Tutorial 2.2 : Getting to know programming with Hadoop MapReduce

Asst. Prof. Dr. Walisa Romsaiyud

Graduate School of Information Technology, Siam University

walisa.rom@siam.edu

IEEE ICSEC 2013 – Tutorial 2.2 (September 4, 2013,13:00-16:00)

Acknowledgments

- Tom White, “Hadoop The Definitive Guide”, O’Reilly Media, 2009.
- IBM
- Teradata (http://www.teradata.com/)
- Hortonworks (http://hortonworks.com/)

Outline of the presentation

1. Big Data!
2. Hadoop
3. The Hadoop Distributed File System (HDFS)
4. Hadoop MapReduce
5. Hive
6. Pig
7. Publication

Big data!

Digital World: New Data Sets, New Possibilities

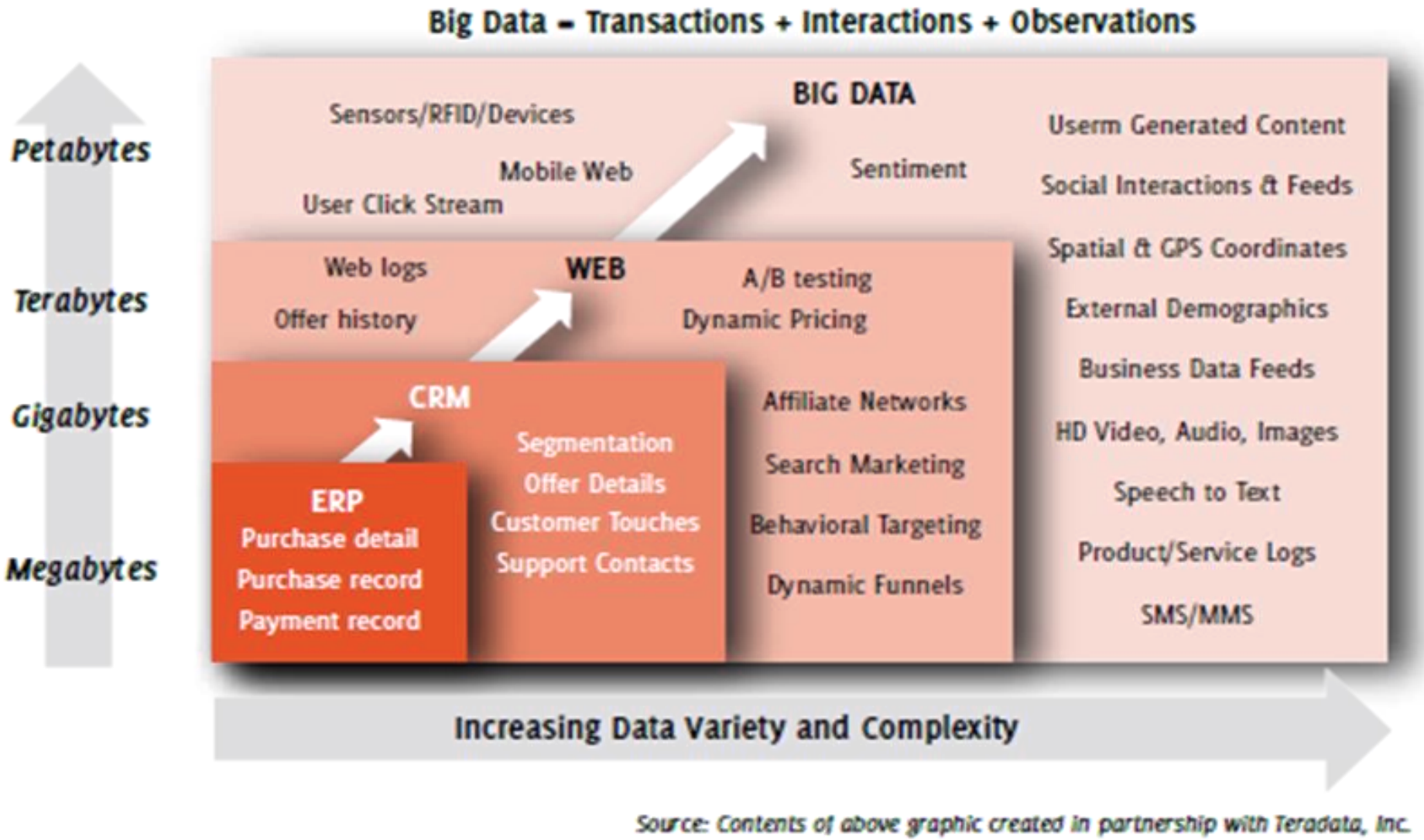
- Data comes from everywhere:
  - Sensors used to gather climate information
  - Posts to social media sites
  - Digital pictures
  - Videos posted online
  - Transaction records of online purchases and
  - Cell phone GPS signals to name a few.

The ‘Big Data’ Phenomenon!!!!

Big Data Drivers:

- The proliferation of data capture and creation technologies.
- Increased “interconnectedness” drives consumption (creating more data).
- Inexpensive storage makes it possible to keep more, longer.
- Innovative software and analysis tools turn data into information.

Examples of Size and Source



Source: Contents of above graphic created in partnership with Teradata, Inc.

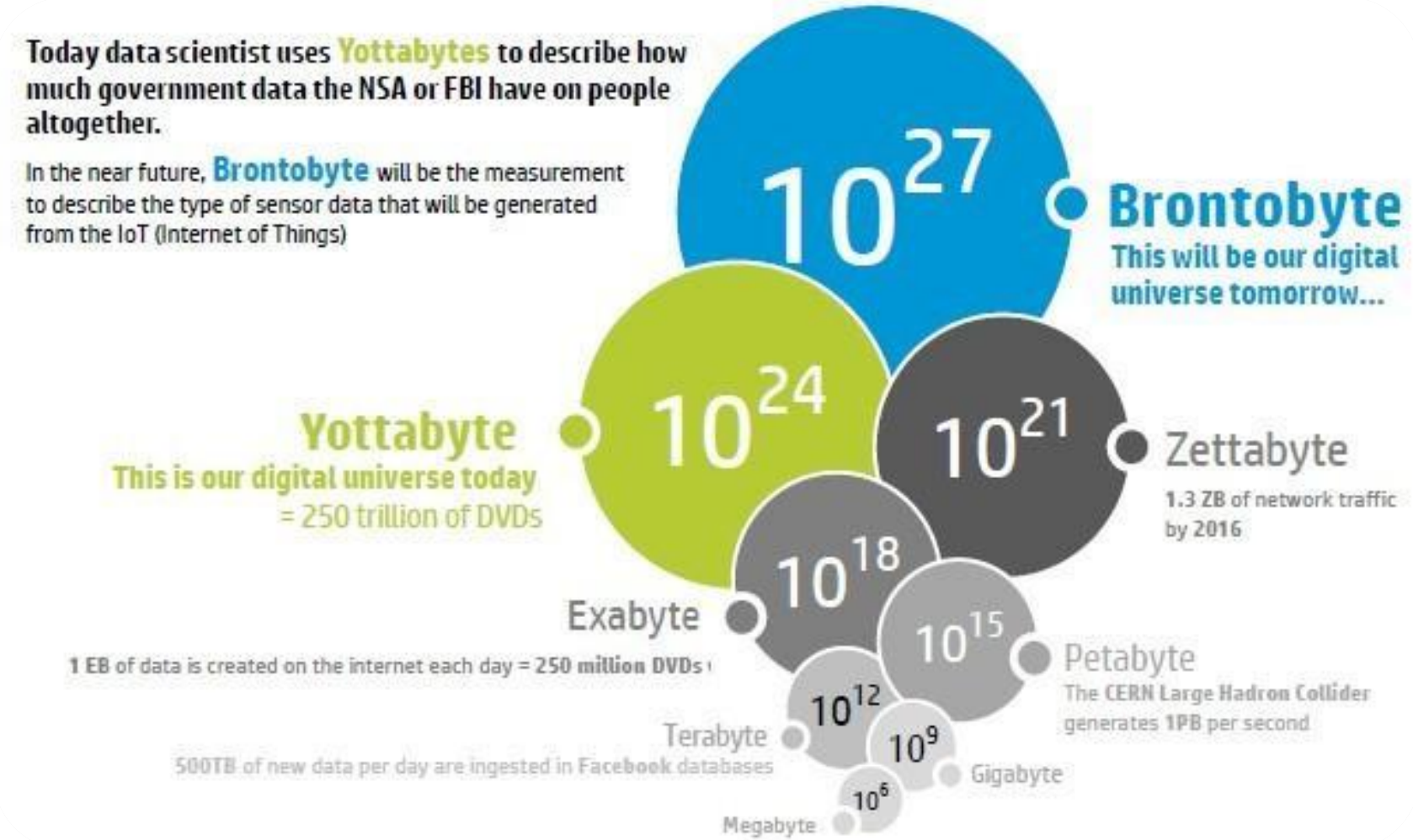
Source: http://hortonworks.com/blog/7-key-drivers-for-the-big-data-market/

Source: http://blog.vint.sogeti.com/wp-content/uploads/2012/07/VINT-Sogeti-on-Big-Data-1-of-4-Creating-Clarity.pdf

Data Size Math

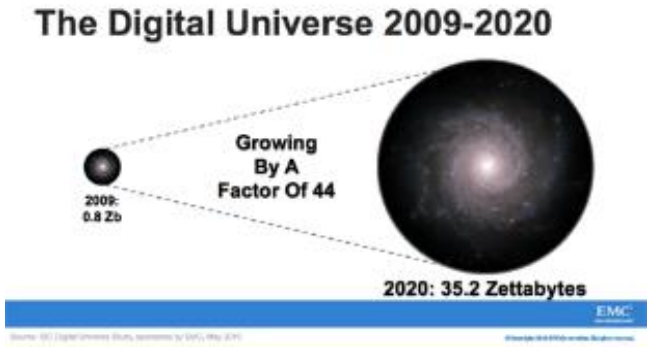
- Kilobyte (KB) – 10^3
- Megabyte (MB) – 10^6
- Gigabyte (GB) – 10^9
- Terabyte (TB) – 10^{12}
- Petabyte (PB) – 10^{15}
- Exabyte (EB) – 10^{18}
- Zettabyte (ZB) – 10^{21}
- Yottabyte (YB) – 10^{24}

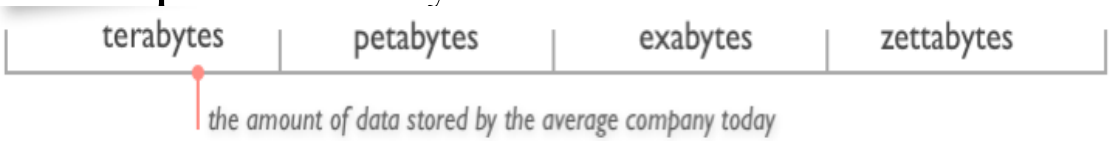
Brontobyte

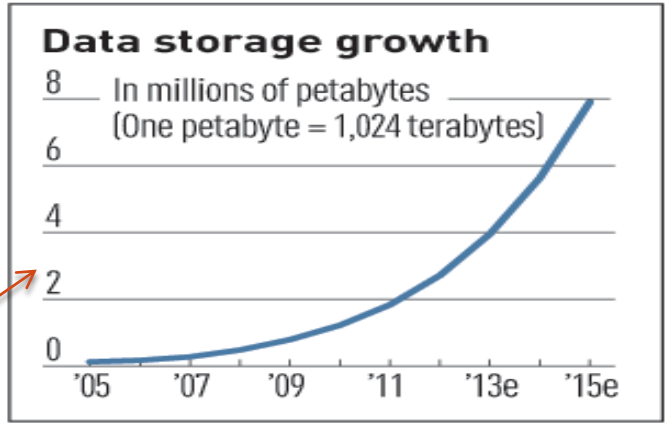

Today data scientist uses Yottabytes to describe how much government data the NSA or FBI have on people altogether.
In the near future, Brontobyte will be the measurement to describe the type of sensor data that will be generated from the IoT (Internet of Things)
10^27
Brontobyte
This will be our digital universe tomorrow...
10^24
Yottabyte
This is our digital universe today
= 250 trillion of DVDs
10^21
Zettabyte
1.3 ZB of network traffic by 2016
10^18
Exabyte
1 EB of data is created on the internet each day = 250 million DVDs
10^15
Petabyte
The CERN Large Hadron Collider generates 1PB per second
10^12
Terabyte
500TB of new data per day are ingested in Facebook databases
10^9
Gigabyte
10^6
Megabyte

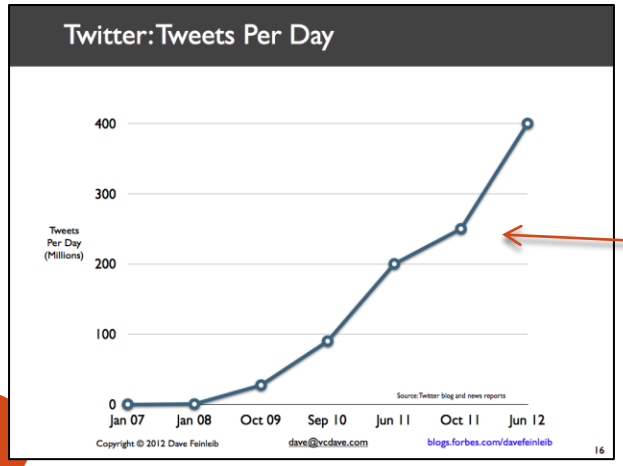
Characteristics of Big Data:
1-Scale (Volume)

- **Data Volume**
  - 44x increase from 2009 - 2020
  - From 0.8 zettabytes to 35zb
- Data volume is increasing exponentially.





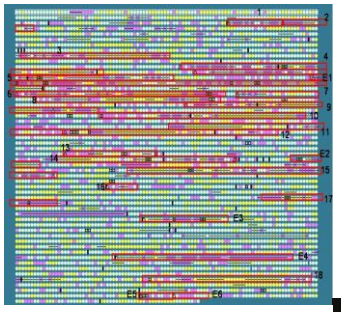


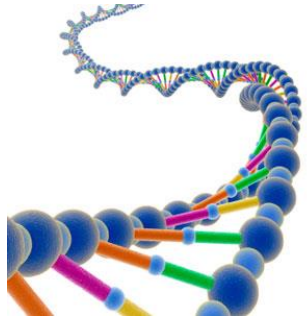


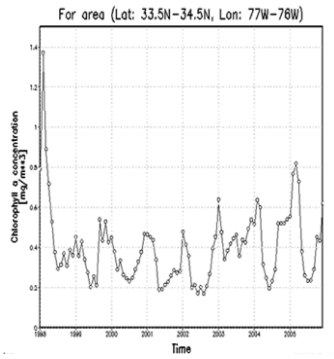
Exponential increase in collected/generated data

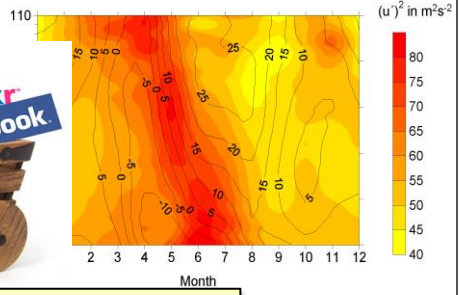
Characteristics of Big Data:
2-Complexity (Varity)

- Various formats, types, and structures.
  - *Text, numerical, images, audio, video, sequences, time series, social media data, multi-dim arrays, etc…*
- Static data vs. streaming data
- A single application can be generating/collecting many types of data.







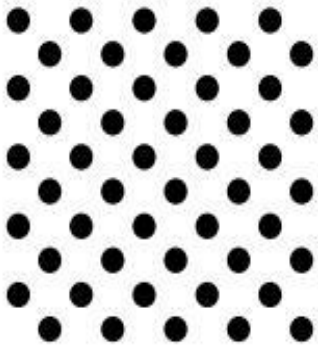
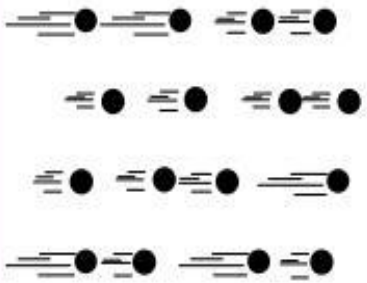
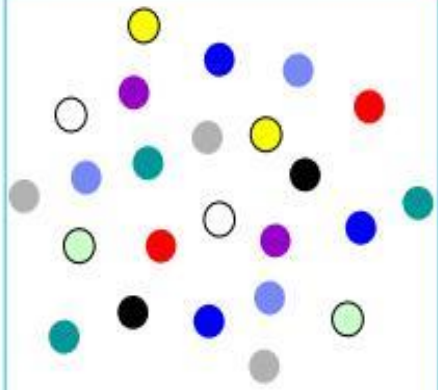
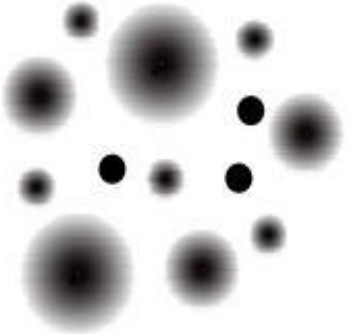


To extract knowledge➔ all these types of data need to linked together

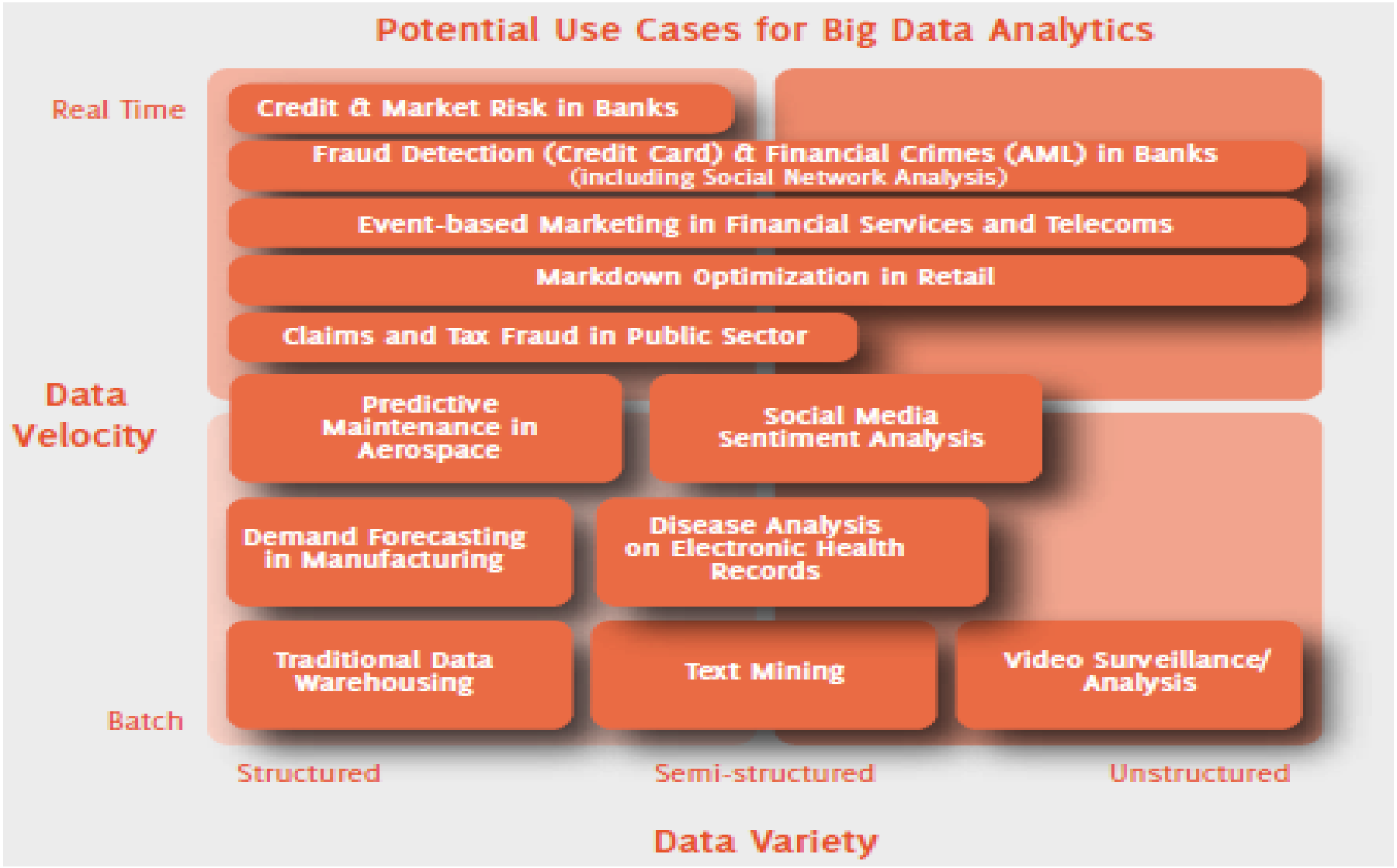
Characteristics of Big Data:
3-Speed (Velocity)

- Data is begin generated fast and need to be processed fast.
- Online Data Analytics.
- Late decisions ➔ missing opportunities
- Examples
  - E-Promotions: Based on your current location, your purchase history, what you like ➔ send promotions right now for store next to you.
  - Healthcare monitoring: sensors monitoring your activities and body ➔ any abnormal measurements require immediate reaction.

