Apache Spark-on-YARN:
Empower Spark Applications on Hadoop Cluster

2014 Hadoop Summit, San Jose, California

PRESENTED BY **Andy Feng (@afeng76), Thomas Graves** | June 5, 2014

Agenda

- Apache Spark
- Spark-on-YARN
- Yahoo Use Cases

YAHOO!

What is Apache Spark?

- Fast and expressive cluster computing system compatible with Apache Hadoop
- Improves efficiency via
  - General execution graphs
  - In-memory storage
- Improves usability via
  - Rich APIs in Java, Scala, Python
  - Interactive shell
- Runs standalone, on YARN, on Mesos, and on Amazon EC2

Apache Spark – Key Idea

- RDD: resilient distributed dataset
  - Collections of objects spread across a cluster
  - Built through parallel transformations
  - Automatically rebuilt on failure
  - Controllable persistence
- Write programs in terms of RDD transformations and actions

Base RDD from HDFS

```
lines = spark.textFile(“hdfs://...”)
errors =  lines.filter(_.startsWith("Error"))
messages = errors.map(_.split('\t')(2))
```

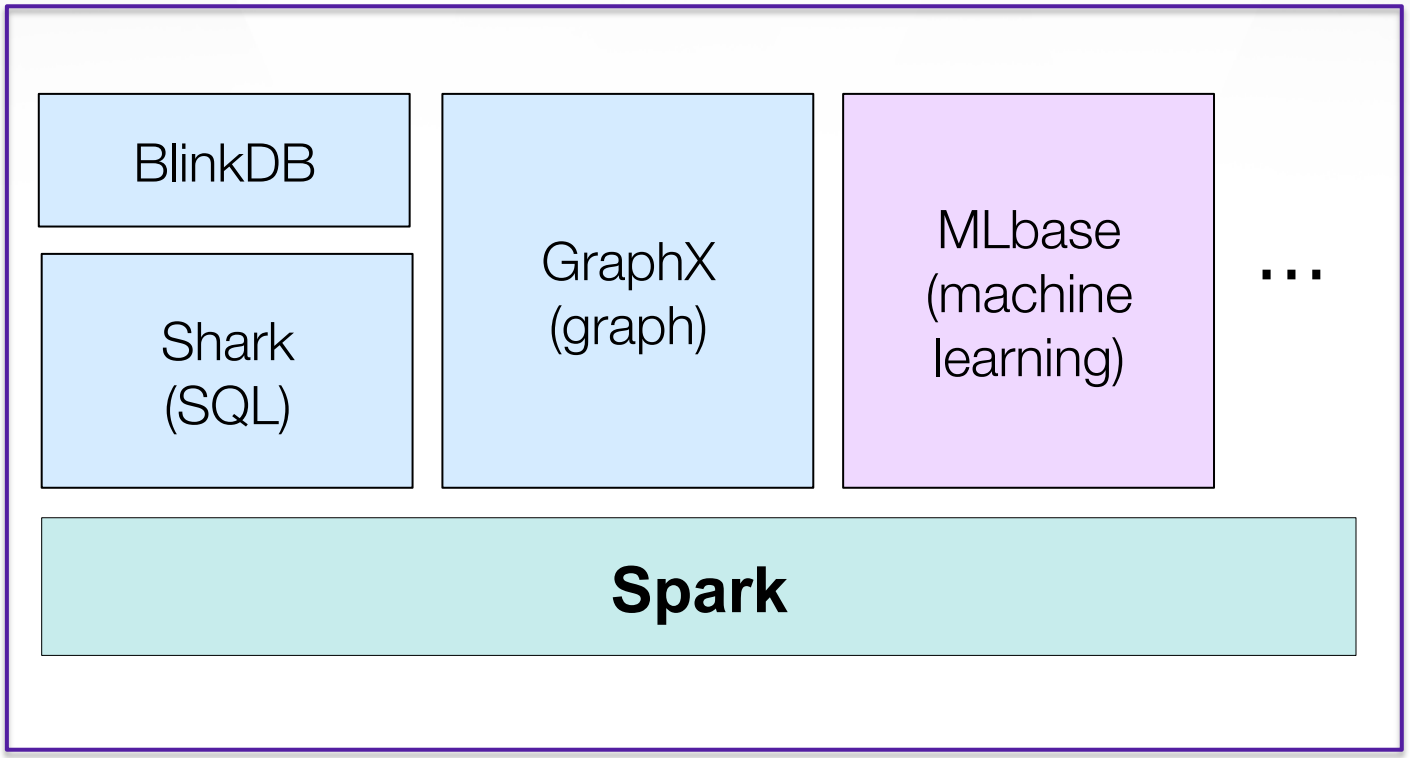
RDD in memory

```
messages.cache()
```

Iterative processing

```
for (str <- Array(“foo”, “bar”))
   messages.filter(_.contains(str)).count()
```

