DICE
ANALYTICS

Apache Pig

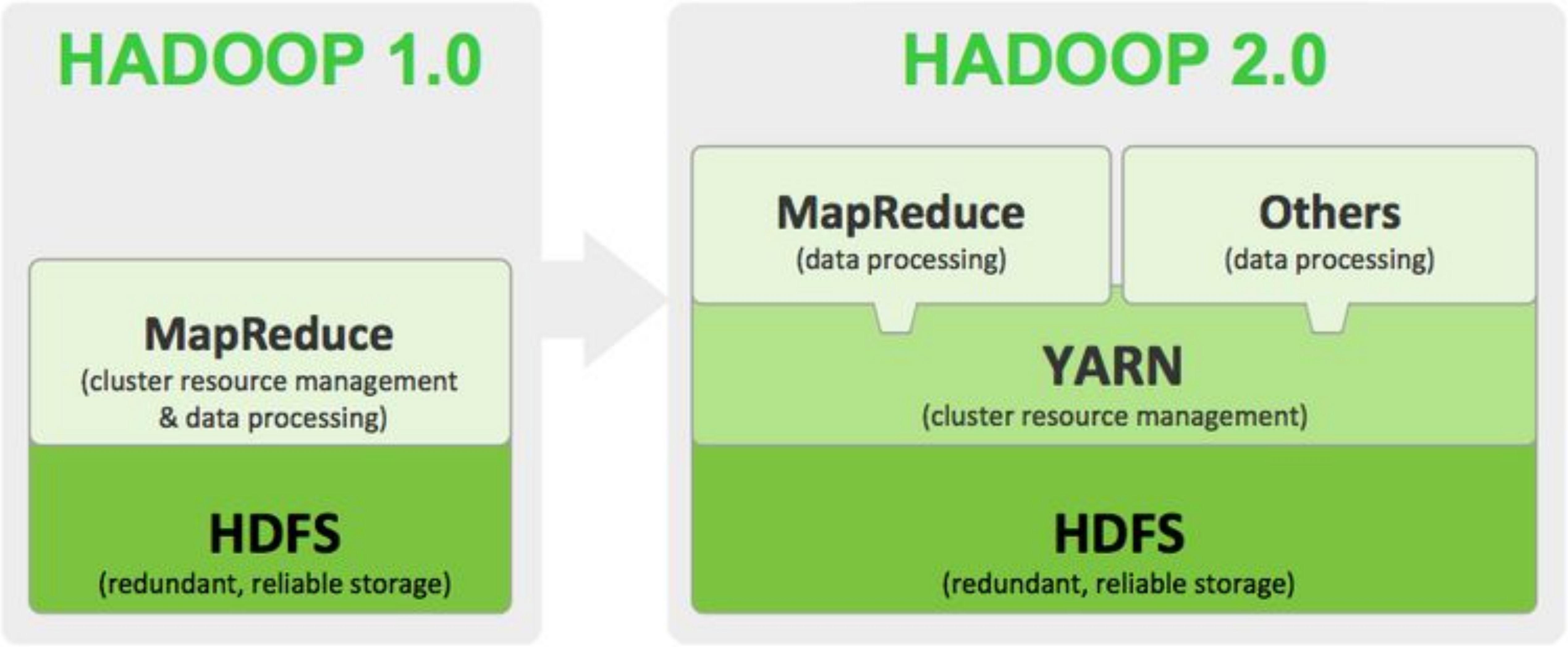
https://www.facebook.com/diceanalytics
/
https://www.linkedin.com/company/13294896
/

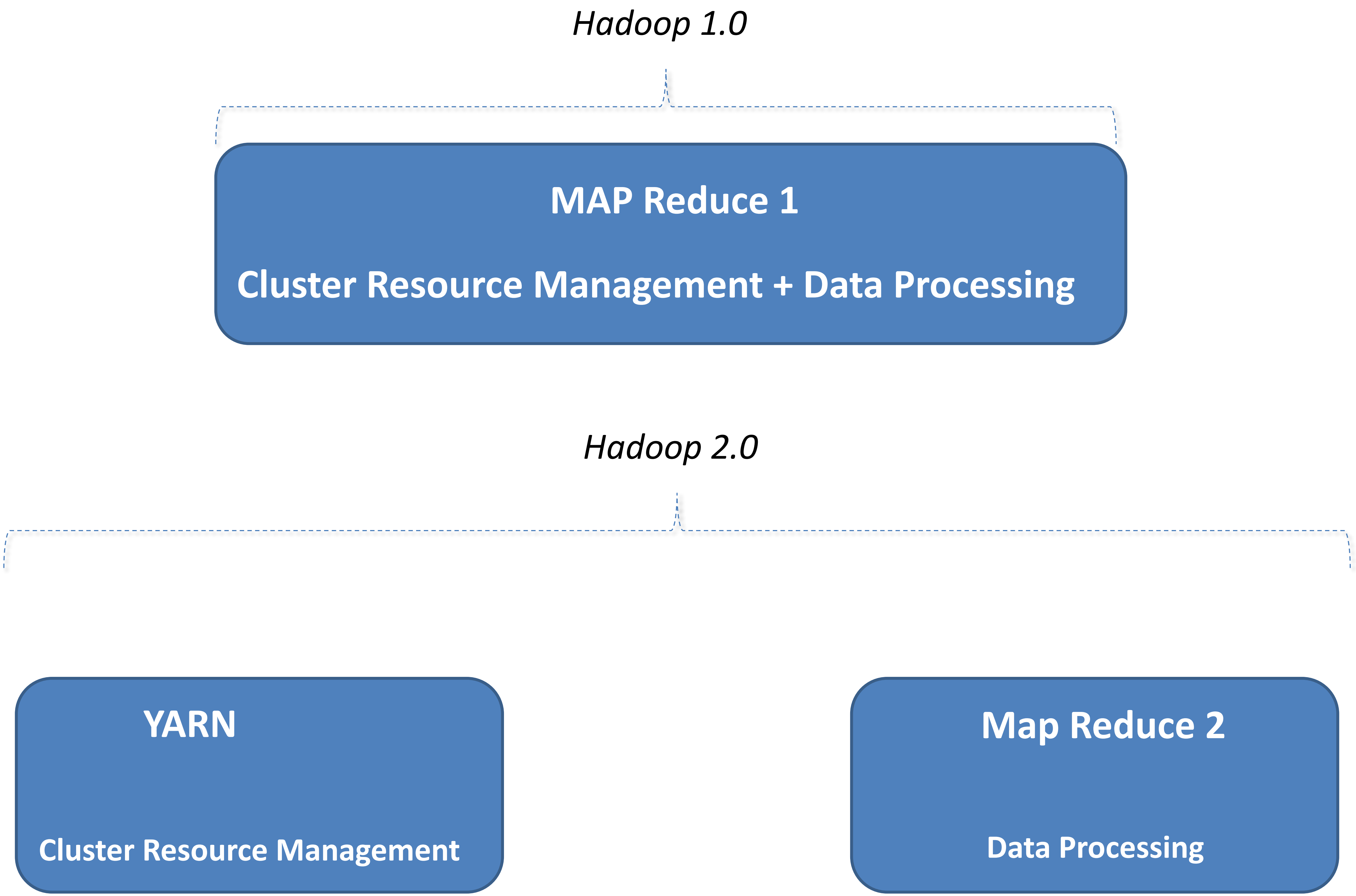
YARN

YET Another Resource Negotiator

https://www.youtube.com/watch?v=DMHf_xiSSg
A

- *Hadoop 2.0 Was Released in 2012*
- *YARN is part of the Hadoop 2.0 and onwards*
- *Previously cluster resource Management was part of the Map Reduce – Hadoop 1.0*


HADOOP 1.0
MapReduce
(cluster resource management & data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MapReduce
(data processing)
Others
(data processing)
YARN
(cluster resource management)
HDFS
(redundant, reliable storage)



Hadoop 1.0

MAP Reduce 1

Cluster Resource Management + Data Processing

Hadoop 2.0

YARN

Cluster Resource Management

Map Reduce 2

Data Processing

YARN to MapReduce: You only do Data Processing because you are best at it. Let me do cluster resource management

