cloudera®

Yarns about YARN: Migrating to MapReduce v2

Kathleen Ting, kate@cloudera.com

Miklos Christine, mwc@cloudera.com

Strata Hadoop San Jose, 19 February 2015

$ whoami

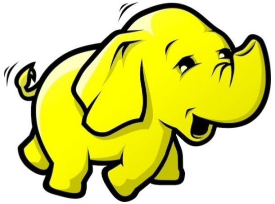
Kathleen Ting

- Joined Cloudera in 2011
- Former customer operations engineer
- Technical account manager
- *Apache Sqoop Cookbook* co-author

Miklos Christine

- Joined Cloudera 2013
- Former customer operations engineer
- Systems engineer
- Apache Spark expert

Cloudera and Apache Hadoop



- Apache Hadoop is an open source software framework for distributed storage and distributed processing of Big Data on clusters of commodity hardware.
- Cloudera is revolutionizing enterprise data management by offering the first unified Platform for Big Data, an enterprise data hub built on Apache Hadoop.
  - Distributes CDH, a Hadoop distribution.
  - Teaches, consults, and supports customers building applications on the Hadoop stack.
  - The world-wide Cloudera Customer Operations Engineering team has closed tens of thousands of support incidents over six years.

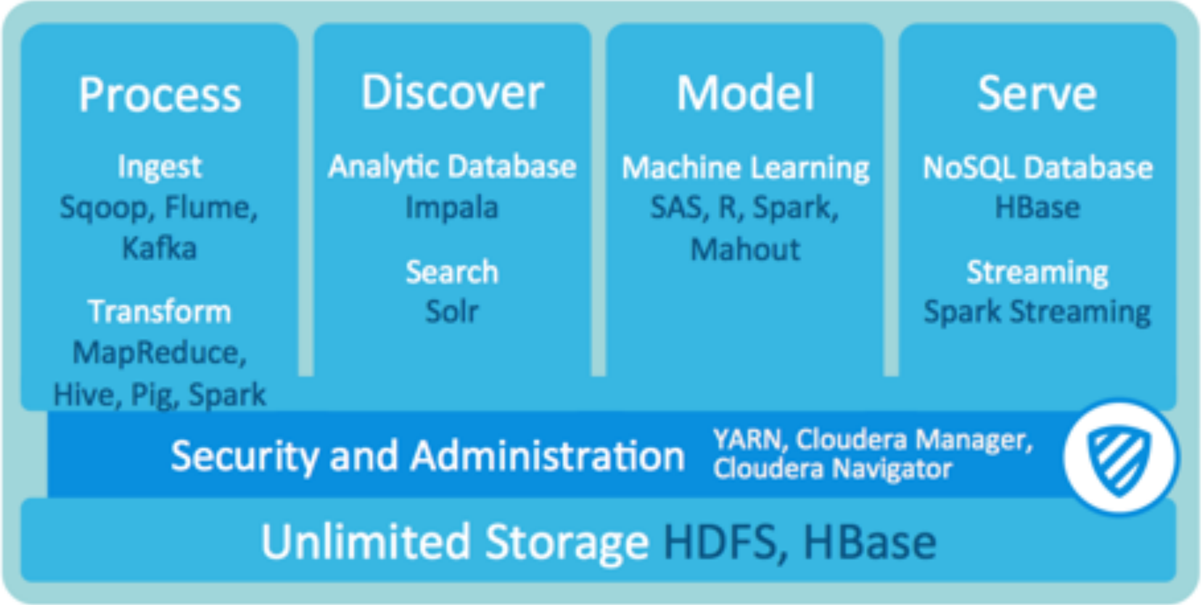
Outline

- YARN motivation
- Upgrading MR1 to MR2
- YARN upgrade pitfalls
- YARN applications