Some Make it 4V's

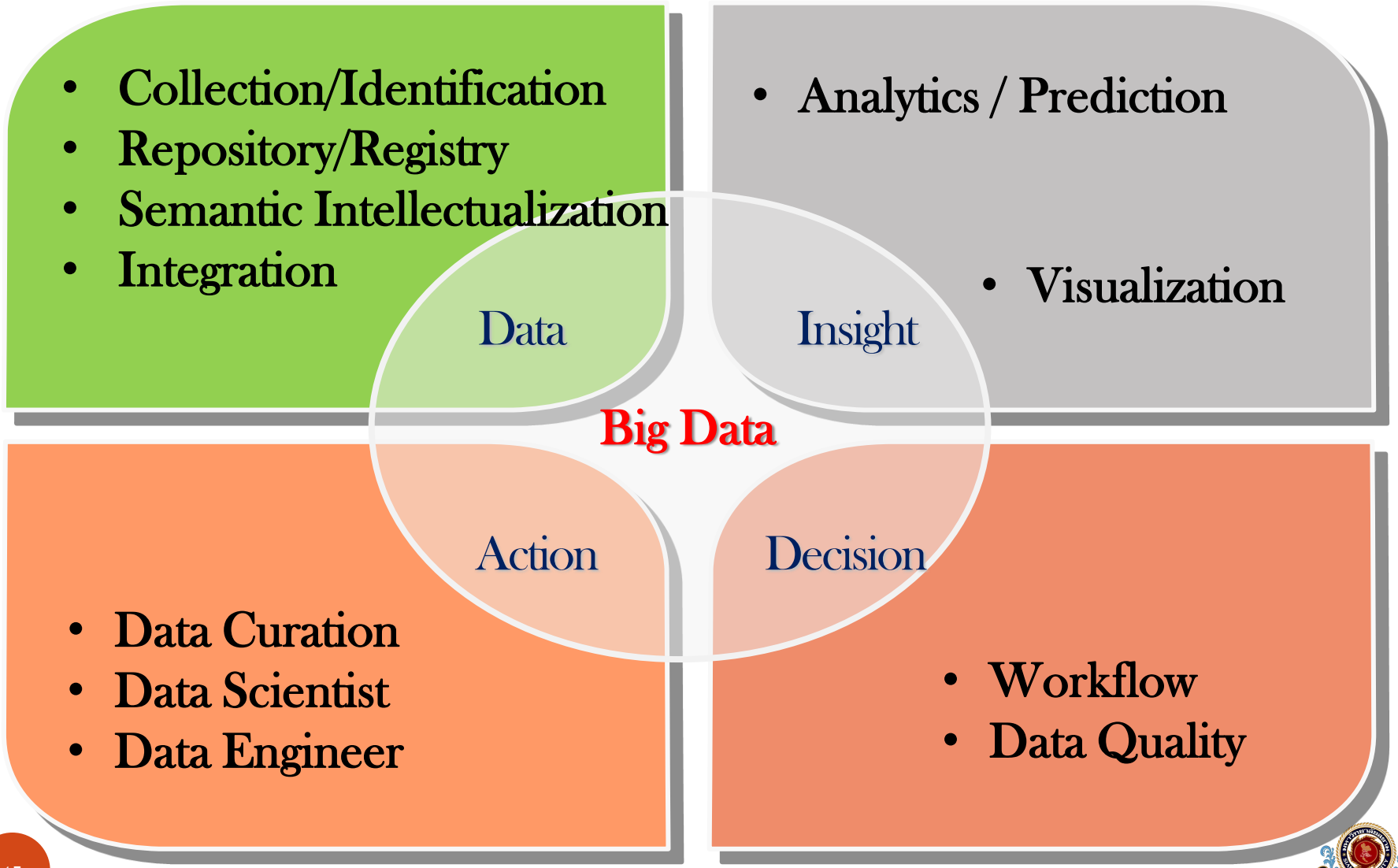
Volume	Velocity	Variety	Veracity*
			
Data at Rest	**Data in Motion**	**Data in Many Forms**	**Data in Doubt**
Terabytes to exabytes of existing data to process	Streaming data, milliseconds to seconds to respond	Structured, unstructured, text, multimedia	Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

Examples of Big Data Analytics



Source: http://blog.vint.sogeti.com/wp-content/uploads/2012/07/VINT-Sogeti-on-Big-Data-1-of-4-Creating-Clarity.pdf page 18

A lifecycle of Big Data



Big Data Problems

- Efficiently storing and accessing large amounts of **data is difficult**. The additional demands of fault tolerance and backups makes things event more complicated.
- Manipulating large data sets involves running immensely parallel processes. Gracefully recovering from any failures during such a run and providing results in a reasonably short period of **time is complex**.
- Managing the continuously evolving schema and metadata for semi-structured and un-structured data, generated by **diverse sources, is a convoluted problem.**

Hadoop

The Origin of the Name "Hadoop"

- The name **Hadoop** is not an acronym; it's a made-up name. The project's creator, **Doug Cutting**, explains how the name came about:

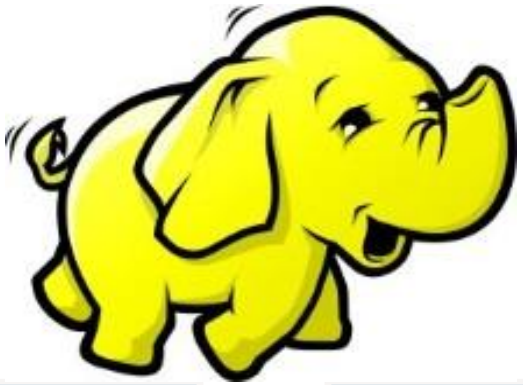
  "The name my kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere: "those are naming criteria.

What is Apache Hadoop?

Apache Hadoop is a platform for data storage and processing that is...

- ✓Scalable
- ✓Fault tolerant
- ✓Open source

Core Hadoop Components



Hadoop Distributed File System (HDFS) + MapReduce

Distributed File System - distributes data
Map/Reduce - distributes application

Distributed File System

- Designed to store large files
- Stores files as large blocks (64 to 256 MB)
- Each block stored on multiple servers
- Data is automatically re-replicated on need
- Accessed from command line, Java API, or C API

```
bin/hadoop fs -put my-file hdfs://node1:50070/foo/bar
Path p = new Path("hdfs://node1:50070/foo/bar");
```


Map-Reduce

- Map-Reduce is a programming model for efficient distributed computing
- It works like a Unix pipeline:

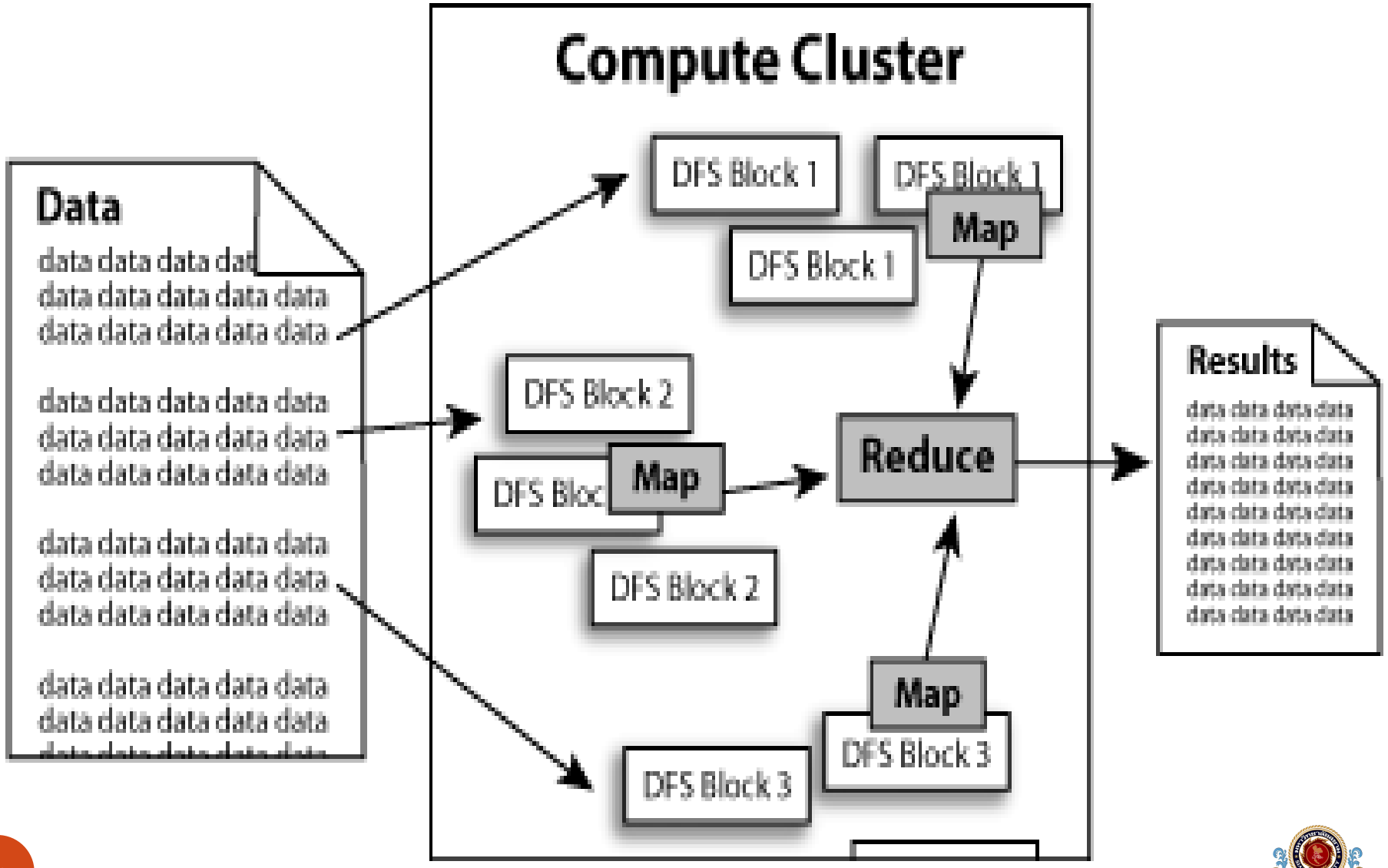
  cat input | grep | sort | unique -c | cat > output

  Input | Map | Shuffle & Sort | Reduce | Output
- Efficiency from
  - Streaming through data, reducing seeks
  - Pipelining
- A good fit for a lot of applications
  - Log processing
  - Web index building

Hadoop Cluster

- A set of machines running HDFS and MapReduce is known as a Hadoop Cluster.
  - Individual machines are known as *nodes*.
  - A cluster can have as few as one node, as many as several thousands.
    - More nodes = better performance!

Clusters of Computers



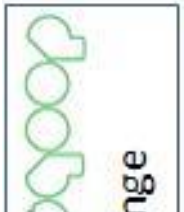
Hadoop: Assumptions

It is written with large clusters of computers in mind and is built around the following assumptions:

- Processing will be run in **batches**.
- Applications that run on HDFS have large **data sets.**
- A typical file in HDFS is **gigabytes** to **terabytes** in size.
- It should provide high aggregate data bandwidth and scale to hundreds of nodes in a single cluster. It should support **tens of millions of files** in a single instance.
- Applications need a **write-once-read-many** access model.
- Moving Computation is **Cheaper than Moving Data.**

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

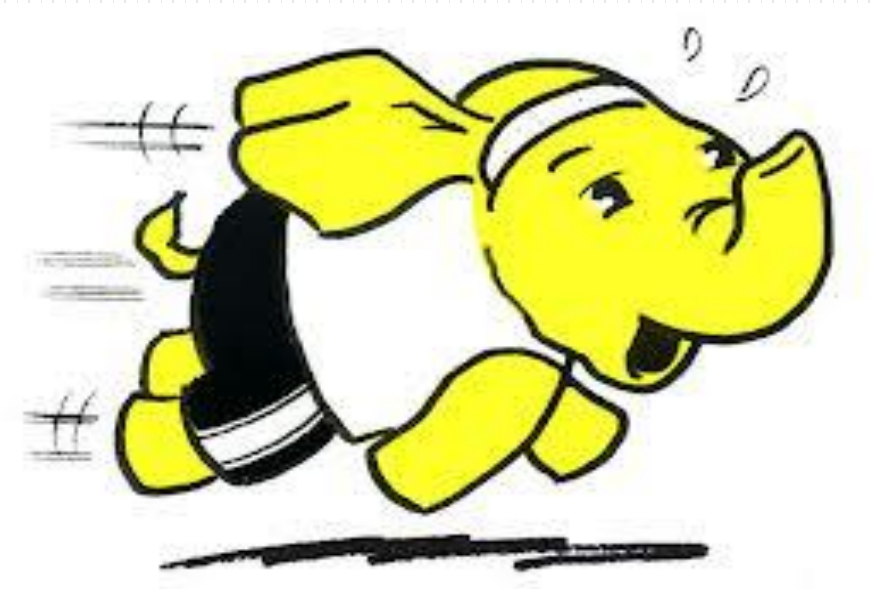
Hive SQL Query

Hbase Columnar Store

YARN Map Reduce v2
Distributed Processing Framework

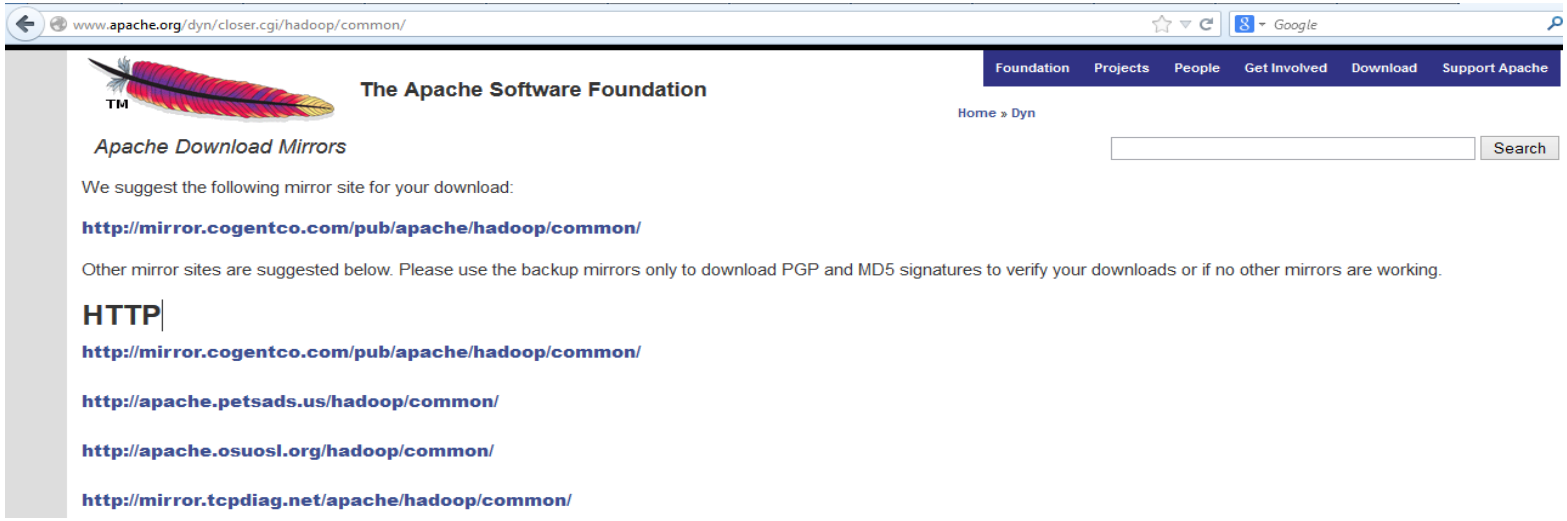
HDFS
Hadoop Distributed File System

Getting Started Hadoop-1



Download Hadoop

- http://hadoop.apache.org/releases.html#Download
- Version
  - **1.2.X** - current stable version, 1.2 release
  - **2.X.X** - current beta version
  - **0.22.X** - does not include security
  - **0.20.X** - old legacy version
- http://www.apache.org/dyn/closer.cgi/hadoop/common/



Prerequisites

- **Supported Platforms**
  - Linux, Mac OS/X, Windows, and Solaris
  - Commodity hardware
- **Required Software**
  - Java™ 1.6.x, preferably from Sun, must be installed.
    - http://www.eclipse.org/downloads/
  - **ssh** must be installed and **sshd** must be running to use the Hadoop scripts that manage remote Hadoop daemons.

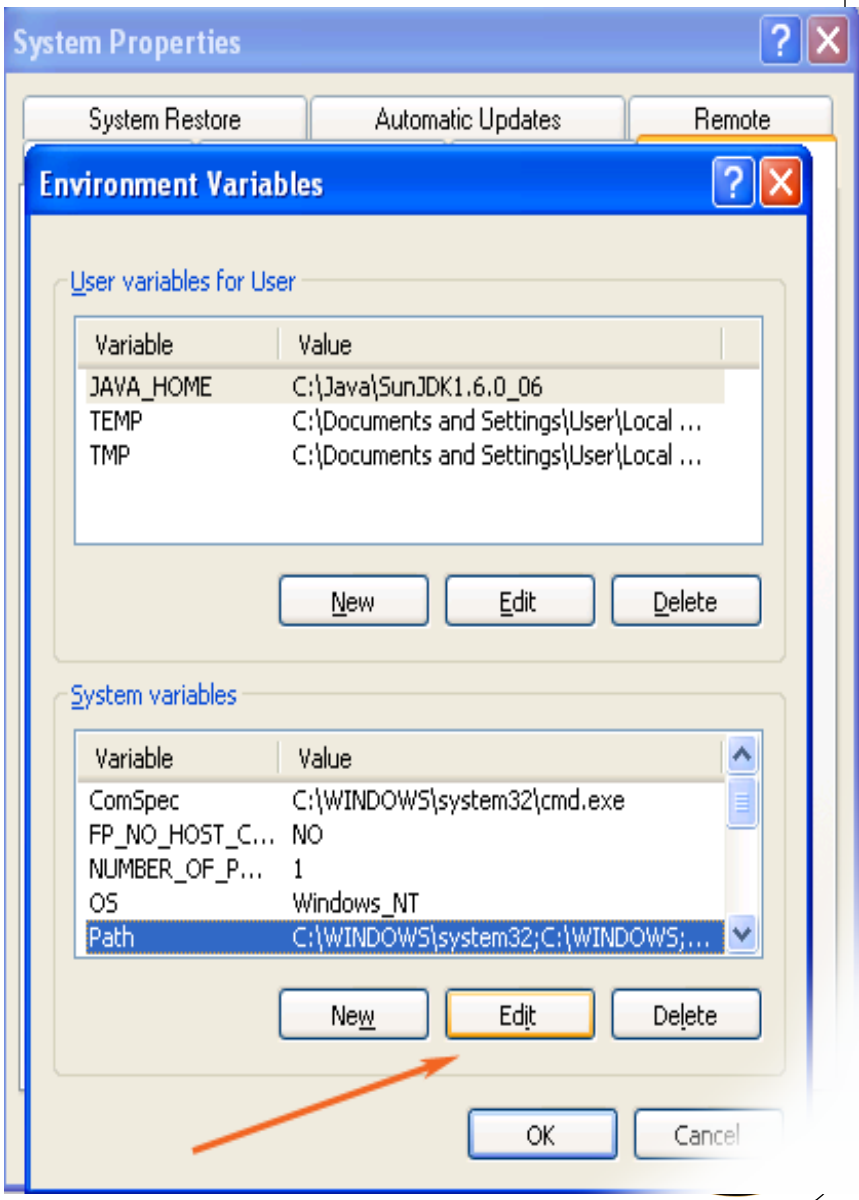
Cygwin

- Additional requirements for Windows include:
  - Cygwin - Required for shell support in addition to the required software above.
  - http://www.cygwin.com/

