Lecture 3 – MapReduce Details

BDAT 1002

Review

Q1

- Which of the below is not a file system?
  - HDFS
  - ext4
  - FS4
  - ext3

Q2

- Assume the cluster (block) size of NTFS is 32 KB. If you are trying to store a file of size 5 KB in your file system, how much space will that file occupy in your hard disk?
  - 5 KB
  - 32 KB
  - 64 KB
  - 64 MB

Q3

- Assume the block size of HDFS is 64 MB and the block size of underlying file system is 1 MB. If you are trying to store a file of size 5 MB in your file system, how much space will that file occupy in your hard disk?
  - 64 MB
  - 5 MB
  - 128 MB
  - 300 MB

Q4

- The concept of HDFS is based on which of the below file system?
  - NTFS
  - ext4
  - ext3
  - GFS

Q5

- What is the default replication factor in HDFS?
  - 5
  - 4
  - 3
  - 2

Q6

- What is the maximum RAM that a 32 bit machine can support?

MapReduce Introduction

MapReduce Introduction

- Example problem: Governor of California wants you to do a census to find the population of California by city
  - You have all the resources you want
- What is your process?
  - Can you do it yourself?
- Sensible thing to do is "divide and conquer"

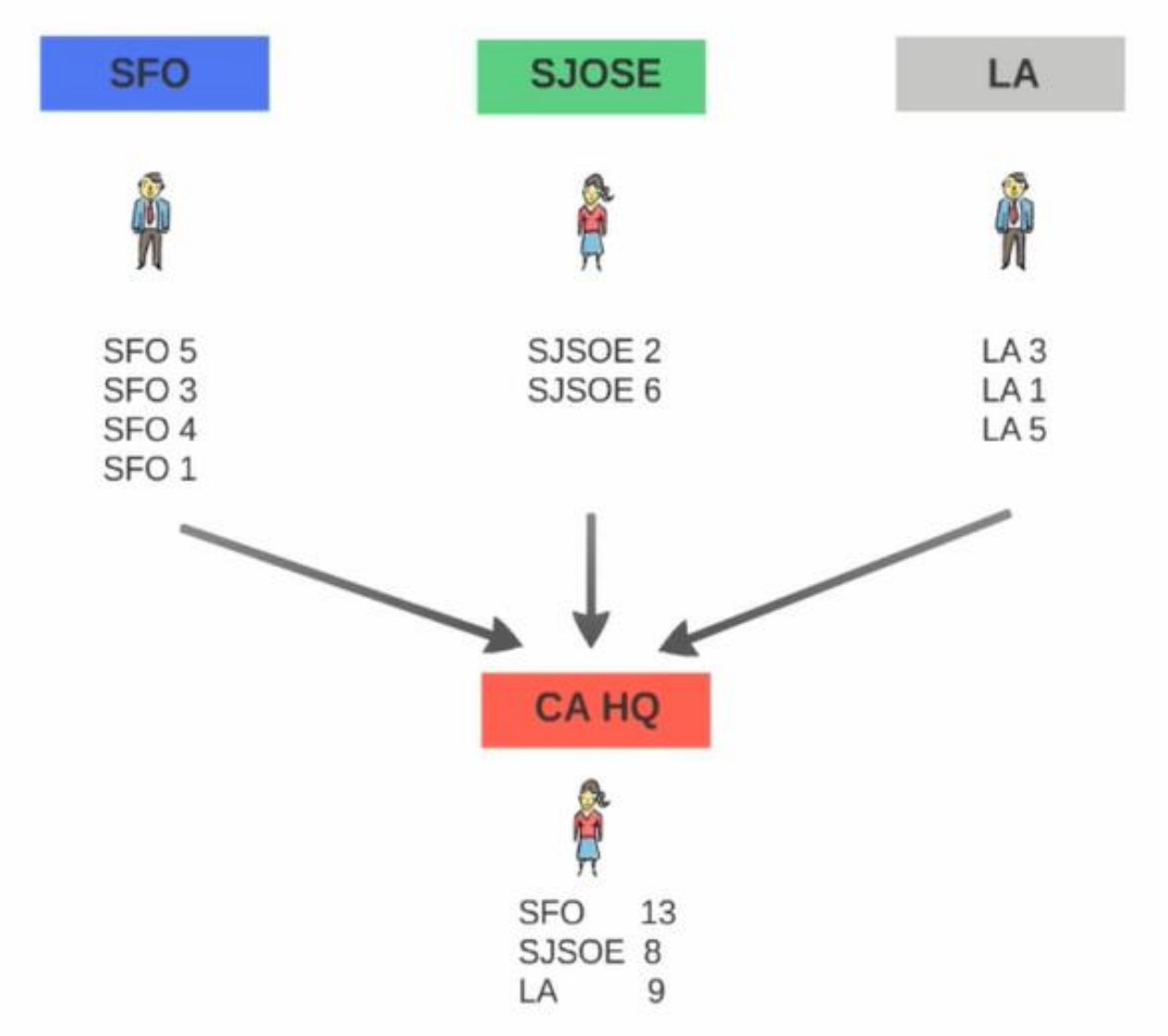
MapReduce Introduction

- For simplicity, let's say there are only three cities
- Have a person in charge of each of the cities
- Each person responsible for finding the population of each city

MapReduce Introduction

- Each person goes home by home and write down the number of people in the household

