Hadoop and R Integration for Large Data Processing

For Short Term Course on
Data Storage and Processing Techniques in Cloud Environment

Koushik Mondal
gemkousk@gmail.com

June 02, 2016

Motivation

- ✓ The main aim is to Design, Build and Maintain a secure long term data processing infrastructure for research.

- ✓ We will use some Open Source Software and framework which will help us
  - to build mathematical models;
  - incorporated large datasets without hassle;
  - produce results and present it through graphics rich window.

- ✓ Both R and Hadoop are Open Source and Data Driven. Thus "Cumulative force" will bring some exciting results

Is there any hurdle which debarring them to work together?

Advantages of choosing the right framework

- ✓ It describes the efficient use of a simple model applied to volumes of data that would be too large for the traditional analytical environment .

  (A simple algorithm with a large volume of data is more accurate than a sophisticated algorithm with little data. The algorithm is not the competitive advantage; the ability to apply it to huge amounts of data, without compromising performance, generates the competitive edge.)

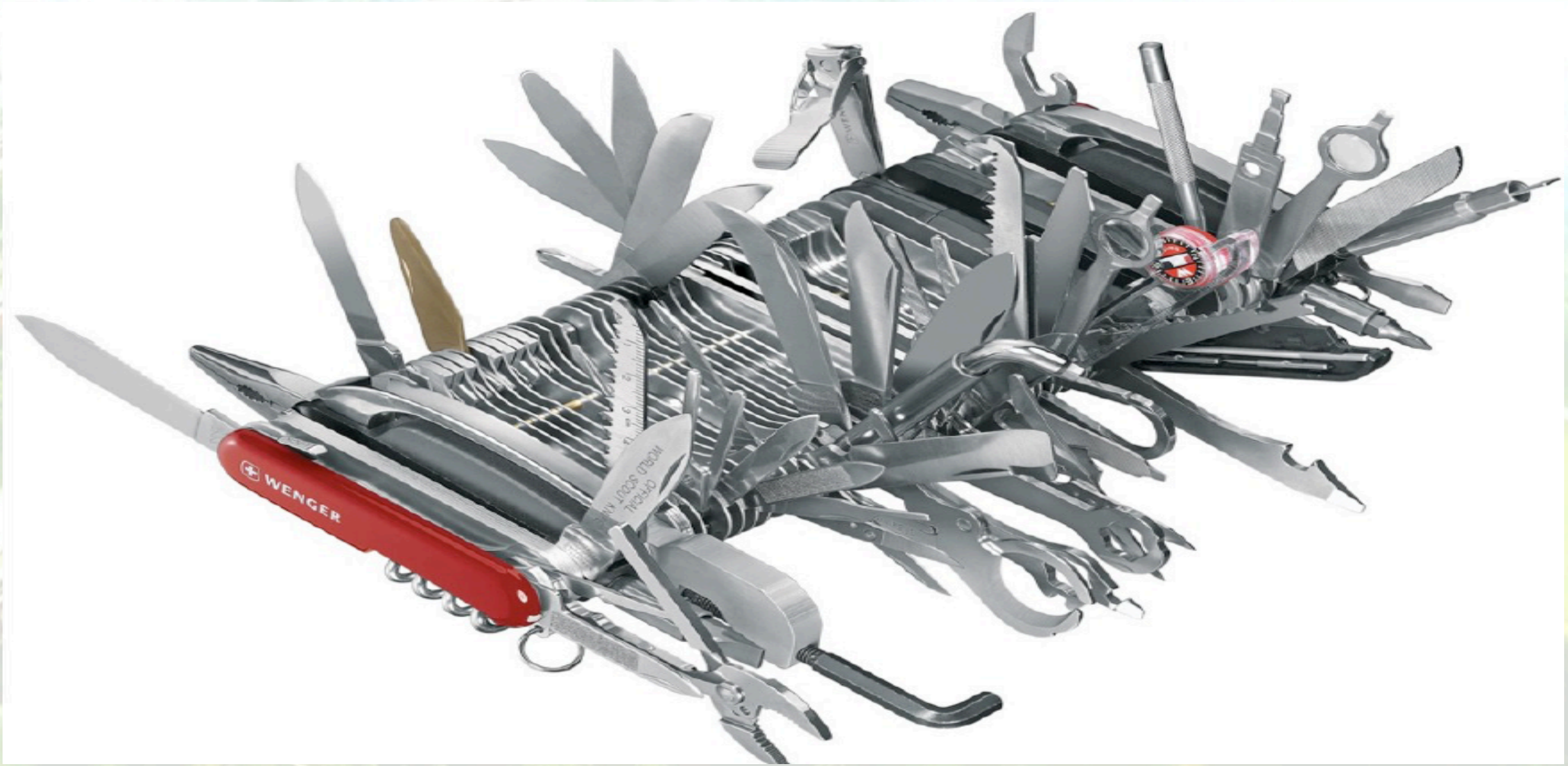
- ✓ It also refers to the sophistication of the model itself.

- ✓ Our main objectives for choosing the right framework are:

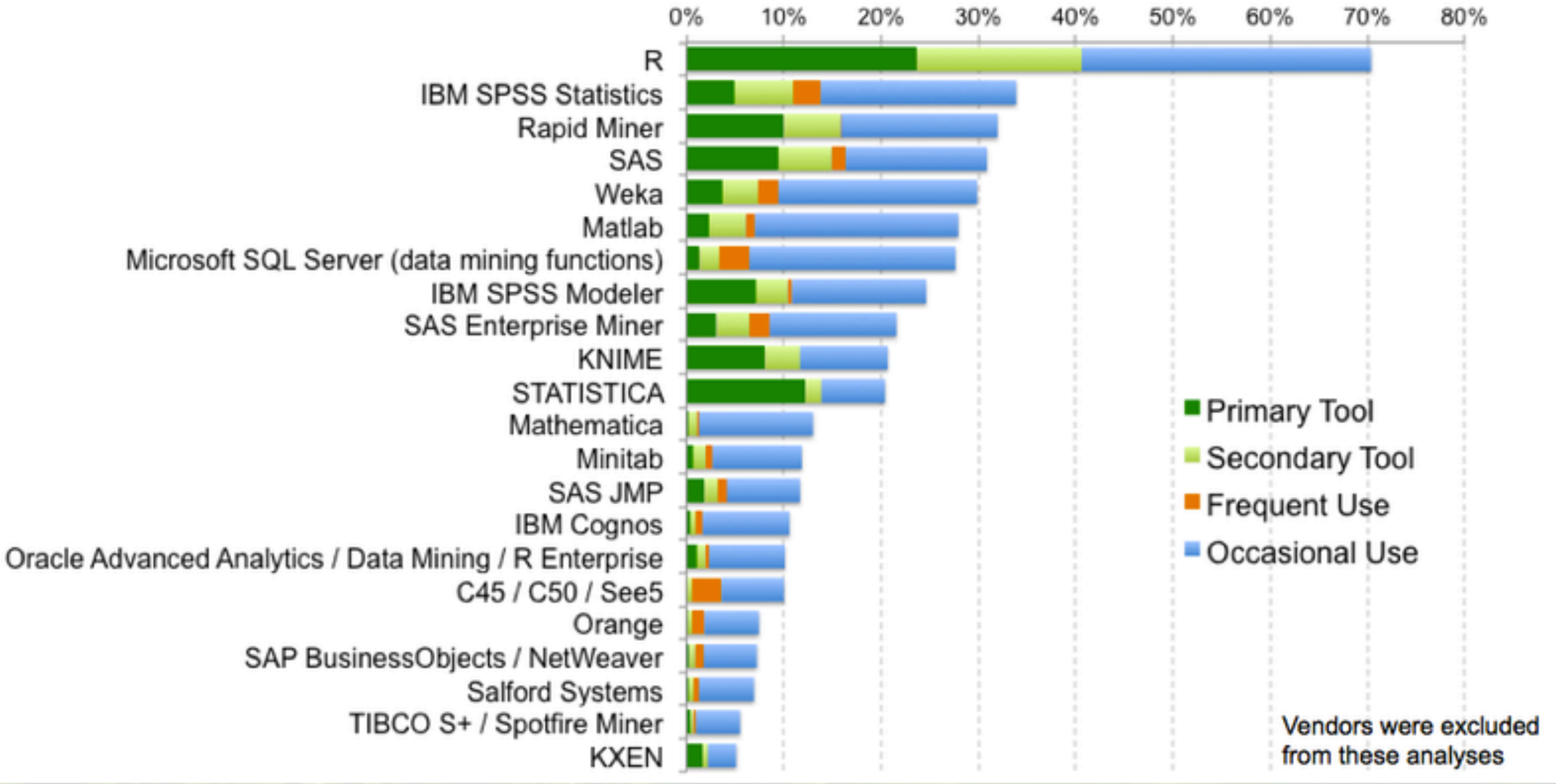
  - Avoid Sampling;
  - Reduce data movement and replication;
  - Bring the analytics as close as possible to data and;
  - Optimize computation speed.

Tools: Why R?

