



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.



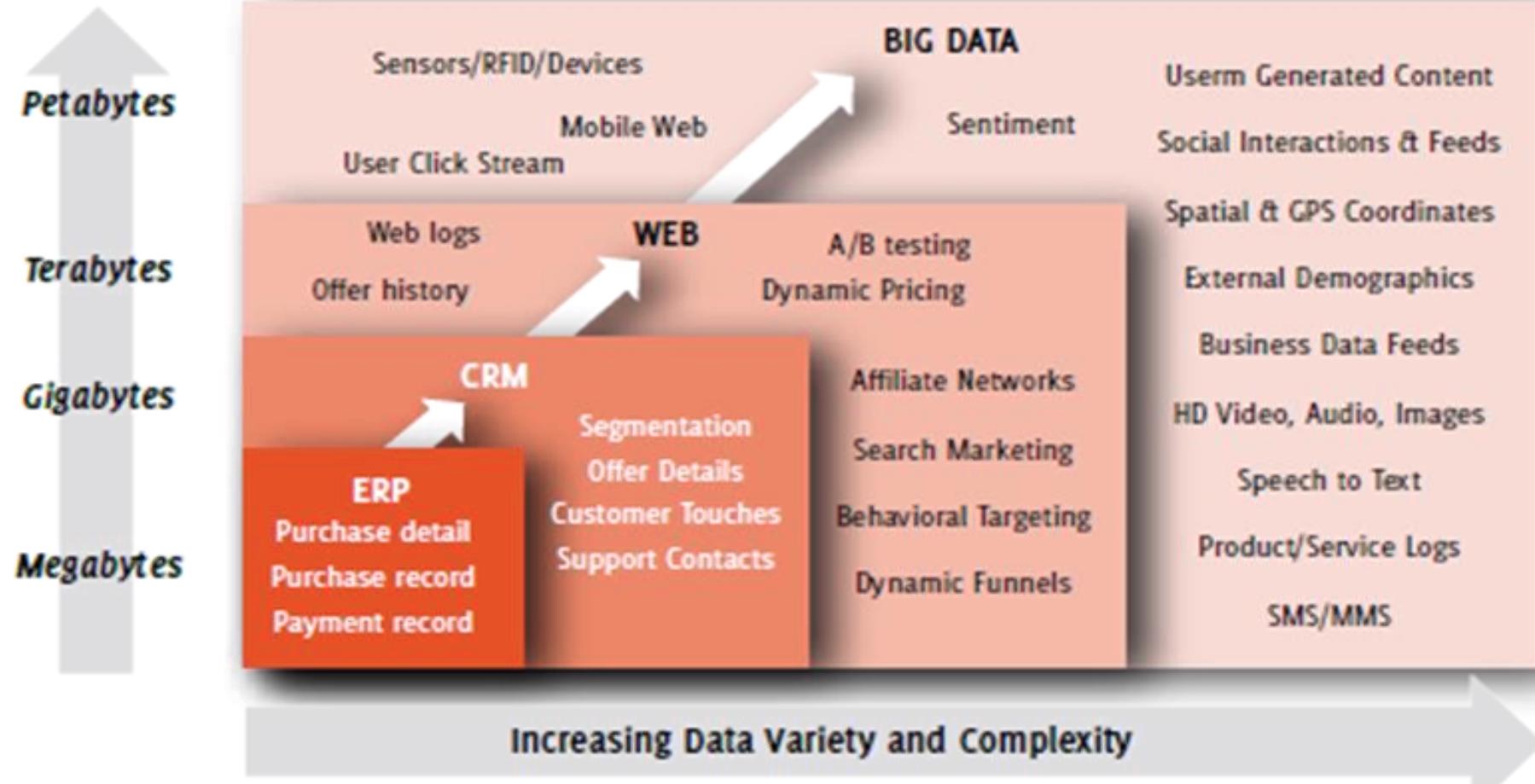
The ‘Big Data’ Phenomenon!!!!

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)

Yottabyte
This is our digital universe today
= 250 trillion of DVDs

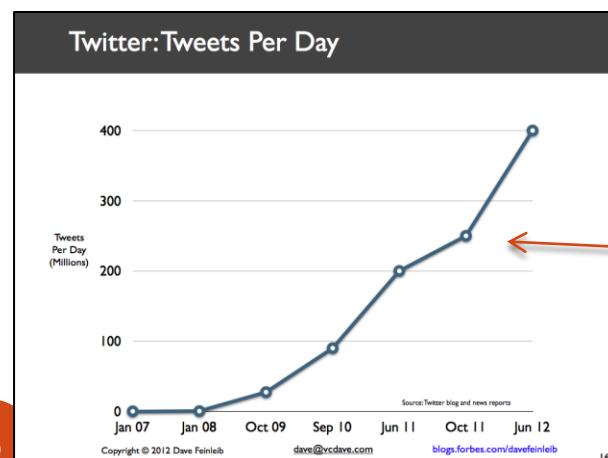
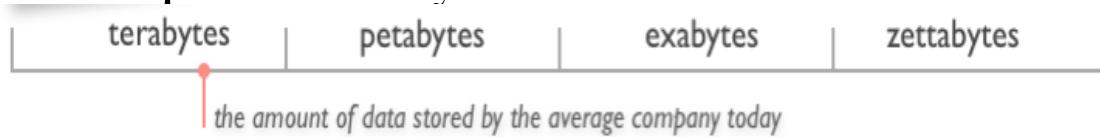
1 EB of data is created on the internet each day = 250 million DVDs!

500TB of new data per day are ingested in Facebook databases

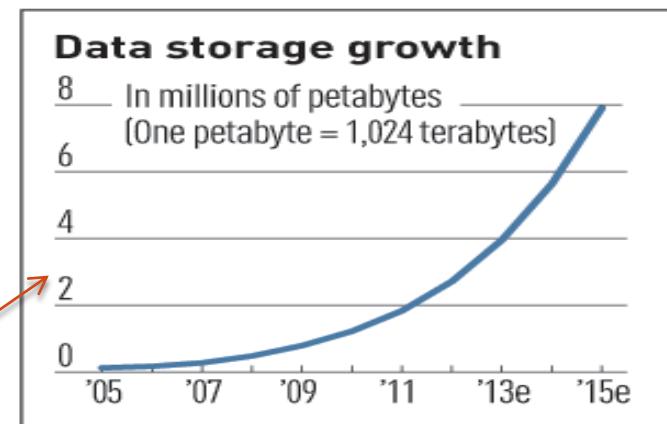
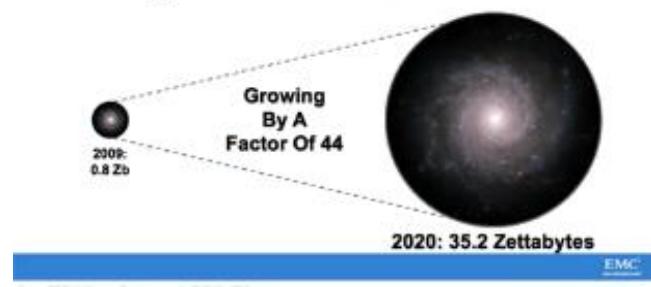


Characteristics of Big Data: 1-Scale (Volume)

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



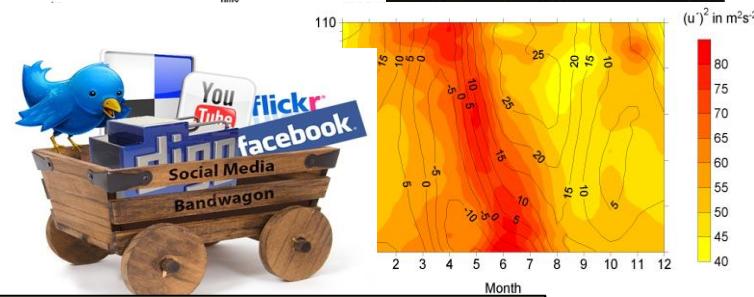
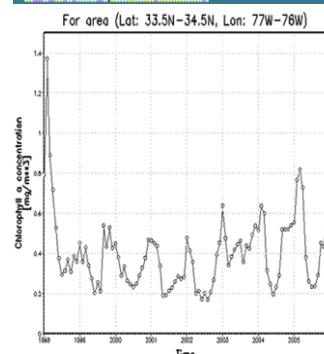
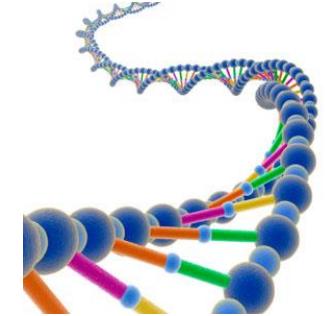
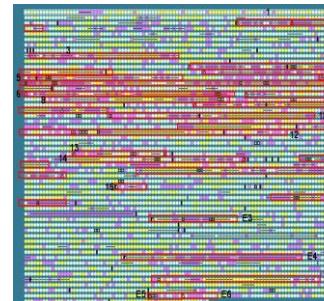
The Digital Universe 2009-2020



*Exponential increase in
collected/generated data*

Characteristics of Big Data: 2-Complexity (Variety)

- 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

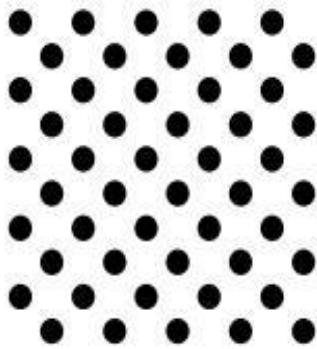
Characteristics of Big Data: 3-Speed (Velocity)



- Data is being 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

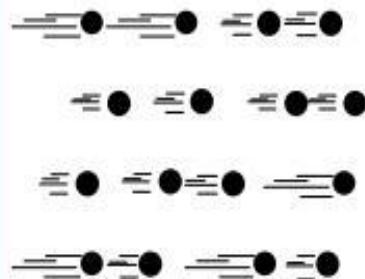
Volume



Data at Rest

Terabytes to exabytes of existing data to process

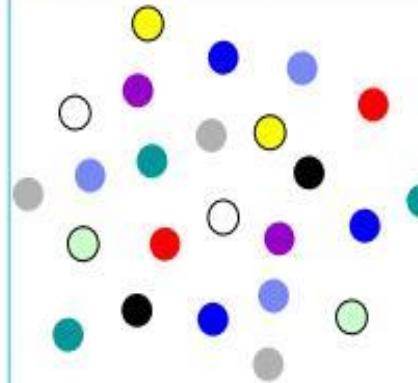
Velocity



Data in Motion

Streaming data, milliseconds to seconds to respond

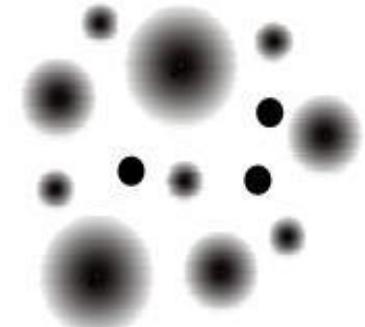
Variety



Data in Many Forms

Structured, unstructured, text, multimedia

Veracity*



Data in Doubt

Uncertainty due to data inconsistency & incompleteness, ambiguities, latency, deception, model approximations

Examples of Big Data Analytics

Potential Use Cases for Big Data Analytics

Real Time

Credit & Market Risk in Banks

Fraud Detection (Credit Card) & Financial Crimes (AML) in Banks
(including Social Network Analysis)

Event-based Marketing in Financial Services and Telecoms

Markdown Optimization in Retail

Claims and Tax Fraud in Public Sector

Predictive Maintenance in Aerospace

Social Media Sentiment Analysis

Demand Forecasting in Manufacturing

Disease Analysis on Electronic Health Records

Traditional Data Warehousing

Text Mining

Video Surveillance/
Analysis

Batch

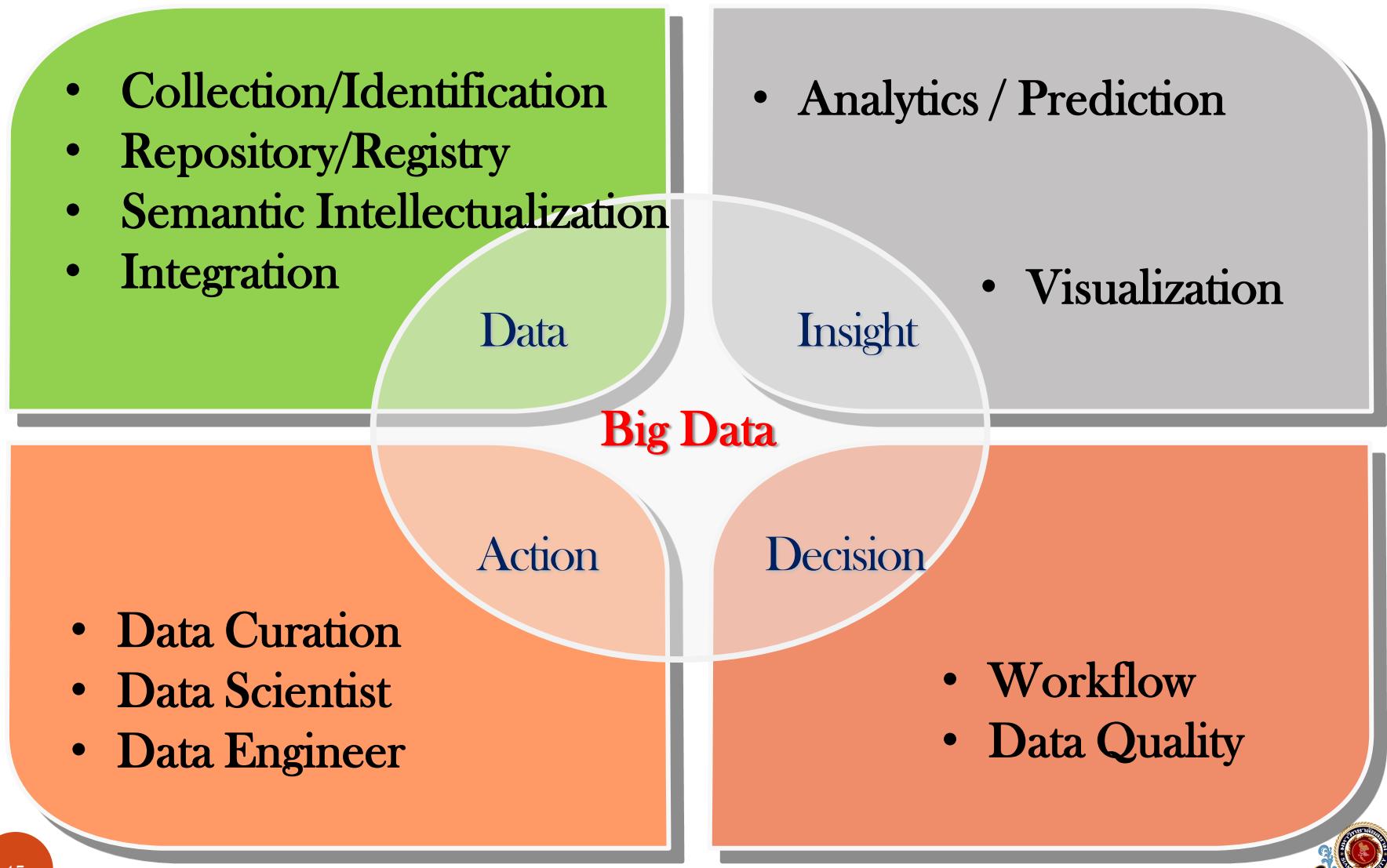
Structured

Semi-structured

Unstructured

Data Variety

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 even 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.

