Hadoop

Yizheng (Ethan) Chen

Advisor: Prof. Aditya Akella

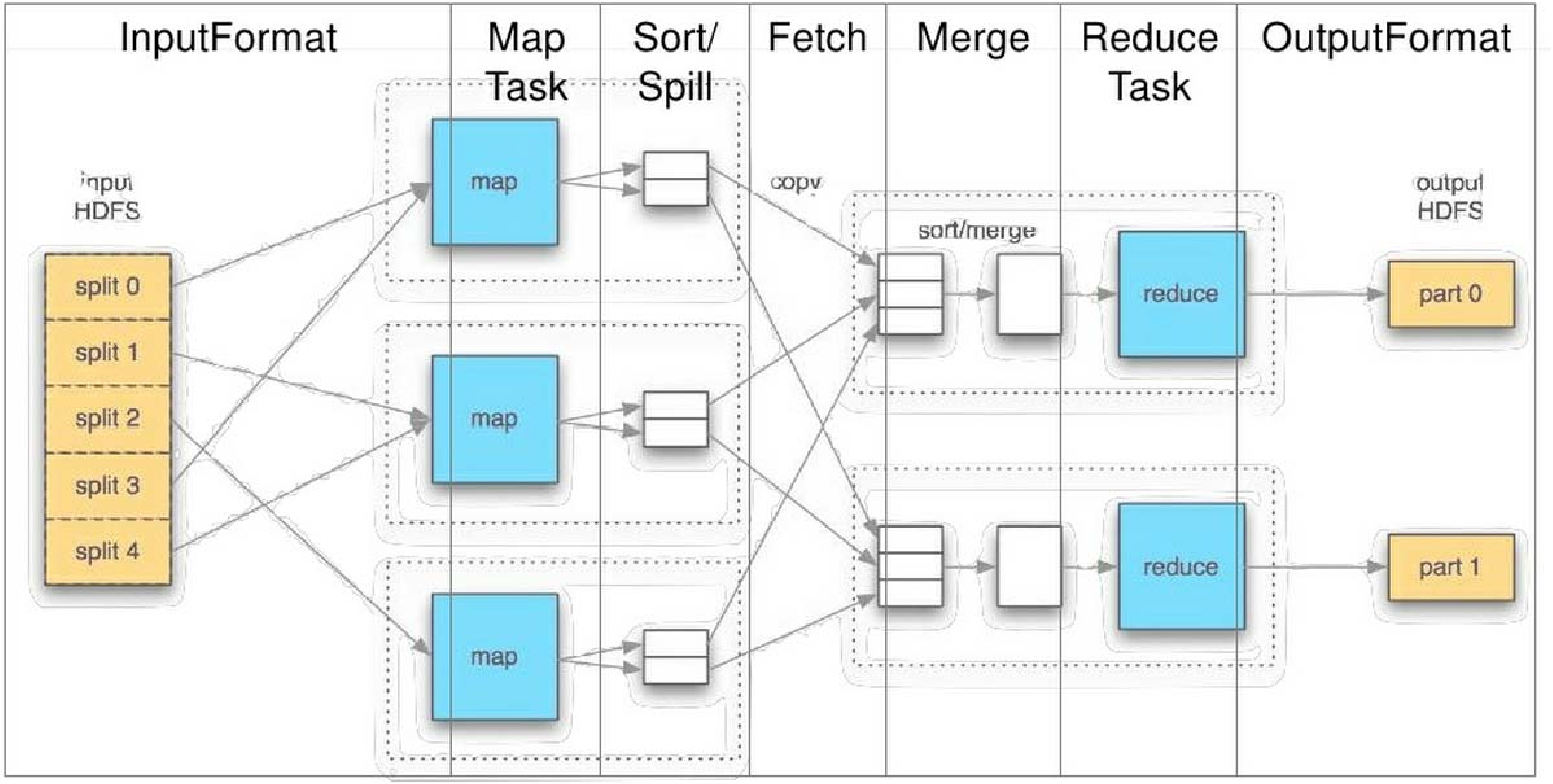
Outline

- Hadoop
- Yarn (NextGen Hadoop)