MapReduce Introduction



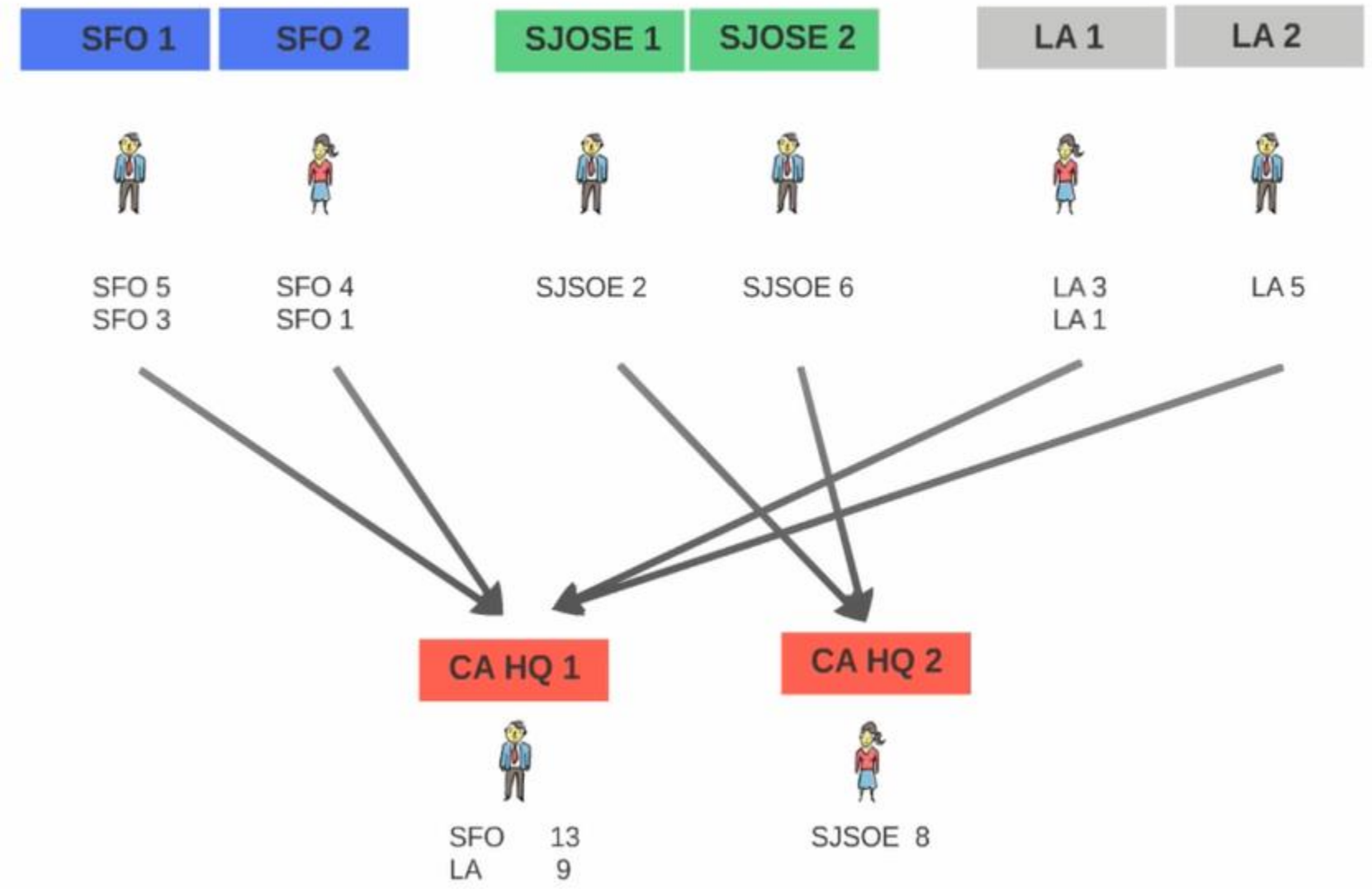
MapReduce Introduction

- Next year, you are now asked to do the same thing BUT with **half the time**
- What to do?
- Remember you have all the resources you want?

MapReduce Introduction

- Double the number of people doing count for each city

MapReduce Introduction



MapReduce Introduction

- There is one crucial detail here
- You want your census takers for each city (ie SFO 1 and SFO2) send the results to **either HQ1 or HQ2**
- We don't want results spread across two headquarters
  - Remember we wanted population by city
- So there must be instructions for census takers in one city to coordinate and send data to same headquarters

MapReduce Introduction

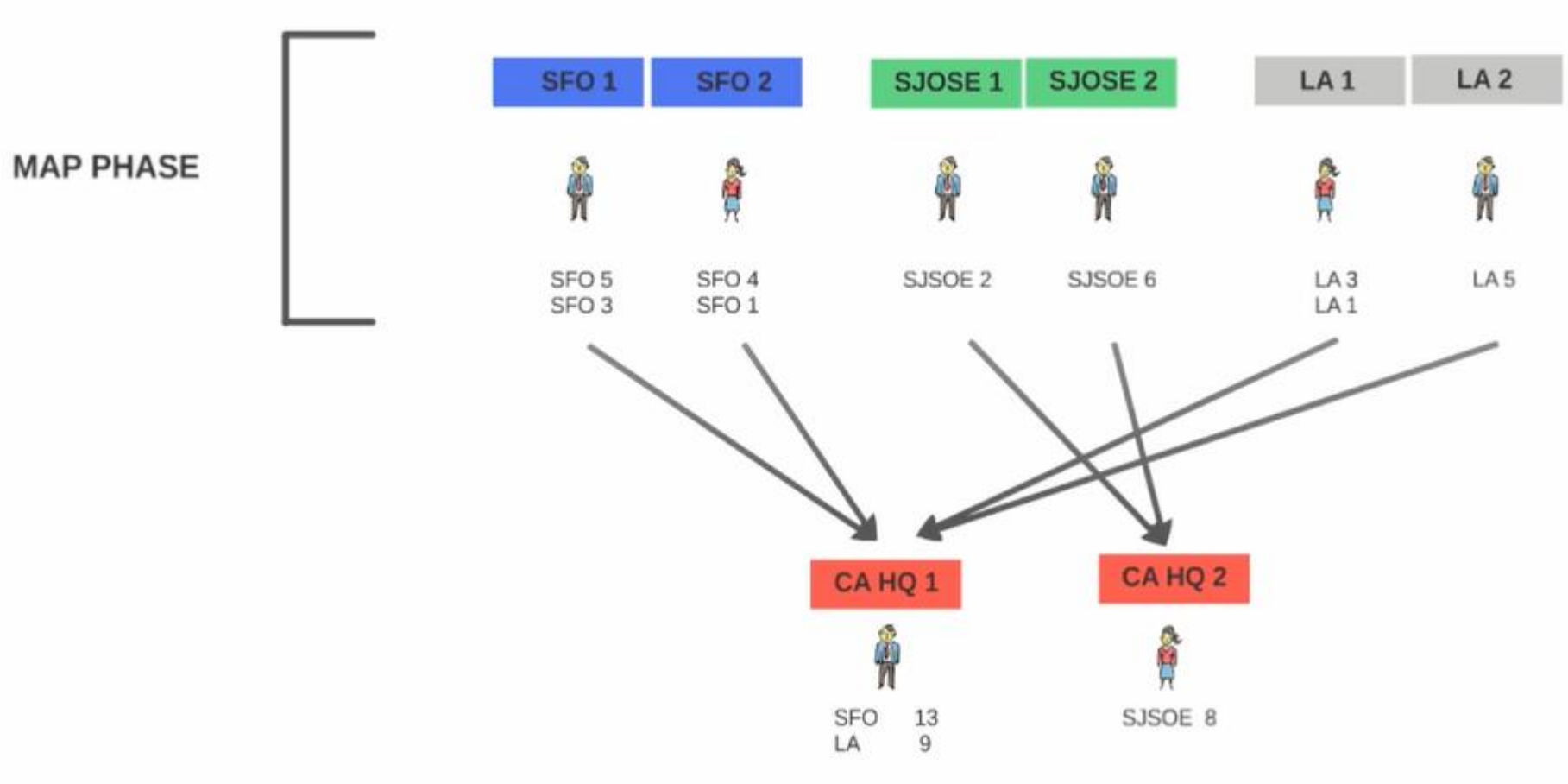
- We can expand this model, more resources, less time
- This model works and **it can scale**
- The model is called MapReduce

What is MapReduce?

