Mahidol University
Wisdom of the Land

BIG DATA PROCESSING

EGCI 466

Week5

Mahidol University
Wisdom of the Land

Early Start

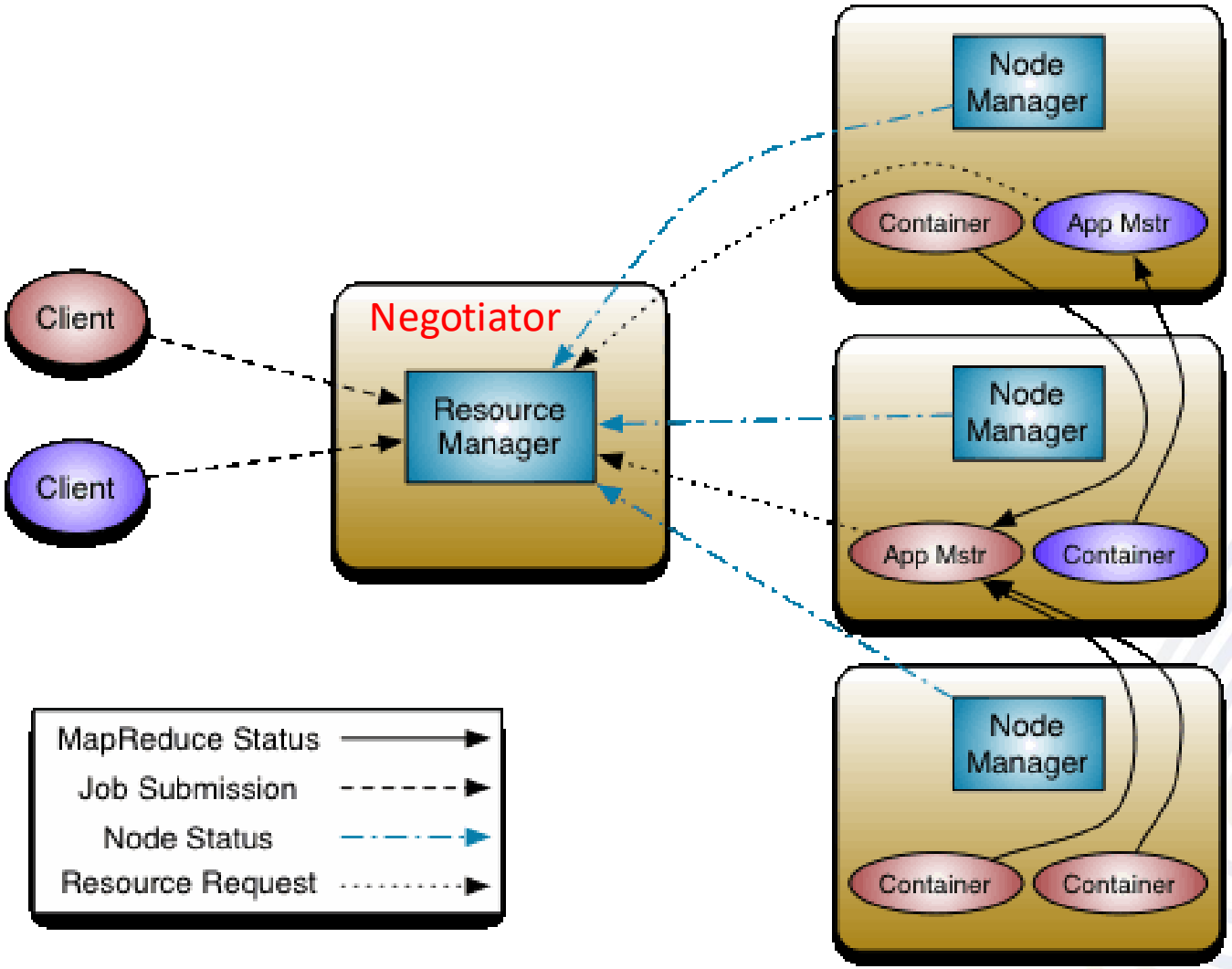
- Cloud Storage
  - Delete all previous dataproc storage (make sure they're deleted)
  - Create your cloud bucket (if you not have one)

- Create a dataproc
  - 1 Master (E2)
  - 2 worker nodes (E2)
  - Storage 50GB/clusters

Pricing: https://cloud.google.com/dataproc/docs/resources/pricing

Mahidol University
Wisdom of the Land

Yarn Architecture



Mahidol University
Wisdom of the Land

Yarn:Benefit

- High resource utilization
- Data → Value
- One data set → Many applications

Mahidol University
Wisdom of the Land

Yarn

- Yet Another Resource Negogiator
- Prepares Hadoop RAM/CPU for
  - Batch
  - Interactive
  - Stream
  - Graph processing

Mahidol University
Wisdom of the Land

HIVE

- Data warehouse software
- Design for managing tabular dataset
- Data analysis

- Scalable and fast and easy to use
- Hive Query Language(HQL) is similar to SQL
- It supports data clasnsing and filtering

