



LEARN. CONNECT. ELEVATE.

YCBS 257 - Data at Scale (Winter 2019)
Instructor: Khaled Tannir



McGill

School of
Continuing Studies

[mcgill.ca
/continuingstudies](http://mcgill.ca/continuingstudies)

School of Continuing Studies
YCBS 257 - Data at Scale (BIG DATA)

Course 2

Hadoop Concepts

Khaled Tannir

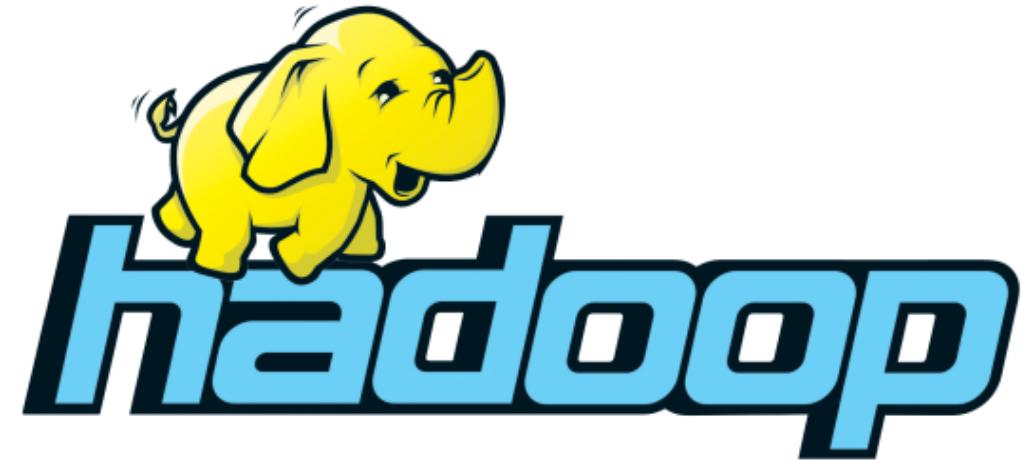


McGill

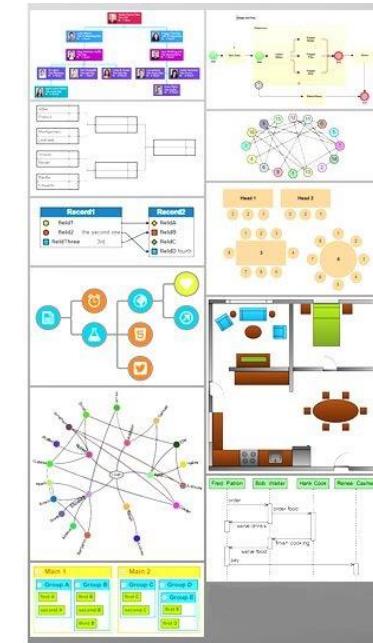
Theme of this Course

Hadoop Concepts

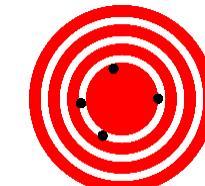
- *Introduction to Hadoop*
- *History, Performances, Core Components*
- *How to install Hadoop*
- *The Hadoop ecosystem*
- *Hadoop Distributions - Hadoop top vendors comparison*



Machine Learning with Small DataSets

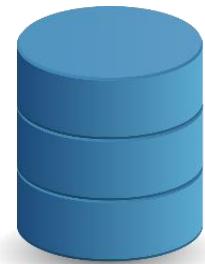
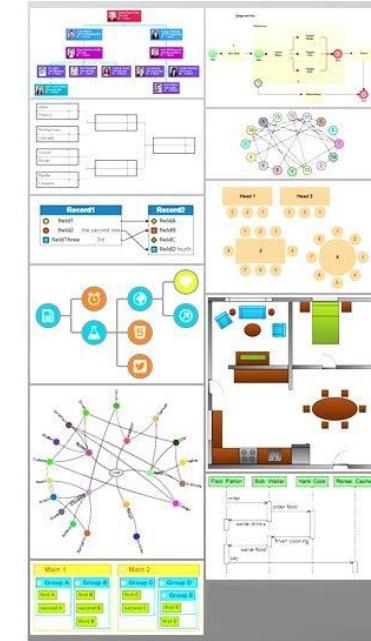


Small DataSet



Low Accuracy

Machine Learning at Scale (Big Data)

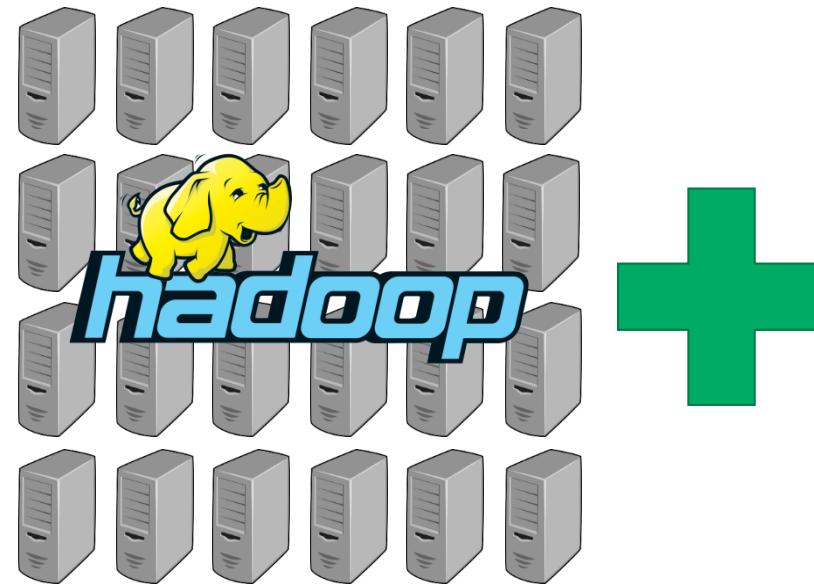


Big DataSet



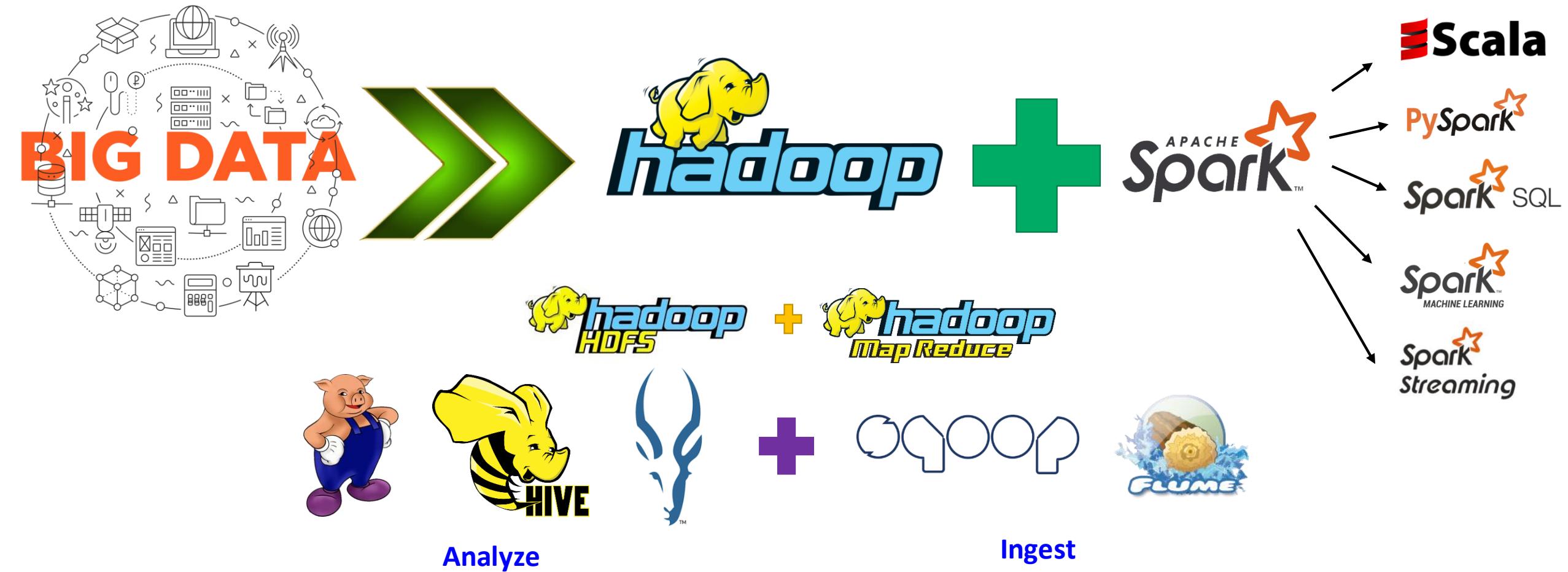
High Accuracy

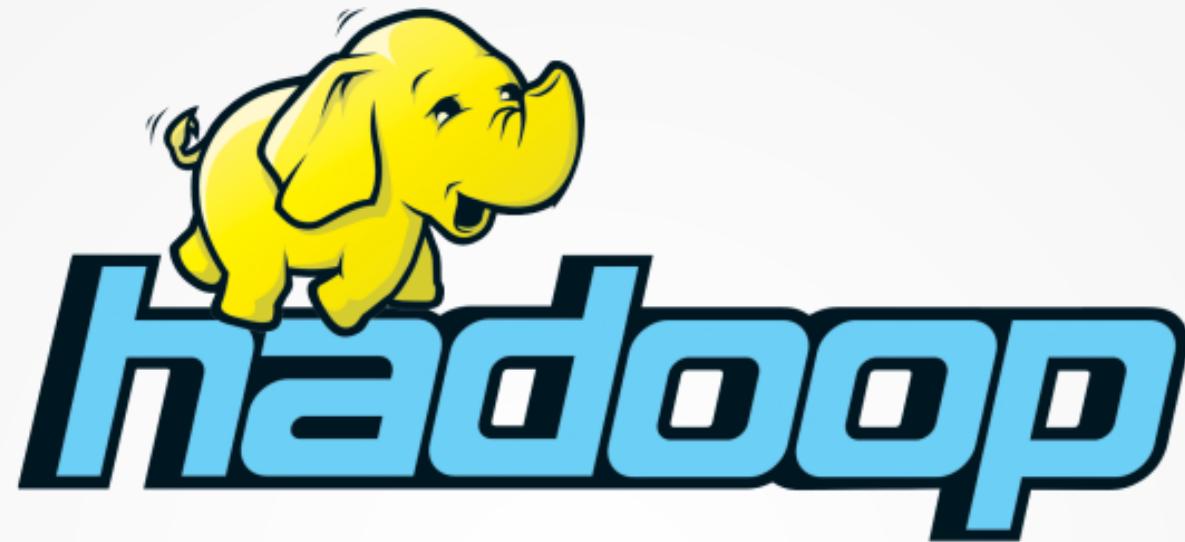
Machine Learning with Big DataSet



ML w/Big Data => Distributed Storage + Distributed Processing

Machine Learning at Scale





Hadoop Concepts



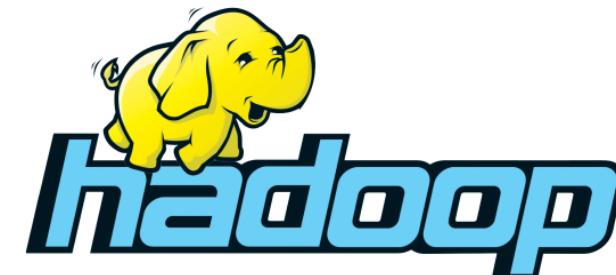
Diving into the Hadoop ecosystem



McGill
FALL 2018

What is Hadoop ?

- Hadoop is Open-source **scalable fault-tolerant** framework for large scale data **storage and processing**
- Written in Java
- Designed for Distributed storage and processing
 - *Distribute data when the data is stored*
 - *Process data where the data is*
 - *Data is replicated*
- Scales horizontally - Shared nothing architecture
- Optimized for high throughput on sequential data access
- Runs on commodity hardware



<http://hadoop.apache.org>

Why do I need Hadoop?



- **Hadoop provides storage for Big Data at reasonable cost**

Enterprises usually spend around \$25,000 to \$50,000 dollars per tera byte per year. With Hadoop this cost drops to few thousands of dollars per tera byte per year. (source : Cloudera)

- **Hadoop allows to capture new or more data**

- **With Hadoop, you can store data and historical data longer**

- **Hadoop provides scalable analytics**

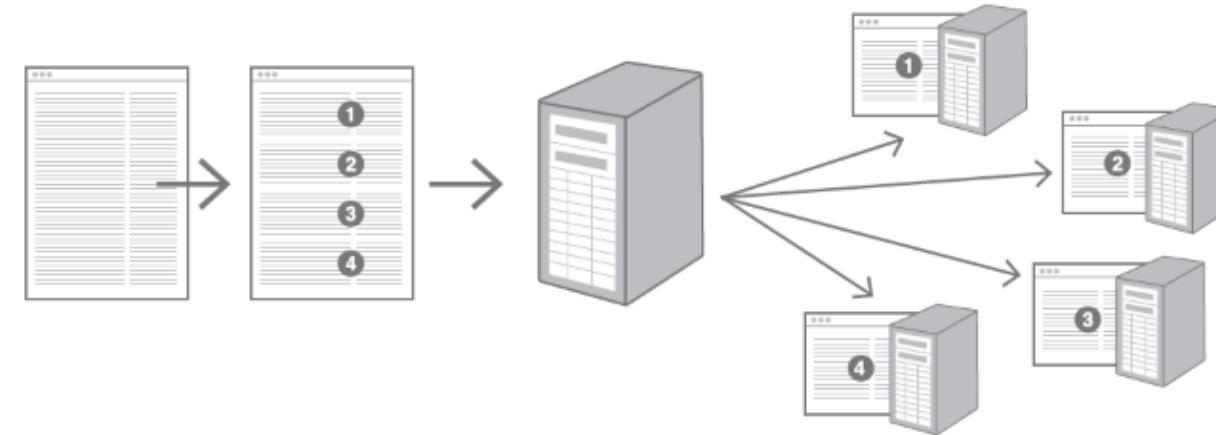
Hadoop provides distributed storage and distributed processing

- **Hadoop provides rich analytics**

Java is the primary language, but other languages like Ruby, Python and R can be used as well

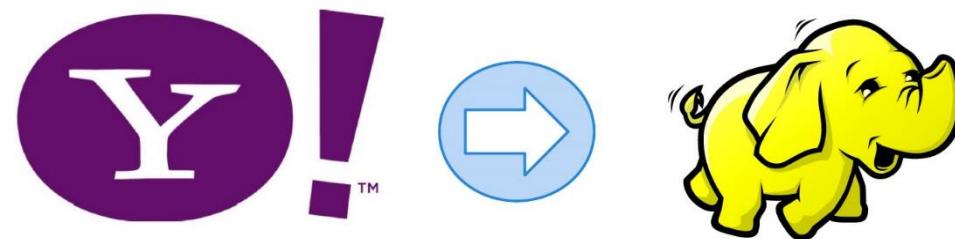
What makes Hadoop unique

- Moving computation to data, instead of moving data to computation
- Simplified programming model
allows user to quickly write and test
- Automatic distribution of data and work across machines