YARN motivation

Yet Another Resource Negotiator

One platform, many workloads: batch, interactive, real-time



Process

Ingest
Sqoop, Flume, Kafka

Transform
MapReduce, Hive, Pig, Spark

Discover

Analytic Database
Impala

Search
Solr

Model

Machine Learning
SAS, R, Spark, Mahout

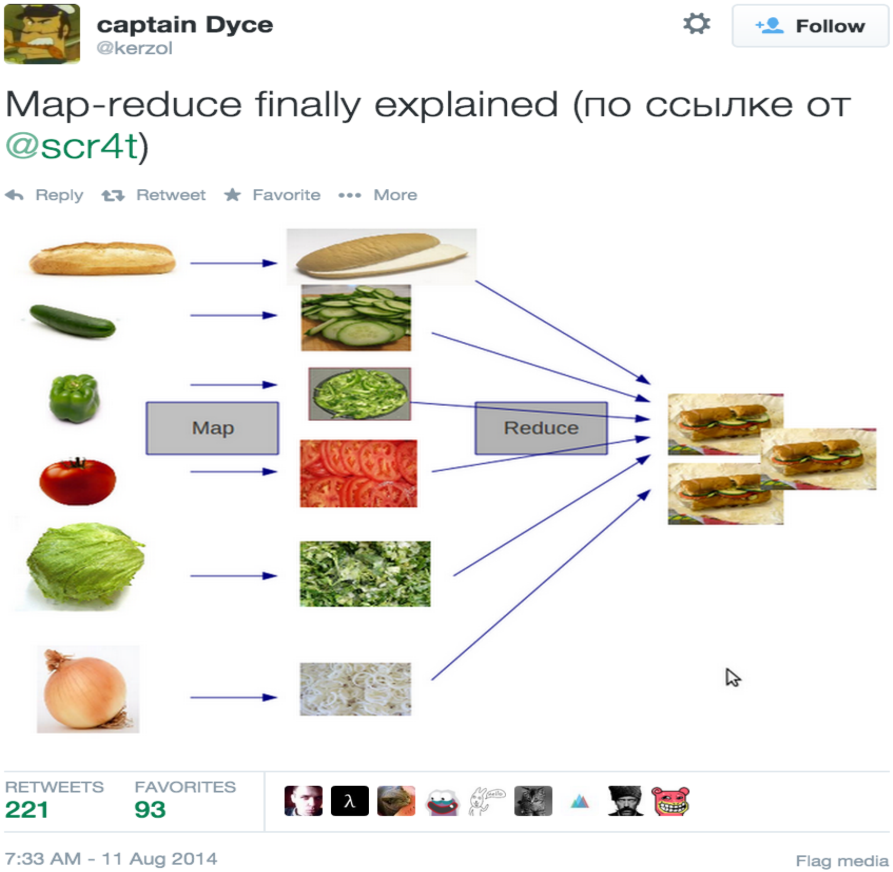
Serve

NoSQL Database
HBase

Streaming
Spark Streaming

Security and Administration YARN, Cloudera Manager, Cloudera Navigator

Unlimited Storage HDFS, HBase



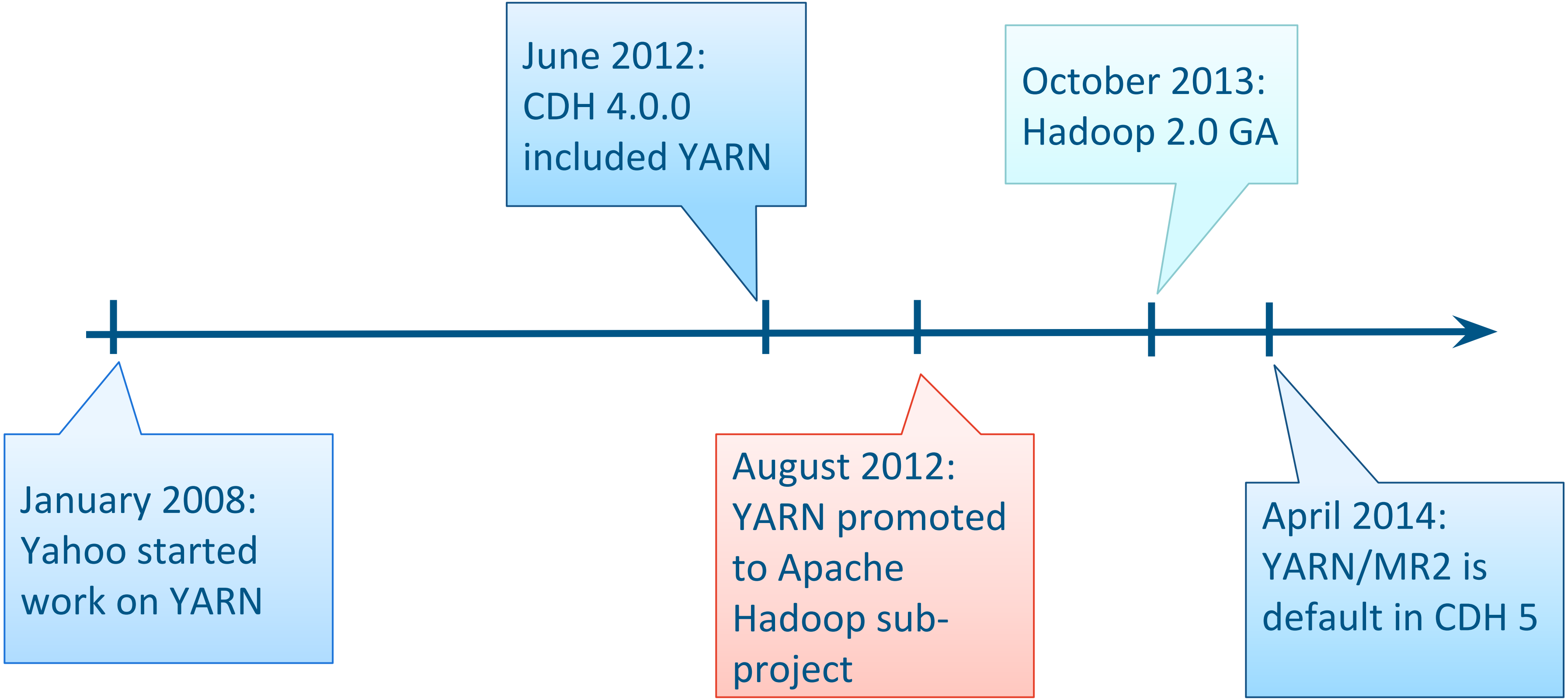
captain Dyce
@kerzol

Map-reduce finally explained (по ссылке от @scr4t)