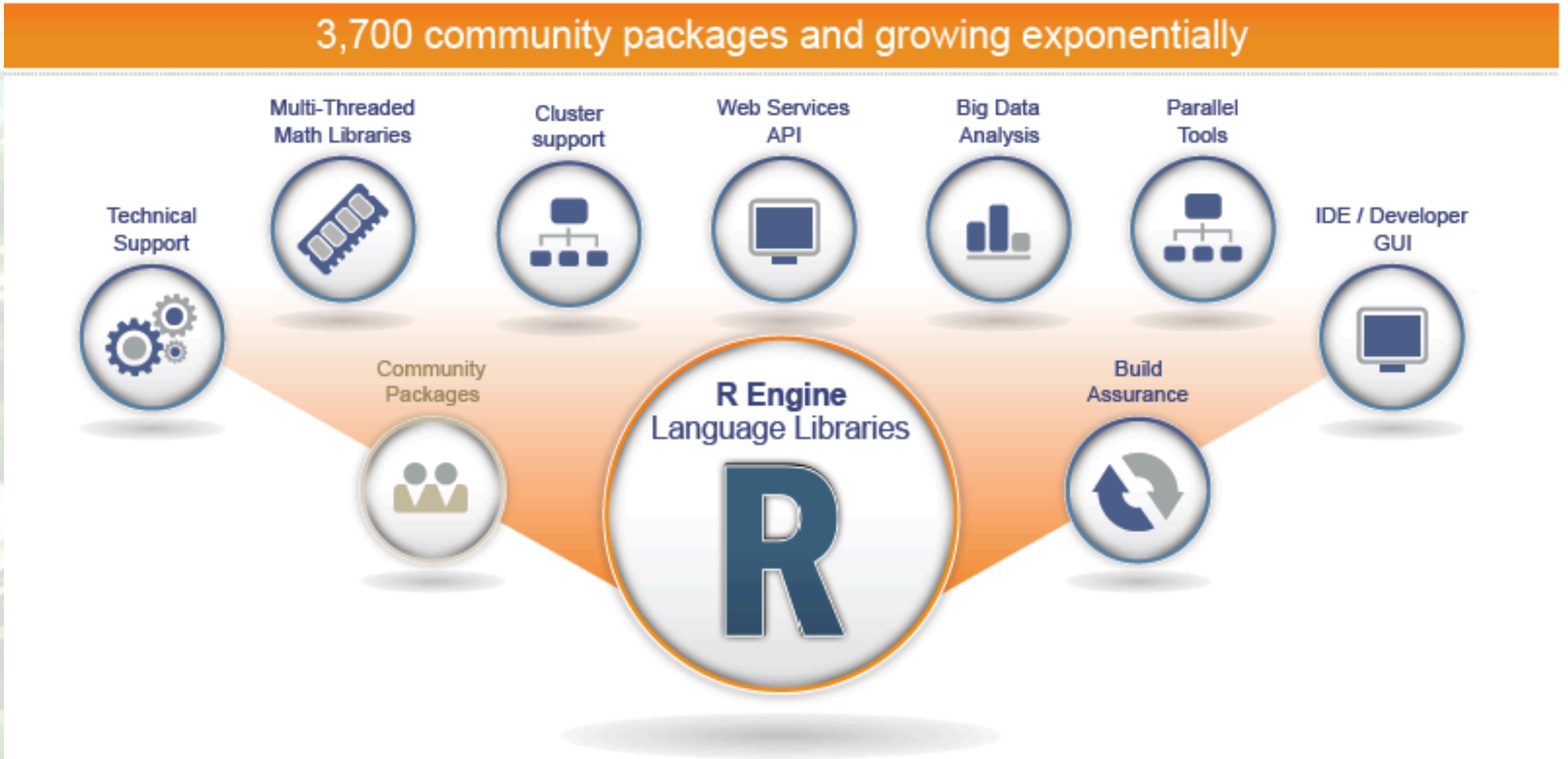
Tools: Why R?



Tools: Why R?


0%
10%
20%
30%
40%
50%
60%
70%
80%
R
IBM SPSS Statistics
Rapid Miner
SAS
Weka
Matlab
Microsoft SQL Server (data mining functions)
IBM SPSS Modeler
SAS Enterprise Miner
KNIME
STATISTICA
Mathematica
Minitab
SAS JMP
IBM Cognos
Oracle Advanced Analytics / Data Mining / R Enterprise
C45 / C50 / See5
Orange
SAP BusinessObjects / NetWeaver
Salford Systems
TIBCO S+ / Spotfire Miner
KXEN
Primary Tool
Secondary Tool
Frequent Use
Occasional Use
Vendors were excluded from these analyses

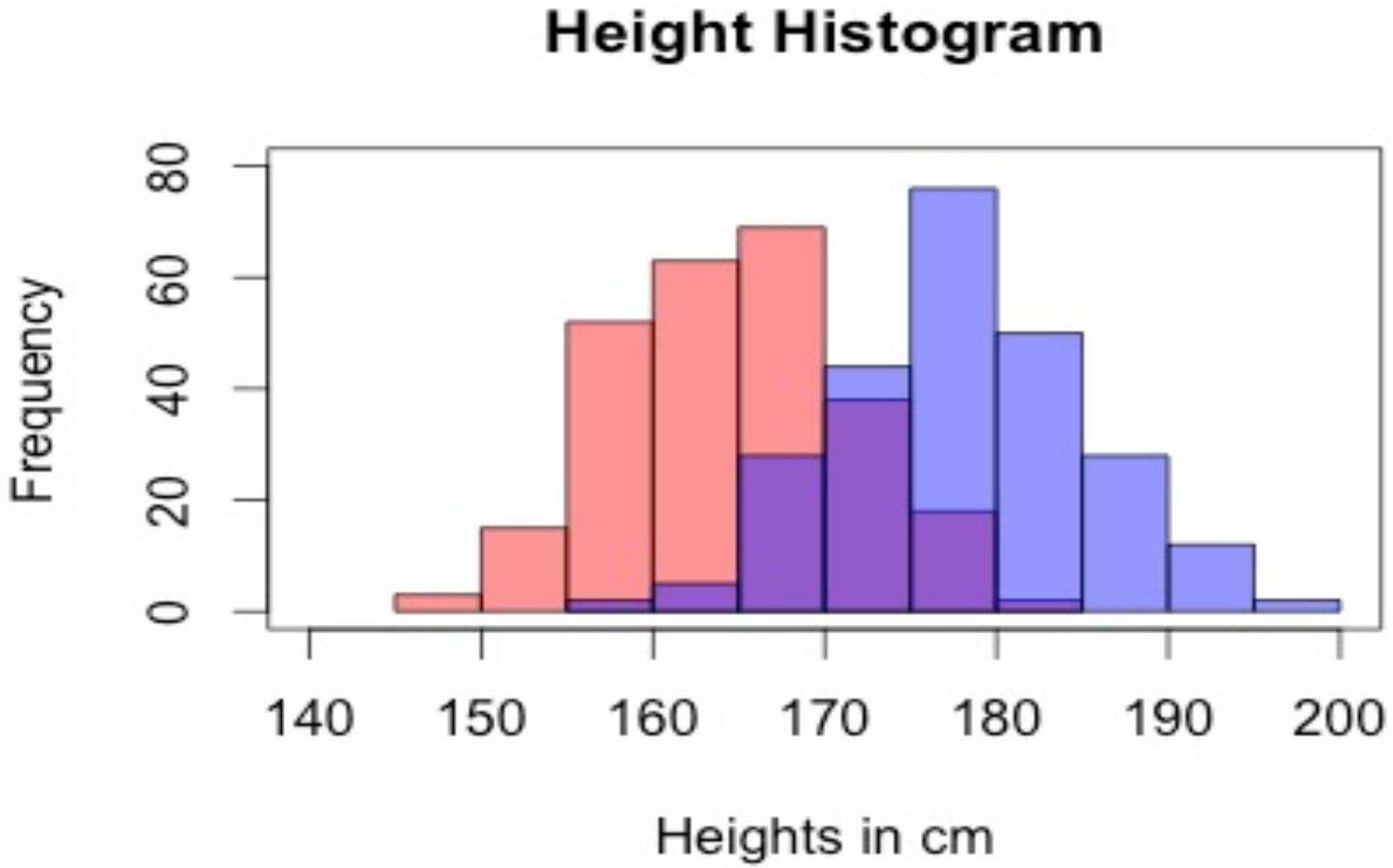
dim(available.packages())



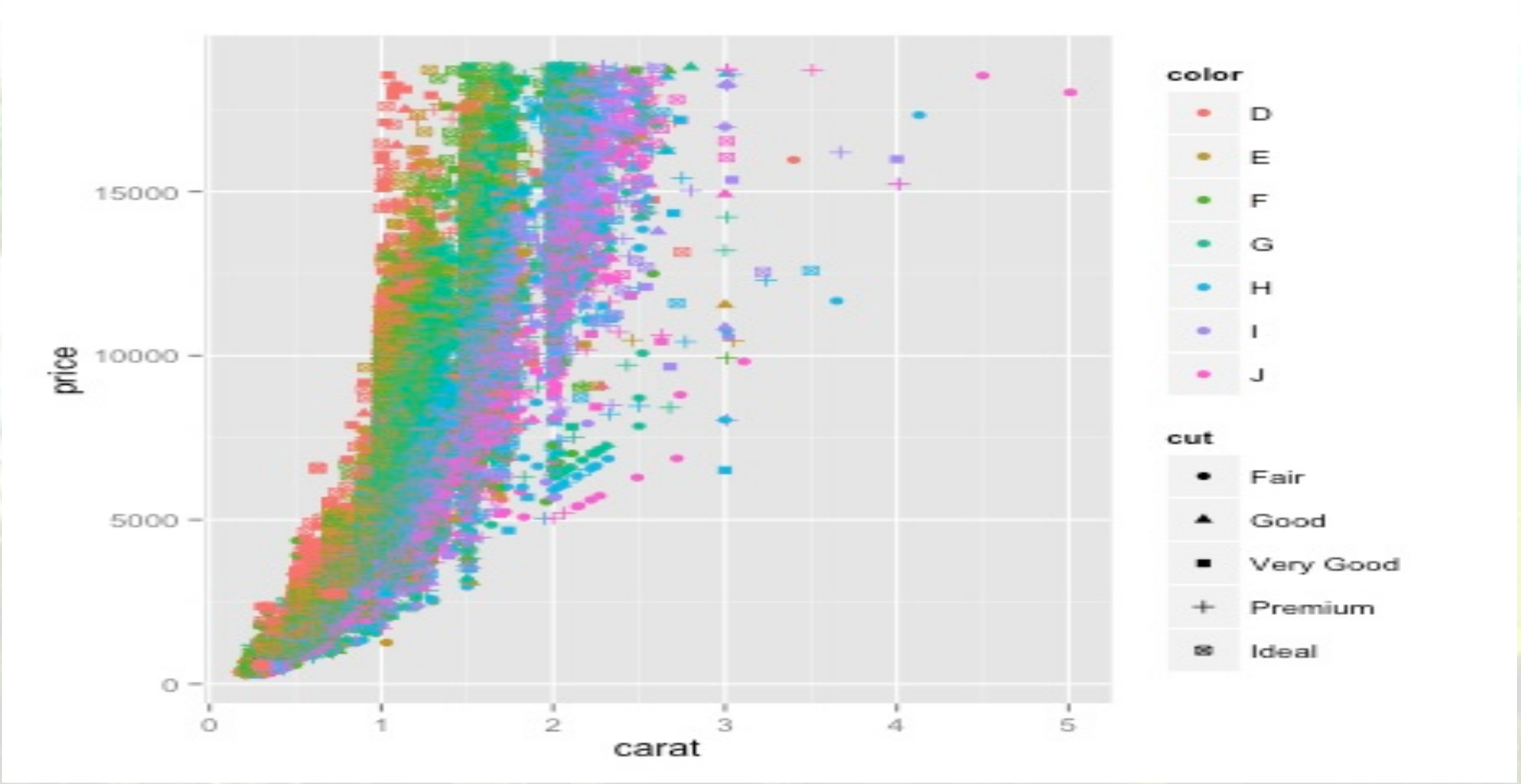
Example:

```
setwd("~/Desktop/ISM/Hadoop/hadoop-1.1.2")

load("bdims.RData")

mdims <- subset(bdims, sex == 1)

fdims <- subset(bdims, sex == 0)

hist(fdims$hgt, main="Height Histogram", xlab="Heights in cm",col=rgb(1,0,0,0.5),xlim=c(140,200),ylim=c(0,80))

hist(mdims$hgt, main="Female Height Histogram", col=rgb(0,0,1,0.5),add=T)

box()
```