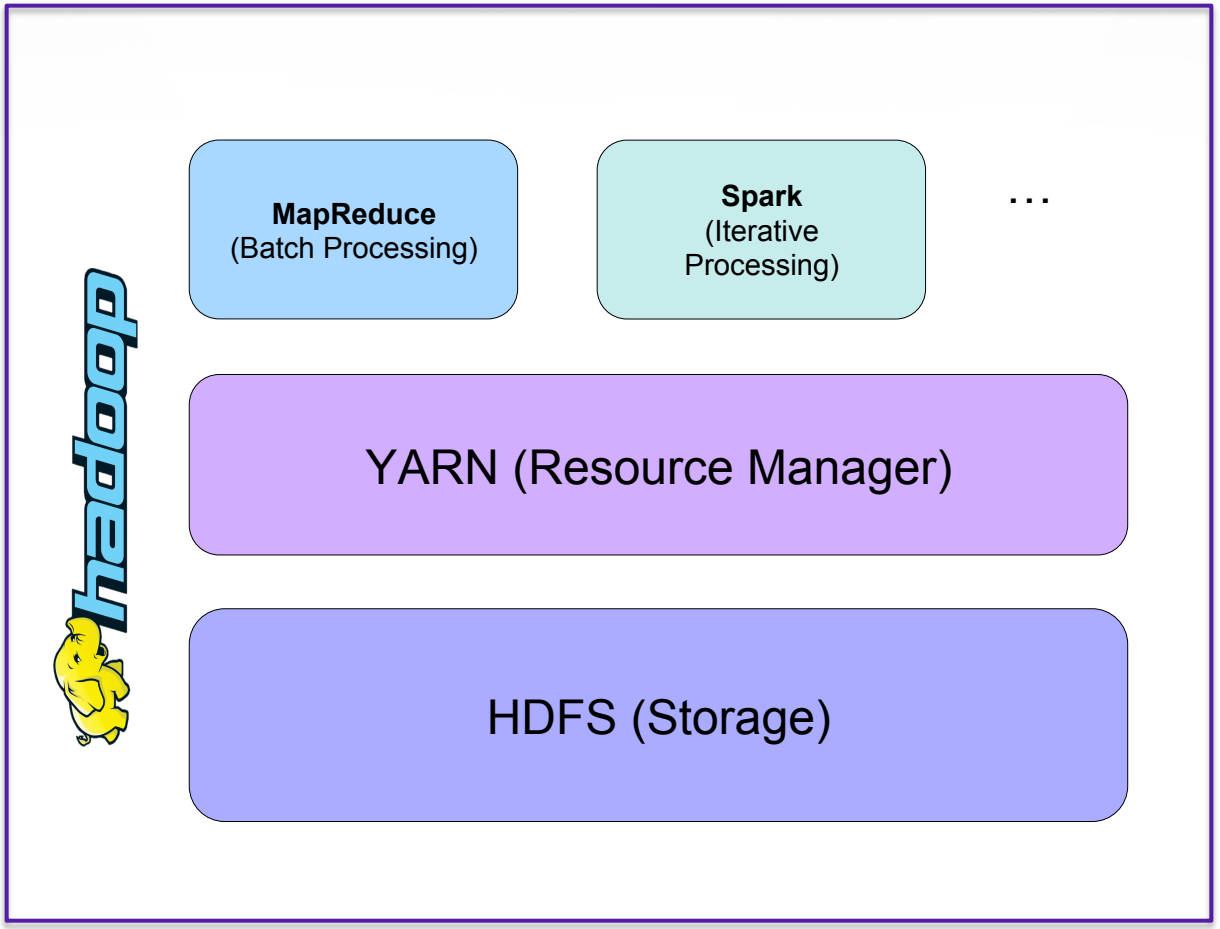
Projects Built on Spark




```
val tweets = spark.textFile(“hdfs://…”)
val points = sql(
  “select latitude, longitude from tweets”)
val model = KMeans.train(points, 10, 50)
```

YAHOO!