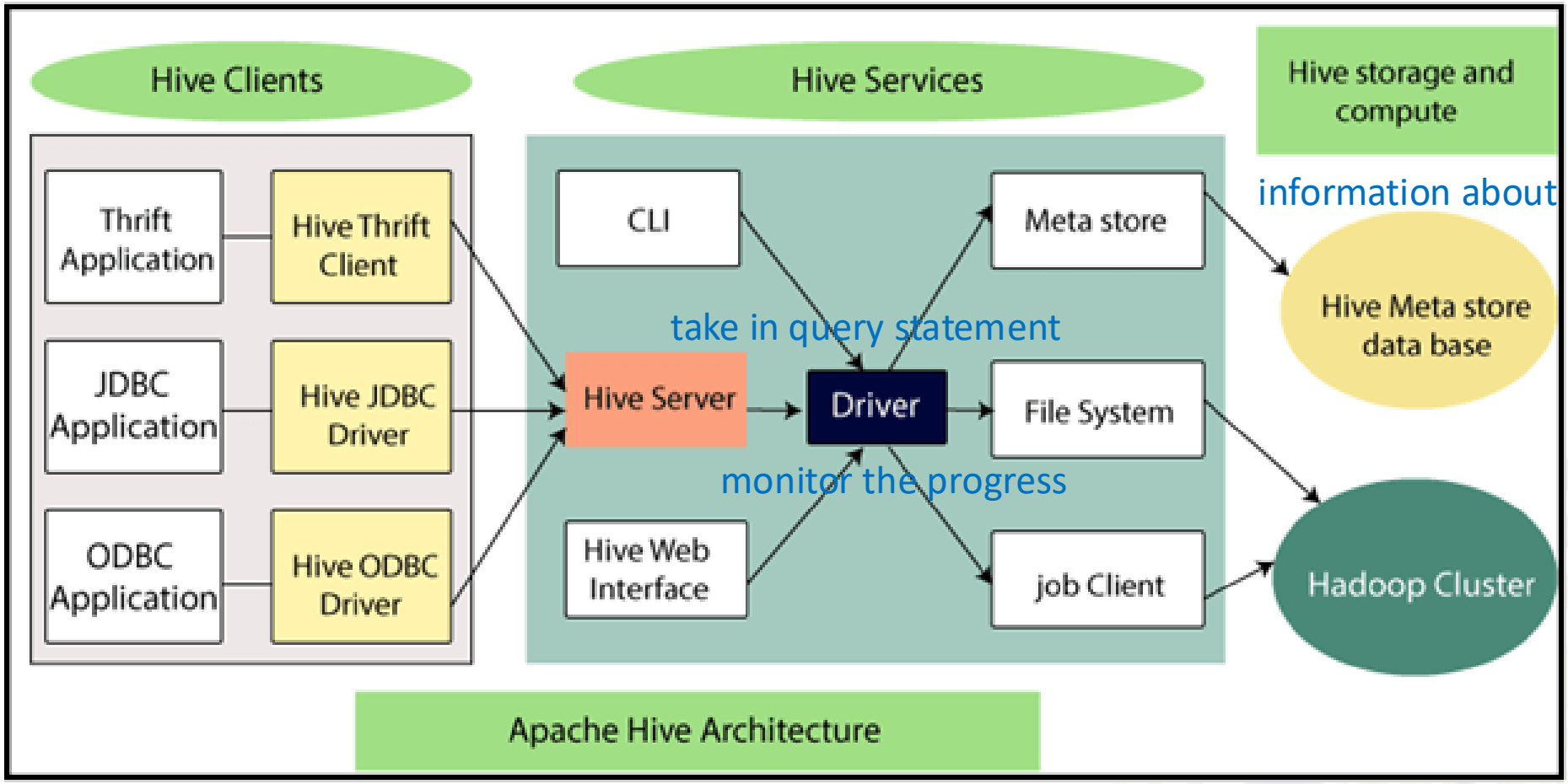
Mahidol University
Wisdom of the Land

Hive and traditional RDBMS

Traditional RDBMS	Hive
Used to maintain a Database and uses SQL	Used to maintain a data warehouse using Hive query language
Suited for real-time/dynamic data analysis like data from sensors	Suited for static data analysis like a text file containing names
Designed to read and write as many times as it needs	Designed on the methodology of write once, read many
Maximum data size it can handle is terabytes	Maximum data size it can handle is petabytes
Enforces that the schema must verify loading data before it can proceed	Doesn't enforce the schema to verify loading data
May not always have built-in for support data partitioning	Supports partitioning

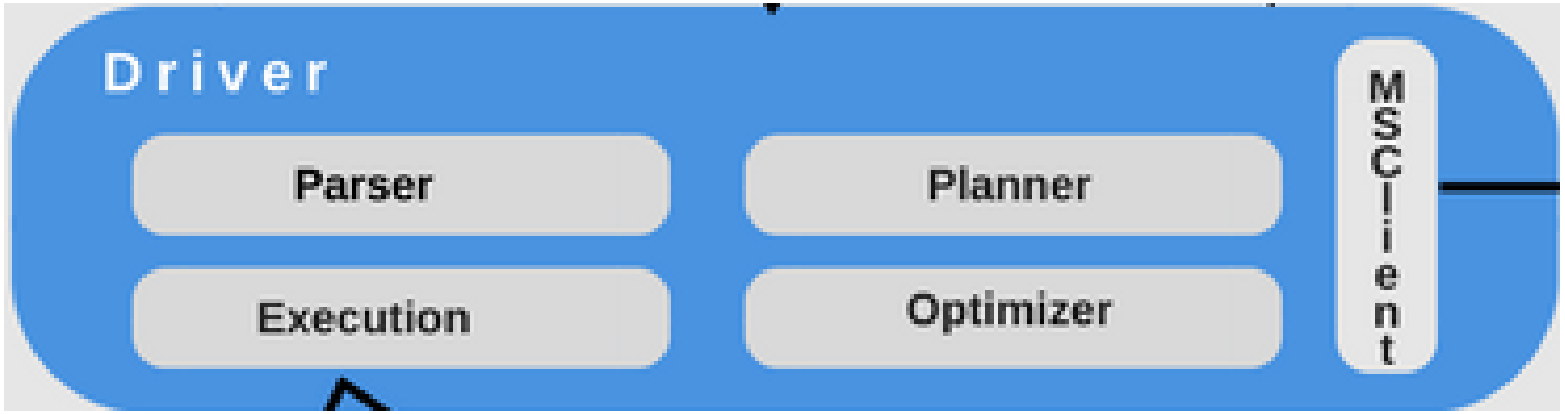
Mahidol University
Wisdom of the Land

Hive architecture



Mahidol University
Wisdom of the Land

Hive Driver



optimizer transformations on the execution plan and splits the tasks to help speed up and improve efficiency.

executor executes tasks after the optimizer has split the tasks.

Mahidol University
Wisdom of the Land

HBASE

- Columnmnar database
- Non-relational database sytem
- Runs on top of HDFS
- Provides a fault-tolerant way of sparse dataset
- Work well with real-time data and random read and write data access

Mahidol University
Wisdom of the Land

HBASE

- Write heavy application
- Backup support for MapReduce jobs
- Consistent read & Write
- No fixed schema
- Easy to use with JAVA API

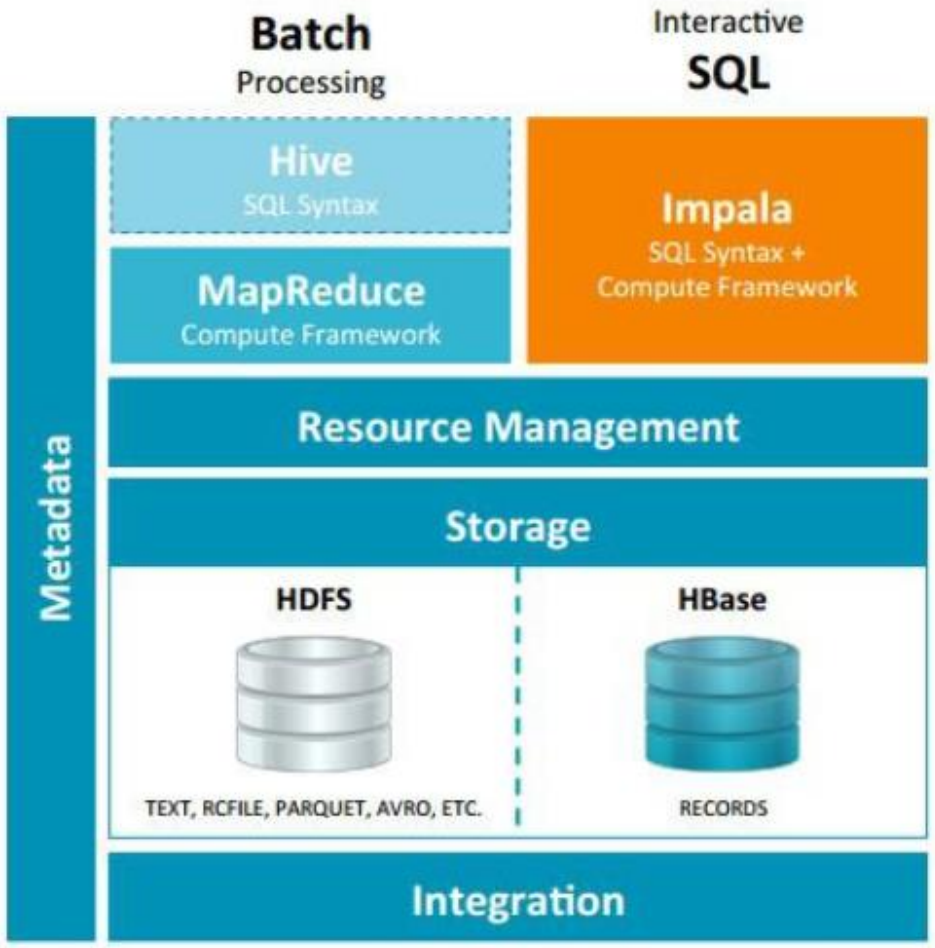
Mahidol University
Wisdom of the Land

HDFS and HBASE

HDFS	HBASE
Stores data in distributed manner accorss different nodes	Store data in form of columns and rows in a table
Does not allow changes	Allow dynamic changes
Write once read many	Random read and write
Storing only	Store and processing big data