Output


Height Histogram
Frequency
0
20
40
60
80
140
150
160
170
180
190
200
Heights in cm

Another Output:



Why Hadoop?

- ✓ Apache Hadoop is an open source software framework for storage and large scale processing of data-sets on clusters of commodity hardware.
- ✓ Inspired by Google's MapReduce-based computational infrastructure.
- ✓ Comprised of several components
  - Hadoop Distributed File System (HDFS)
  - MapReduce processing framework, job scheduler, etc.
  - Ingest/outgest services (Sqoop, Flume, etc.)
- ✓ Higher level languages and libraries (Hive, Pig, Cascading, Mahout)
- ✓ Written in Java, first opened up to alternatives through its Streaming API
  - → If your language of choice can handle stdin and stdout, you can use it to write MapReduce jobs

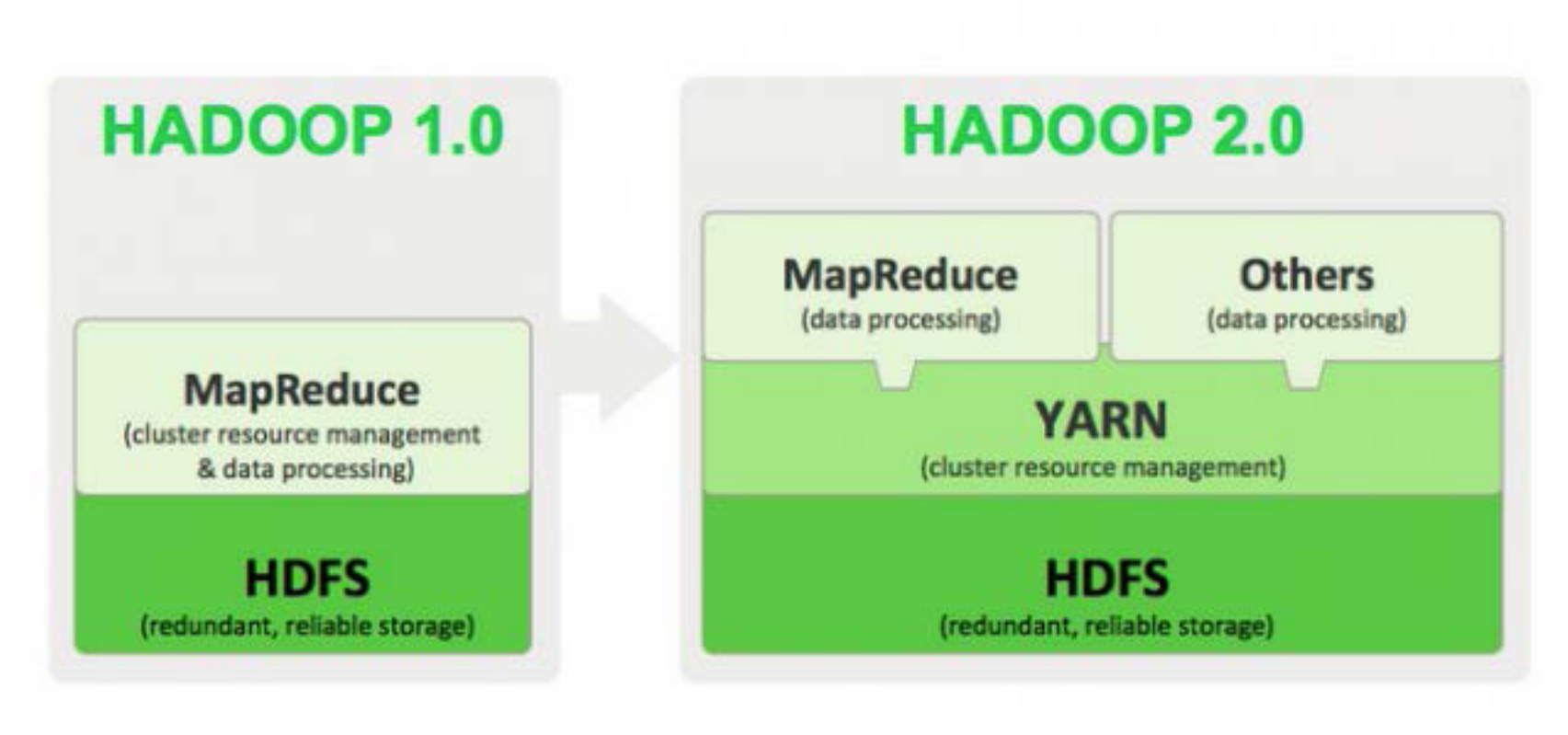
Apache Framework Basic Modules

- ✓ Hadoop Common
- ✓ Hadoop Distributed File System(HDFS)

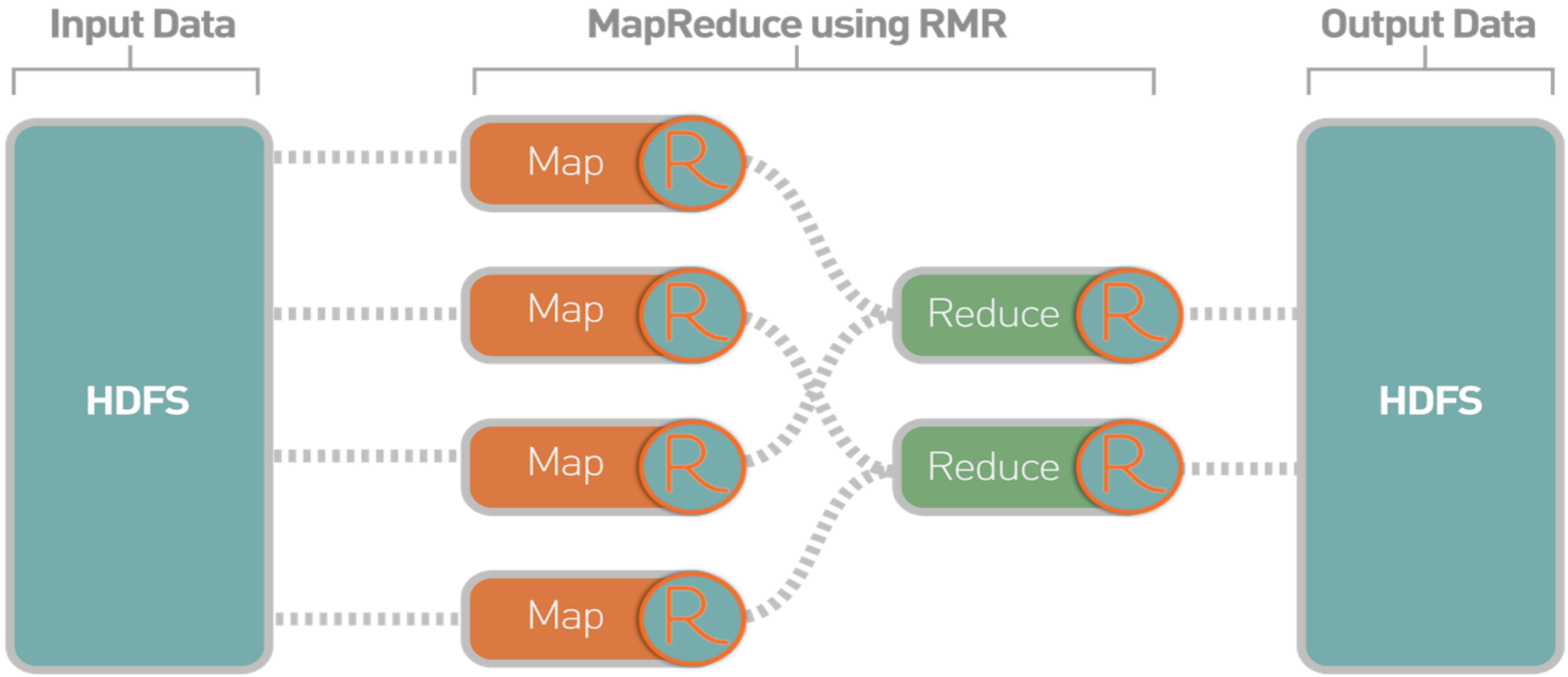
  Distributed, scalable, and portable file-system written in Java for the Hadoop framework

