



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

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))
```

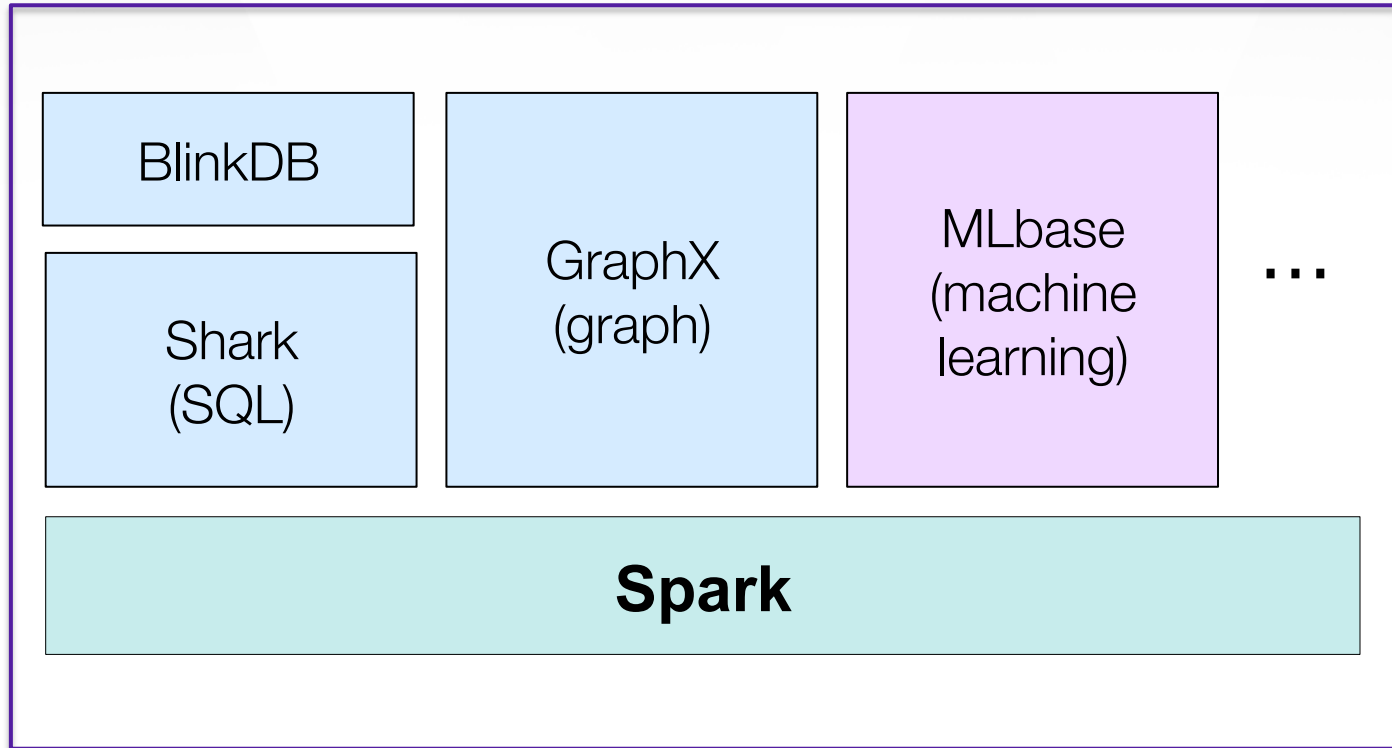
RDD in memory