Hadoop 2.0

Said Good Bye to

JOB
Tracke
r

TASK
Tracke
r

NOW

its

MAP Reduce 2

Which Only Does Data Processing

- YARN Divide Node Resources into ***Containers***

Container = Fixed **CPU** + Fixed **Memory**

YARN

YET Another Resource Negotiator

Global Resource Manager

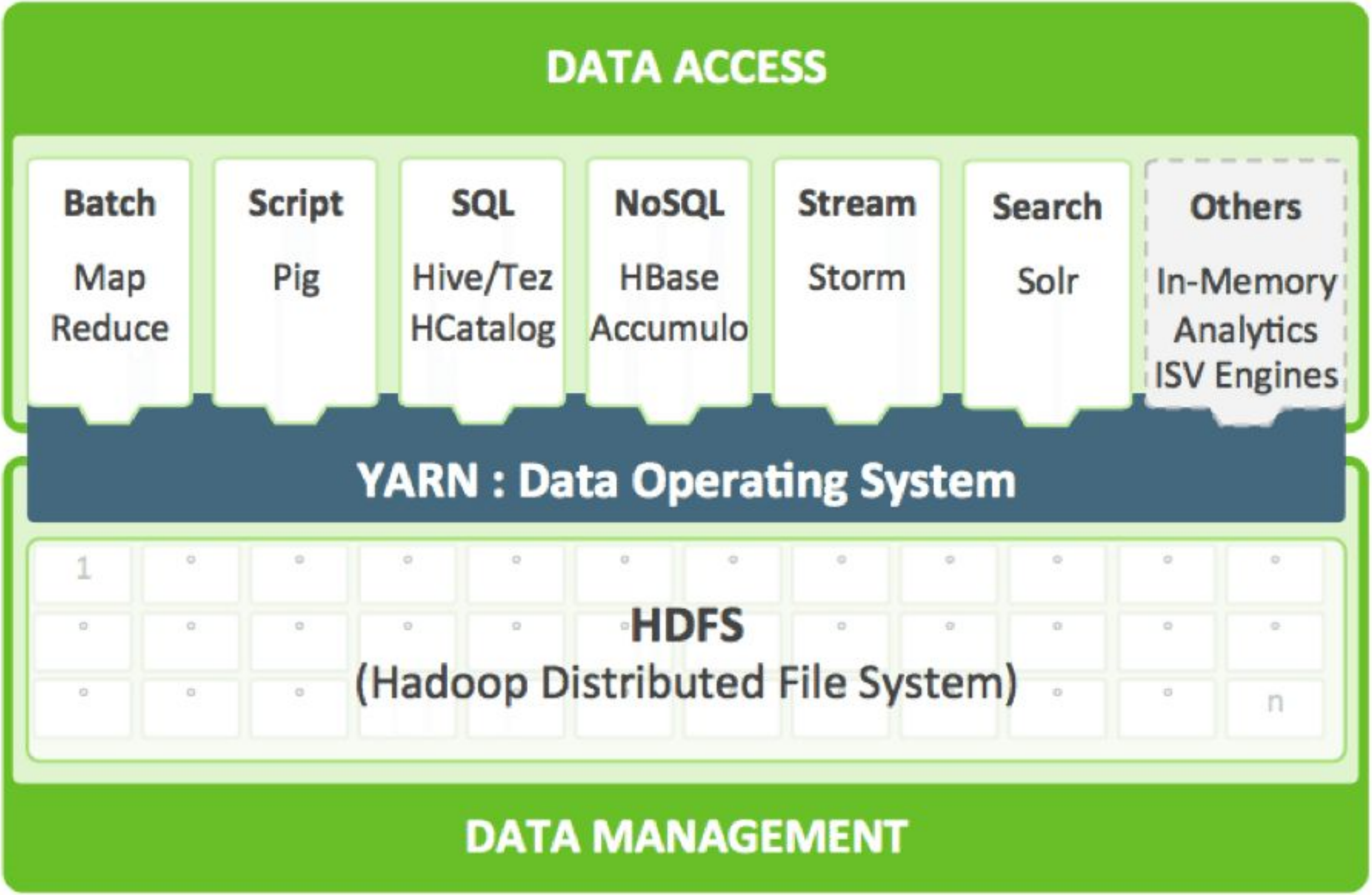
- ✔ Per Cluster Service
- ✔ Authority that distributes resources among applications

Node Manager

- ✔ Per Machine Service
- ✔ Creates Execution Containers & monitor their resources and reports to Resource Manager

Application Master

- ✔ Per Application Framework
- ✔ Negotiate resources with the Resource Manager
- ✔ Works with the Node Manager to execute and monitor the tasks