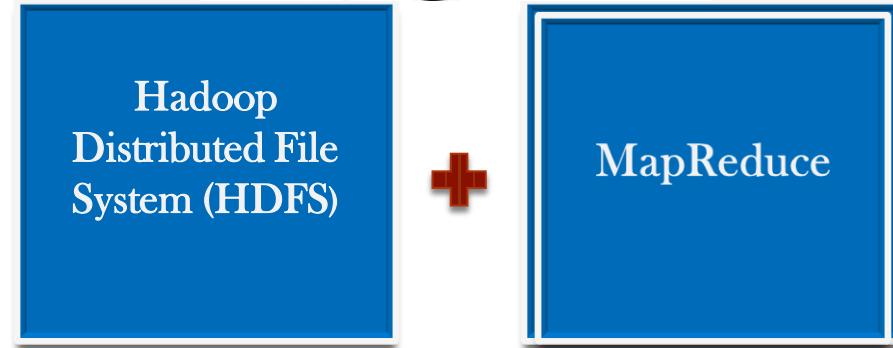
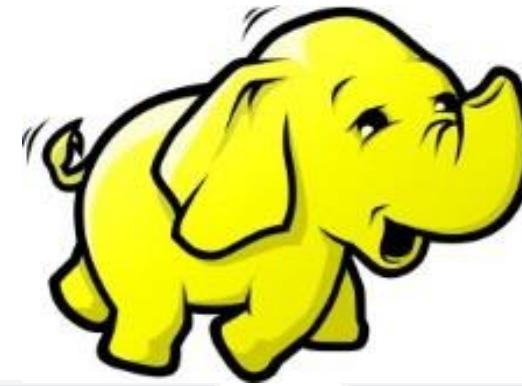


What is Apache Hadoop?

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

- ✓ Scalable
- ✓ Fault tolerant
- ✓ Open source

Core Hadoop Components



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

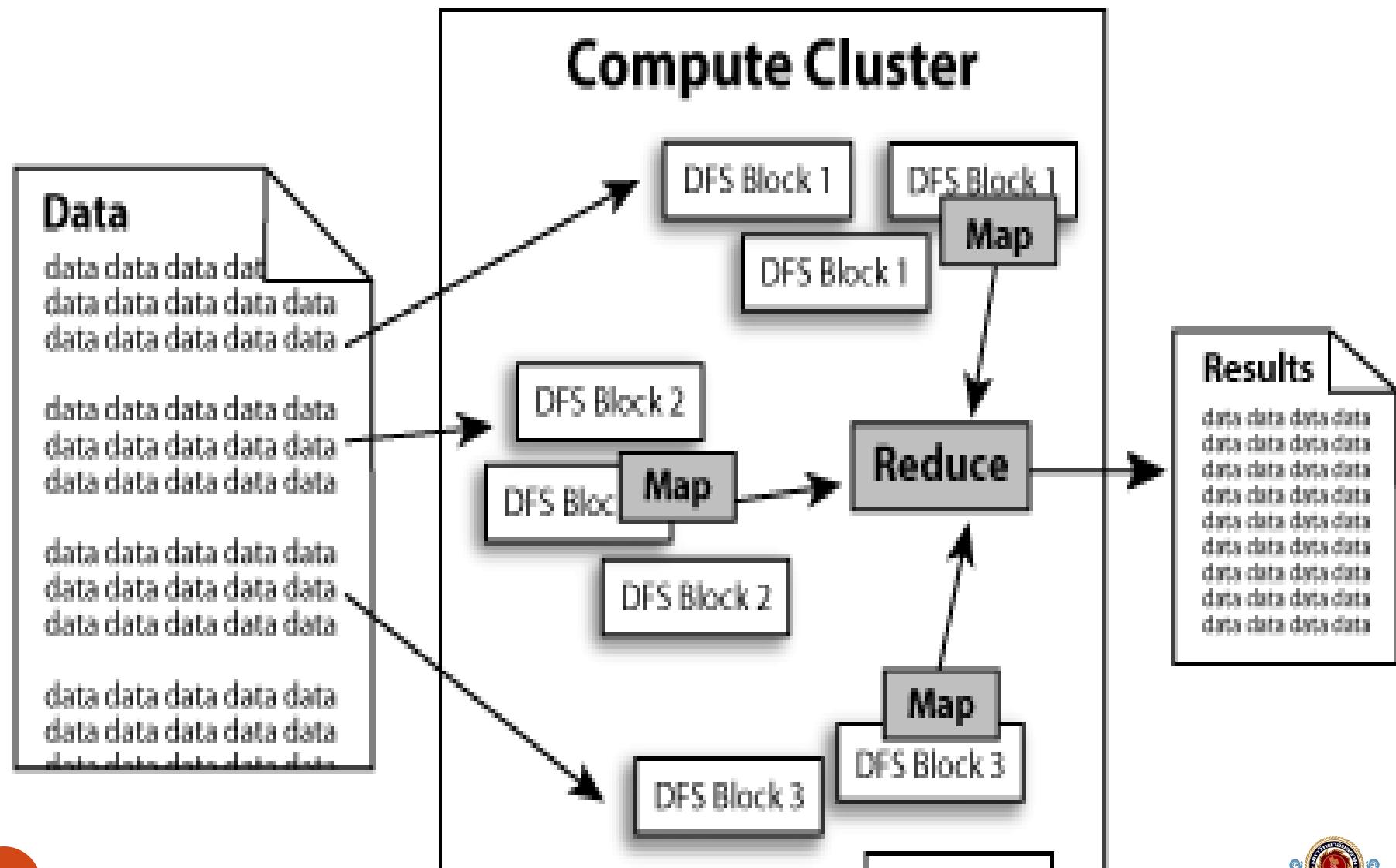


Hadoop Cluster

- 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

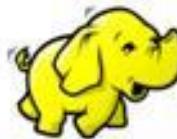


Hadoop: Assumptions

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



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/>

The screenshot shows a web browser displaying the Apache Software Foundation's download mirrors page for Hadoop. The page features the Apache feather logo and the text "The Apache Software Foundation". A navigation bar at the top includes links for Foundation, Projects, People, Get Involved, Download, and Support Apache. Below the navigation bar, a search bar and a "Home > Dyn" link are visible. The main content area displays a list of mirror sites for Hadoop, starting with "Apache Download Mirrors" and "We suggest the following mirror site for your download: <http://mirror.cogentco.com/pub/apache/hadoop/common/>". It also lists other mirror sites and notes to use backup mirrors for PGP and MD5 signatures.

Apache Download Mirrors

We suggest the following mirror site for your download:

<http://mirror.cogentco.com/pub/apache/hadoop/common/>

Other mirror sites are suggested below. Please use the backup mirrors only to download PGP and MD5 signatures to verify your downloads or if no other mirrors are working.

HTTP

<http://mirror.cogentco.com/pub/apache/hadoop/common/>

<http://apache.petsads.us/hadoop/common/>

<http://apache.osuosl.org/hadoop/common/>

<http://mirror.tcpdiag.net/apache/hadoop/common/>



Prerequisites

- Supported Platforms
 - Linux, Mac OS/X, Windows, and Solaris
 - Commodity hardware
- Required Software
 - JavaTM 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

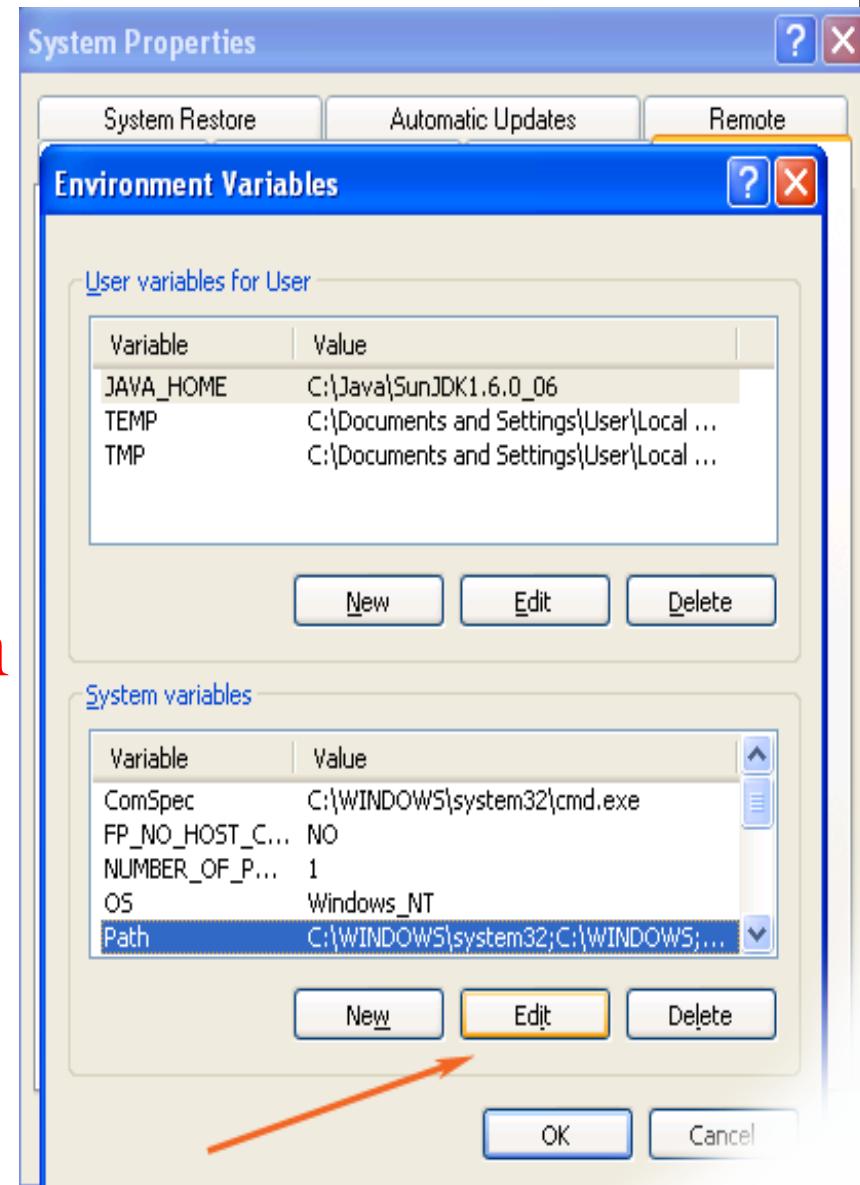
- Additional requirements for Windows include:
 - Cygwin - Required for shell support in addition to the required software above.
 - <http://www.cygwin.com/>

The screenshot shows a web browser displaying the Cygwin installation page at www.cygwin.com/install.html. The page features a large red 'Cygwin' logo and the tagline 'Get that *Linux* feeling - on Windows!'. A sidebar on the left contains links for Cygwin, Cygwin/X, Community, Documentation, and API Reference. The main content area includes sections for 'Installing and Updating Cygwin Packages', 'Installing and Updating Cygwin for 32-bit versions of Windows', and 'Installing and Updating Cygwin for 64-bit versions of Windows'. A circular orange button in the bottom left corner contains the number '29'.