```
messages.cache()
```

Iterative processing

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

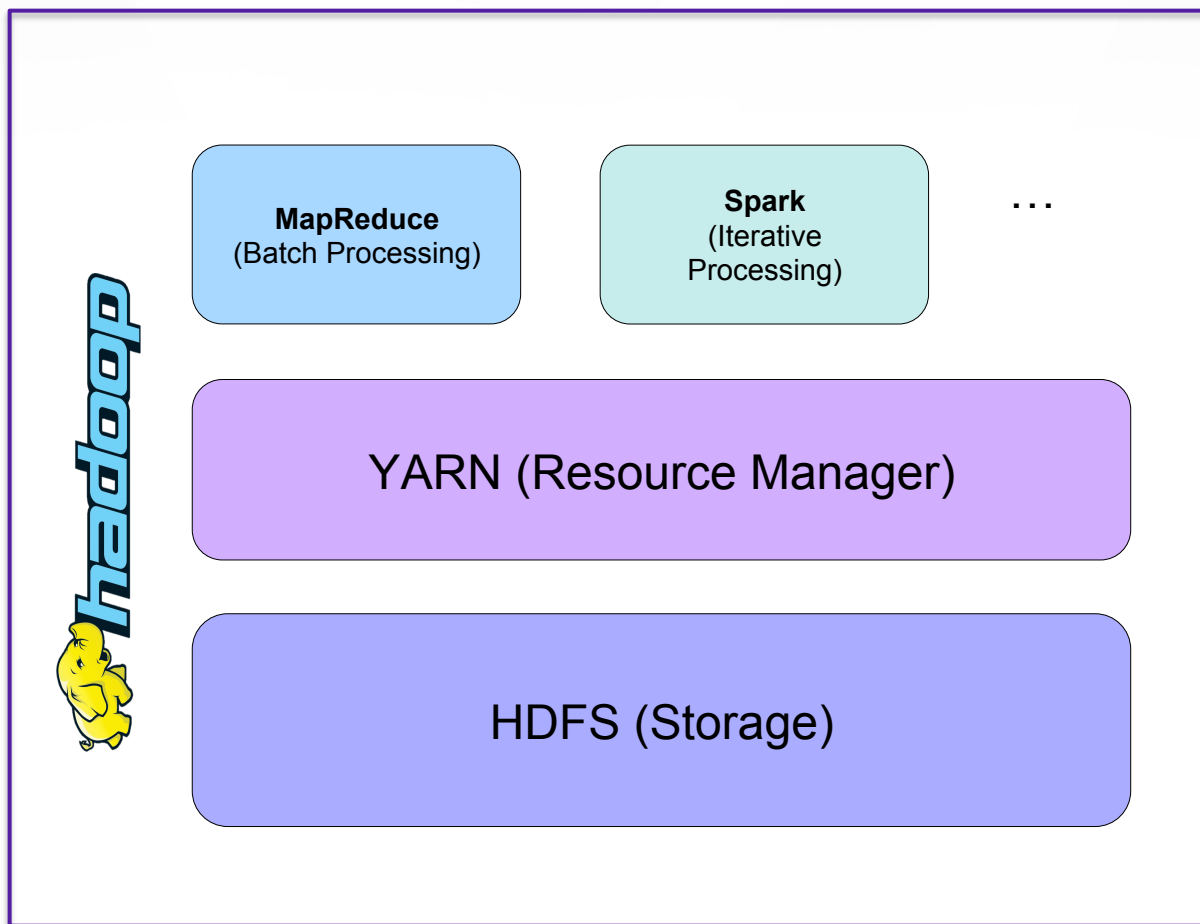
Projects Built on Spark