- ✓ Hadoop MapReduce
- ✓ Hadoop YARN

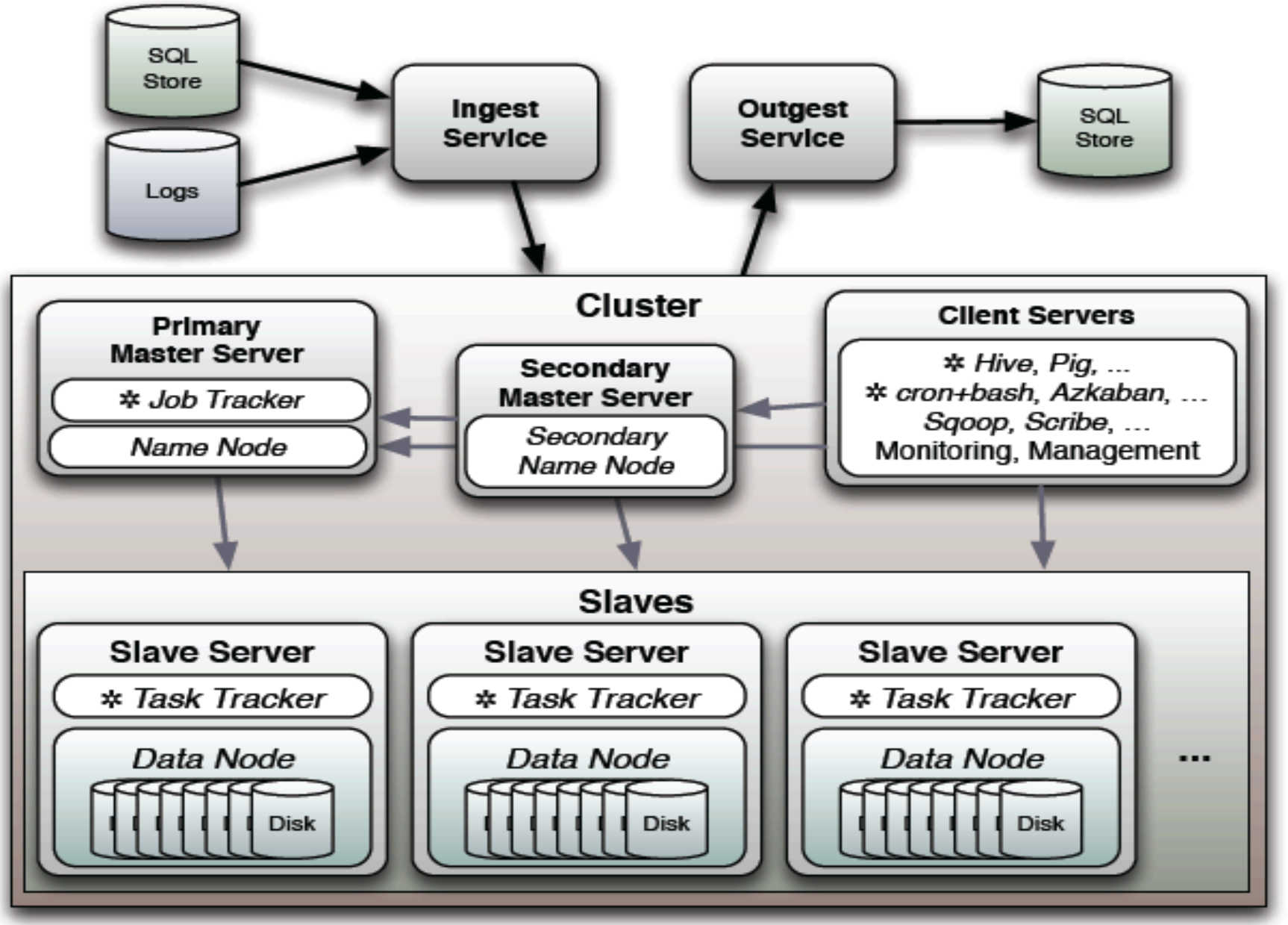
Hadoop Versions



MapReduce in Action



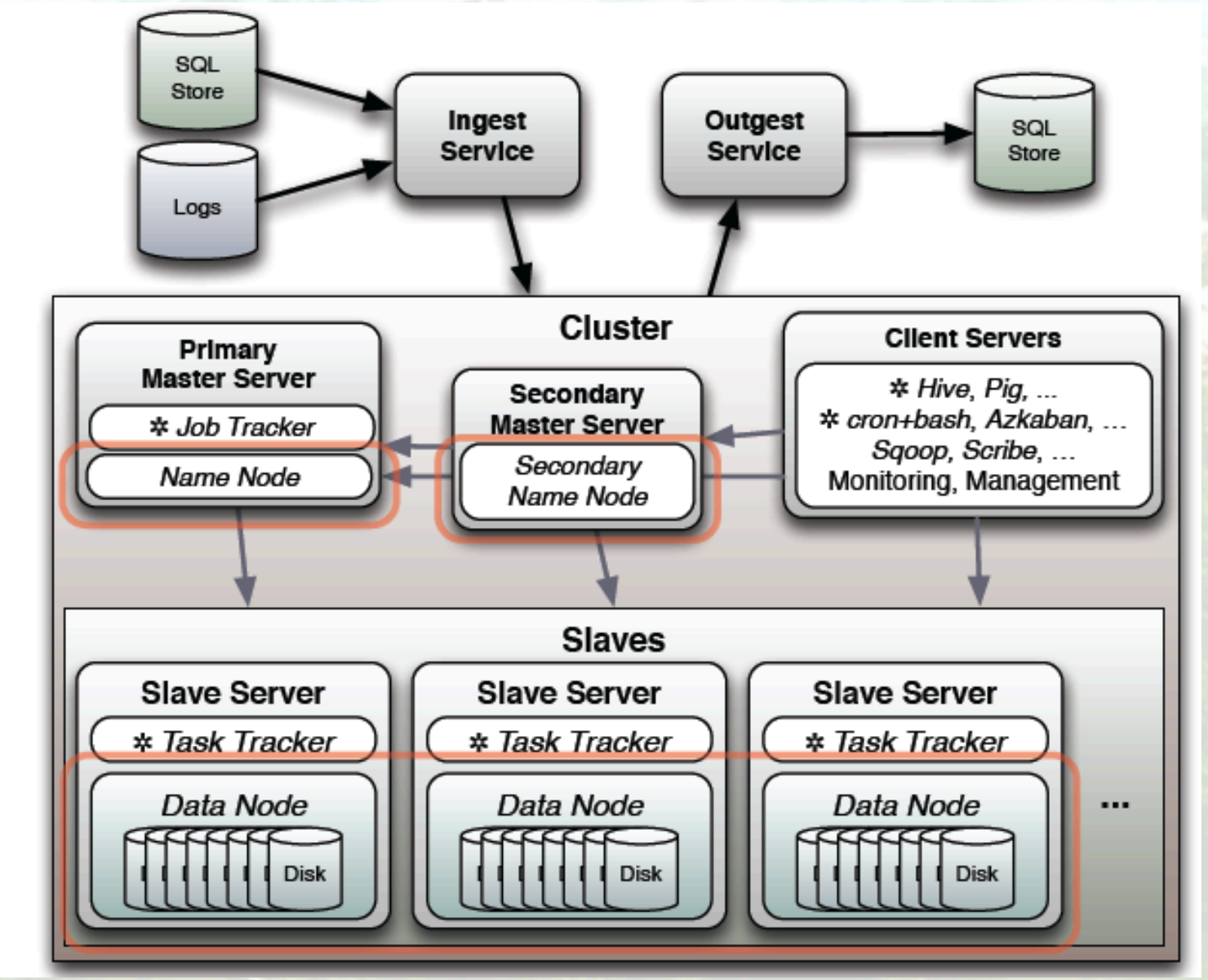
Hadoop Cluster Components



Italics: process

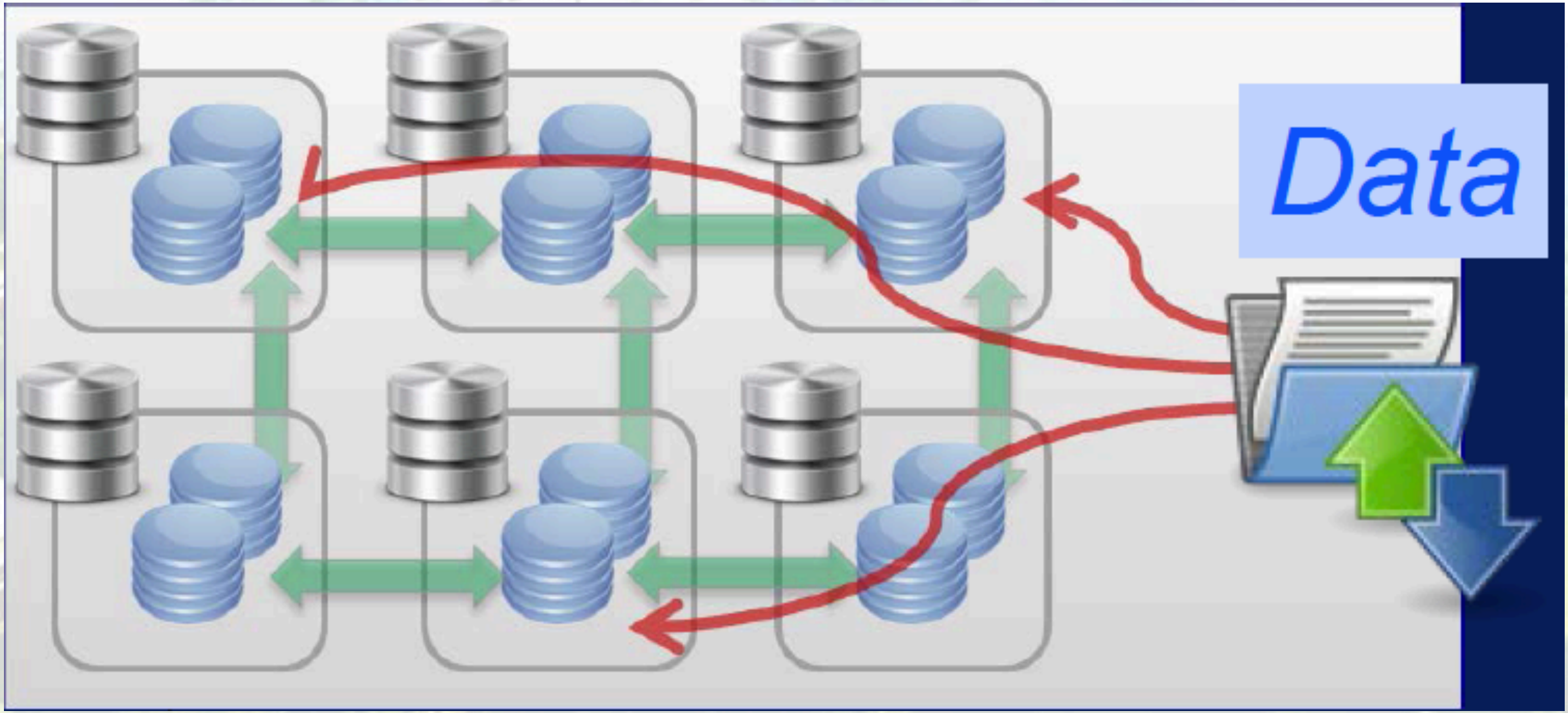
*** : MR Jobs**

Hadoop's Distributed File System

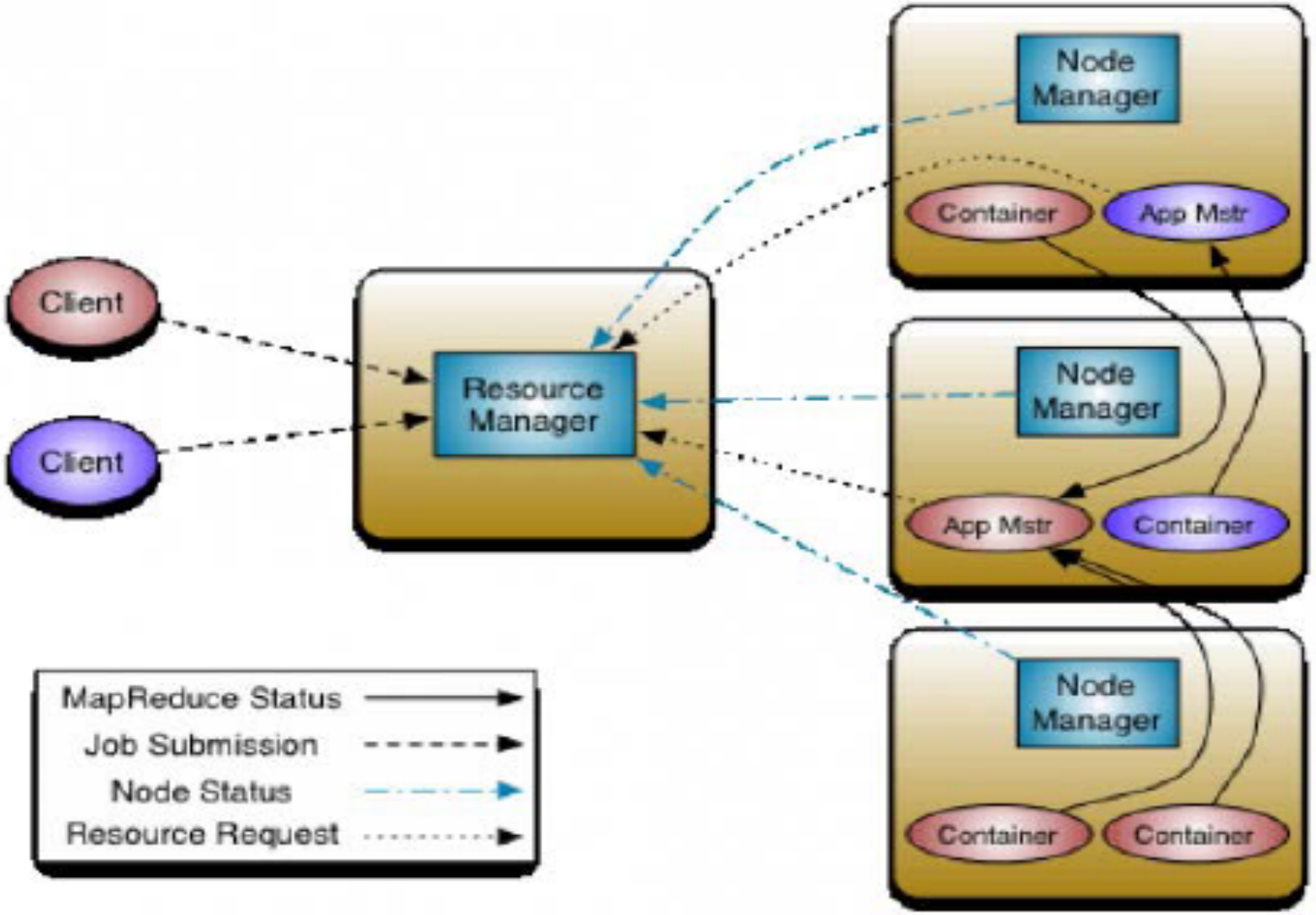


- Primary Name Node
- Secondary Name Node
- Job Tracker
- Data Node
- Task Tracker