Set Environment Variables

- JAVA
  - JAVA_HOME
  - C:\Java\SunJDK1.6.0_06
- Cygwin
  - System > Edit Path

    ;c:\cygwin\bin;c:\cygwin\usr\bin



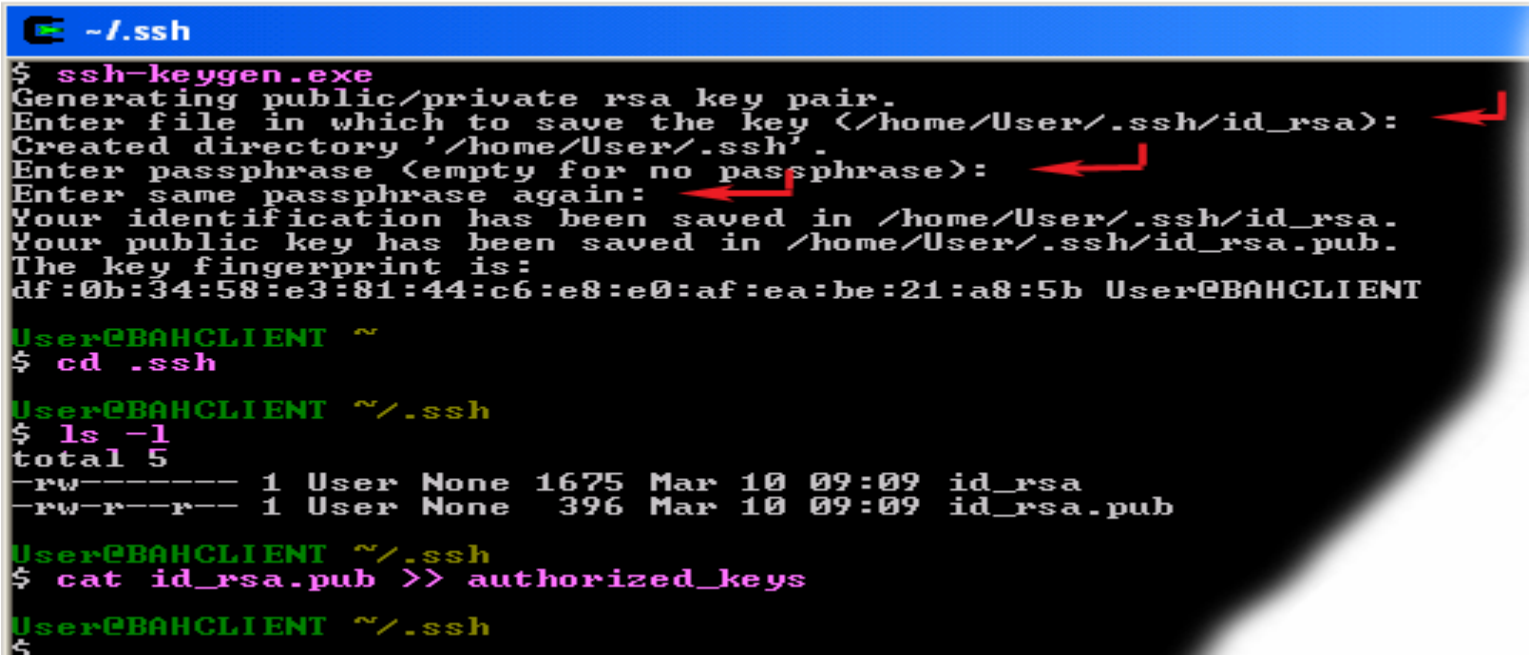
Setup SSH daemon

- ssh-host-config

```
User@BAHCLIENT ~
$ ssh-host-config
Generating /etc/ssh_host_key
Generating /etc/ssh_host_rsa_key
Generating /etc/ssh_host_dsa_key
Generating /etc/ssh_config file
Privilege separation is set to yes by default since OpenSSH 3.3.
However, this requires a non-privileged account called 'sshd'.
For more info on privilege separation read /usr/share/doc/openssh/README.privse
.

Should privilege separation be used? (yes/no) no
Generating /etc/sshd_config file
Added ssh to C:\WINDOWS\system32\drivers\etc\services


Warning: The following functions require administrator privileges!

Do you want to install sshd as service?
(Say "no" if it's already installed as service) (yes/no) yes

Which value should the environment variable CYGWIN have when
sshd starts? It's recommended to set at least "ntsec" to be
able to change user context without password.
Default is "ntsec".  CYGWIN=ntsec

The service has been installed under LocalSystem account.
To start the service, call `net start sshd' or `cygrunsrv -S sshd'.

Host configuration finished. Have fun!
```

Setup Authorization Keys

- ssh-keygen
- cd ~/.ssh
- cat id_rsa.pub >> authorized_keys
- ssh localhost



HDFS

Hadoop Distributed File System (HDFS)

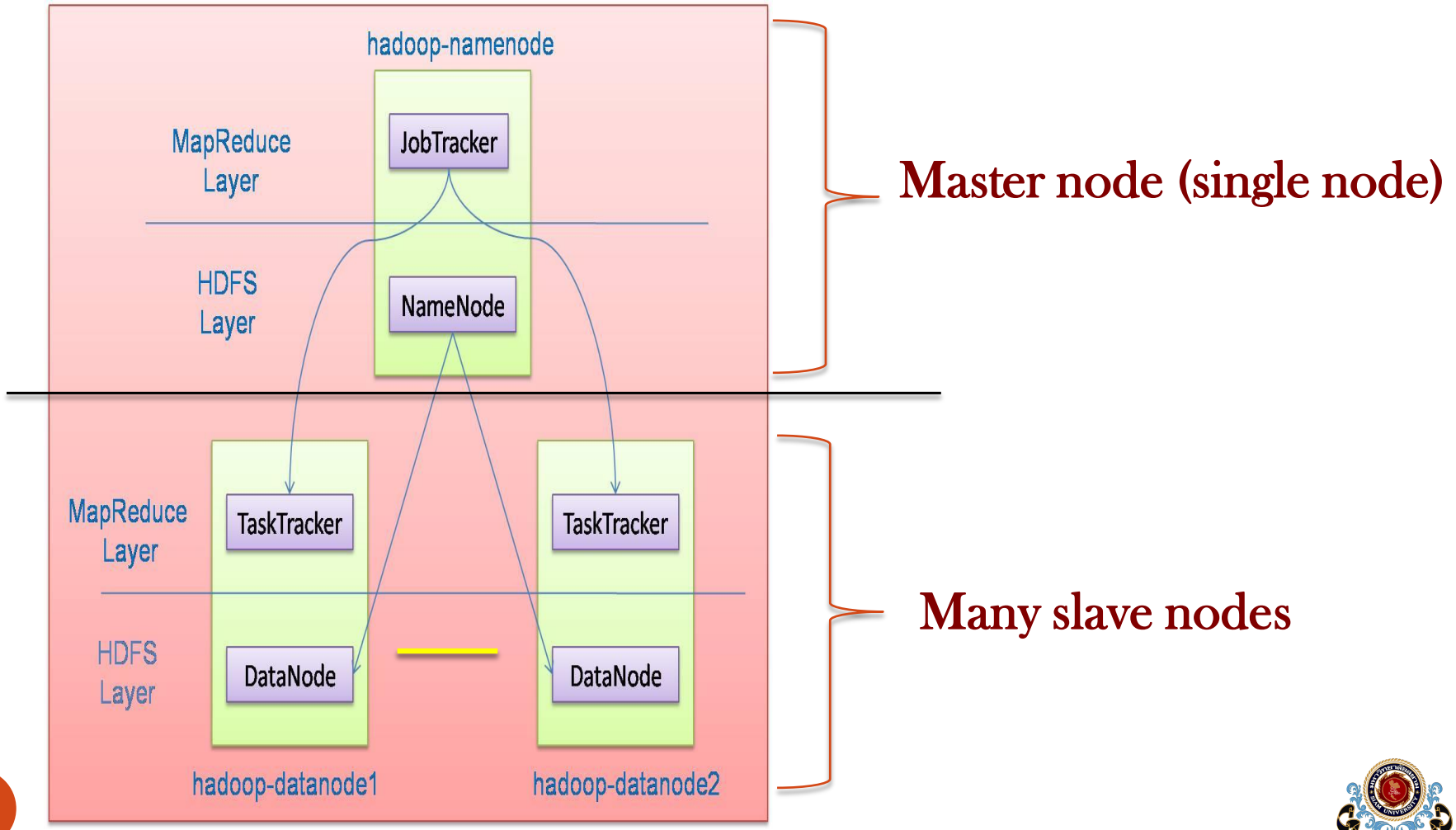
What is HDFS*?

- The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware.
  - It has many similarities with existing distributed file systems.
- Data in a **Hadoop** cluster is broken down into smaller pieces (called blocks) and distributed throughout the cluster.
- HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

* http://www-01.ibm.com/software/data/infosphere/hadoop/hdfs/

Hadoop Master/Slave Architecture

- Hadoop is designed as a **master-slave shared-nothing** architecture

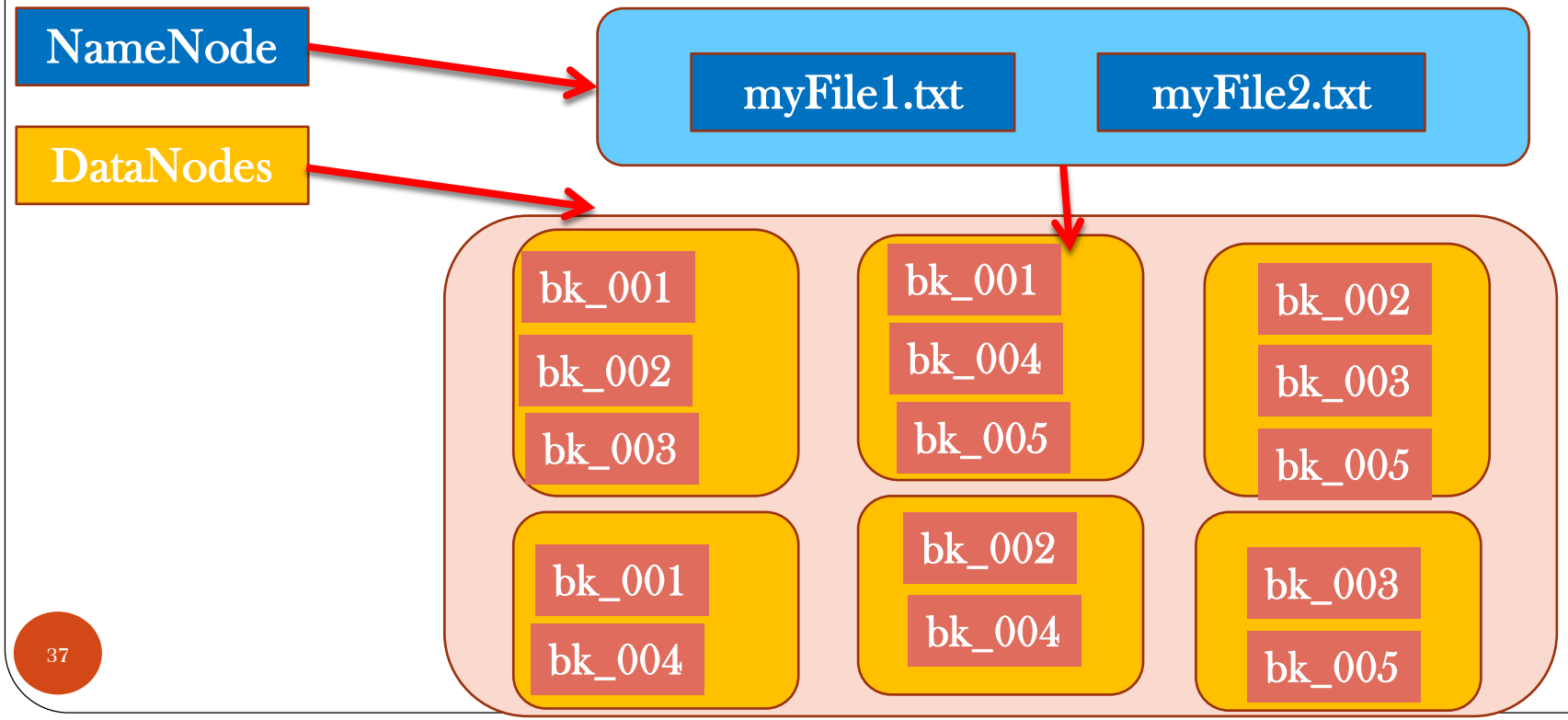


Hadoop Server Role

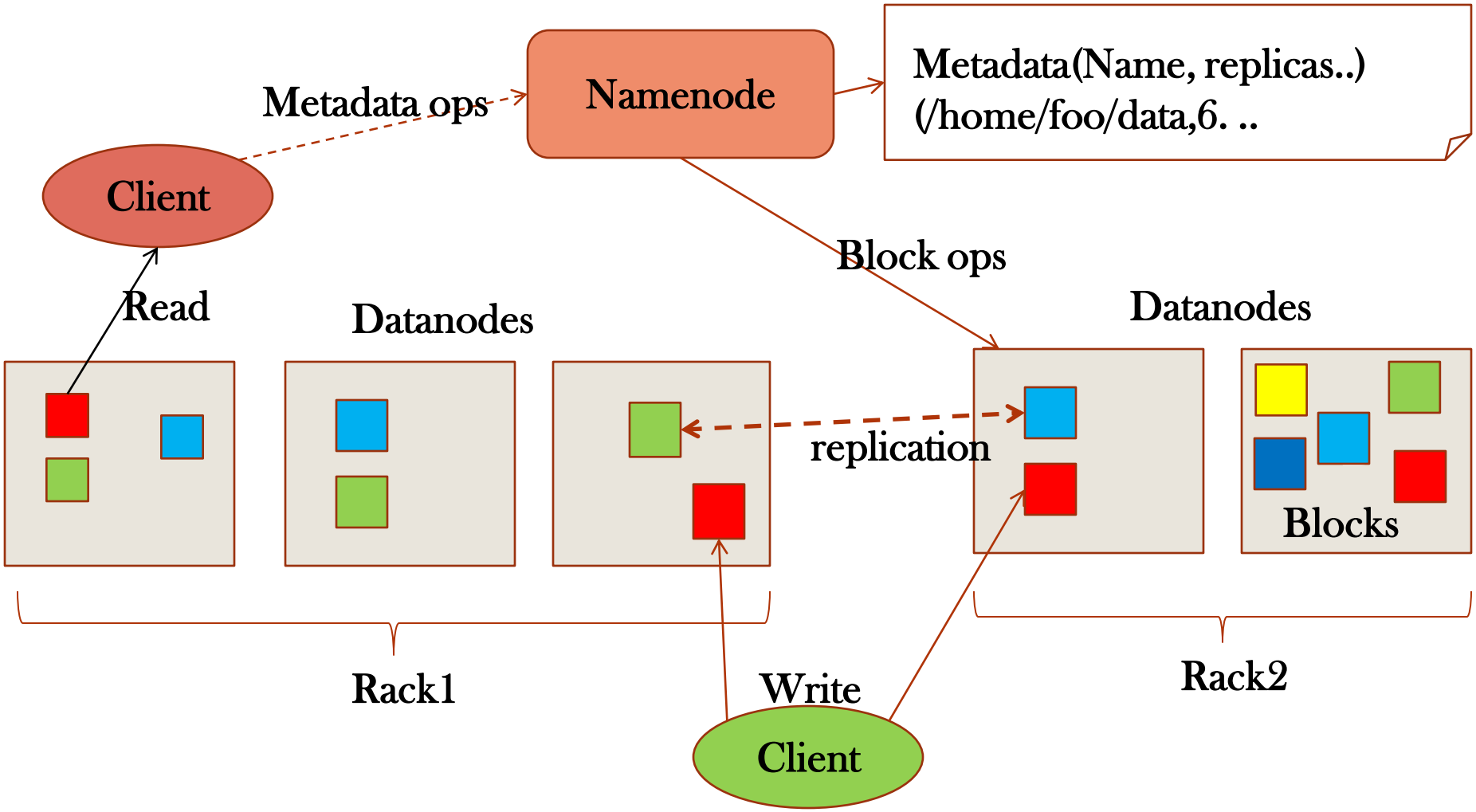
- NameNode
  - An HDFS cluster consists of a single NameNode which manages the file system namespace and regulates access to files by clients.
- JobTracker
  - The JobTracker daemon is the liaison between the application and Hadoop.
- SecondaryNameNode (SNN)
  - SSN is an assistant daemon for monitoring the state of the HDFS cluster.
- DataNodes
  - Each slave machine in the cluster will host a DataNode daemon to perform the grunt work of the distributed file system for both reading and writing.
- TaskTracker
  - manages the execution of individual tasks on each slave node.

NameNode and DataNodes

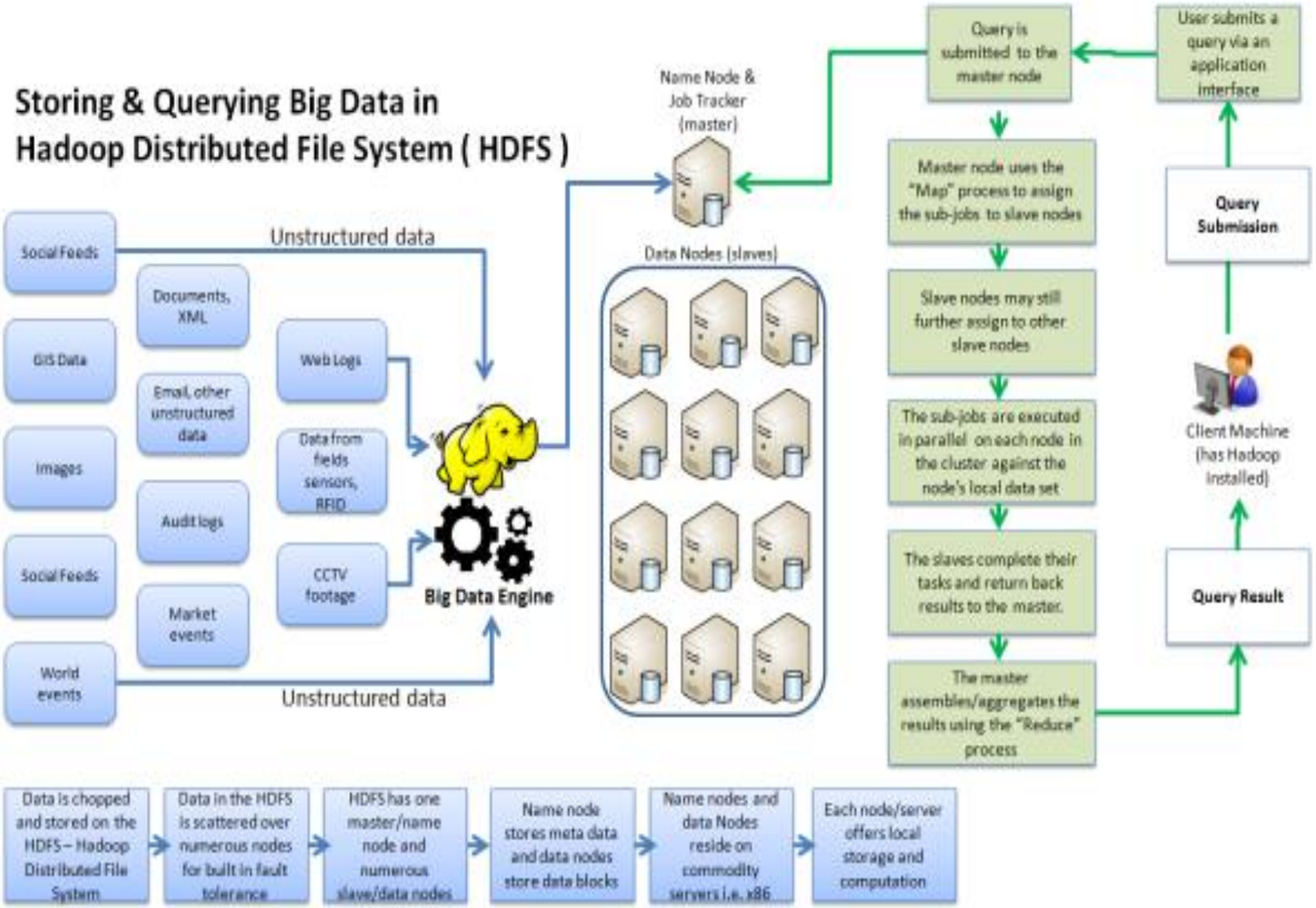
- Example:
  - NameNode holds metadata for the two files
    - myFile1.txt : bk_001, bk_002, bk_003
    - myFile2.txt: bk_004, bk_005
  - DataNodes hold the actual blocks 64 – 256 MB size.