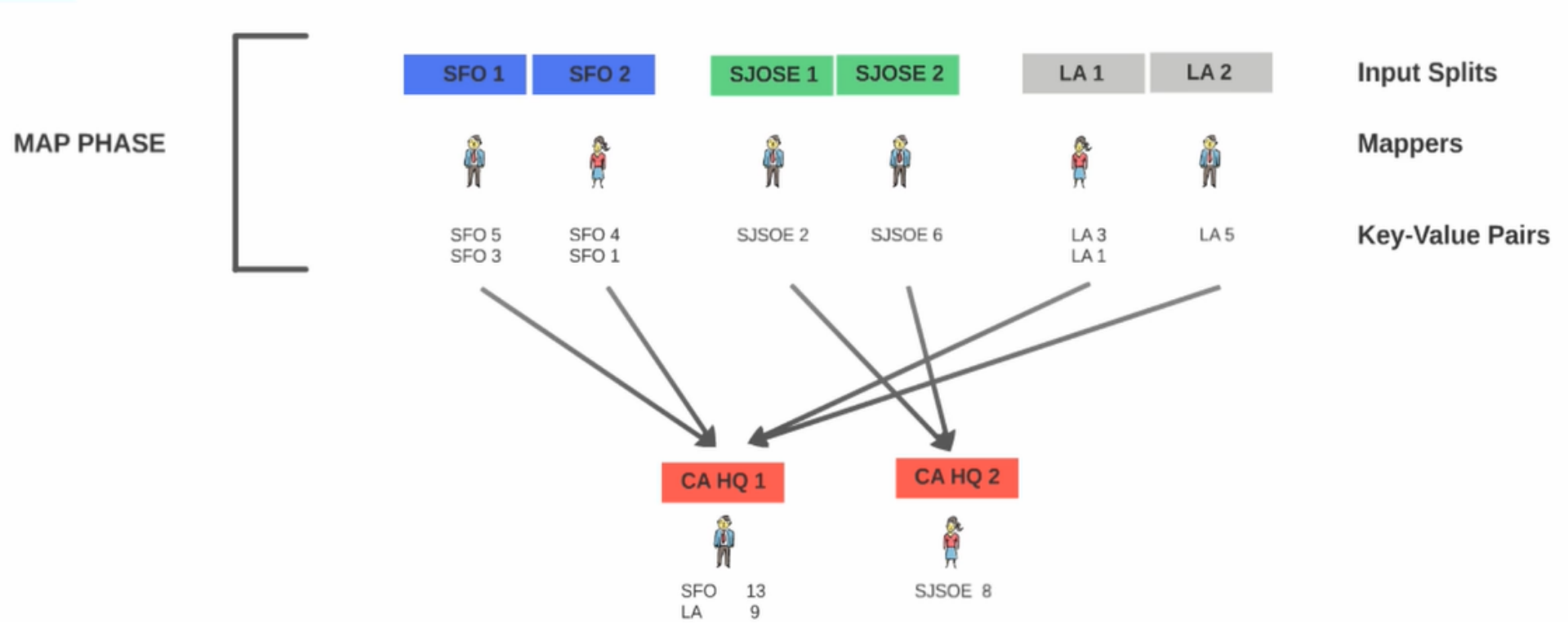
- MapReduce is a programming model for ***distributive computing***
  - It is not a programming language