```
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



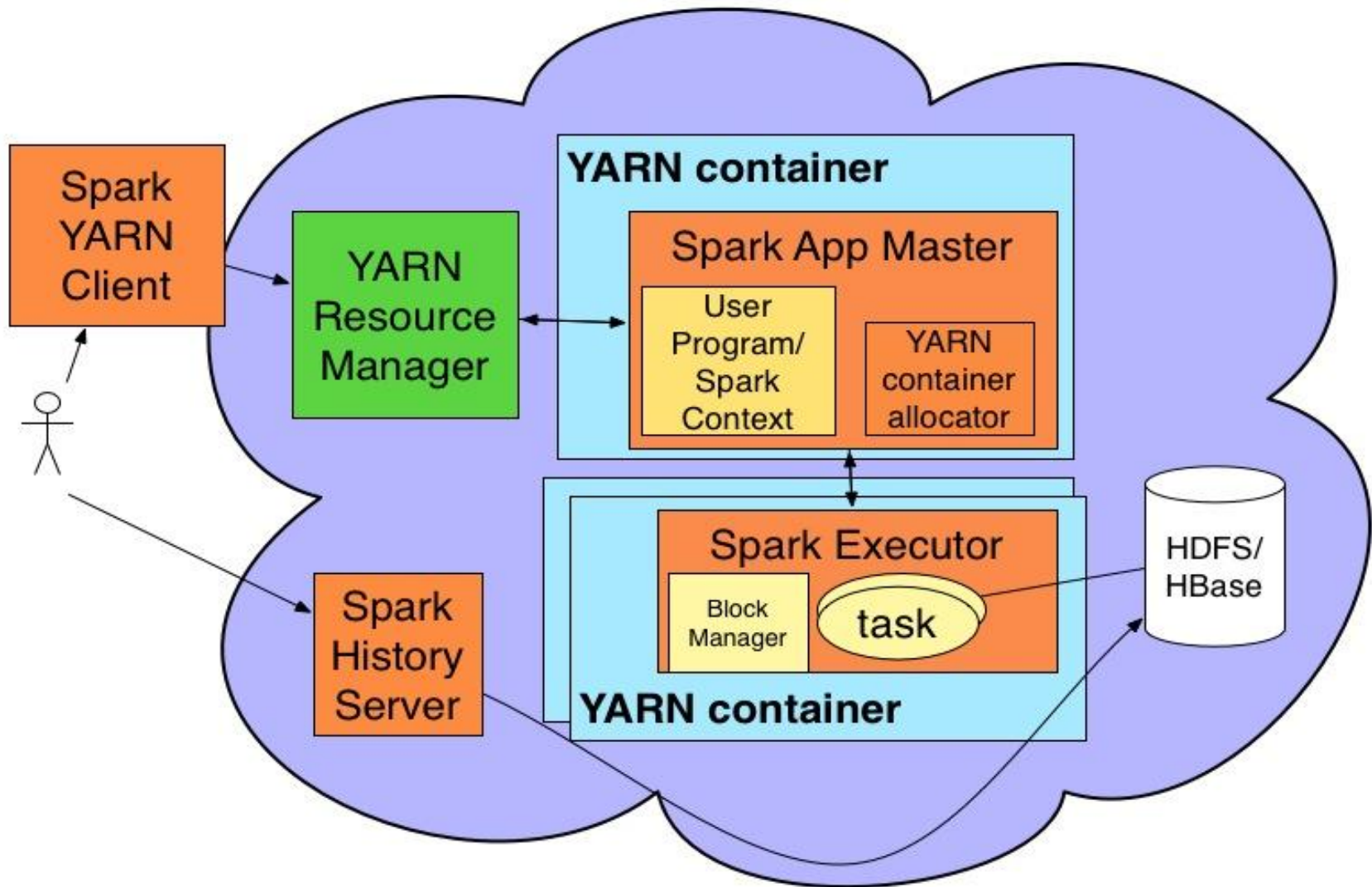
Why Spark-on-YARN?