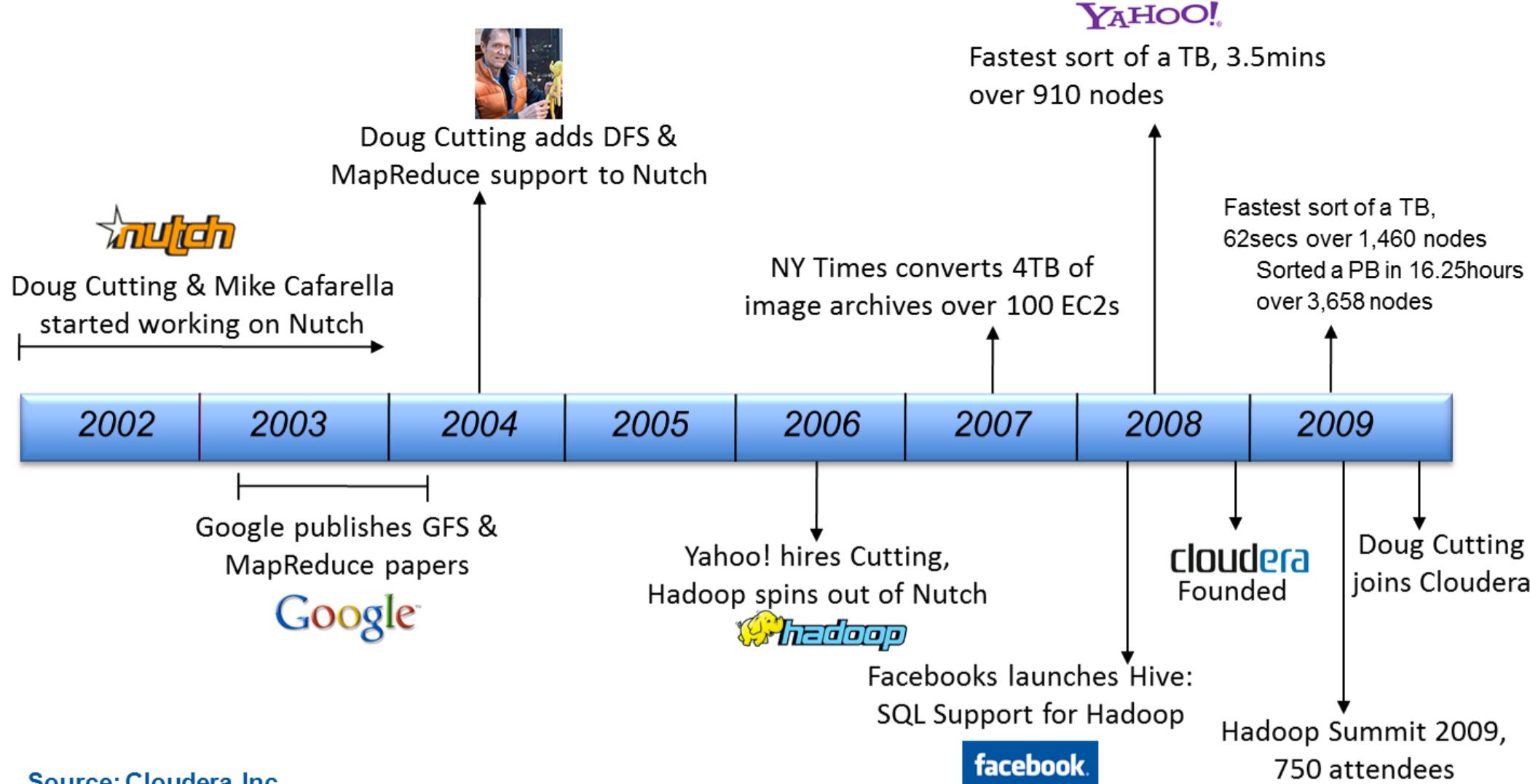
Hadoop History

- Created by Doug Cutting in 2005 at Yahoo
- Originally built as a Infrastructure for the “Nutch” project.
- Based on Google’s MapReduce and Google’s File System.
- Named after his son’s toy yellow elephant.



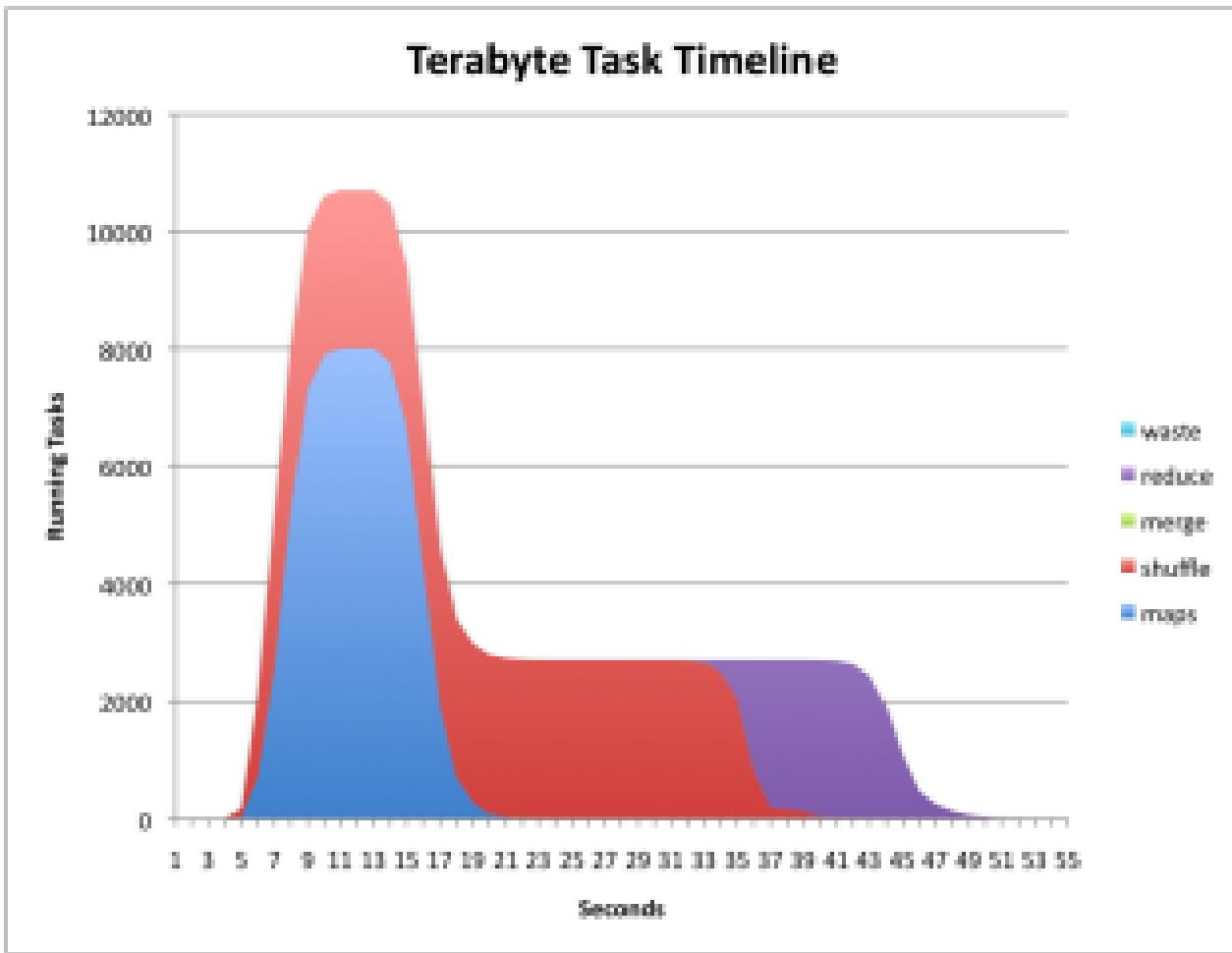
Hadoop = High-Availability Distributed Object-Oriented Platform

Hadoop History – 10+ years !

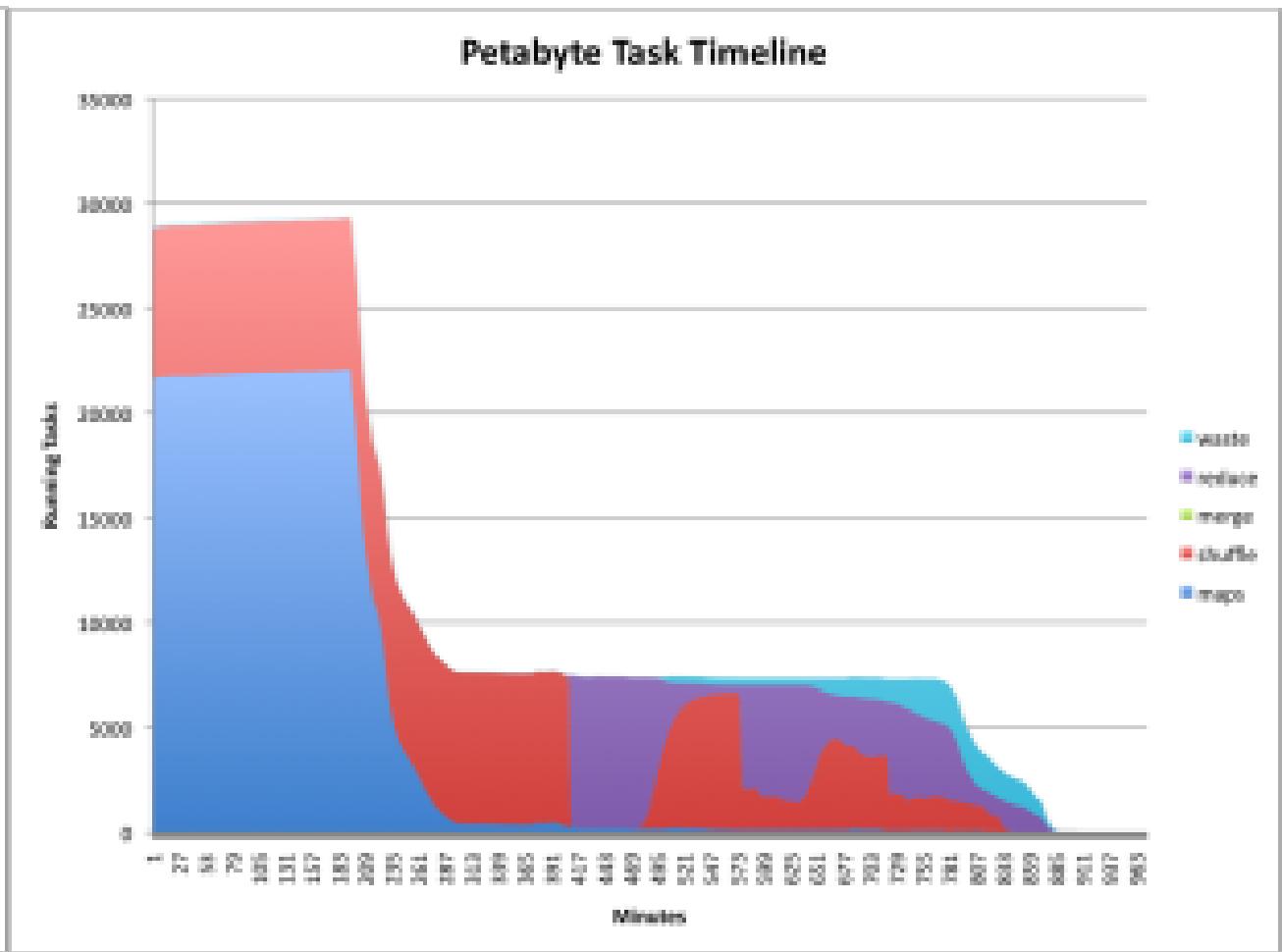


Hadoop Performances

1 Terabytes = 62 secondes



1 Petabytes = 16h 25 minutes



Hadoop v1.x Terminology



● **NameNode**

- Master of the system
- Maintains and manages the blocks which are present on the **DataNodes**

● **DataNodes**

- Slaves which are deployed on each machine and provide the actual storage
- Responsible for serving read and write requests for the clients

● **Secondary NameNode**

- Take checkpoints of the file system metadata present on **NameNode**. It is not a backup namenode

● **Jobtracker**

- Takes care of all the job scheduling and assign tasks to **TaskTrackers**.

● **TaskTrackers**

- One per node in the cluster – it accepts Map, Reduce tasks from the **Jobtracker**

Hadoop Core Components



Hadoop v1.x = **HDFS (Storage)** + **MapReduce (Processing)**



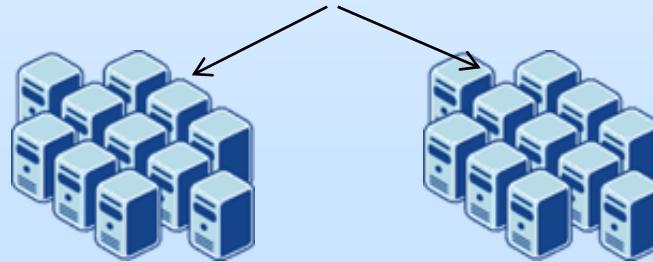
Running a Hadoop Job

- Place input files into HDFS
- Run MapReduce either normal or streaming* version

Hadoop Core Components



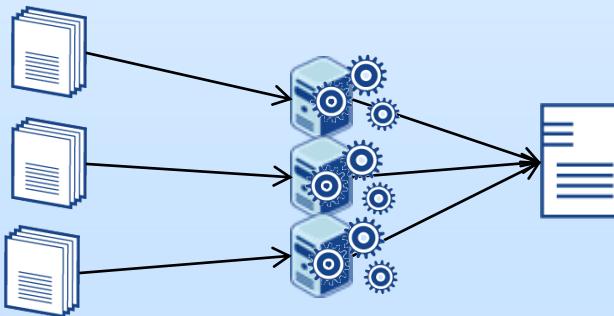
Storage



HDFS (*Hadoop Distributed File System*)

HDFS is a distributed file system. It is run on commodity hardware and it is highly fault tolerant. It was designed to reliably store very large files across machines in a large cluster.

Processing



MapReduce

MapReduce is a programming model for writing applications that can process Big Data in parallel on multiple nodes. MapReduce provides analytical capabilities for analyzing huge volumes of complex data.