Mahidol University
Wisdom of the Land

Simple architecture



Mahidol University
Wisdom of the Land

Programming models

- Support Big data operations
  - Split volume data
  - Fast data access
  - Distribute computations to nodes
- Handle Fault Tolerance
  - Replicate data partition
  - Recover files when needed

Mahidol University
Wisdom of the Land

Traditional Parallel programming

- Semaphorse
- Threads
- Monitors
- Message Passing
- Shared Memory
- Locks

Mahidol University
Wisdom of the Land

Programming model for Big Data

- Enable Adding more racks
  - Add more resources
  - For increase data storage faster data access
  - Scaling out

- Optimized for specific data types
  - Table
  - Stream
  - Image
  - Document
  - Key-Model

Functional Programming Example

- **Function creator**

f(x) =x+1

apply(f(x),

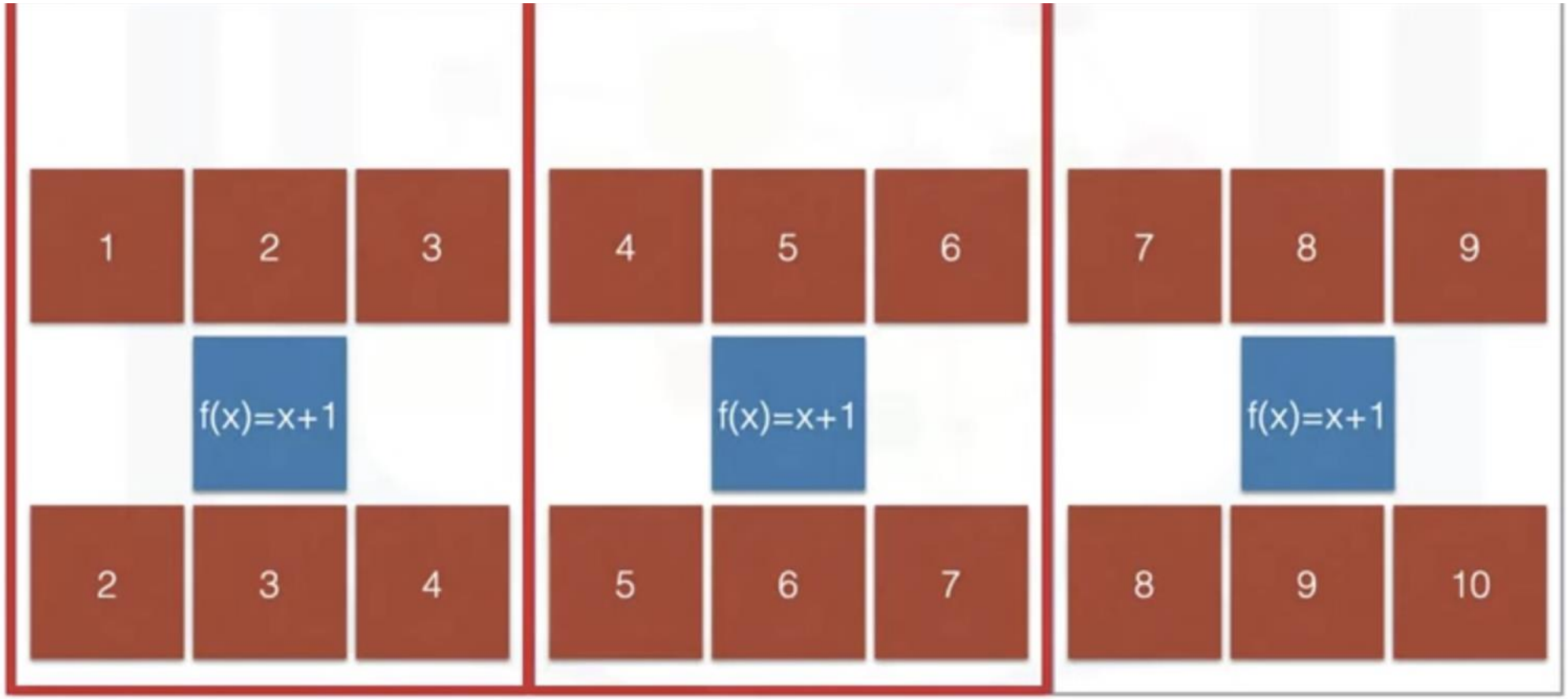
2
3
4
5
6

)=

3
4
5
6
7

Mahidol University
Wisdom of the Land

Parallerlization



Mahidol University
Wisdom of the Land

MapReduce

- Programming model used in Hadoop for Big Data Processing
- Processing technique for distributed computing
- Consist of Map and Reduce task
- Can be coded in Java, C++, Python, Ruby and R

Mahidol University
Wisdom of the Land

Map Reduce

- Map
  - Apply operation to all elements
  - f(x)=y
  - Provide format of the operation for eachh data element
- Reduce
  - Summarize operation on elements

Mahidol University
Wisdom of the Land

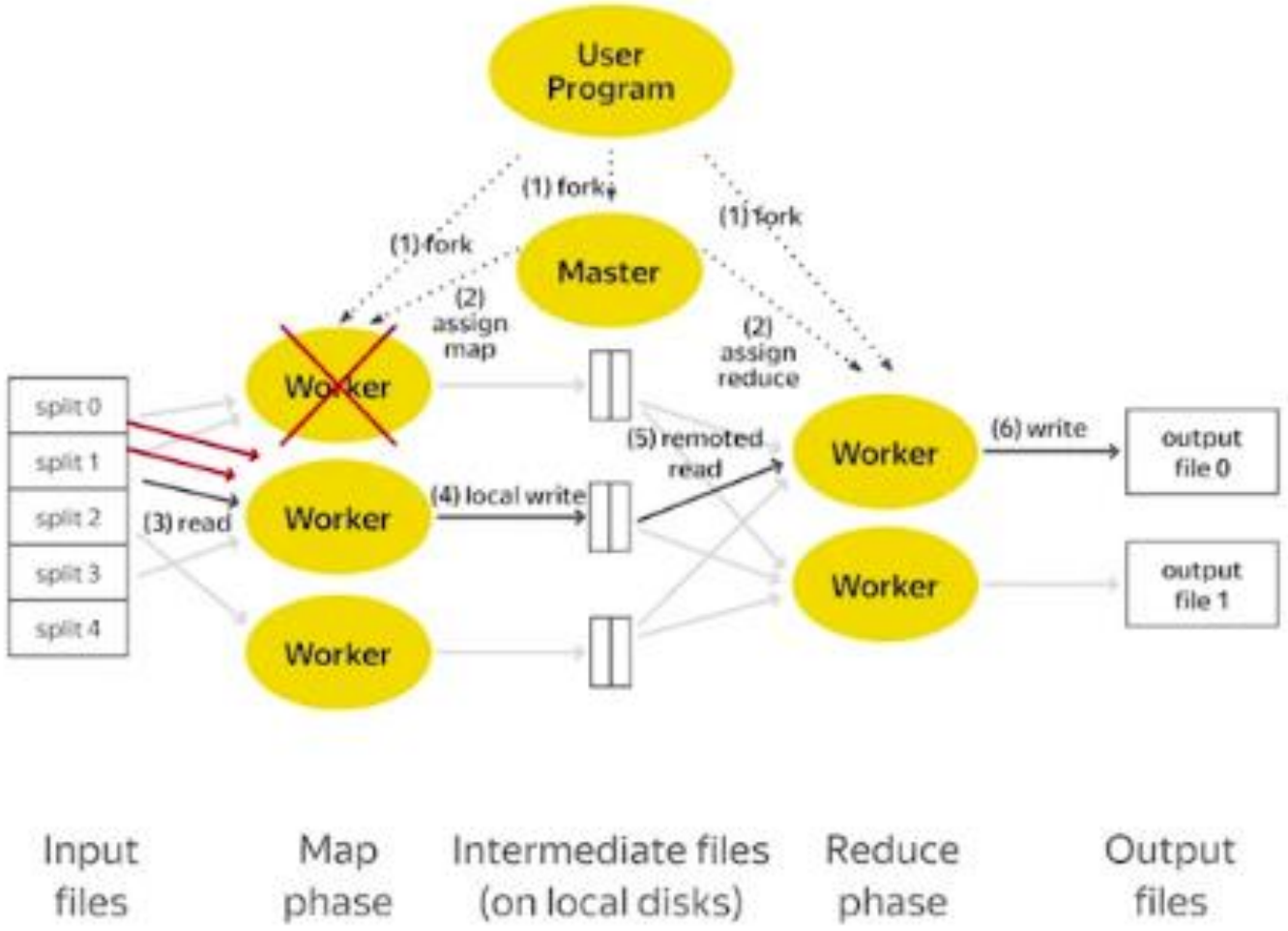
Map and Reduce steps

File	• input File
Map	• Process data map to • Key-Value pair
Rearrange	• sort • reorganize
Reducer	• Aggregate and compute result • Produce Final Output

