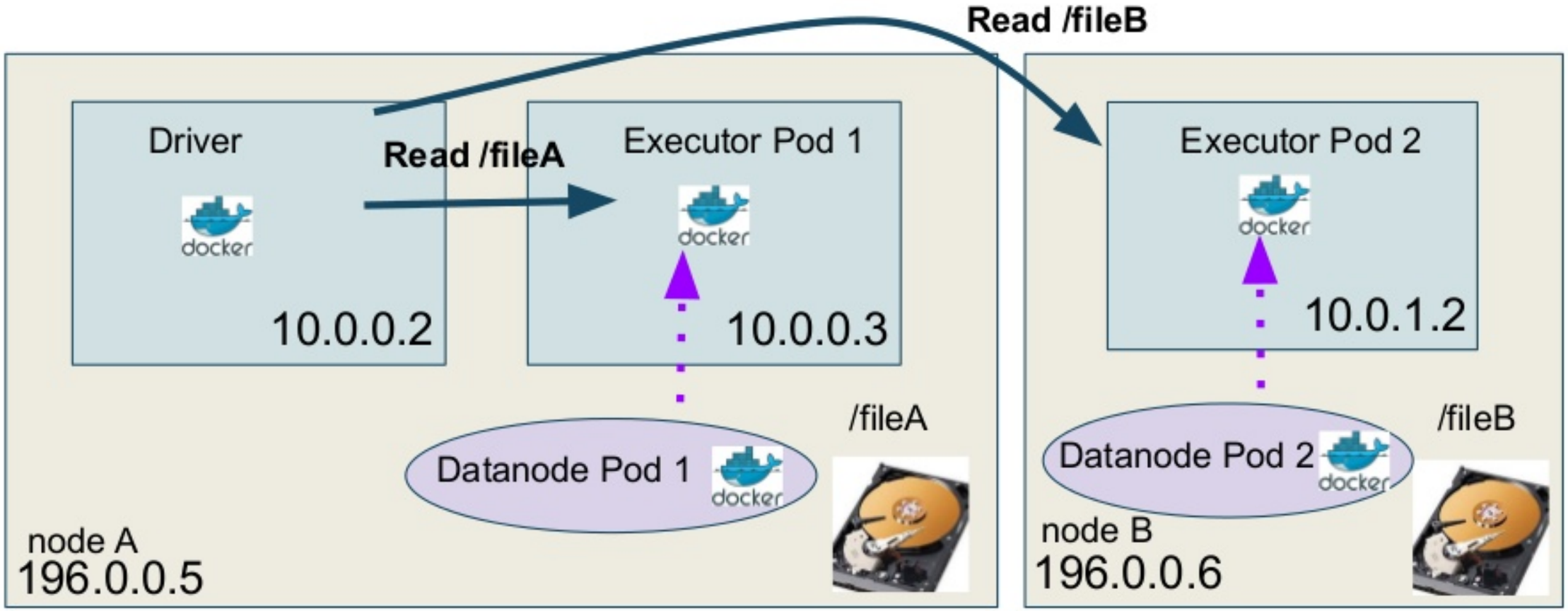


Fix Spark Driver code

1. Ask Kubernetes master to find the cluster node where the executor pod is running.
2. Get the node IP.
3. Compare with the datanode IPs.