Phases in MapReduce

Phases of MapReduce



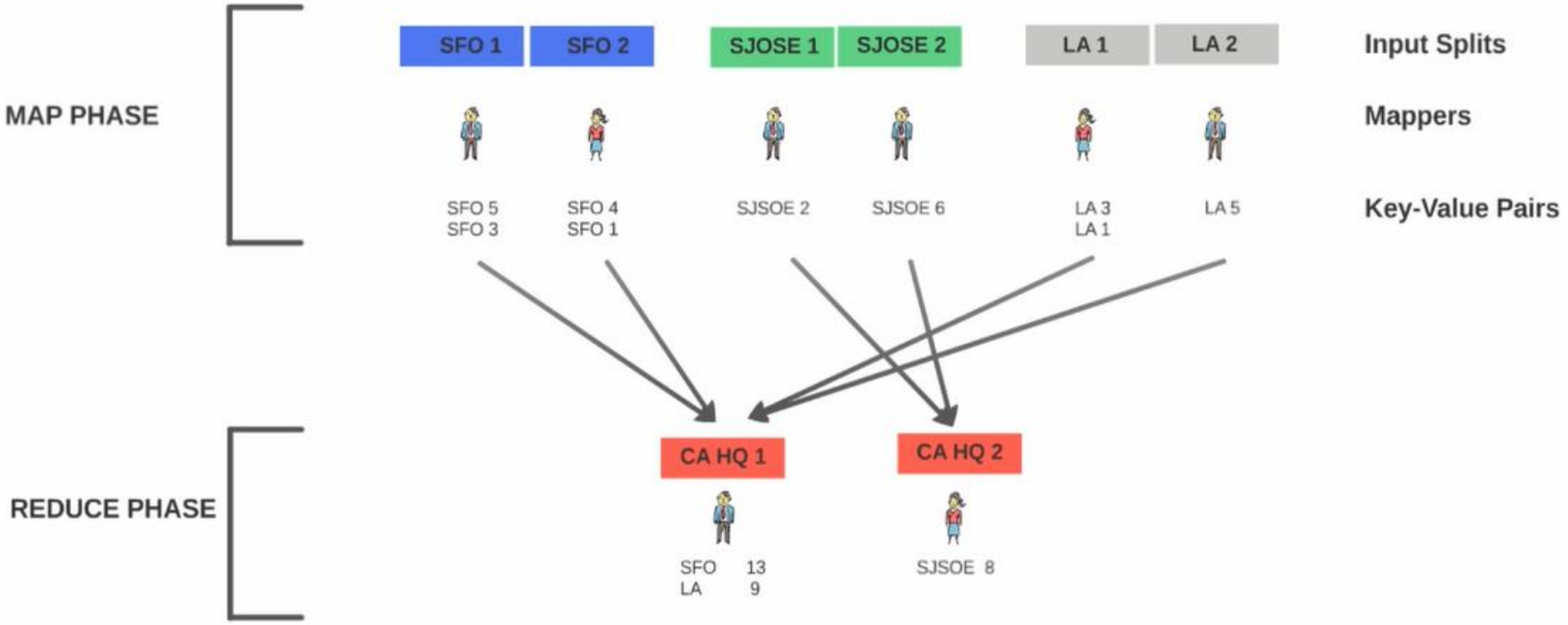
Phases of MapReduce



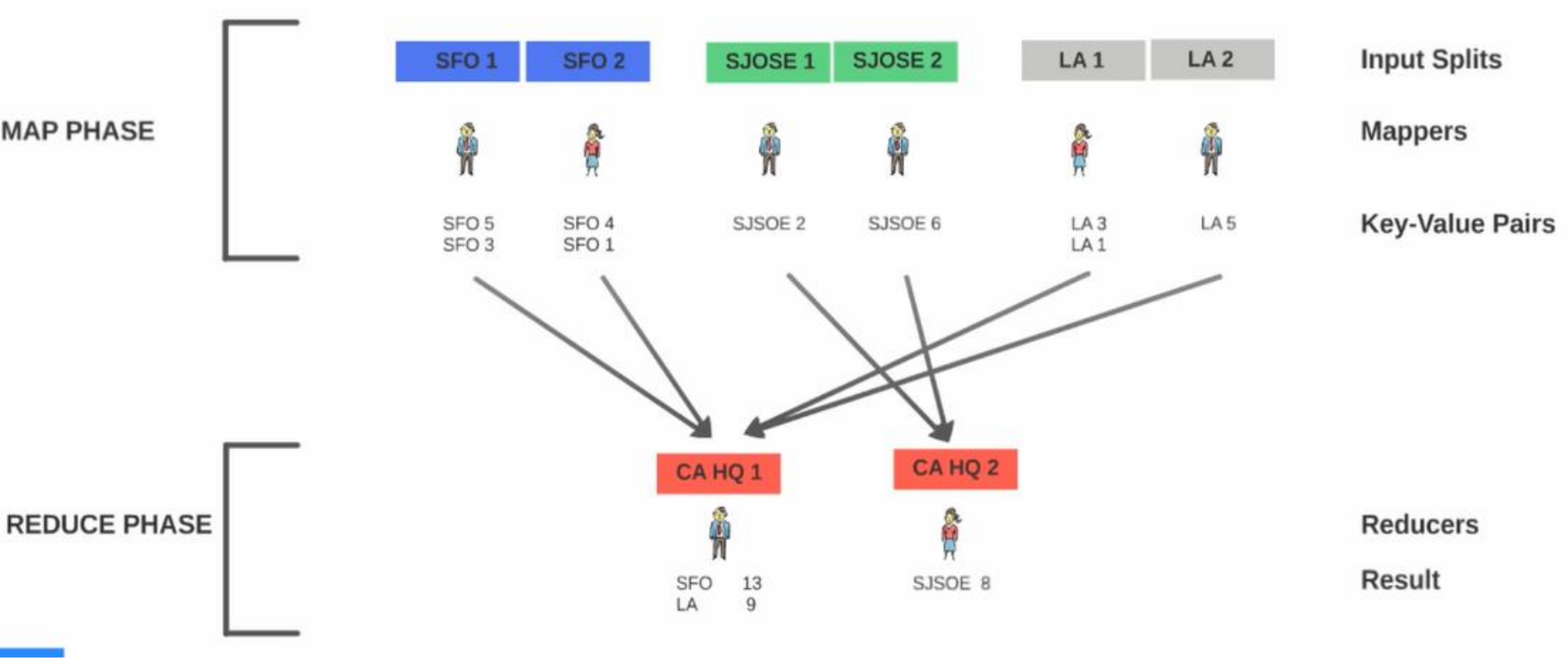
Phases of MapReduce

- The phase where individuals collect the population of individual cities is called the **Map Phase**
- Individual person involved in the calculations is called **the Mapper**
- The city or part of the city they are working in is called **Input splits**
- The output from each pair is **key-value pair**

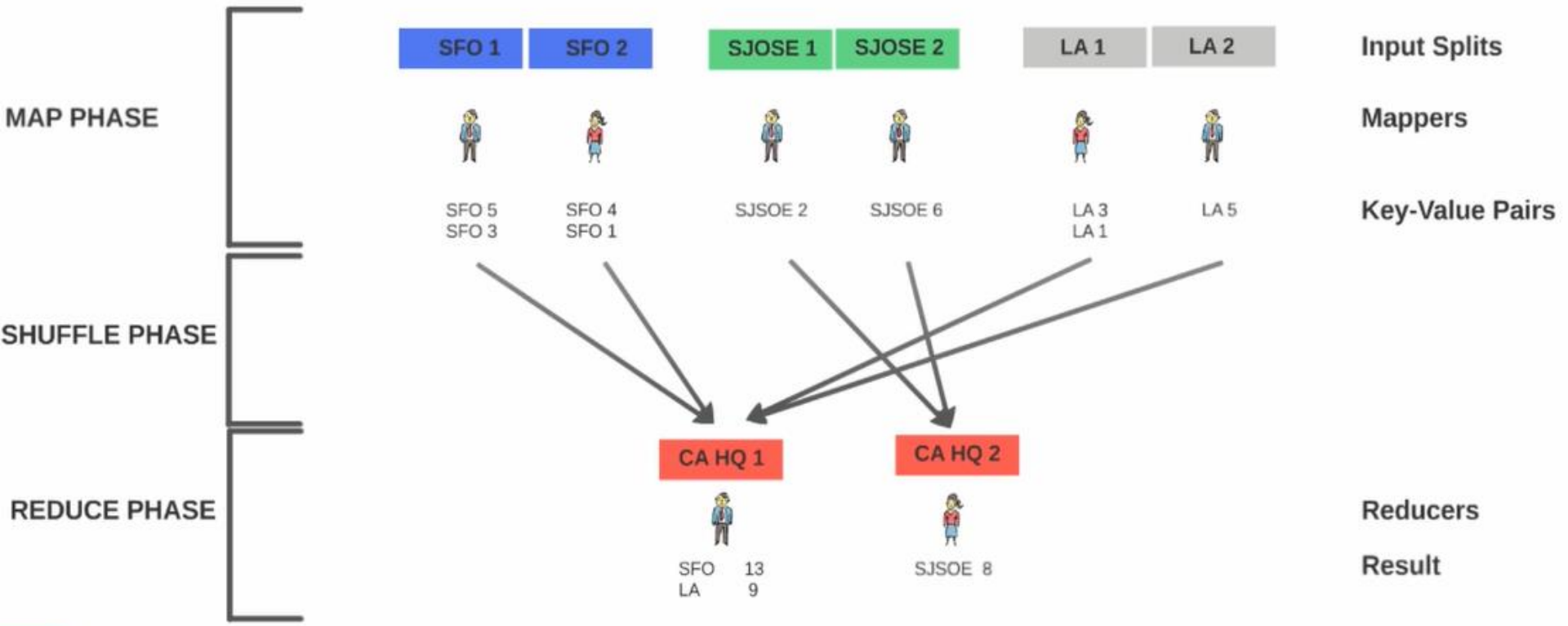
Phases of MapReduce



Phases of MapReduce



Phases of MapReduce



Phases of MapReduce

- The phase you aggregate the results for each city is called the **reduce phase**
- The individuals working in the headquarters are known as the **reducers**
- Each reducer will produce a **result set**
- The phase where the values for the different mappers is copied or transferred to the reducers is known as the **shuffle phase**

Summary

- MapReduce is a programming model for processing large data
  - Distributed processing
  - Developed at Google
  - Adapted by Hadoop
- Can be implemented in any programming language
  - Hadoop supports many languages
  - Java, C++, Python, Scala

MapReduce in Detail

Sample Big Data Problem

- Sample stock dataset
- Each record has symbol, date, open, close, max, min, volume, etc
- Want to find the maximum closing price for each symbol