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

```
E -/ssh
$ ssh-keygen.exe
Generating public/private rsa key pair.
Enter file in which to save the key </home/User/.ssh/id_rsa>: ↗
Created directory '/home/User/.ssh'.
Enter passphrase (empty for no passphrase): ↗
Enter same passphrase again: ↗
Your identification has been saved in /home/User/.ssh/id_rsa.
Your public key has been saved in /home/User/.ssh/id_rsa.pub.
The key fingerprint is:
df:0b:34:58:e3:81:44:c6:e8:e0:af:ea:be:21:a8:5b User@BAHCLIENT

User@BAHCLIENT ~
$ cd .ssh

User@BAHCLIENT ~/.ssh
$ ls -l
total 5
-rw----- 1 User None 1675 Mar 10 09:09 id_rsa
-rw-r--r-- 1 User None 396 Mar 10 09:09 id_rsa.pub

User@BAHCLIENT ~/.ssh
$ cat id_rsa.pub >> authorized_keys

User@BAHCLIENT ~/.ssh
```



HDFS

Hadoop Distributed File System (HDFS)



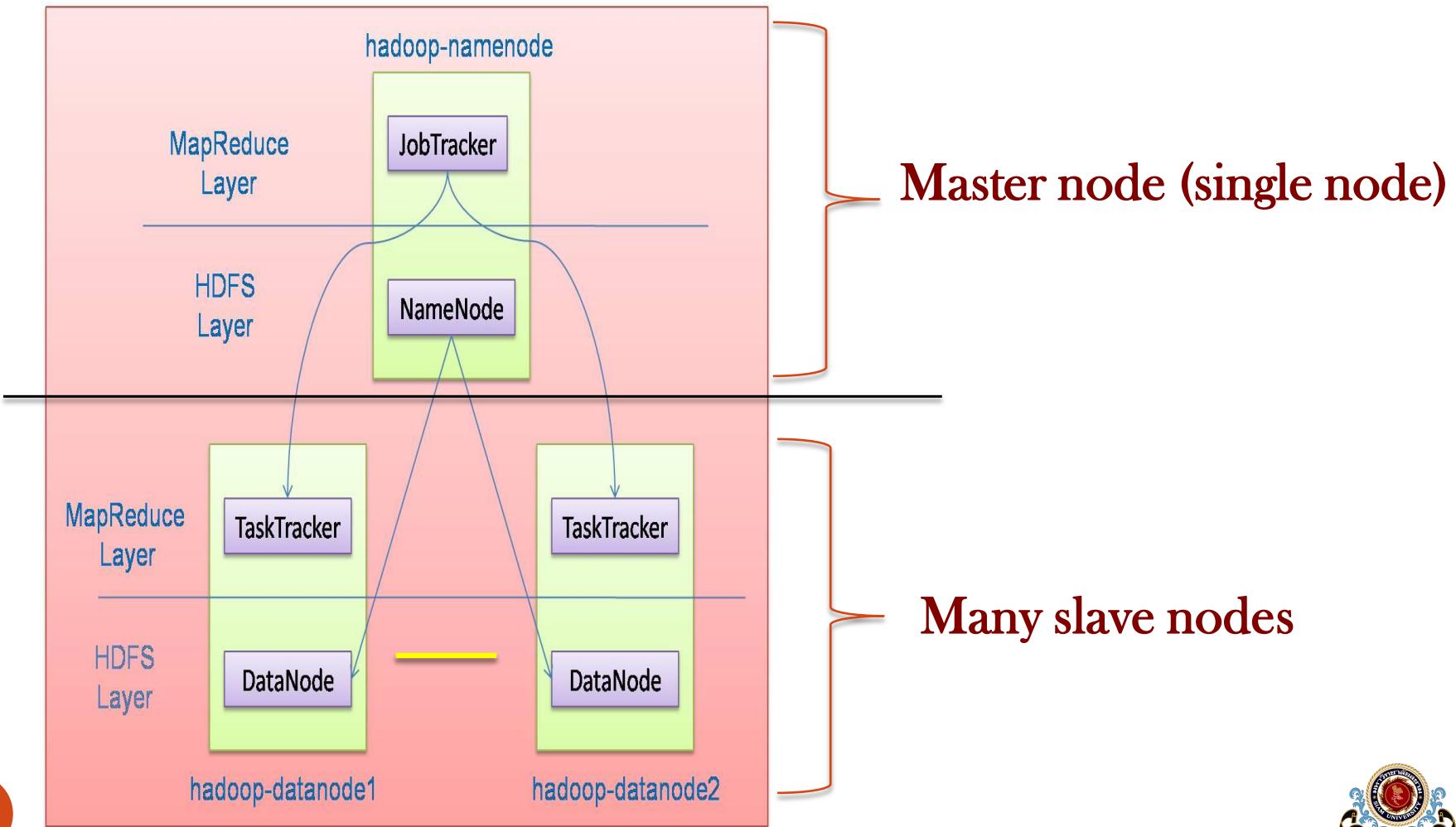
What is HDFS*?

- 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.



Hadoop Master/Slave Architecture

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



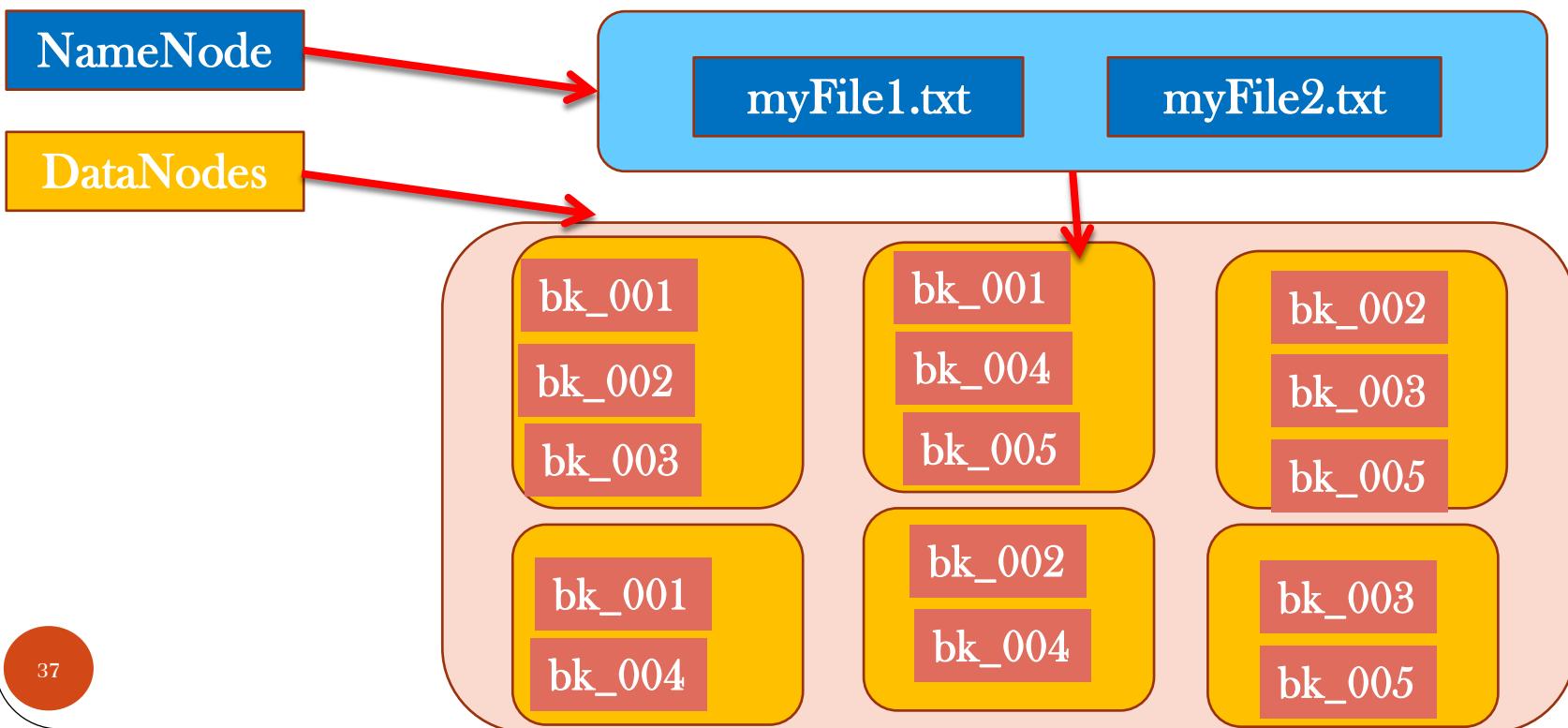
Hadoop Server Role

- **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)**
 - SNN 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.

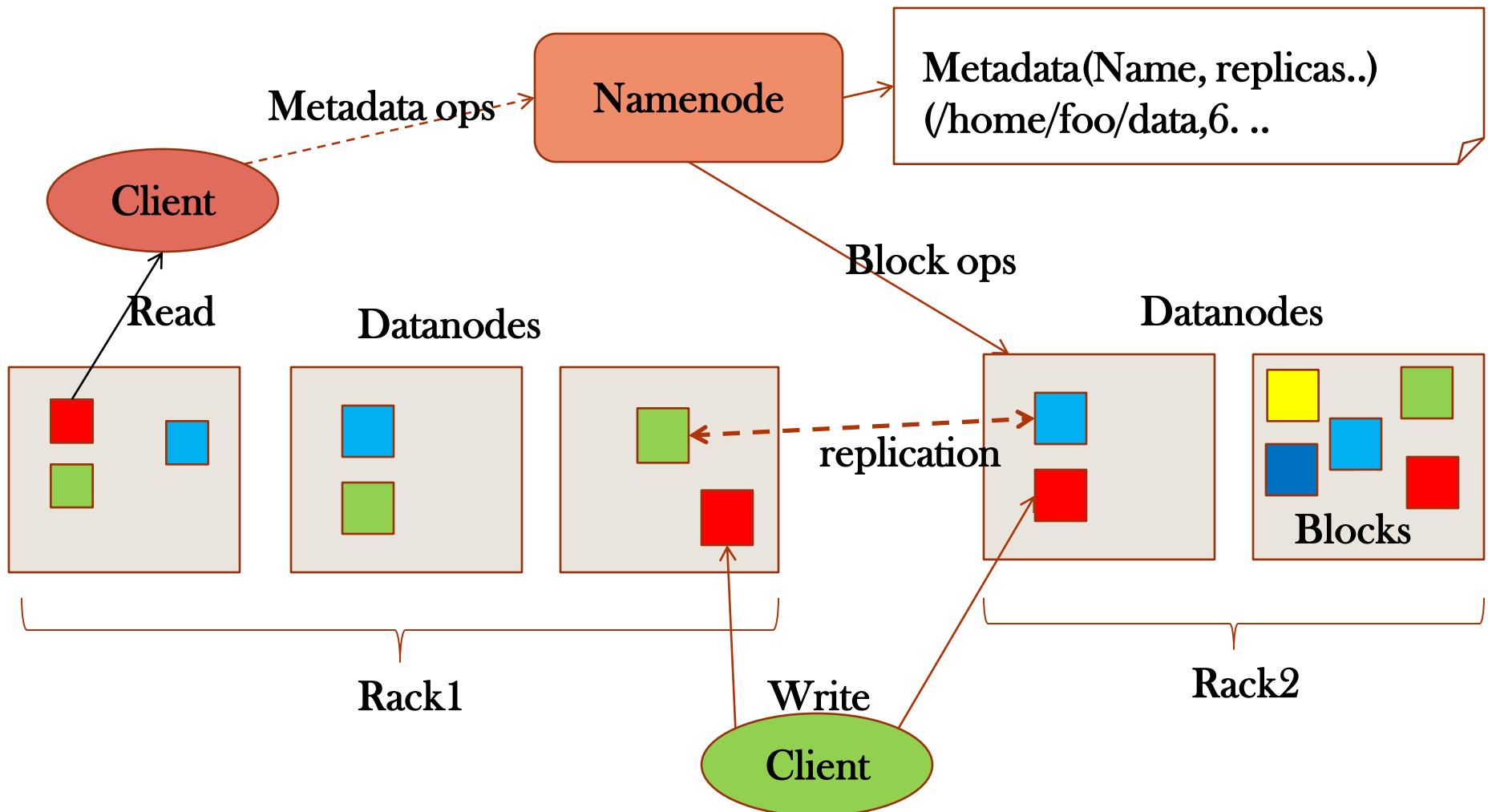


NameNode and DataNodes

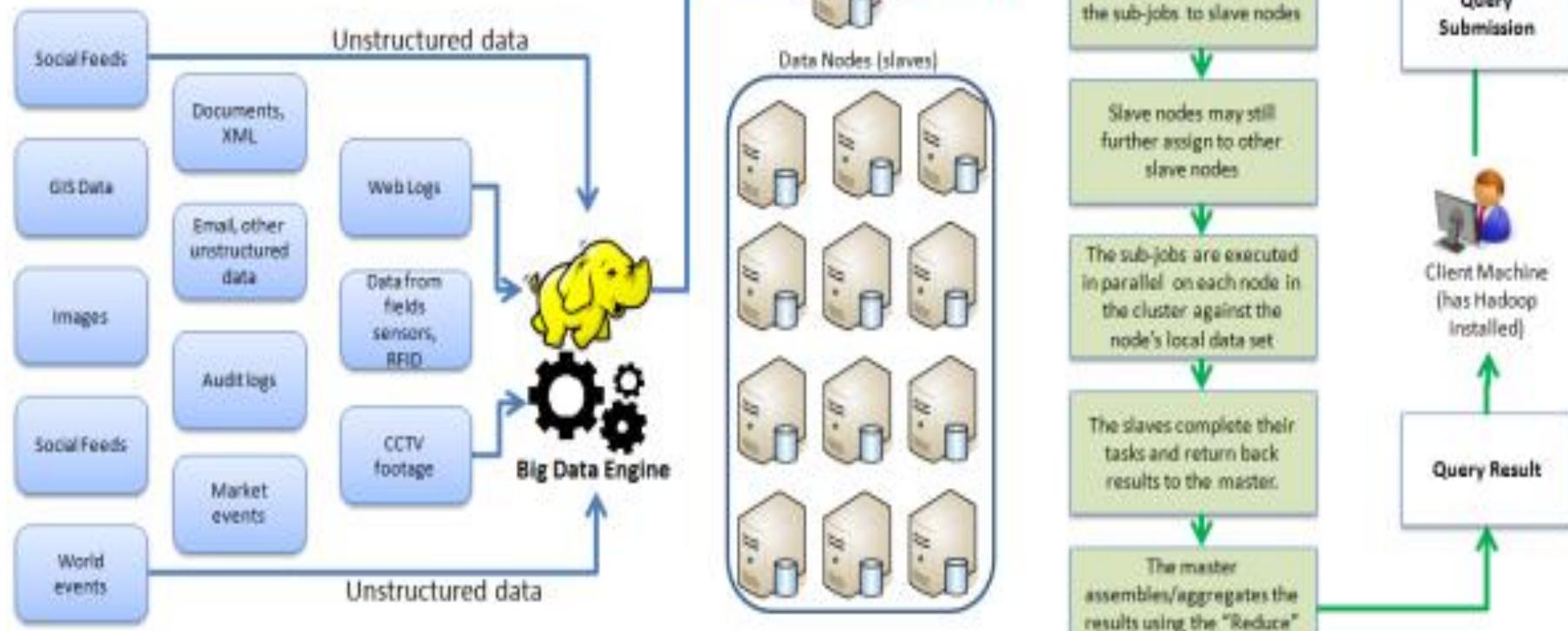
- Example:
 - NameNode holds metadata for the two files
 - **myFile1.txt : bk_001, bk_002, bk_003**
 - **myFile2.txt: bk_004, bk_005**
 - DataNodes hold the actual blocks 64 - 256 MB size.



HDFS Architecture



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



Data is chopped and stored on the HDFS – Hadoop Distributed File System.

Data in the HDFS is scattered over numerous nodes for built in fault tolerance.