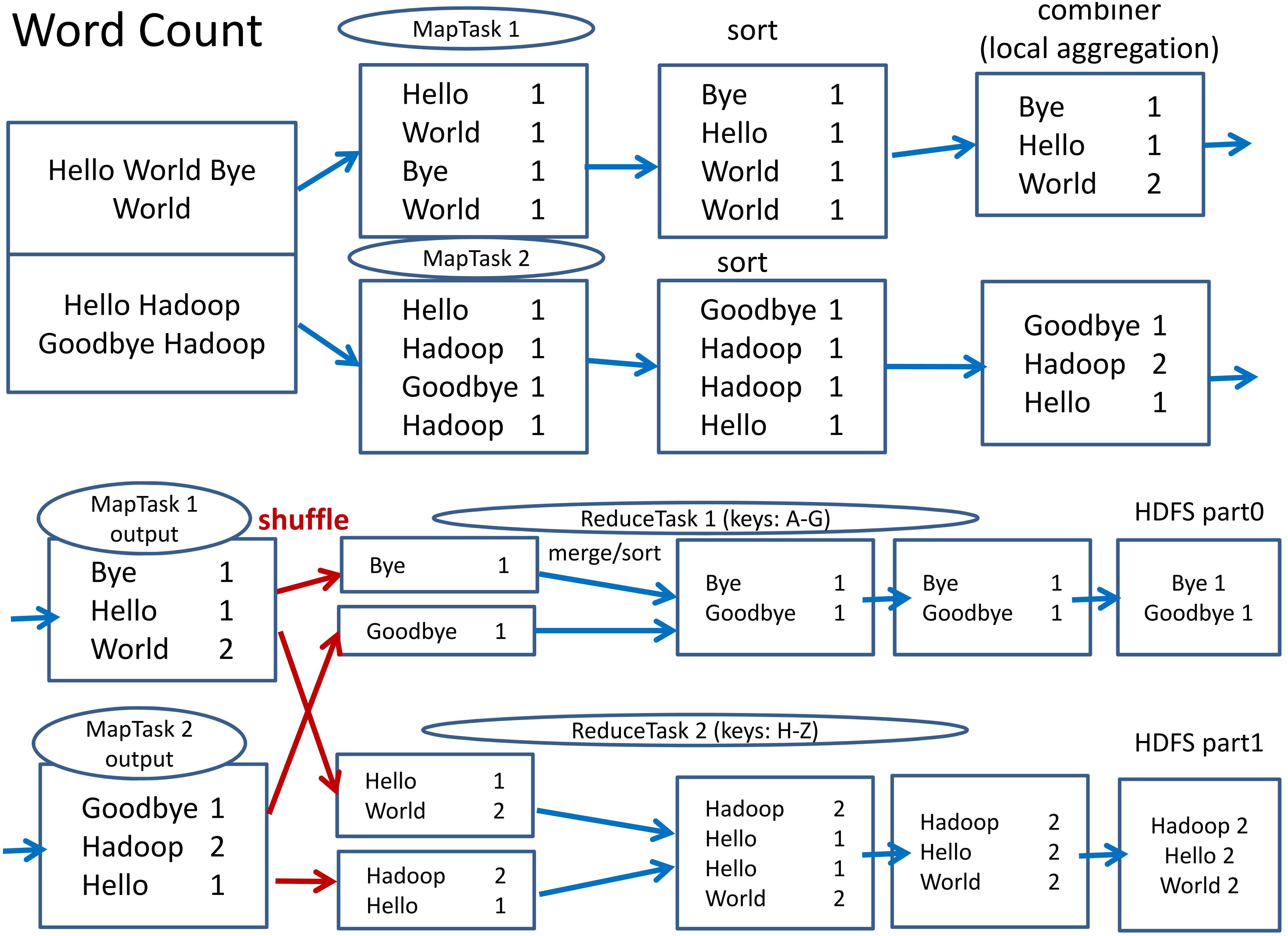
Hadoop

- What is Apache Hadoop
  - A framework (open-source software) for reliable, scalable, distributed computing
    - Hadoop MapReduce
      - A system for parallel processing of large data sets
    - Hadoop Distributed File System (HDFS™)
      - A distributed file system that provides high-throughput access to application data
      - Similar to GFS
  - http://hadoop.apache.org/
- Why Hadoop?

Hadoop Job Execution



Word Count



Hello World Bye World

Hello Hadoop Goodbye Hadoop

MapTask 1

Hello 1
World 1
Bye 1
World 1

sort

Bye 1
Hello 1
World 1
World 1

combiner (local aggregation)

Bye 1
Hello 1
World 2

MapTask 2

Hello 1
Hadoop 1
Goodbye 1
Hadoop 1

sort

Goodbye 1
Hadoop 1
Hadoop 1
Hello 1

Goodbye 1
Hadoop 2
Hello 1

MapTask 1 output

Bye 1
Hello 1
World 2

shuffle

ReduceTask 1 (keys: A-G)

Bye 1

Goodbye 1

merge/sort

Bye 1
Goodbye 1

Bye 1
Goodbye 1

HDFS part0

Bye 1
Goodbye 1

MapTask 2 output

Goodbye 1
Hadoop 2
Hello 1

ReduceTask 2 (keys: H-Z)