Spark-on-YARN: Hadoop + Spark



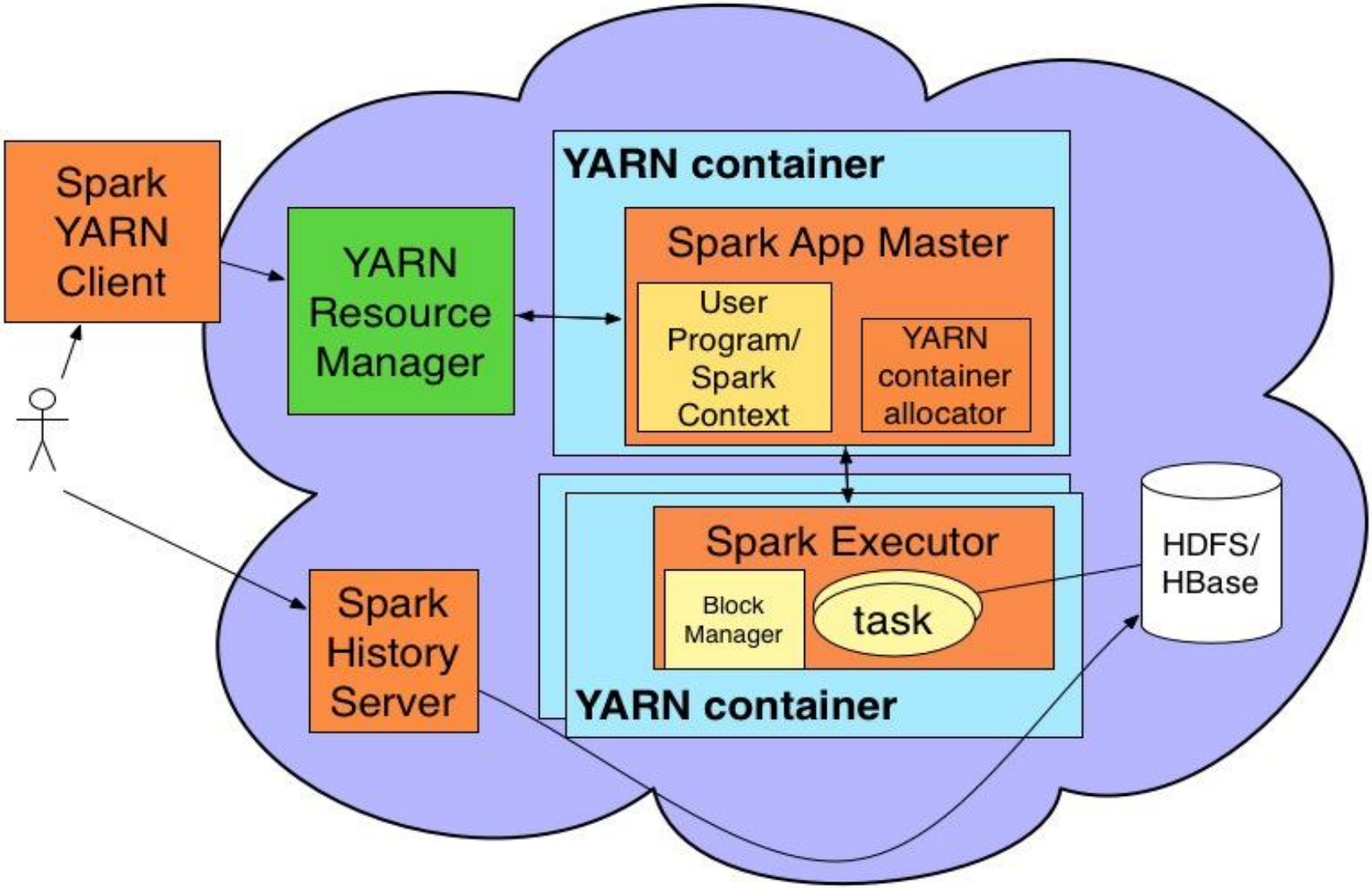
Why Spark-on-YARN?