Moving Computation to Data

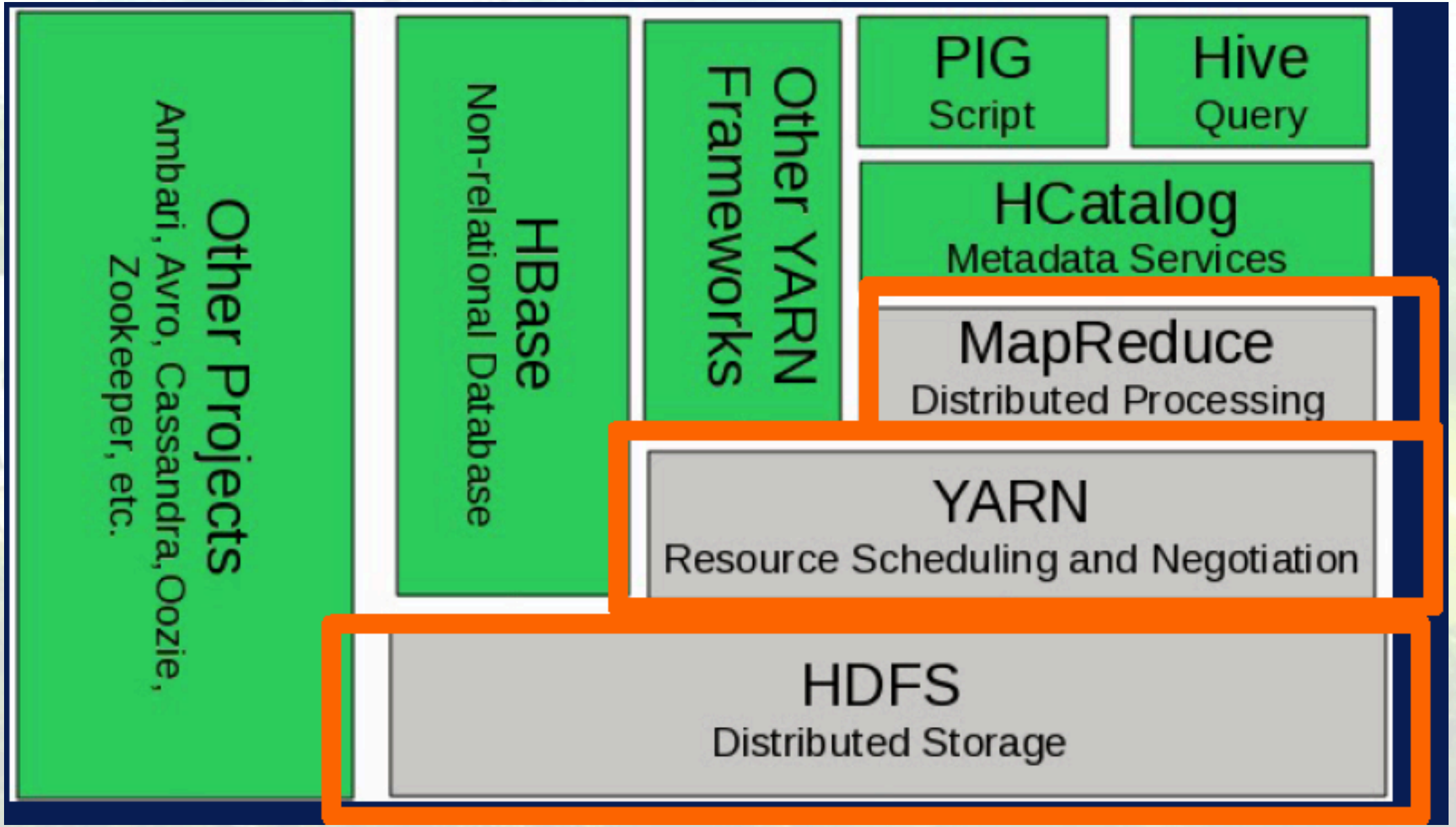


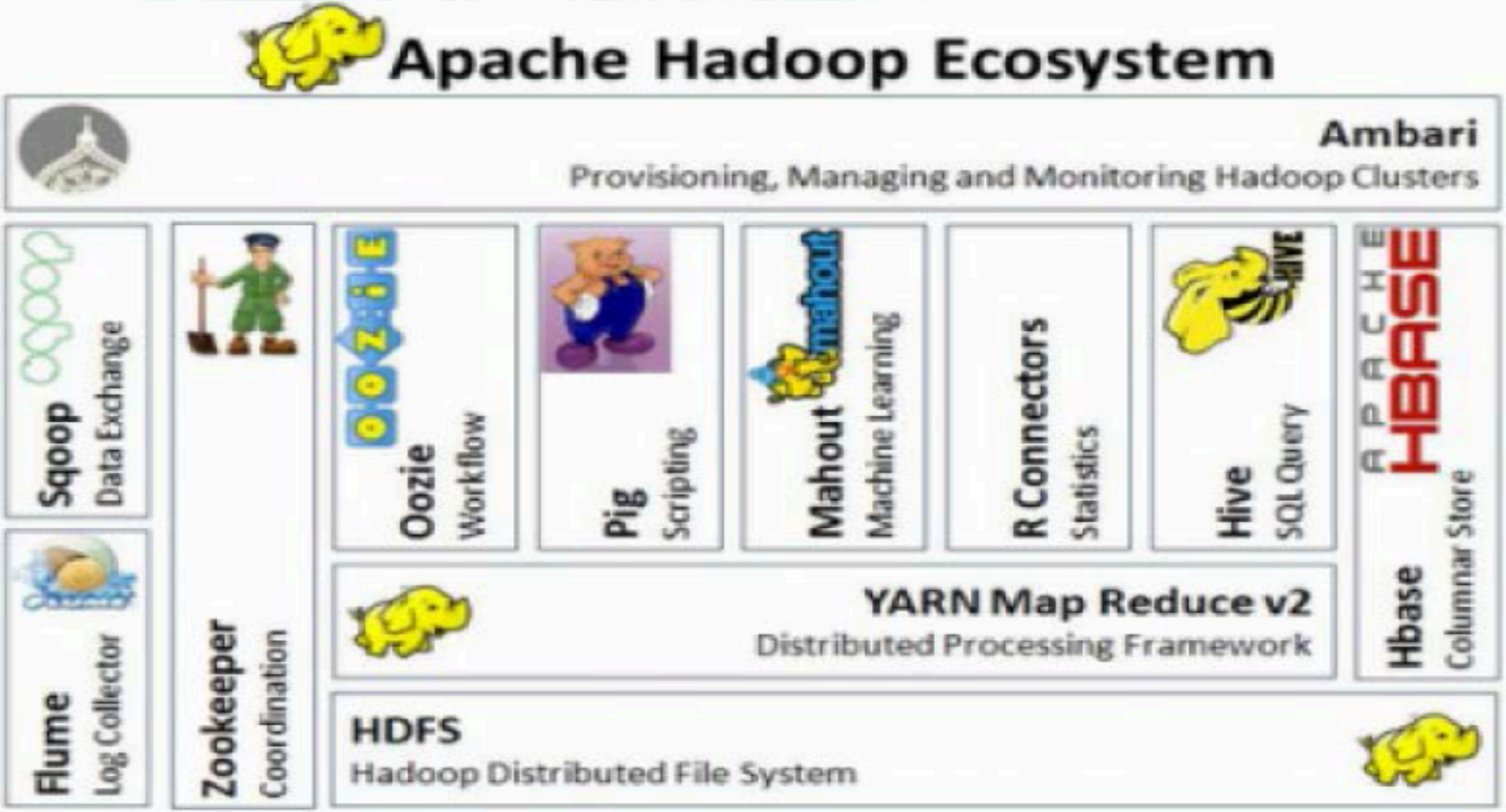
YARN



Scalability
Improved cluster Utilization
MapReduce Compatibility
Supports other workloads

Framework Stack




Apache Hadoop Ecosystem
Ambari
Provisioning, Managing and Monitoring Hadoop Clusters
Sqoop
Data Exchange
Flume
Log Collector
Zookeeper
Coordination
Oozie
Workflow
Pig
Scripting
Mahout
Machine Learning
R Connectors
Statistics
Hive
SQL Query
Hbase
Columnar Store
YARN Map Reduce v2
Distributed Processing Framework
HDFS
Hadoop Distributed File System

Details

Sqoop: Tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.

HBase: Column-oriented database management system
Not a Relational DBMS