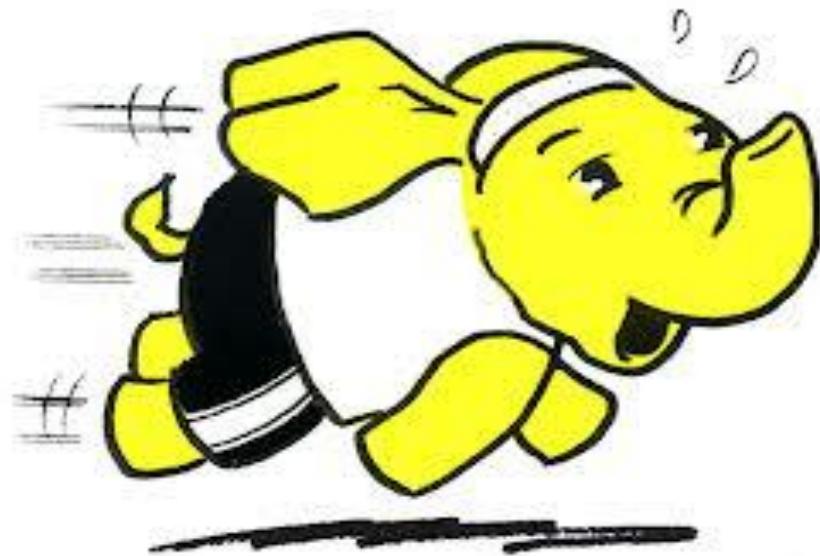
HDFS has one master/name node and numerous slave/data nodes.

Name node stores meta data and data nodes store data blocks.

Name nodes and data Nodes reside on commodity servers i.e. x86.

Each node/server offers local storage and computation.

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

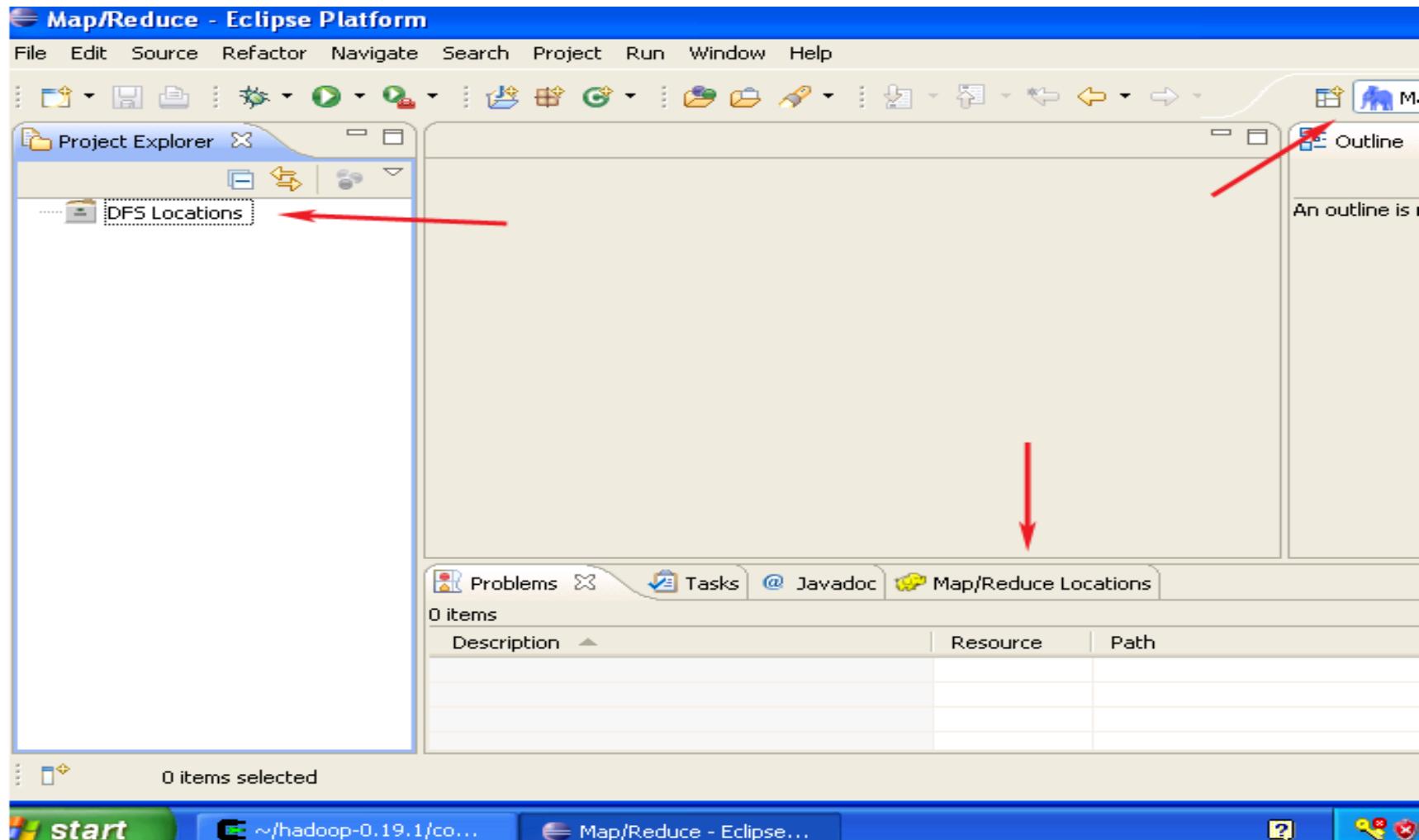
- Start NameNode
 - cd hadoop-0.19.1
bin/hadoop namenode
- Start JobTracker
 - cd hadoop-0.19.1
bin/hadoop jobtracker
- Start DataNode
 - cd hadoop-0.19.1
bin/hadoop datanode
- Start TaskTracker
 - cd hadoop-0.19.1
bin/hadoop tasktracker

The screenshot shows a Windows desktop environment with four terminal windows and one Cygwin window.

- Terminal 1:** Shows logs for the NameNode starting up. It includes log entries for FSNamesystem audit, permission changes, and the JobTracker registration from port 50010.
- Terminal 2:** Shows logs for the JobTracker starting up. It includes log entries for the DataNode assignment and its initial delay.
- Terminal 3:** Shows logs for the DataNode starting up. It includes log entries for the DataNode's initial delay and startup.
- Terminal 4:** Shows logs for the TaskTracker starting up. It includes log entries for the WebApplicationContext startup, the Jetty server, and the SecondaryNameNode's startup.
- Cygwin Window:** Shows the Eclipse icon, indicating the Java Development Kit (JDK) is installed.



Run 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.

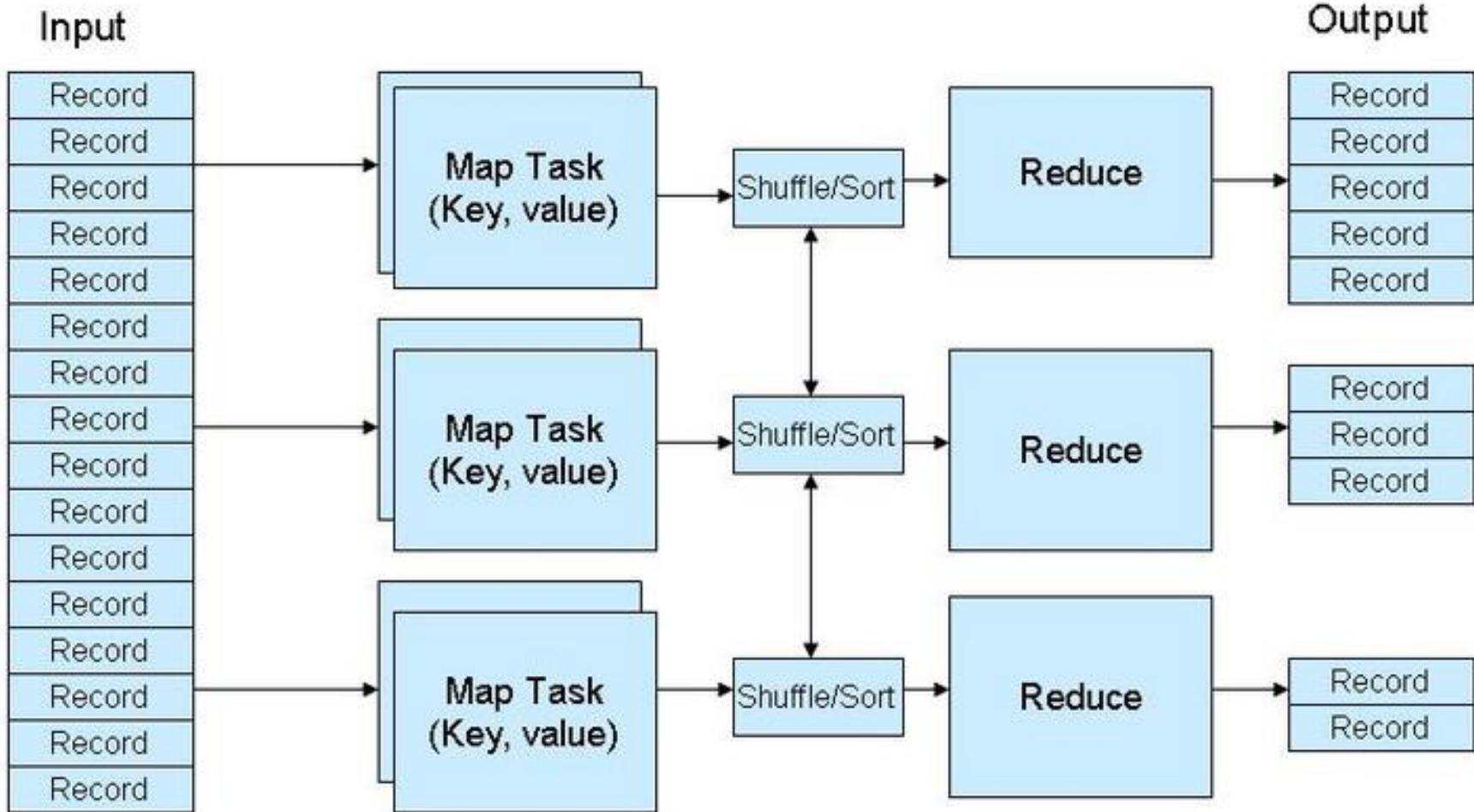


Map & Reduce Functions

- **The Mapper:**
 - Input: Read input from disk.
 - Output: Create a single or multiple pairs of data $\langle\text{key}, \text{value}\rangle$, known as intermediate pairs
- **The Reducer:**
 - Input: a list of $\langle\text{key}, \text{value}\rangle$ pairs with a unique key.
 - Output: Single or multiple of $\langle\text{key}, \text{values}\rangle$

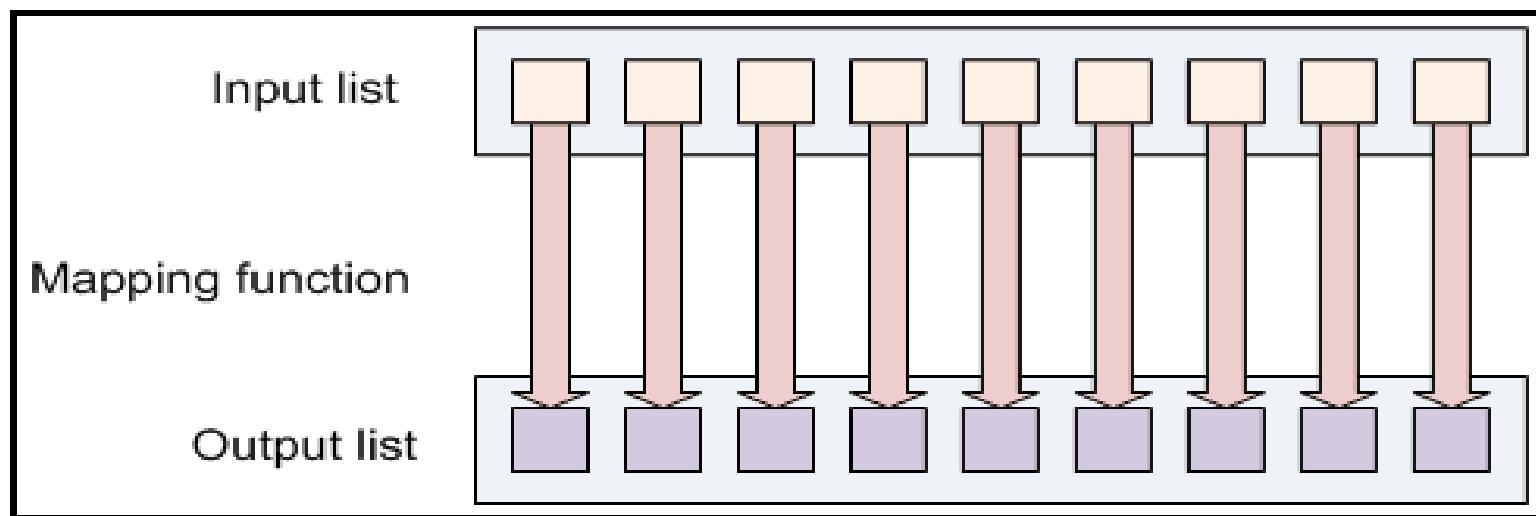


MapReduce Architecture *



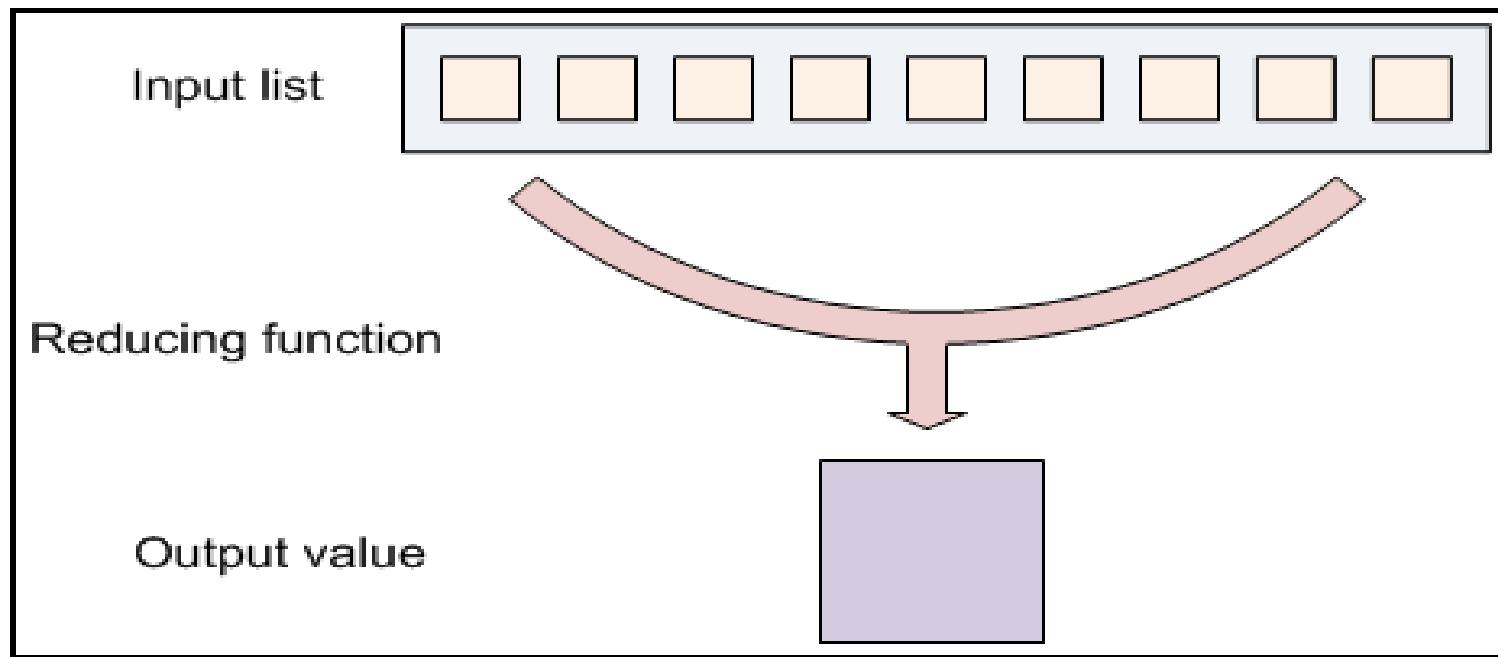
Mapping Lists

- 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.