- 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

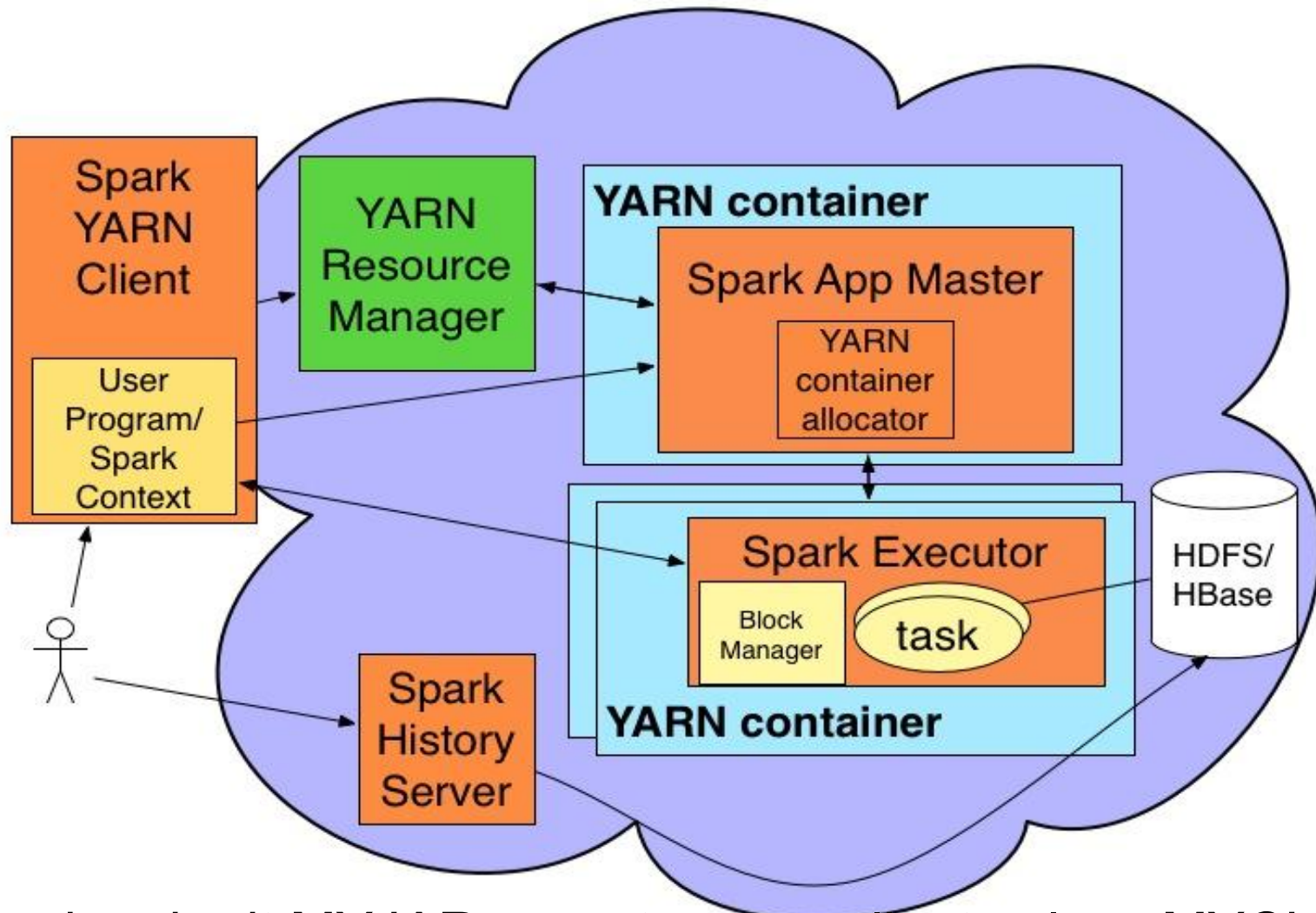
- YARN-client and YARN-cluster modes
- Integration w/ Spark history server
- Secure HDFS access
- Authentication between Spark components
- Support running application JARs in HDFS
- Hadoop Distributed cache (files/archives) supported
- Hadoop 0.23 and Hadoop 2.x supported

Spark-on-YARN: Cluster Mode



`spark-submit MYJAR --master yarn-cluster --class MYCLASS`
YAHOO!