Key-value store
Based on Google's Big Table

PIG: High level programming on top of Hadoop MapReduce (ETL framework)

HIVE: Data warehouse software facilitates querying and managing large datasets residing in distributed storage

Oozie: Workflow scheduler system to manage Apache Hadoop jobs

Zookeeper: Provides operational services for a Hadoop cluster group services

Contd…

Flume: Distributed, reliable and available services for efficiently collecting, aggregating, and moving large amounts of log data.

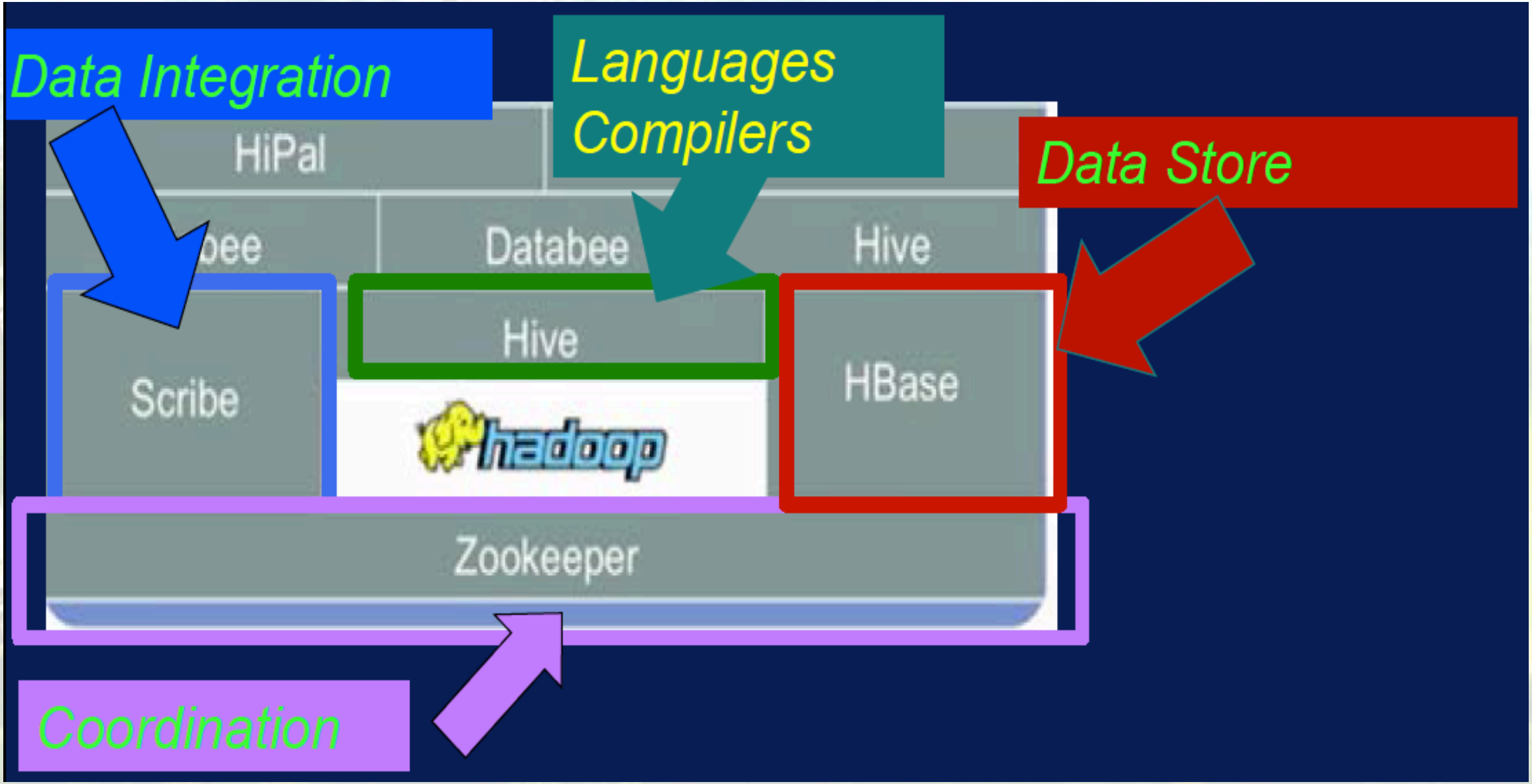
Spark: Apache Spark is a fast and general engine for large-scale data processing

And Important attribute…

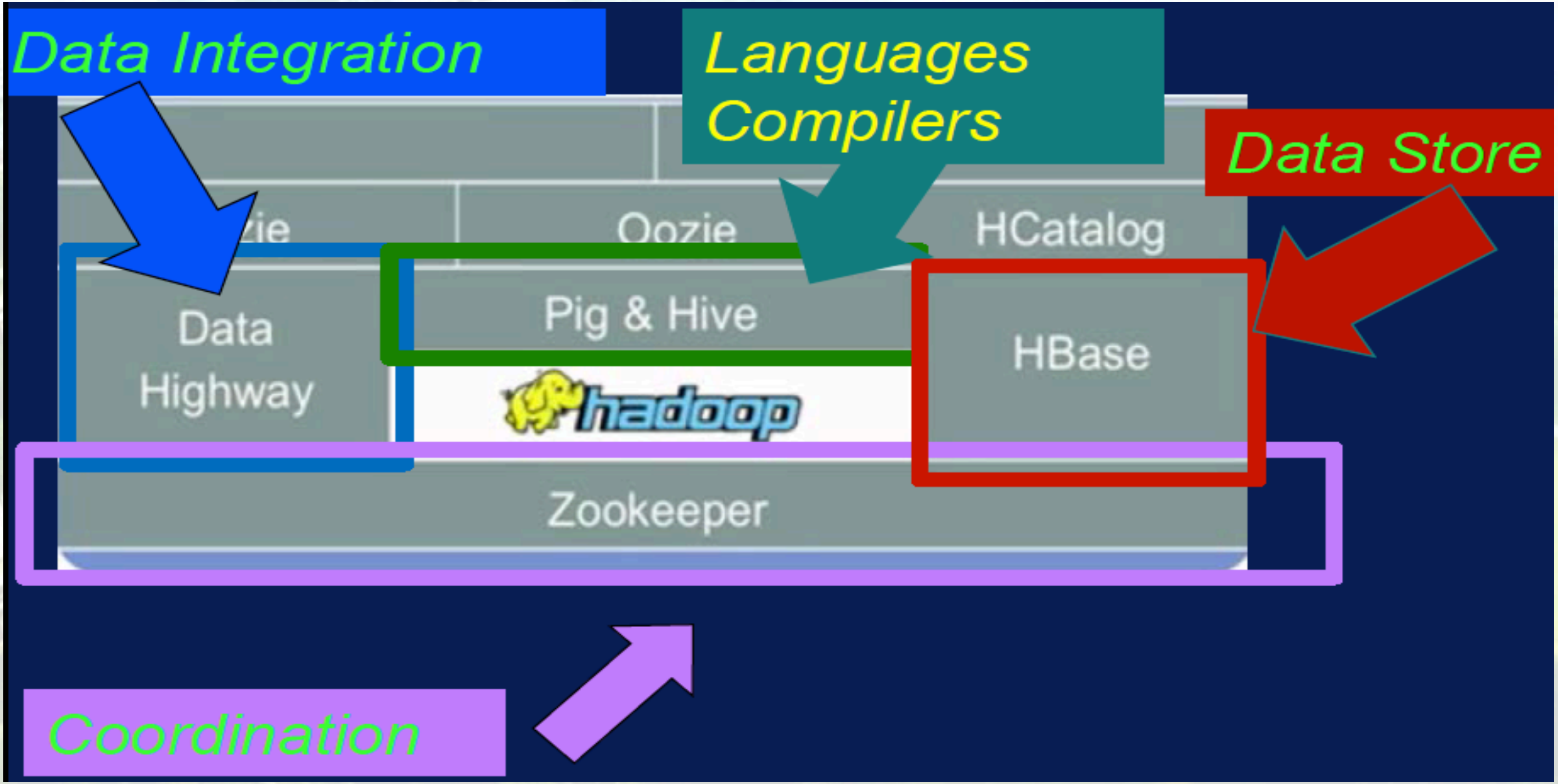
R Connectors???