Hello 1
World 2

Hadoop 2
Hello 1

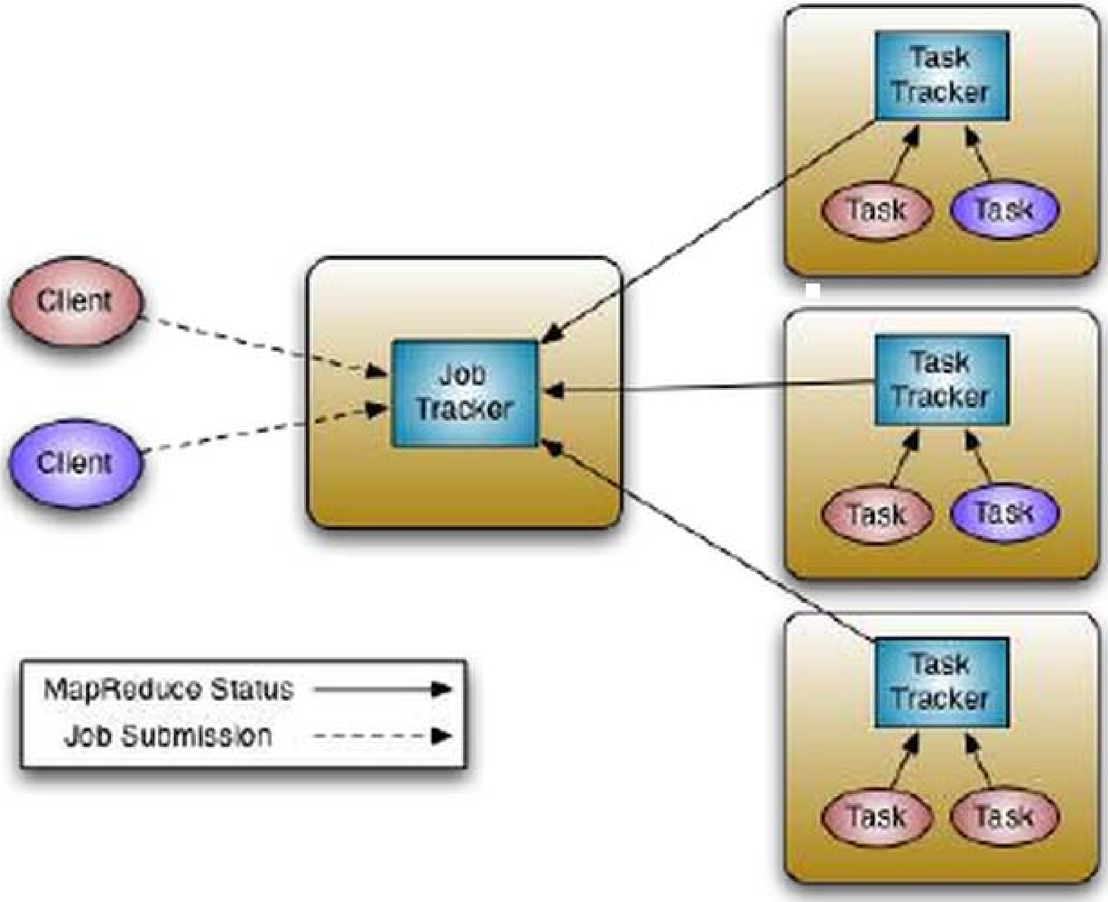
Hadoop 2
Hello 1
Hello 1
World 2

Hadoop 2
Hello 2
World 2

HDFS part1

Hadoop 2
Hello 2
World 2

Hadoop MapReduce

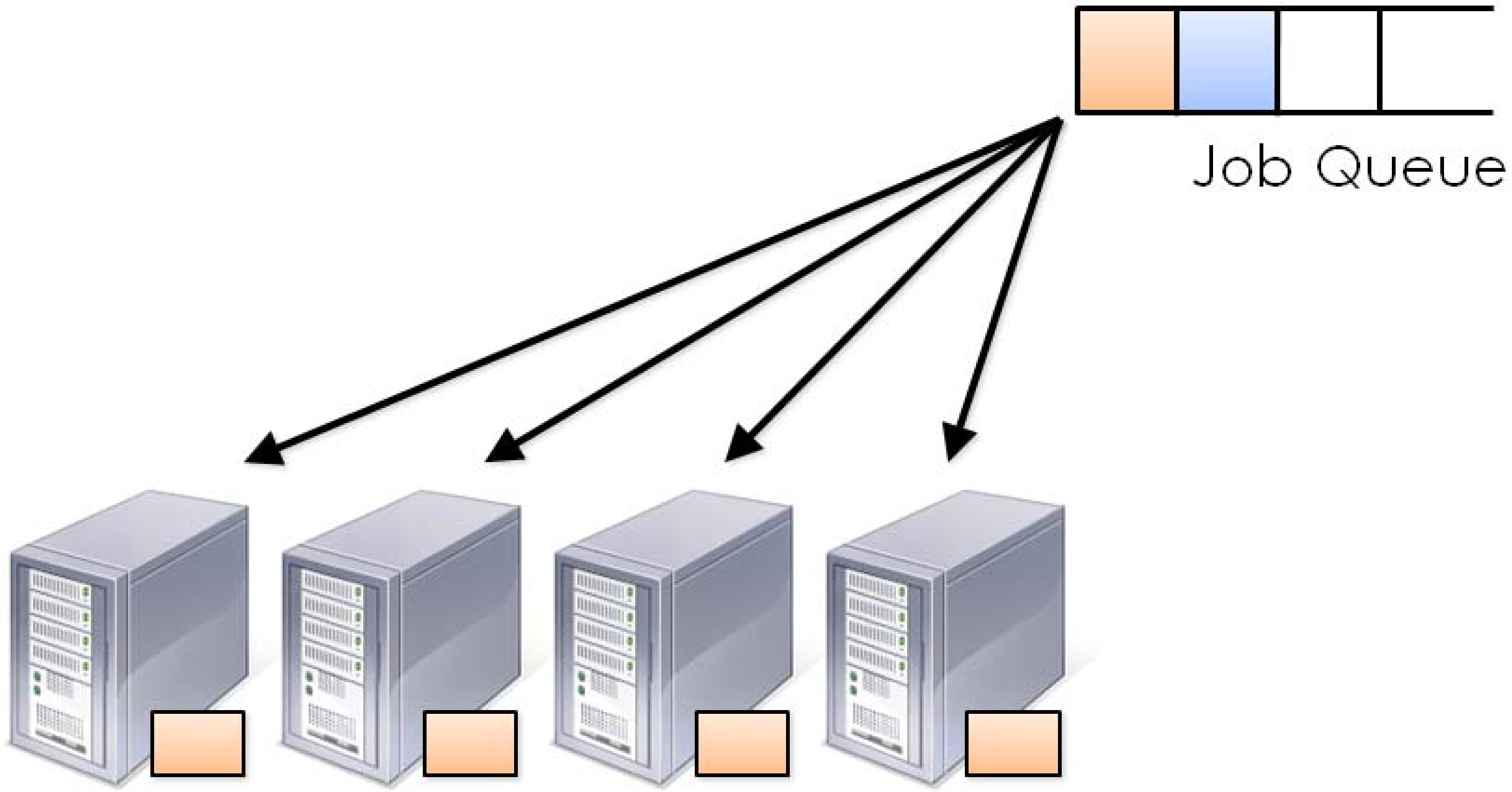


Hadoop Schedulers

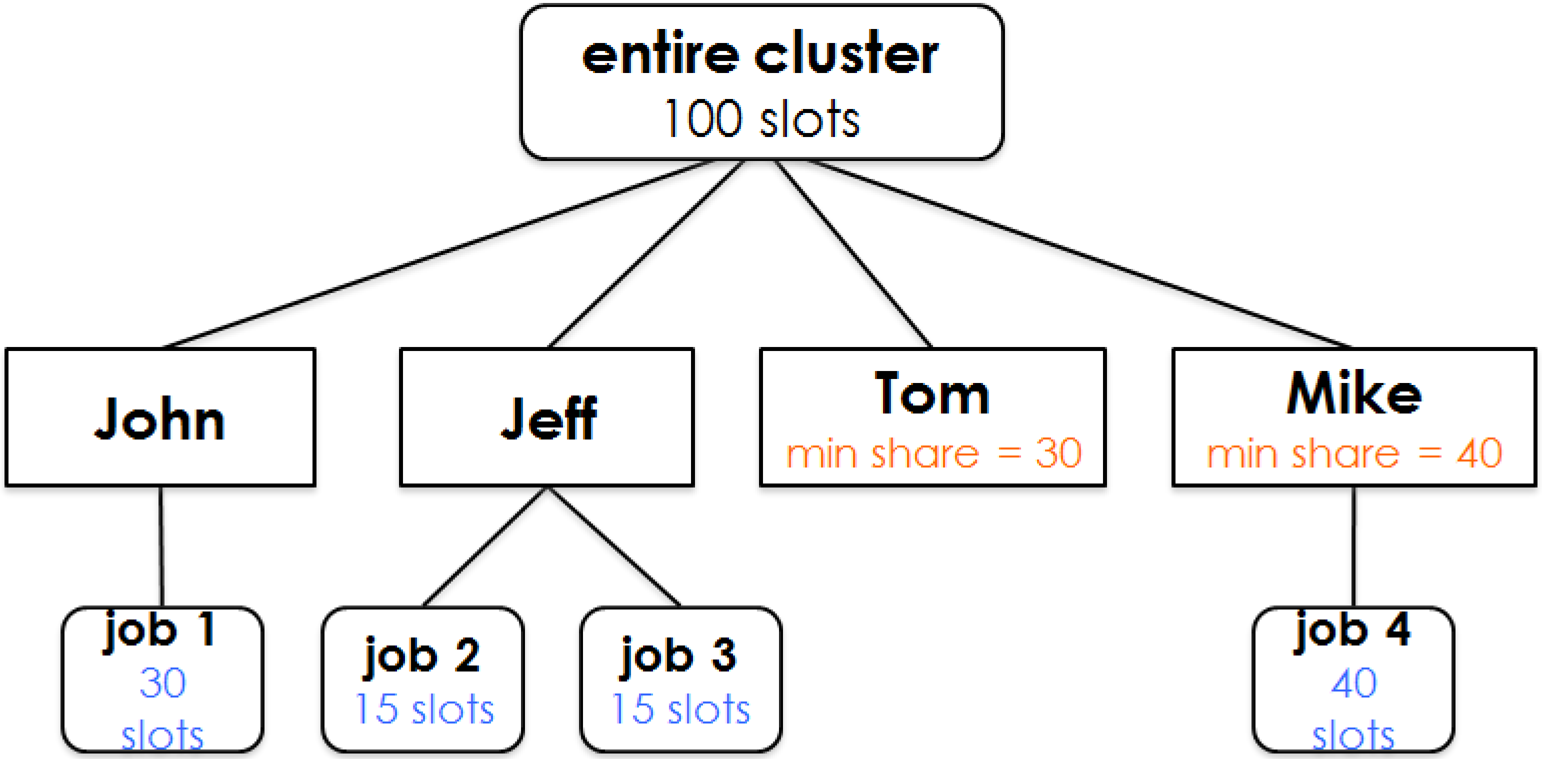
- A pluggable framework for job scheduling algorithm available since Hadoop 0.19
  - FIFO
  - Fair Scheduler (Facebook)
  - Capacity Scheduler (Yahoo!)