Hadoop Features



Flexibility

Schema on read

Fault-Tolerant

HDFS: Data still accessible

MapReduce : Able to handle failures including crashes



Scalability

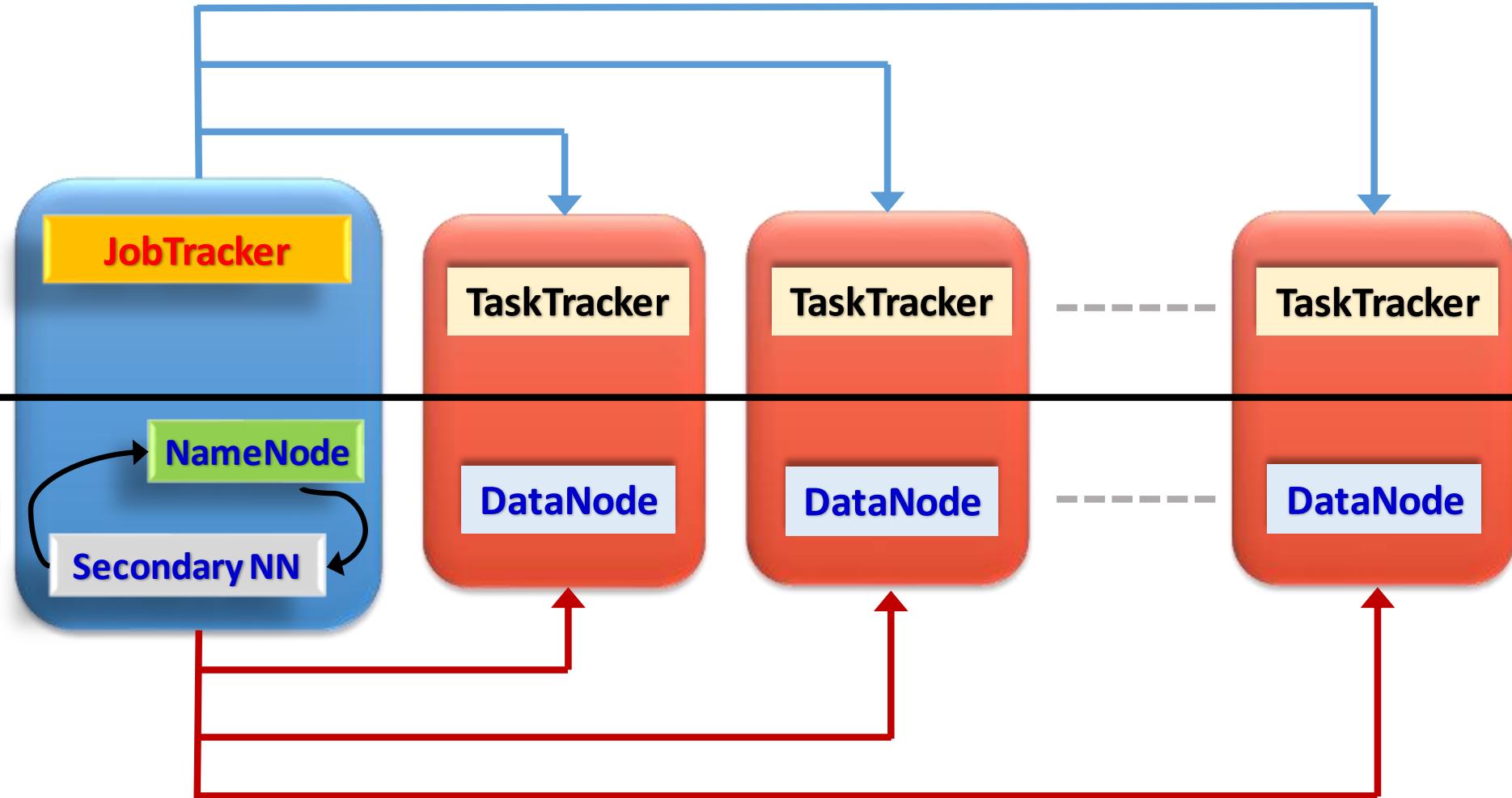
Scale-out easily

Economic

Runs on commodity hardware



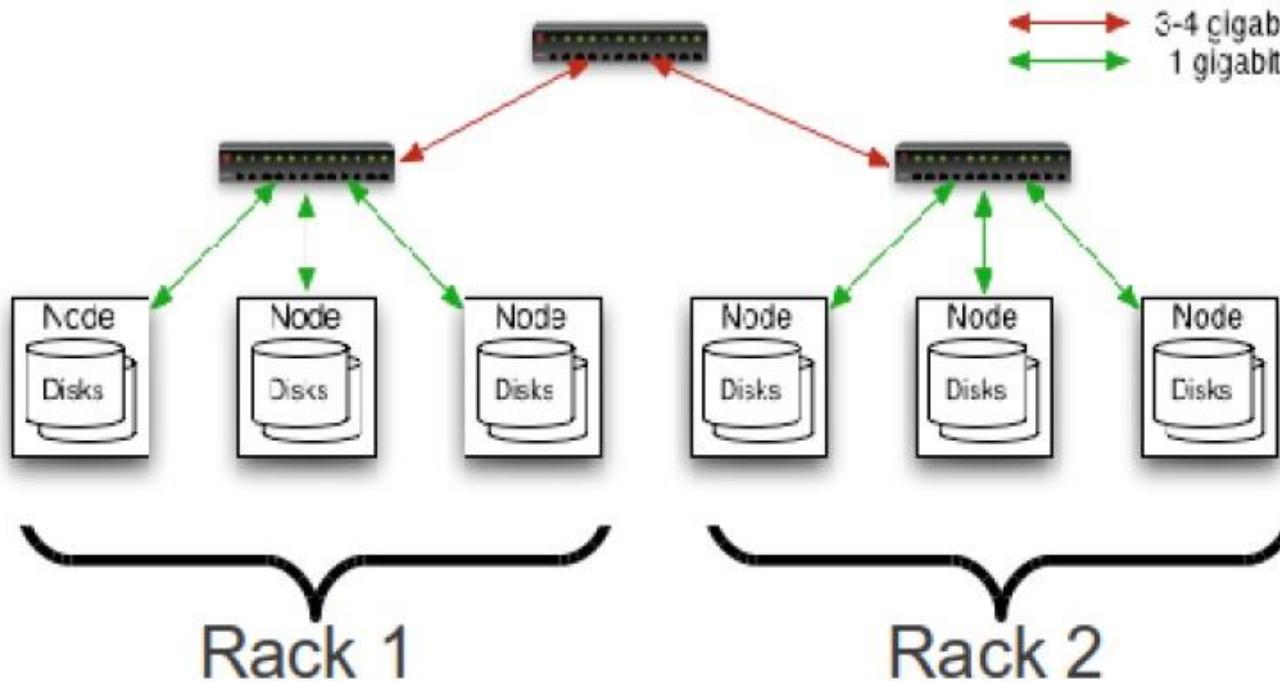
Hadoop : High Level Architecture



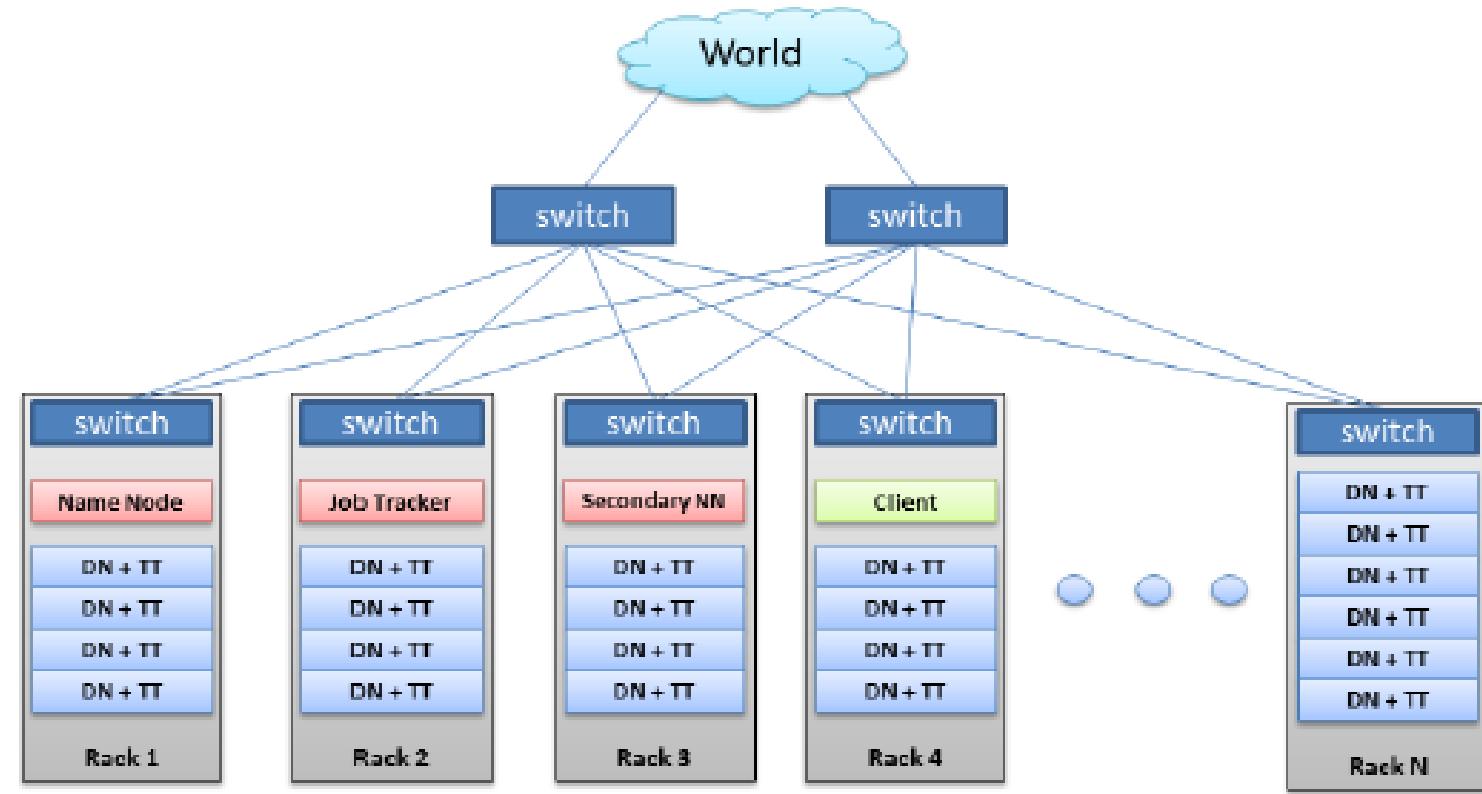
Hadoop Physical Architecture

- Client / Server based architecture
- Machines are grouped per *Rack*

(1 Rack \approx 1 -> 32 machines)



Hadoop Cluster Architecture



Yahoo Hadoop Cluster : 42,000+ Nodes



3 Nodes Hadoop Cluster (Commodity Hardware)

Hadoop Running Modes



Standalone (Local)

- Single machine
- All services run in the same JVM (*Java Virtual Machine*)
- No HDFS



Pseudo-distributed

- Single machine
- Each service runs in its own JVM
- HDFS is available

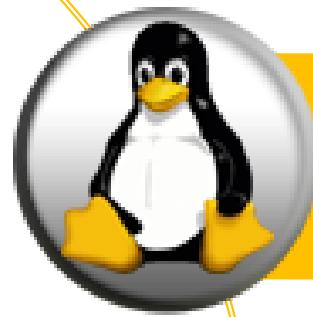


Fully-distributed

- Two machines at least
- Each service runs in its own JVM
- HDFS is available



Setting up Hadoop



Linux : Ubuntu, CentOS (recommended), ...
Windows : possible in native mode



Java JDK 1.6+



SSH : (Secure Shell)

You need a computer with Linux installed, internet connection and patience

Hadoop Installation Steps

- **STEP 1 – Create Separate Login/Group**
- **STEP 2 – Install JAVA JDK 1.6+**
- **STEP 3 – Install and configure SSH**
- **STEP 4 – Download, Unzip and Install Hadoop**
- **STEP 5 – Setting up Hadoop configuration files**
~/.bashrc, hadoop-env.sh, core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml
- **STEP 6 – Format the Hadoop file system (namenode)**
- **STEP 7 – Start Hadoop daemons**