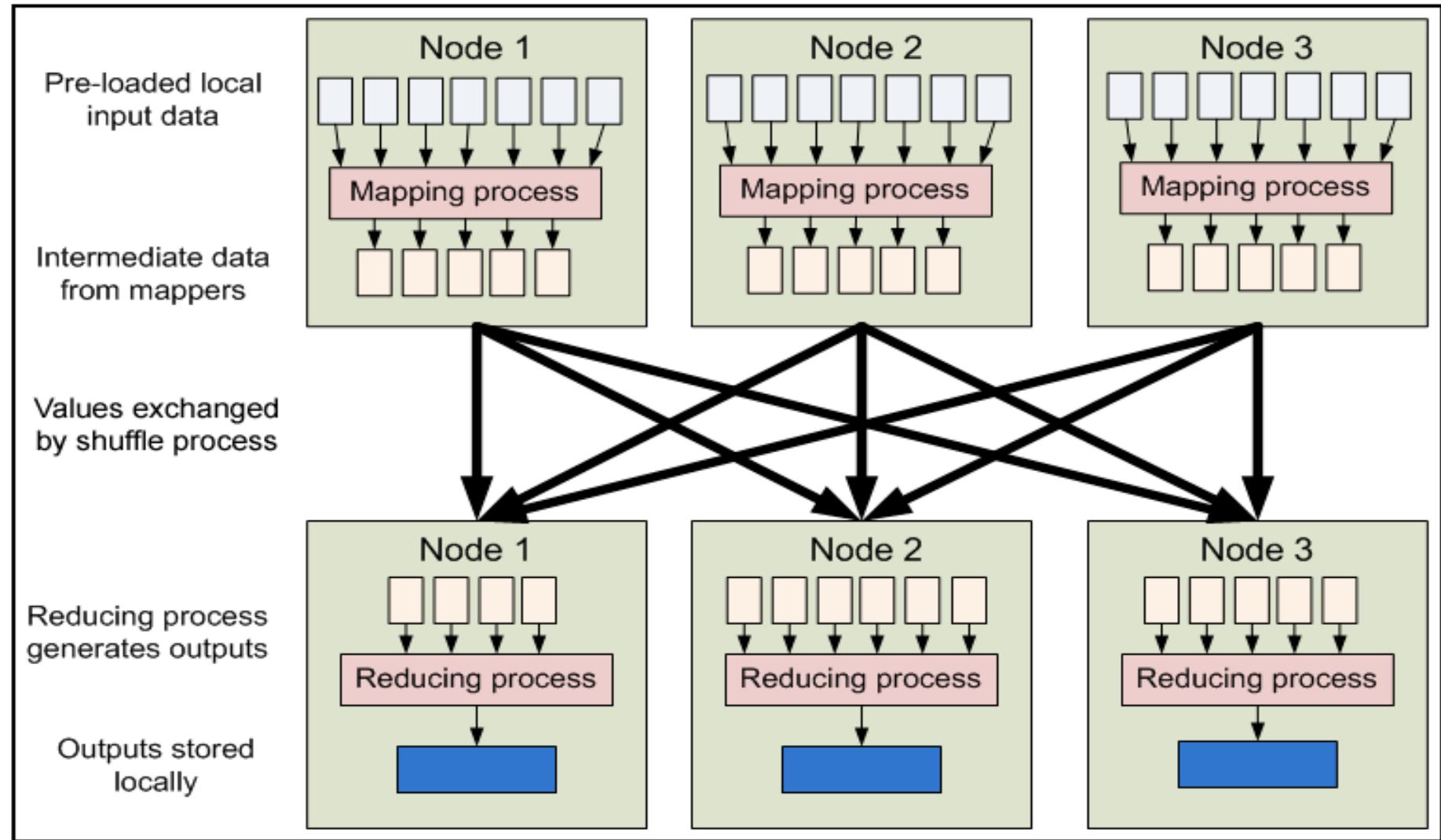


Reducing Lists

- 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.file.buffer.size	Buffer size used to read/write (intermediate) sequence files	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 filesystem	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