RETWEETS 221
FAVORITES 93

7:33 AM - 11 Aug 2014

An Apache YARN timeline



January 2008: Yahoo started work on YARN

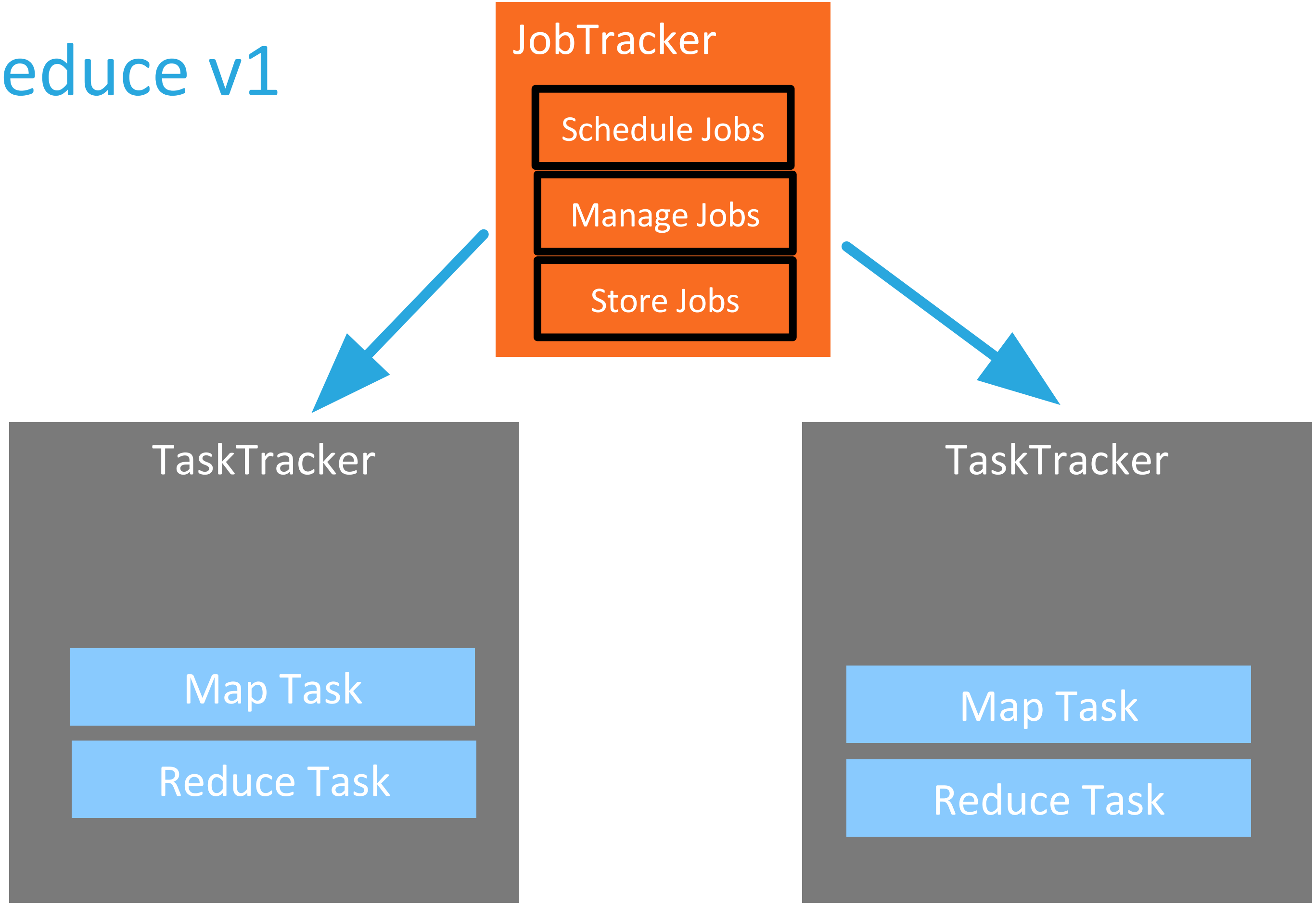
June 2012: CDH 4.0.0 included YARN

August 2012: YARN promoted to Apache Hadoop sub-project

October 2013: Hadoop 2.0 GA

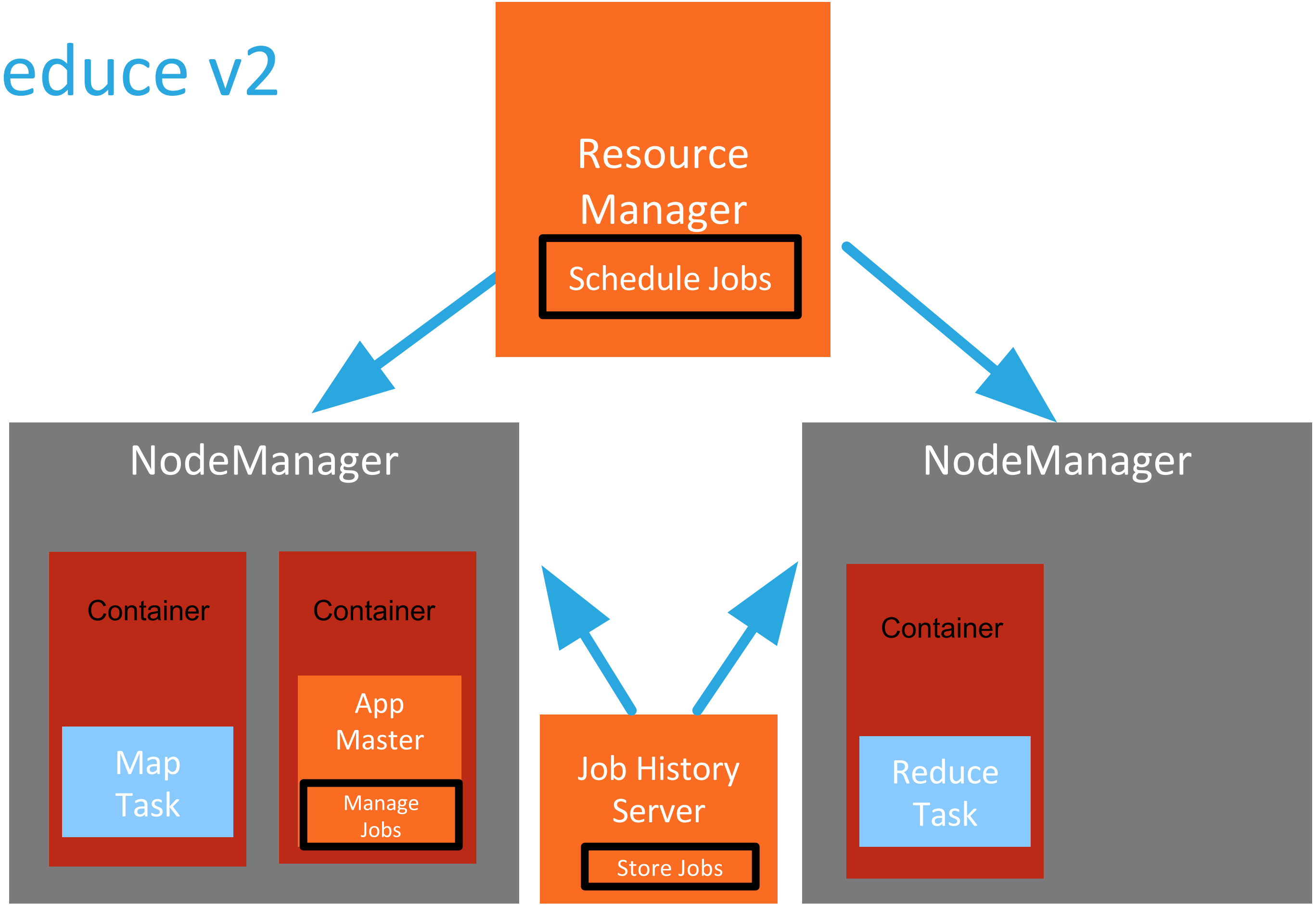
April 2014: YARN/MR2 is default in CDH 5

MapReduce v1



cloudera

MapReduce v2



cloudera

YARN motivation

	MR1	MR2
Scalability	JobTracker tracks all jobs, tasks Max out at 4k nodes , 40k tasks	Split up tracking between ResourceManager, ApplicationMaster Scale up to 10k nodes, 100k tasks
Availability	JT HA	RM HA & for per-application basis
Utilization	Fixed size slots for map, reduce	Allocate only as many resources as needed, allows cluster utilization > 70%
Multi-tenancy	N/A	Cluster resource management system Data locality & lowered operational costs from sharing resources between frameworks

cloudera

Upgrading MR1 to MR2

MR1 to MR2 functionality mapping

- Completely revamped architecture in MR2 on YARN
- While most translate, some configurations don’t

MR2 on YARN	Applications on YARN
Memory > heap to account for overhead: Memory per Container: mapreduce.[map|reduce].memory.mb (1.5 GB) Map/Reduce Task Maximum Heap Size: mapreduce.[map|reduce].java.opts.max.heap (1GB) CPU per Container: mapreduce.[map|reduce].cpu.vcores	Mem/CPU thresholds: Container Memory Minimum: yarn.scheduler.minimum-allocation-mb Container Memory Maximum: yarn.scheduler.maximum-allocation-mb Container Virtual CPU Cores Minimum: yarn.scheduler.minimum-allocation-vcores Container Virtual CPU Cores Maximum: yarn.scheduler.maximum-allocation-vcores

YARN compatibility

Migration path	Binary support
MR1 (CDH4) to MR2 (CDH5)	✔
MR1 (CDH4) to MR1 (CDH5)	✔
MR2 (CDH4) to MR1/MR2 (CDH5)	✖