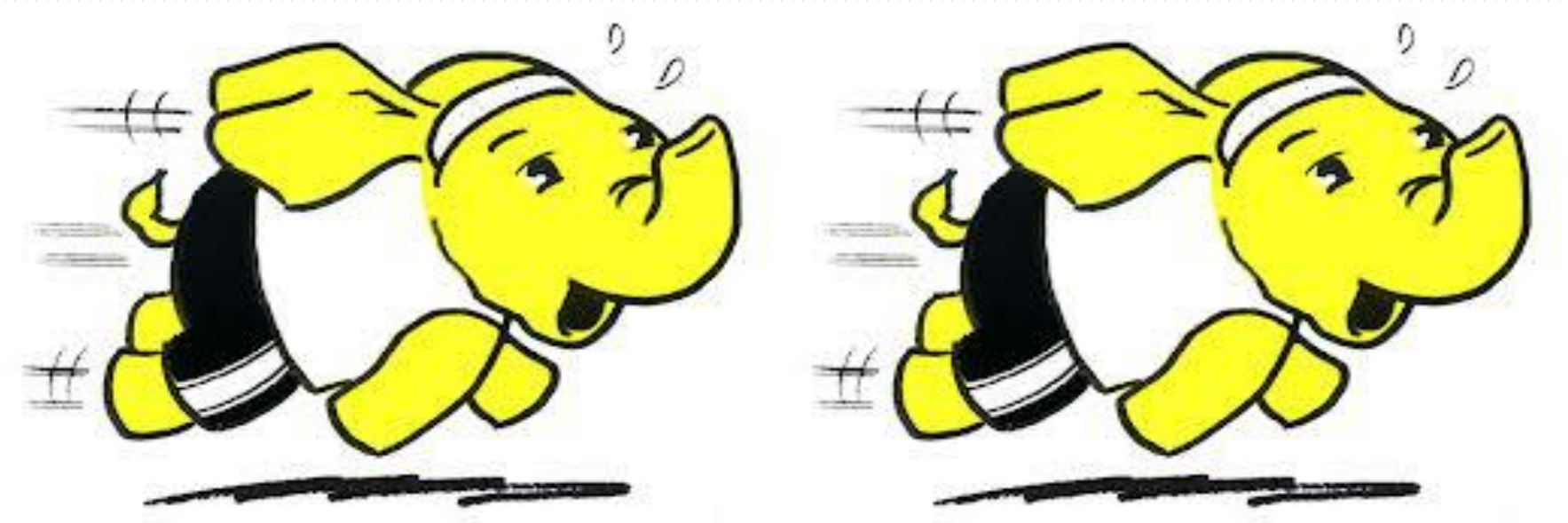
HDFS Architecture



Storing & Querying Big Data in Hadoop Distributed File System (HDFS)



Getting Started Hadoop-2

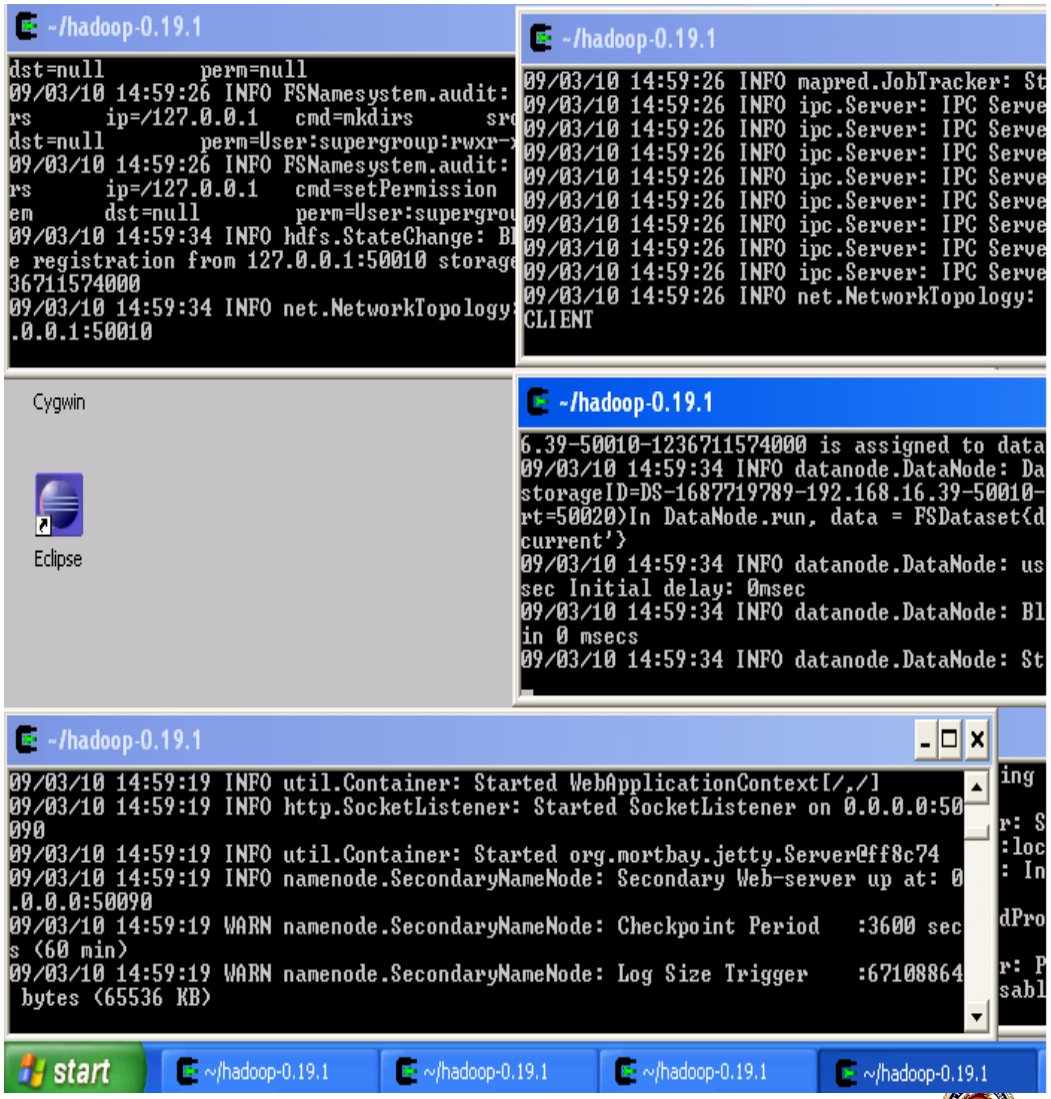


Configure the hosts file (All machines)

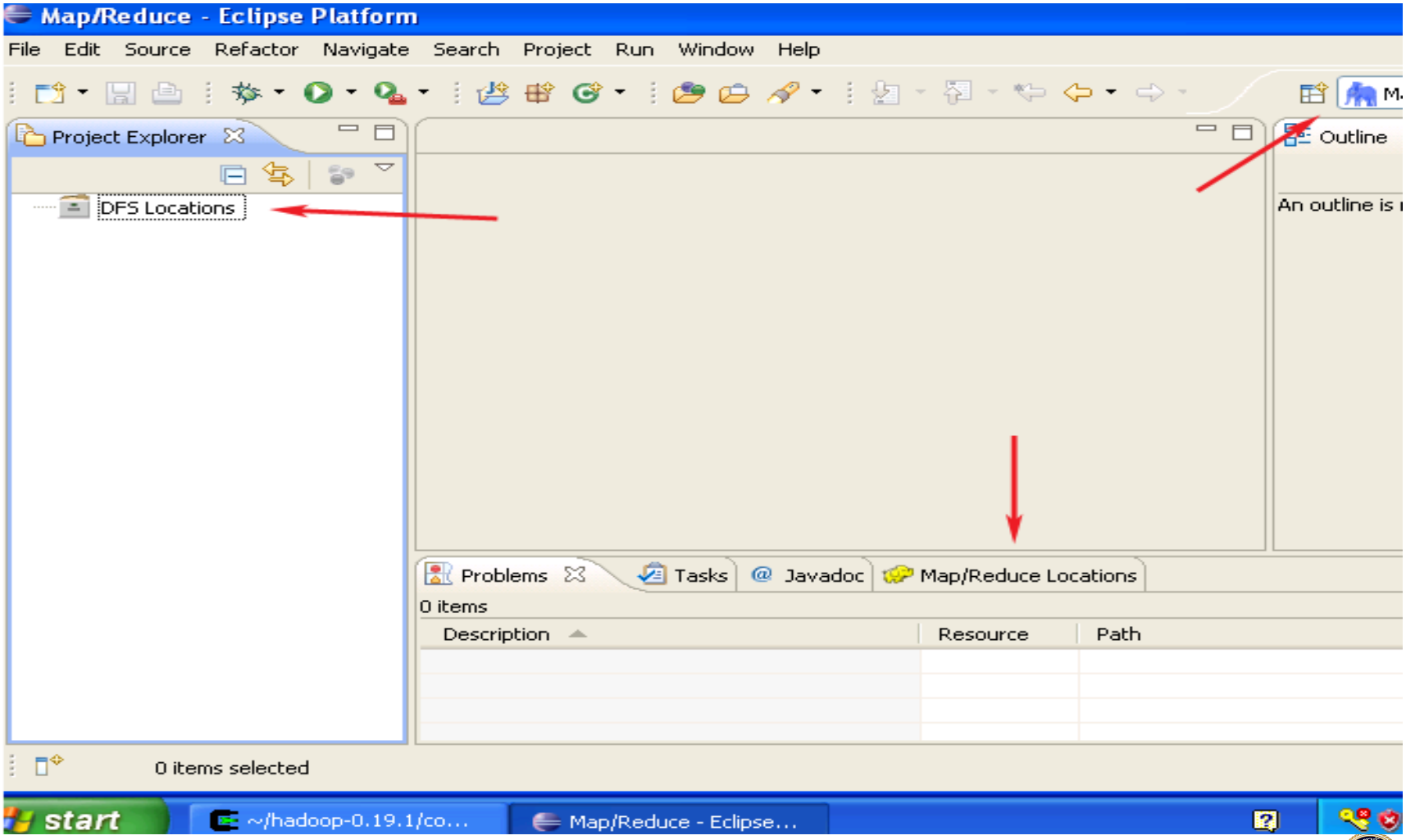
- Open the Windows hosts file located at;
  c:\windows\system32\drivers\etc\hosts
  master NNN.NNN.NNN.NNN
  slave NNN.NNN.NNN.NNN
- Install Cygwin and Configure OpenSSH
- Extract Hadoop
  tar -xzf hadoop-0.19.1.tar.gz
  ls -l

Start the local Hadoop Cluster

- Start NameNode
  - cd hadoop-0.19.1
    bin/hadoop namenode
- Start JobTracker
  - cd hadoop-0.19.1
    bin/haoop jobtracker
- Start DataNode
  - cd hadoop-0.19.1
    bin/haoop datanode
- Start TaskTracker
  - cd hadoop-0.19.1
    bin/haoop tasktracker



Run Eclipse


Map/Reduce - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
DFS Locations
Outline
An outline is
Problems
Tasks
Javadoc
Map/Reduce Locations
0 items
Description
Resource
Path
0 items selected
start
~/hadoop-0.19.1/co...
Map/Reduce - Eclipse...

MapReduce

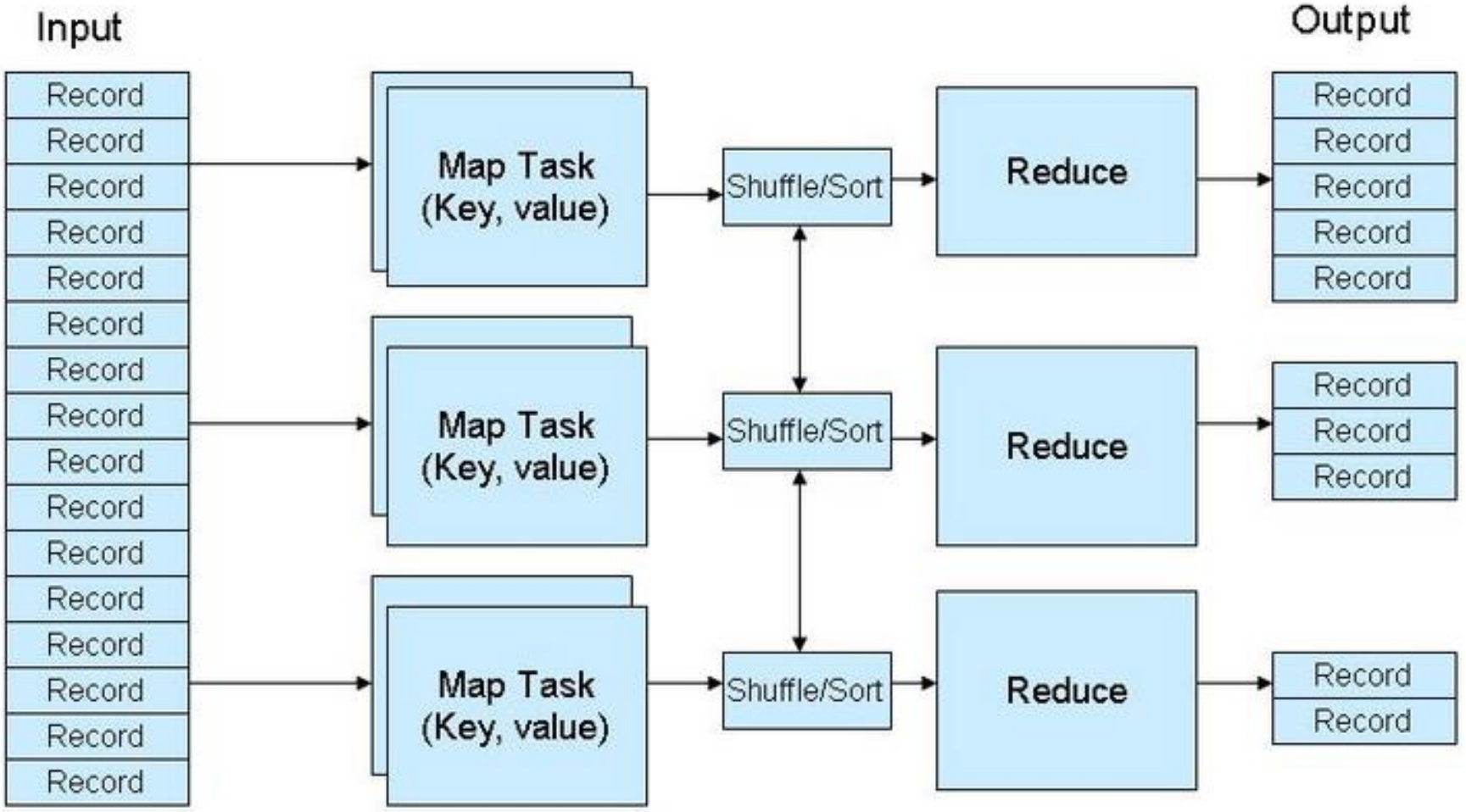
What is MapReduce?

- MapReduce framework, the developer has to specify based on two simple functions as **map function** and a **reduce function**.
- The map function is invoked with a **key/value-pair for each data record** and **returns intermediate key/value pairs**, which are used by the reduce function to merge all intermediate values with the same intermediate key.
  - **Map(key, value) → list(ikey, ivalue)**
  - **Reduce(iKey, list(ivalue) → list(fvalue)**
- Thereby
  - (key, value) is one record of input data.
  - (ikey, ivalue) an intermediate key/value pair.
  - Fvalue a final result value.

Map & Reduce Functions

- **The Mapper:**
  - Input: Read input from disk.
  - Output: Create a single or multiple pairs of data <key, value>, known as intermediate pairs
- **The Reducer:**
  - Input: a list of <key,value> pairs with a unique key.
  - Output: Single or multiple of <key, values>

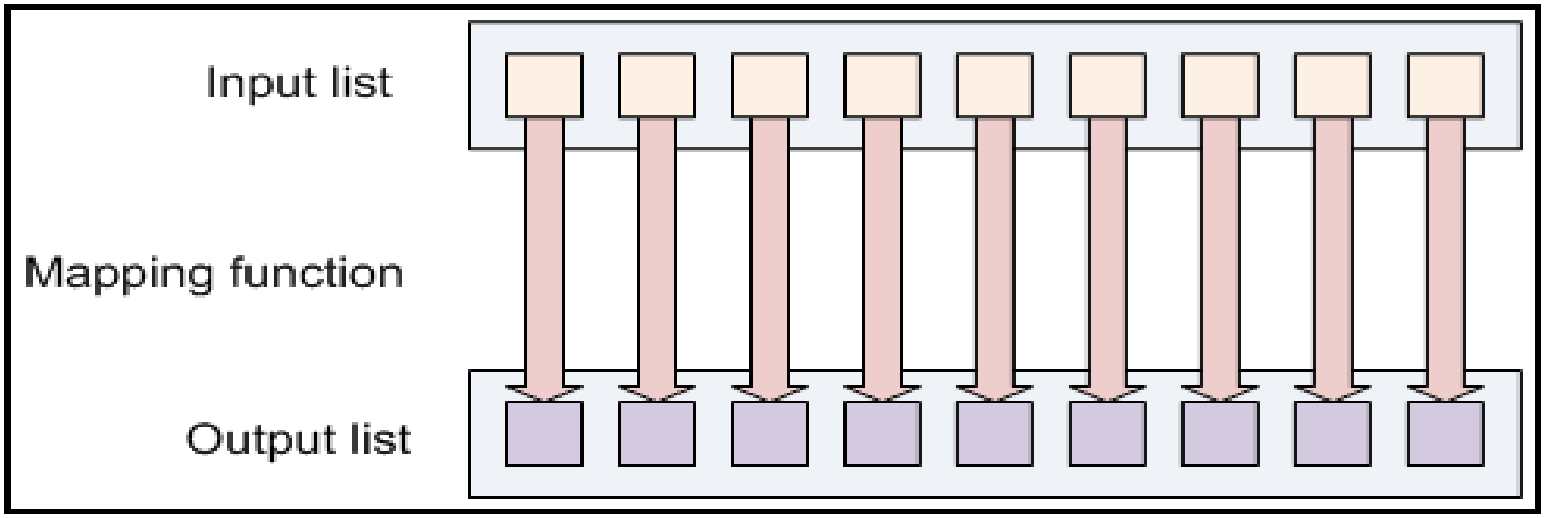
MapReduc Architecture*



*http://mapreduce-deepak4278.blogspot.com/2010/08/hdfs-and-map-reduce-architecture.html

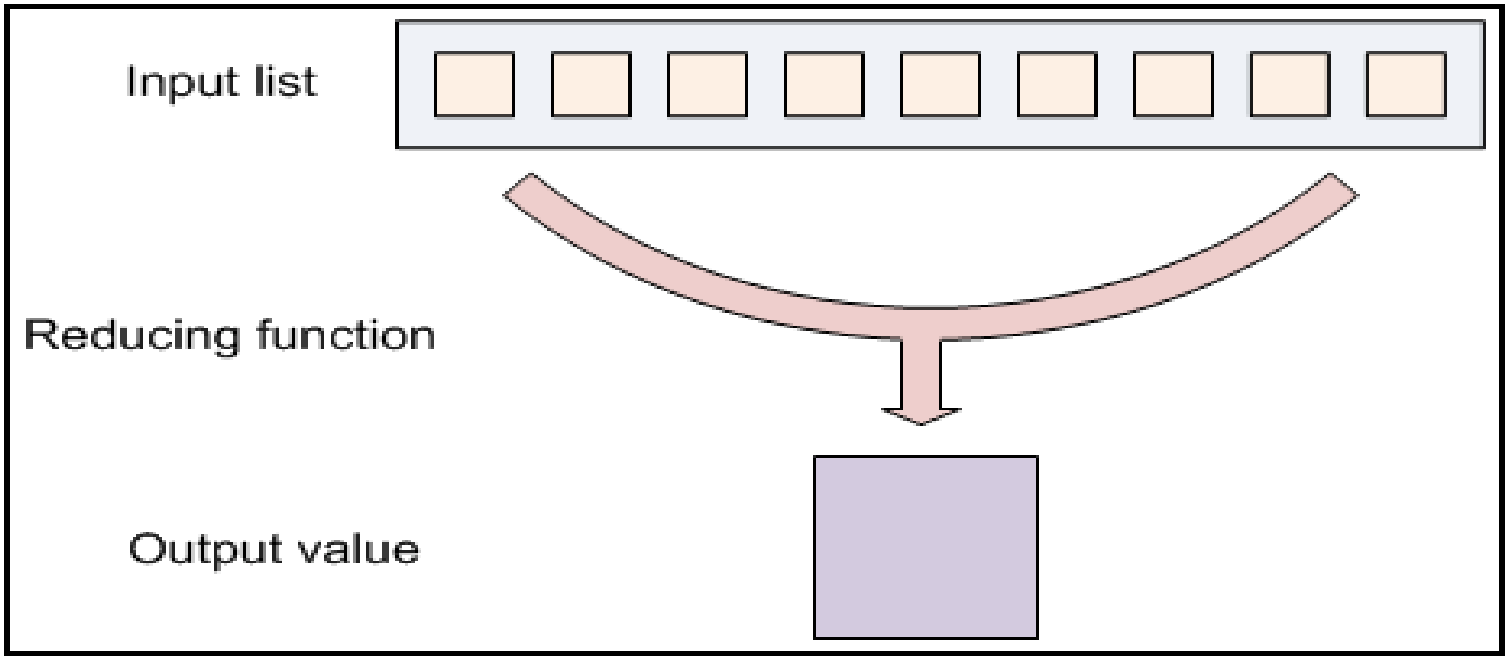
Mapping Lists

- The first phase of a MapReduce program is called ***mapping***.
- A list of data elements are provided, one at a time, to a function called the ***Mapper***, which transforms each element individually to an output data element.