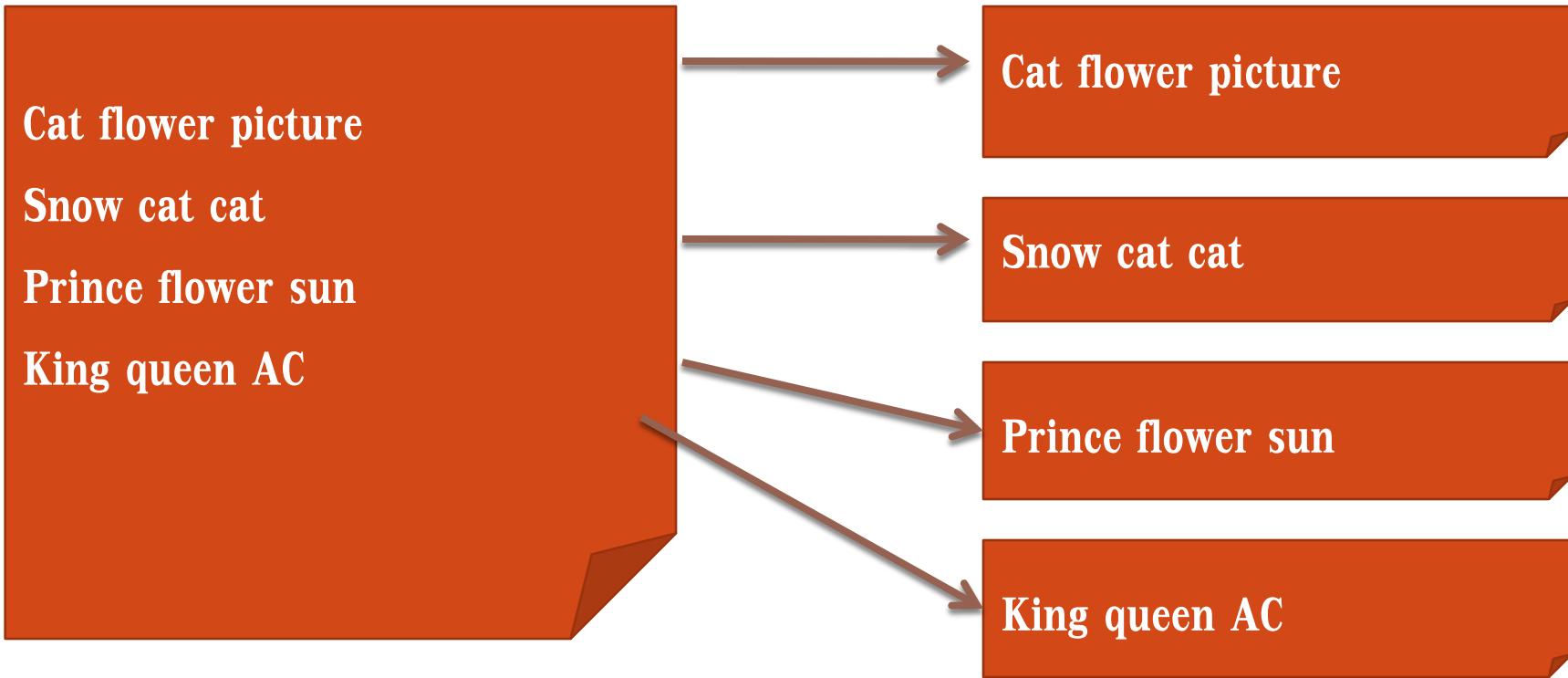
Prince flower sun

King queen AC



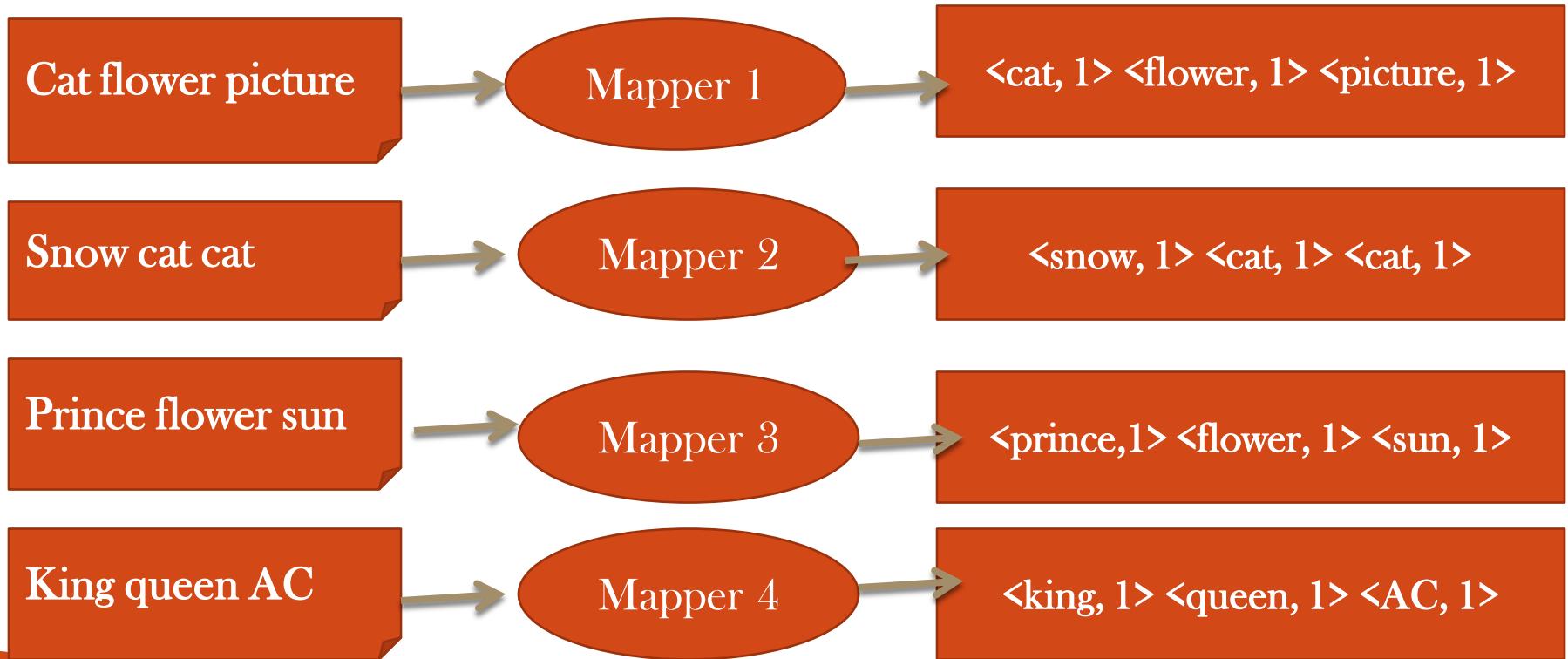
Example - Word Count-1

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



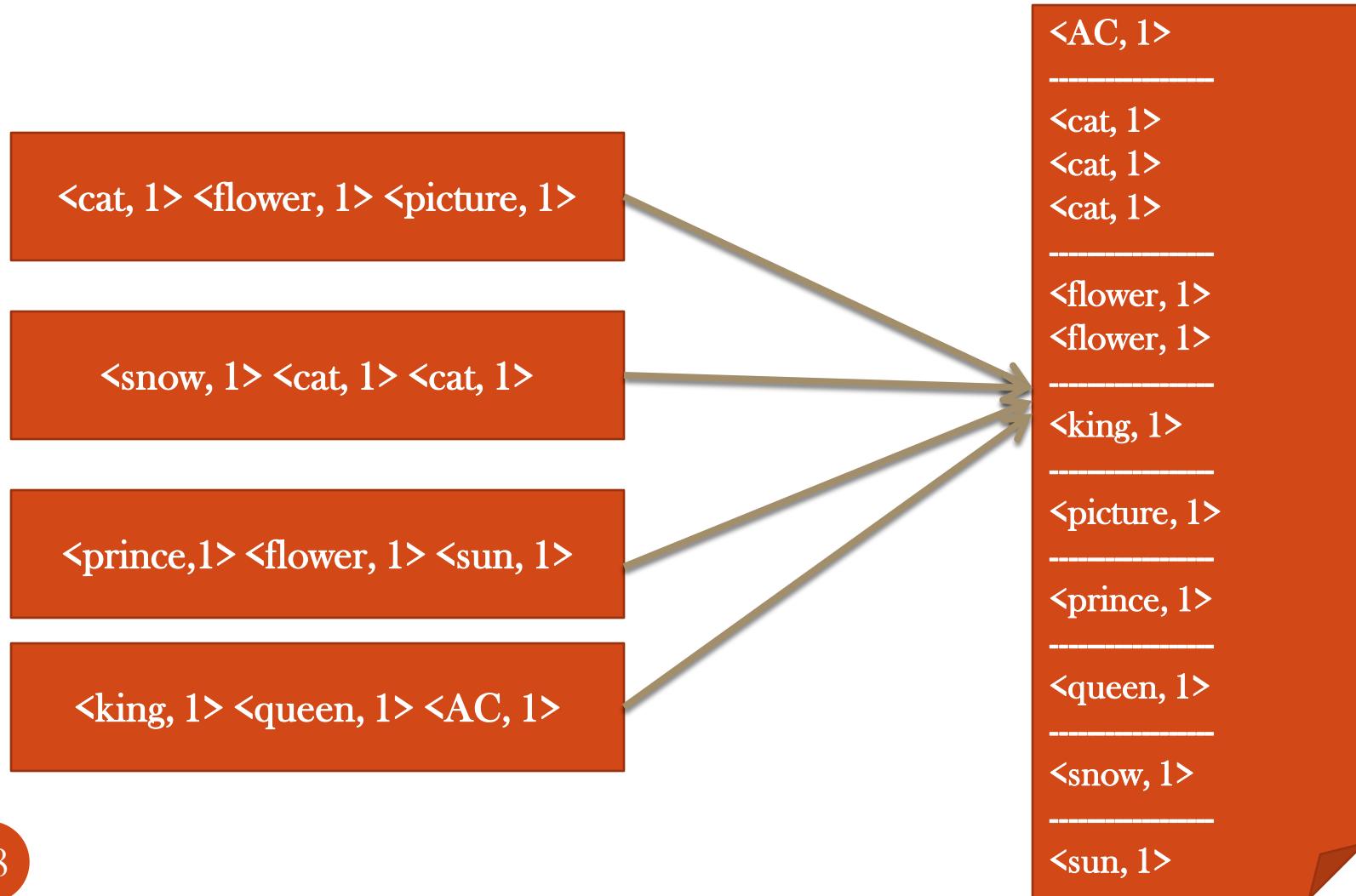
Example - Word Count-2

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



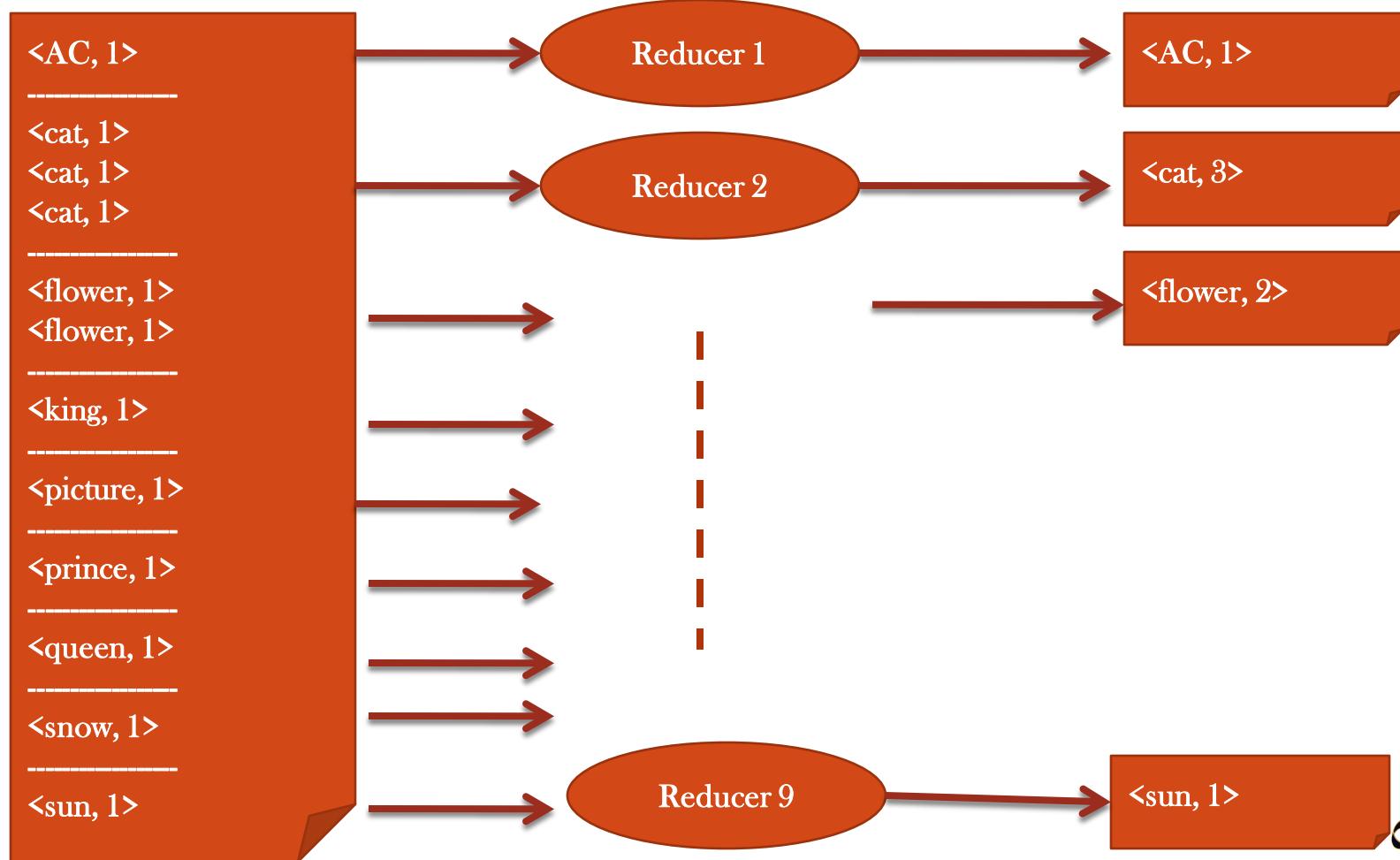
Example - Word Count-3

- Step3: Wait for all Map workers to finish.



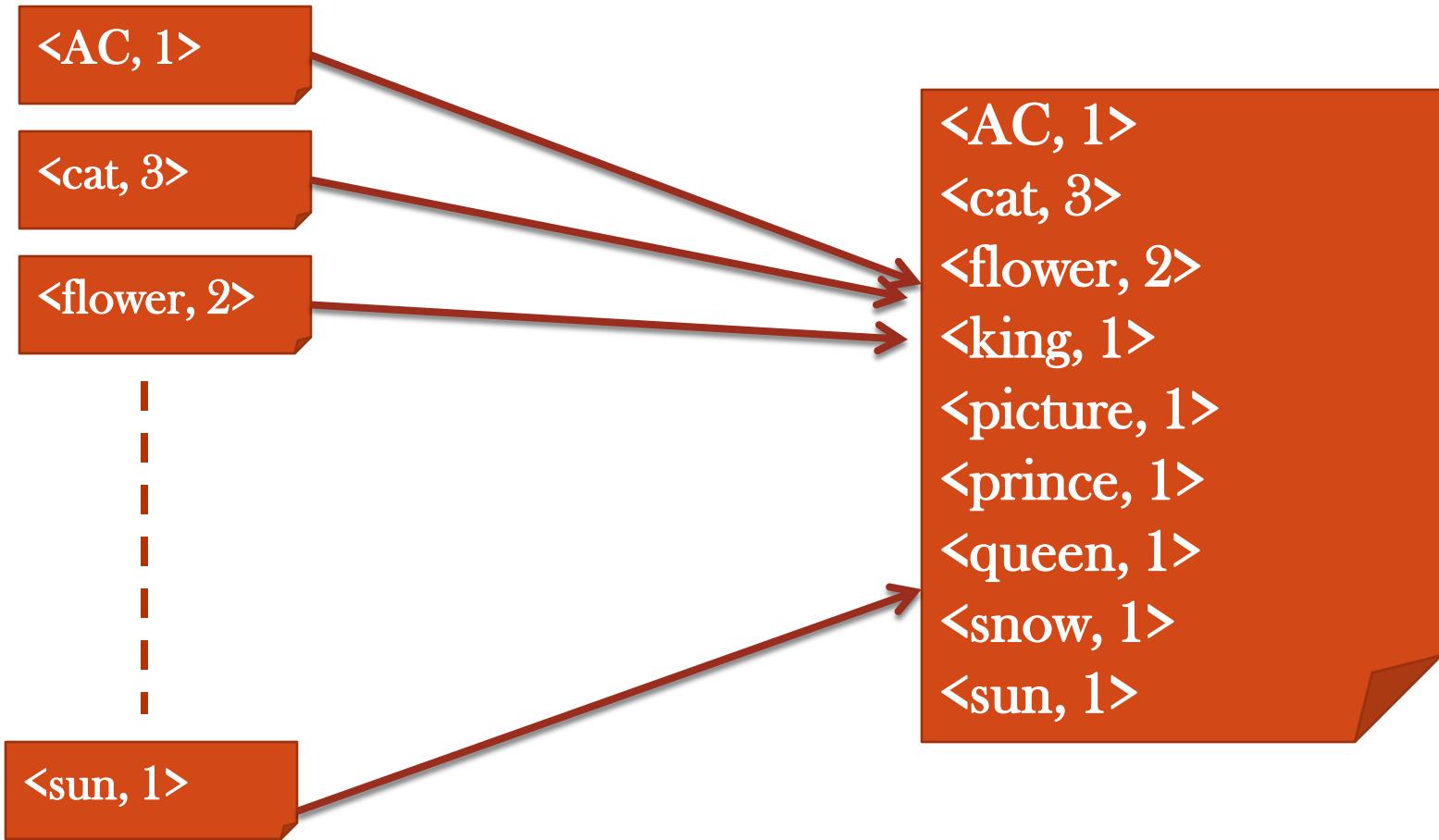
Example - Word Count-4

- Step4: The partition intermediate pairs into R regions.

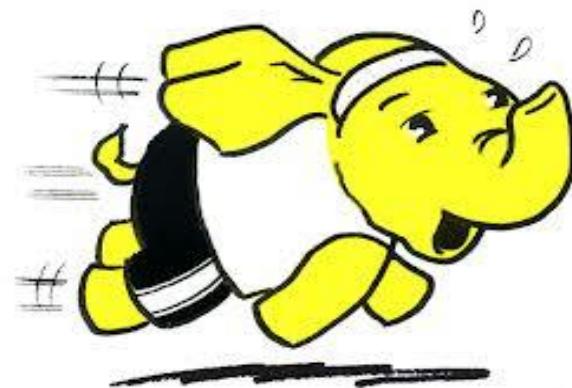
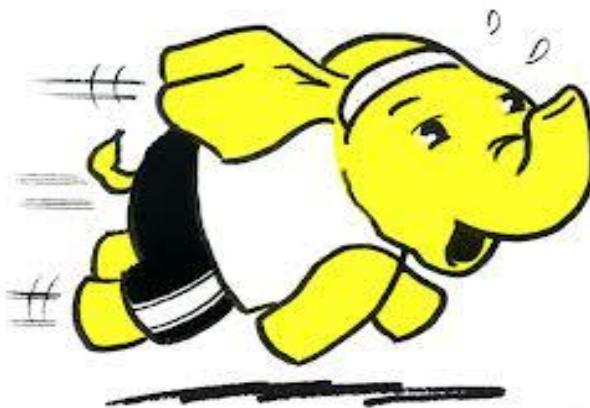


Example - Word Count-5

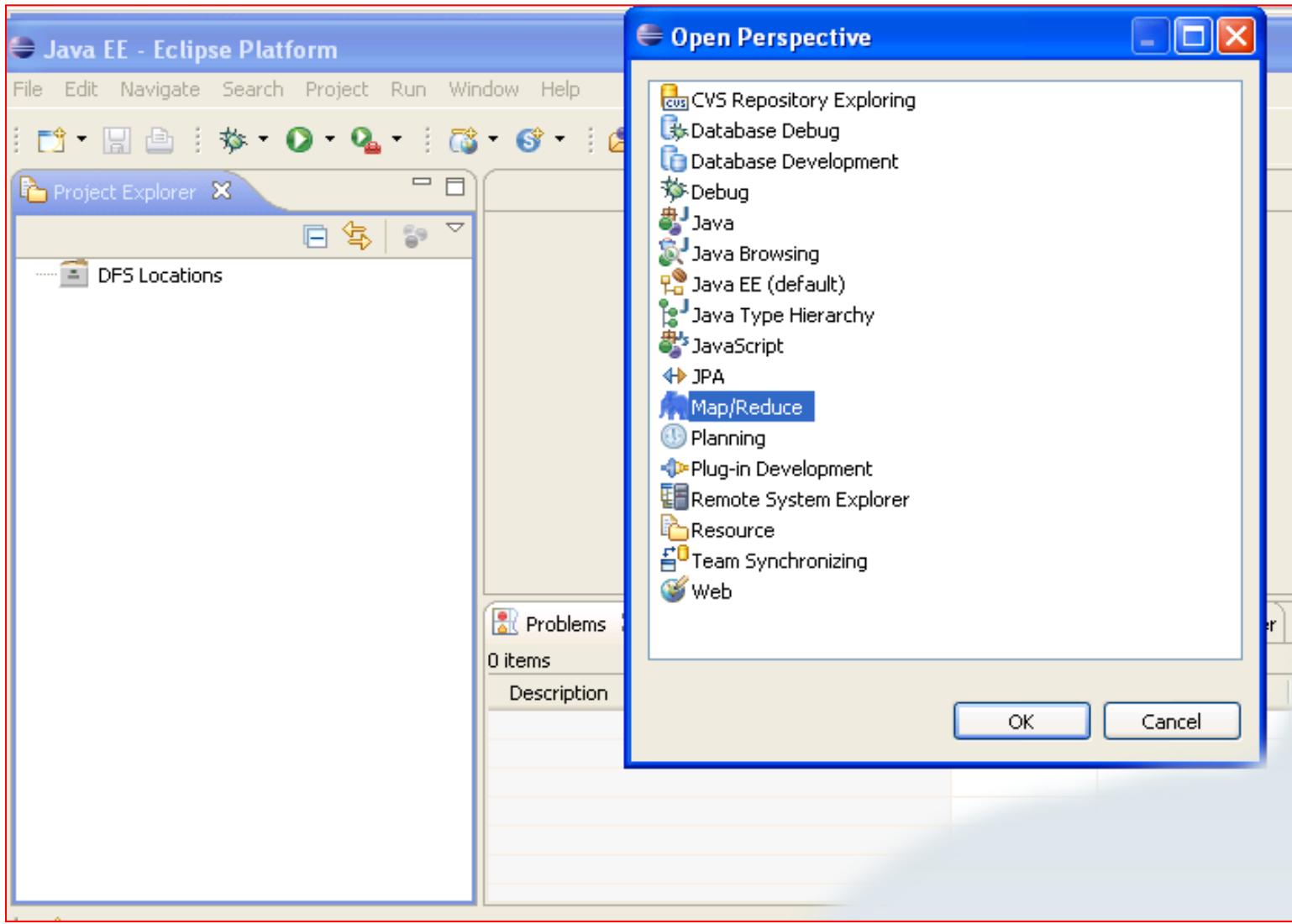
- Step5: Write the output of reduce workers to file(s).