- Access to HDFS dataset
  - Ex. 150PB data on Yahoo Hadoop clusters
- Leverage existing Hadoop clusters
  - Ex. 32,000 compute nodes in Yahoo
  - Familiar by Hadoop users
  - No extra deployment and operation costs
- Easy to get started

Spark-on-YARN: Features

- YARN-client and YARN-cluster modes
- Integration w/ Spark history server
- Secure HDFS access
- Authentication between Spark components
- Support running application JARs in HDFS
- Hadoop Distributed cache (files/archives) supported
- Hadoop 0.23 and Hadoop 2.x supported

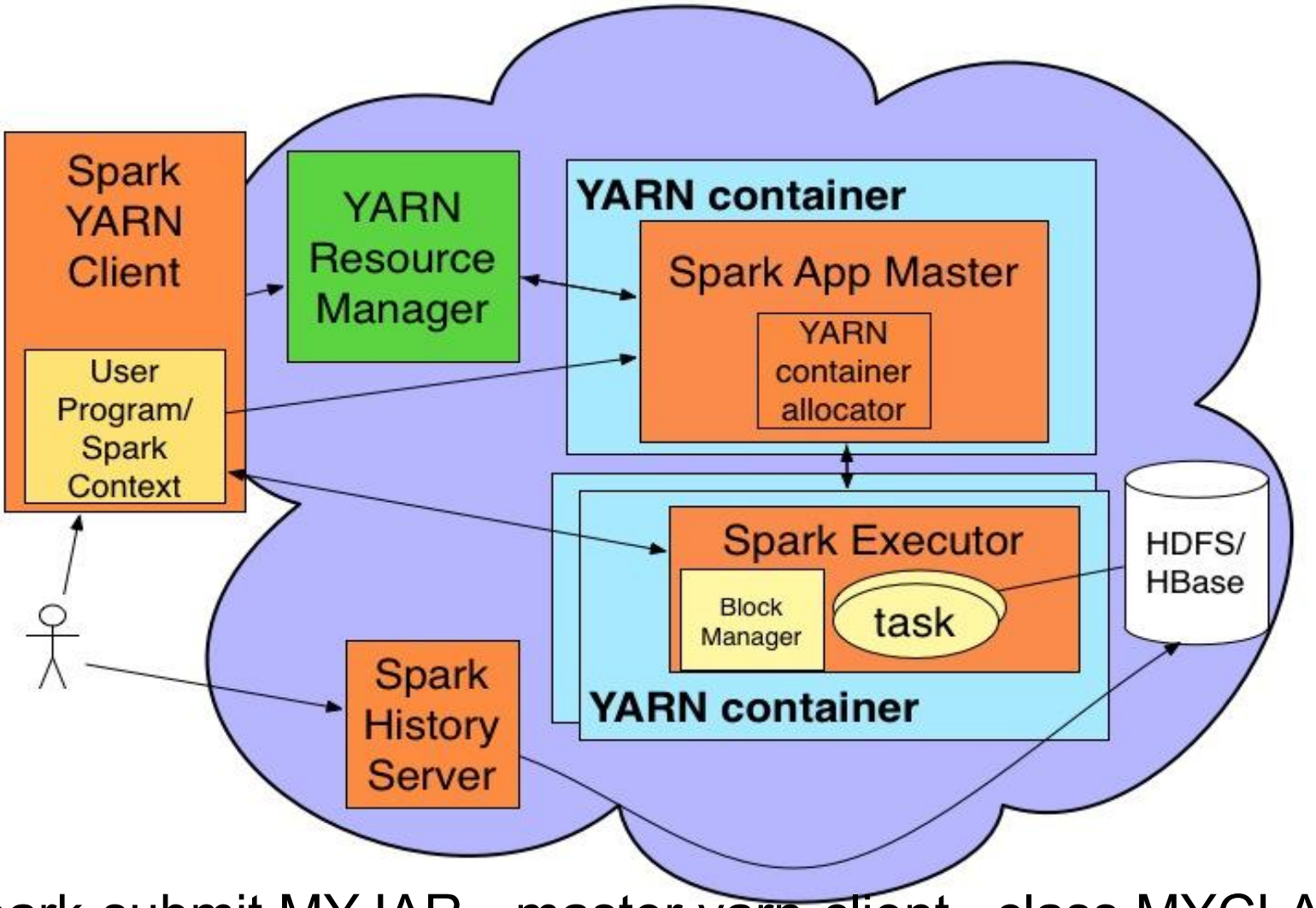
Spark-on-YARN: Cluster Mode



spark-submit MYJAR --master yarn-cluster –class MYCLASS

YAHOO!