DATA ACCESS

Batch
Map Reduce

Script
Pig

SQL
Hive/Tez
HCatalog

NoSQL
HBase
Accumulo

Stream
Storm

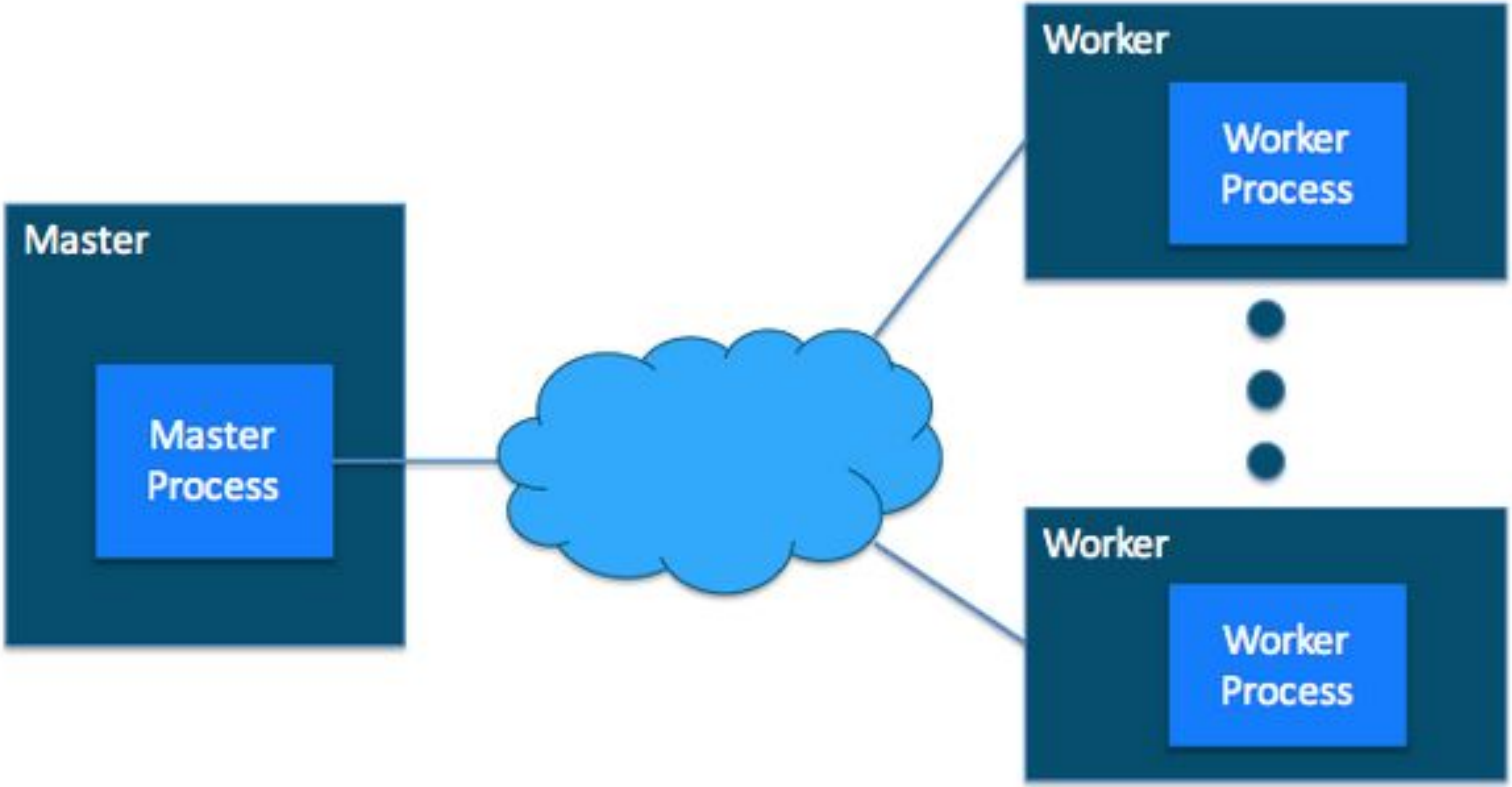
Search
Solr

Others
In-Memory
Analytics
ISV Engines

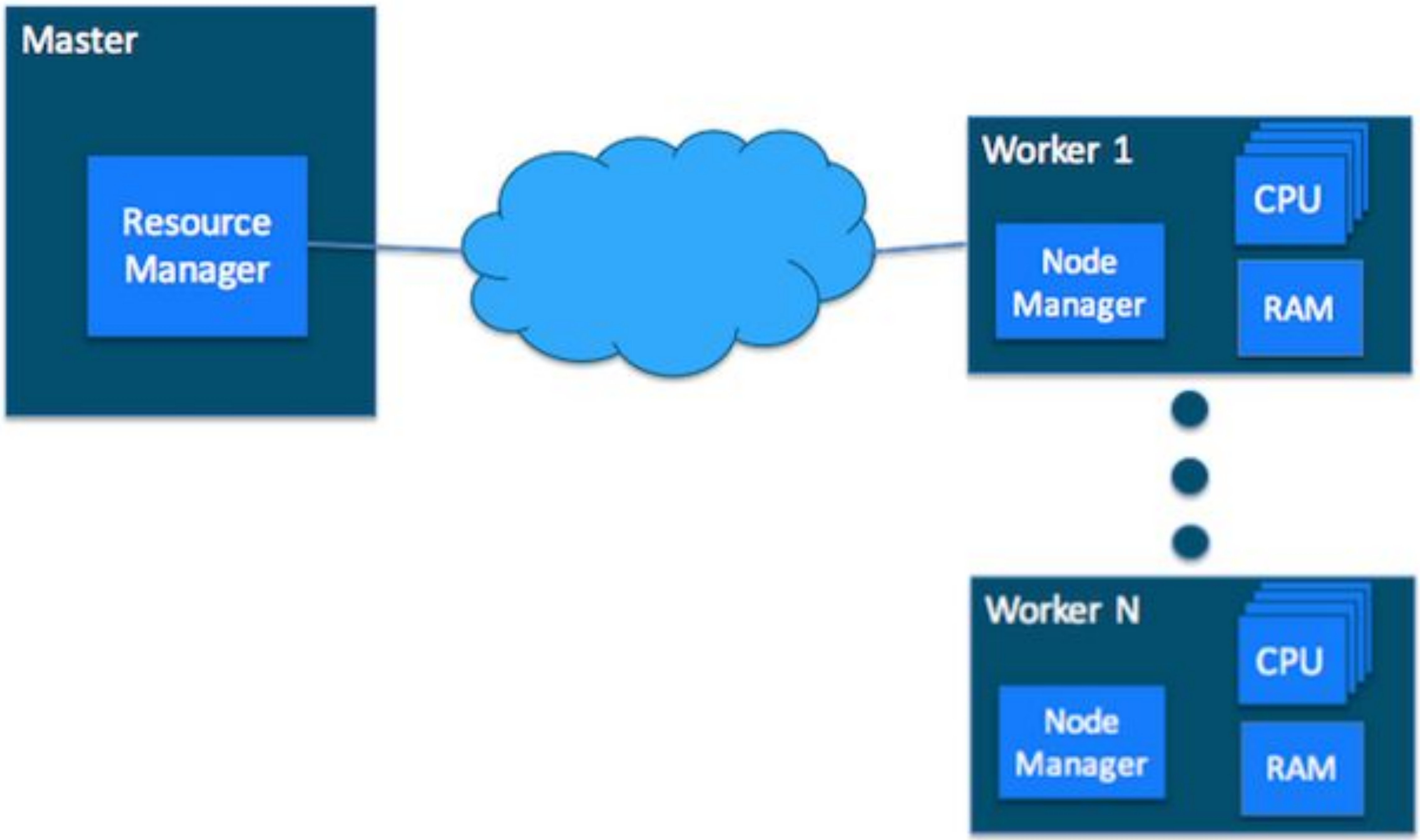
YARN : Data Operating System

HDFS
(Hadoop Distributed File System)

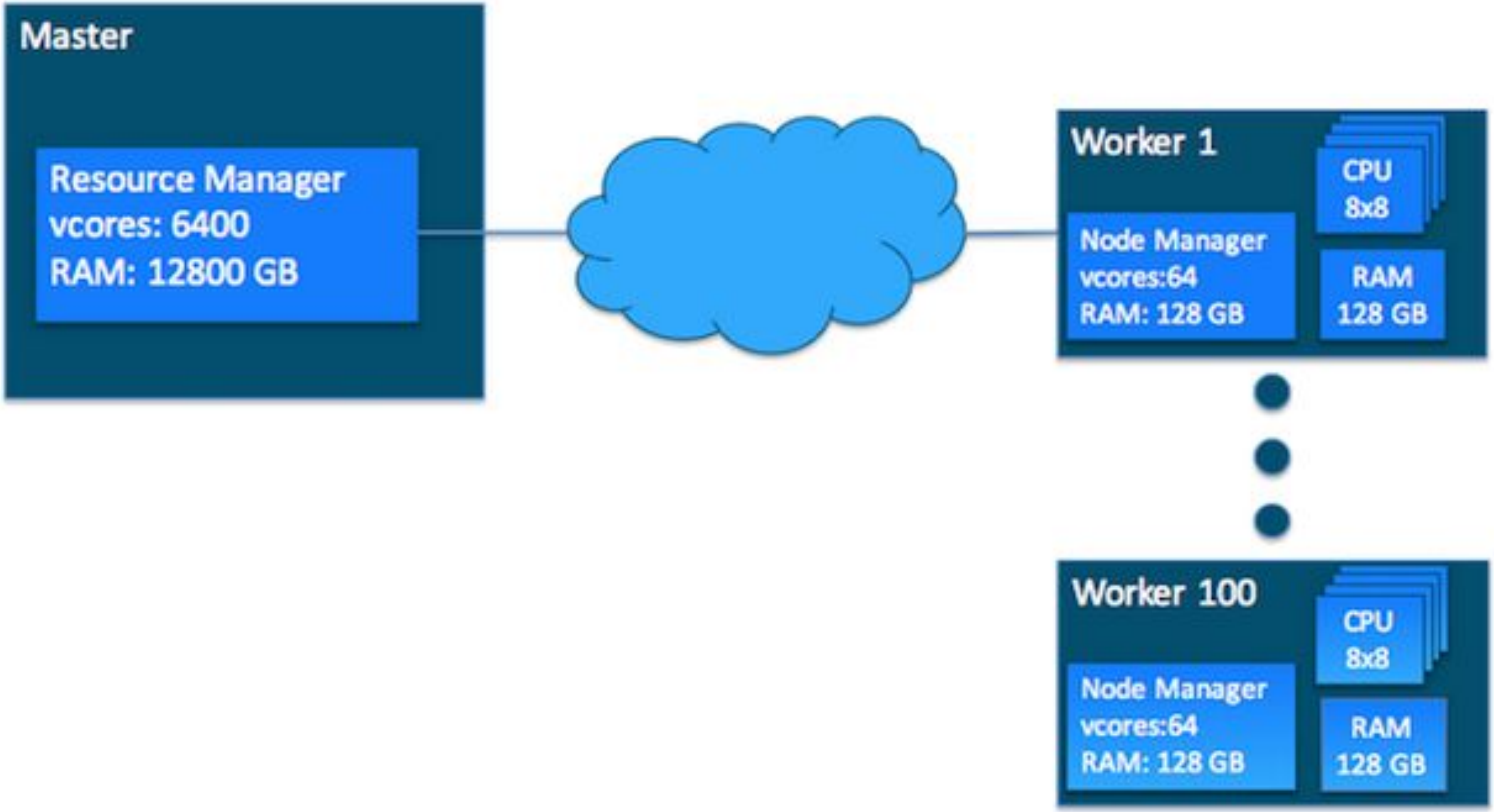
DATA MANAGEMENT



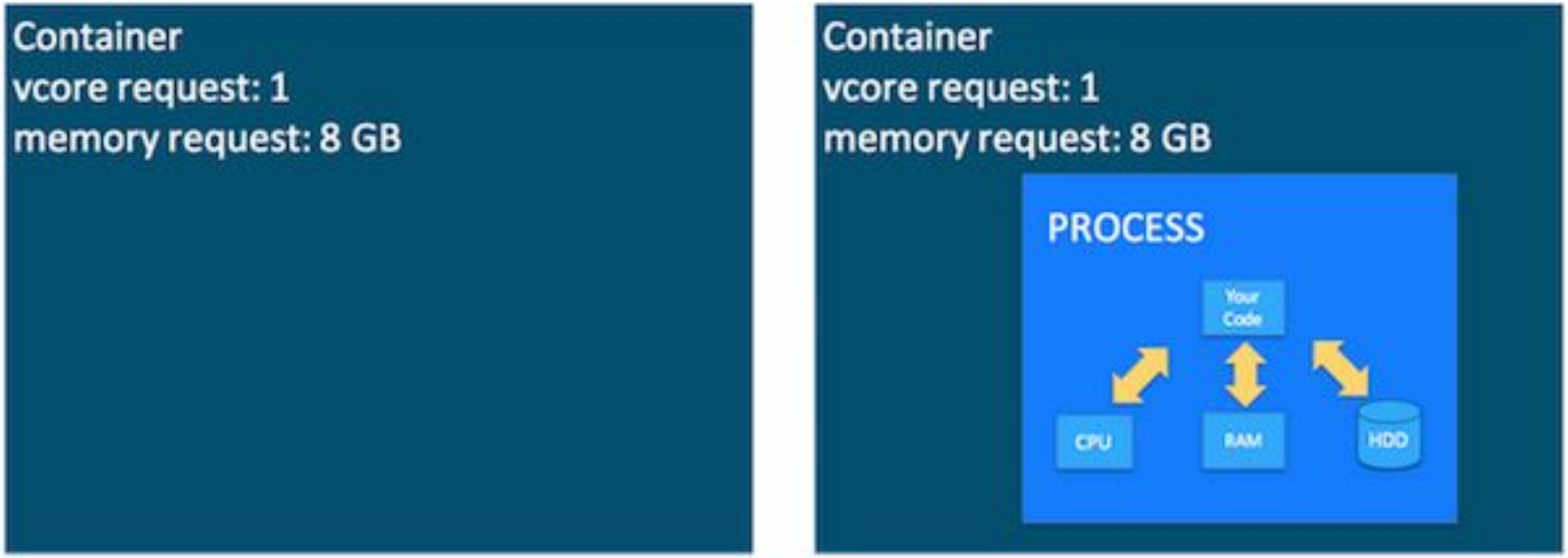
https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



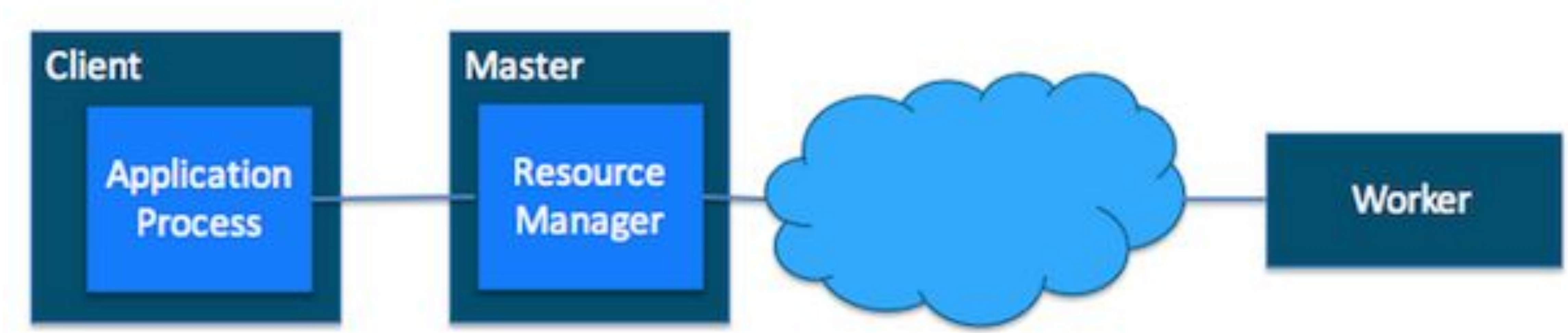
https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