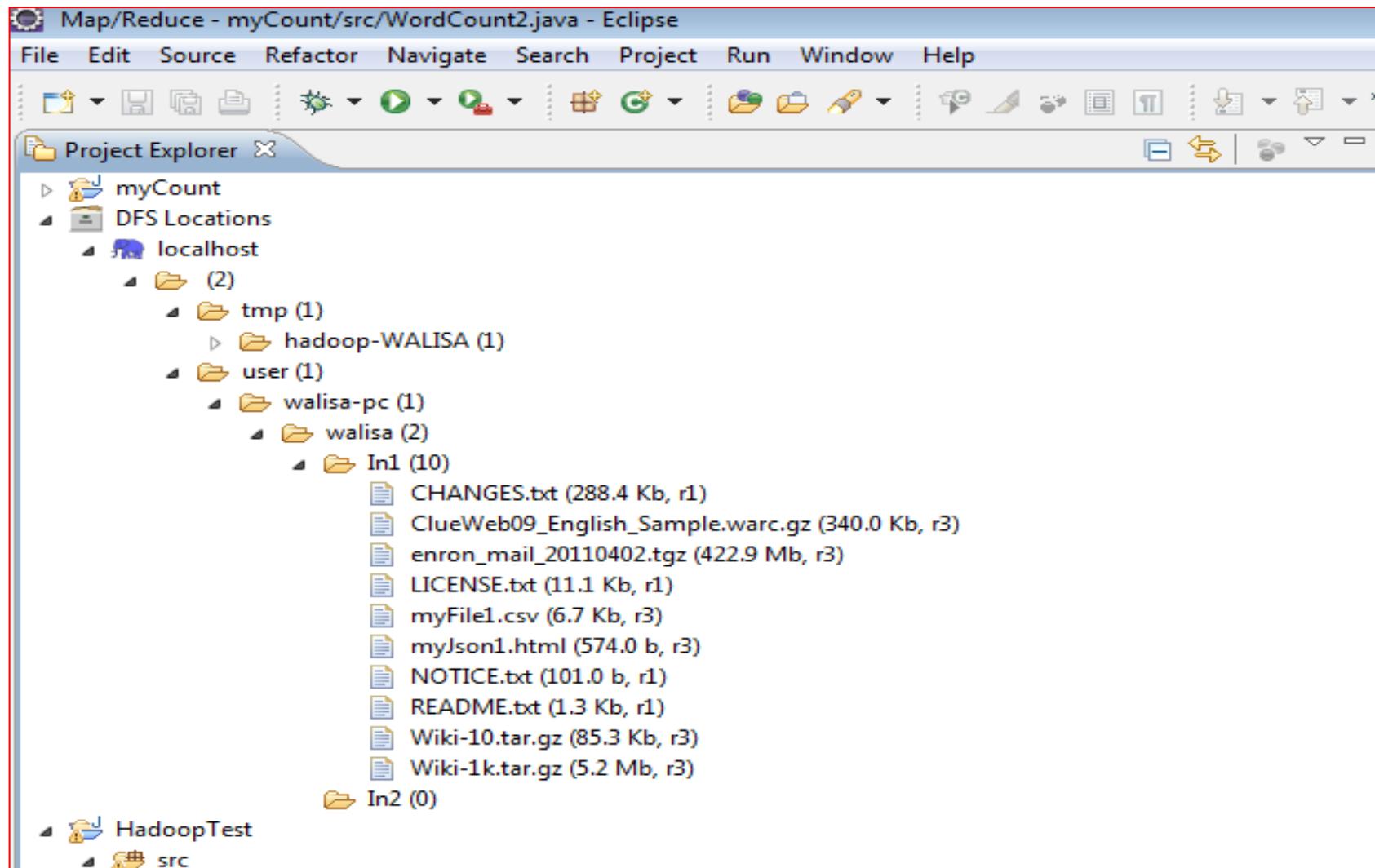
Getting Started Hadoop-3



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

HDFS:/user/com-pc1/com/ln - Windows Internet Explorer
EN English (United States) Help
http://127.0.0.1:50075/browseDirectory.jsp?dir=%2Fuser%2Fcom-pc1%2Fcom%2Fln&namenodeInfoPort=50070

File Edit View Favorites Tools Help
Convert Select
Favorites Suggested Sites Web Slice Gallery
HDFS... User man... HDFS/... Hadoop... The Face... SIGSNA... Enron Em... Tastes, Ti... Internet E... Connecti...
Find: thailand Previous Next Options

Contents of directory /user/com-pc1/com/ln

Goto : /user/com-pc1/com/ln go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
CHANGES.txt	file	280.32 KB	1	64 MB	2012-02-16 16:40	rw-r--r--	com-pc1\com	supergroup
LICENSE.txt	file	11.09 KB	1	64 MB	2012-02-16 16:40	rw-r--r--	com-pc1\com	supergroup
NOTICE.txt	file	0.1 KB	1	64 MB	2012-02-16 16:40	rw-r--r--	com-pc1\com	supergroup
README.txt	file	1.33 KB	1	64 MB	2012-02-16 16:40	rw-r--r--	com-pc1\com	supergroup
movies.csv	file	171.08 KB	1	64 MB	2012-02-17 11:33	rw-r--r--	com-pc1\com	supergroup
myBig1.htm	file	6.84 MB	1	64 MB	2012-02-18 10:28	rw-r--r--	com-pc1\com	supergroup
myData.txt	file	0.02 KB	1	64 MB	2012-02-17 17:20	rw-r--r--	com-pc1\com	supergroup
myFC1.htm	file	251.99 KB	1	64 MB	2012-02-18 10:40	rw-r--r--	com-pc1\com	supergroup
ratings.dat	file	23.45 MB	1	64 MB	2012-02-17 11:28	rw-r--r--	com-pc1\com	supergroup

Go back to DFS home

Local logs

[Log](#) directory
[Hadoop](#), 2012.

Internet | Protected Mode: On
11:07 AM 2/18/2012



Run the result-2

Hadoop NameNode 127.0.0.1:9100 - Windows Internet Explorer
EN English (United States) ? Help
File Edit View Favorites Tools Help
Convert Select
Favorites Suggested Sites Web Slice Gallery
Hadoop User man... HDFS/us... HDFS/us... Hadoop The Face... SIGSNA r... Enron Em... Tastes, Ti... Internet E... Connecti...
Find: thailand Previous Next Options
Compiled: Fri Nov 14 03:12:29 UTC 2008 by ndaley
Upgrades: There are no upgrades in progress.

Browse the filesystem
[Namenode Logs](#)

Cluster Summary

90 files and directories, 54 blocks = 144 total. Heap Size is 8.78 MB / 992.31 MB (0%)

Configured Capacity	:	221.61 GB
DFS Used	:	85.62 MB
Non DFS Used	:	42.82 GB
DFS Remaining	:	178.71 GB
DFS Used%	:	0.04 %
DFS Remaining%	:	80.64 %
Live Nodes	:	1
Dead Nodes	:	0

Live Datanodes : 1

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
192.168.193.61	0	In Service	221.61	0.08	42.82	178.71	0.04	—	80.64	54

Dead Datanodes : 0

[Hadoop](#), 2012.

Done Internet | Protected Mode: On 100% 11:10 AM 2/18/2012



Run the result-3

HDFS:/user/com-pc1/com/output4/part-00000 - Windows Internet Explorer
EN English (United States) Help
File Edit View Favorites Tools Help
Convert Select
Favorites Suggested Sites Web Slice Gallery
HDFS:/user/com-pc1/c... User management syste... HDFS:/user/com-pc1/c... HDFS:/user/com-pc1/c... Connecting...
Find: thailand Previous Next Options
File: /user/com-pc1/com/output4/part-00000

Goto : \user\com-pc1\com\output\go

[Go back to dir listing](#)
[Advanced view/download options](#)

[View Next chunk](#)

```
1000::1023::5::975041651 1
1000::1029::3::975041859 1
1000::1036::4::975040964 1
1000::1104::5::975042421 1
1000::110::5::975040841 1
1000::1196::3::975040841 1
1000::1198::5::975040841 1
1000::1200::4::975041125 1
1000::1201::5::975041025 1
1000::1210::5::975040629 1
1000::1214::4::975040919 1
1000::1220::4::975040964 1
1000::1221::4::975040919 1
1000::1222::3::975040964 1
1000::1233::5::975040841 1
1000::1240::4::975040919 1
1000::1282::4::975041651 1
1000::1287::4::975040919 1
1000::1291::5::975040964 1
1000::1299::4::975042337 1
1000::1304::4::975040919 1
1000::1356::3::975041346 1
1000::1374::5::975041125 1
1000::1387::4::975040841 1
1000::1610::5::975040841 1
```

Done Internet | Protected Mode: On 100% 11:16 AM 2/18/2012



Run the result-4

The screenshot shows the Eclipse IDE interface with the title "Map/Reduce - part-00000 (25.4 Mb, r1) - Eclipse".

- Project Explorer:** Shows the file structure under "DFS Locations / localhost".
 - In (9):** CHANGES.txt, LICENSE.txt, movies.csv, myBig1.htm, myData.txt, myFC1.htm, NOTICE.txt, ratings.dat, README.txt.
 - Output Directories:** output1 through output8, each containing part-00000 (25.4 Mb, r1).
- Outline:** An outline is not available.
- Console:** Displays the following Java stack trace:

```
<terminated> TestDriver [Java Application] C:\Program Files\Java\jre1.6.0\bin\javaw.exe (Feb 16, 2012 5:30:05 PM)
12/02/16 17:30:06 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the sa
12/02/16 17:30:06 WARN mapred.JobClient: No job jar file set. User classes may not be found. See JobConf(Class) or JobConf#setJar(String)
org.apache.hadoop.mapred.InvalidInputException: Input path does not exist: file:/C:/myJAVA/HadoopTest/In
    at org.apache.hadoop.mapred.FileInputFormat.listStatus(FileInputFormat.java:179)
    at org.apache.hadoop.mapred.FileInputFormat.getSplits(FileInputFormat.java:190)
    at org.apache.hadoop.mapred.JobClient.submitJob(JobClient.java:782)
    at org.apache.hadoop.mapred.JobClient.runJob(JobClient.java:1127)
    at TestDriver.main(TestDriver.java:41)
```



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" = "([^\"]*) ([^\"]*) ([^\"]*) (-|\\[^\n\\]*\\) ([^ \"]* |\"[^\"]*\") (-|[0-9]*) (-|[0-9]*)(?: ([^\n\"]* |\".*\") ([^\n\"]* |\".*\"))?",
    "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

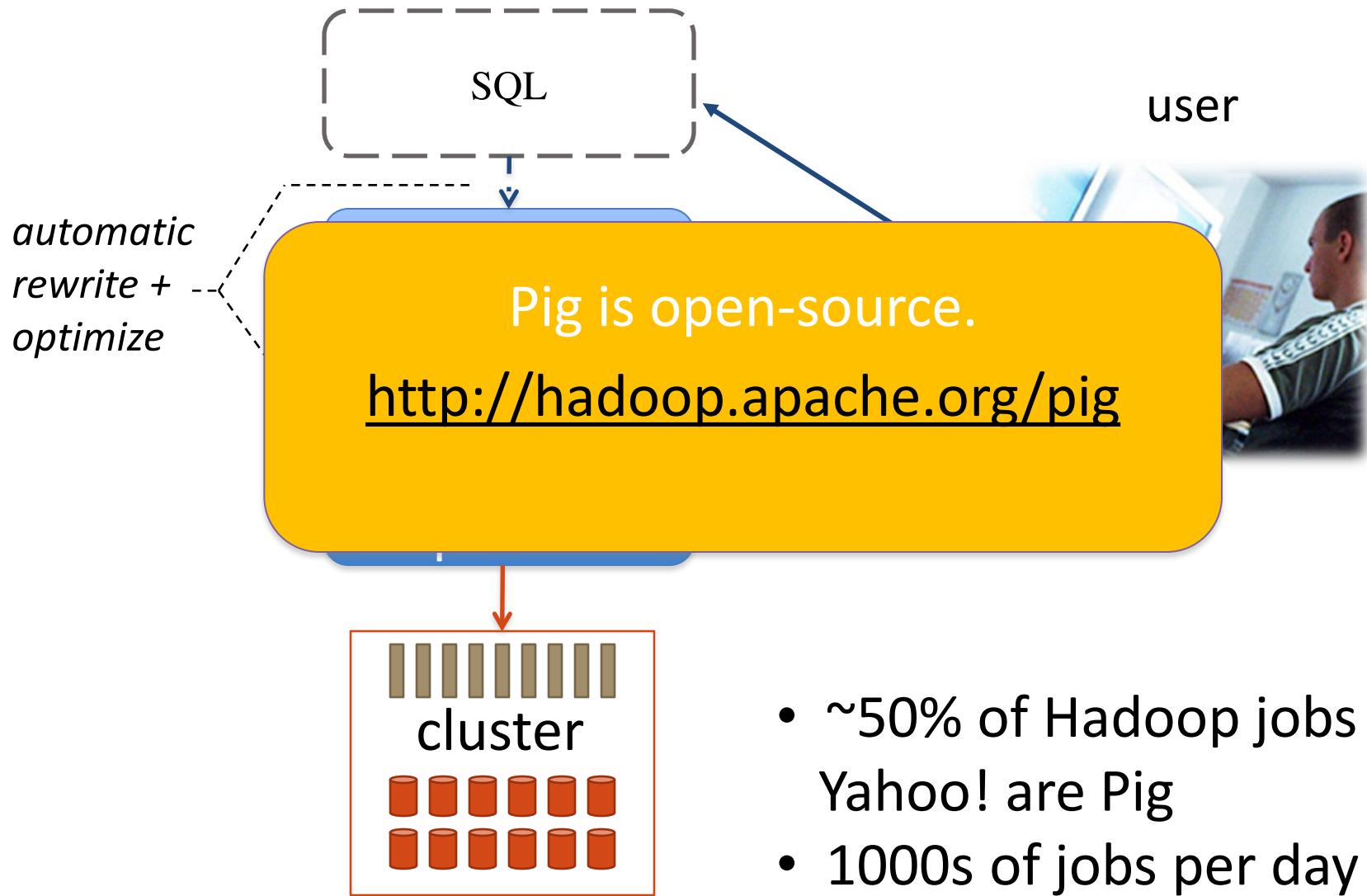
- Usage statistics:
 - Total Users: ~140 (about 50% of engineering !) in the last 1 ½ months
 - Hive Data (compressed): 80 TB total, ~1TB incoming per day
- Job statistics:
 - ~1000 jobs/day
 - ~100 loader jobs/day



Pig



Pig



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.