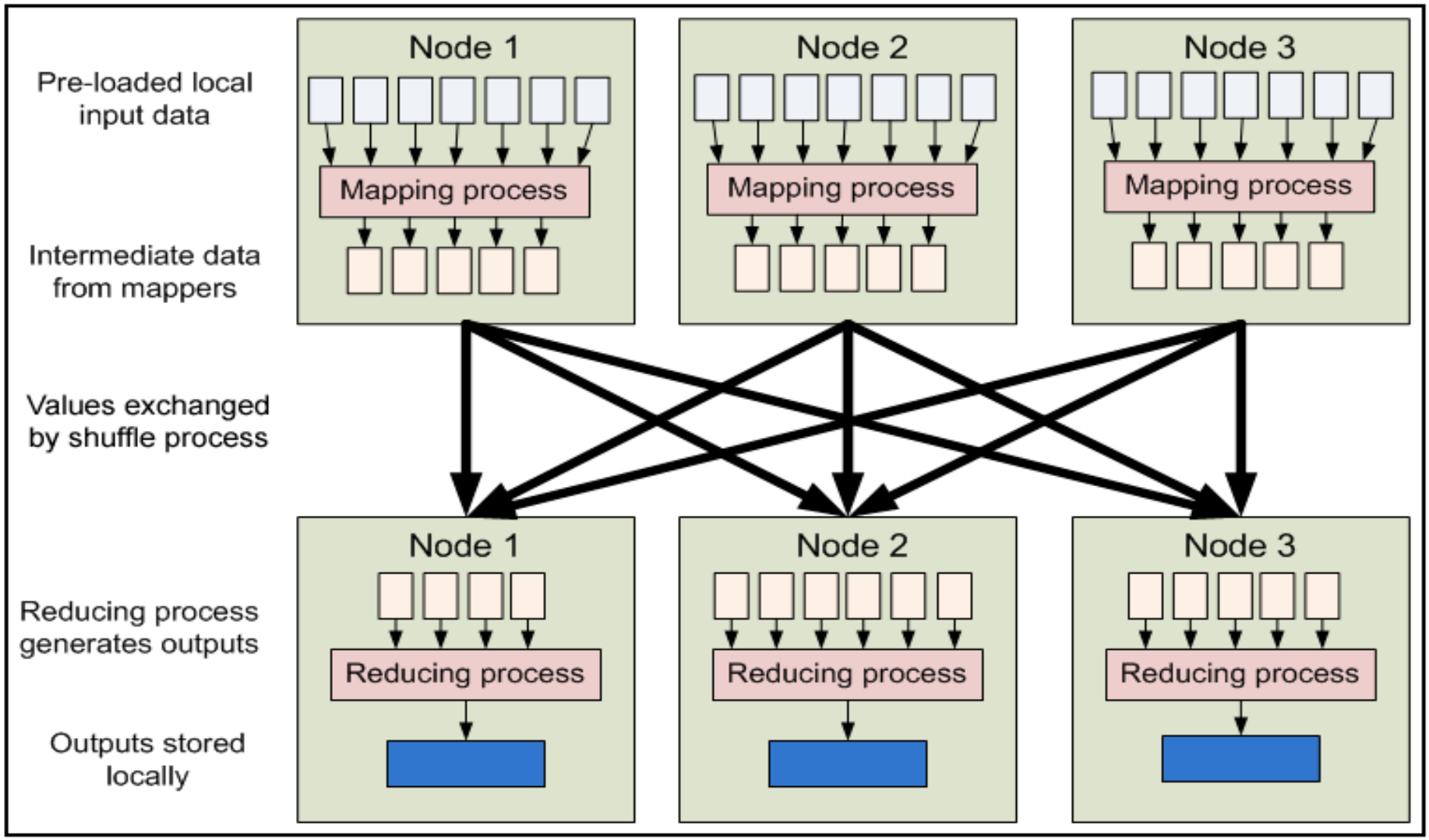


Reducing Lists

- Reducing lets you aggregate values together.
- A *reducer function* receives an iterator of input values from an input list. It then combines these values together, returning a single output value.



MapReduce Data Flow



MapReduce Types and Formats

- Input/Output Types
  - TextInputFormat (default)
  - TextObject:
    - On the top of the Crumpetty Tree
    - The Quangle Wangle sat,
    - But his face you could not see.
  - Key-Value Pairs:
    - (0, On the top of the Crumpetty Tree)
    - (31, The Quangle Wangle sat,)
    - (55, But his face you could not see.)
  - XML
  - Binary
  - Multiple
  - Database

Parameters Varies in Experiments

Parameter Name	Description and Use	Default Value	Values Considered
mapred.reduce.tasks	Number of reducer tasks	1	[5,300]
io.sort.factor	Number of sorted streams to merge at once during sorting	10	[10,500]
io.sort.mb	Size in MegaBytes of map-side buffer for sorting	100	[100,200]
io.sort.record.percent	Fraction of io.sort.mb dedicated to metadata storage	0.05	[0.05,0.15]
io.fi le.buffer.size	Buffer size used to read/write (intermediate) sequence fi les	4K	32K
mapred.child.java.opts	Java control options for all mapper and reducer tasks	-Xmx200m	-Xmx[200m,300m]
mapred.inmem.merge.threshold	Reduce-side trigger for in-memory merging; off when 0	1000	0
mapred.job.shuffle.input.buffer.percent	% of reducer task's heap to buffer map outputs	0.7	{0.7,0.8}
mapred.job.shuffle.merge.percent	Usage threshold of mapred.job.shuffle.input.buffer.percent to trigger reduce-side merge in parallel with the copying of map outputs	0.66	{0.66,0.8}
mapred.job.reduce.input.buffer.percent	% of reducer task's heap to buffer map outputs while applying reduce	0	{0,0.8}
dfs.replication	Block replication factor in Hadoop's HDFS fi lesystem	3	2
dfs.block.size	HDFS block size (equal to amount of data processed per mapper task)	64MB	128MB

Parameters Affecting Performance

- **dfs.block.size:** File system block size - Default: 67108864 (bytes)
- Example; Input data size = 160 GB and
- **dfs.block.size** = 64 MB then
  - The minimum no. of maps= **xxx** maps.
- If **dfs.block.size** = 128 MB
  - The minimum no. of maps= **xxx** maps.
- If **dfs.block.size** = 256 MB
  - The minimum no. of maps= **xxx** maps.

How MapReduce Work - Application

1. Partition input file into **M** partitions.
2. Create **M** Map tasks, read **M** partitions in parallel and create intermediate pairs. Store them into local storage.
3. Wait for all Map workers to finish, and partition intermediate pairs into **R** regions.
4. Start **R** reduce workers, each reads a list of intermediate with a unique key from remote disks.
5. Write the output of reduce workers to file(s).

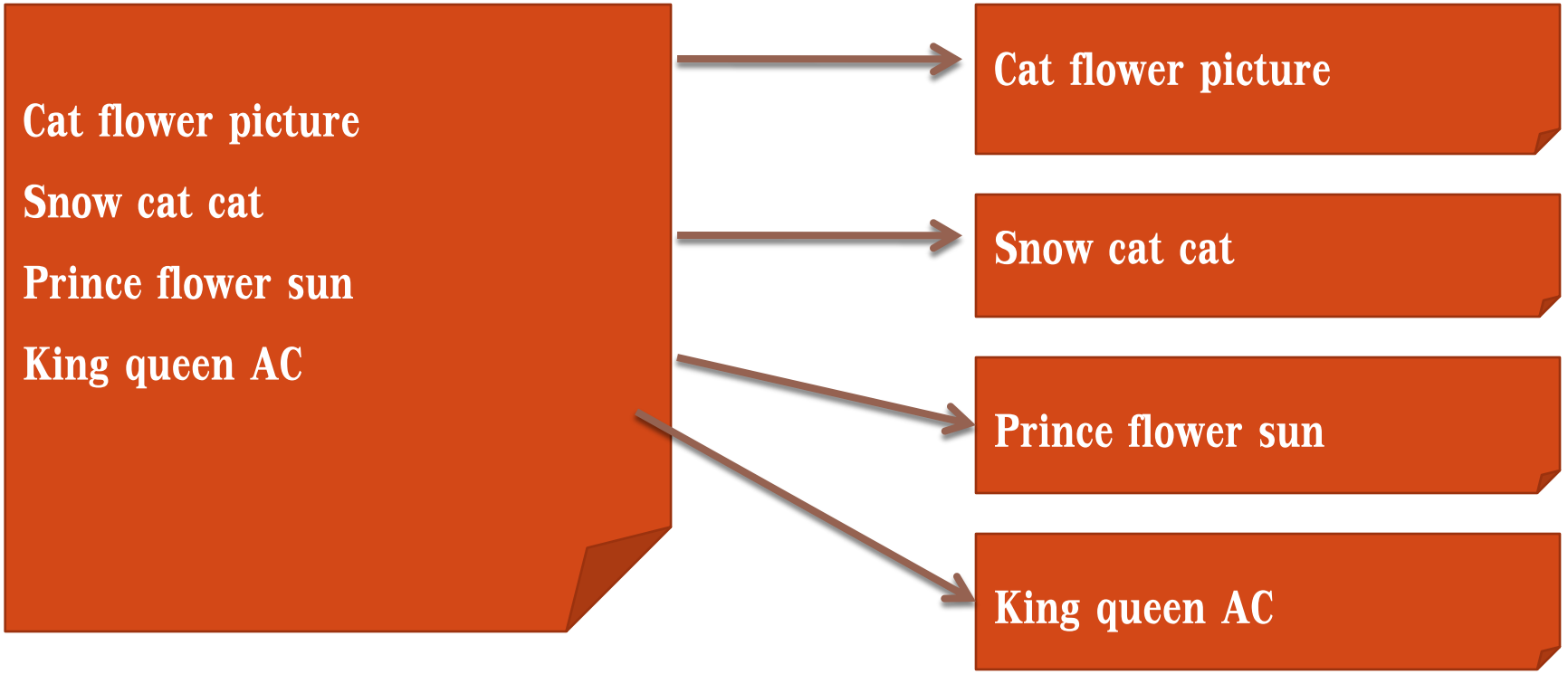
Example – Word Count-0

- Assume an input as following;

```
Cat flower picture
Snow cat cat
Prince flower sun
King queen AC
```

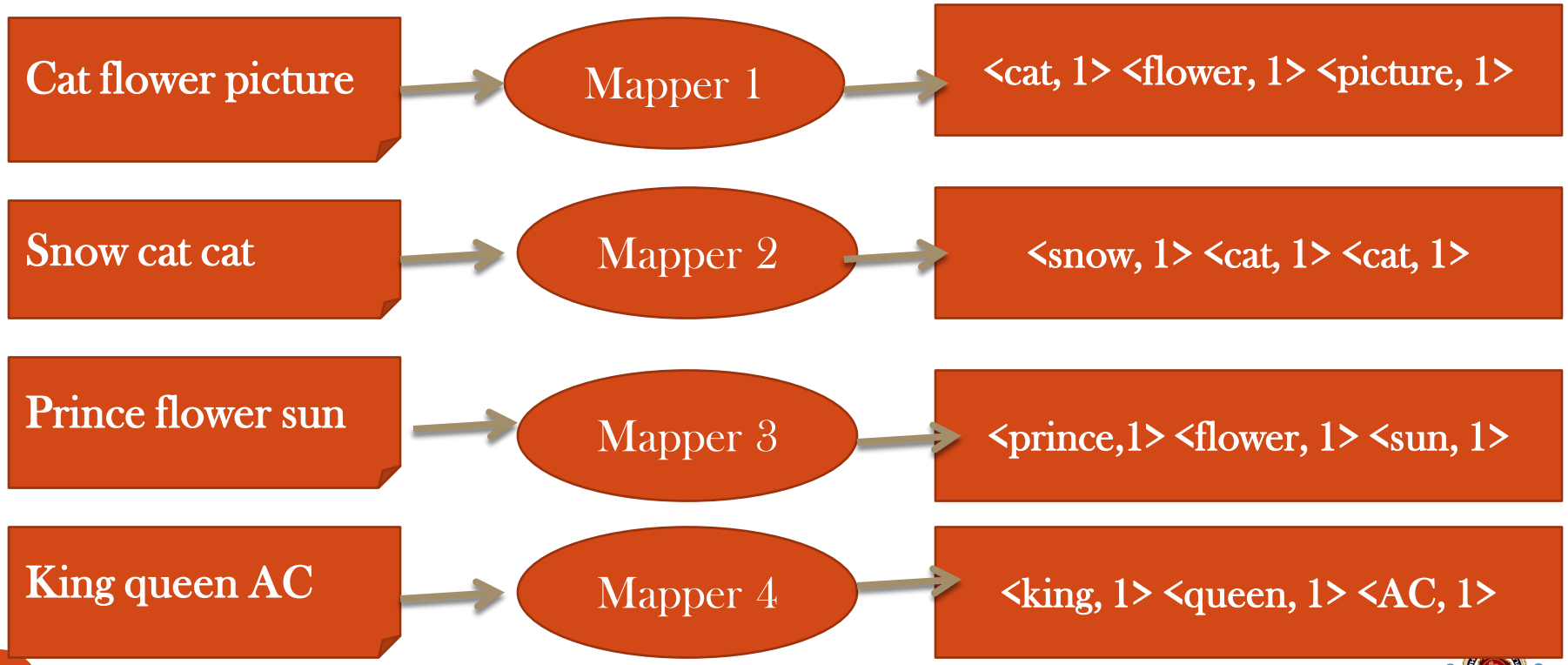
Example – Word Count-1

- Step1: Partition input file into **M** partitions.



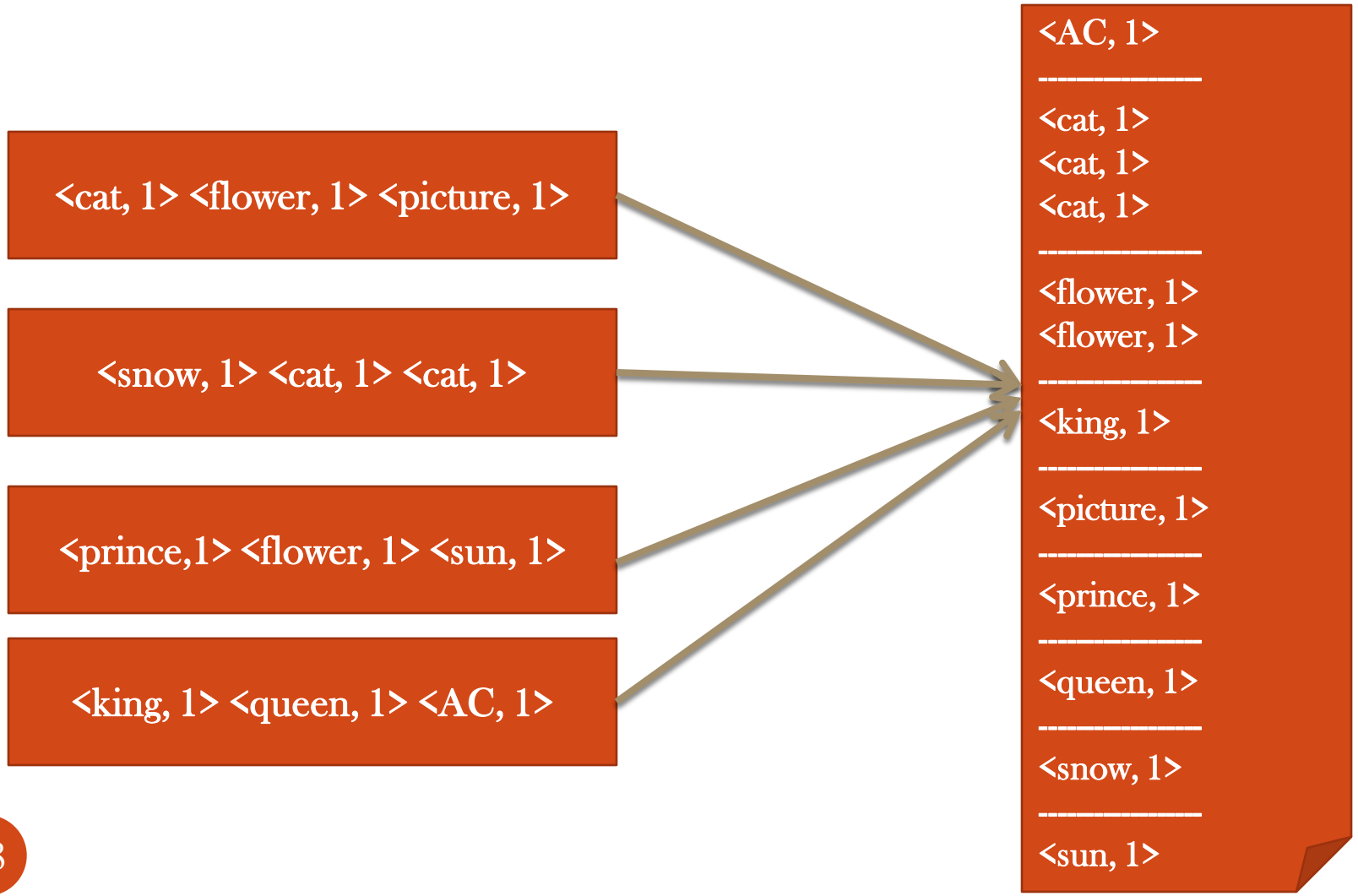
Example – Word Count-2

- Step2: Create M Map tasks, read M partitions in parallel and create intermediate pairs. Store them into local storage file into M partitions.



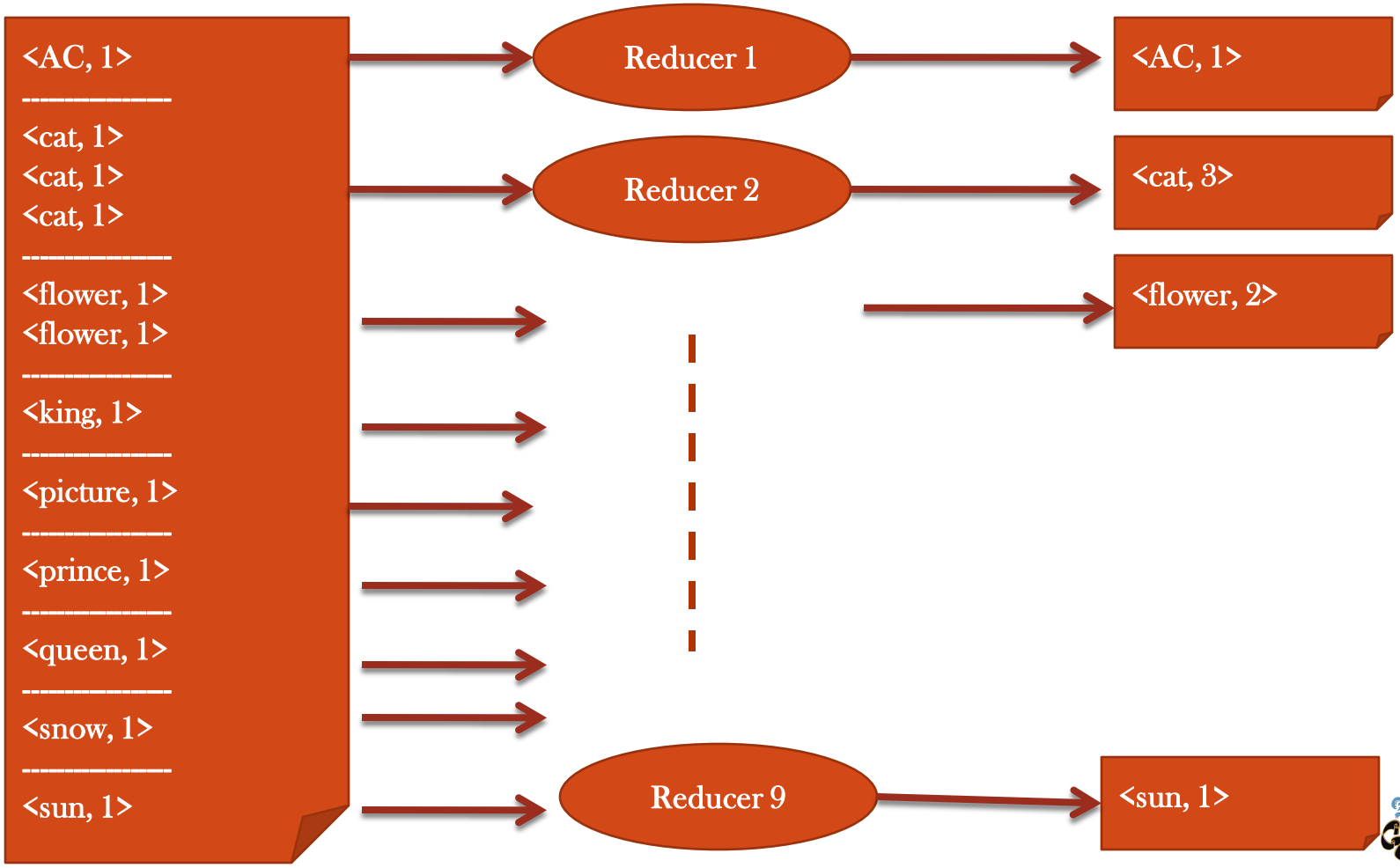
Example – Word Count-3

- Step3: Wait for all Map workers to finish.