To keep track of its tasks → create a unique key

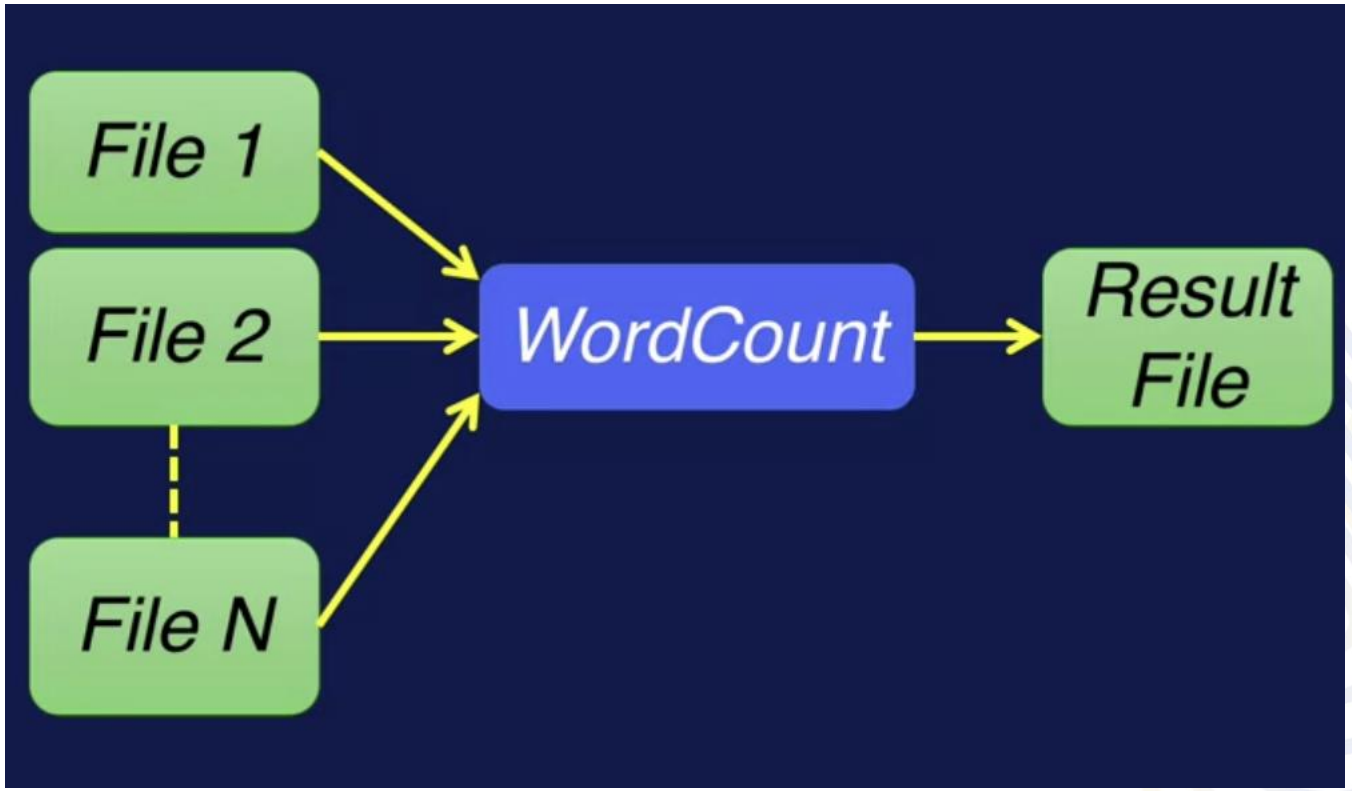
Mahidol University
Wisdom of the Land

Map Reduce



Mahidol University
Wisdom of the Land

WordCount Example



Mahidol University
Wisdom of the Land

Mapper

```
public static class TokenizerMapper
     extends Mapper<Object, Text, Text, IntWritable>{

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  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);
    }
  }
```

Key process

- Extends **mapper – defined** in Mapreduce Framework

Input

- *key* Unique. words
- *Value* whole line of text

Output

- map function: **key, value**
- IntWritable is always1

(my,1) (bigger,1)
(car,1) (than,1)
(is,1) (your,1)
(car,1)

Mahidol University
Wisdom of the Land

Reducer

```
public static class IntSumReducer
     extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();

  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);
  }
}
```

Key process

- Extends **mapper – defined** in Mapreduce Framework

Input

- *key* Unique words
- *Value* =1

(my,1) (bigger,1)
(car,1) (than,1)
(is,1) (your,1)
(car,1)

Output

- *key* Unique words
- *Value* sum of all words

(my,1) (bigger,1)
(car,2) (than,1)
(is,1) (your,1)