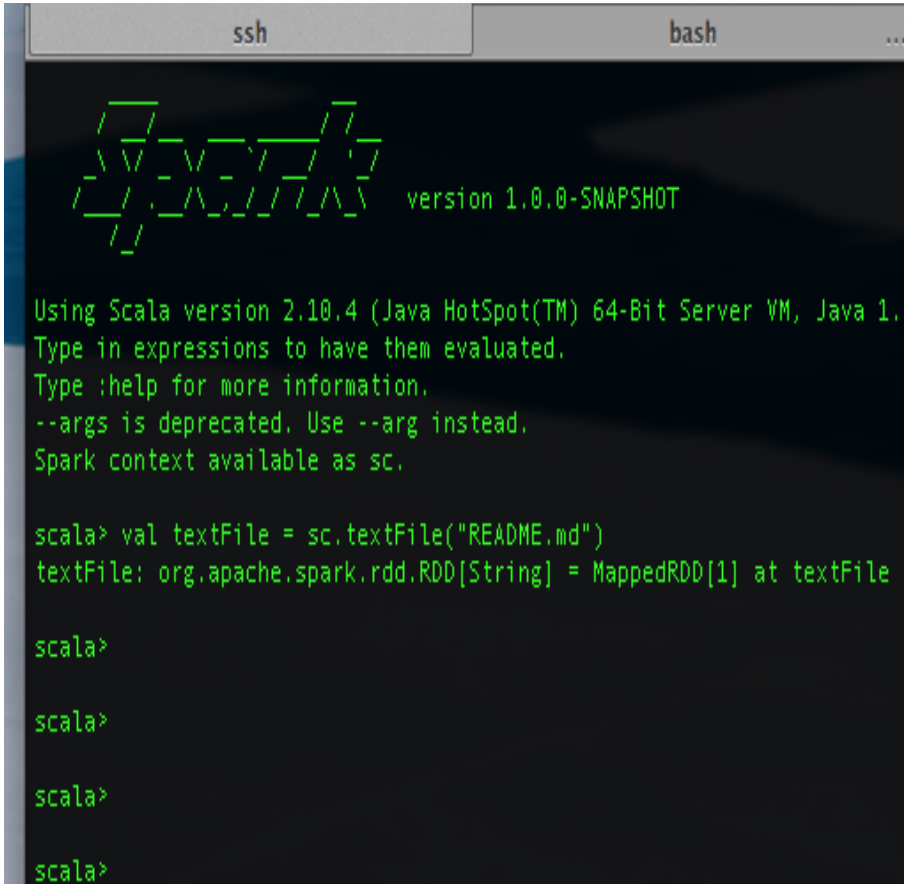
Spark-YARN: Client Mode



spark-submit MYJAR --master yarn-client –class MYCLASS

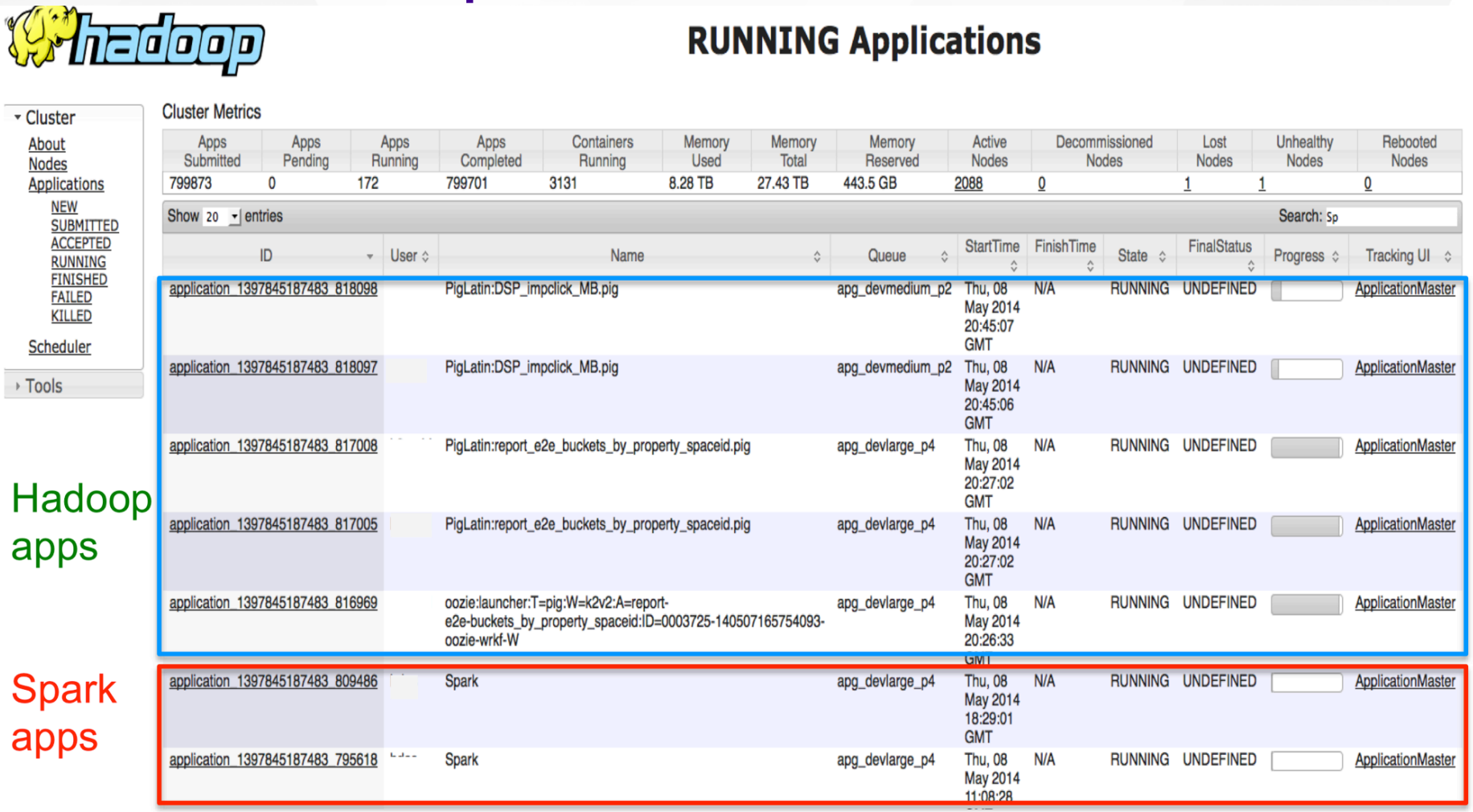
Interactive Shell on YARN

- Spark shell
  - Easy way to learn spark
  - Great for adhoc queries
- Launch shell w/ Spark-on-YARN
  - Scala
    - MASTER=yarn-client spark-shell
  - Python
    - MASTER=yarn-client pyspark



YAHOO!

Spark-on-YARN UI



Hadoop apps

Spark apps

YAHOO!

Spark-on-YARN UI

Spark Stages

Total Duration: 2.36 h
Scheduling Mode: FIFO
Active Stages: 1
Completed Stages: 174
Failed Stages: 0

Active Stages (1)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Shuffle Read	Shuffle Write
174	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:50:16	23.8 s	295/700		

Completed Stages (174)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Shuffle Read	Shuffle Write
173	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:49:12	1.1 m	700/700		
172	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:47:57	1.3 m	700/700		
171	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:46:39	1.3 m	700/700		
170	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:45:46	52.3 s	700/700		
169	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:44:56	49.8 s	700/700		
168	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:44:07	48.5 s	700/700		
167	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:43:21	46.6 s	700/700		
166	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:42:34	46.1 s	700/700		
165	reduce at SparkFastDecisionTreeAlgorithm.scala:703	2014/05/08 20:42:05	29.2 s	700/700		
164	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:41:21	43.6 s	700/700		
163	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:40:12	1.2 m	700/700		