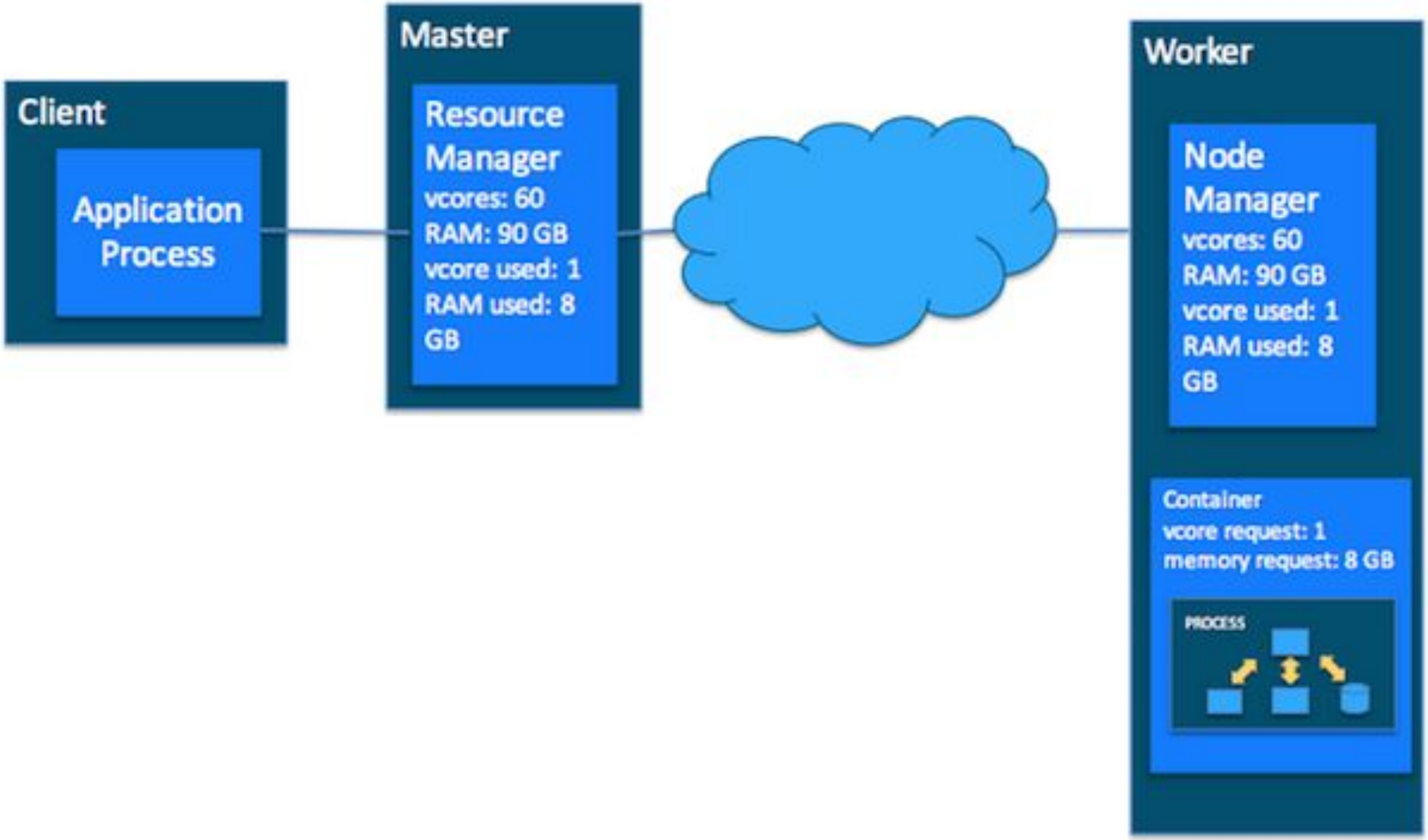
https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



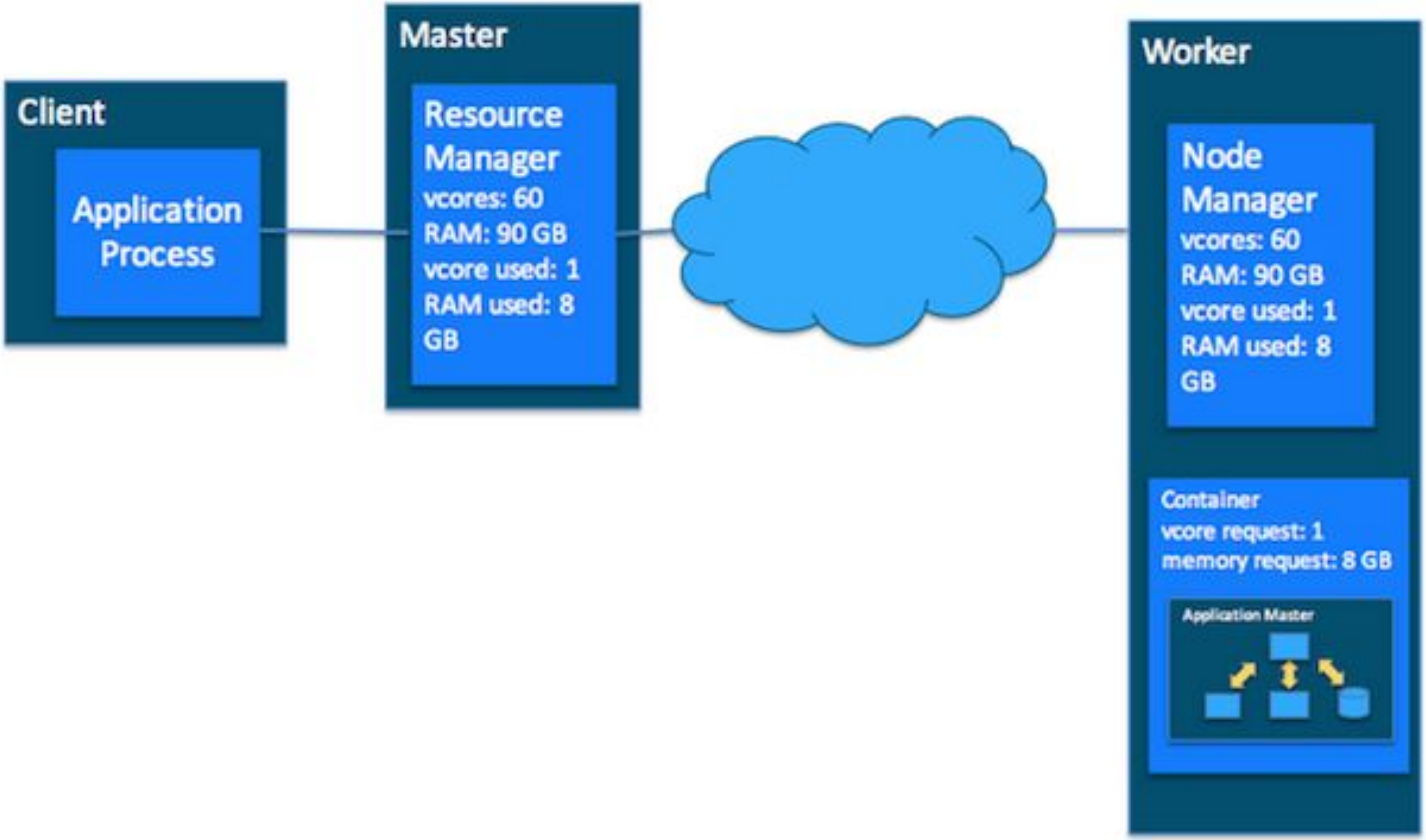
https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