FIFO scheduler

Originally optimized for large batch jobs(web index construction)
FIFO order + priority queues

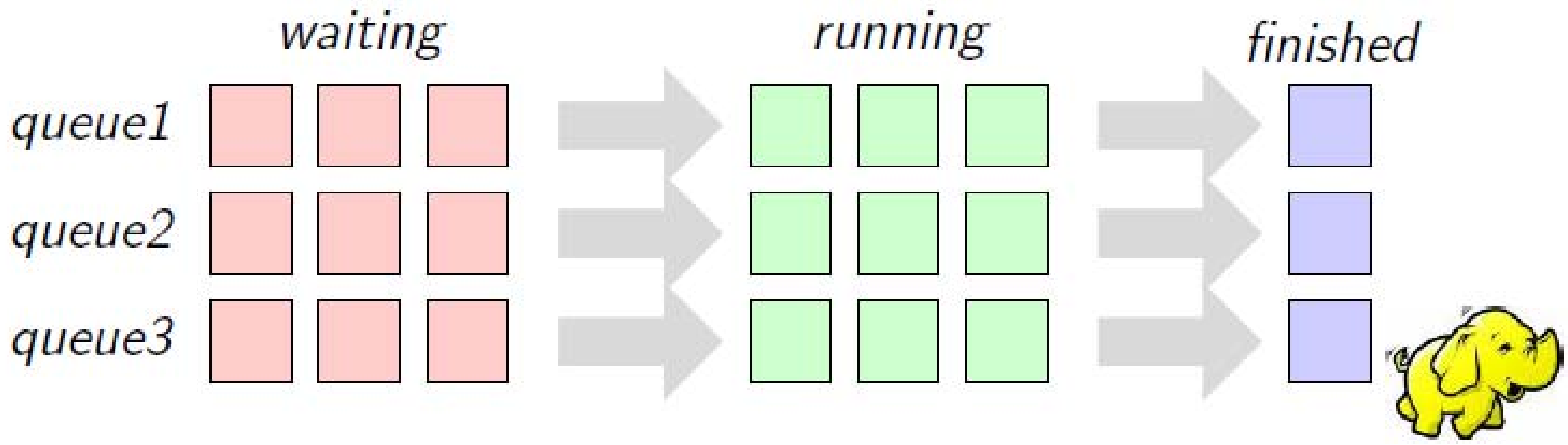


Fair Scheduler

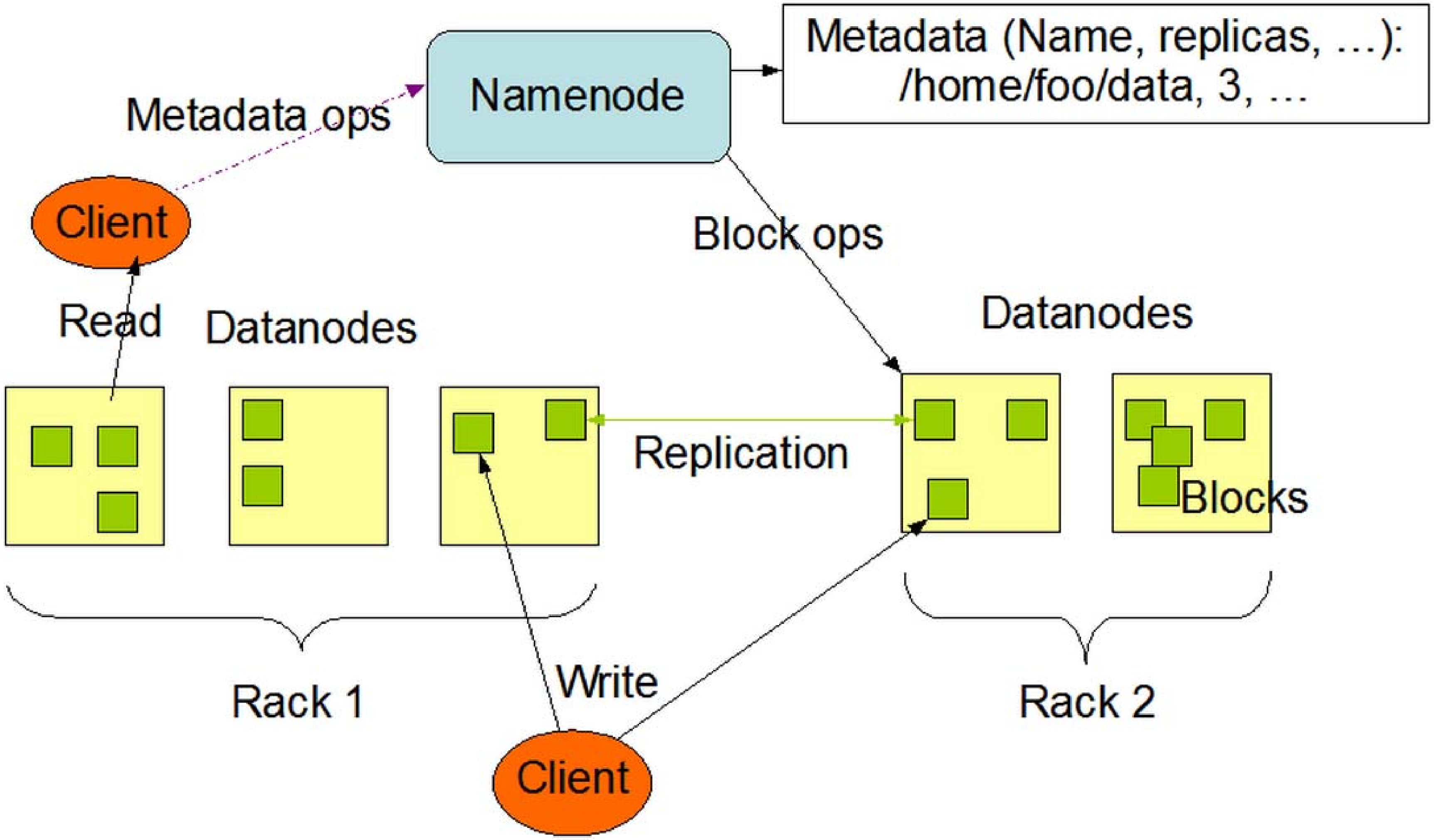


Capacity Scheduler

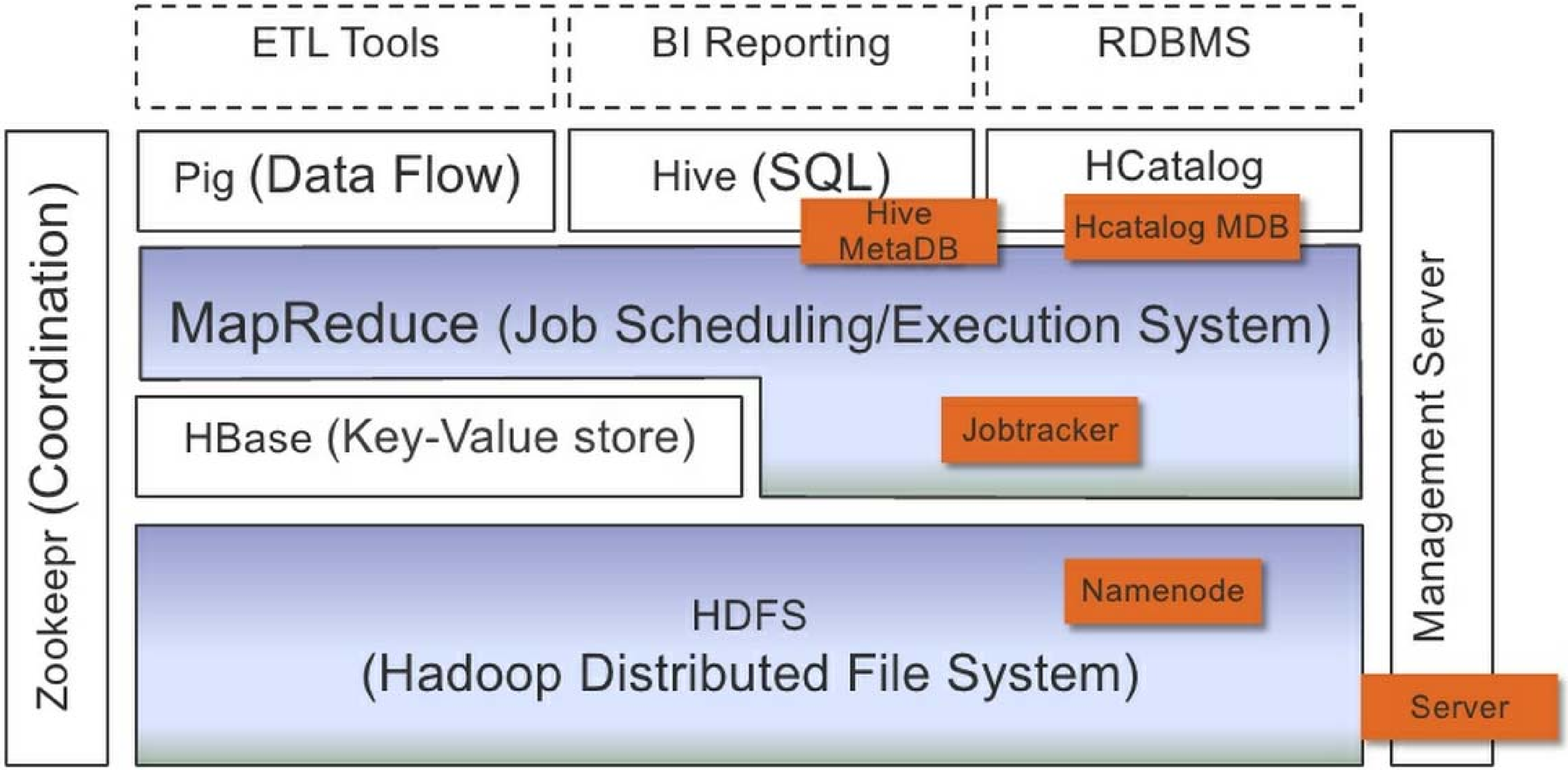
- Organizes jobs into queues
- Queue shares as %'s of cluster