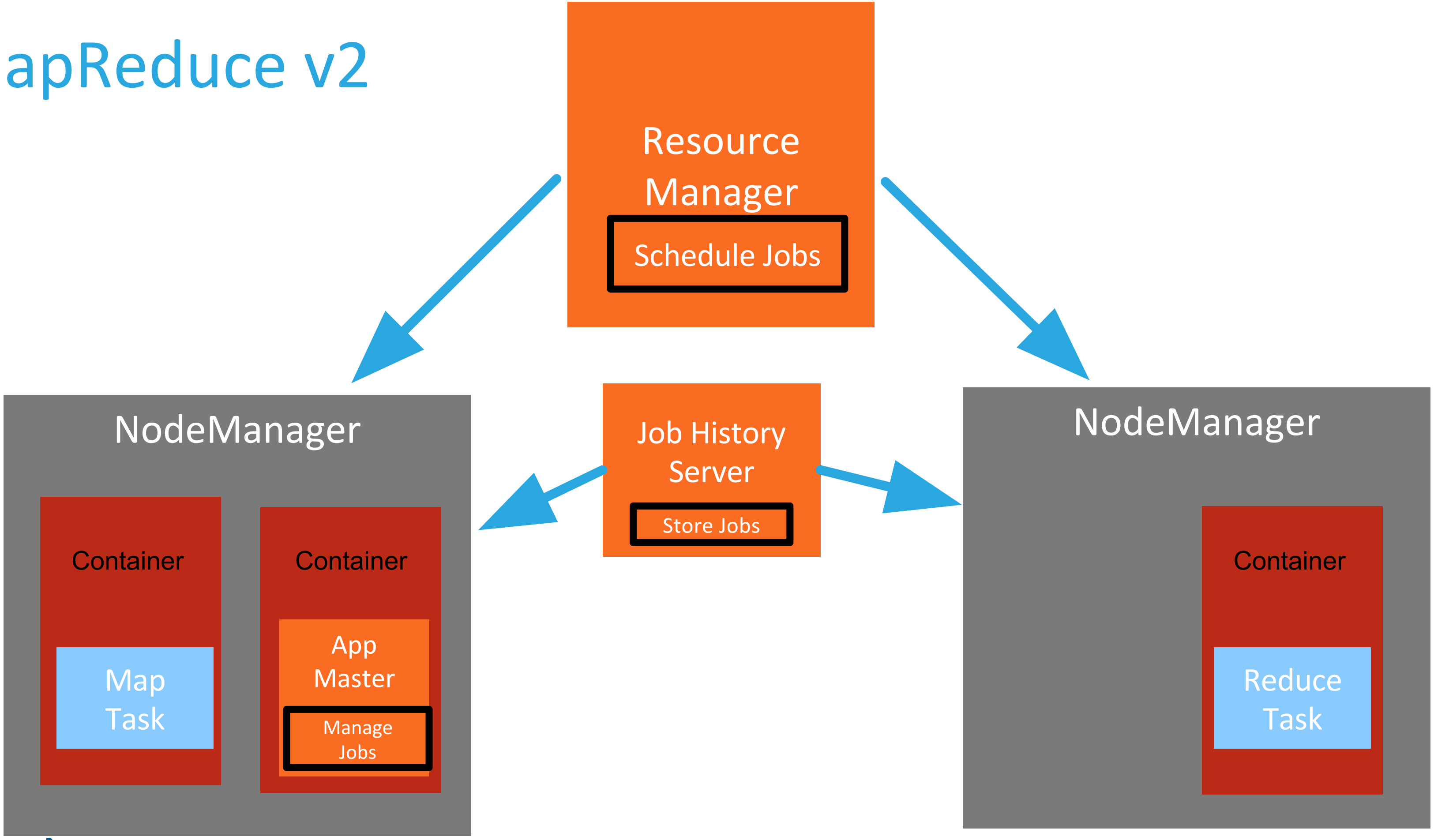
CDH has complete binary/source compatibility for almost all programs.

Virtually every job compiled against MR1 in CDH 4 will be able to run without any modifications on an MR2 cluster.

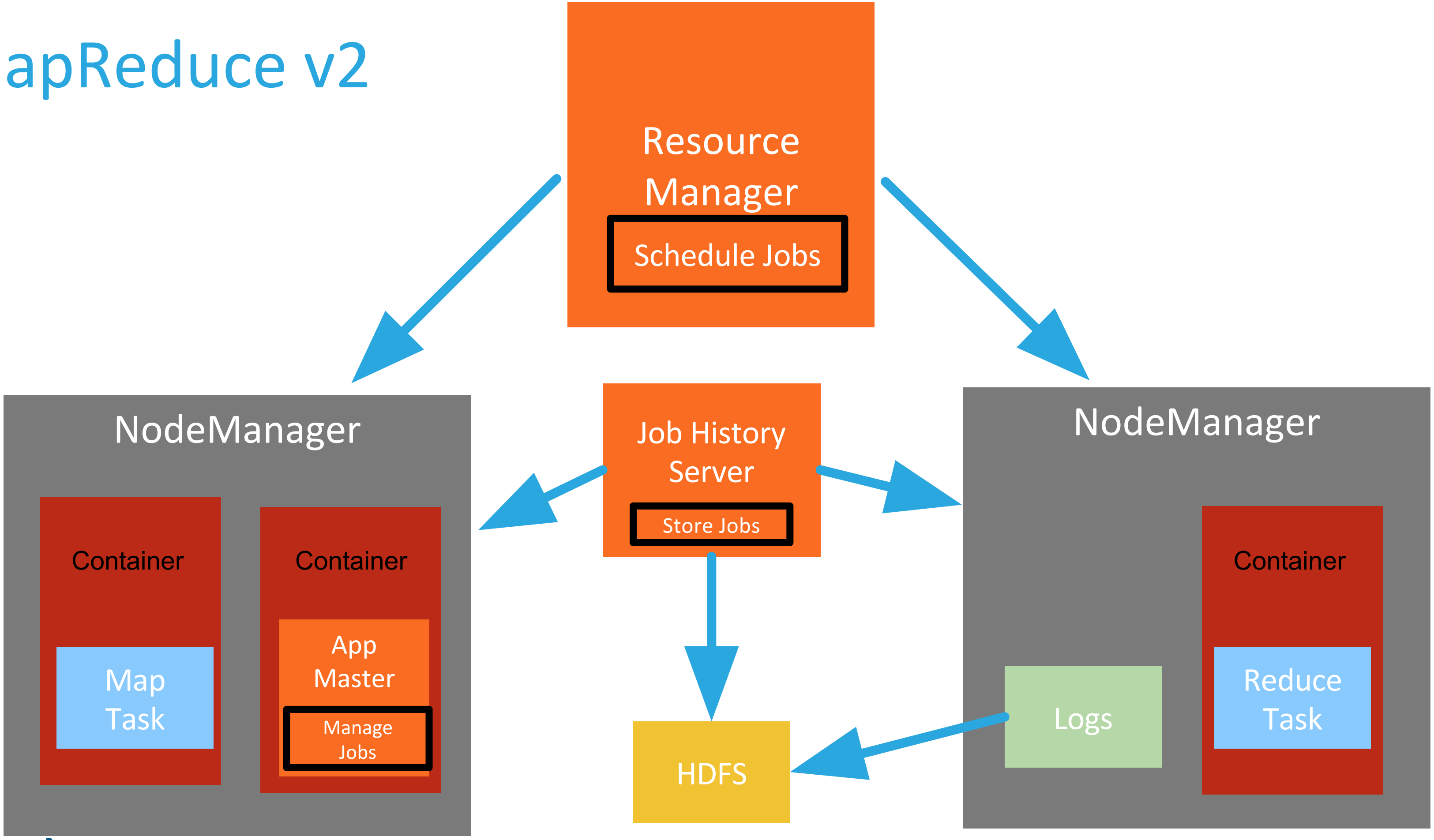
- Migrating to MR2 on YARN
  - For Operators: http://blog.cloudera.com/blog/2013/11/migrating-to-mapreduce-2-on-yarn-for-operators/
  - For Users: http://blog.cloudera.com/blog/2013/11/migrating-to-mapreduce-2-on-yarn-for-users/
  - http://blog.cloudera.com/blog/2014/04/apache-hadoop-yarn-avoiding-6-time-consuming-gotchas/
- Getting MR2 Up to Speed
  - http://blog.cloudera.com/blog/2014/02/getting-mapreduce-2-up-to-speed/
- Avoiding YARN Gotchas
  - http://blog.cloudera.com/blog/2014/04/apache-hadoop-yarn-avoiding-6-time-consuming-

YARN upgrade pitfalls

MapReduce v2



MapReduce v2



cloudera

General log related configuration properties

Log configuration parameter	What it does
yarn.nodemanager.log-dirs	Determines where the container-logs are stored on the node when the containers are running. Default is ${yarn.log.dir}/userlogs. For MapReduce applications, each container directory will contain the files stderr, stdin, and syslog generated by that container.
yarn.log-aggregation-enable	Whether to enable log aggregation or not. If disabled, NMs will keep the logs locally and not aggregate them.