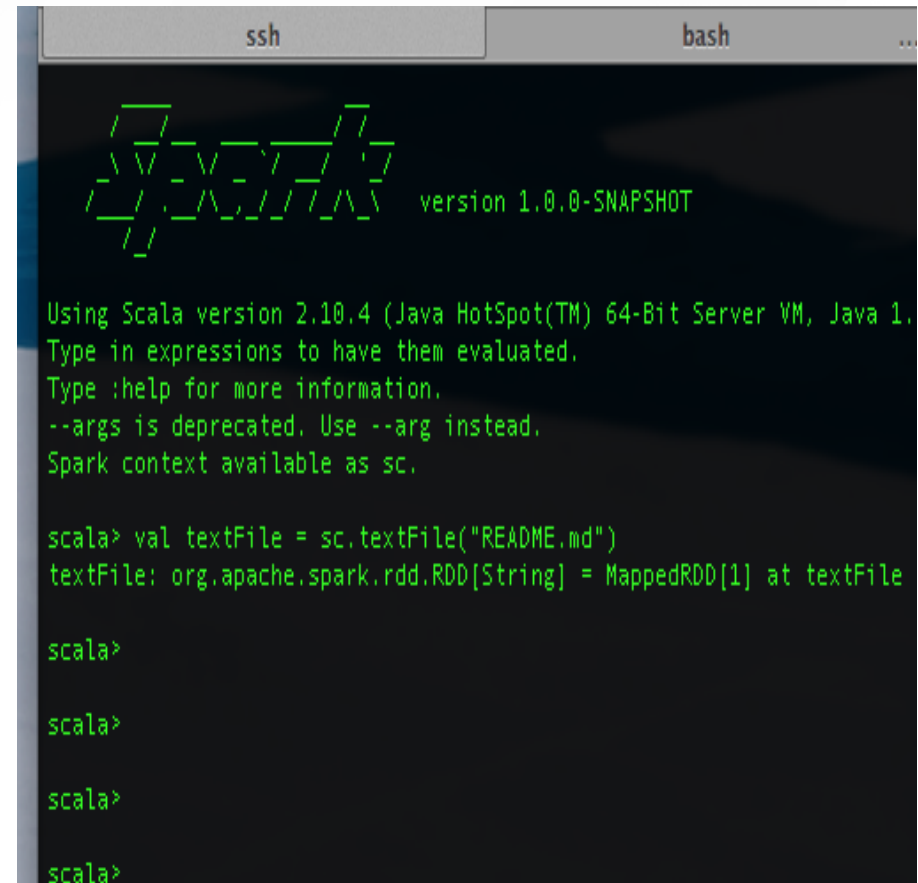
Spark-YARN: Client Mode



spark-submit MYJAR --master yarn-client --class MYCLASS
YAHOO!

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



The screenshot shows a terminal window with two tabs: 'ssh' and 'bash'. The terminal displays the Spark logo and the text 'version 1.0.0-SNAPSHOT'. Below this, it says 'Using Scala version 2.10.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_75)'. It then provides instructions: 'Type in expressions to have them evaluated.', 'Type :help for more information.', and '--args is deprecated. Use --arg instead.'. It also states 'Spark context available as sc.'. The prompt 'scala>' is shown, followed by the command 'val textFile = sc.textFile("README.md")'. The output is 'textFile: org.apache.spark.rdd.RDD[String] = MappedRDD[1] at textFile'. The prompt 'scala>' is shown again, followed by another 'scala>' prompt.

```
ssh bash ...  
  
Spark version 1.0.0-SNAPSHOT  
  
Using Scala version 2.10.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_75)  
Type in expressions to have them evaluated.  
Type :help for more information.  
--args is deprecated. Use --arg instead.  
Spark context available as sc.  
  
scala> val textFile = sc.textFile("README.md")  
textFile: org.apache.spark.rdd.RDD[String] = MappedRDD[1] at textFile  
  
scala>  
  
scala>  
  
scala>
```

Spark-on-YARN UI

RUNNING Applications



- Cluster
 - About
 - Nodes
 - Applications
 - NEW
 - SUBMITTED
 - ACCEPTED
 - RUNNING
 - FINISHED
 - FAILED
 - KILLED
 - Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
799873	0	172	799701	3131	8.28 TB	27.43 TB	443.5 GB	2088	0	1	1	0