Sample Big Data Problem

- Each line in the data set is a **record**

```
ABCSE,B7J,2008-10-28,6.48,6.74,6.22,6.72,44300,5.79
ABCSE,B7J,2008-10-27,6.21,6.78,6.21,6.40,55200,5.51
ABCSE,B7J,2008-10-24,6.39,6.66,6.21,6.40,67400,5.51
ABCSE,B7J,2008-10-23,6.95,6.95,6.50,6.59,59400,5.68
ABCSE,B7J,2008-10-22,6.92,7.17,6.80,6.80,55300,5.86
ABCSE,B7J,2008-10-21,7.20,7.30,7.10,7.10,54400,6.11
ABCSE,B7J,2008-10-20,6.94,7.31,6.94,7.12,45700,6.13
ABCSE,B7J,2008-10-17,6.43,6.93,6.42,6.90,57700,5.94
ABCSE,B7J,2008-10-16,6.61,6.69,6.21,6.53,83200,5.62
ABCSE,B7J,2008-10-15,6.84,6.90,6.36,6.36,78900,5.48
ABCSE,B7J,2008-10-14,7.15,7.32,6.93,6.96,74700,5.99
ABCSE,B7J,2008-10-13,6.00,6.57,6.00,6.57,75700,5.66
ABCSE,B7J,2008-10-10,5.05,5.72,4.79,5.72,158400,4.93
ABCSE,B7J,2008-10-09,6.30,6.41,6.00,6.02,140500,5.18
ABCSE,B7J,2008-10-08,5.60,6.47,5.60,6.28,292000,5.41
ABCSE,B7J,2008-10-07,7.59,7.59,6.66,6.69,89900,5.76
ABCSE,B7J,2008-10-06,7.83,7.90,7.00,7.40,159600,6.37
```

Sample Big Data Problem

- How to solve this problem?
  - Forget about MapReduce etc
- Parse each record
- Check closing price of stock
- If price higher than a current maximum price, change price
- Else go to next record
- Stop when you reach last line (record)

Sample Big Data Problem

- As we discussed in Lecture 1, the problem with this approach is time
- With no parallelization, it may take very long time to complete task

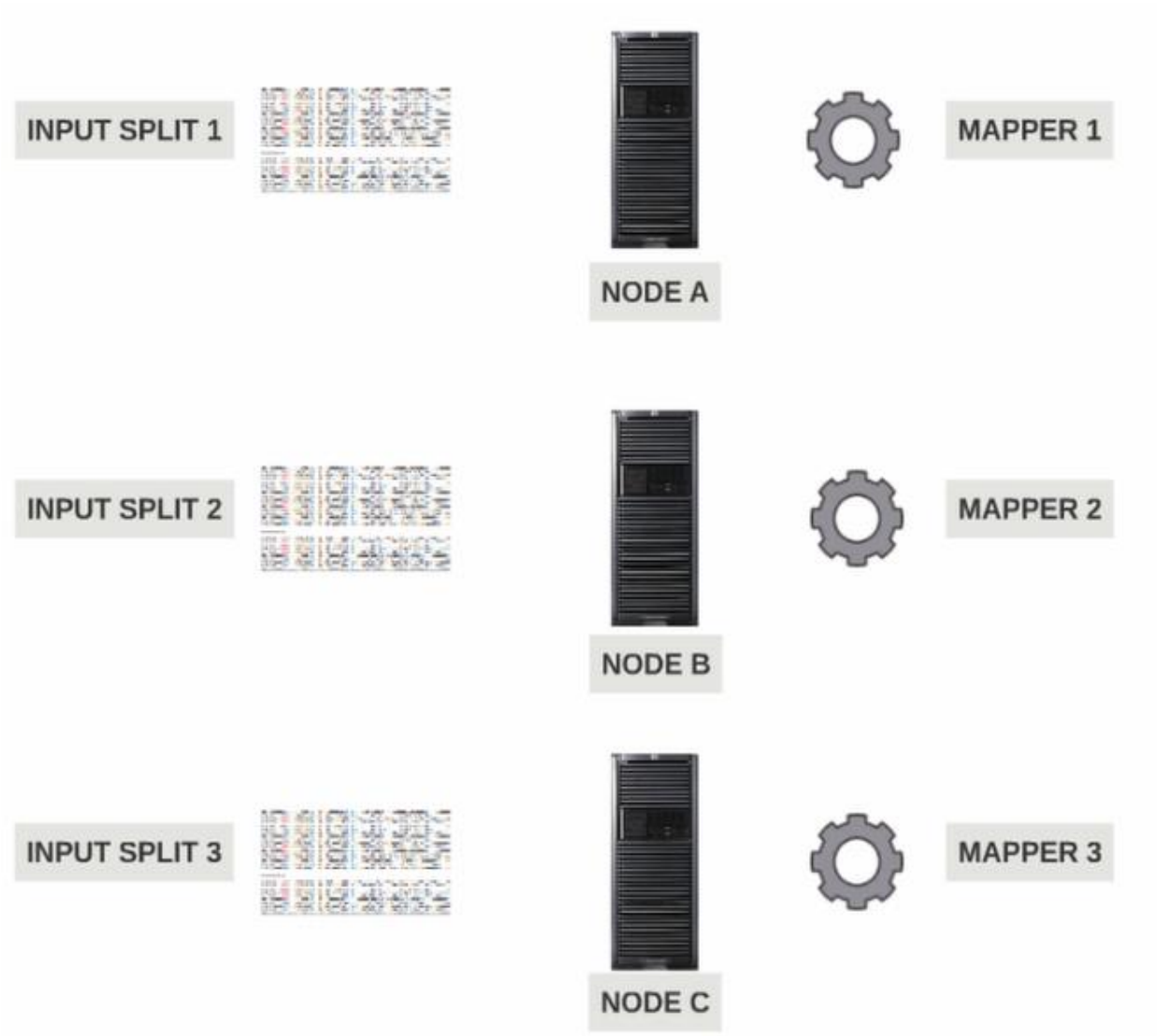
Solution in MapReduce World

- We will go through the phases we discussed in previous section
  - Kind of abstract with census example
  - Now we want to see details with an actual problem
- First talk about **Map phase**

Map Phase

- Divide the data set into chunks
- Have a separate process working on each chunk of the data
- Some technical jargon now
  - Chunks are called **input splits**
  - The process working on the chunks are called **mappers**