Configuration Files settings example

conf/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

conf/core-site.xml:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

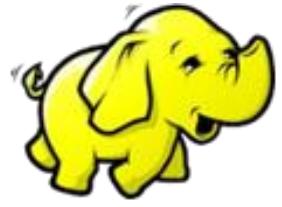


conf/mapred-site.xml

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
</configuration>
```

Configuration files are located in :
hadoop/conf/

Hadoop and Data Velocity



On-Demand Processing

Data at Rest

- Data is uploaded into Hadoop cluster and processed using the Hadoop ecosystem tools.
- Java is the primary language to write your code.



Real-Time / Near Real-Time

Data in Motion

- Data is processed in memory and it is very fast.
- Suitable to process data in motion (streaming) or at rest (batch) with low or high velocity.
- You can write your code using Java, Python, R and Scala.



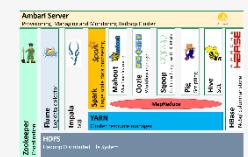
APACHE
STORM™

Real-Time

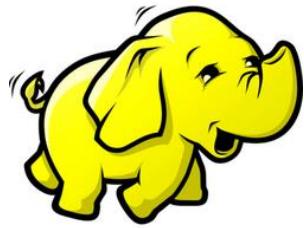
Data in Motion

- Data processing is done in real-time.
- Very flexible and scalable architecture.
- Suitable for high velocity data
- You need to code in Java to design your Storm topology.

Choose the right tool / ecosystem to process your data



The Hadoop Ecosystem



2006 - Hadoop

APACHE
HBASE



2007 - HBase



2009 - Hive / Pig

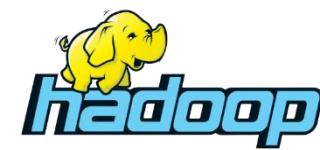


2011 - Storm



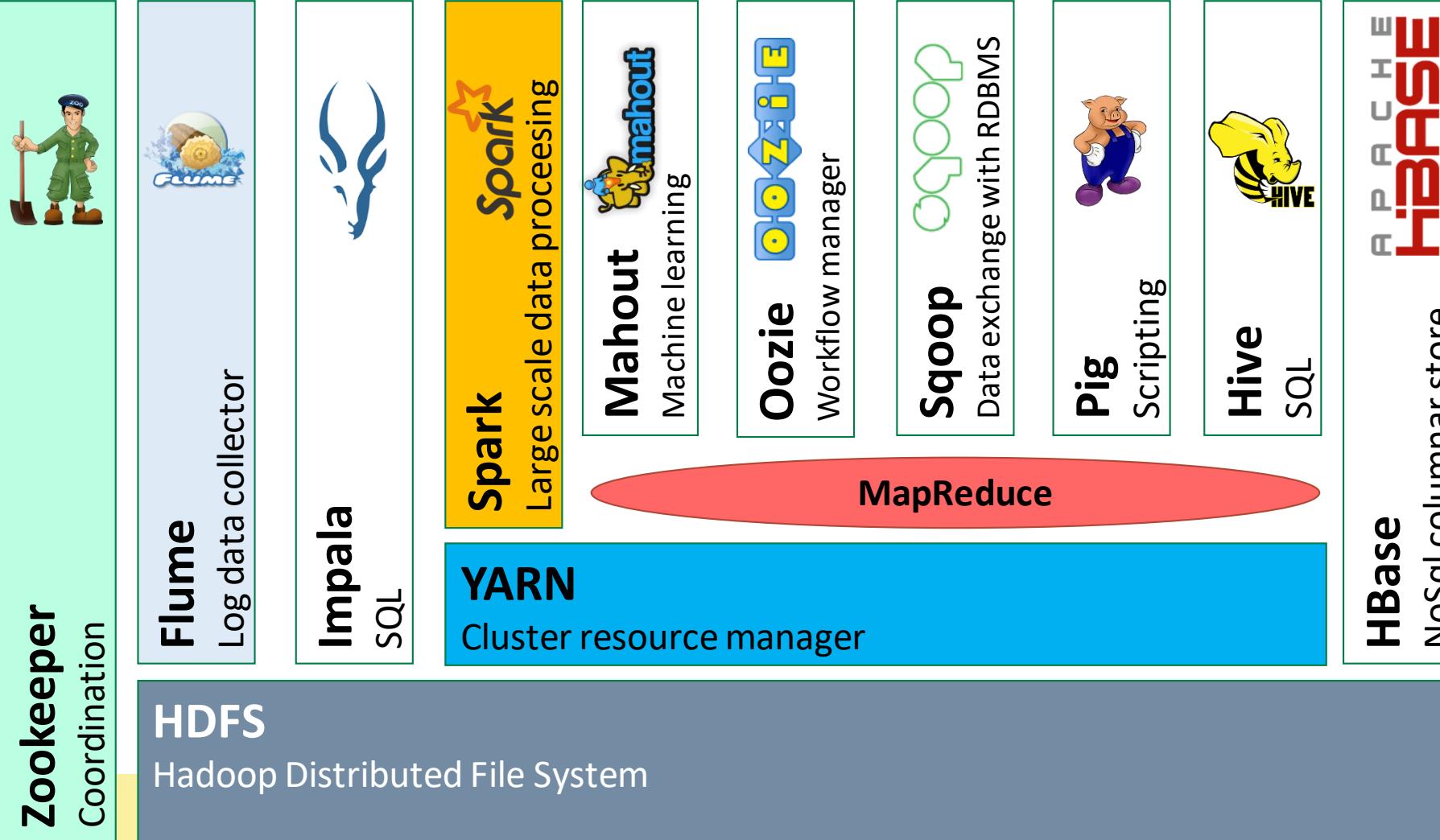
2012 - Spark / Impala

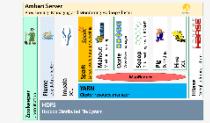
Hadoop Ecosystem - A Set of Components



Ambari Server

Provisioning, Managing and Monitoring Hadoop Cluster





Some Hadoop Ecosystem components



- **HBase**

Hadoop database for random read/write access



- **Hive**

SQL-like queries and tables on large datasets



- **Pig**

Data flow language and compiler



- **Oozie**

Workflow for interdependent Hadoop jobs



- **Sqoop**

Integration of databases and data warehouses with Hadoop



- **Flume**

Configurable streaming data collection



- **ZooKeeper**

Coordination service for distributed applications



Hadoop Yarn



Diving into the Hadoop core



McGill
FALL 2018

What is YARN?



- YARN (**Y**et **A**nother **R**esource **N**egotiator) has been introduced to Hadoop with version 2.0
- YARN aims to solve some issues with the resources scheduling of MapReduce in version 1.0
- Remedies the scalability shortcomings of “classic” MapReduce



Current Hadoop Map Reduce Limitations



Scalability

- Maximum Cluster Size – 4000 Nodes
- Maximum Concurrent Tasks – 40 000
- Coarse synchronization in Job Tracker
- One JobTracker per cluster

Single point of failure

- Failure kills all queued and running jobs
- Jobs need to be resubmitted by users

Resource management

- ‘slots’ can be assigned to map or reduce not both
- Underutilization of cluster when more map or reduce tasks are running
- Can’t share resources with non MR applications

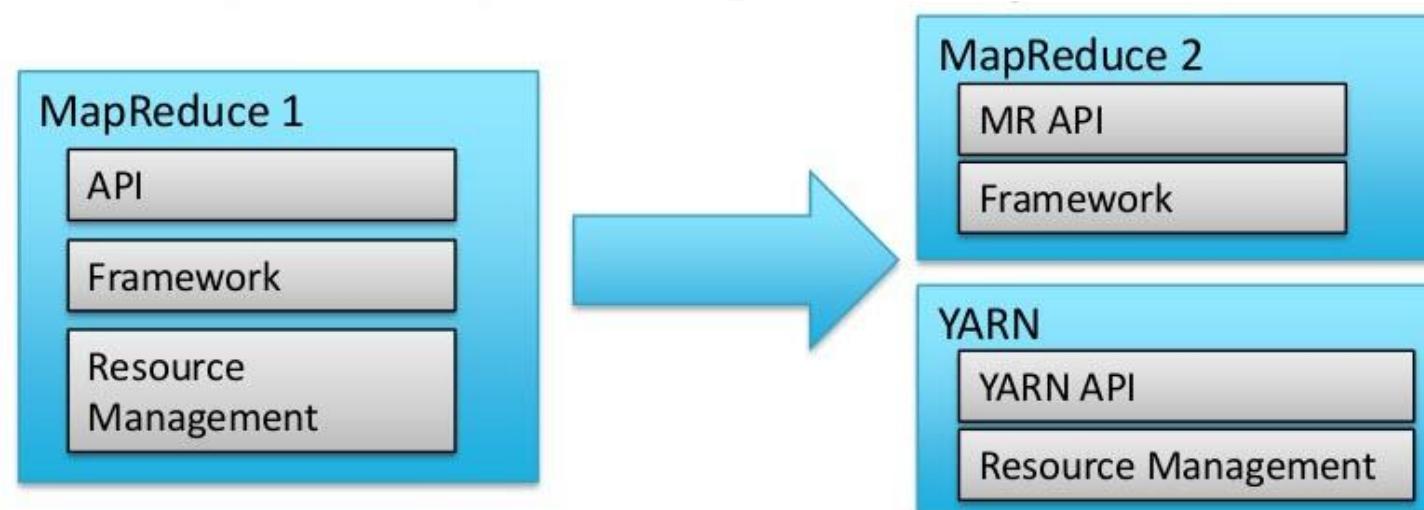
MapReduce v1 and MapReduce v2



- MapReduce v1 has three main components

- MR Core Framework and runtime services
- Java API for user level programming MR application
- Resource management – monitoring nodes, allocating resources and scheduling jobs

- MapReduce v2 moves Resource management into YARN

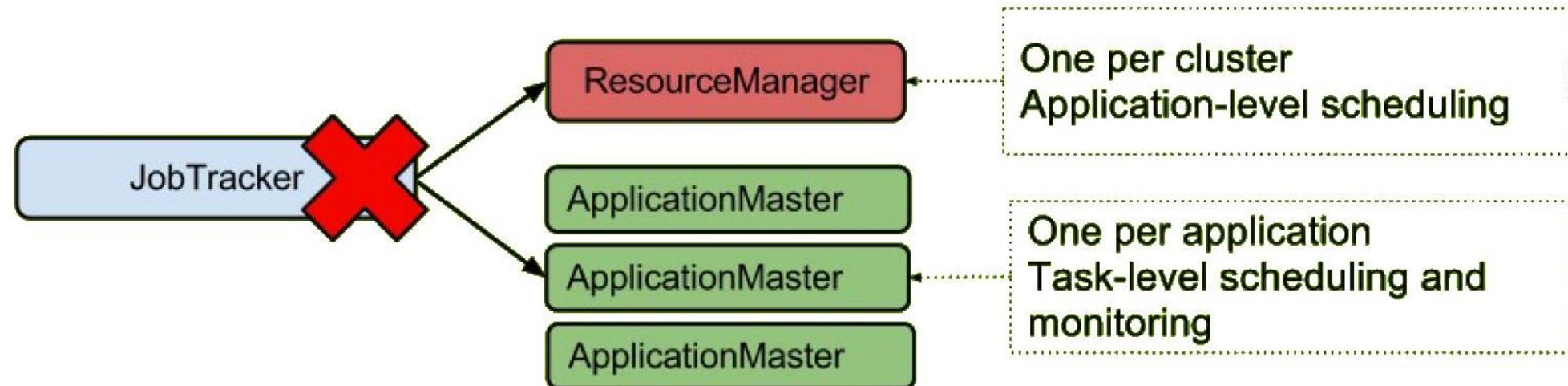


YARN Solution



- No more ‘slots’ – resources (CPU, Memory) are allocated when requested
- Support MapReduce and non MapReduce applications running on the same cluster
- Moves the two major functions of JobTracker to **ApplicationMaster and ResourceManager**

One cluster can have many ApplicationMaster(s)

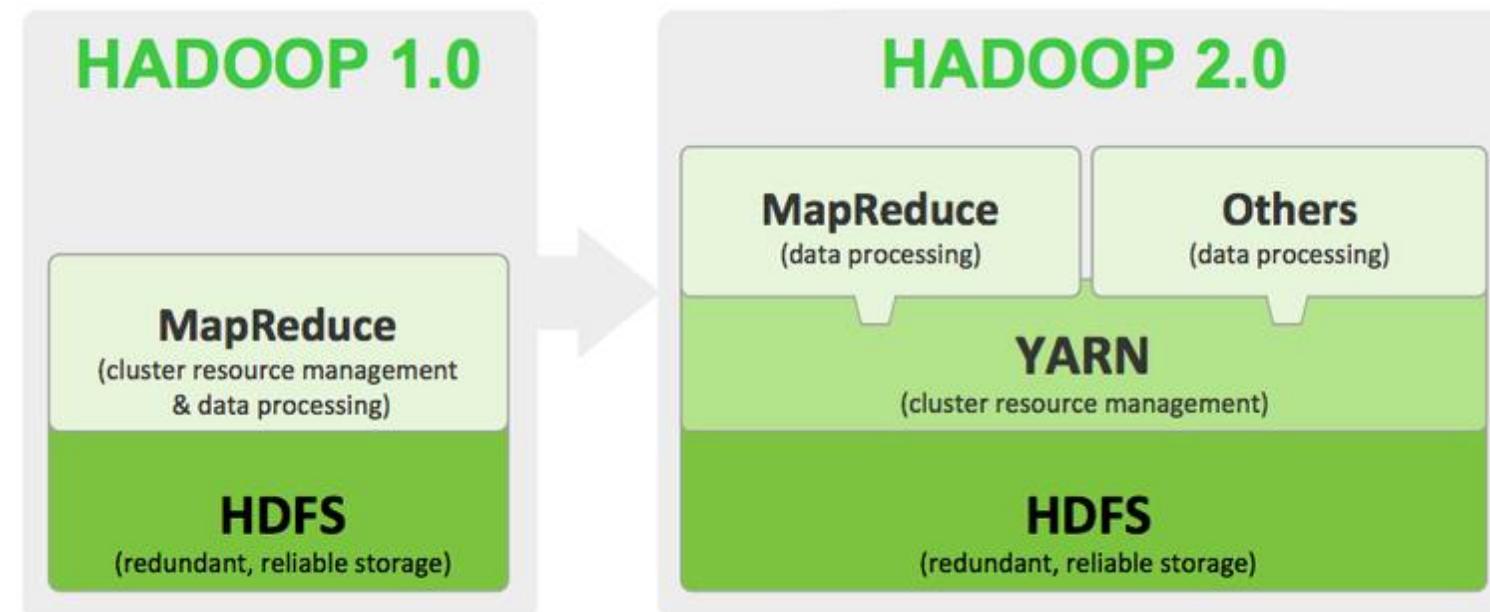
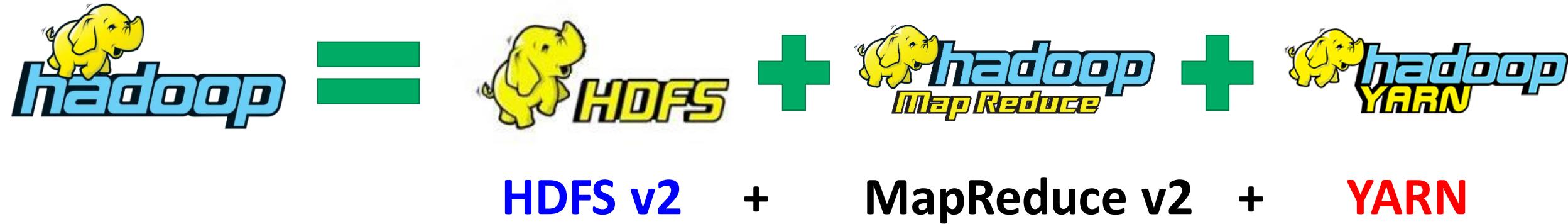


YARN Solution

- Tasktracker moves to NodeManager (NM)
- NodeManager is a new per-node slave
- NodeManager is responsible for
 - Launching the applications' containers
 - Monitoring their resource usage (cpu, memory, disk, network)
 - Reporting to the Resource Manager
- YARN maintains compatibility with existing MapReduce applications and users