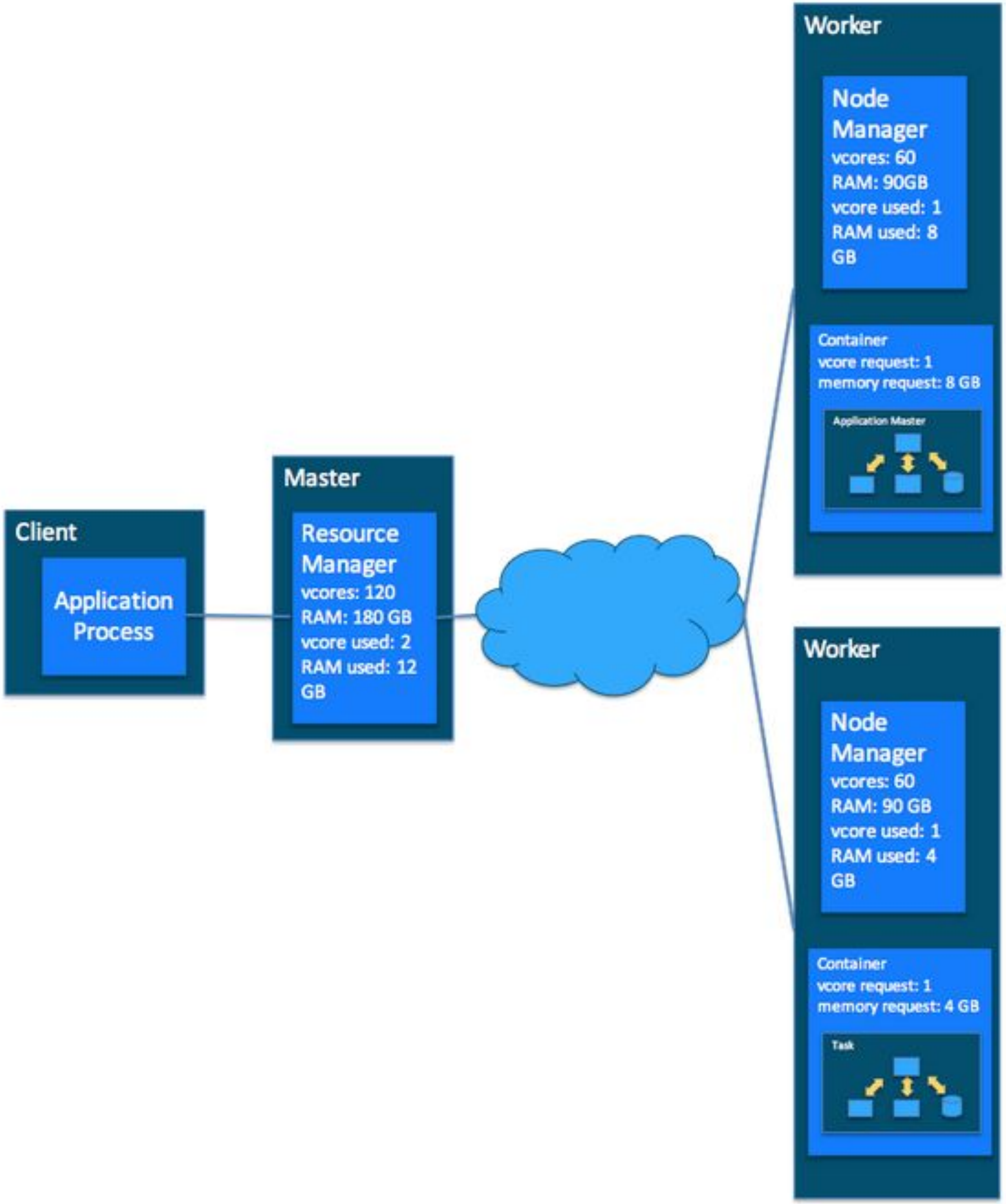
https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