Graphic View



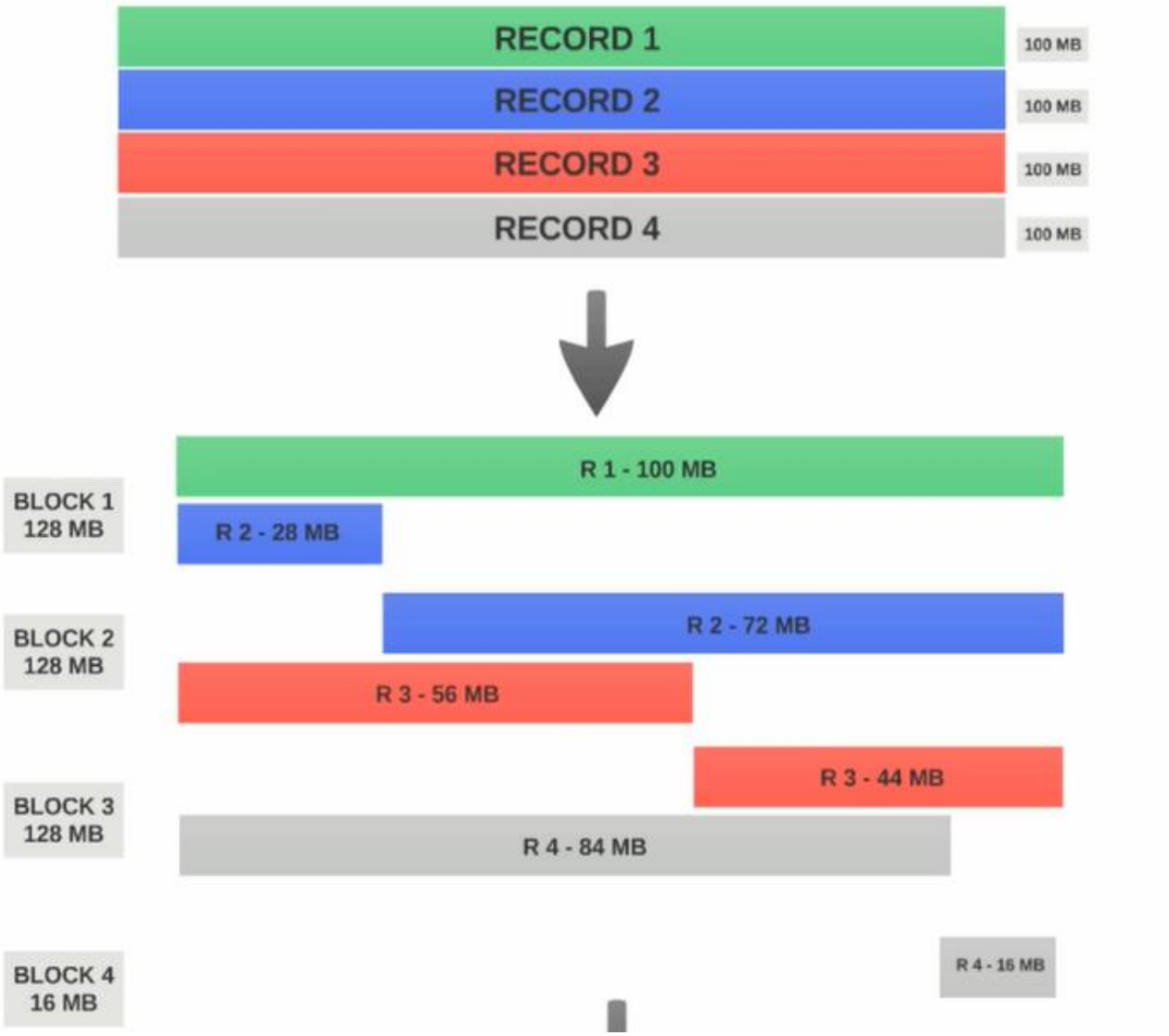
Map Phase

- Each **mapper** processes one **record** at a time
- Each mapper executes the same set of code on each record
- The output of the mapper will be a key-value pair

What is the input split?

- Same as the block?
  - No!
- Input split is not the same as the block
- A block is a hard division of data
- So a block can end BEFORE a record ends
  - Remember a record is a single line of data (stock information for one day in our example)

Graphically

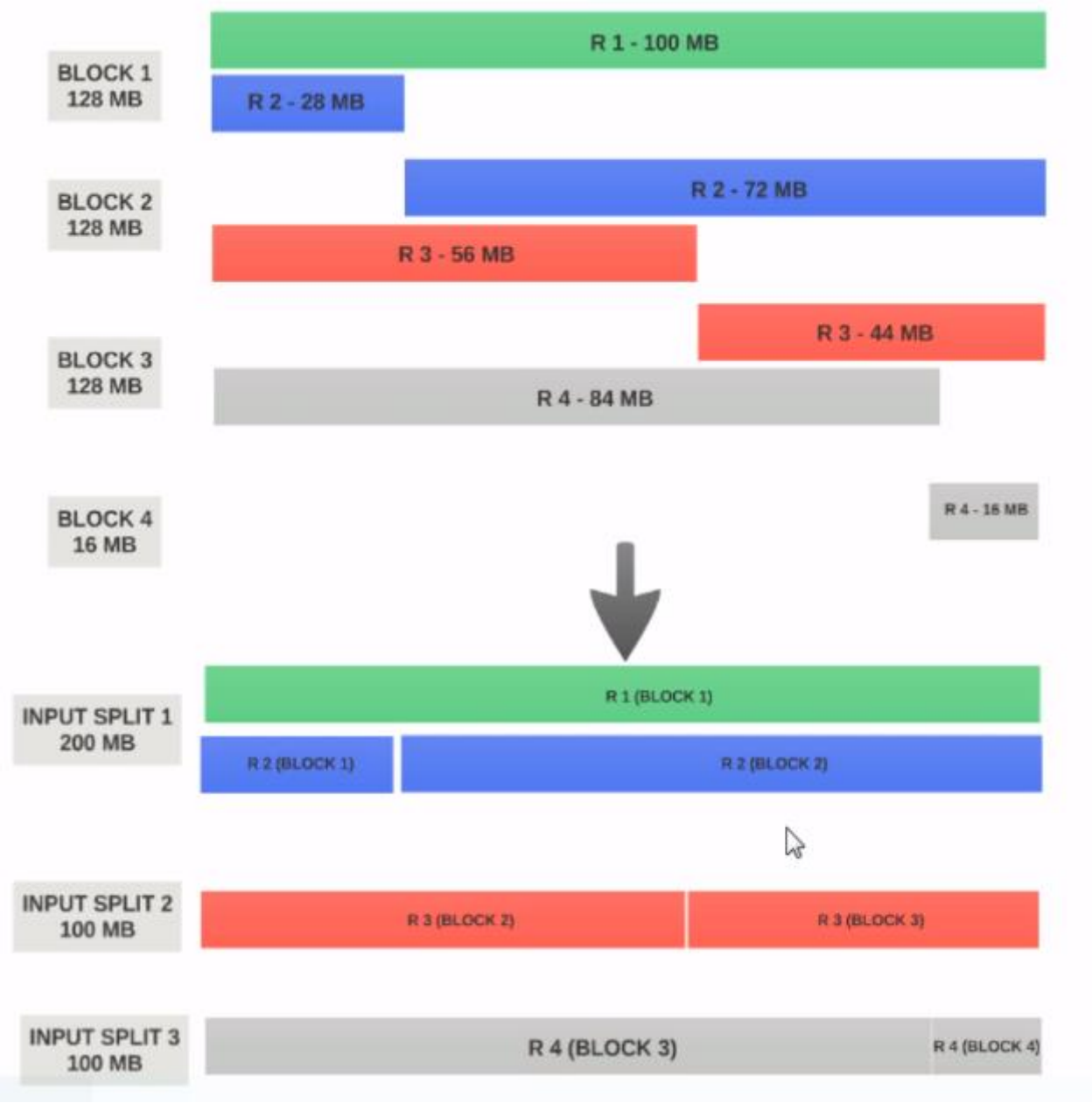


What is the input split?

- If you assign a Mapper to Block 1, the mapper cannot process Record 2, because block 1 only has a partial record
- Input split ***is not physical chunks*** of data
- It is an abstraction
  - A Java class behind the scenes with pointers to start and end location within blocks
- When a mapper tries to read the data, it knows where to start and where to end

What is the input split?