YARN applications

Llama, Slider, Spark

YARN applications

- Llama (Low Latency Application MAster)
  - Reserves memory in YARN for short-lived processes (e.g. Impala)
  - Registers one long-lived AM per YARN pool
  - Caches resources allocated by YARN for a short time, so that they can be quickly re-allocated to Impala queries
  - Long-term solution is to run Impala on YARN but currently recommend setting up admission control

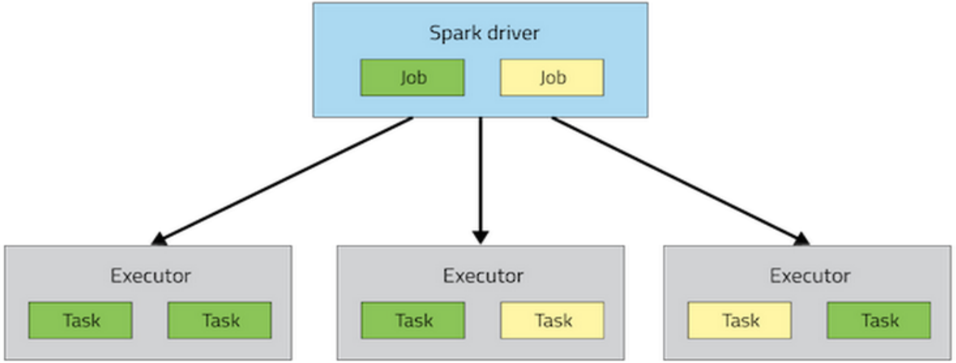
YARN applications

- Apache Slider (incubating) née Hoya
  - Runs long-lived persistent services on YARN (e.g. HBase)
  - Not currently recommended as it doesn't provide IO isolation

Spark on YARN

Spark Overview

- Application corresponds to an instance of the SparkContext class
- Executors are long lived processes
- Applications take up resource until the app completes



Why Spark on Yarn?

- Built in scheduler for resource management (Isolation, Prioritization)
- Sharing resources within a cluster (MapReduce, Spark)
- YARN is the only cluster manager for Spark that supports security (Kerberized Hadoop).

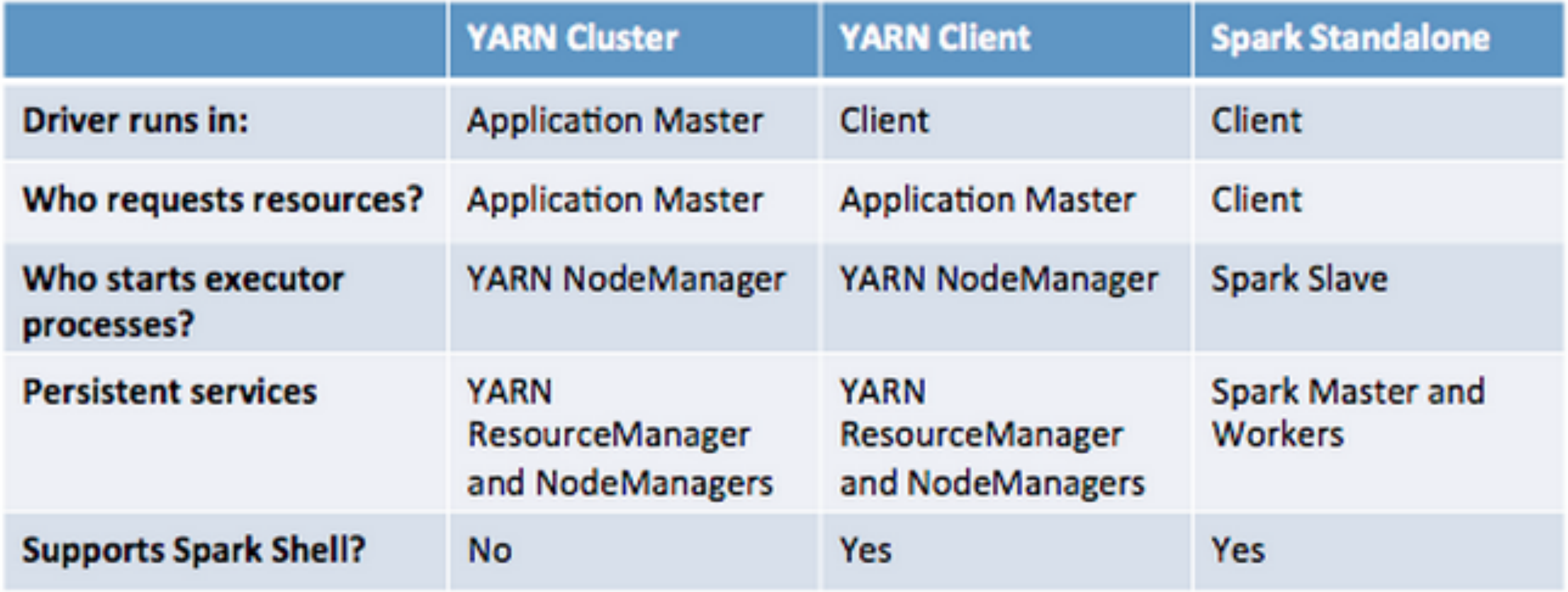
Configuring YARN for Spark

- Designed for interactive queries and iterative algorithms
  - In-memory caching, DAG engine, and APIs
- Set yarn.scheduler.maximum-allocation-mb as high as 64G on a machine with 192GB of memory
- Won’t run with small (< 1 GB) containers due to overhead

Reference:

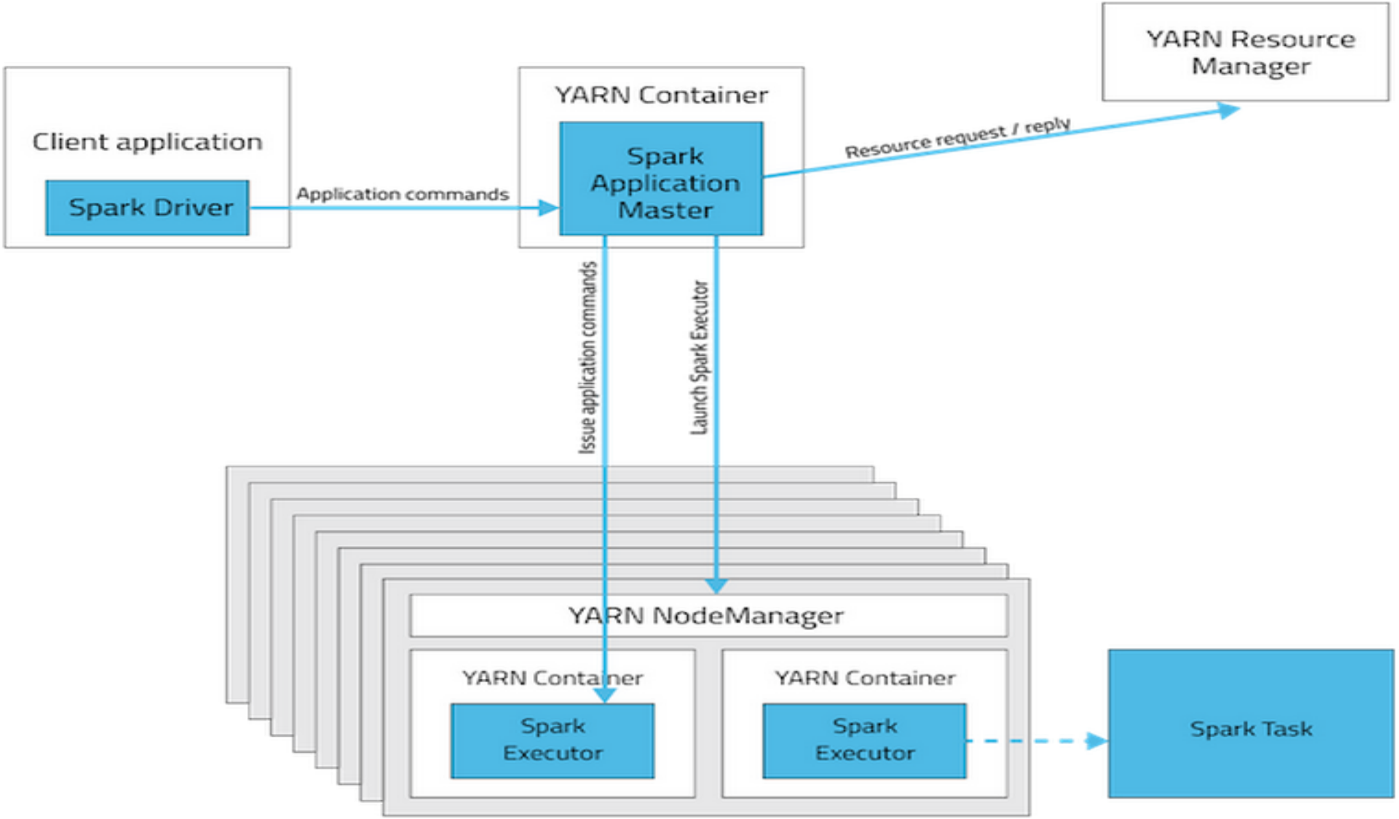
http://blog.cloudera.com/blog/2014/05/apache-spark-resource-management-and-yarn-app-models/

Deploying Spark Jobs



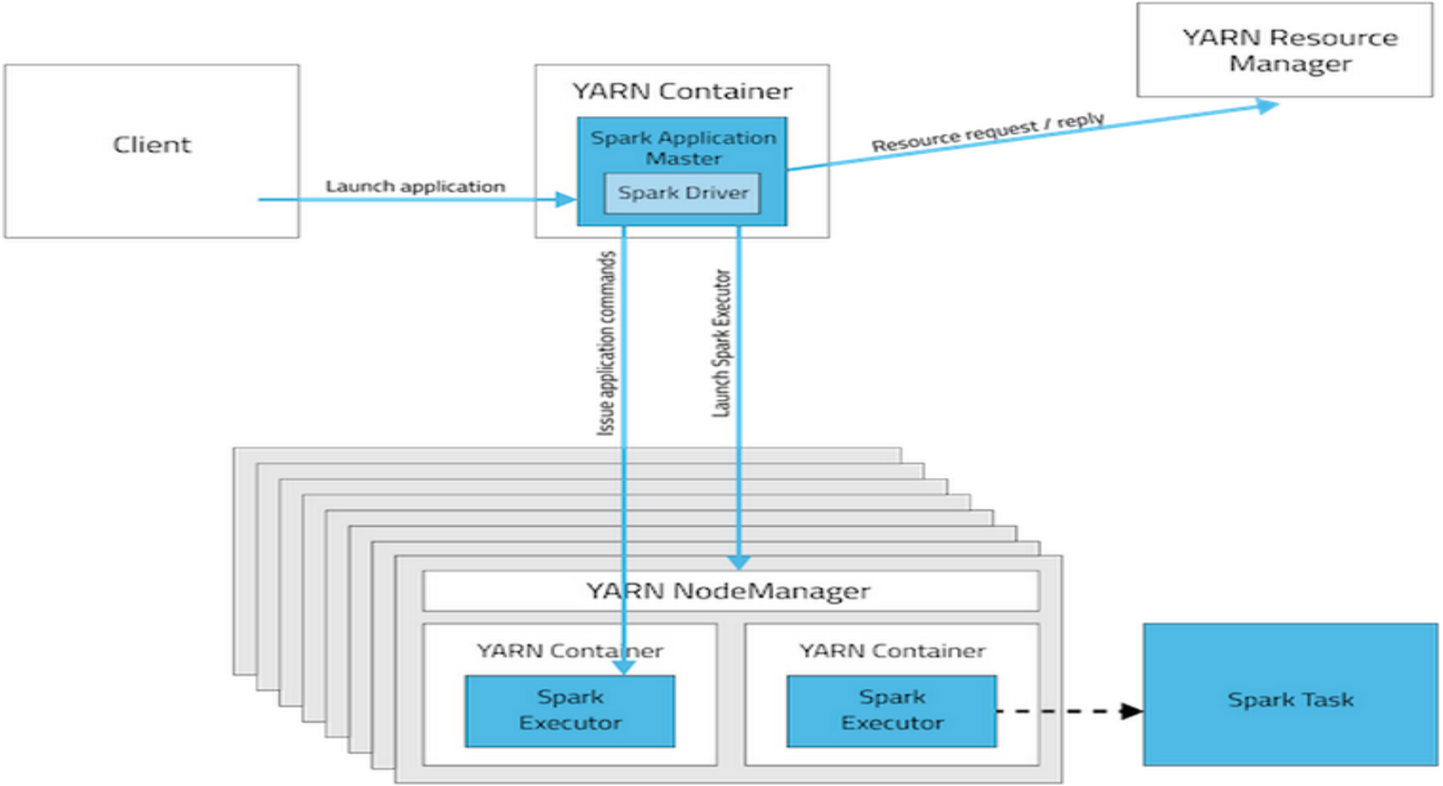
	YARN Cluster	YARN Client	Spark Standalone
Driver runs in:	Application Master	Client	Client
Who requests resources?	Application Master	Application Master	Client
Who starts executor processes?	YARN NodeManager	YARN NodeManager	Spark Slave
Persistent services	YARN ResourceManager and NodeManagers	YARN ResourceManager and NodeManagers	Spark Master and Workers
Supports Spark Shell?	No	Yes	Yes