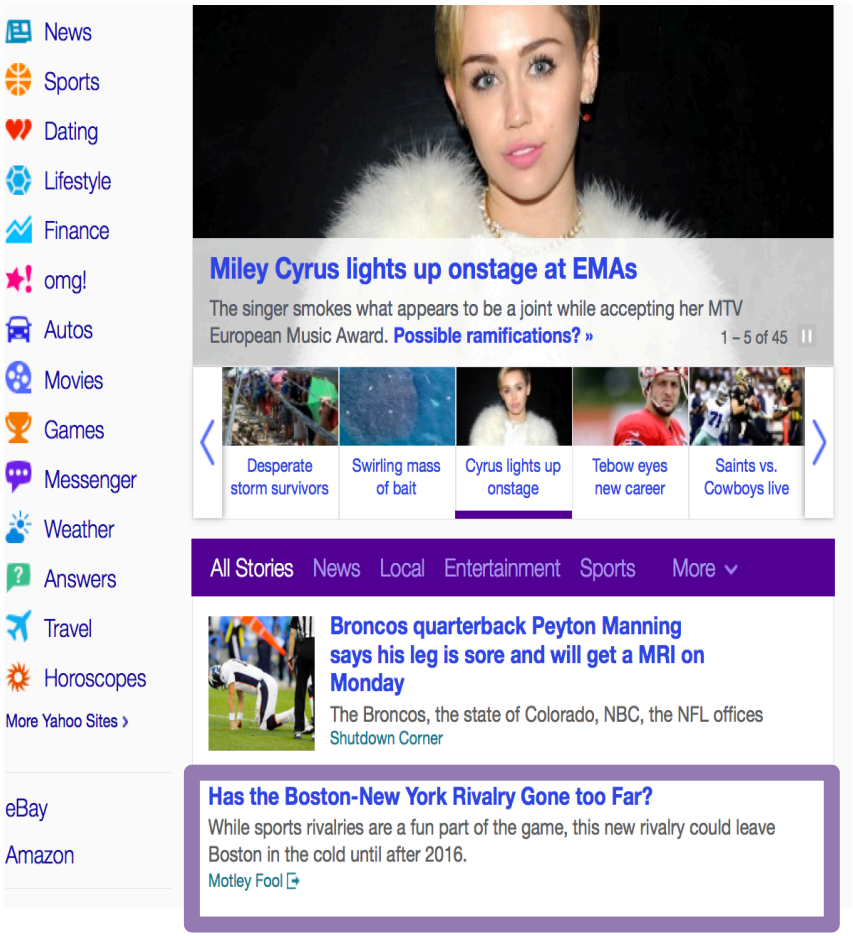
YAHOO!

Spark-on-YARN: Upcoming

- Support long running jobs
  - Ex. long-lived machine learning jobs
  - Ex. Shark
- Dynamic resource allocation
  - # of containers
- Integrate with Hadoop enhancements
  - Generic History Server (*Timeline Server*)
  - Pre-emption
  - …

YAHOO!

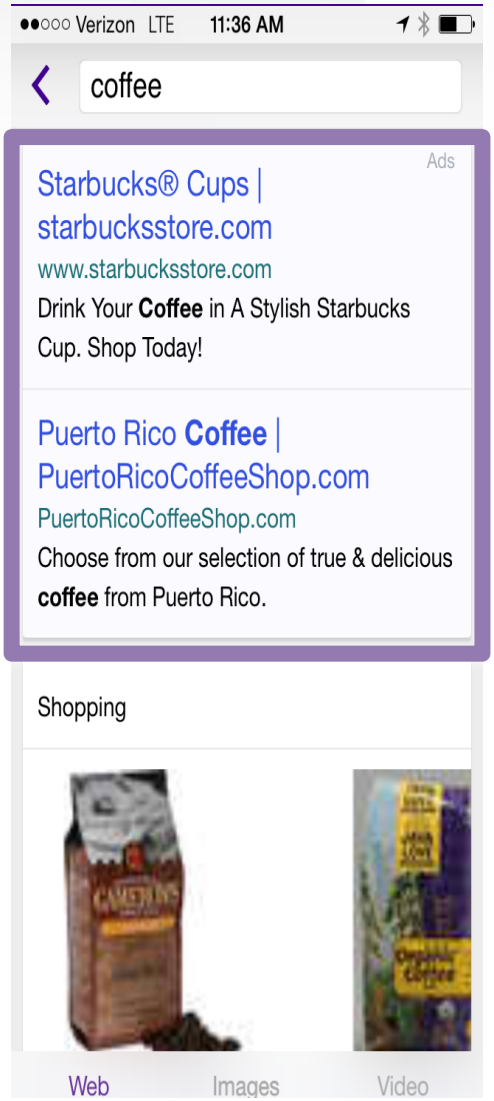
Use Case: Yahoo Native Ads POC



- Logistic regression algorithm
  - 120 LOC in Spark/Scala
  - 30 min. on model creation for 100M samples and 13K features
- Initial version launched within 2 hours after Spark/YARN announcement
  - Compared: Several days on hardware acquisition, system setup and data movement

YAHOO!