- The start position of a input split can start in one block and end in another block
- So this is why we have the concept of input split
- Input split ***respects logical record boundaries***
- During MapReduce executions, Hadoop goes through the blocks and creates input splits that respect record boundaries


BLOCK 1
128 MB
R 1 - 100 MB
R 2 - 28 MB
BLOCK 2
128 MB
R 2 - 72 MB
R 3 - 56 MB
BLOCK 3
128 MB
R 3 - 44 MB
R 4 - 84 MB
BLOCK 4
16 MB
R 4 - 16 MB
INPUT SPLIT 1
200 MB
R 1 (BLOCK 1)
R 2 (BLOCK 1)
R 2 (BLOCK 2)
INPUT SPLIT 2
100 MB
R 3 (BLOCK 2)
R 3 (BLOCK 3)
INPUT SPLIT 3
100 MB
R 4 (BLOCK 3)
R 4 (BLOCK 4)

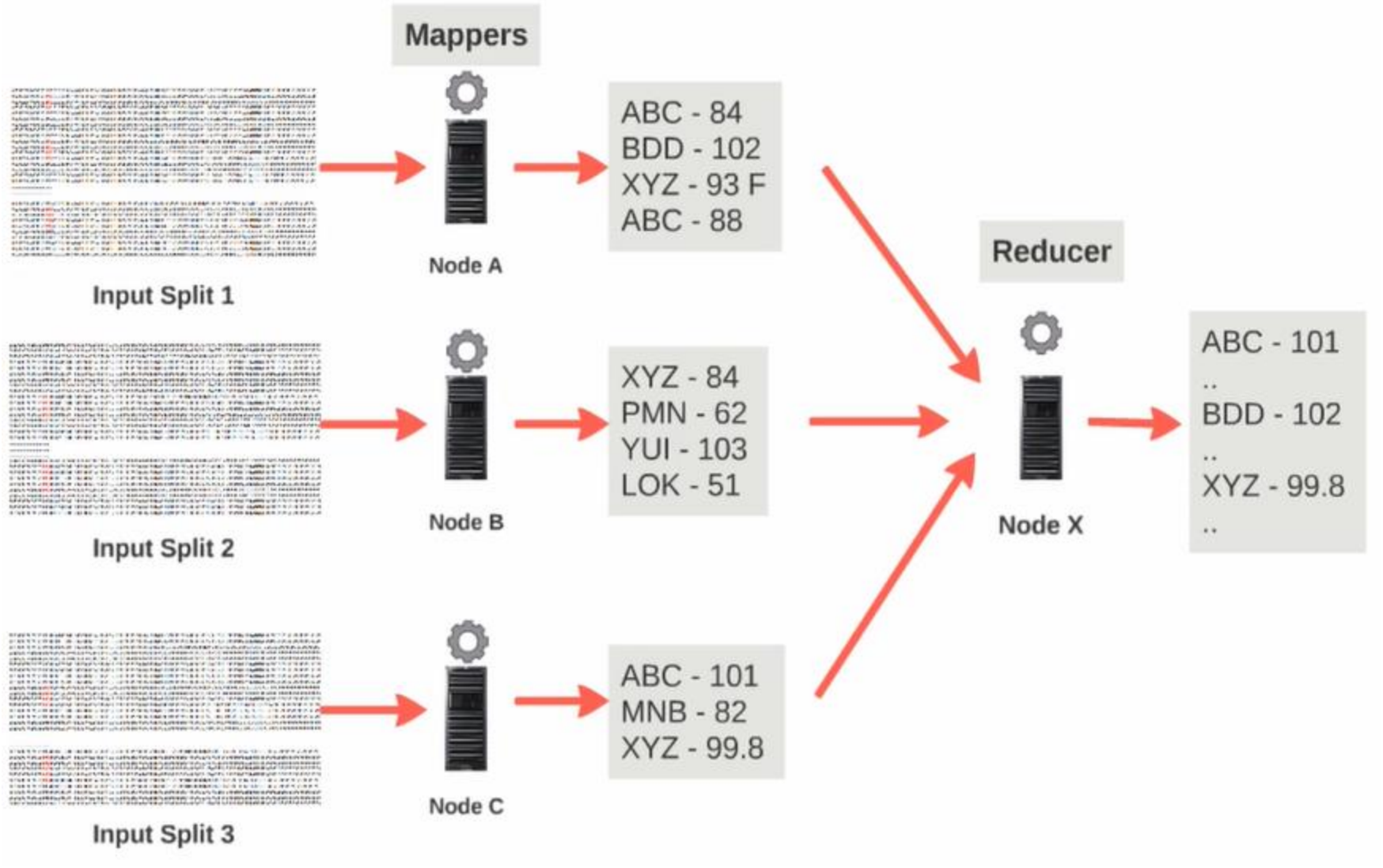
Map Phase

- Mapper is a program
  - Can be written in many different programming languages
  - In our case, it will be Java
- Mapper is therefore a Java program that is invoked by the Hadoop framework ***once per every record in the input split***
- So if you have 10 records in your input split, the mapper will be executed 10 times

Interview Question

- How many mappers does Hadoop create for a Data Set?
  - Entirely dependent on the number of input splits
  - A mapper is invoked for every single record in the input split