Spark – YARN-CLIENT



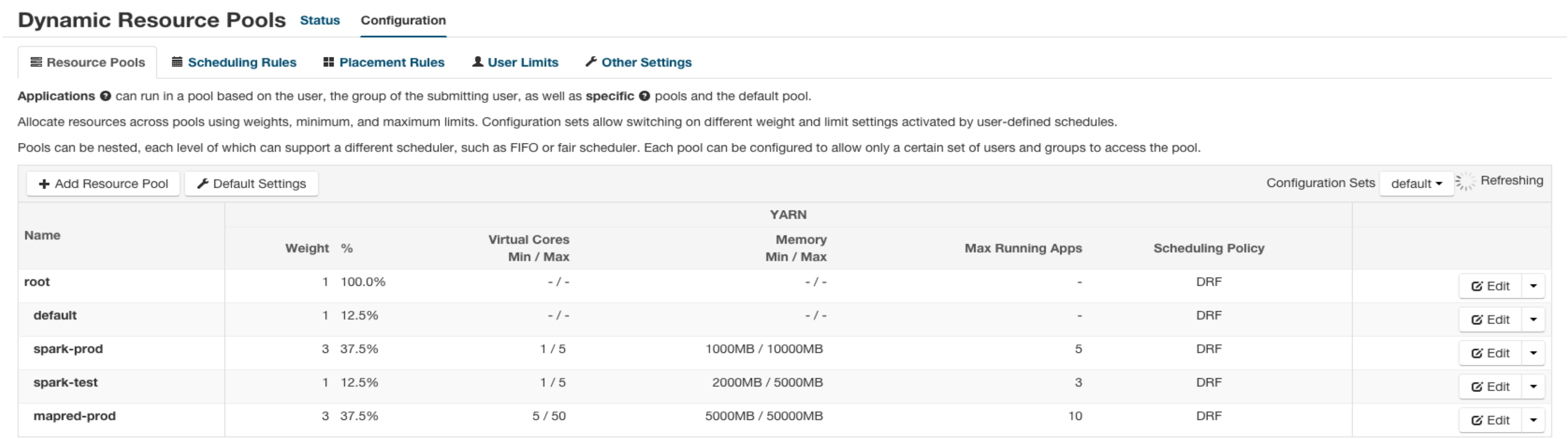
cloudera

Spark – YARN-CLUSTER



Spark Scheduling

- Fair scheduler for resource sharing
  - spark.yarn.queue
- Standalone cluster mode currently only supports a simple FIFO scheduler across applications



Applications can run in a pool based on the user, the group of the submitting user, as well as **specific** pools and the default pool.

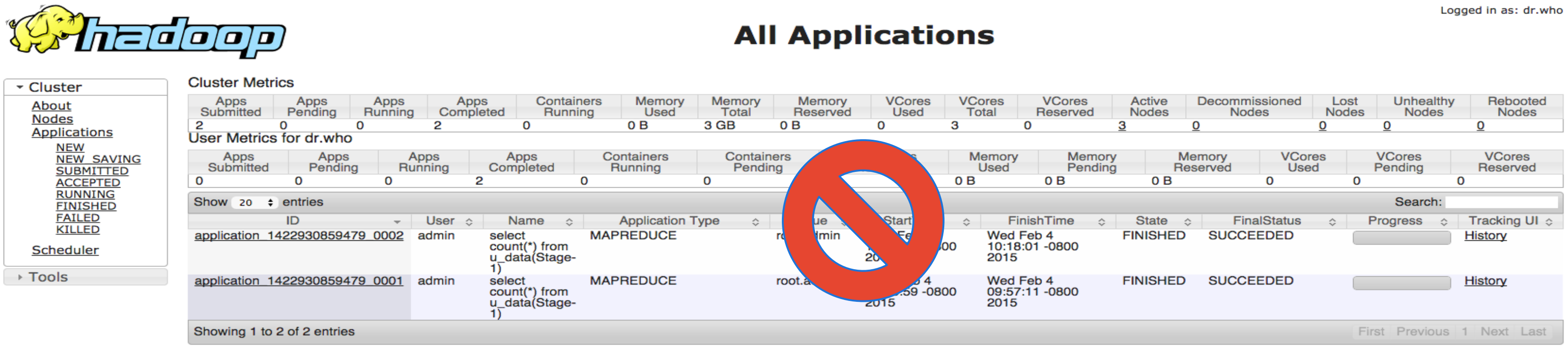
Allocate resources across pools using weights, minimum, and maximum limits. Configuration sets allow switching on different weight and limit settings activated by user-defined schedules.

Pools can be nested, each level of which can support a different scheduler, such as FIFO or fair scheduler. Each pool can be configured to allow only a certain set of users and groups to access the pool.

	YARN					
Name	Weight	%	Virtual Cores Min / Max	Memory Min / Max	Max Running Apps	Scheduling Policy
root	1	100.0%	- / -	- / -	-	DRF
default	1	12.5%	- / -	- / -	-	DRF
spark-prod	3	37.5%	1 / 5	1000MB / 10000MB	5	DRF
spark-test	1	12.5%	1 / 5	2000MB / 5000MB	3	DRF
mapred-prod	3	37.5%	5 / 50	5000MB / 50000MB	10	DRF

Spark Not Running On Yarn?

- **Symptom**:
  - Use spark-submit to run a python job, but only see the resources being used on one machine.



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
2	0	0	2	0	0 B	3 GB	0 B	0	3	0	3	0	0	0	0

User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	[illegible]	Memory Used	Memory Pending	Memory Reserved	VCores Used	VCores Pending	VCores Reserved
0	0	0	2	0	0	[illegible]	0 B	0 B	0 B	0	0	0

ID	User	Name	Application Type	[illegible]ue	Start[illegible]	FinishTime	State	FinalStatus	Progress	Tracking UI
application_1422930859479_0002	admin	select count(*) from u_data(Stage-1)	MAPREDUCE	r[illegible]dmin	[illegible] Fe[illegible] [illegible]00 20[illegible]	Wed Feb 4 10:18:01 -0800 2015	FINISHED	SUCCEEDED		History
application_1422930859479_0001	admin	select count(*) from u_data(Stage-1)	MAPREDUCE	root.a[illegible]	[illegible] 4 [illegible]:59 -0800 2015	Wed Feb 4 09:57:11 -0800 2015	FINISHED	SUCCEEDED		History

Showing 1 to 2 of 2 entries

Spark Not Running On Yarn?

- **Workaround**:
  - Ensure that you have the options in the right position
    - **Cause**:

      -
        ```
        $ spark-submit pi.py –master yarn-client
        ```
    - **Fix**:

      -
        ```
        $ spark-submit --master yarn-client pi.py 1000
        ```
  - Usage:
    ```
    spark-submit [options] <app jar | python file> [app options]
    ```
  - Lot of improvements made to Spark 1.2 for spark-submit SPARK-1652

PySpark on Yarn Limitation

- **Symptom**:

```
$ spark-submit --master yarn-cluster pi.py 1000

Error: Cluster deploy mode is currently not
supported for python applications.
Run with --help for usage help or --verbose for
debug output
```

PySpark on Yarn Limitation

- **Workaround**:

```
$ spark-submit --master yarn-client pi.py 1000
…
Pi is roughly 3.132290
15/02/11 09:41:34 INFO SparkUI: Stopped Spark web UI
at http://sparktest-1.ent.cloudera.com:4040
15/02/11 09:41:34 INFO DAGScheduler: Stopping
DAGScheduler
```

- Future work: SPARK-5162 / SPARK-5173

Lost Spark Executors

- **Symptom**:
  - Spark Driver WARN Messages

```
14/12/08 17:11:08 WARN scheduler.TaskSetManager: Lost task 205.0 in
stage 2.0 (TID 352, test-1.cloudera.com: ExecutorLostFailure (executor
lost)
```

  - NodeManager Logs

```
2014-12-08 17:10:32,860 WARN org.apache.hadoop.yarn.server.nodemanager.
containermanager.monitor.ContainersMonitorImpl: Container
[pid=26842,containerID=container_1418059756626_0010_01_000093_01]
is running beyond physical memory limits.
Current usage: 26.2 GB of 26 GB physical memory used;
27.1 GB of 54.6 GB virtual memory used. Killing container.
```