Hadoop v2.x High Level Architecture



YARN main components



Resource Manager (RM)

- One per cluster
- Central, global view
- Runs on master node
- Global resource scheduler
- Tracks hearbeats from NodeManagers
- Handles Application Master requests for resources
 - *Job requests are submitted to RM*
 - *To start a job, RM finds a container to spawn AM*
- Manage security (supports Kerberos)



YARN main components



Node Manager (NM)

- Runs on slave nodes
- One per node
- Launch application processes on request
- Report resources (memory, CPU, etc...) to RM

Send heartbeats and containers status

- Configure the environment for task execution
- Garbage collection / Authentication
- Auxiliary services

Output intermediate data between map and reduce tasks

Node Manager



Node Manager



YARN main components



Application Master (AM)

- The head of a job
- One per application
- Runs as a container
- Requests resources from the RM to run application tasks
- Dynamically changing resource consumption
- Can run any user code (Dryad, MapReduce, Tez, REEF...etc)
- Determines the semantics of the success or failure of the container

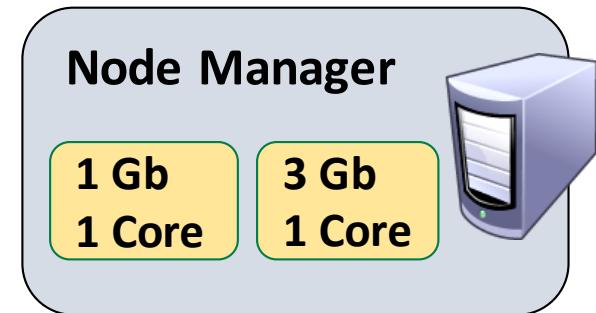


YARN main components

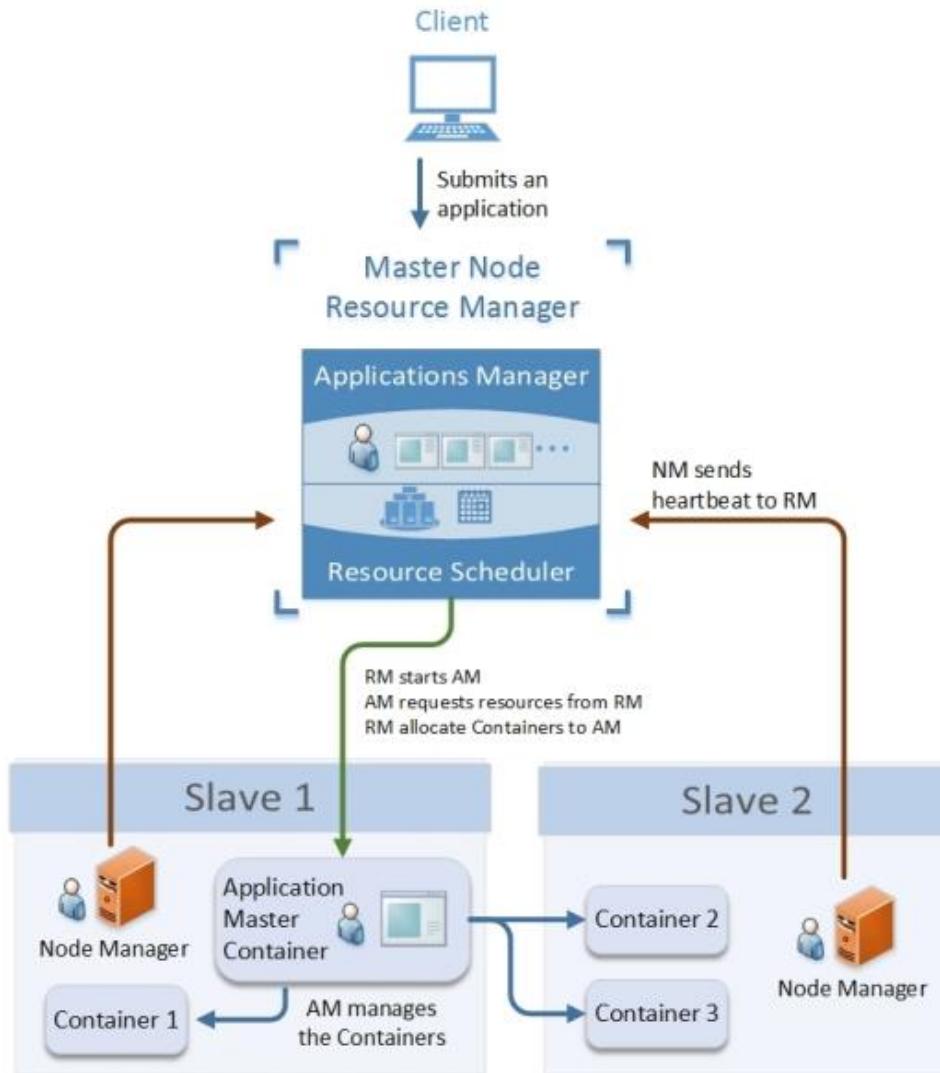


Containers

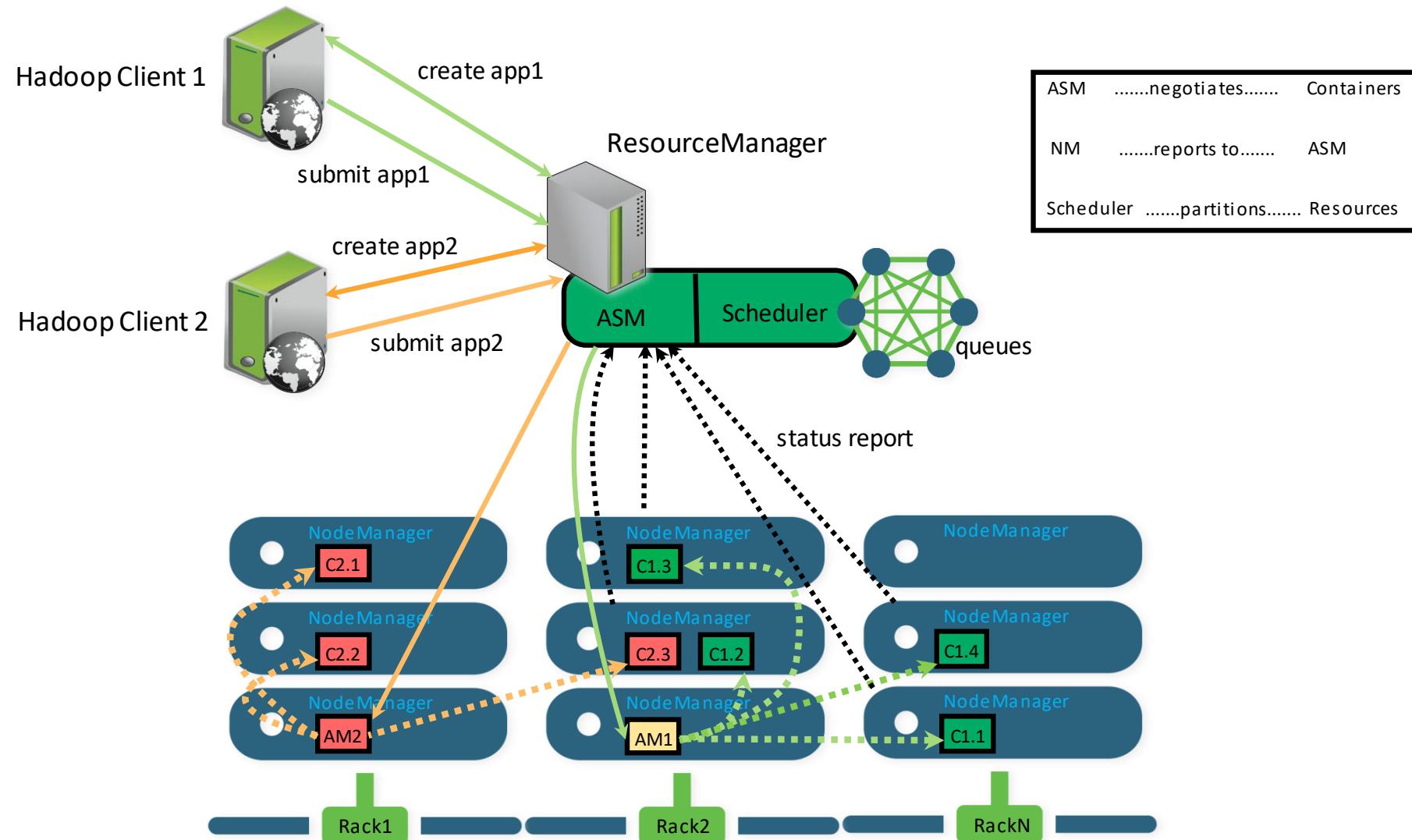
- Logical bundle of resources (CPU/memory)
- Created by the RM upon request
- Allocate a certain amount of resources (CPU, Memory) on slave node
- Applications run in one or more containers



YARN: High Level Architecture



YARN Cluster : Running an Application



Fault Tolerance

Server Failure

- NM heartbeats to RM

If server fails, RM lets all affected AMs know, and AMs take action

- NM keeps track of each task running at its server

If task fails while in-progress, mark the task as idle and restart it

- AM heartbeats to RM

On failure, RM restarts AM, which then syncs up with its running tasks

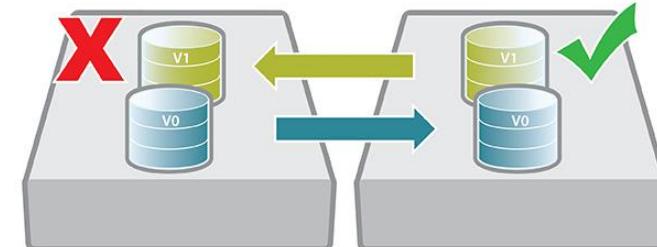


RM Failure

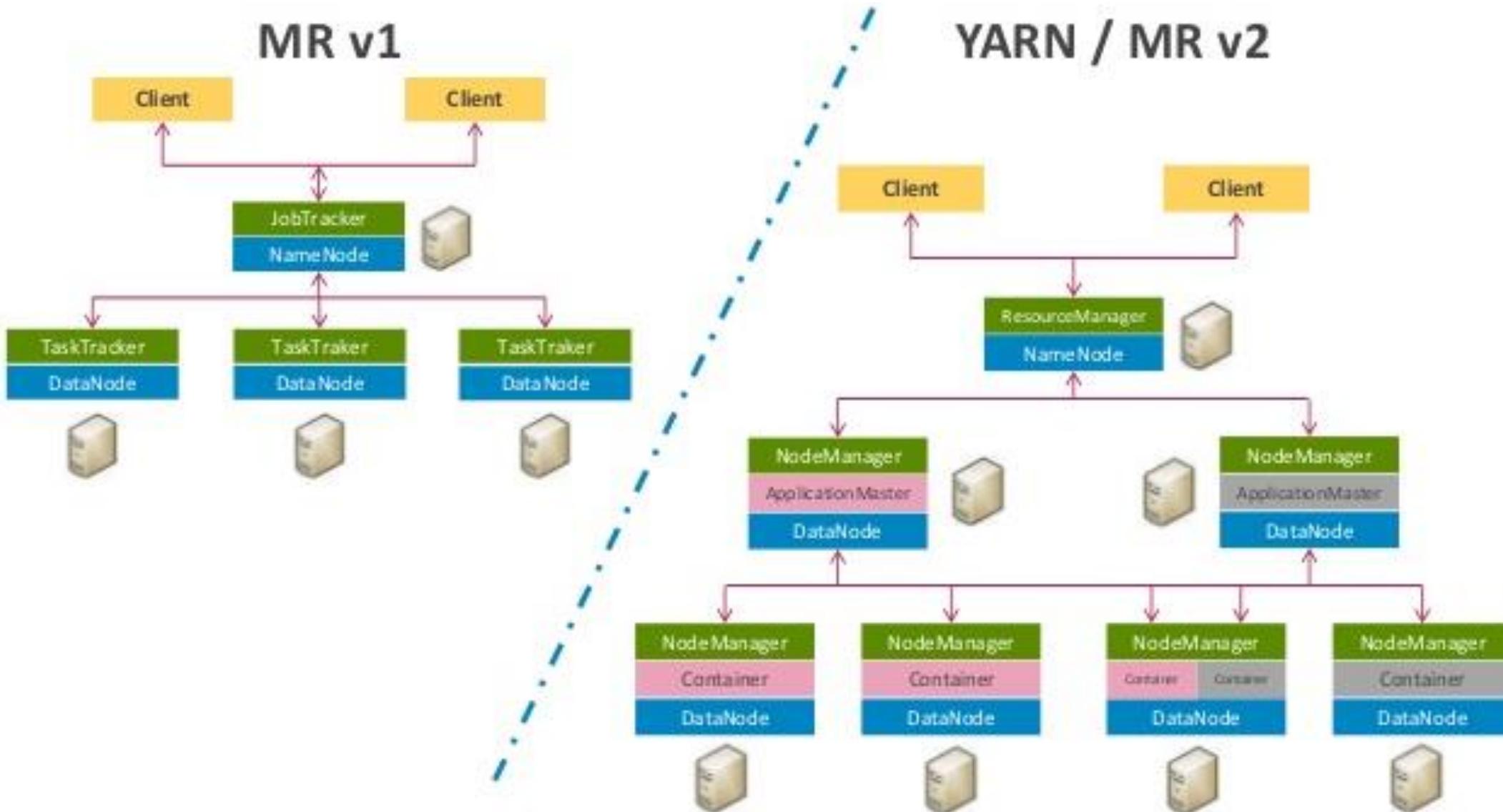
- Use old checkpoints and bring up secondary RM

Heartbeats also used to piggyback container requests

- Avoids extra messages



Hadoop v1 vs Hadoop v2



Benefits of HDFS Federation



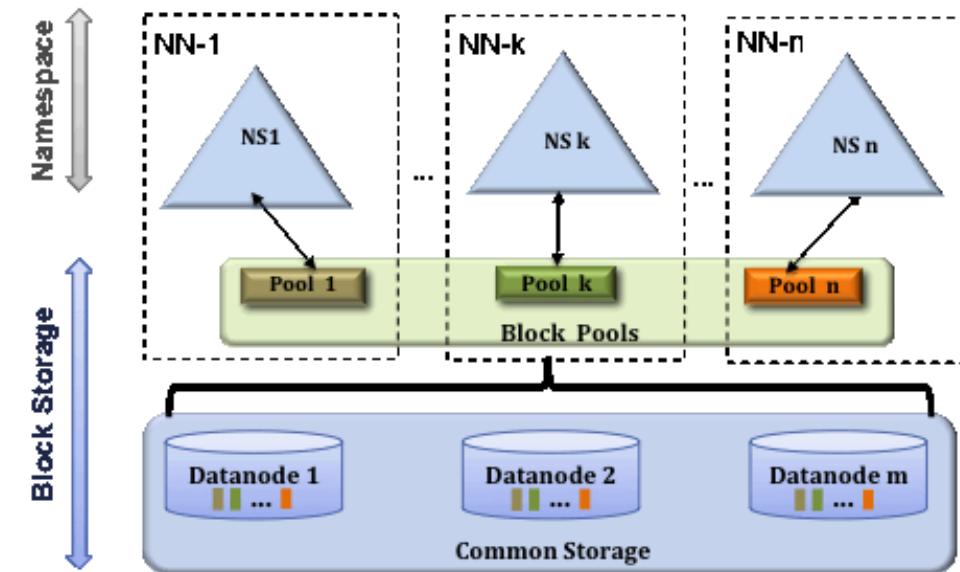
- Here are some benefits of HDFS Federation in the HDFS v2.x

- Logically Separate Namespaces
- Ease NameNode Bottlenecks

Having all data index through a single NameNode can be eliminated by creating multiple NameNodes.

- Options for Tiering Performance

Segmenting different NameNodes and namespaces by customer requirements instead of setting up multiple complicated performance quotas is now an option



Start / Stop YARN

To Start YARN

- From the `hadoop/sbin` folder use the commande `start-yarn.sh`
- This command starts YARN Resource Manager



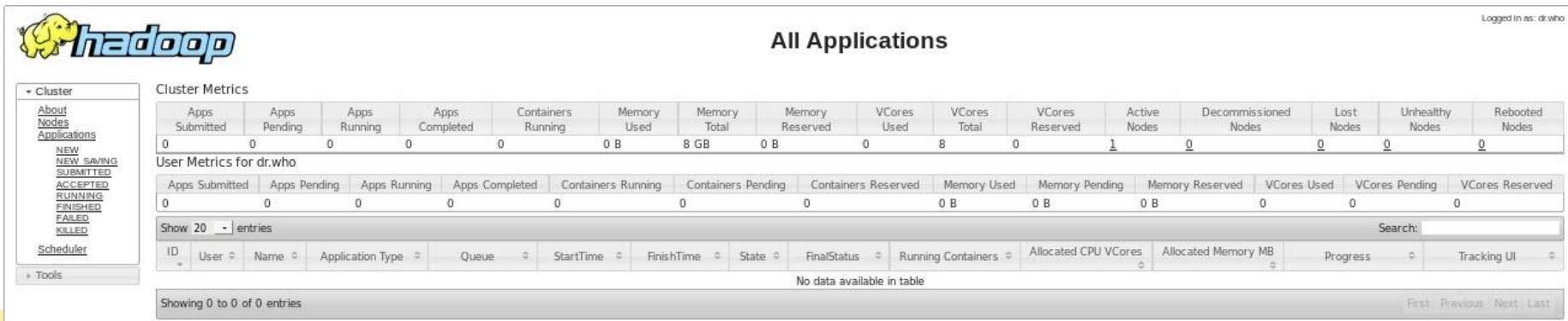
To Stop YARN

- From the `hadoop/sbin` folder use the commande `stop-yarn.sh`
- This command stops YARN Resource Manager



ResourceManager web interface:

<http://localhost:8088>



The screenshot shows the ResourceManager's web interface at <http://localhost:8088>. The title bar says "All Applications". The left sidebar has a "Cluster Metrics" section with tabs for "About", "Nodes", "Applications", "Scheduler", and "Tools". The "Applications" tab is selected, showing a table with columns: 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, and Rebooted Nodes. Below this is a "User Metrics for dr.who" table with similar columns. At the bottom, there is a search bar and a table header for "Applications" with columns: ID, User, Name, Application Type, Queue, StartTime, FinishTime, State, FinalStatus, Running Containers, Allocated CPU VCores, Allocated Memory MB, Progress, and Tracking UI.

Run a Jar file

- To run a jar file use this command

yarn jar examples.jar myjob [...]

yarn command

archive name

application name

application arguments