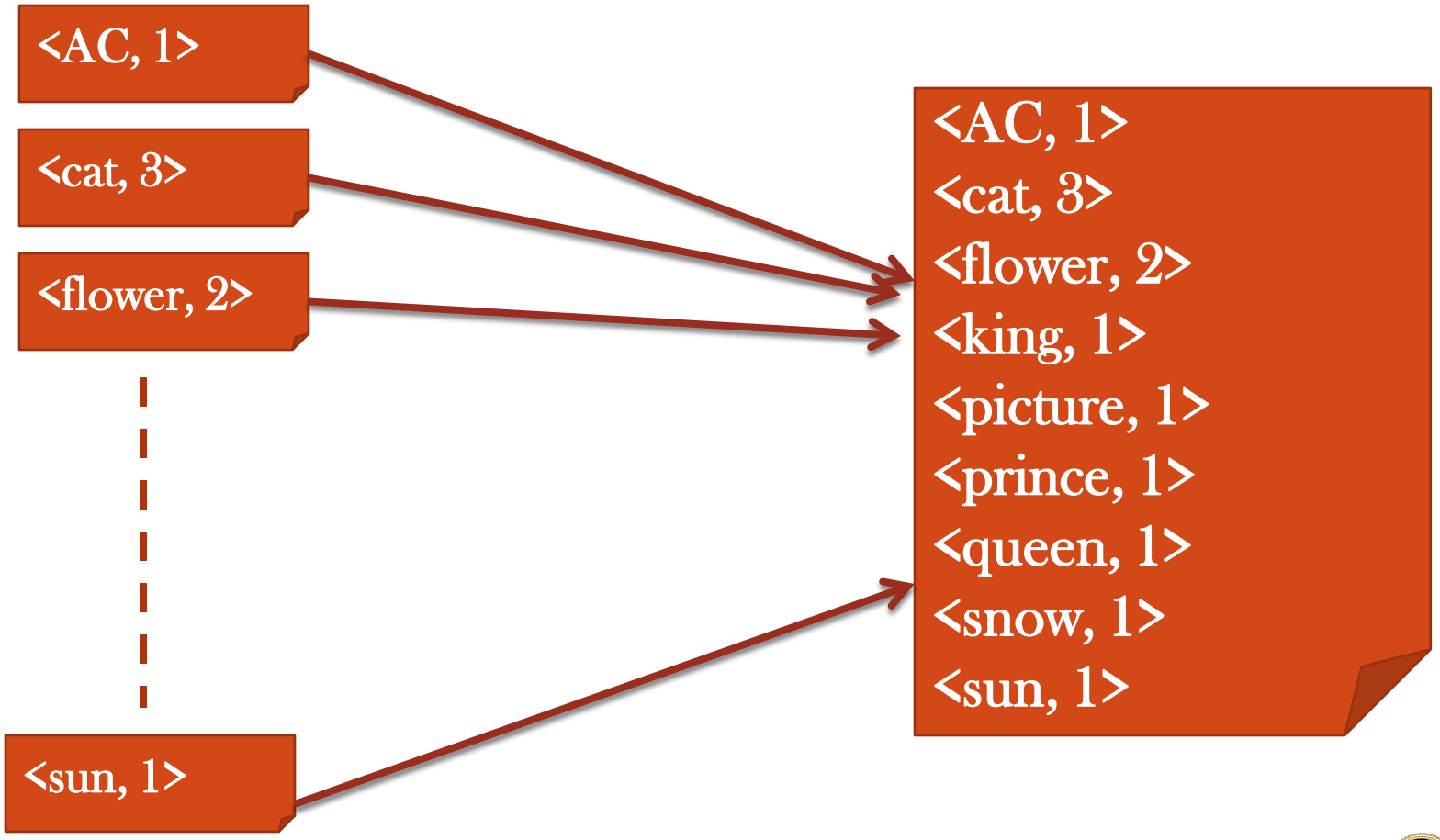
Example – Word Count-4

- Step4: The partition intermediate pairs into R regions.

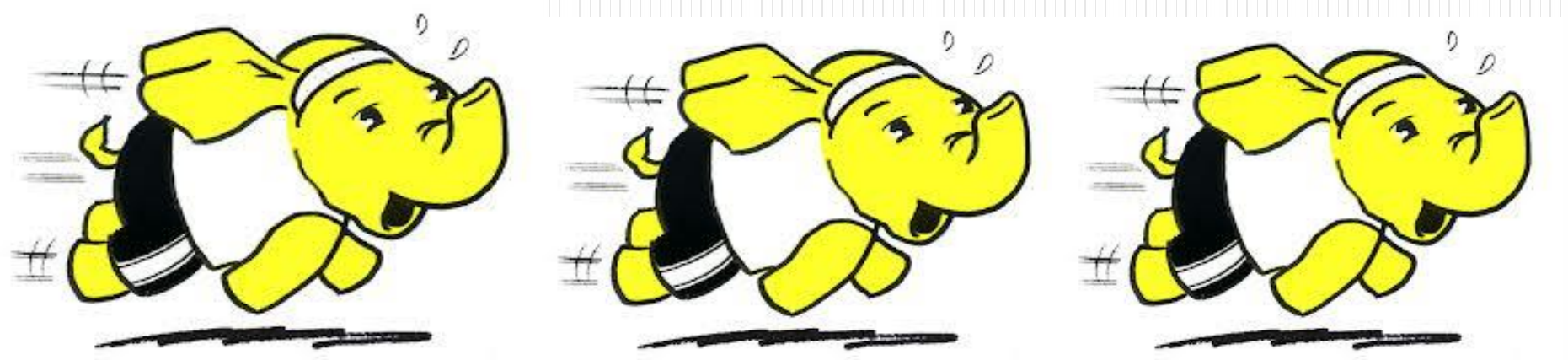


Example – Word Count-5

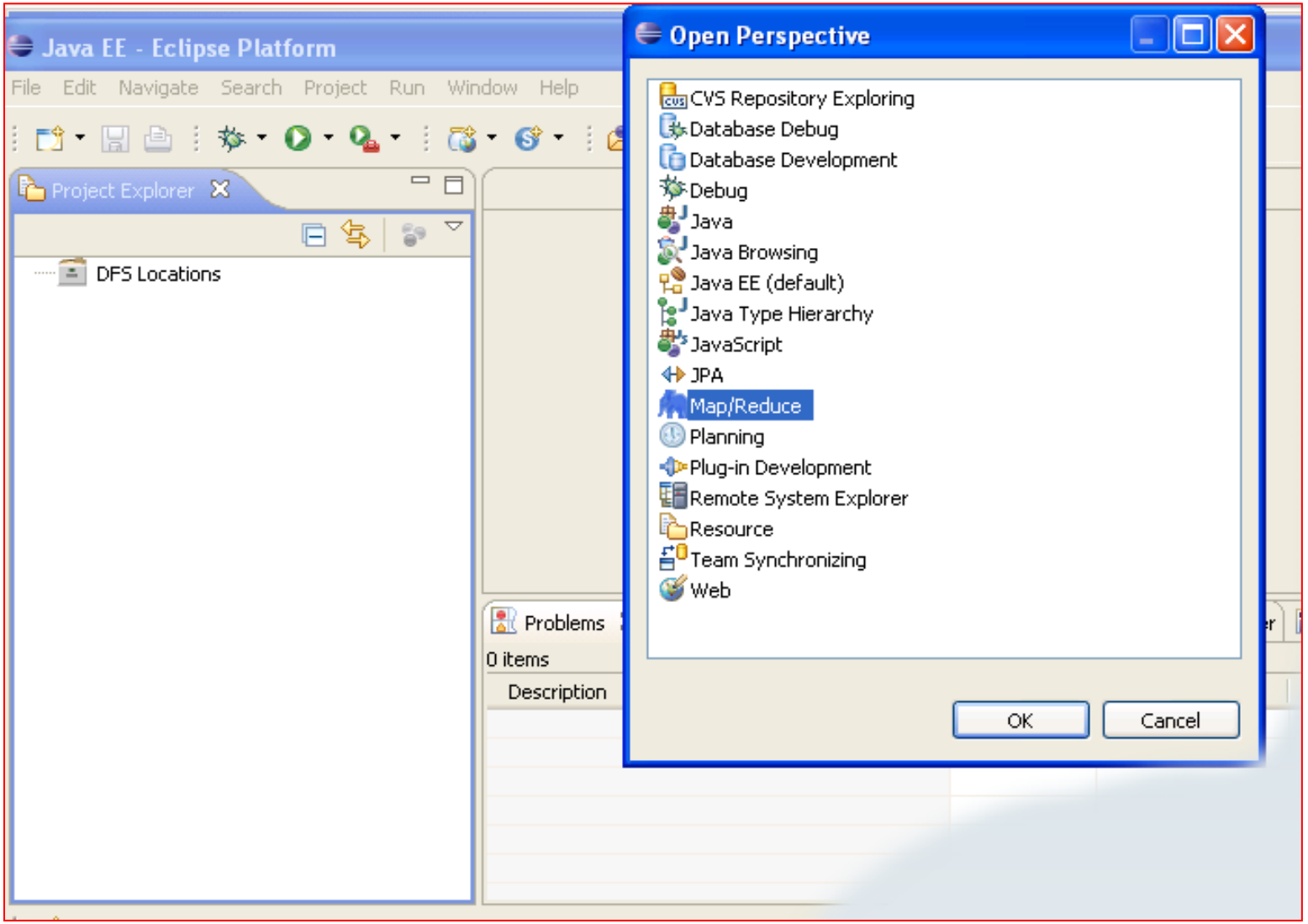
- Step5: Write the output of reduce workers to file(s).



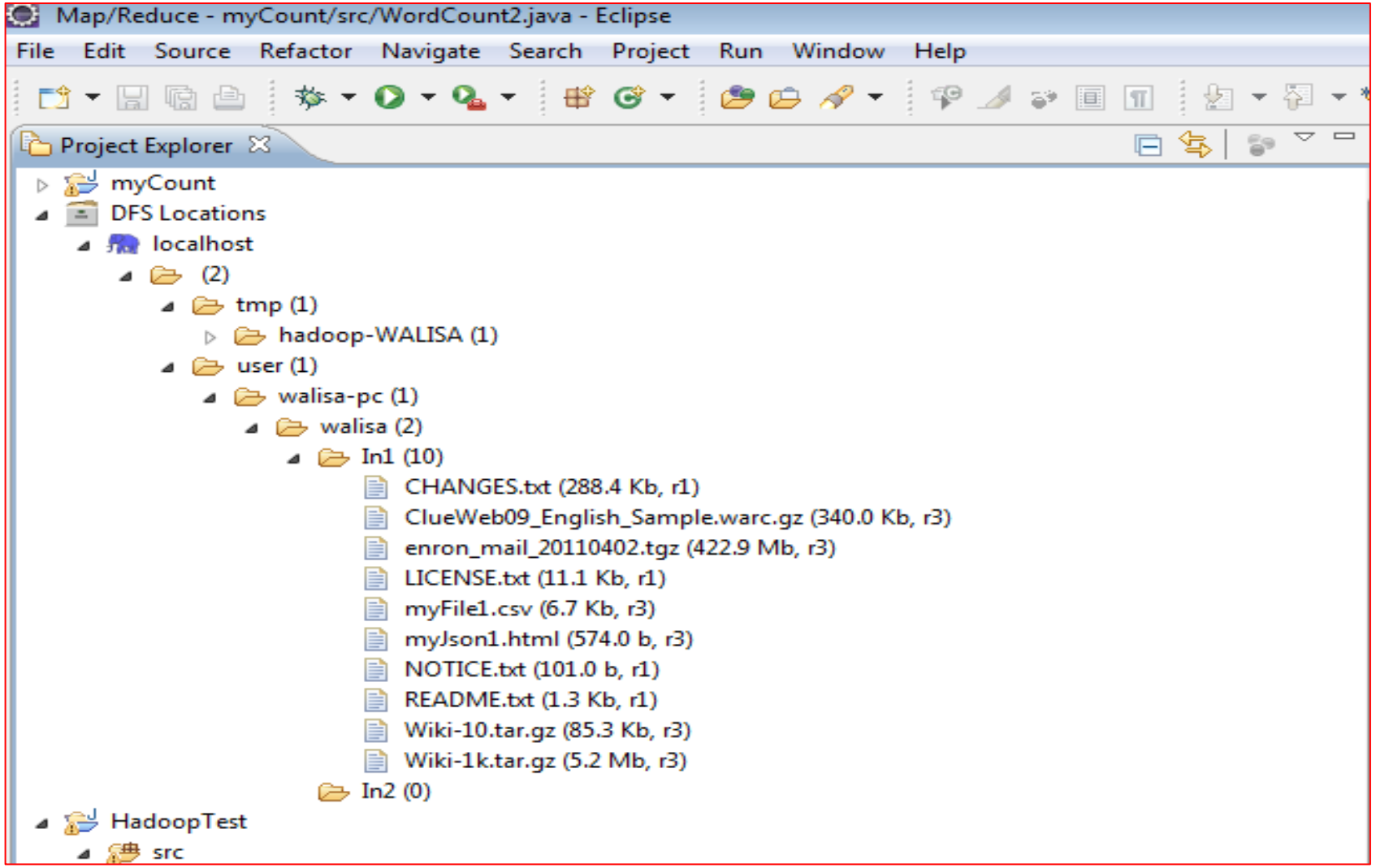
Getting Started Hadoop-3



MapReduce Project



DFS Location



Java : WordCount-1

```
public class WordCount {
public static class TokenizerMapper
extends Mapper<Object, Text, Text, IntWritable>
{
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();
```

Download WordCount : http://grepcode.com/file/repository.cloudera.com/content/repositories/releases/com.cloudera.hadoop/hadoop-examples/0.20.2-320/org/apache/hadoop/examples/AggregateWordCount.java?av=f

Java : WordCount-2

```
public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException
{
StringTokenizer itr = new
StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }
```

Java : WordCount-3

```
public void reduce(Text key, Iterable<IntWritable> values,
                   Context context
                   ) throws IOException, InterruptedException
{
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
}
```


Run Jar File

WALISA@WALISA-PC ~/hadoop-0.19.1

$ bin/hadoop jar WordCount.jar org.apache.hadoop.examples.WordCount /user/walisa-pc/walisa/In1/enron_mail_20110402.tgz /user/walisa-pc/walisa/myMail

55/05/09 15:59:15 INFO mapred.FileInputFormat: Total input paths to process : 1

55/05/09 15:59:15 INFO mapred.JobClient: Running job: job_255505091229_0015

55/05/09 15:59:16 INFO mapred.JobClient: map 0% reduce 0%

....

55/05/09 16:01:53 INFO mapred.JobClient: HDFS bytes read=443494101

55/05/09 16:01:53 INFO mapred.JobClient: HDFS bytes written=810486137

55/05/09 16:01:53 INFO mapred.JobClient: Local bytes read=1695775337

55/05/09 16:01:53 INFO mapred.JobClient: Local bytes written=2543280422

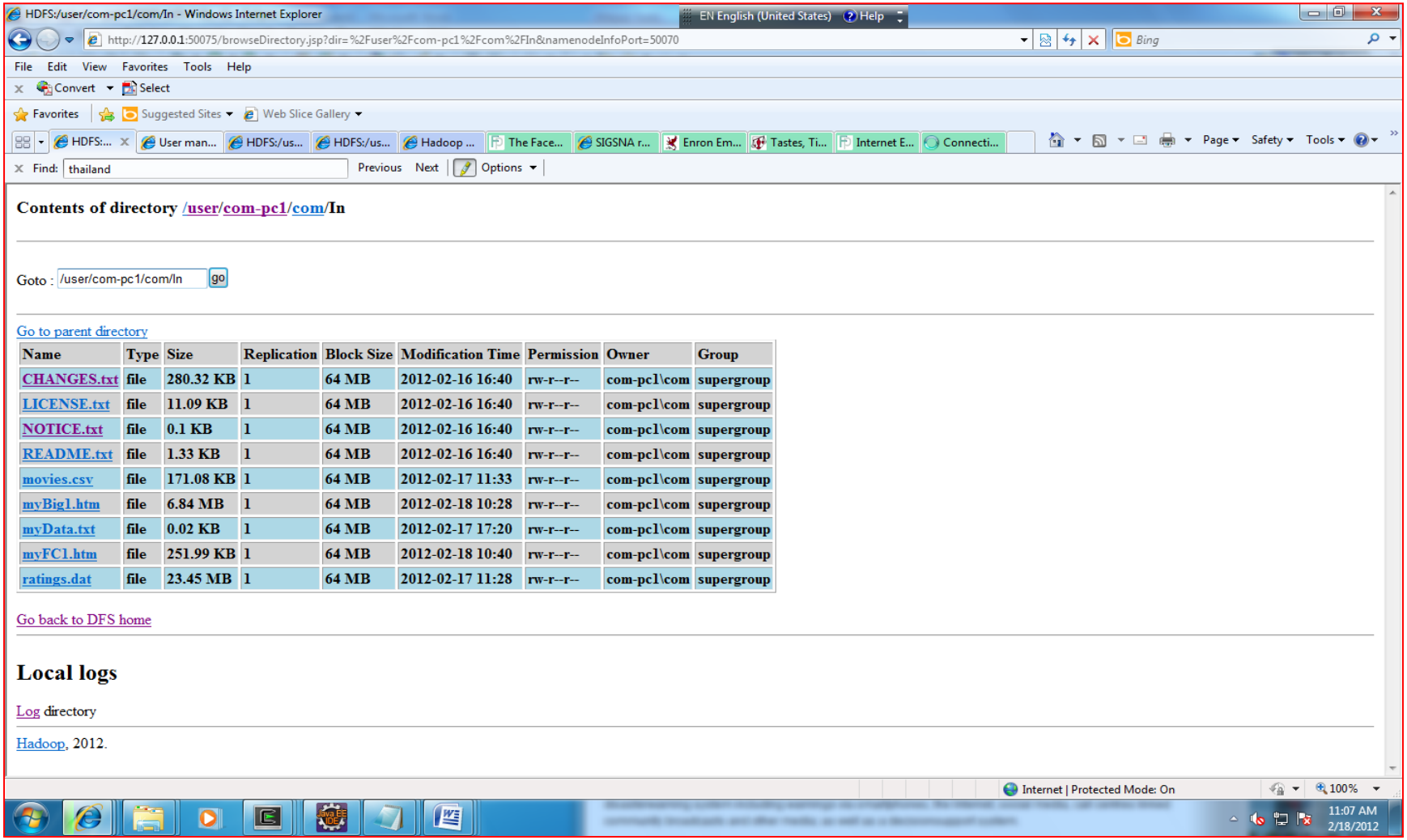
55/05/09 16:01:53 INFO mapred.JobClient: Launched reduce tasks=1