Rescued data locality

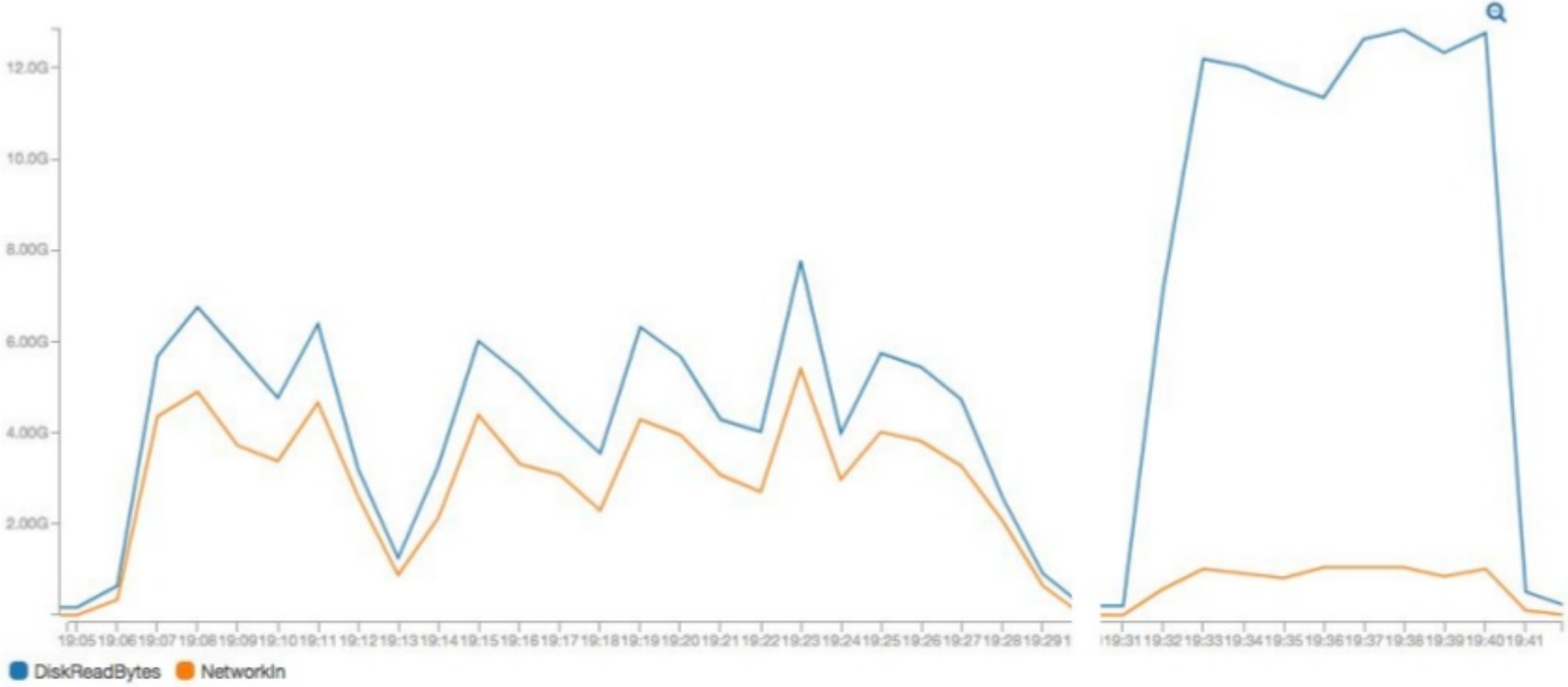
(/fileA → Datanode 1 → 196.0.0.5) == (Executor 1 →**10.0.0.3** → **196.0.0.5**)



Rescued data locality!

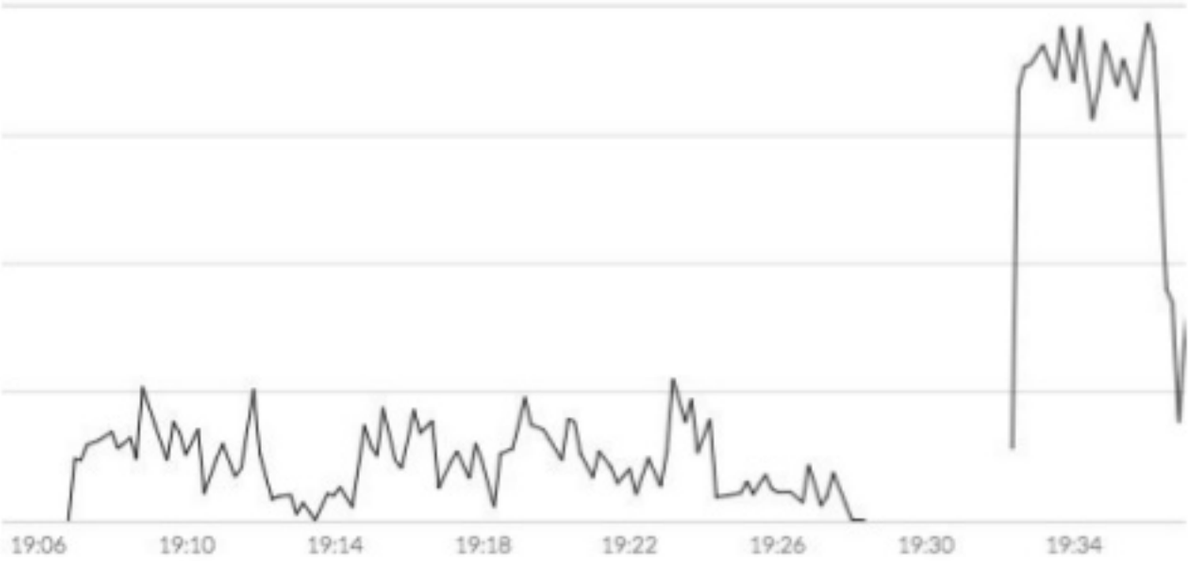
without data locality fix
- duration: 25 minutes

with data locality fix
- duration: 10 minutes



pepperdata kube-test Dashboard Charts Reports Tables Alarms

Summary of HDFS Local Read Bytes Sec



Summary of HDFS Remote Read Bytes Sec

SPARK SUMMIT 2017

Recap

Got HDFS up and running.

Basic data locality works.

Open problems:

- Remaining data locality issues -- rack locality, node preference, etc
- Kerberos support
- Namenode High Availability

Join us!

- github.com/apache-spark-on-k8s
- pepperdata.com/careers/

More questions?

- Come to Pepperdata booth #101
- Mail kimoon@pepperdata.com