Original Google Stack

Facebook's version of the Stack



Yahoo's version of the Stack



Main Configuration Files:

Check JAVA Version:

Koushik-Mondals-MacBook-Pro:~ koushikmondal$ javac -version

javac 1.6.0_65

Creating Password less ssh:

Koushik-Mondals-MacBook-Pro:~ koushikmondal$ ssh-keygen -t rsa -P ""

Generating public/private rsa key pair.

Enter file in which to save the key (/Users/koushikmondal/.ssh/id_rsa):

/Users/koushikmondal/.ssh/id_rsa already exists.

Overwrite (y/n)? Y

cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

ssh localhost

$ tar -xzvf hadoop-1.1.2.tar.gz x hadoop-1.1.2/

$ bin/hadoop namenode -format

Main Configuration Files:

Under $HADOOP_HOME/conf :

hadoop-env.sh : export JAVA_HOME=/Library/Java/Home
export HADOOP_HEAPSIZE=2000

Try the following command:
$ bin/hadoop
This will display the usage documentation for the hadoop script.

Now you are ready to start your Hadoop cluster in one of the three supported modes:

Local (Standalone) Mode
Pseudo-Distributed Mode
Fully-Distributed Mode

Standalone Mode:

```
$ mkdir input
$ cp conf/*.xml input
$ bin/hadoop jar hadoop-examples-*.jar grep input output 'dfs[a-z.]+'
$ cat output/*
```

But we are interested in Pseudo distributed Mode as if we are working in large cluster

Main Configuration Files:

conf/core-site.xml:

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

conf/hdfs-site.xml:

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

Main Configuration Files:

conf/mapred-site.xml:

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

Format a new distributed-filesystem:

```
$ bin/hadoop namenode -format
```

Start the hadoop daemons:

```
$ bin/start-all.sh
```

Browse the web interface for the NameNode and the JobTracker; by default they are available at:

NameNode - http://localhost:50070/

JobTracker - http://localhost:50030/

Main Configuration Files:

```
Koushik-Mondals-MacBook-Pro:hadoop-1.1.2 koushikmondal$ jps
80557 SecondaryNameNode
80470 DataNode
80710 TaskTracker
80623 JobTracker
80375 NameNode
80735 Jps

$ bin/hadoop fs -put conf input
$ bin/hadoop jar hadoop-examples-*.jar grep input output 'dfs[a-z.]+'      //example job 1
$ bin/hadoop jar hadoop-examples-1.1.2.jar pi 10 100                       //example job 2
$ /user/koushikmondal/output/_logs
$ bin/stop-all.sh
```

Main Configuration Files:

```
$ bin/start-all.sh
starting namenode, logging to /Users/koushikmondal/Desktop/ISM/Hadoop/hadoop-1.1.2/
libexec/../logs/hadoop-koushikmondal-namenode-Koushik-Mondals-MacBook-Pro.local.out
localhost: starting datanode, logging to /Users/koushikmondal/Desktop/ISM/Hadoop/
hadoop-1.1.2/libexec/../logs/hadoop-koushikmondal-datanode-Koushik-Mondals-MacBook-
Pro.local.out
localhost: starting secondarynamenode, logging to /Users/koushikmondal/Desktop/ISM/
Hadoop/hadoop-1.1.2/libexec/../logs/hadoop-koushikmondal-secondarynamenode-Koushik-
Mondals-MacBook-Pro.local.out
starting jobtracker, logging to /Users/koushikmondal/Desktop/ISM/Hadoop/hadoop-1.1.2/
libexec/../logs/hadoop-koushikmondal-jobtracker-Koushik-Mondals-MacBook-Pro.local.out
localhost: starting tasktracker, logging to /Users/koushikmondal/Desktop/ISM/Hadoop/
hadoop-1.1.2/libexec/../logs/hadoop-koushikmondal-tasktracker-Koushik-Mondals-MacBook-
Pro.local.out
```

Running Hadoop Job (Example Job 2)

```
Koushik-Mondals-MacBook-Pro:hadoop-1.1.2 koushikmondal$ bin/hadoop jar hadoop-examples-1.1.2.jar pi 10 100
Number of Maps  = 10
Samples per Map = 100
Wrote input for Map #0
Wrote input for Map #1
…  …
Wrote input for Map #8
Wrote input for Map #9
Starting Job
16/05/30 07:44:38 INFO mapred.FileInputFormat: Total input paths to process : 10
16/05/30 07:44:39 INFO mapred.JobClient: Running job: job_201605300742_0001
16/05/30 07:44:40 INFO mapred.JobClient:  map 0% reduce 0%
16/05/30 07:44:48 INFO mapred.JobClient:  map 20% reduce 0%
…    ….
16/05/30 07:45:13 INFO mapred.JobClient:    Total committed heap usage (bytes)=1931190272
16/05/30 07:45:13 INFO mapred.JobClient:    Map input bytes=240
16/05/30 07:45:13 INFO mapred.JobClient:    Reduce input records=20
16/05/30 07:45:13 INFO mapred.JobClient:    Combine output records=0
16/05/30 07:45:13 INFO mapred.JobClient:    Reduce output records=0
16/05/30 07:45:13 INFO mapred.JobClient:    Map output records=20
Job Finished in 34.679 seconds
Estimated value of Pi is 3.14800000000000000000
```

Example Job 2 : Jobtracker

localhost:50030/jobtracker.jsp

Quick Links

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

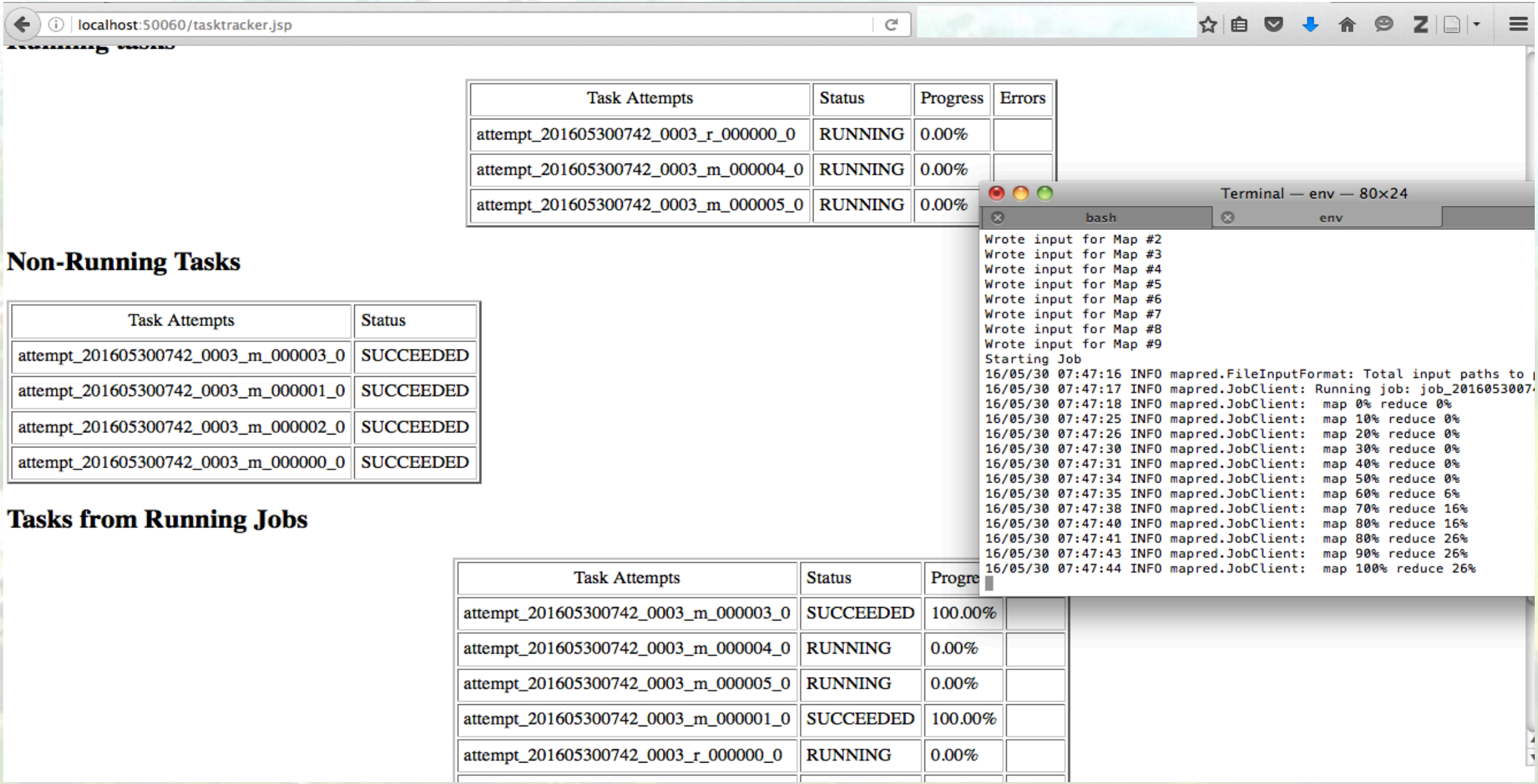
Running Jobs

none

Completed Jobs

Jobid	Started	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information	Diagnostic Info
job_201605300742_0001	Mon May 30 07:44:38 GMT+05:30 2016	NORMAL	koushikmondal	PiEstimator	100.00%	10	10	100.00%	1	1	NA	NA
job_201605300742_0002	Mon May 30 07:45:59 GMT+05:30 2016	NORMAL	koushikmondal	PiEstimator	100.00%	10	10	100.00%	1	1	NA	NA
job_201605300742_0003	Mon May 30 07:47:16 GMT+05:30 2016	NORMAL	koushikmondal	PiEstimator	100.00%	10	10	100.00%	1	1	NA	NA