Show 20 entries										Search: sp	
ID	User	Name	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI		
application_1397845187483_818098		PigLatin:DSP_impclick_MB.pig	apg_devmedium_p2	Thu, 08 May 2014 20:45:07 GMT	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster		
application_1397845187483_818097		PigLatin:DSP_impclick_MB.pig	apg_devmedium_p2	Thu, 08 May 2014 20:45:06 GMT	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster		
application_1397845187483_817008		PigLatin:report_e2e_buckets_by_property_spaceid.pig	apg_devlarge_p4	Thu, 08 May 2014 20:27:02 GMT	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster		
application_1397845187483_817005		PigLatin:report_e2e_buckets_by_property_spaceid.pig	apg_devlarge_p4	Thu, 08 May 2014 20:27:02 GMT	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster		
application_1397845187483_816969		oozie:launcher:T=pig:W=k2v2:A=report-e2e-buckets_by_property_spaceid:ID=0003725-140507165754093-oozie-wrkf-W	apg_devlarge_p4	Thu, 08 May 2014 20:26:33 GMT	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster		
application_1397845187483_809486		Spark	apg_devlarge_p4	Thu, 08 May 2014 18:29:01 GMT	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster		
application_1397845187483_795618		Spark	apg_devlarge_p4	Thu, 08 May 2014 11:08:28 GMT	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster		

Hadoop apps

Spark apps

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	<div><div>295/700</div></div>		

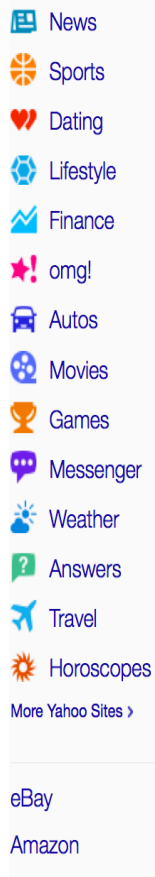
Completed Stages (174)

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	<div><div>700/700</div></div>		
172	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:47:57	1.3 m	<div><div>700/700</div></div>		
171	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:46:39	1.3 m	<div><div>700/700</div></div>		
170	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:45:46	52.3 s	<div><div>700/700</div></div>		
169	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:44:56	49.8 s	<div><div>700/700</div></div>		
168	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:44:07	48.5 s	<div><div>700/700</div></div>		
167	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:43:21	46.6 s	<div><div>700/700</div></div>		
166	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:42:34	46.1 s	<div><div>700/700</div></div>		
165	reduce at SparkFastDecisionTreeAlgorithm.scala:703	2014/05/08 20:42:05	29.2 s	<div><div>700/700</div></div>		
164	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:41:21	43.6 s	<div><div>700/700</div></div>		
163	reduce at SparkFastDecisionTreeAlgorithm.scala:448	2014/05/08 20:40:12	1.2 m	<div><div>700/700</div></div>		

Spark-on-YARN: Upcoming

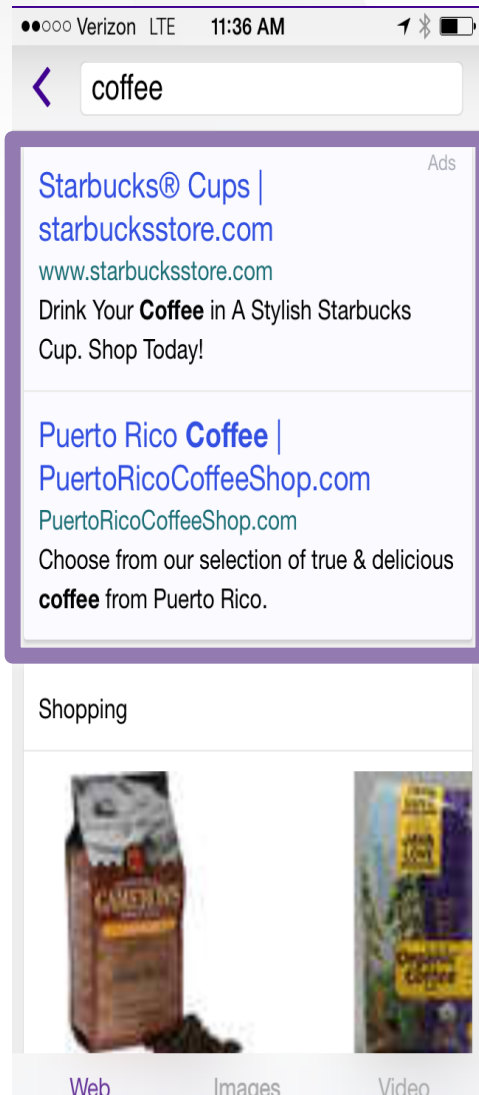
- Support long running jobs
 - Ex. long-lived machine learning jobs
 - Ex. Shark
- Dynamic resource allocation
 - # of containers
- Integrate with Hadoop enhancements
 - Generic History Server (*Timeline Server*)
 - Pre-emption
 - ...