55/05/09 16:01:53 INFO mapred.JobClient: Launched map tasks=7

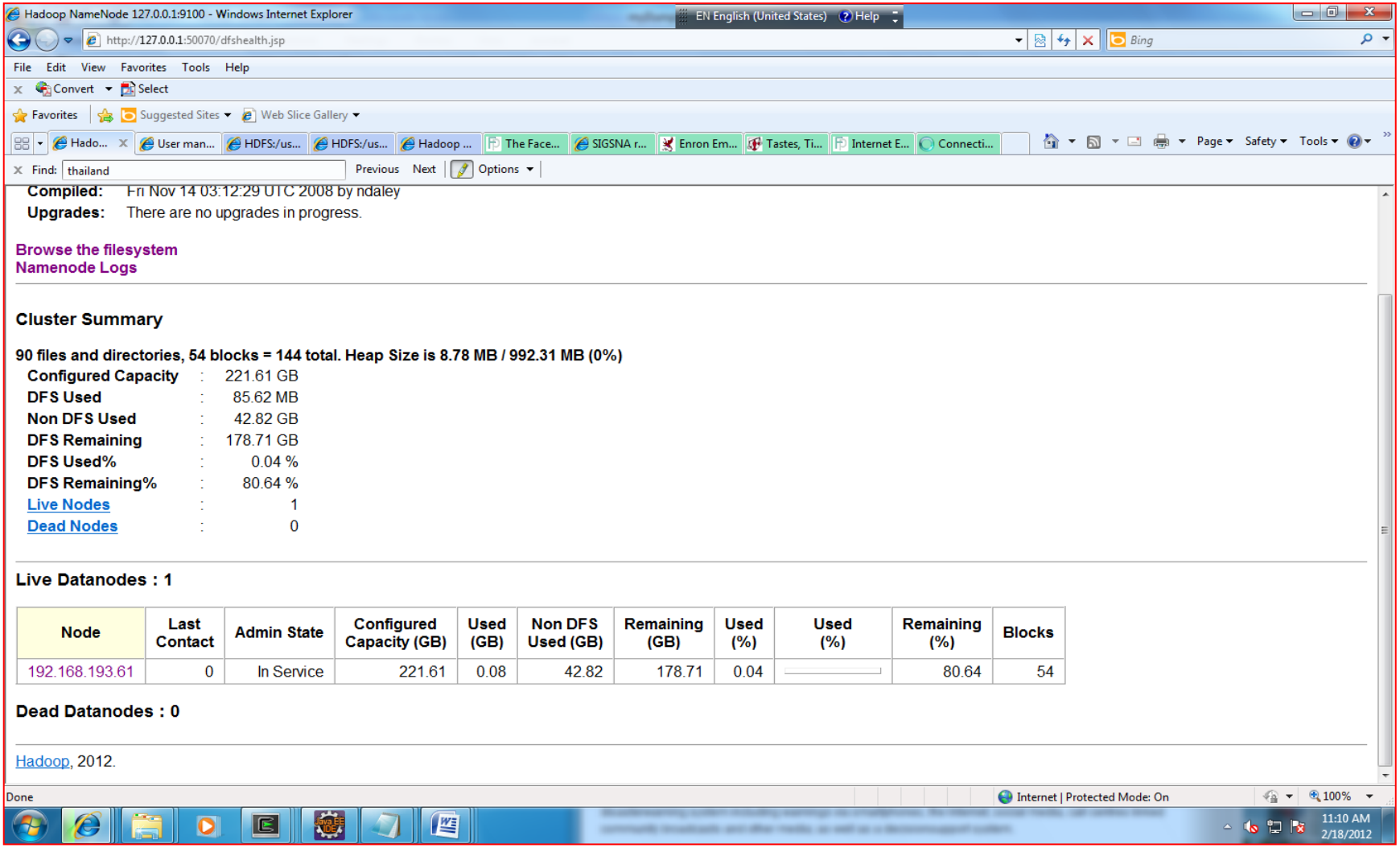
55/05/09 16:01:53 INFO mapred.JobClient: Data-local map tasks=7

55/05/09 16:01:53 INFO mapred.JobClient: Reduce input records=7659462

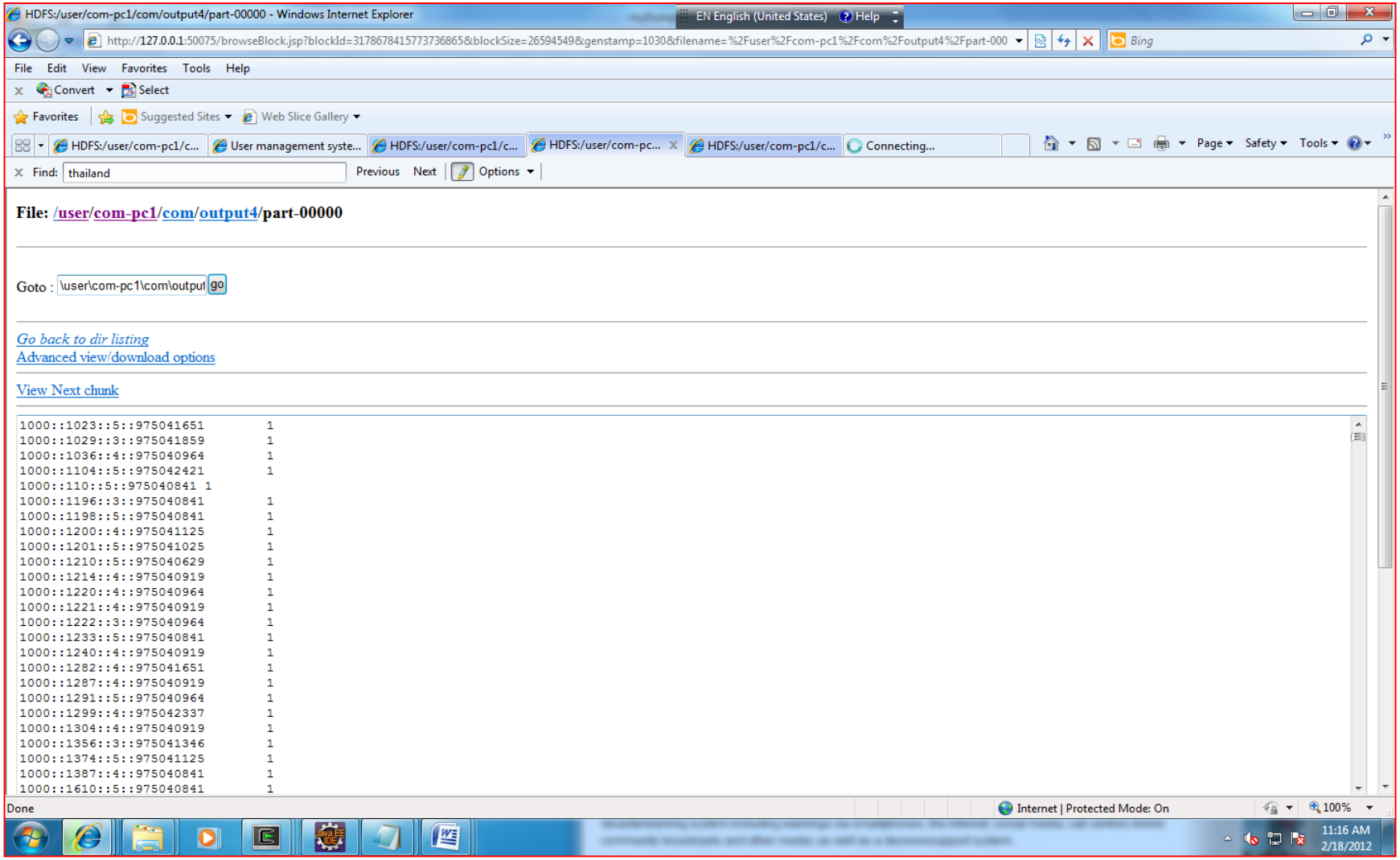
Run the result-1



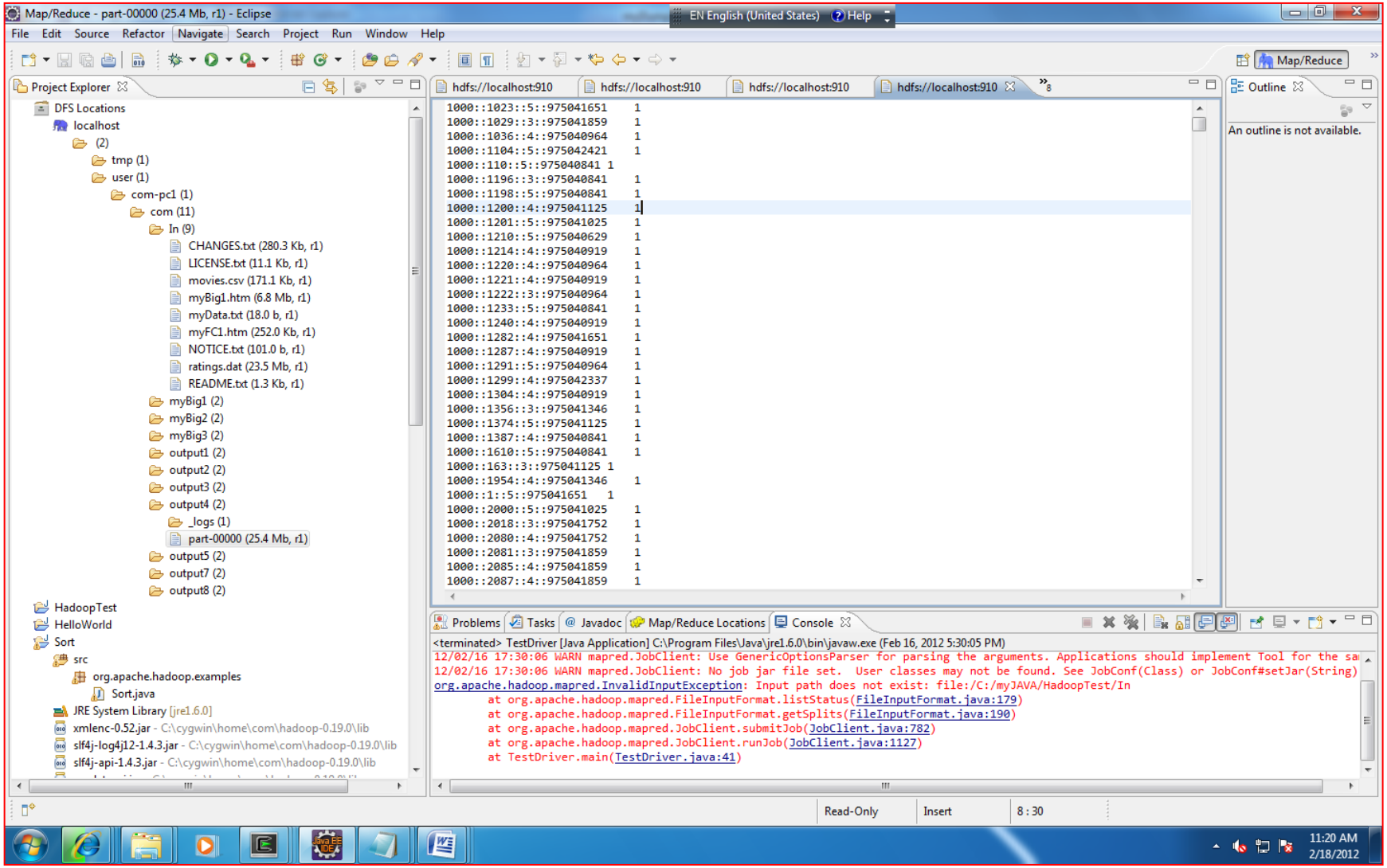
Run the result-2



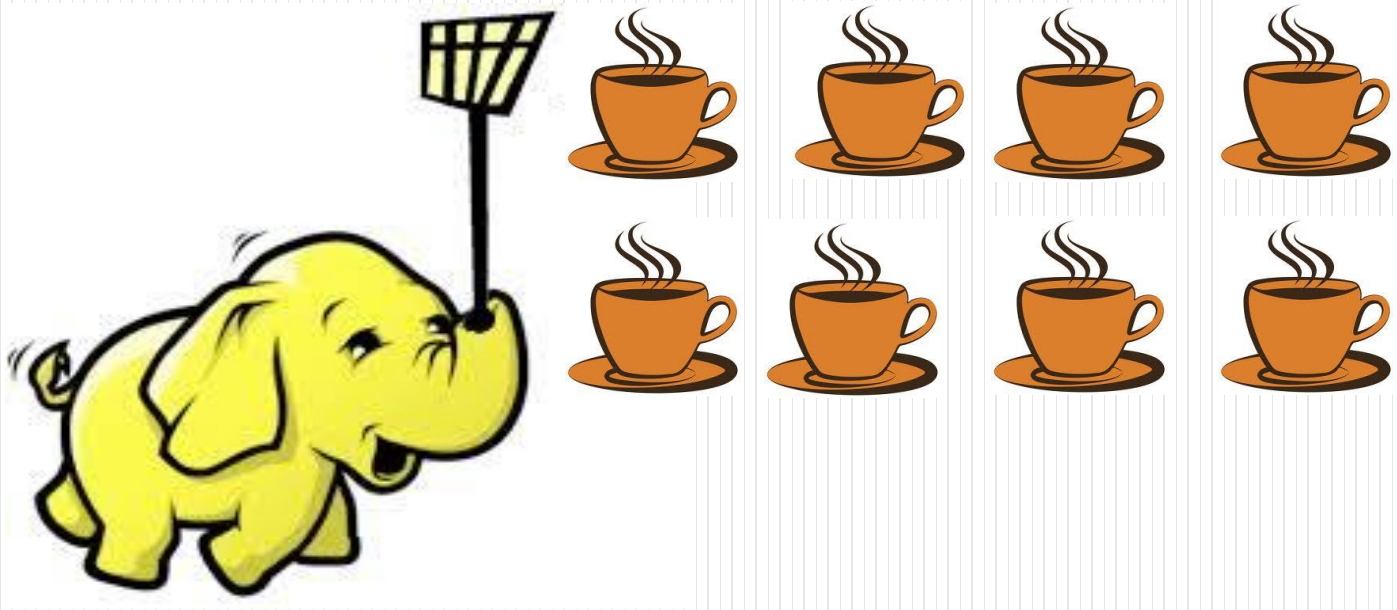
Run the result-3



Run the result-4



Taking a Coffee Break !!!



Hive

Hive Refresher

- Apache Hive in a few words:

  "*A data warehouse infrastructure built on top of Apache Hadoop*"

- Used for:
  - Ad-hoc querying and analyzing large data sets without having to learn MapReduce
- Main features:
  - SQL-like query language called QL
  - Built-in user defined functions (UDFs) to manipulate dates, strings, and other data-mining tools
  - Plug-in capabilities for custom mappers, reducers, and UDFs
  - Support for different storage types such as plain text, RCFiles, HBase, and others
  - Multiple clients like a shell, JDBC, Thrift

Installation and Configuration

- Download Hive
  - http://hive.apache.org/releases.html#Download
- **Requirements**
  - Java 1.6
  - Hadoop 0.20.x.
- Set Path

  $ export PATH=$HIVE_HOME/bin:$PATH

Starting the Hive Shell

- Create Table into Hive:
  hive> CREATE TABLE shakespeare (freq
  INT, word STRING) ROW FORMAT
  DELIMITED FIELDS TERMINATED BY '\t'
  STORED AS TEXTFILE;
- Load dataset into Hive:
  hive> LOAD DATA INPATH "shakespeare_freq"
  INTO TABLE shakespeare
- Selecting Data
  hive> SELECT * FROM shakespeare
  WHERE freq > 100 SORT BY freq ASC
  LIMIT 10;

Apache Weblog Data

```
CREATE TABLE apachelog (
  host STRING,
  identity STRING,
  user STRING,
  time STRING,
  request STRING,
  status STRING,
  size STRING,
  referer STRING,
  agent STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
   "input.regex" = "([^]*) ([^]*) ([^]*) (-|\\[^\\]*\\]) ([^ \"]*|\"[^\"]*\") (-|[0-9]*) (-|[0-9]*)(?: ([^
\"]*|\".*\") ([^\"]*|\".*\"))?",
  "output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s"
)
STORED AS TEXTFILE;
```

Hive Applications

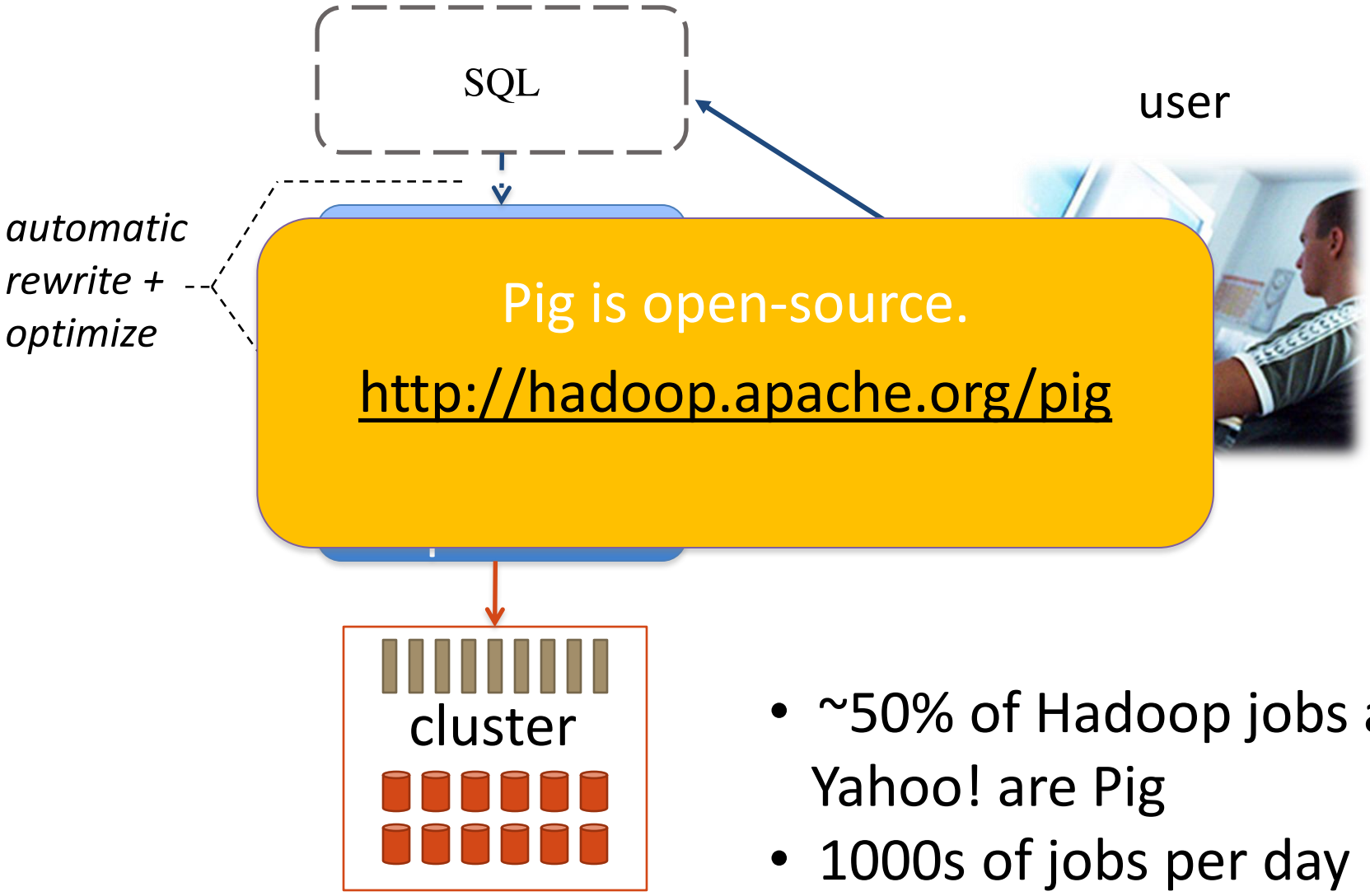
- Log processing
- Text mining
- Document indexing
- Customer-facing business intelligence (e.g., Google Analytics)
- Predictive modeling, hypothesis testing

Hive Usage @Facebook

- Usage statistics:
  - Total Users: ~140 (about 50% of engineering !) in the last 1 ½ months
  - Hive Data (compressed): 80 TB total, ~1TB incoming per day
  - Job statistics:
    - ~1000 jobs/day
    - ~100 loader jobs/day

Pig

Pig



- ~50% of Hadoop jobs at Yahoo! are Pig
- 1000s of jobs per day

Installation and Configuration

- Download Pig

  http://pig.apache.org/

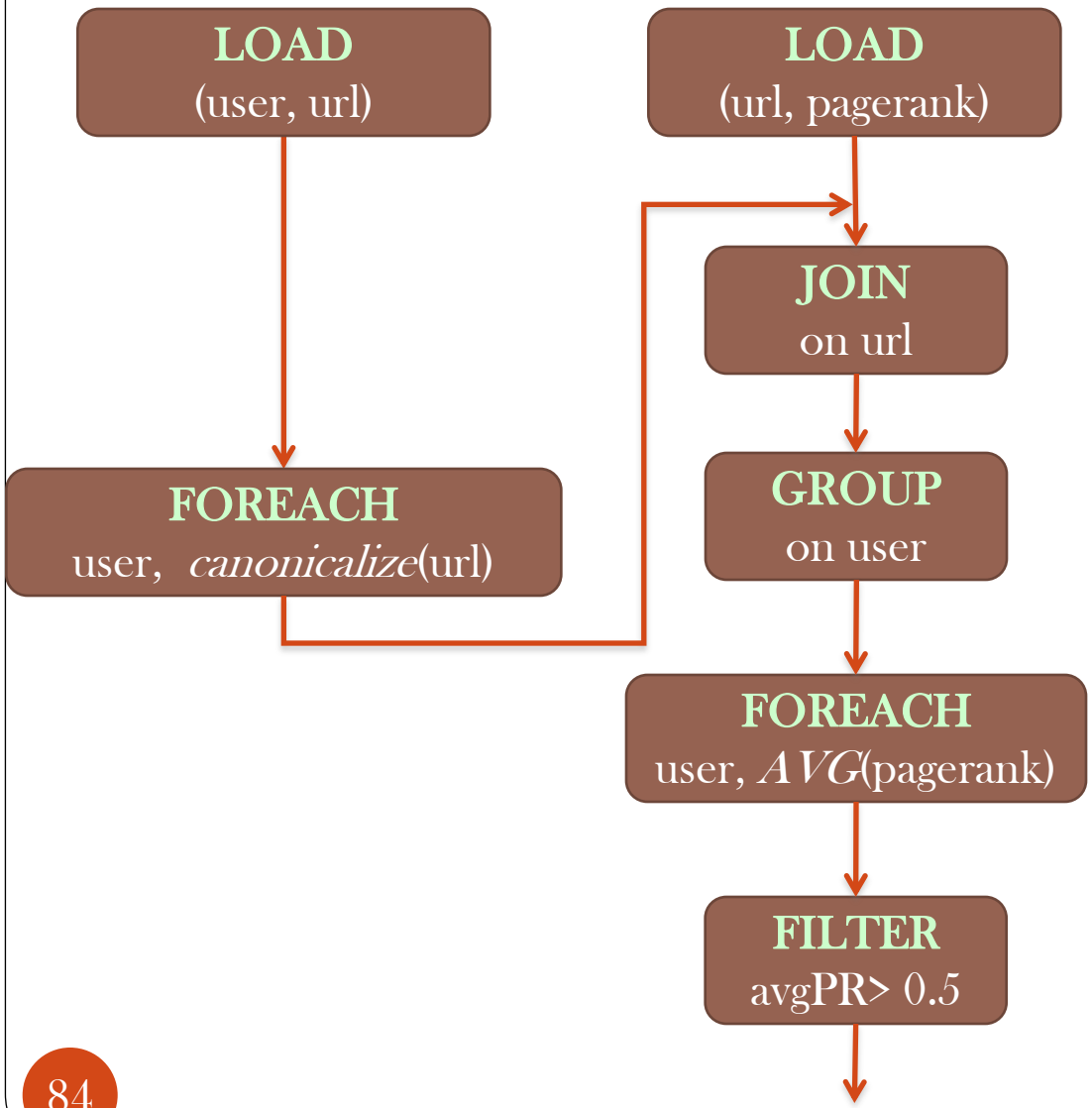
- **Requirements**
  - Java 1.6
  - Hadoop 0.20.0
  - The JAVA_HOME environment variable is set the root of your Java installation.