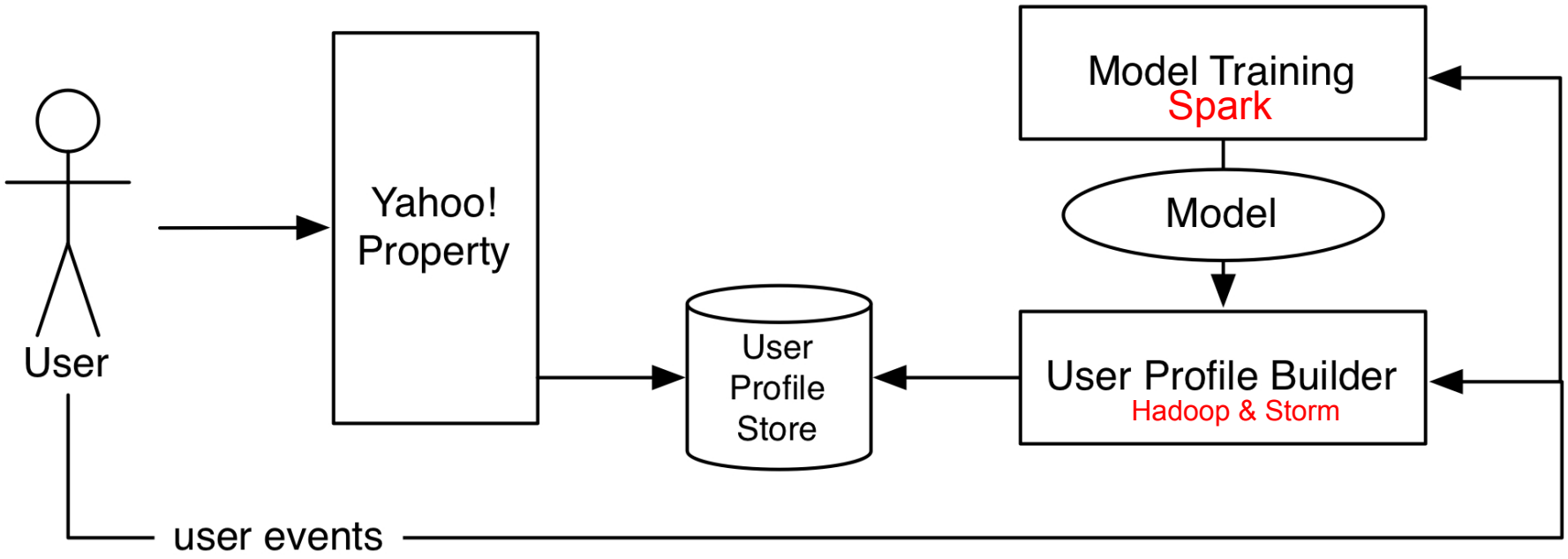
Use Case: Mobile Search Ads POC



- Learn from mobile search ads clicks data
  - 600M labeled examples on HDFS
  - 100M sparse features
- Spark programs for Gradient Boosting Decision Trees
  - 6 hours for model training with 100 workers
  - Model w/ accuracy very close to heavily-manually-tuned Logistic Regression models
- See us @ Spark Summit 2014
  - M. Amde, H. Das, E. Sparks, and A. Talwalkar: “Scalable Distributed Decision Trees in Spark MLLib”

YAHOO!

Use Case: Audience Expansion for Ad Targeting



- G. Li, J. Kim, and A. Feng: “Yahoo Audience Expansion: Migration from Hadoop Streaming to Spark”, Spark Summit 2013.

YAHOO!

Acknowledgement

- Spark-on-YARN contributors
  - Matei and his team at DataBricks & AMPLab
  - Mridul, Thomas, Bobby, Paul etc. from Yahoo
  - Sandy, Marcelo etc. from Cloudera

- Spark-on-YARN users
  - Hirakendu, Amit, Gavin, Jaebong and many Yahoo users
  - Production launch in Taobao

YAHOO!

Thank You

@afeng76

We are hiring!

Stop by **Kiosk P9**
or reach out to us at
bigdata@yahoo-inc.com.