```
[cloudera@quickstart ~]$ yarn
Usage: yarn [--config confdir] COMMAND
where COMMAND is one of:
  resourcemanager -format-state-store    deletes the RMStateStore
  resourcemanager                         run the ResourceManager
                                         Use -format-state-store for deleting the RMStateStore.
                                         Use -remove-application-from-state-store <appId> for
                                         removing application from RMStateStore.

  nodemanager
  timelinerunner
  radmin
  version
  jar <jar>
  application

  applicationattempt
  container
  node
  queue
  logs
  classpath

  daemonlog

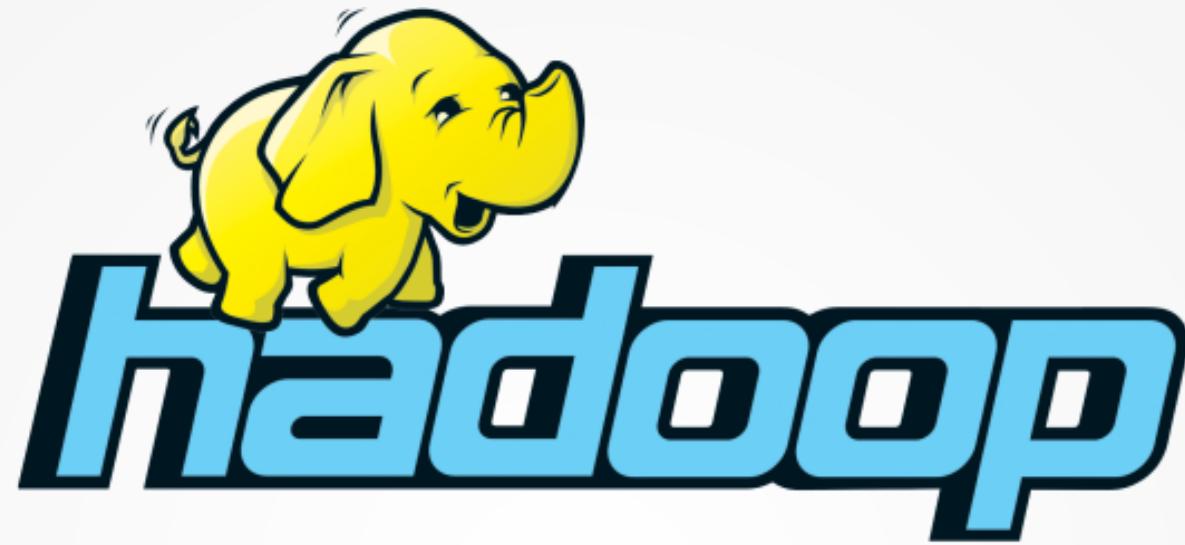
  top
  or
  CLASSNAME                         run the class named CLASSNAME