  Optionally, can limit maximum resources per queue
- FIFO scheduling within each queue
- Contributed by Yahoo!



HDFS Architecture



Hadoop Ecosystem



HBase: BigTable-like

Hive: Data summarization and ad hoc querying

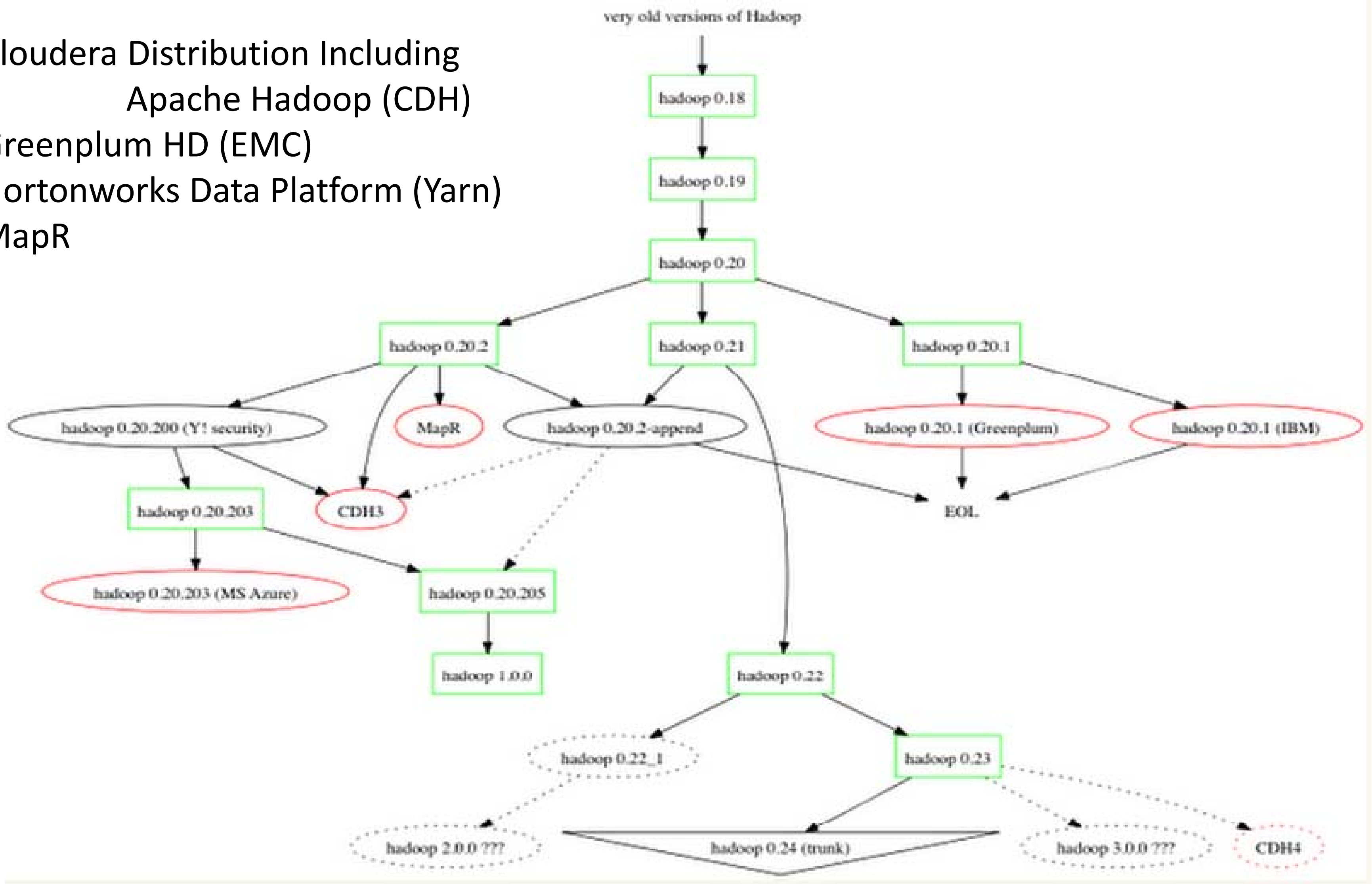
Pig: A high-level data-flow language and execution framework for parallel computation