Pig Latin

Pig provides a higher level language, Pig Latin, that:

- Increases productivity.
  - In one test 10 lines of Pig Latin ≈200 lines of Java.
  - What took 4 hours to write in Java took 15 minutes in Pig Latin.
- Opens the system to non-Java programmers.
- Provides common operations like join, group, filter, sort.

Example Dataflow Program



Find users that tend to visit high-pagerank pages

How It Works

Pig Latin

```
A = LOAD 'myfile'
    AS (x, y, z);
B = FILTER A by x > 0;
C = GROUP B BY x;
D = FOREACH A GENERATE
    x, COUNT(B);
STORE D INTO 'output';
```

pig.jar:

- parses
- checks
- optimizes
- plans execution
- submits jar to Hadoop
- monitors job progress

Execution Plan

Map:
Filter

Reduce:
Count

Pig Word Count Script

- A = load './input.txt';
- B = foreach A generate flatten(TOKENIZE((chararray)$0)) as word;
- C = group B by word;
- D = foreach C generate COUNT(B), group;
- store D into './wordcount';

Running Pig Word Count Script

- Local Mode

  bin/pig -x local wordcount.pig

- MapRedce Mode

  hadoop dfs -copyFromLocal input.txt input/input.txt

  bin/pig -x mapreduce wordcount.pig

Pig Applications

- Web log processing.
- Data processing for web search platforms.
- Ad hoc queries across large data sets.
- Rapid prototyping of algorithms for processing large data sets

Publication

International Journal of Intelligent Systems -Wiley

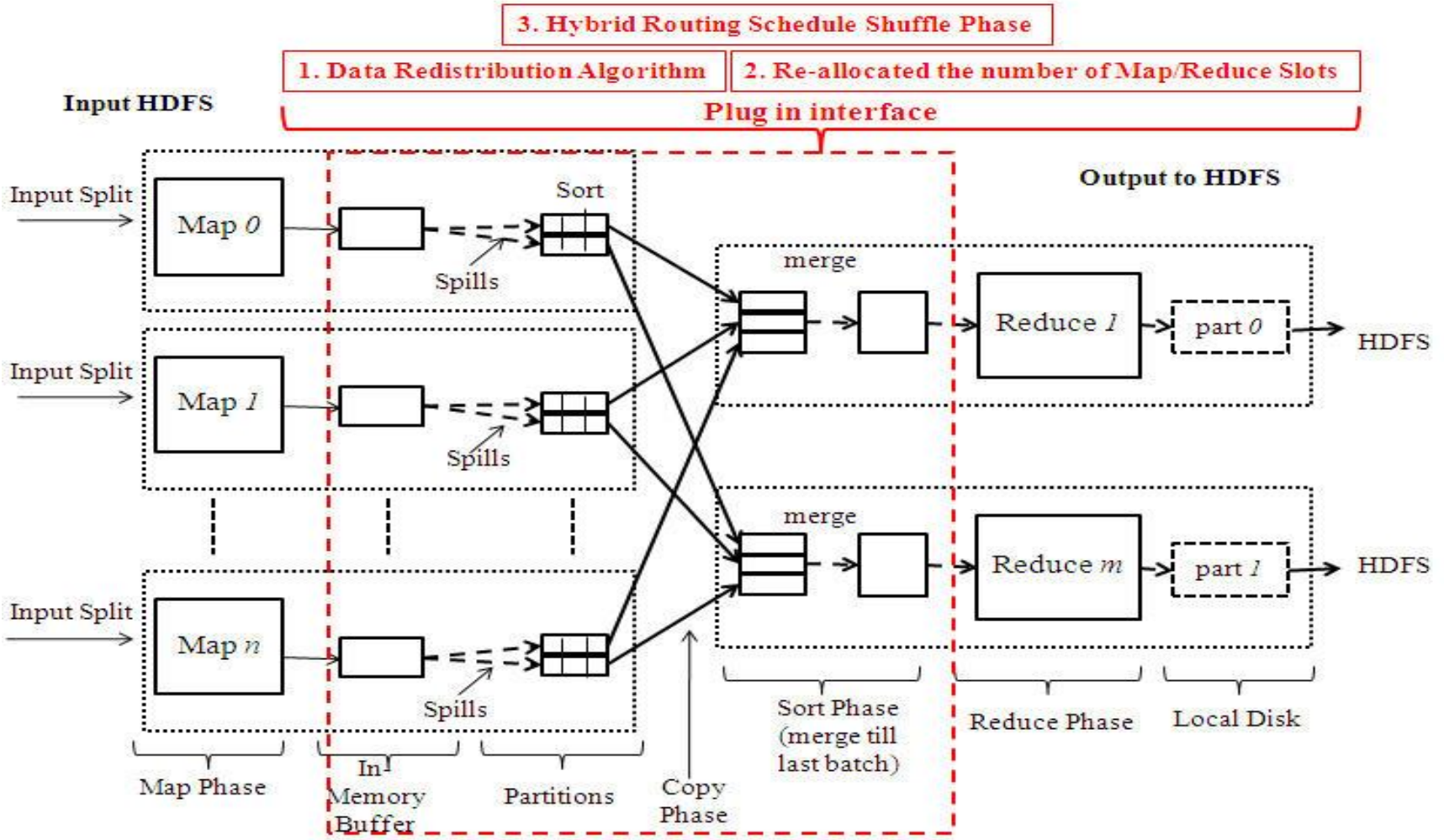
- Optimizing and Tuning MapReduce Jobs to Improve the Large-Scale Data Analysis Process
  - Wichian Premchaiswadi and Walisa Romsaiyud
  - Graduate School of Information Technology, Siam University.
  - Vol. 28, Issue 2, 2013, pp.185-200.

- Edited By: Ronald R. Yager
- Impact Factor: 1.416
- ISI Journal Citation Reports © Ranking: 2012: 49/114 (Computer Science Artificial Intelligence)
- Online ISSN: 1098-111X

Abstract

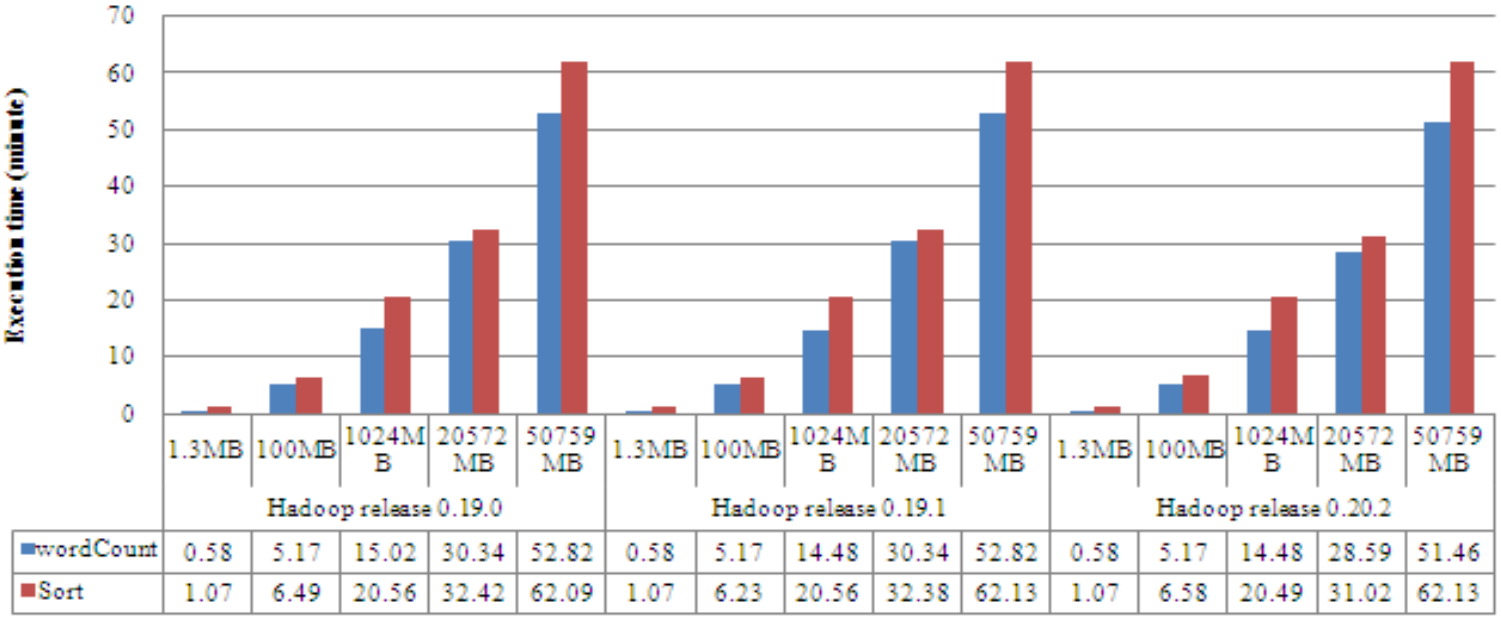
- Data-intensive applications process large volumes of data using a parallel processing method. Additionally, MapReduce is a programming model designed for data-intensive applications for massive datasets and an execution framework for large-scale data processing on clusters of commodity servers.
- While fault tolerance, easy programming structure and high scalability are considered strong points of MapReduce, however its configuration parameters must be fine-tuned to the specific deployment which makes it more complex in configuration and performance.
- This paper explains tuning of the Hadoop configuration parameters which directly affects MapReduce's job workflow performance under various conditions to achieve maximum performance. Based on the empirical data we collected it became apparent that three main methodologies can affect the execution time of MapReduce running on cluster systems.
- Therefore, in this paper, we present a model that consists of three main modules;
  - 1) We extended a Data Redistribution to find the high performance nodes,
  - 2) We utilized the Number of Map/Reduce Slots to make it more efficient in terms of execution time,
  - 3) We developed a new Hybrid Routing Schedule Shuffle Phase to define the scheduler task to reduce the memory management level.

THE SYSTEM DESIGN



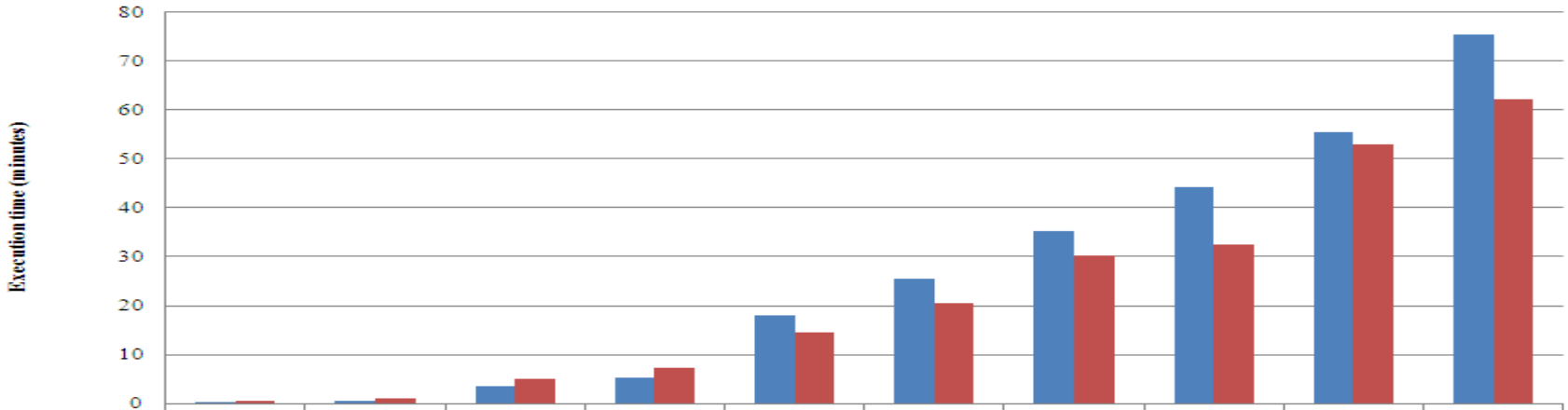
Efficiency Results-1

	Hadoop release 0.19.0					Hadoop release 0.19.1					Hadoop release 0.20.2				
Application (MB)	1.3	100	1024	20572	50759	1.3	100	1024	20572	50759	1.3	100	1024	20572	50759
wordCount	0.58	5.17	15	30.34	52.82	0.58	5.17	14.5	30.34	52.82	0.58	5.17	14.5	28.59	51.46
Sort	1.07	6.49	20.6	32.42	62.09	1.07	6.23	20.6	32.38	62.13	1.07	6.58	20.5	31.02	62.13



Efficiency Results-2

- We ran our program on Hadoop version 0.19.1 testing the three plug-in modules using Hadoop MapReduce. The wordCount and sort applications were run and show the comparison of the execution time of WordCount in native and our program called XMR(eXtensible-MapReduce).

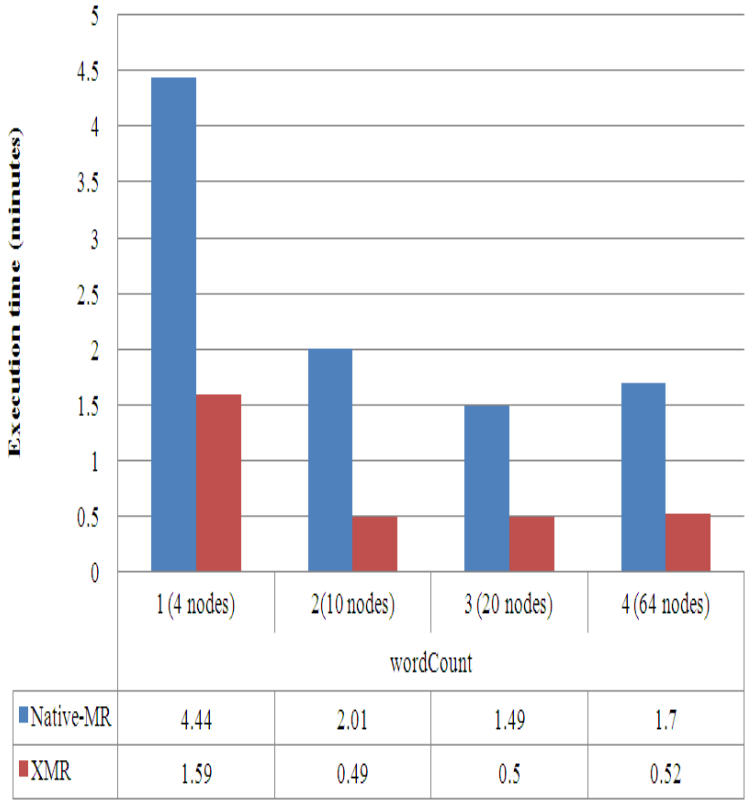


	wordCount	Sort	wordCount	Sort	wordCount	Sort	wordCount	Sort	wordCount	Sort
	1.3MB		100MB		1024MB		20572MB		50759MB	
Native-MR	0.44	0.57	3.49	5.27	18.06	25.44	35.22	44.16	55.39	75.45
XMR	0.58	1	5.17	7.23	14.48	20.56	30.34	32.38	52.82	62.13

Efficiency Results-3

We executed three modules with Hadoop in cluster 10 nodes. Table summarizes the characteristics of test sets used in this paper. For each test set configuration, we compared the performance of XMR with that of Hadoop in the following subsections. As in the overall performance, we measured the response time to complete the corresponding workload.

	1	2	3	4
Workload	wordCount	wordCount	wordCount	wordCount
#nodes	4	10	20	64
#map tasks	18	18	18	50
#reduce tasks	1	1	1	1
Input file size	27 GB	27 GB	27 GB	27 GB
Split data size	128MB	128MB	128MB	128MB



Conclusions

- This paper presented an extension to the "Hadoop MapReduce" model featuring a hierarchical reduction phase.
- This model is called XMR, which can automatically configure "Hadoop MapReduce" through tuning a number of parameters that influence the performance in terms of execution time. XMR includes three modules as follows:
- 1) data redistribution,
- 2) re-allocation of the number of Map/Reduce slots,
- 3) introducing a new hybrid routing schedule shuffle phase.
- Furthermore, we conducted extensive experiments by analyzing the efficiency and performance compared with Native MapReduce (Native-MR) founded on Hadoop release 0.19.0, 0.19.1, 0.20.2 and etc.
- The experimental results show that XMR improves the performance on execution time by 30.68% for a single large file and 16.35% for a cluster system. Additionally, the hybrid routing schedule on shuffle phase achieves the highest performance when the file size is more than 1024MB.
- These results indeed demonstrate the effectiveness of parameter configuration on "Hadoop MapReduce" using XMR. Also, as an advantage, the methodology developed here works well with the particular parameter tuning in "Hadoop MapReduce".

Sample DataSets

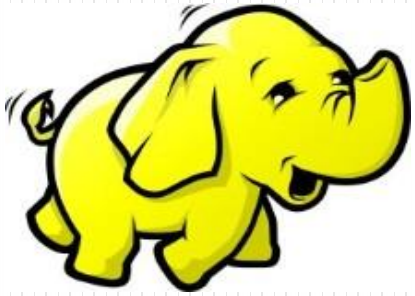
- Road
  - http://www.census.gov/geo/maps-data/data/tiger-geodatabases.html
- Infochimps
  - http://www.infochimps.com/tags/hadoop
- University of Southern California (Voting record, Tiwtter and Flickr)
  - http://www.isi.edu/integration/people/lerman/downloads.html
- Public Datasets
  - http://www.scaleunlimited.com/datasets/public-datasets/

Reference WebSites

- Cloudera
  - http://www.cloudera.com/content/cloudera/en/home.html
- Apache
  - http://hadoop.apache.org/
- IBM
  - http://www-01.ibm.com/software/data/infosphere/hadoop/
  - http://www.ibm.com/big-data/us/en/downloads/
- Yahoo
  - http://developer.yahoo.com/hadoop/tutorial/index.html
- Microsoft
  - http://www.microsoft.com/en-us/sqlserver/solutions-technologies/business-intelligence/big-data.aspx
  - http://www.microsoft.com/en-us/download/details.aspx?id=37134
- Amazon WebServices
  - http://aws.amazon.com/

Thank you very much.

walisa.rom@siam.edu



Walisa and the gang