Lost Spark Executors

- **Workaround**:
  - Increase `spark.yarn.[executor|driver].memoryOverhead`
    - Test for your specific use case. 1GB to 4GB

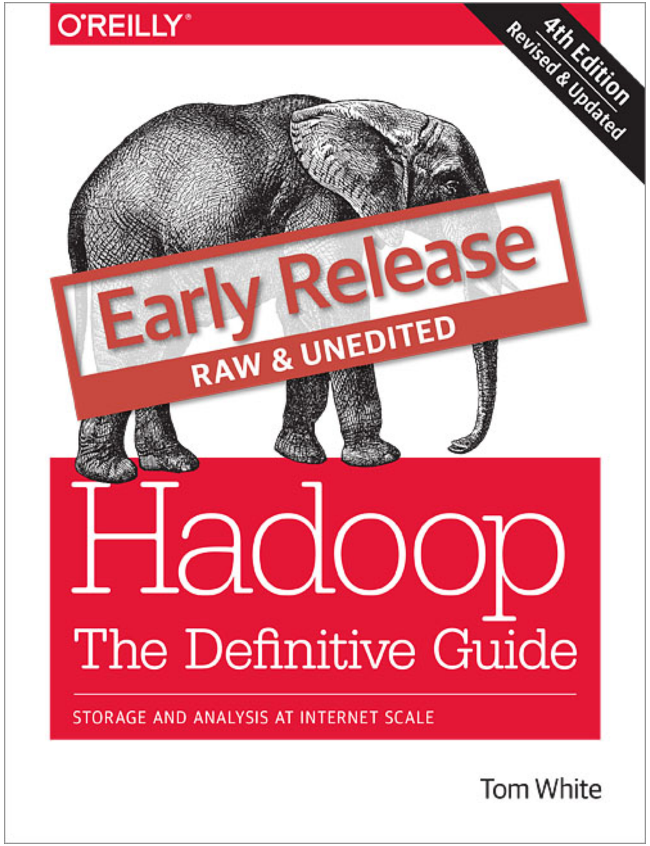
Spark Improvements

- Spark 1.2 / CDH 5.3 : Prefer RDDs that are cached locally in HDFS
- Spark 1.2 : Dynamically release unused resources via spark. dynamicAllocation.enabled
  - Only support via YARN currently.
- Spark Streaming save incoming data to a WAL (write-ahead log) on HDFS, preventing any data loss on driver failure.

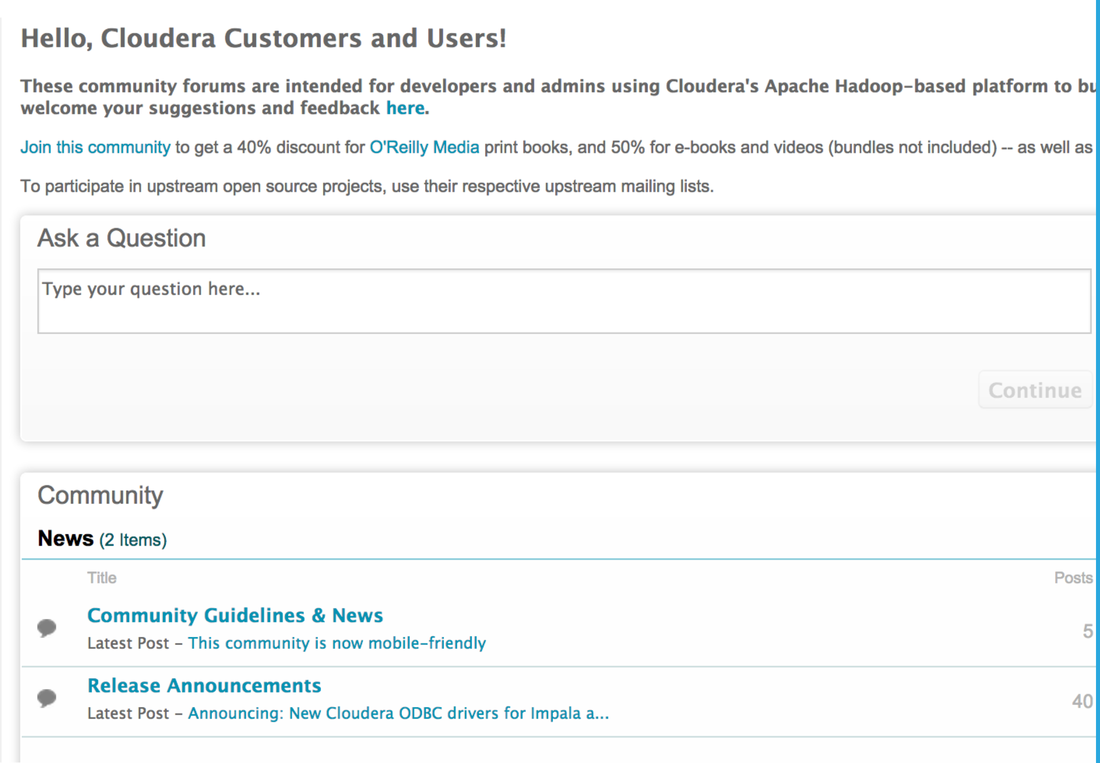
Conclusion

YARN performance

- Improved cluster utilization
  - Can run more jobs in smaller clusters
  - Run in uber mode for smaller jobs (reduces AM overhead)
- Dynamic resource sharing between frameworks
  - One framework can use the entire cluster
- Tom White's *Hadoop: The Definitive Guide 4th Ed* (book signing @6:30pm)
  - Chapter 4 is on YARN



Join the Discussion



Hello, Cloudera Customers and Users!

These community forums are intended for developers and admins using Cloudera's Apache Hadoop-based platform to bu welcome your suggestions and feedback here.

Join this community to get a 40% discount for O'Reilly Media print books, and 50% for e-books and videos (bundles not included) -- as well as

To participate in upstream open source projects, use their respective upstream mailing lists.

News (2 Items)

Title	Posts
Community Guidelines & News Latest Post – This community is now mobile-friendly	5
Release Announcements Latest Post – Announcing: New Cloudera ODBC drivers for Impala a...	40

Get community help or provide feedback

cloudera.com/community

cloudera

cloudera LIVE
The fastest way to get started with Hadoop

Visit us at Booth #809

HIGHLIGHTS:

Apache Kafka is now fully supported with Cloudera

Learn why Cloudera is the leader for security in Hadoop

cloudera

BOOK SIGNINGS

THEATER SESSIONS

TECHNICAL DEMOS

GIVEAWAYS

cloudera

Questions?

@kate_ting

@miklos_c

Spark Tuning Parameters

- spark.shuffle.consolidateFiles=true
- spark.yarn.executor.memoryOverhead
- spark.yarn.driver.memoryOverhead
- spark.shuffle.manager=SORT
- spark.rdd.compress=true
- spark.serializer=org.apache.spark.serializer.KryoSerializer

YARN vs Mesos: Resource Manager’s role

YARN	Mesos
Asks for resources	Offers resources
Evolved into a resource manager	Evolved into managing Hadoop
Written in Java	Written in C++
Locality aware	More customizable