Most commands print help when invoked w/o parameters.
[cloudera@quickstart ~]$
```

It's time for a break

Grab some coffee, We'll be back in 15min





Hadoop Distributions

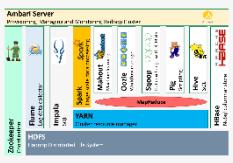


A panorama of top Hadoop vendors



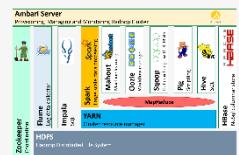
McGill
FALL 2018

Introduction to Hadoop Distributions



- Installing Hadoop is not for the faint heart. It needs to be patient and very rigorous
 - Installing on a large number of machines requires a lot of effort
 - Configuring each tool and take care of different versions might be very complex
 - Needs to track all configuration settings and parameters

What is the solution?



1

Automate installations using scripts and third-party tools



2

Use a commercial solution or a “Distribution”

cloudera®

Hortonworks

IBM BigInsights



Hortonworks Powers
Microsoft HDInsight

MAPR®

amazon
EMR

Pivotal Big Data Suite



DATASTAX®

Vendors Distributions Approach



- Different approach to cluster management
- Different level of “Open Source” – vendor lock-in
- Different approach to near to real-time analytics
- Proprietary components

cloudera®



Hortonworks



MAPR®



IBM BigInsights



**amazon
EMR**

The Case for Hadoop Distributions



- **Vendors provide easy to install mediums (Linux RPMs)**
packages make it easy for system administrators to manage effectively
- **Vendors package multiple components that work well together**
The Hadoop ecosystem contains a lot of components. Vendors bundle versions of components that work well together (HBase, Pig, Hive, Zookeeper, etc.).
- **Tested**
Vendors strive to ensure good quality components.
- **Performance patches**
- **Predictable upgrade path**
- **And most importantly . . SUPPORT !!**

Vendors Distributions partnerships

Most popular vendors list tends to shrink to Cloudera, Hortonworks, and MapR

- Intel ditched its Hadoop distribution and backed Cloudera in 2014

<https://venturebeat.com/2014/03/26/intel-cloudera-hadoop/>

- Pivotal switched to resell Hortonworks Data Platform (HDP)

<https://www.zdnet.com/article/hortonworks-announces-new-alliances-and-releases-hadoop-comes-to-fork-in-road/>

- IBM announced that it would stop shipping its own Hadoop distribution and partner with Hortonworks instead

<http://blogs.gartner.com/merv-adrian/2017/06/21/ibm-ends-hadoop-distribution-hortonworks-expands-hybrid-open-source/>

Type of Commercial Hadoop Distribution

Commercial Hadoop distributions are majorly categorized in three primary kinds:

- 1. Distributions that provide paid support and training**
- 2. Distributions that offer a set of supporting tools for deployment and management**
- 3. Distributions that enable adding vendor specific features and code, paid enhancements**

Hadoop Distributions / Big Data Suites



Apache Hadoop



Less

Hadoop Distributions



- Test & Packaging
- Installation
- Monitoring
- Business Support

Hadoop Ecosystem

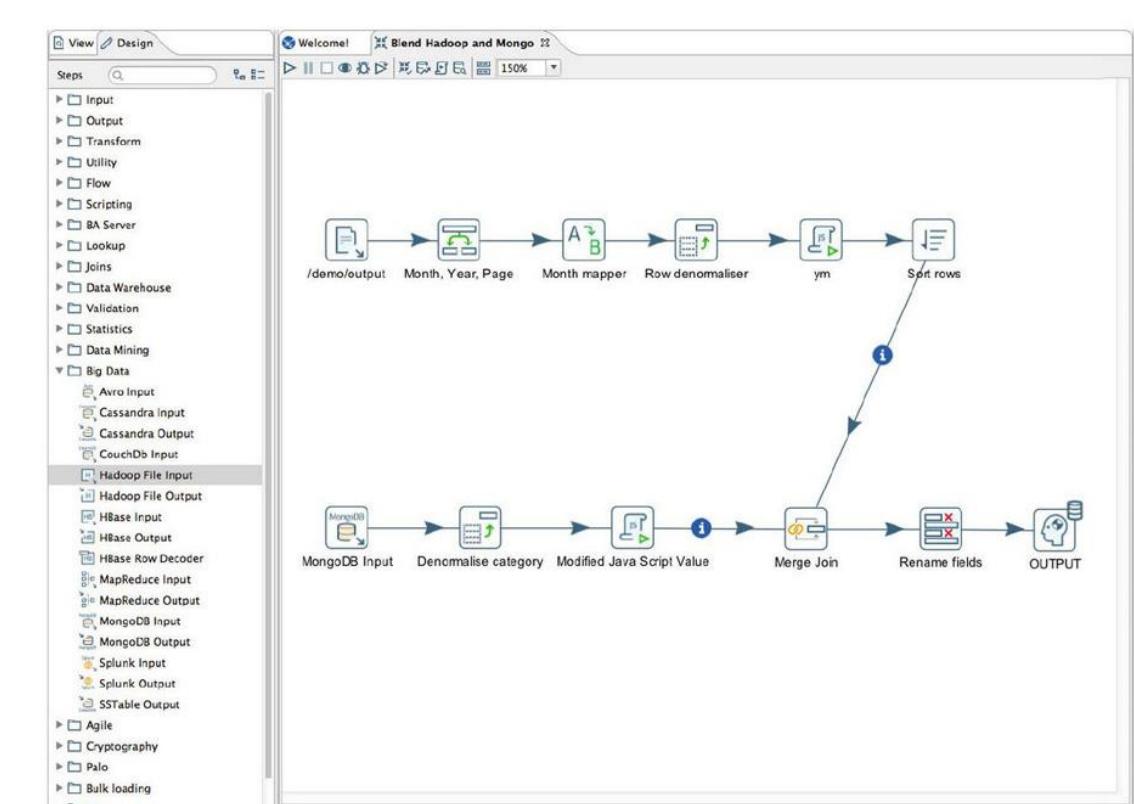
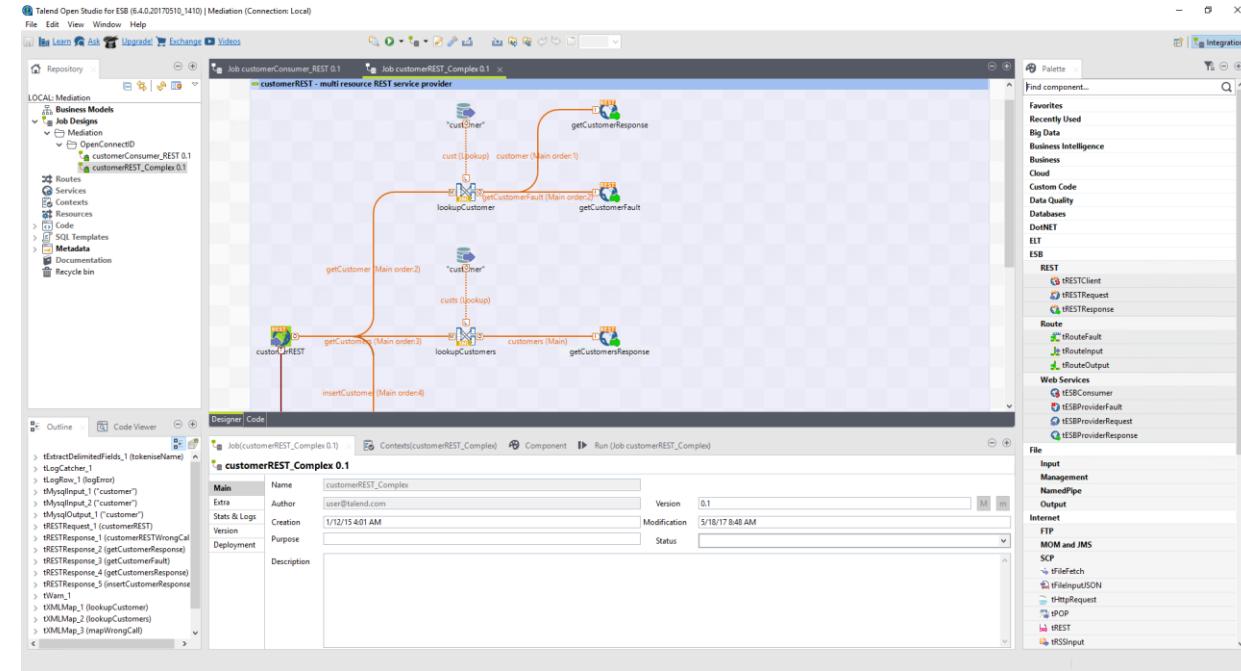
Big Data Suites



- Integrated Environment
- Visualization
- (Near) Real-time analytics
- Modeling
- ETL & Connectors

More +

Big Data Suites examples



 talend

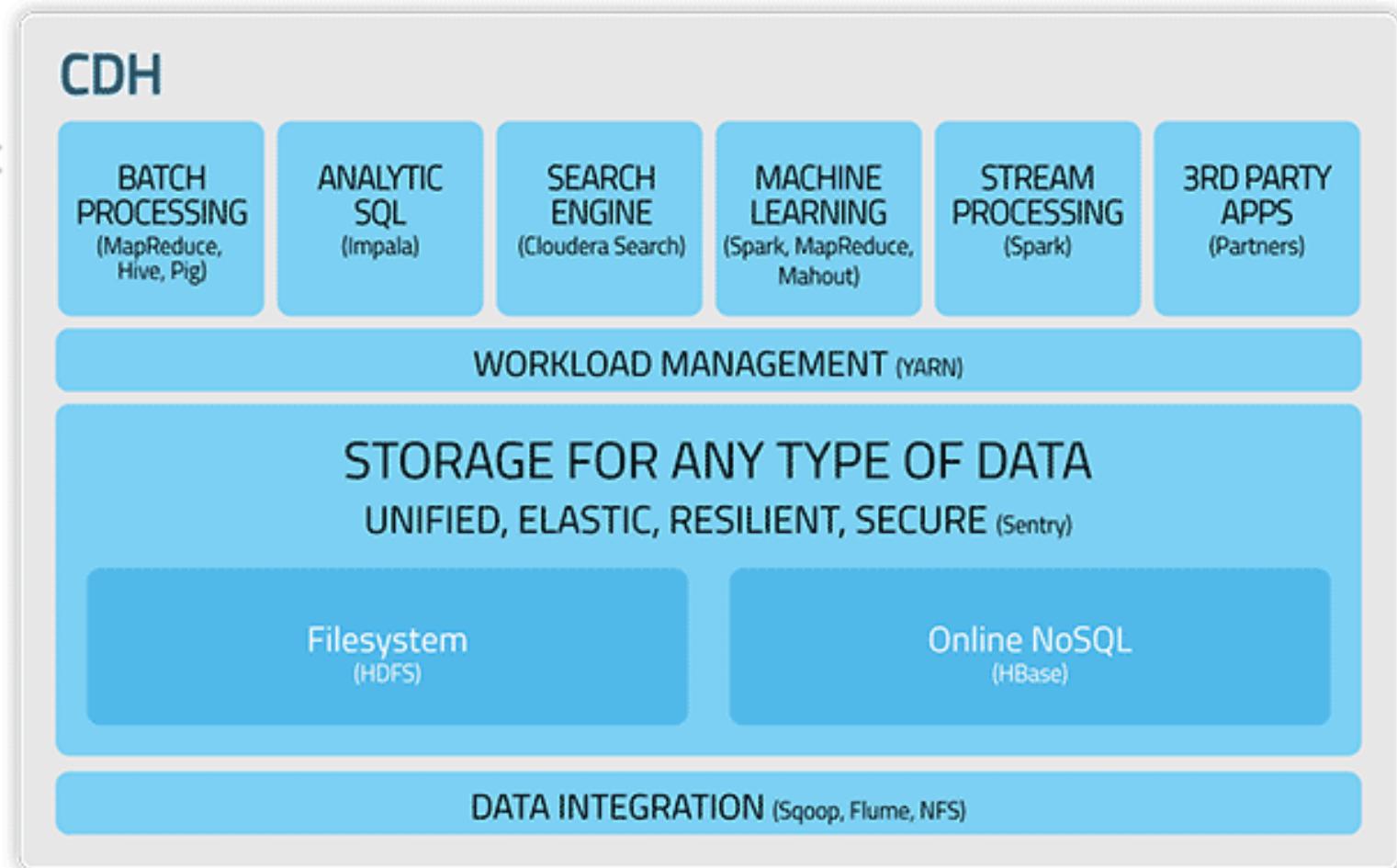
 pentaho®

Cloudera Distribution including Hadoop (CDH)

- Founded by Intel in 2008
- 1st Hadoop distribution editor
- Doug Cutting is the chief architect
- Hadoop vendors leader

- Cloudera Enterprise
- Support / Premium Services
- **Not full open-source**

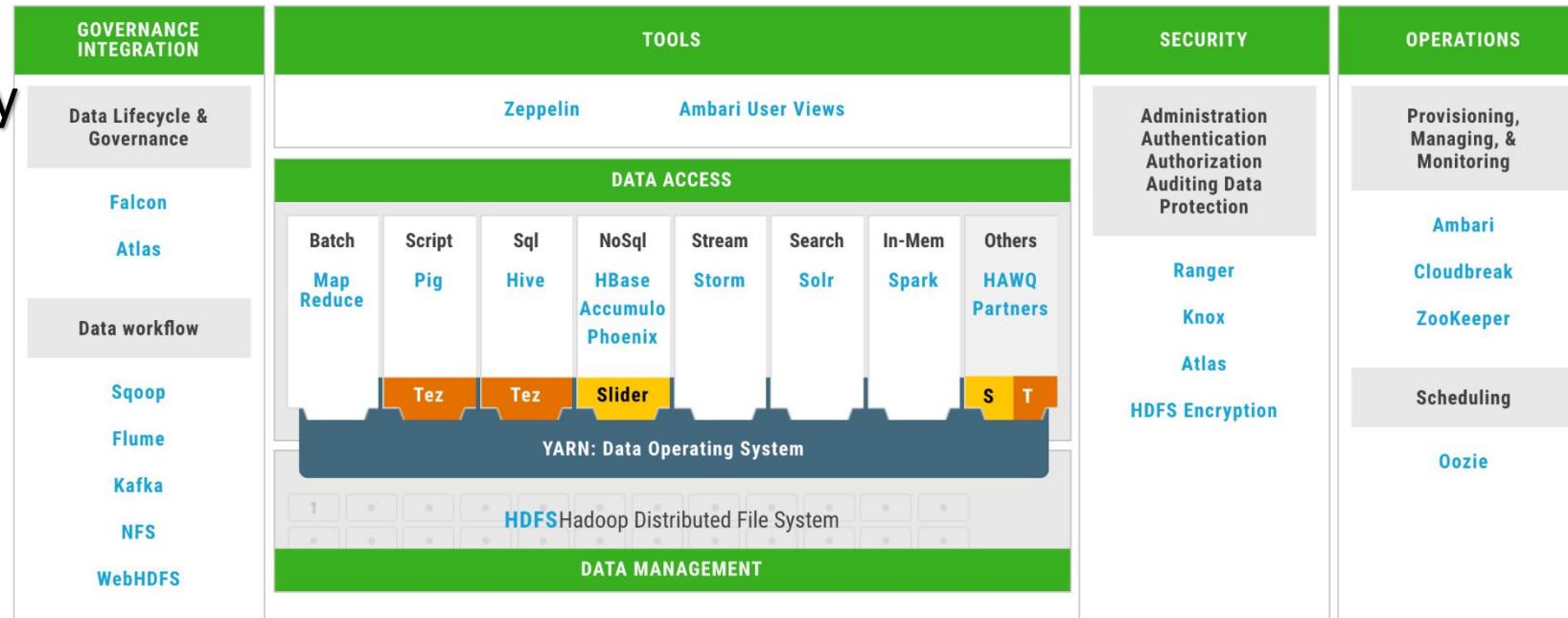
- Hadoop Projects:
 - Hadoop Common (Utilities)
 - Hue (SDK IHM)
 - Impala (real-time queries)
 - Whirr (SAAS Hadoop)
 - Sentry (Security)
 - Cloudera Manager (*Cluster management, provisioning, monitoring..*) (and others)



cloudera®

Hortonworks Data Platform (HDP)

- Founded by Yahoo in 2011
- Largest open source community
- Largest Hadoop committers
- **Full open-source**
- Support / Premium Services
- Hadoop projects:
 - YARN
 - HCatalog
 - Ambari Server (*Cluster management, provisioning, monitoring..*)
 - ... (*and many others*)