Mahidol University
Wisdom of the Land

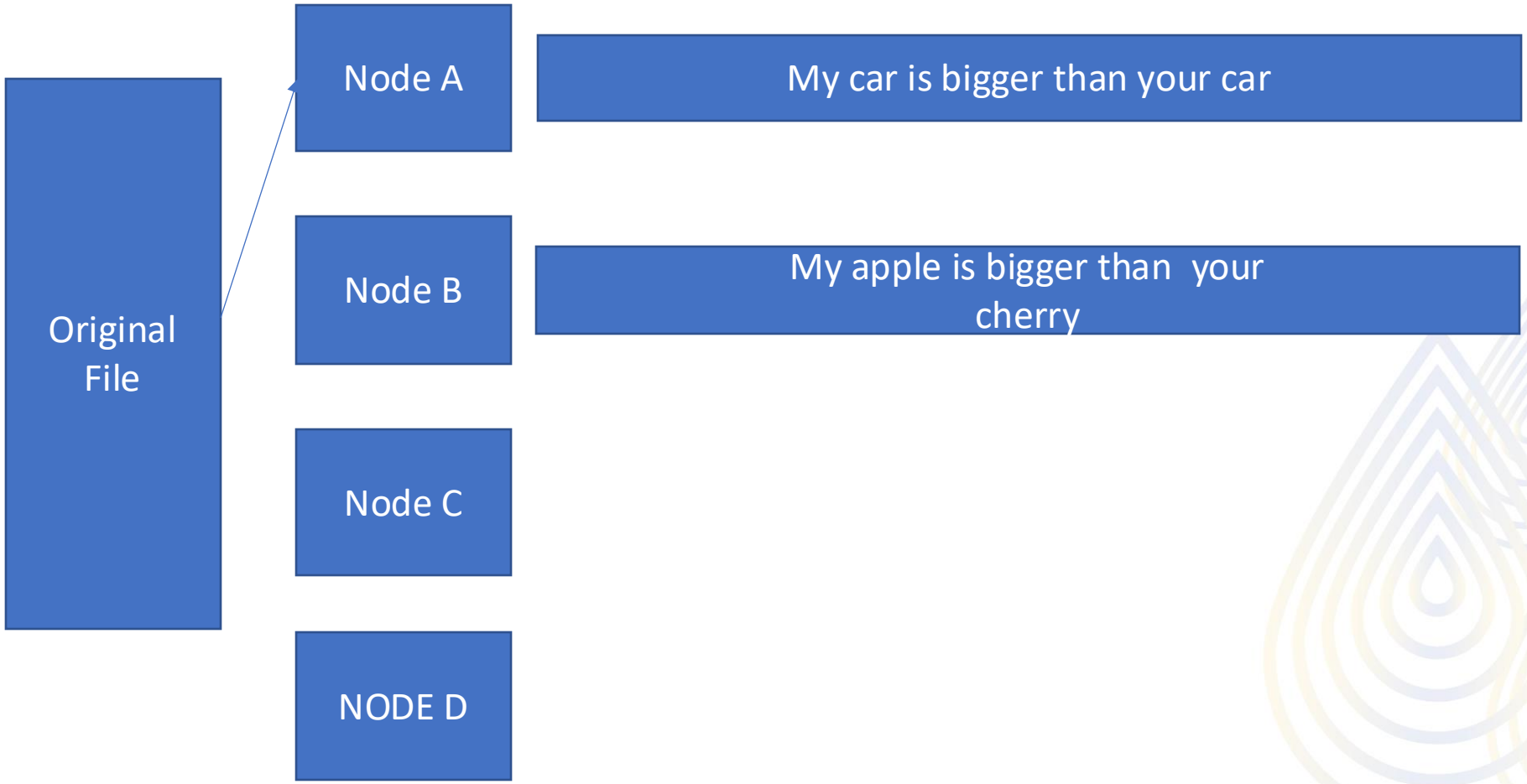
Driver Code

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

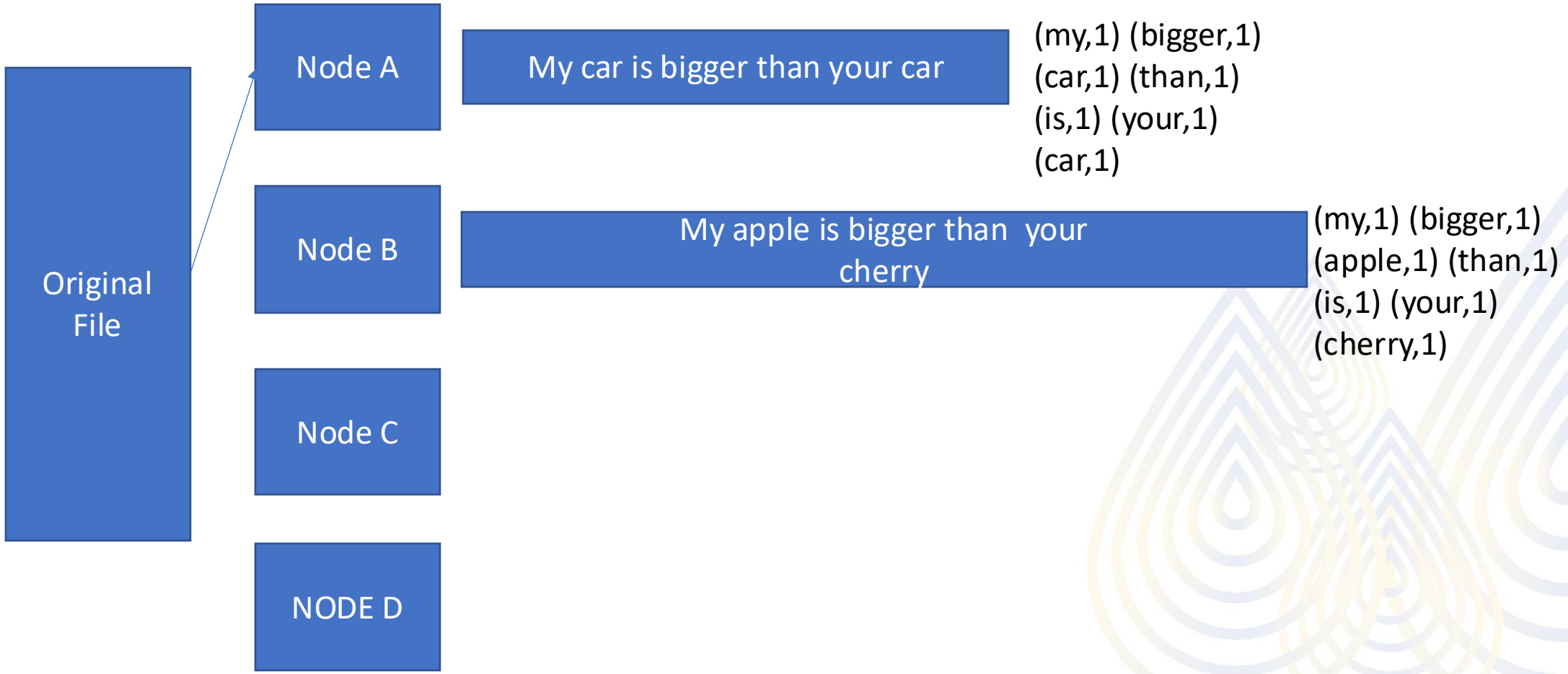
Mahidol University
Wisdom of the Land

How does it work?