HCatalog: Table and storage management service (table abstraction of data)

Zookeeper: A high-performance coordination service for distributed applications

Hadoop and Hadoop-derived Distributtions

*Cloudera Distribution Including
Apache Hadoop (CDH)
*Greenplum HD (EMC)
*Hortonworks Data Platform (Yarn)
*MapR



https://blogs.apache.org/bigtop/entry/all_you_wanted_to_know

Outline

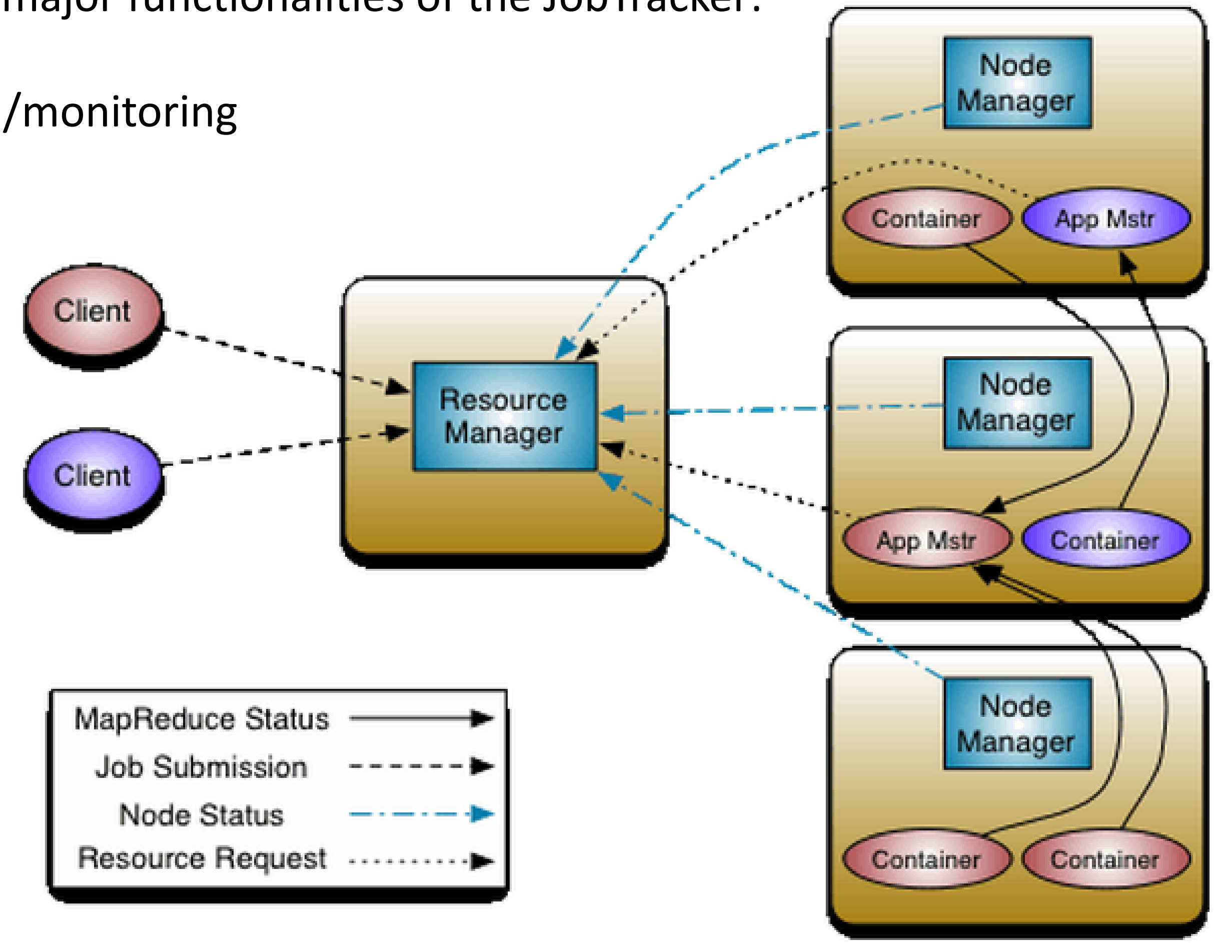
- Hadoop
- Yarn (NextGen Hadoop)

Yarn (NextGen Hadoop)

Split up the two major functionalities of the JobTracker:

* Management

* Job scheduling/monitoring



ResourceManager:

*Scheduler: allocate resources to the various running applications (pluggable policy plug-in)

*ApplicationsManager : accept job-submissions/launch the first container for ApplicationMaster