Use Case: Yahoo Native Ads POC



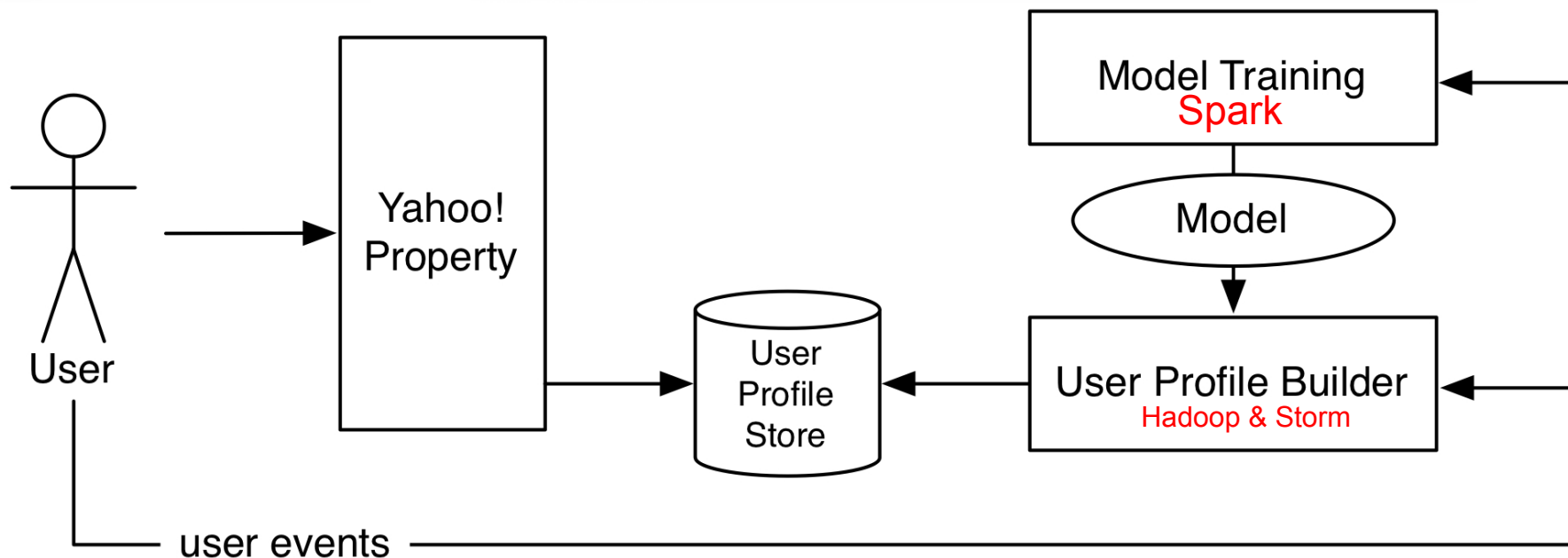
- Logistic regression algorithm
 - 120 LOC in Spark/Scala
 - 30 min. on model creation for 100M samples and 13K features
- Initial version launched within 2 hours after Spark/YARN announcement
 - Compared: Several days on hardware acquisition, system setup and data movement

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"

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.

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!

Committed to
pioneering new
grounds for
Apache Hadoop



Thank You

@afeng76

We are hiring!

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