Example Job 2 : Tasktracker



RHadoop

- ✓ RHadoop is an open source project sponsored by Revolution Analytics
- ✓ Package Overview
  - rmr2 - all MapReduce-related functions
  - rhdfs - interaction with Hadoop's HDFS file system
  - rhbase - access to the NoSQL HBase database
- ✓ rmr2 uses Hadoop's Streaming API to allow R users to write MapReduce jobs in R
  handles all of the I/O and job submission for you (no while(<stdin>)-like loops!)
- ✓ Modular
  - Packages group similar functions Only load (and learn!) what you need
  - Minimizes prerequisites and dependencies
- ✓ Open Source Cost: Low (no) barrier to start using
- ✓ Transparency: Development, issue tracker, Wiki, etc. hosted on
  github: https://github.com/RevolutionAnalytics/Rhadoop/

Setting Environment for RHadoop

In Rstudio:

```
Sys.setenv("HADOOP_HOME"="/Users/koushikmondal/Desktop/ISM/Hadoop/hadoop-1.1.2")

Sys.setenv("JAVA_HOME"="/Library/Java/Home")

Sys.setenv("HADOOP_STREAMING"="hadoop-1.1.2/contrib/streaming/hadoop-streaming-1.1.2.jar")

install.packages(“rhbase”)   install.packages(“rhdfs”)   install.packages(“rmr2”)

library(rhbase)   library(rhdfs)   library(rmr2)
```

Word Count in RHadoop

localhost:50060/tasktracker.jsp

tracker_localhost:localhost/127.0.0.1:51408 Task Tracker Status

Version: 1.1.2, r1440782
Compiled: Thu Jan 31 02:03:24 UTC 2013 by hortonfo

Running tasks

Task Attempts	Status	Progress	Errors

Non-Running Tasks

Task Attempts	Status

Tasks from Running Jobs

File: /user/koushikmondal/wordcount/data

Goto : /user/koushikmondal/word go

Go back to dir listing
Advanced view/download options

View Next chunk

Local Logs

```
holy bible authorized king james version textfile 890904
in the beginning god created the heaven and the earth
and the earth was without form and void and darkness was upon the face of the deep and the spirit of god moved upon the face of the waters
and god said let there be light and there was light
and god saw the light that it was good and god divided the light from the darkness
and god called the light day and the darkness he called night and the evening and the morning were the first day
and god said let there be a firmament in the midst of the waters and let it divide the waters from the waters
and god made the firmament and divided the waters which were under the firmament from the waters which were above the firmament and it was so
and god called the firmament heaven and the evening and the morning were the second day
and god said let the waters under the heaven be gathered together unto one place and let the dry land appear and it was so
and god called the dry land earth and the gathering together of the waters called he seas and god saw that it was good
and god said let the earth bring forth grass the herb yielding seed and the fruit tree yielding fruit after his kind whose seed is in itself upon the earth and it
was so
and the earth brought forth grass and herb yielding seed after his kind and the tree yielding fruit whose seed was in itself after his kind and god saw that it was
good
and the evening and the morning were the third day
and god said let there be lights in the firmament of the heaven to divide the day from the night and let them be for signs and for seasons and for days and years
and let them be for lights in the firmament of the heaven to give light upon the earth and it was so
and god made two great lights the greater light to rule the day and the lesser light to rule the night he made the stars also
and god set them in the firmament of the heaven to give light upon the earth
and to rule over the day and over the night and to divide the light from the darkness and god saw that it was good
and the evening and the morning were the fourth day
and god said let the waters bring forth abundantly the moving creature that hath life and fowl that may fly above the earth in the open firmament of heaven
and god created great whales and every living creature that moveth which the waters brought forth abundantly after their kind and every winged fowl after his kind
and god saw that it was good
and god blessed them saying be fruitful and multiply and fill the waters in the seas and let fowl multiply in the earth
```

Download this file
Tail this file

tracker_localhost:localhost/127.0.0.1:51408 Task Tracker Status

Version: 1.1.2, r1440782
Compiled: Thu Jan 31 02:03:24 UTC 2013 by hortonfo

Running tasks

Task Attempts	Status	Progress	Errors
attempt_201605291327_0001_r_000000_0	RUNNING	33.33%	

Non-Running Tasks

Task Attempts	Status
attempt_201605291327_0001_m_000000_0	SUCCEEDED
attempt_201605291327_0001_m_000001_0	SUCCEEDED

Task Attempts	Status	Progress	Errors
attempt_201605291327_0001_r_000000_0	RUNNING	33.33%	
attempt_201605291327_0001_m_000000_0	SUCCEEDED	100.00%	
attempt_201605291327_0001_m_000001_0	SUCCEEDED	100.00%	

Network Setup

Cluster Summary (Heap Size is 81.06 MB/1.95 GB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots
0	1	1	1	0	1	0	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

Jobid	Started	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduli Informati
job_201605291221_0001	Sun May 29 12:26:02 GMT+05:30 2016	NORMAL	koushikmondal	grep-search	100.00%	16	16	25.00%	1	0	NA

Retired Jobs

```
16/05/29 12:27:02 INFO mapred.JobClient:   File Output Format C
16/05/29 12:27:02 INFO mapred.JobClient:     Bytes Written=52
16/05/29 12:27:02 INFO mapred.JobClient:   FileSystemCounters
16/05/29 12:27:02 INFO mapred.JobClient:     FILE_BYTES_READ=82
16/05/29 12:27:02 INFO mapred.JobClient:     HDFS_BYTES_READ=30
16/05/29 12:27:02 INFO mapred.JobClient:     FILE_BYTES_WRITTEN
16/05/29 12:27:02 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN
16/05/29 12:27:02 INFO mapred.JobClient:   Map-Reduce Framework
16/05/29 12:27:02 INFO mapred.JobClient:     Map output materia
16/05/29 12:27:02 INFO mapred.JobClient:     Map input records=
16/05/29 12:27:02 INFO mapred.JobClient:     Reduce shuffle byt
16/05/29 12:27:02 INFO mapred.JobClient:     Spilled Records=6
16/05/29 12:27:02 INFO mapred.JobClient:     Map output bytes=7
16/05/29 12:27:02 INFO mapred.JobClient:     Total committed he
269619200
16/05/29 12:27:02 INFO mapred.JobClient:     Map input bytes=94
16/05/29 12:27:02 INFO mapred.JobClient:     Combine input reco
16/05/29 12:27:02 INFO mapred.JobClient:     SPLIT_RAW_BYTES=12
16/05/29 12:27:02 INFO mapred.JobClient:     Reduce input recor
16/05/29 12:27:02 INFO mapred.JobClient:     Reduce input groups
16/05/29 12:27:02 INFO mapred.JobClient:     Combine output rec
16/05/29 12:27:02 INFO mapred.JobClient:     Reduce output reco
16/05/29 12:27:02 INFO mapred.JobClient:     Map output records=
Koushik-Mondals-MacBook-Pro:hadoop-1.1.2 koushikmondal$
```

Hadoop job_201605291221_0001 on localhost — http://localhost:50030/jobdetails.jsp?jobid=job_201605291221_0001&refre

Hadoop job_201605291221_0001 on localhost

User: koushikmondal
Job Name: grep-search
Job File: hdfs://localhost:9000/tmp/koushikmondal/hadoop-koushikmondal/mapred/staging/koushikmondal/.staging/job_201605291221_0001/job.xml
Submit Host: Koushik-Mondals-MacBook-Pro.local
Submit Host Address: 127.0.0.1
Job-ACLs: All users are allowed
Job Setup: Successful
Status: Succeeded
Started at: Sun May 29 12:26:02 GMT+05:30 2016
Finished at: Sun May 29 12:26:45 GMT+05:30 2016
Finished in: 42sec
Job Cleanup: Successful

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	16	0	0	16	0	0 / 0
reduce	100.00%	1	0	0	1	0	0 / 0

	Counter	Map	Reduce	Total
File Input Format Counters	Bytes Read	27,600	0	27,600
Job Counters	SLOTS_MILLIS_MAPS	0	0	60,640
	Launched reduce tasks	0	0	1
	Total time spent by all reduces waiting after reserving slots (ms)	0	0	0
	Total time spent by all maps waiting after reserving slots (ms)	0	0	0
	Launched map tasks	0	0	16
	Data-local map tasks	0	0	16
	SLOTS_MILLIS_REDUCES	0	0	33,658
File Output Format Counters	Bytes Written	0	180	180
FileSystemCounters	FILE_BYTES_READ	0	82	82
	HDFS_BYTES_READ	29,466	0	29,466
	FILE_BYTES_WRITTEN	952,338	59,451	1,011,789
	HDFS_BYTES_WRITTEN	0	180	180
Map-Reduce Framework	Map output materialized bytes	172	0	172
	Map input records	776	0	776
	Reduce shuffle bytes	0	172	172
	Spilled Records	3	3	6
	Map output bytes	70	0	70
	Total committed heap usage (bytes)	2,953,904,128	85,000,192	3,038,904,320
	Map input bytes	27,600	0	27,600
	Combine input records	3	0	3
	SPLIT_RAW_BYTES	1,866	0	1,866
	Reduce input records	0	3	3
	Reduce input groups	0	3	3
	Combine output records	3	0	3
	Reduce output records	0	3	3
	Map output records	3	0	3

gemkousk@gmail.com