Pure play Hadoop company that drives open source Hadoop distributions

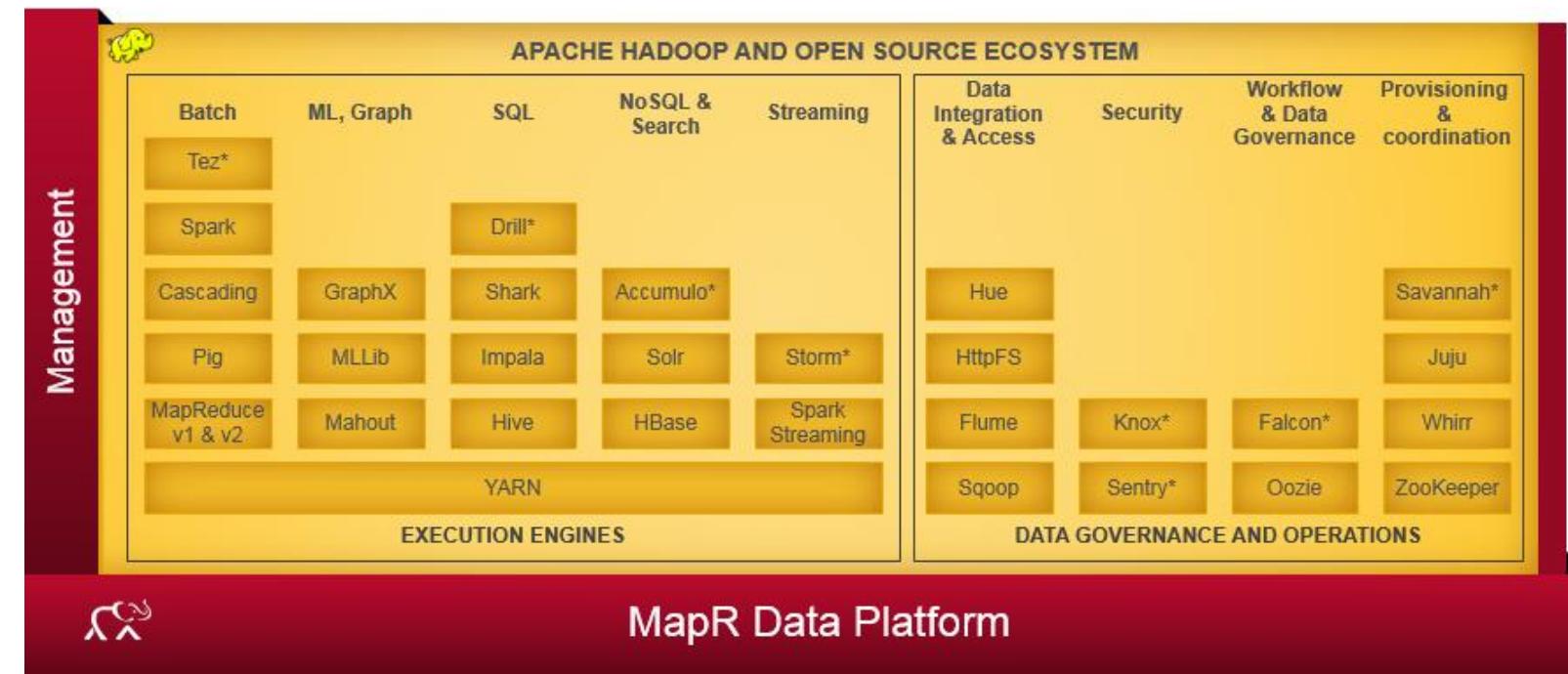


MapR Data Platform

- Founded in 2009
- Full data protection, no Spof
- Easy to use
- Dedicated file system
- **Not Full open-source**

- Support / Premium Services

- Hadoop projects:
 - Cascading (Java dev factory)
 - Vaidya (Perf, Benchmark)
 - Drill (Real-time queries)
 - HBase, Pig, Hive, Mahout
 - Sqoop, Flume, ...



MapR Data Platform

- MapR Distributions:
 - M3 - Free
 - M5
 - M7

MAPR

Amazon Elastic MapReduce (EMR)

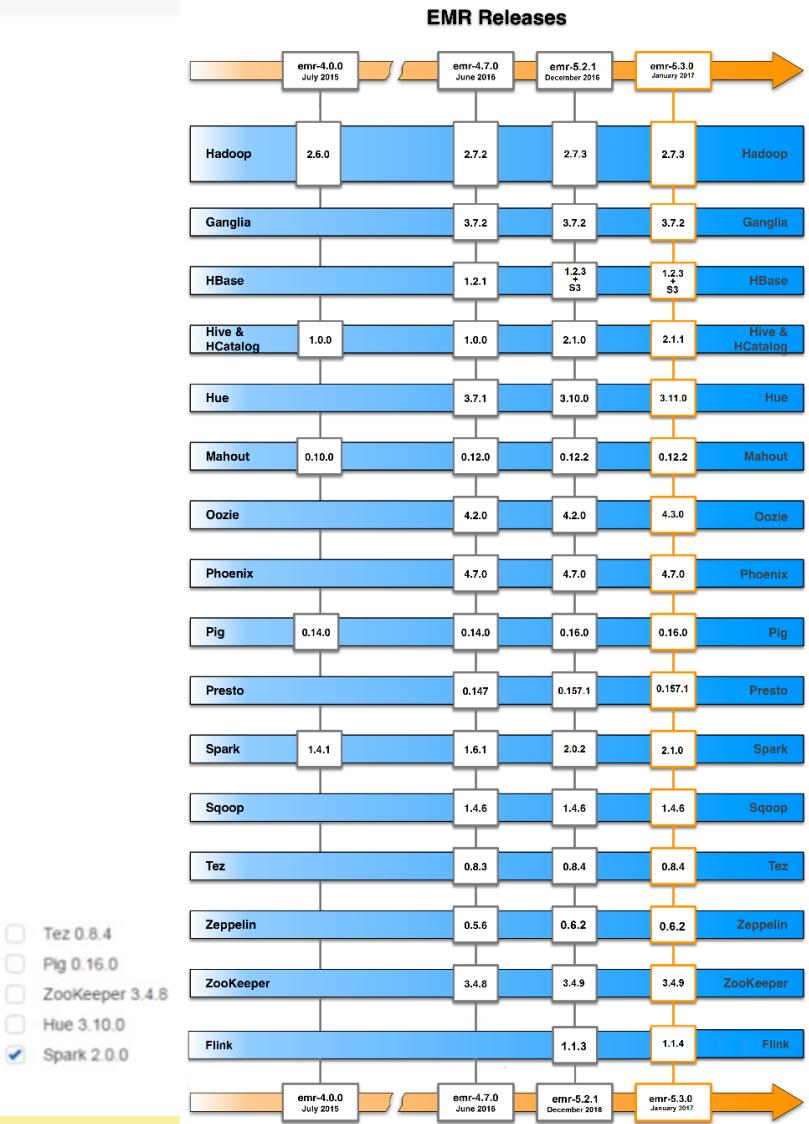
- First commercial offer on Amazon Web Services (AWS)
- Hadoop in the Cloud
 - Amazon EC2 (*Elastic Compute Cloud*) for processing
 - Amazon S3 (*Simple Storage Service*) for storage
 - Amazon Kinesis (data streaming)
- Good integration with Amazon Redshift
- Auto scaling
- Support many NoSQL databases and BI tools
- Low cost



Hadoop 2.7.2
 Ganglia 3.7.2
 Hive 2.1.0
 Sqoop 1.4.6
 Phoenix 4.7.0
 HCatalog 2.1.0

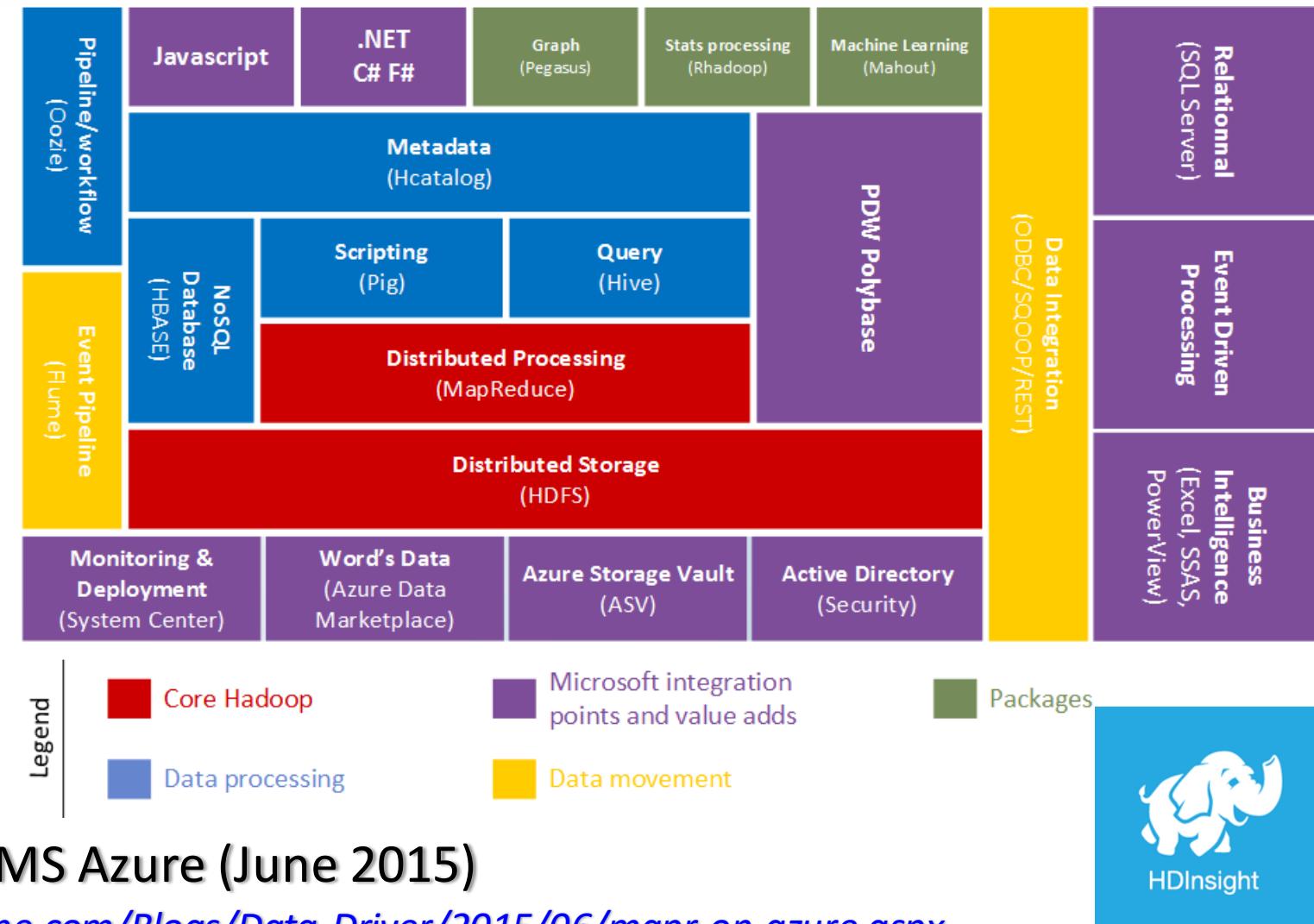
Zeppelin 0.6.1
 HBase 1.2.2
 Presto 0.150
 Mahout 0.12.2
 Hue 3.10.0
 Oozie 4.2.0

EMR 5.0 - Applications



Microsoft Big Data Solution (HDInsight)

- Based on a partnership with Hortonworks
- Runs on Windows Azure
- The only Hadoop distribution that runs on Windows platform
- Use MS Azure ‘Blob Storage’
- Designed to operate with MS Windows Server, MS SQL Server, Polybase and PowerPivot

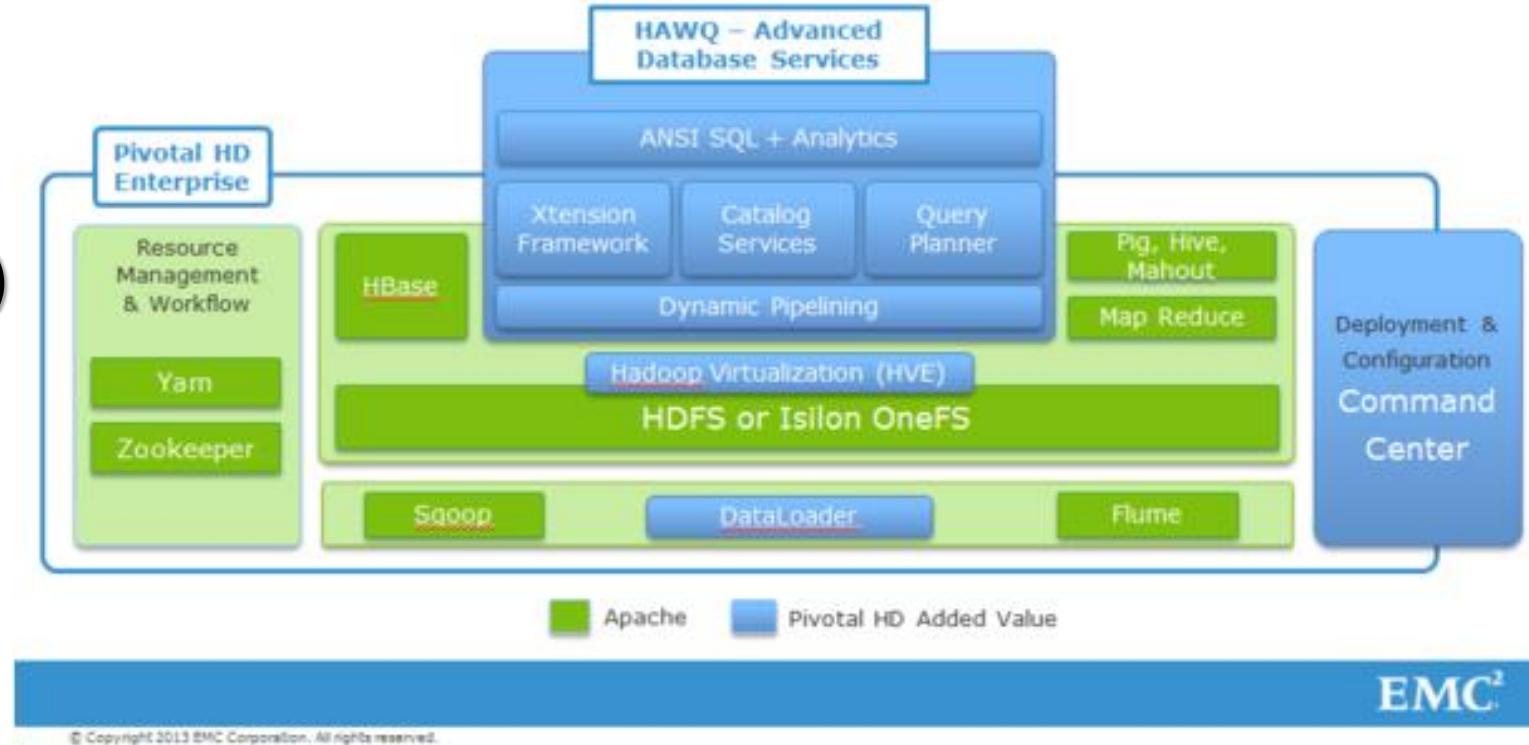


- MapR Hadoop is also available on MS Azure (June 2015)

<https://visualstudiomagazine.com/Blogs/Data-Driver/2015/06/mapr-on-azure.aspx>

Pivotal Big Data Suite

- Founded by EMC and Vmware
- Greenplum Technology (Database)

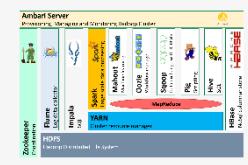


- Hadoop projects:
 - HAWQ - SQL on Hadoop using MPP engine



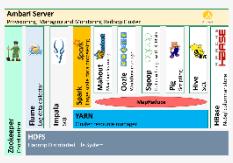
Pivotal **Big Data Suite**

Choosing the right distribution CheckList



- You need to close the gap between what Hadoop do generally and what it can specifically for your business (here some questions to answer)
 1. *What business problems do you want to solve?*
 2. *What data do you want to bring into Hadoop? data format, volume and velocity?*
 3. *What data will be analyzed and what methods of analysis do you plan to use?*
 4. *What is the acceptable time for you to process your data?*
 5. *Is the Hadoop infrastructure that you're considering flexible enough for all your use cases?*
 6. *How will your use of Hadoop expand across users, applications, ...?*
 7. *What existing tools and infrastructure do you want to integrate with Hadoop?*
 8. *Will you consider Hadoop on premise or Hadoop in the cloud?*
 9. *Will your administrators need management tools ? Hadoop's core distribution does not provide management capabilities. Do you plan train your administrators or to hire for managing Hadoop?*

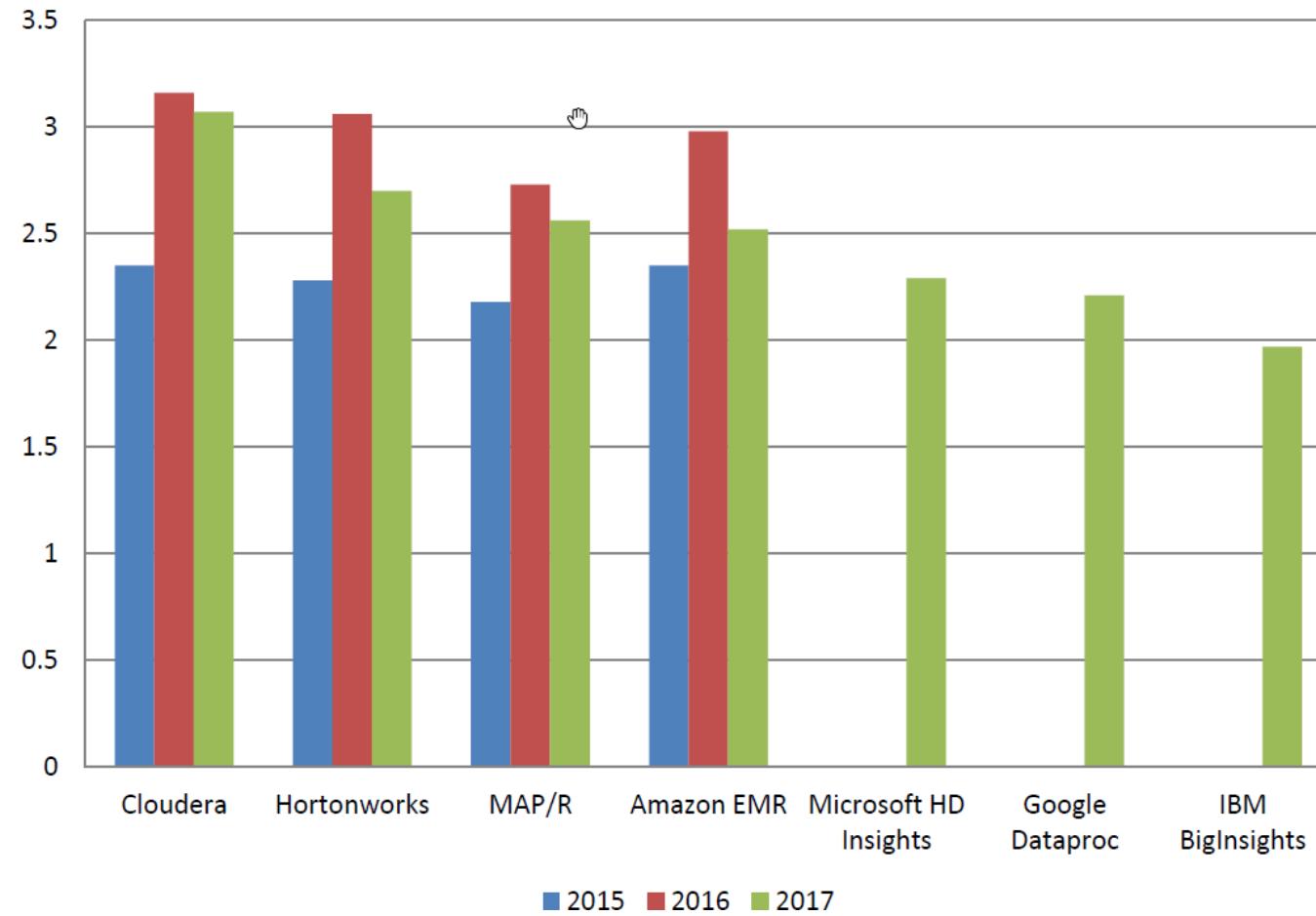
Evaluating Hadoop Vendors Offers



- **Hadoop core**
Ensure the distribution includes all Hadoop tools you plan to use
 - **Management capabilities**
Vendors provide software that helps administrators configure, monitor, and tune Hadoop. It can be open-source or not
 - **Enterprise reliability and integration**
You need to consider high availability, scalability and reliability. You may need to buy from third-party vendors connectors to existing enterprise systems if your vendors do not provide them.
 - **Support**
Most of the Hadoop vendors provide technical guidance and assistance

Hadoop Vendors Market Share

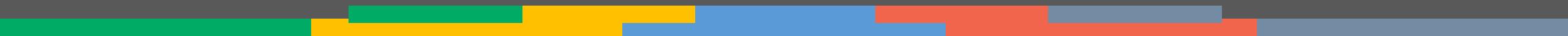
Big Data Distributions 2015-2017



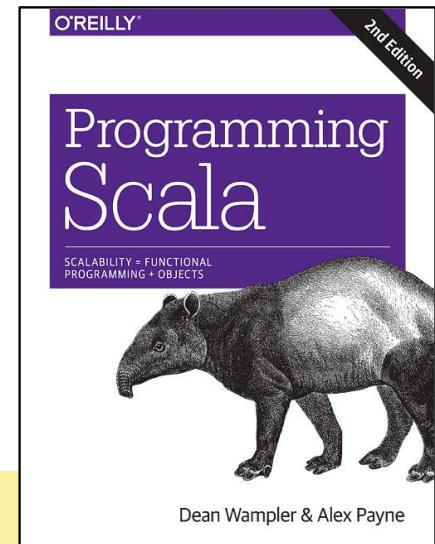
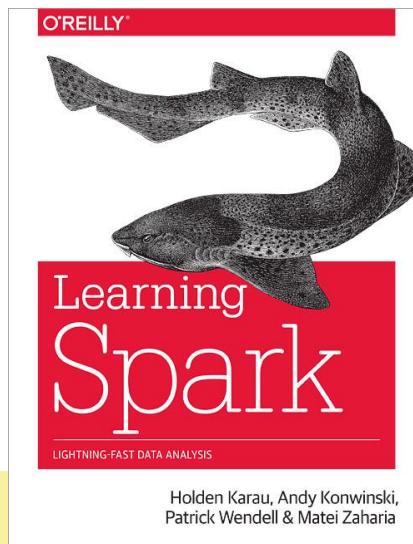
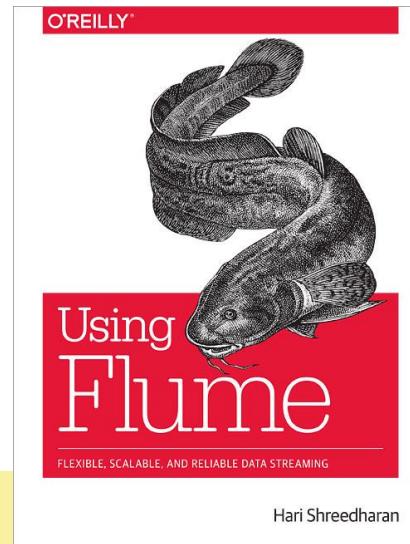
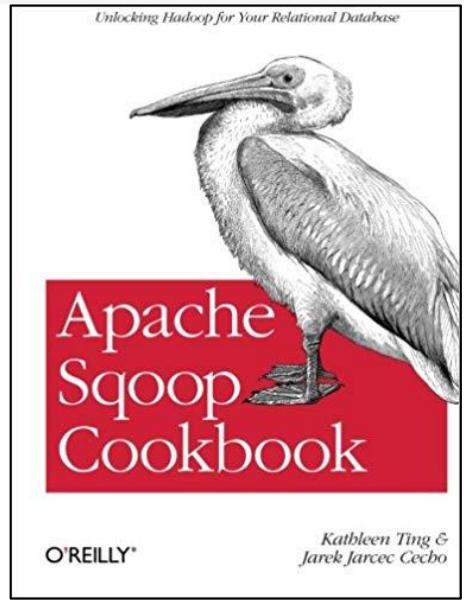
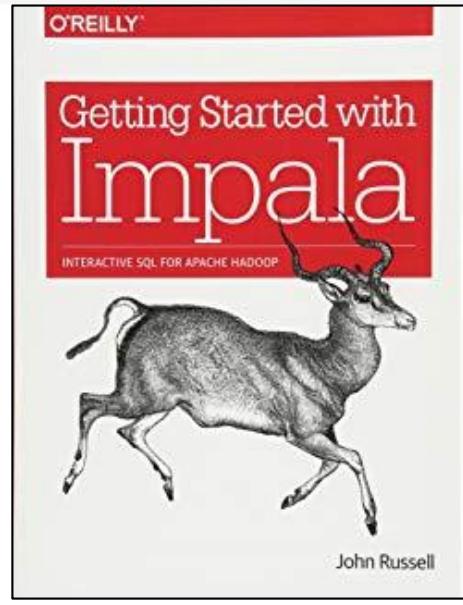
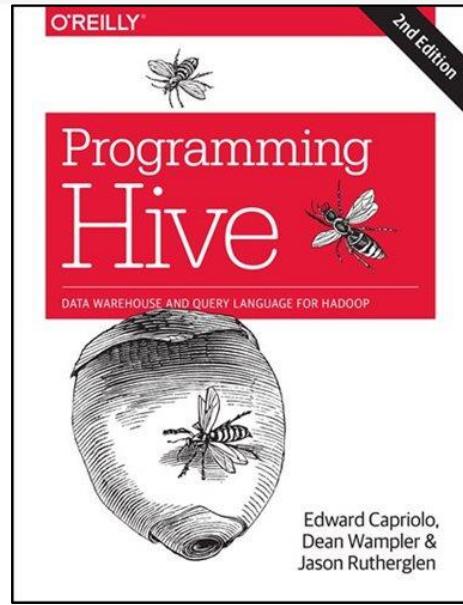
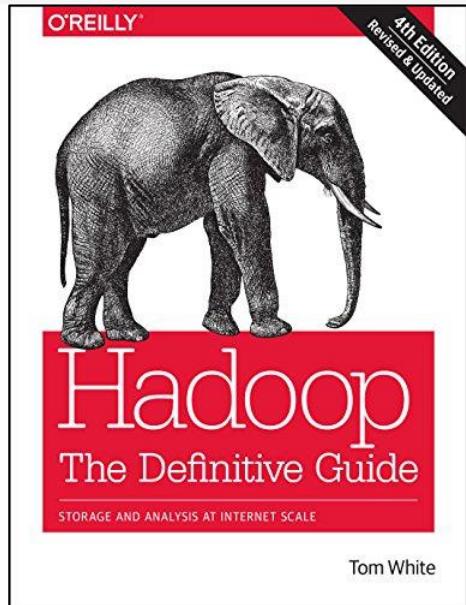
Full report:

https://www.microstrategy.com/getmedia/cd052225-be60-49fd-ab1c-4984ebc3cde9/Dresner-Report-Big_Data_Analytic_Market_Study-WisdomofCrowdsSeries-2017.pdf

Preparing The Technical Environment



Resources



Thank You