Client

Application Process

Master

Resource Manager
vcores: 120
RAM: 180 GB
vcore used: 2
RAM used: 12 GB

Worker

Node Manager
vcores: 60
RAM: 90GB
vcore used: 1
RAM used: 8 GB

Container
vcore request: 1
memory request: 8 GB

Application Master

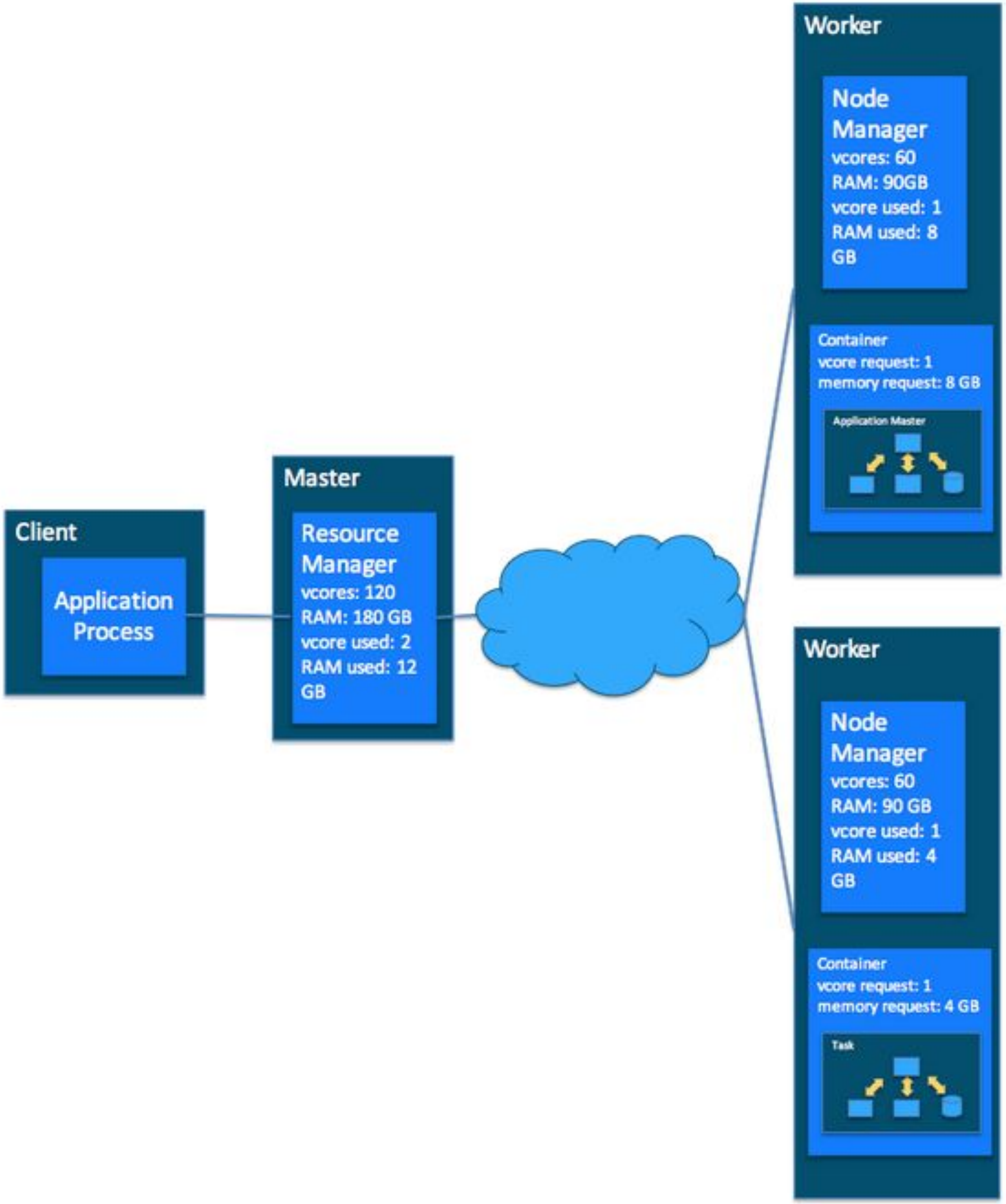
Worker

Node Manager
vcores: 60
RAM: 90 GB
vcore used: 1
RAM used: 4 GB

Container
vcore request: 1
memory request: 4 GB

Task

https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



Client

Application Process

Master

Resource Manager
vcores: 120
RAM: 180 GB
vcore used: 2
RAM used: 12 GB

Worker

Node Manager
vcores: 60
RAM: 90GB
vcore used: 1
RAM used: 8 GB

Container
vcore request: 1
memory request: 8 GB

Application Master

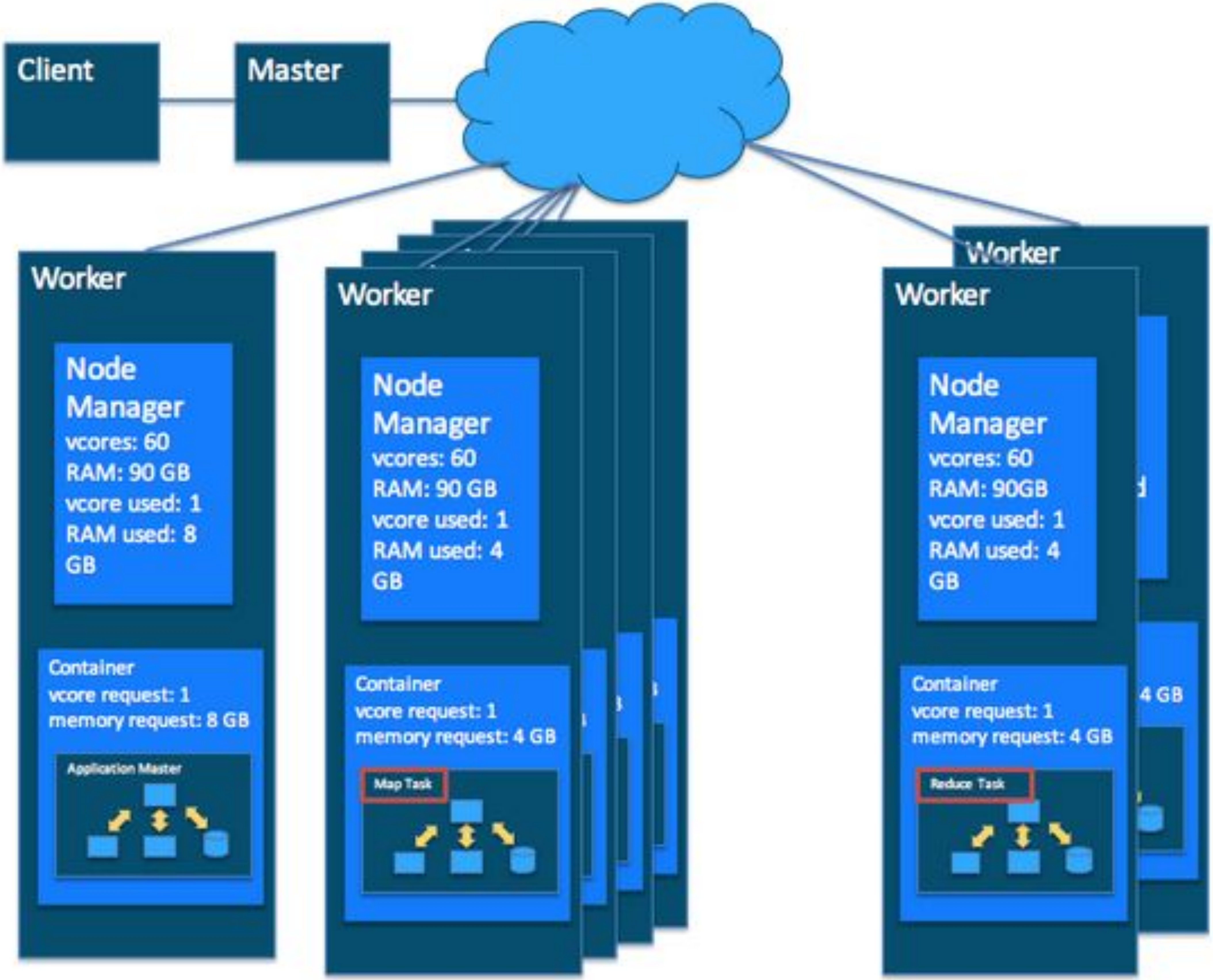
Worker

Node Manager
vcores: 60
RAM: 90 GB
vcore used: 1
RAM used: 4 GB

Container
vcore request: 1
memory request: 4 GB

Task

https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



Client

Master

Worker

Node Manager
vcores: 60
RAM: 90 GB
vcore used: 1
RAM used: 8 GB

Container
vcore request: 1
memory request: 8 GB

Application Master

Worker

Node Manager
vcores: 60
RAM: 90 GB
vcore used: 1
RAM used: 4 GB

Container
vcore request: 1
memory request: 4 GB

Map Task

Worker

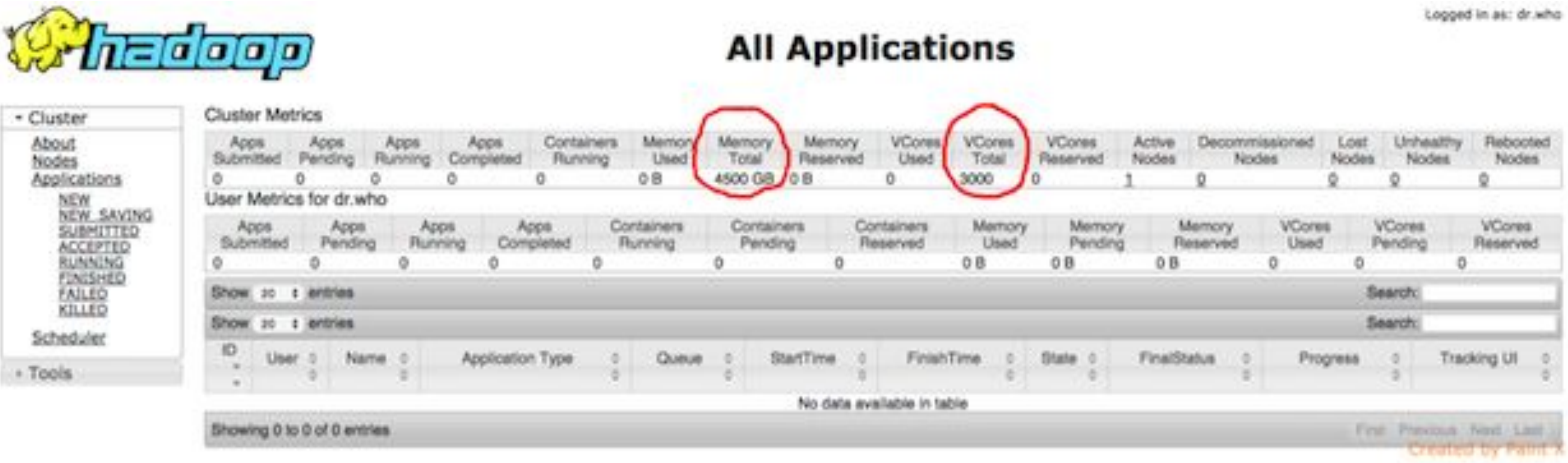
Worker

Node Manager
vcores: 60
RAM: 90GB
vcore used: 1
RAM used: 4 GB

Container
vcore request: 1
memory request: 4 GB

Reduce Task

https://blog.cloudera.com/blog/2015/09/untangling-apache-hadoop-yarn-part-1/



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
0	0	0	0	0	0 B	4500 GB	0 B	0	3000	0	1	0	0	0	0

User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved	VCores Used	VCores Pending	VCores Reserved
0	0	0	0	0	0	0	0 B	0 B	0 B	0	0	0

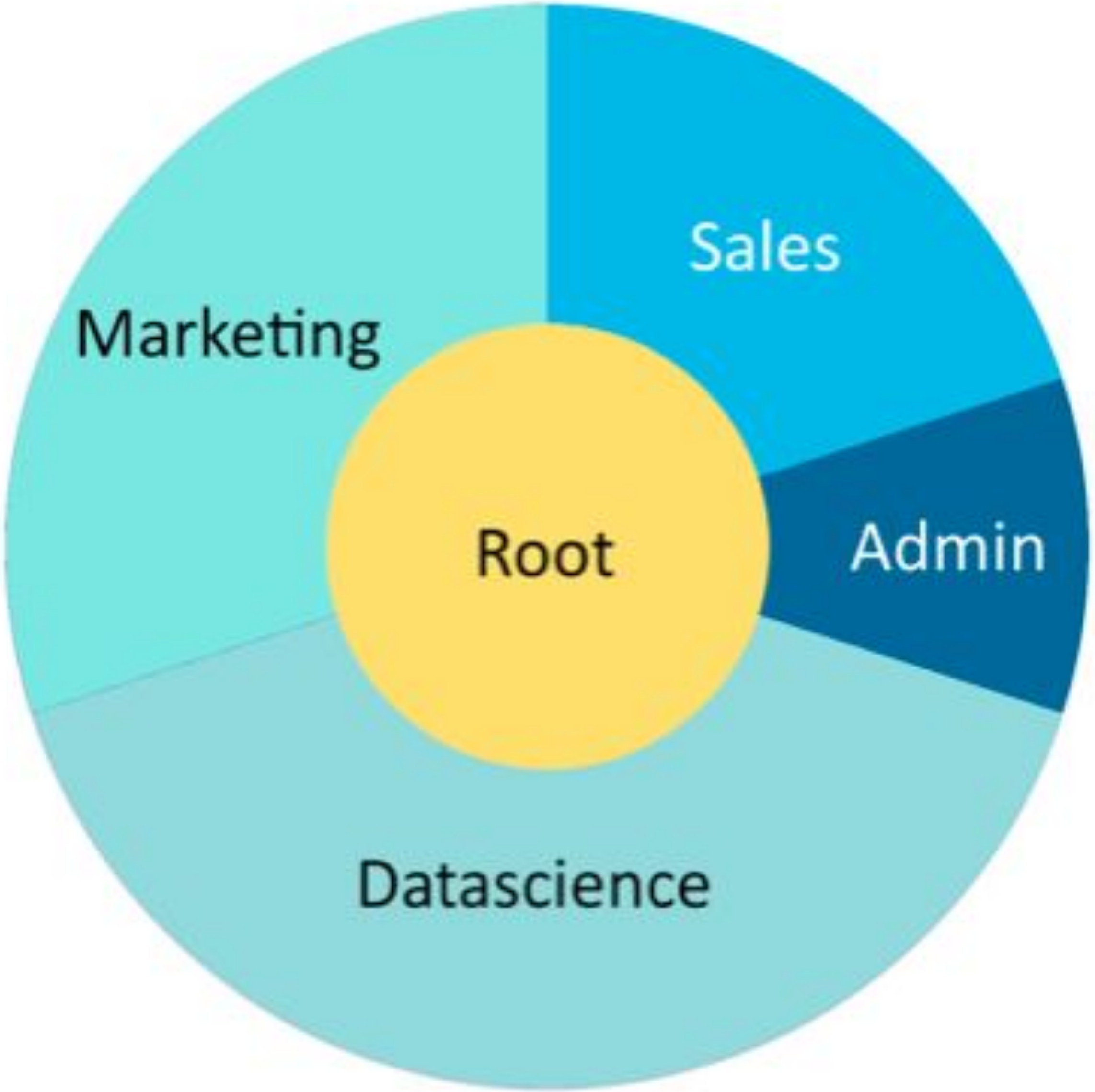
Show 20 entries Search:

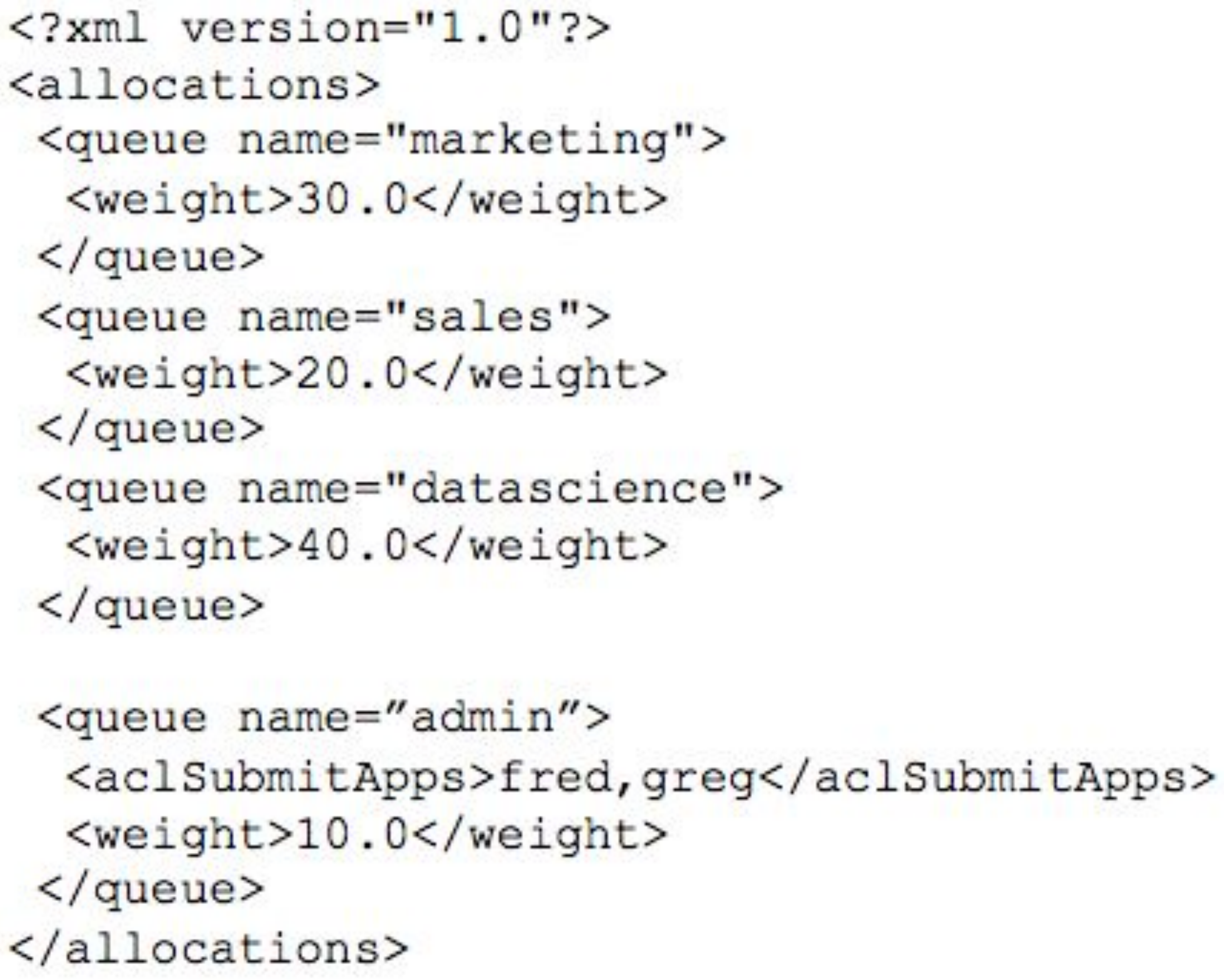
Show 20 entries Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
No data available in table										

Showing 0 to 0 of 0 entries First Previous Next Last

http://blog.cloudera.com/blog/2015/10/untangling-apache-hadoop-yarn-part-2/

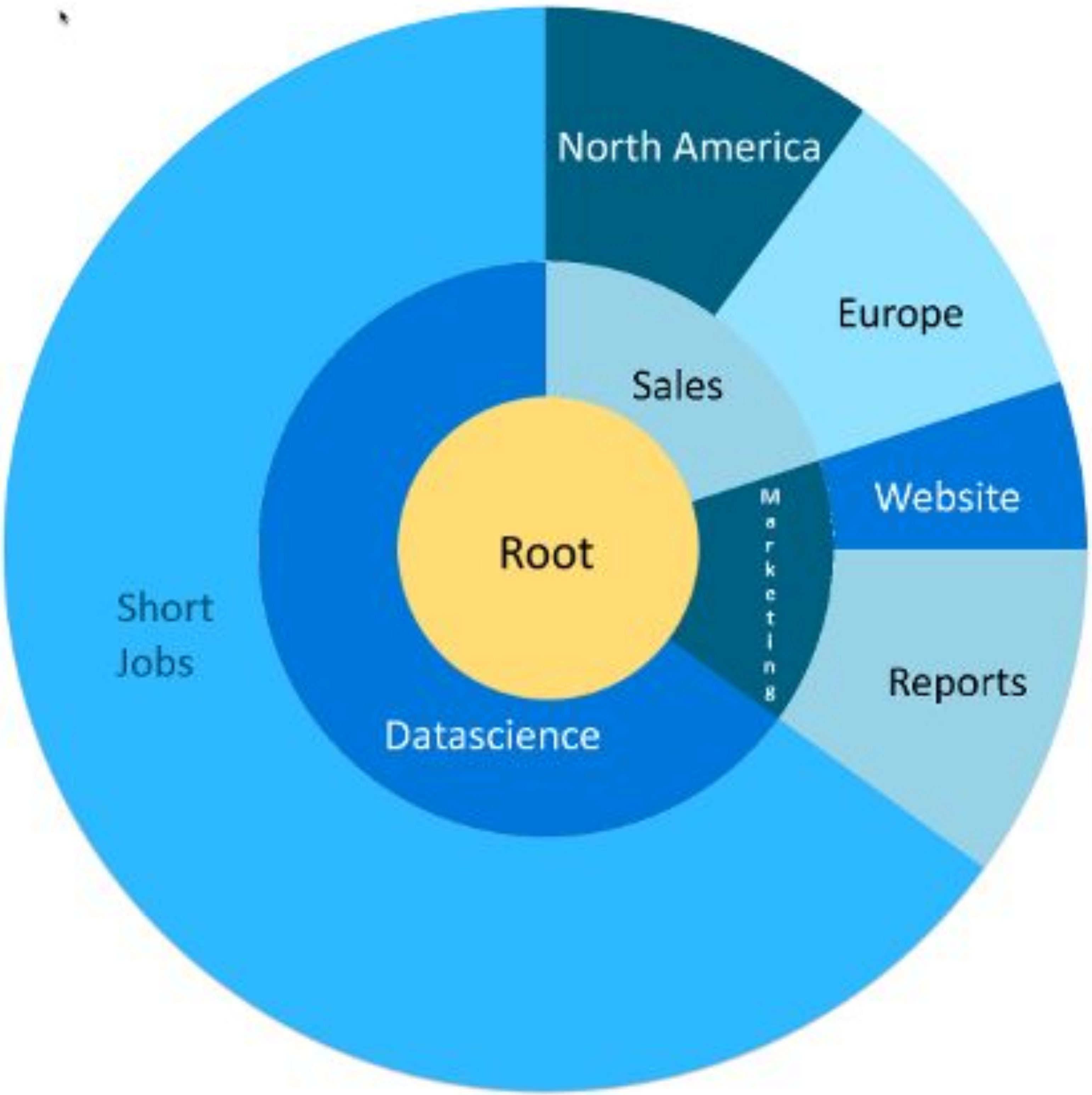




```
<?xml version="1.0"?>
<allocations>
 <queue name="marketing">
  <weight>30.0</weight>
 </queue>
 <queue name="sales">
  <weight>20.0</weight>
 </queue>
 <queue name="datascience">
  <weight>40.0</weight>
 </queue>

 <queue name=”admin”>
  <aclSubmitApps>fred,greg</aclSubmitApps>
  <weight>10.0</weight>
 </queue>
</allocations>
```

http://blog.cloudera.com/blog/2015/10/untangling-apache-hadoop-yarn-part-2/



```xml
<?xml version="1.0"?>
<allocations>
 <queue name="marketing">
  <weight>3.0</weight>
  <queue name="reports">
   <weight>40.0</weight>
  </queue>
  <queue name="website">
   <weight>20.0</weight>
  </queue>
 </queue>
 <queue name="sales">
  <weight>4.0</weight>
  <queue name="northamerica">
   <weight>30.0</weight>
  </queue>
  <queue name="europe">
   <weight>30.0</weight>
  </queue>
 </queue>
 <queue name="datascience">
  <weight>13.0</weight>
  <queue name="short_jobs">
   <weight>100.0</weight>
  </queue>
  <queue name="best_effort_jobs">
   <weight>0.0</weight>
  </queue>
 </queue>
</allocations>
```

Assignment

Fair & Capacity Scheduler

LIFO & FIFO Concept in YARN

https://docs.hortonworks.com/HDPDocuments/Ambari-2.6.0.0/bk_ambari-views/content/ch_using_yarn_queue_manager_view.html

BRAD PITT

Rolling Stone PETER TRAVERS "ONE OF THE BEST FILMS OF THE YEAR."