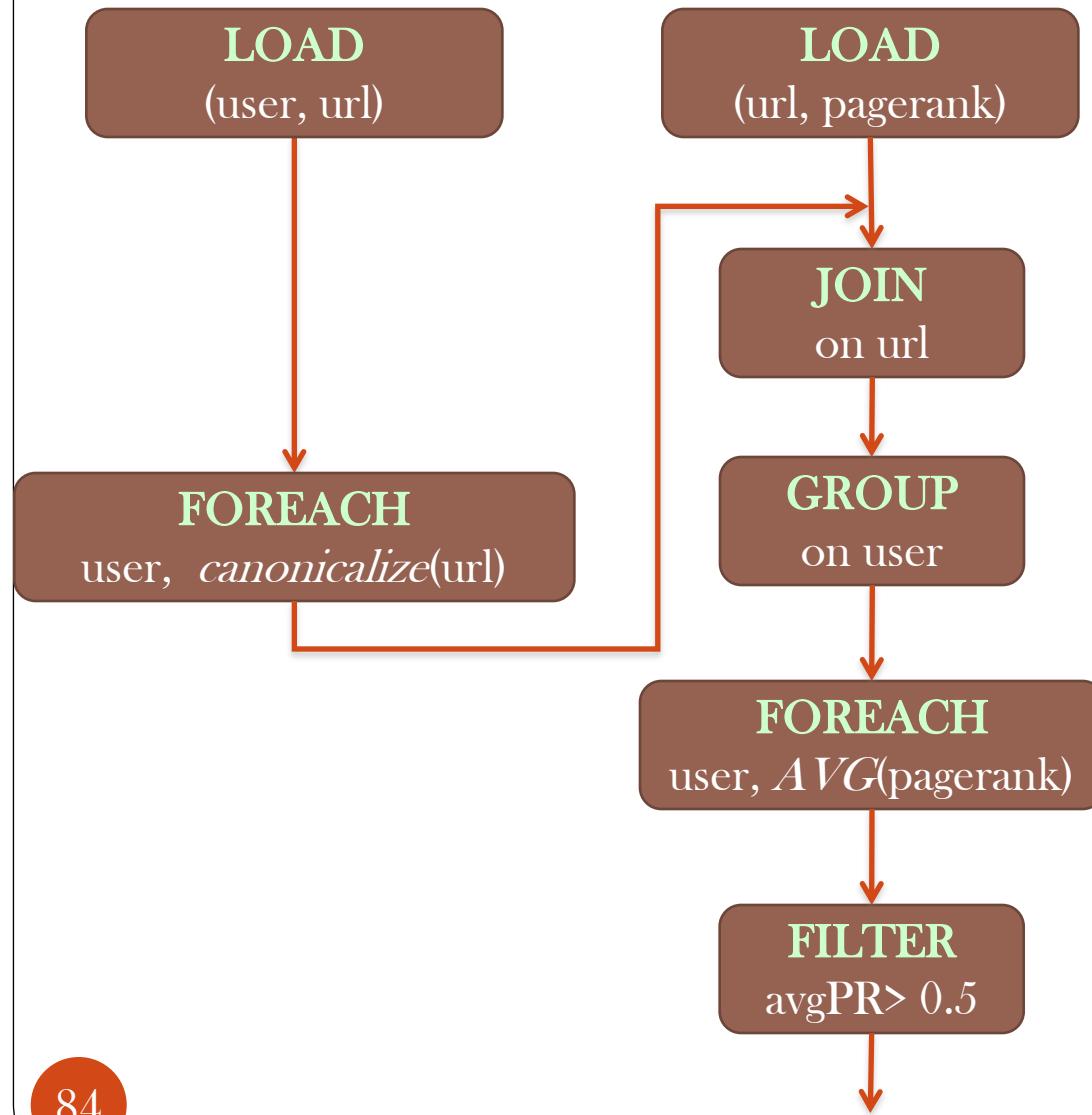
Pig Latin

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

- Increases productivity.
 - In one test 10 lines of Pig Latin \approx 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

- MapReduce 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

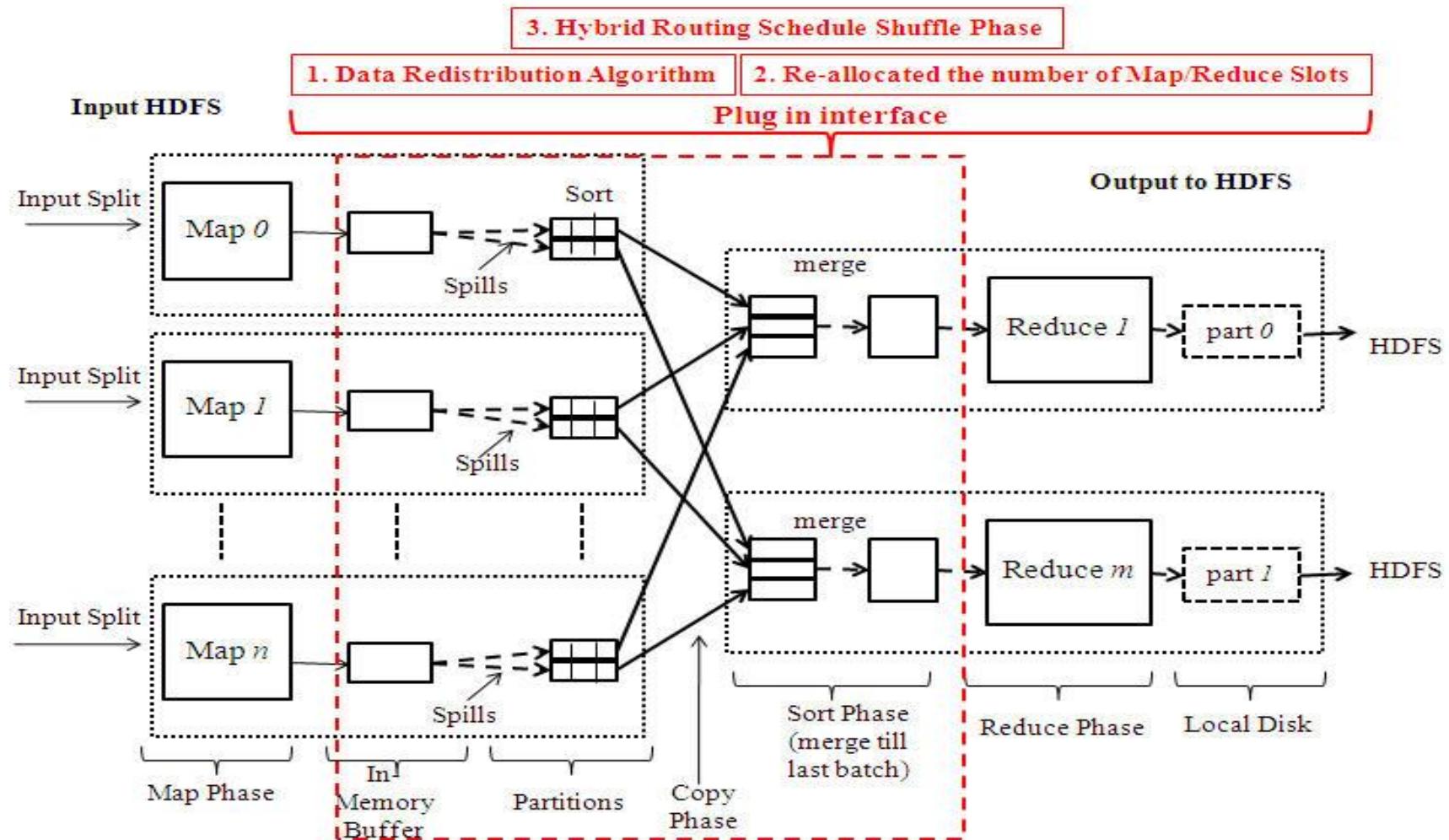


Abstract

- 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



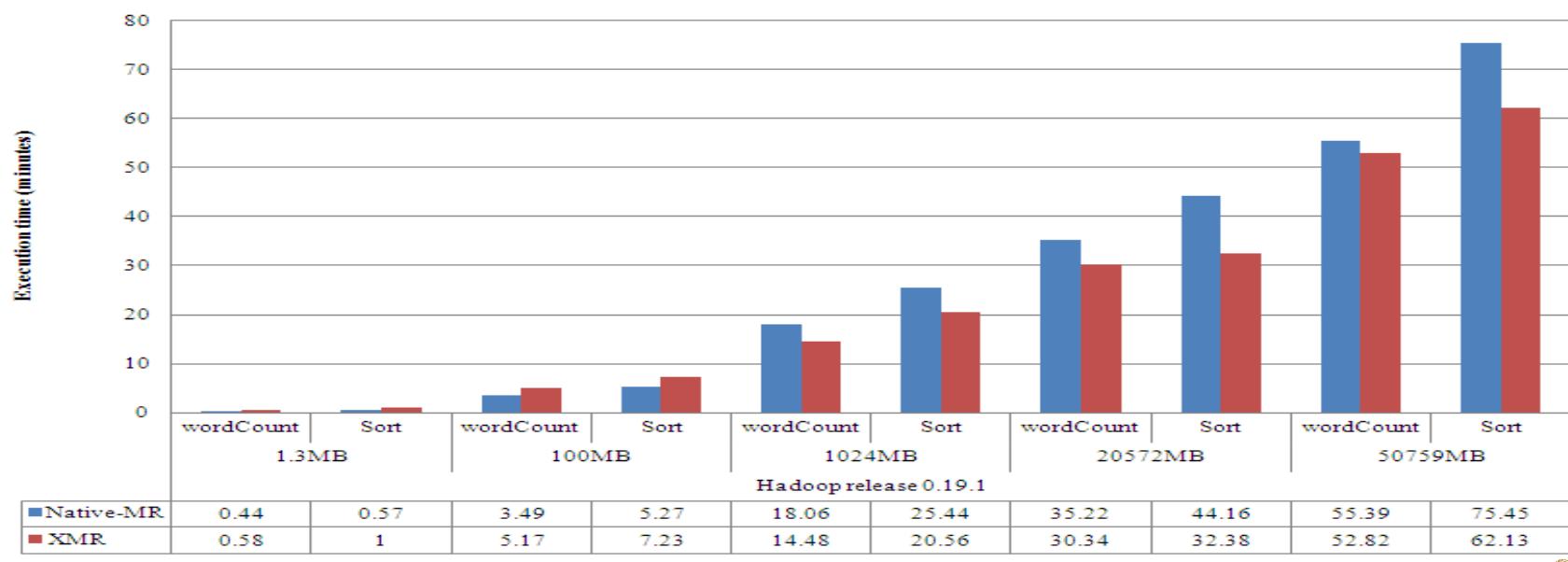
Efficiency Results-1

	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

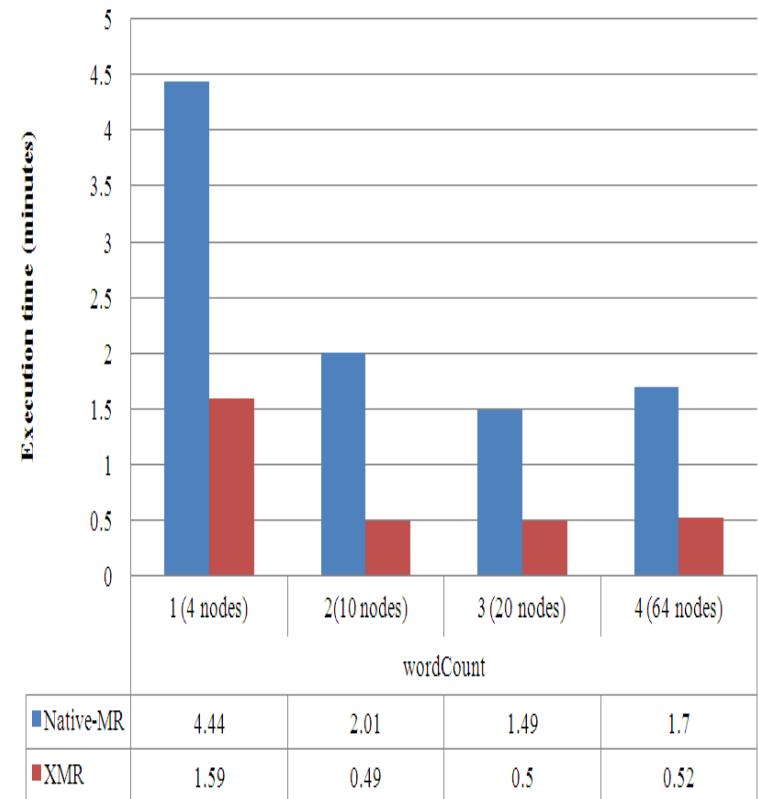
- We ran our program on Hadoop version 0.19.1 testing the three plug-in modules using Hadoop MapReduce. The wordCount and sort applications were run and show the comparison of the execution time of WordCount in native and our program called XMR(eXtensible-MapReduce).



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, Twitter and Flickr)
 - <http://www.isi.edu/integration/people/lerman/downloads.html>
- Public Datasets
 - <http://www.scaleunlimited.com/datasets/public-datasets/>



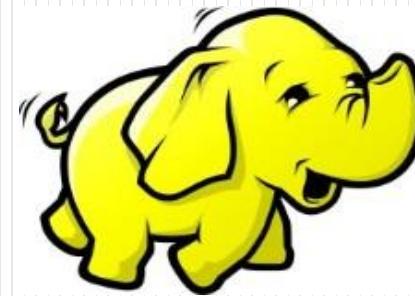
Reference WebSites

- 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

APACHE
HBASE



mahout