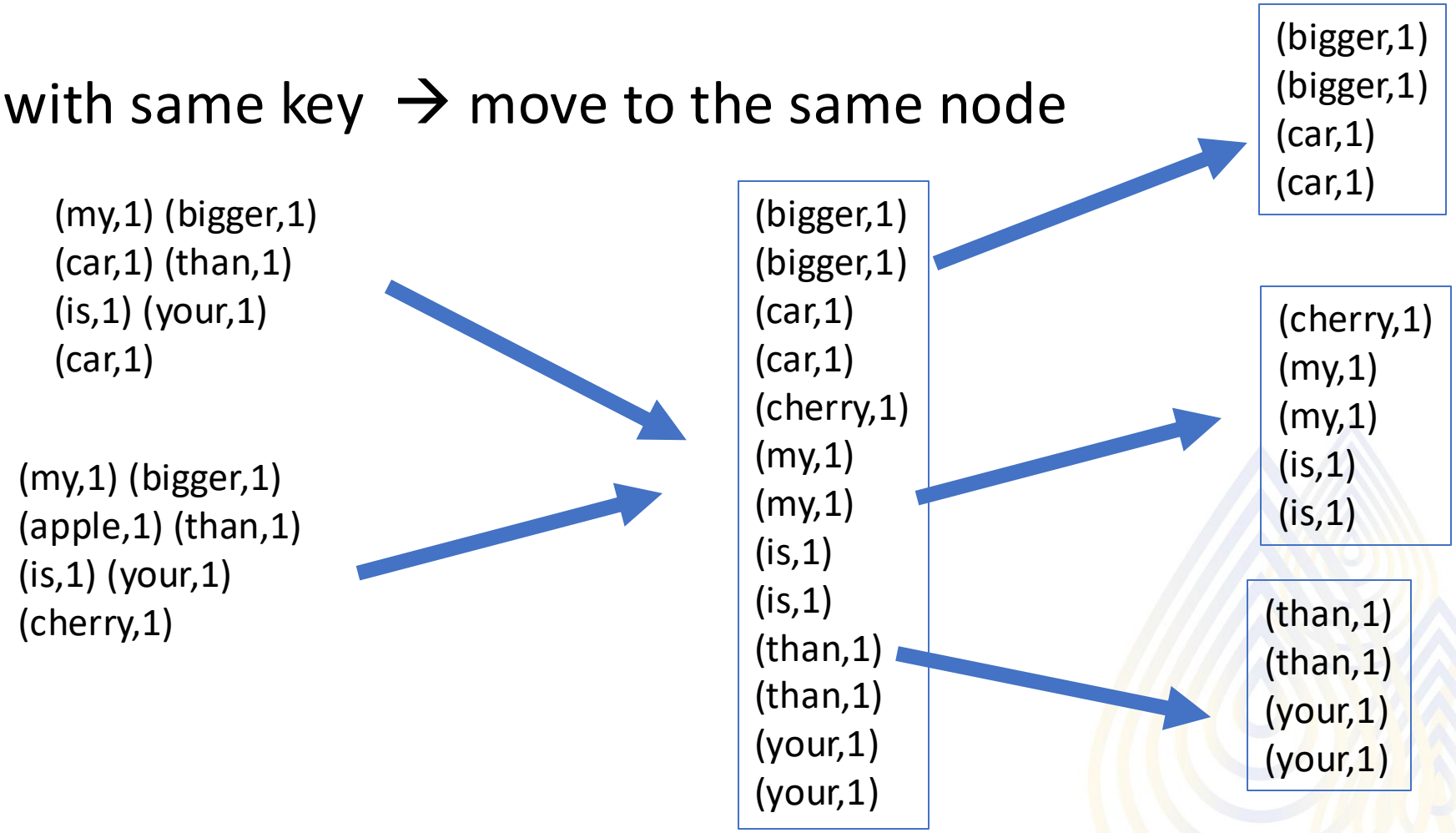
Mahidol University
Wisdom of the Land

Map step



sort and shuffle

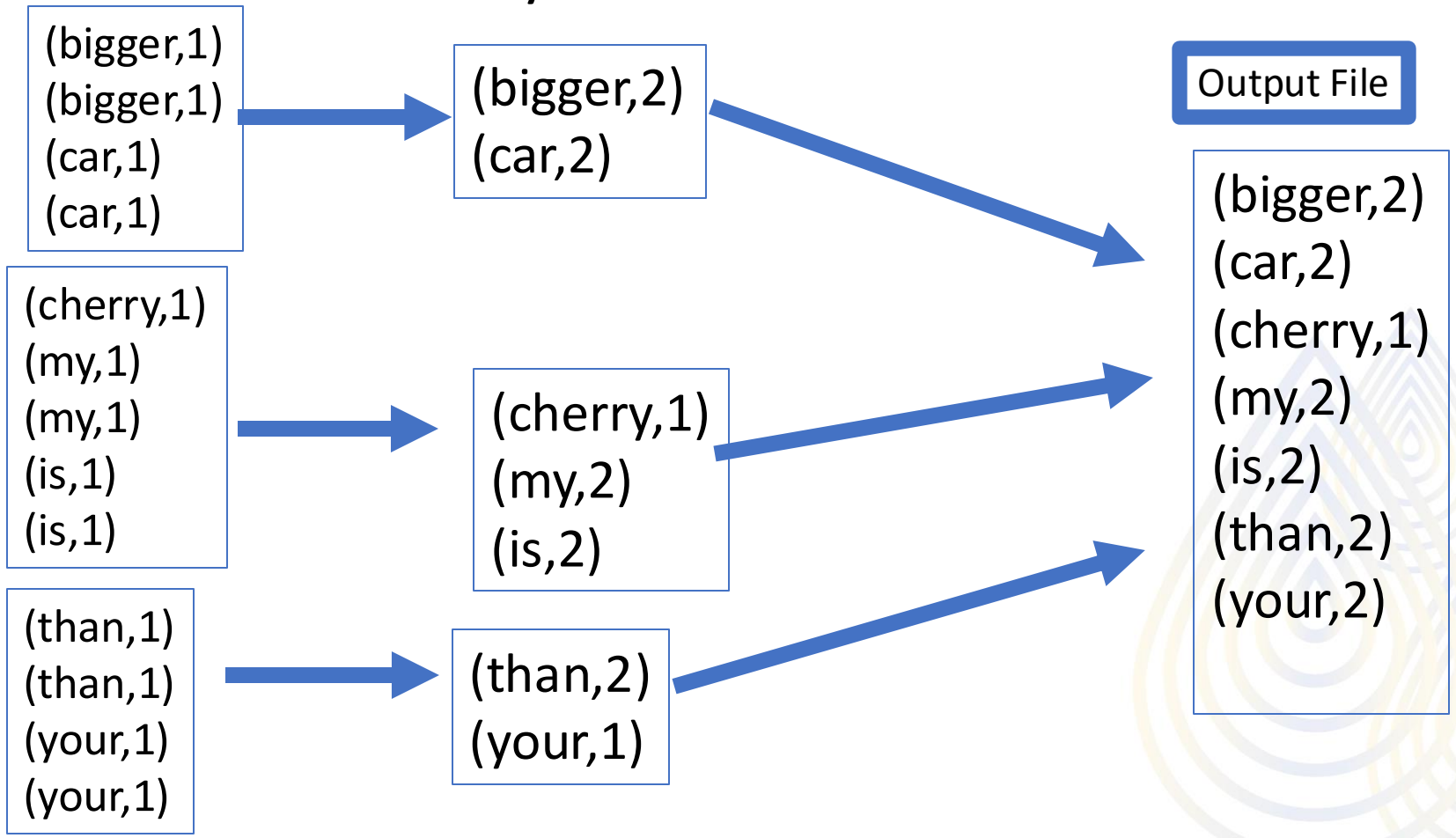
- Pairs with same key → move to the same node



Mahidol University
Wisdom of the Land

Reduce

- Add value to the same key



Mahidol University
Wisdom of the Land

Experiment 2: Manual Job creation

Setup the environment

- `hadoop fs -mkdir /user/<user_name>`
- `export PATH=${JAVA_HOME}/bin:${PATH}`
- `export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar`

Mahidol University
Wisdom of the Land

Manual Job creation(2)

Upload a file

Wordcount.java into your ssh (not hdfs)

Compile a file

- Hadoop com.sun.tools.javac.Main./WordCount.java
- jar cfwordcount.jar WordCount*.class

Run

- hadoop jar WordCount.jar WordCount covid.txt out

Explore the new directory

Download (for further use)

Put to your cloud storage (to make it safe)

Explore output: Run the map-reduce

- *hadoop fs -ls out/*
- *hadoop fs -cat out/partxx*
- *hadoop fs -copyToLocal out/partxxxx*

Or to your bucket (hadoop fs –cp file gs://)