"BRAD PITT IS SENSATIONAL." TIME RICHARD CORLISS

MONEYBALL

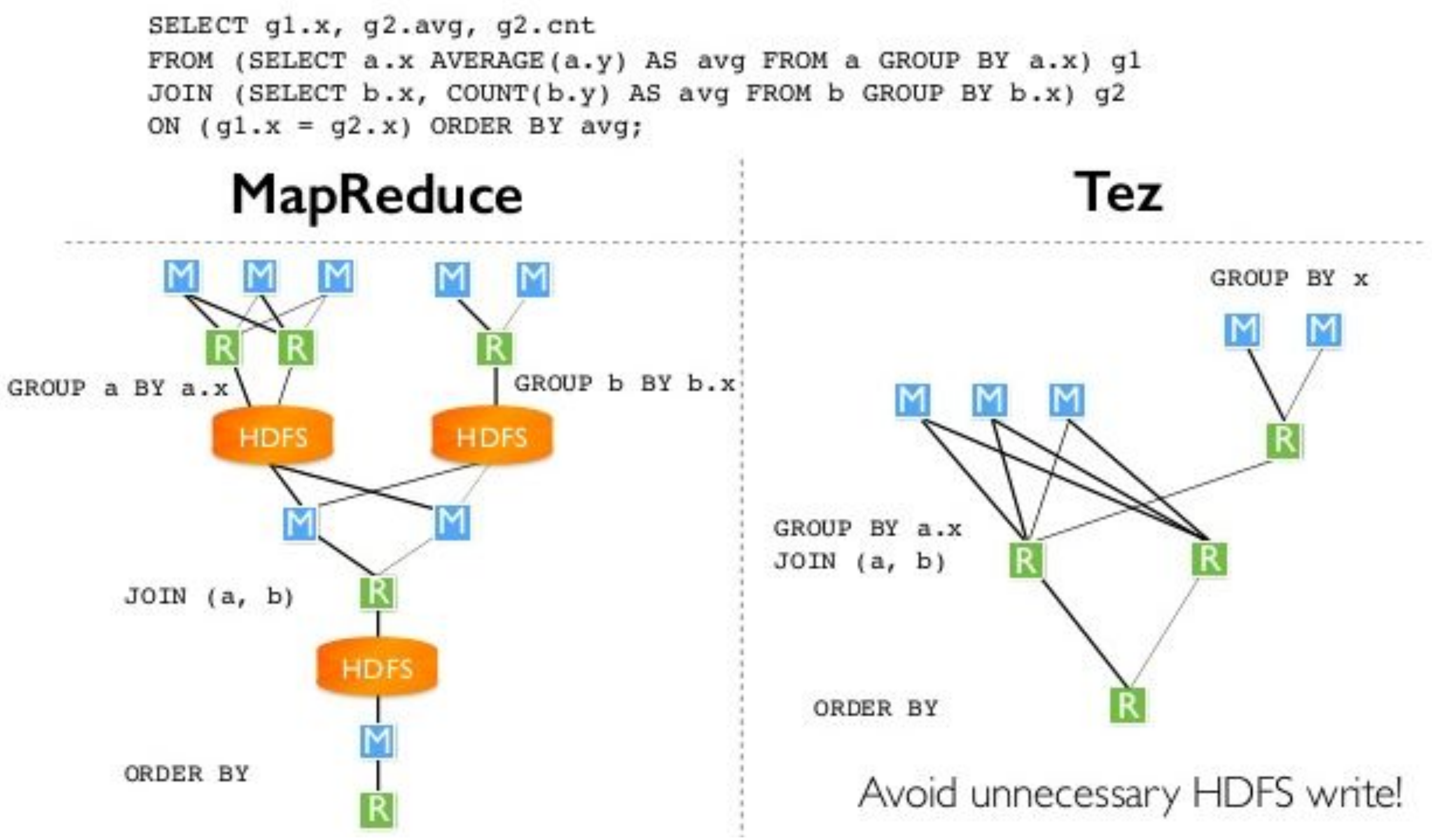
JONAH HILL PHILIP SEYMOUR HOFFMAN

BASED ON A TRUE STORY

IN THEATERS SEPTEMBER 23

TEZ

Tez

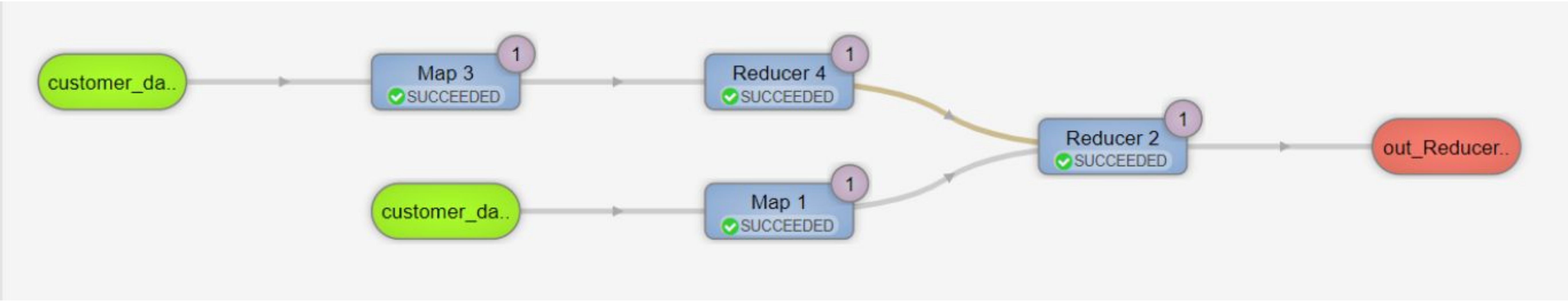


```
SELECT g1.x, g2.avg, g2.cnt
FROM (SELECT a.x AVERAGE(a.y) AS avg FROM a GROUP BY a.x) g1
JOIN (SELECT b.x, COUNT(b.y) AS avg FROM b GROUP BY b.x) g2
ON (g1.x = g2.x) ORDER BY avg;
```

Avoid unnecessary HDFS write!

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0051-6

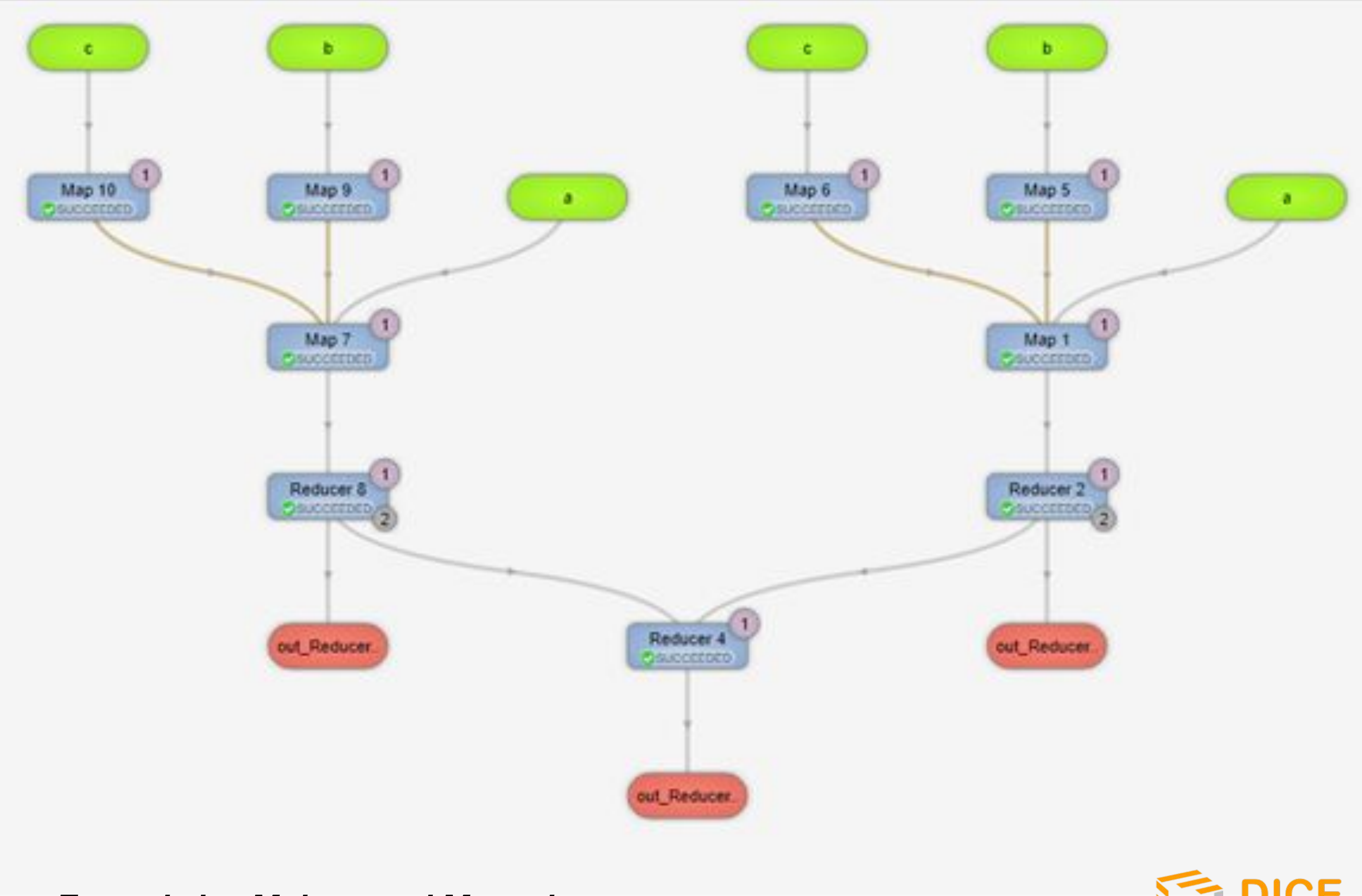
```
SELECT
      *
FROM
      (SELECT PROVINCE,COUNT(*) AS CNT FROM customer_data GROUP BY PROVINCE ) AS A
LEFT JOIN
      (SELECT PROVINCE,COUNT(*) AS CNT FROM customer_data GROUP BY PROVINCE ) AS B
ON
A.PROVINCE = B.PROVINCE;
```



Example by: Sheharyar Alam BD-03

DICE ANALYTICS

```
Select
a.name,count(*) from testdb.mytable
a,testdb.mytable b,testdb.mytable c
where a.name=b.name
and a.name=c.name
group by a.name
union
select
a.name,count(*) from testdb.mytable
a,testdb.mytable b,testdb.mytable c
where a.name=b.name
```



Example by: Muhammad Muneeb
BD-03

Hive Optimizations

http://chennaihug.org/knowledgebase/673/

https://www.slideshare.net/Hadoop_Summit/w-235phall1pandey