Resource Allocation

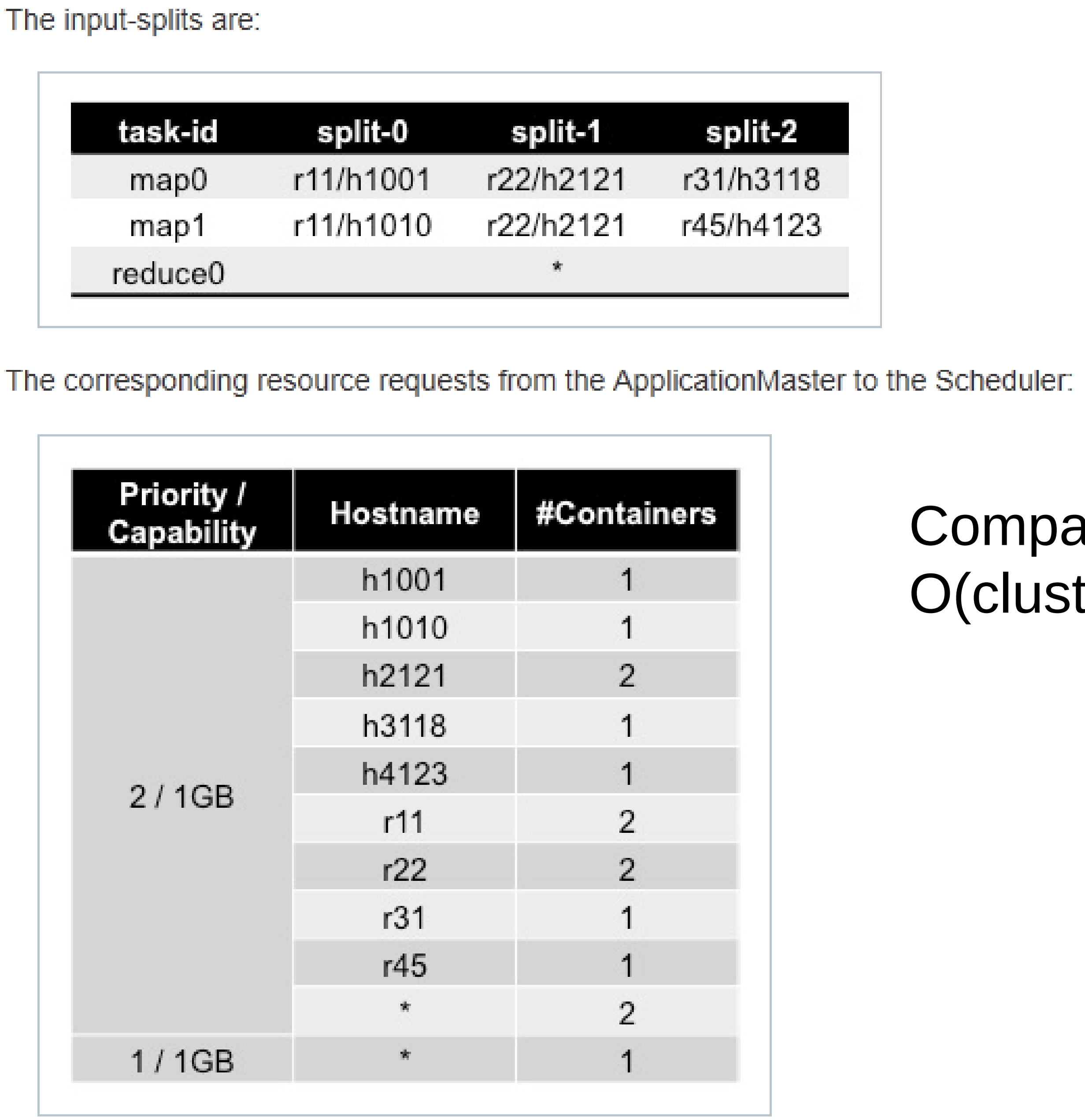
- the resource request understood by the Scheduler is of the form:

<priority, (hostname/rackname/*), capability, #containers>

- **Scheduler API**

There is a single API between the Scheduler and the ApplicationMaster:

(List <Container> newContainers, List <ContainerStatus> containerStatuses) allocate (List <ResourceRequest> ask, List<Container> release)



The input-splits are:

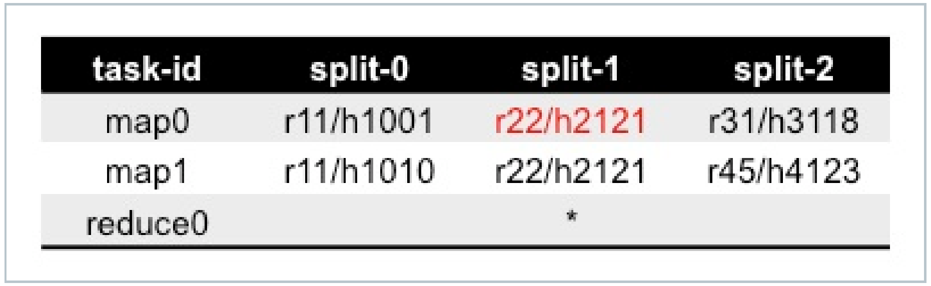
task-id	split-0	split-1	split-2
map0	r11/h1001	r22/h2121	r31/h3118
map1	r11/h1010	r22/h2121	r45/h4123
reduce0		*	

The corresponding resource requests from the ApplicationMaster to the Scheduler:

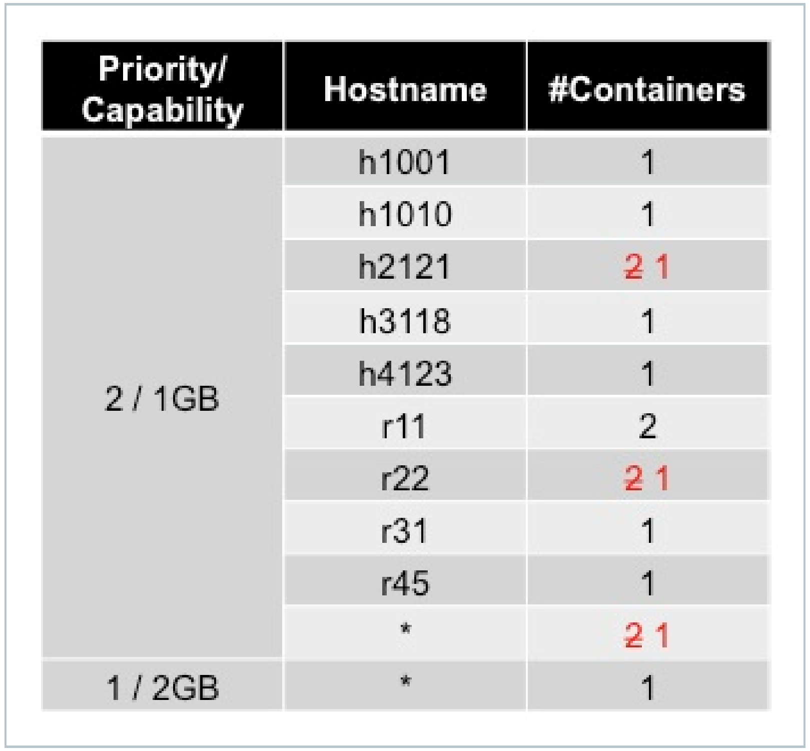
Priority / Capability	Hostname	#Containers
2 / 1GB	h1001	1
	h1010	1
	h2121	2
	h3118	1
	h4123	1
	r11	2
	r22	2
	r31	1
	r45	1
	*	2
1 / 1GB	*	1

Compact:
O(clustersize)

If a container is available on `r22/h2121`, the Scheduler allocates it to the application and invalidates requests on `h2121`, `r22` and `*` by updating the resource request list as follows:



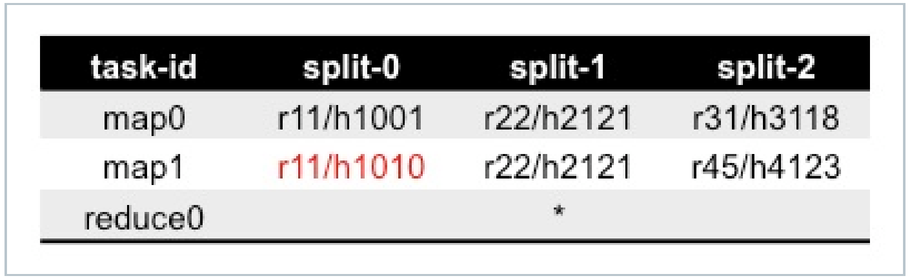
task-id	split-0	split-1	split-2
map0	r11/h1001	r22/h2121	r31/h3118
map1	r11/h1010	r22/h2121	r45/h4123
reduce0		*	



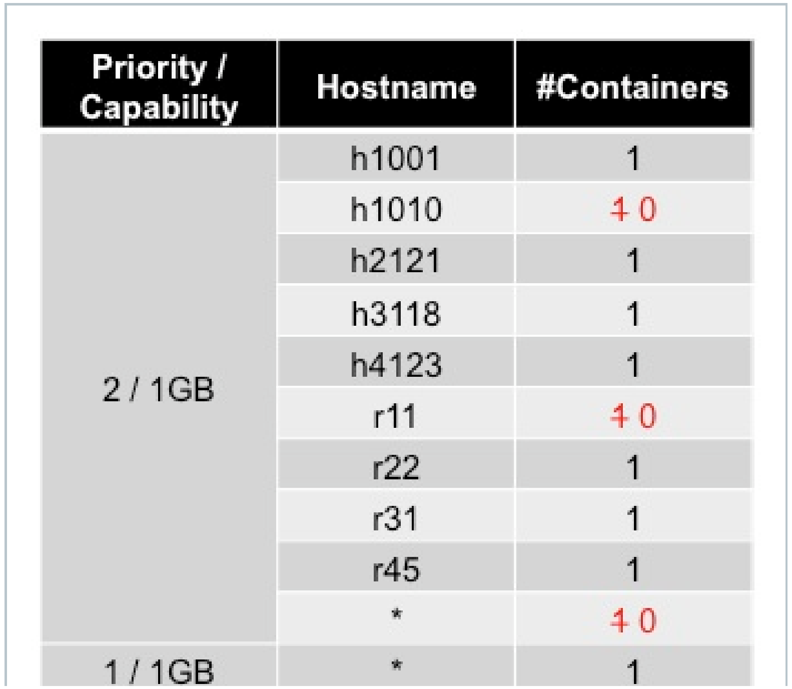
Priority/ Capability	Hostname	#Containers
2 / 1GB	h1001	1
	h1010	1
	h2121	~~2~~ 1
	h3118	1
	h4123	1
	r11	2
	r22	~~2~~ 1
	r31	1
	r45	1
	*	~~2~~ 1
1 / 2GB	*	1

The ApplicationMaster can then, optionally, invalidate requests for `h1001`, `h3118`, `r11` and `r31` after it decides to run *map0* on `h2121`.

Now, if a container is available on `r11/h1010`, the Scheduler allocates it to the application and invalidates requests on `h1010`, `r11` and `*` as follows:

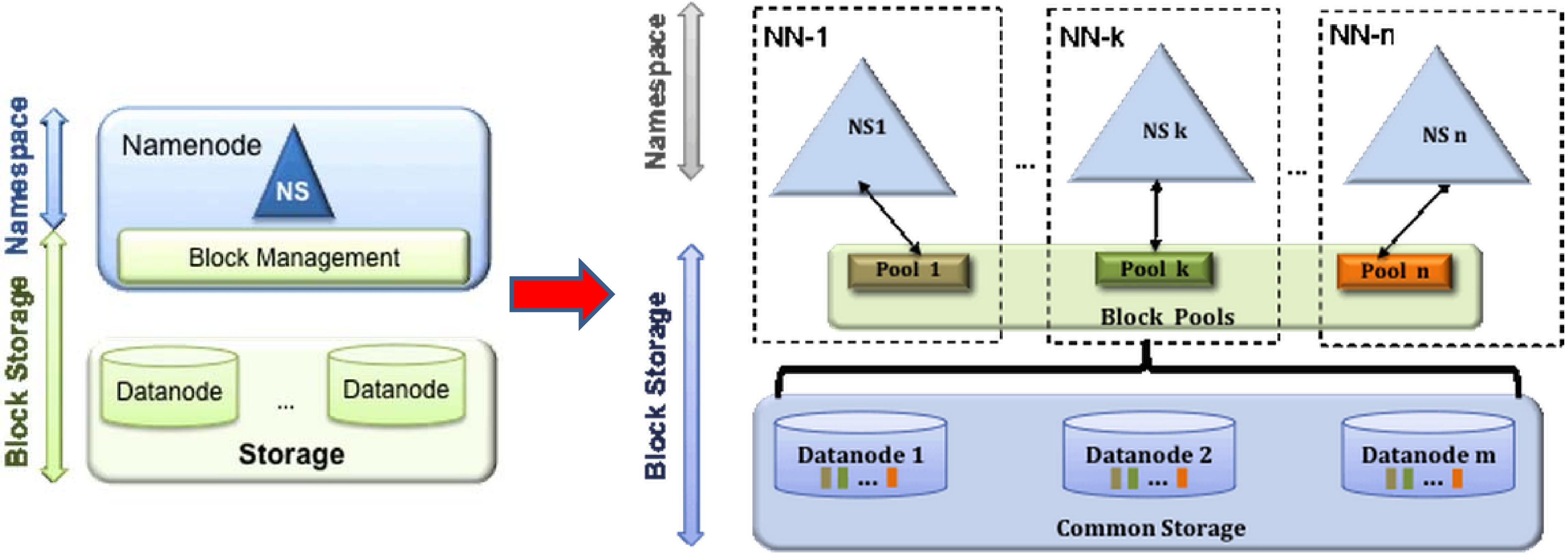


task-id	split-0	split-1	split-2
map0	r11/h1001	r22/h2121	r31/h3118
map1	r11/h1010	r22/h2121	r45/h4123
reduce0		*	



Priority / Capability	Hostname	#Containers
2 / 1GB	h1001	1
	h1010	~~1~~ 0
	h2121	1
	h3118	1
	h4123	1
	r11	~~1~~ 0
	r22	1
	r31	1
	r45	1
	*	~~1~~ 0
1 / 1GB	*	1

HDFS Federation



Benefits:
- * Namespace Scalability
- * Performance
- * Isolation

Q & A