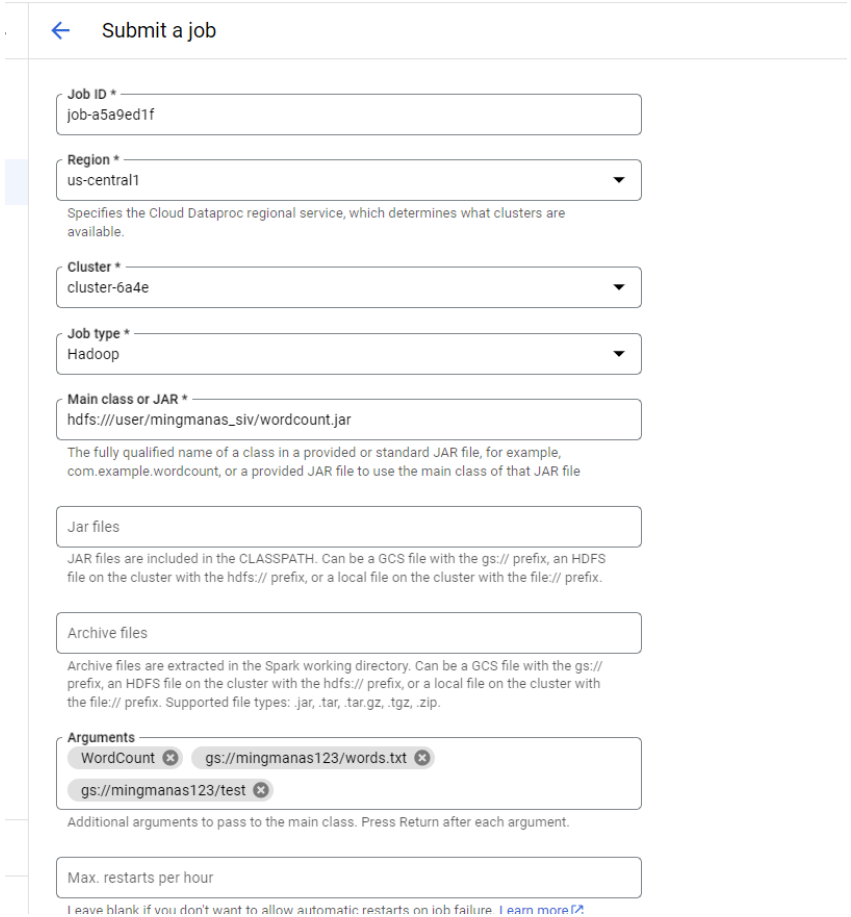
- *→Check the output*

Mahidol University
Wisdom of the Land

Why MapReduce

- Parallel Computing
- Divide-> Run Tasks -> Combine -> DONE
- Process data in structured/un-structured file formats
- Support different programming languages
- Support platform for analysis

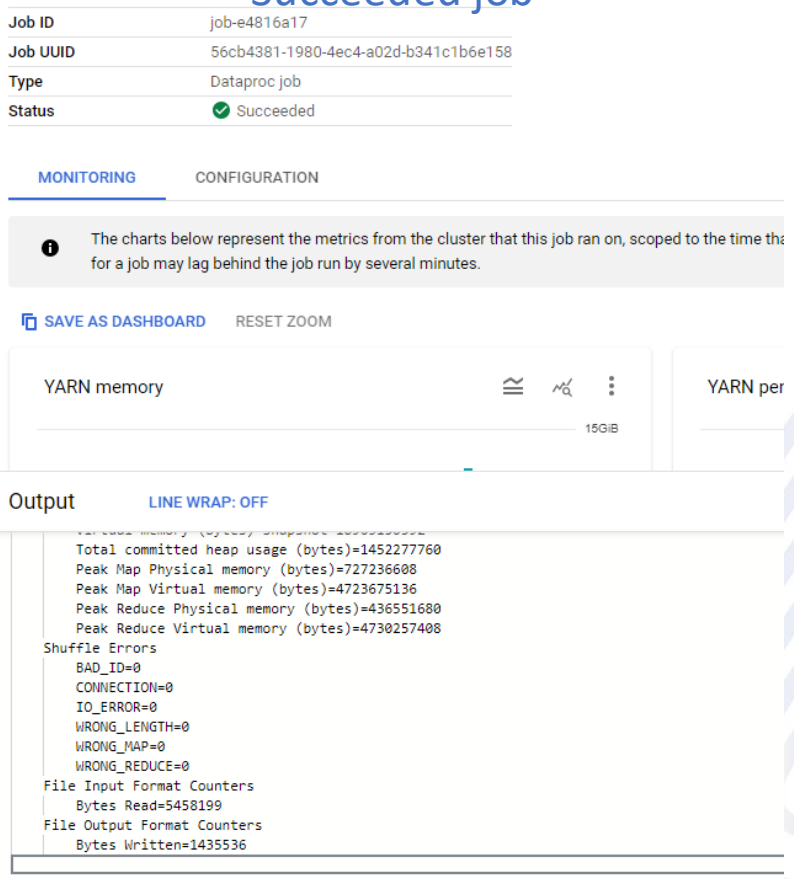
Submit job



Jar files

Related Arguments

Succeeded job



Mahidol University
Wisdom of the Land

Not familiar with JAVA?

- Python streamers
  - hadoop jar ///usr/lib/hadoop/hadoop-streaming.jar \
    -files mapper.py,reducer.py \
    -mapper 'python mapper.py' \
    -numReduceTasks 1 \
    -reducer 'python reducer.py' \
    -input covid.txt \
    -output wc1/

```
Combine output records=0
Reduce input groups=2
Reduce shuffle bytes=30
Reduce input records=2
Reduce output records=1
Spilled Records=4
```

Mahidol University
Wisdom of the Land

mapper and reducer files

```
import sys
line_count=0
for line in sys.stdin:
          line_count+=1
print '%d' % line_count
```

mapper.py

```
import sys
line_count=0
for line in sys.stdin:

     line_count+=int(line)
print '%d' % line_count
```

reducer.py

Mahidol University
Wisdom of the Land

Standard Program

- **Use this file**

  ///usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples-3.3.6.jar

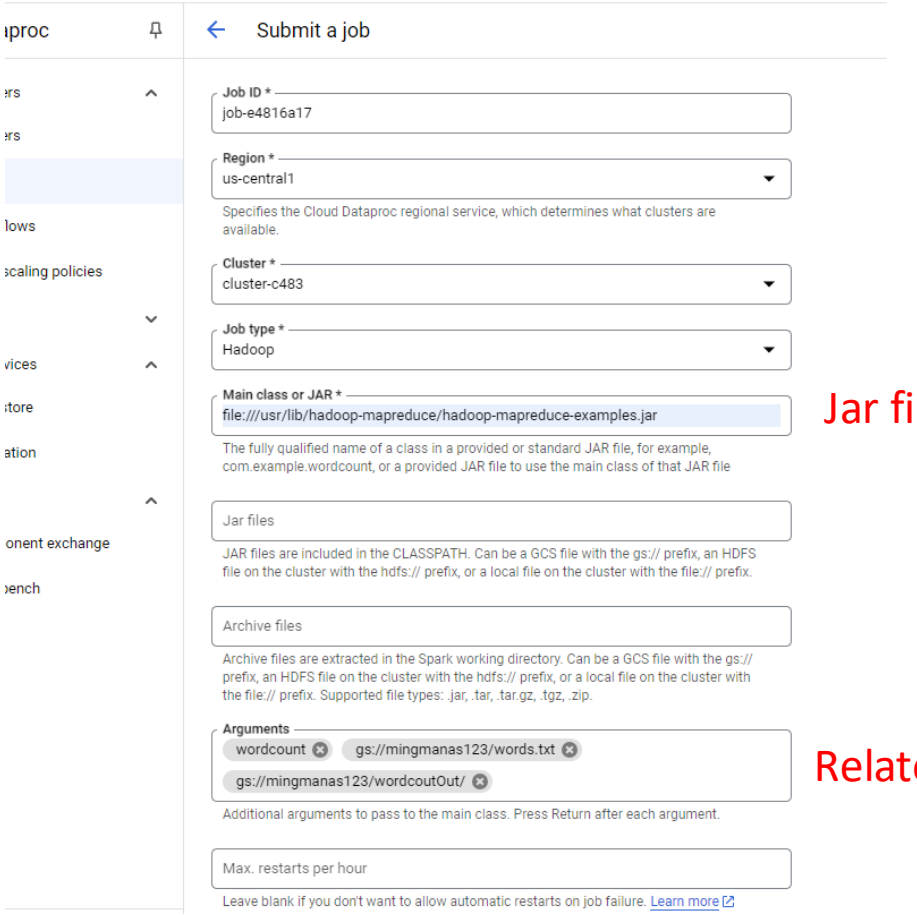
- **Explore its usage**
  - *hadoop jar [your file]*
- **Run the example**
  - *hadoop jar* [your file] ***wordcount [your gs://file] [your folder]***

- **Explore output**

- **Check Status in web interface**

Mahidol University
Wisdom of the Land

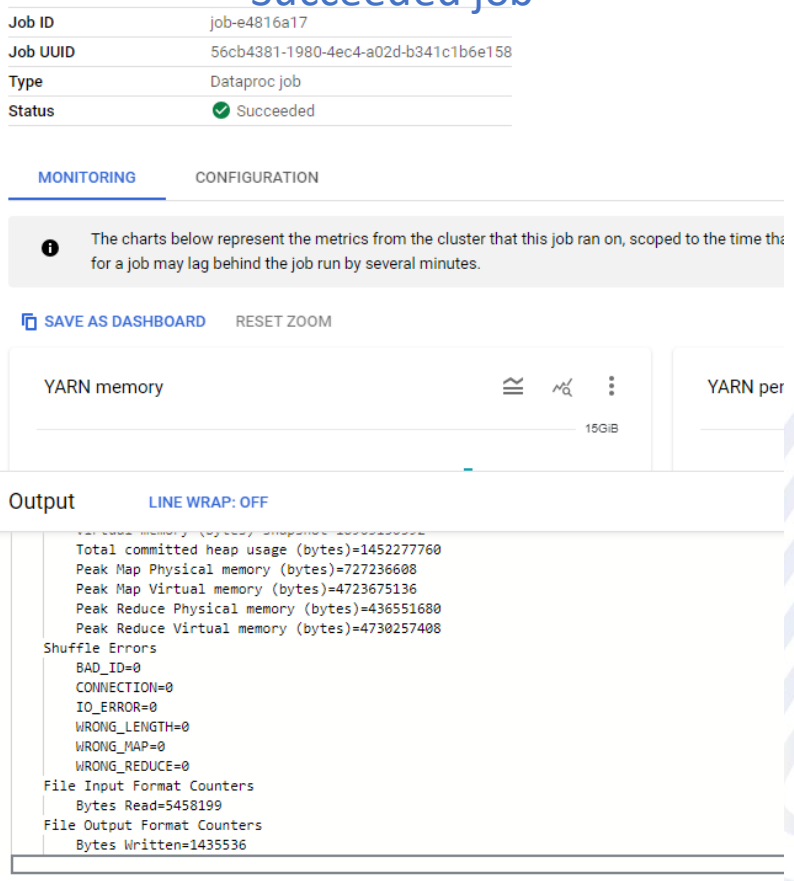
Submit job



Jar files

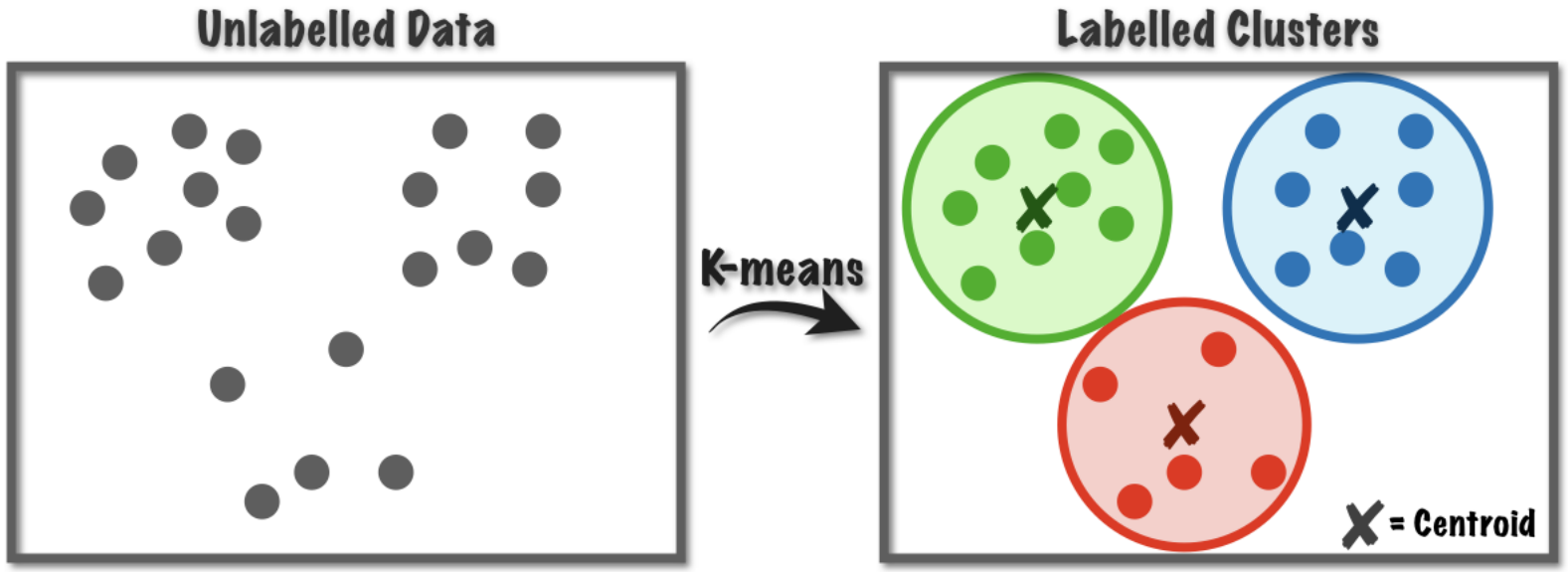
Related Arguments

Succeeded job



Mahidol University
Wisdom of the Land

MORE EXAMPLE:K MEAN



Mahidol University
Wisdom of the Land

K-mean Examples

Classify: Assign observations to closest cluster center

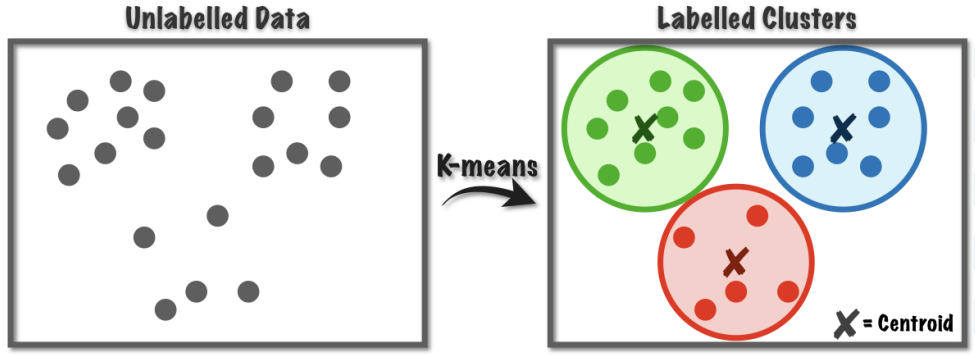
$$z_i \leftarrow \arg\min_j ||\mu_j - \mathbf{x}_i||_2^2$$

Map: For each data point, given ($\{\mu_j\}$,$\mathbf{x}_i$), emit(z_i,$\mathbf{x}_i$)

Recenter: Revise cluster centers as mean of assigned observations

$$\mu_j = \frac{1}{n_j} \sum_{i:z_i=k} \mathbf{x}_i$$

Reduce: Average over all points in cluster j (z_i=k)



Mahidol University
Wisdom of the Land

Examples

- Distributed grep
- Count of URL Access Frequency
- Reverse Web-link graph:
  - (target,source) →(target,list(source))
- Term-vector per Host
  - <word,frequencty> → <hostname, term vector>
- Inverted index
- Distributed Sort

Mahidol University
Wisdom of the Land

More Examples

- Joining data
- Vector multiplication
  - <index, value>
- Feature selection
- Clustering
- word frequency

Mahidol University
Wisdom of the Land

Additional Reading

Google: Summary

Google White Paper: https://research.google.com/archive/mapreduce-osdi04.pdf