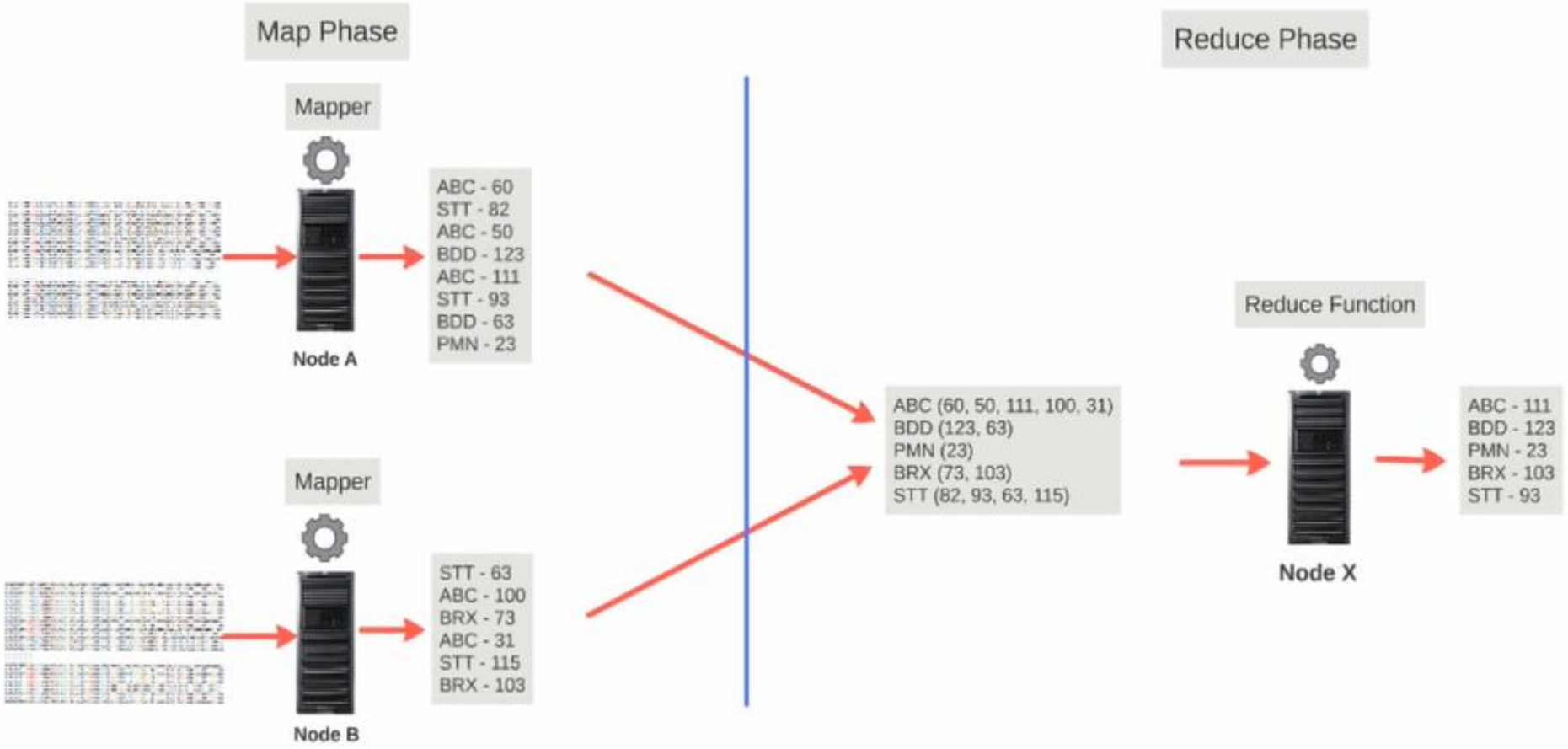
Map Phase


Mappers
Input Split 1
Node A
ABC - 84
BDD - 102
XYZ - 93 F
ABC - 88
Input Split 2
Node B
XYZ - 84
PMN - 62
YUI - 103
LOK - 51
Input Split 3
Node C
ABC - 101
MNB - 82
XYZ - 99.8
Reducer
Node X
ABC - 101
..
BDD - 102
..
XYZ - 99.8
..

Map Phase

- Again, a mapper is invoked for every single record in the input split
- The output of the mapper should be a key-value pair
- In our sample data set, every line is a record
- How do you decide what should be the key and the value for the key?
  - The reduce phase will give you an answer

Reduce Phase



Reduce Phase

- The reducers work on the output of the mappers
- The output of the individual mappers are grouped by the key and passed to the reducer
- Reducer will receive a key and list of values as input
  - The keys will be grouped

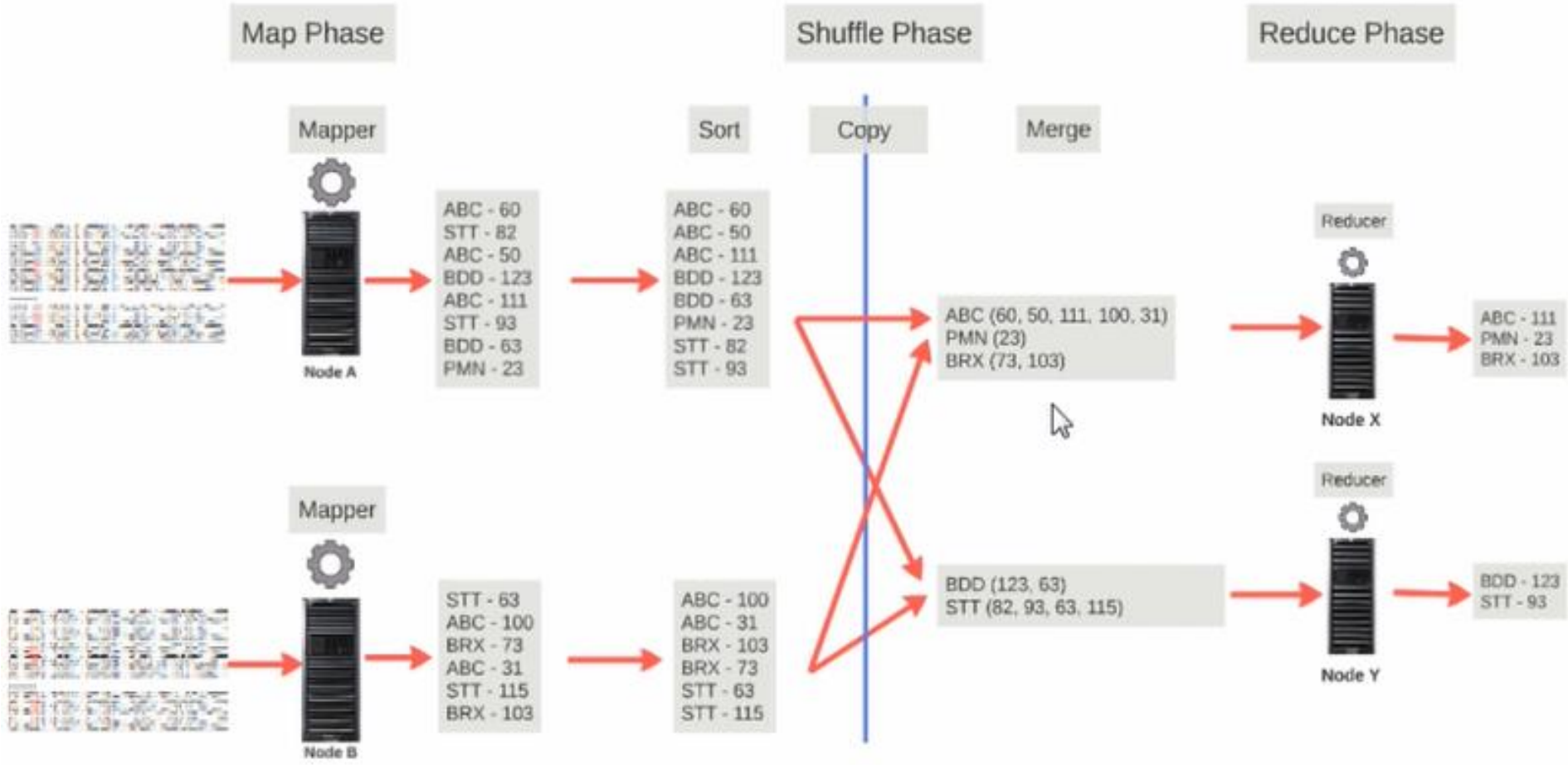
Interview Question

- How many reducers is there for a given job?
  - Set by the user
- Reducer can be a bottle neck
- So it is advisable to have more than one reducer

Shuffle

- What happens if you have more than one reducer?
- How does the output of the individual mappers got grouped by symbols and reached reducer?
- The magic happens in the **shuffle phase**
- This is the process in which the output of the mappers is transferred to the reducers is known as a shuffle

Shuffle


Map Phase
Shuffle Phase
Reduce Phase
Mapper
Sort
Copy
Merge
Reducer
ABC - 60
STT - 82
ABC - 50
BDD - 123
ABC - 111
STT - 93
BDD - 63
PMN - 23
Node A
ABC - 60
ABC - 50
ABC - 111
BDD - 123
BDD - 63
PMN - 23
STT - 82
STT - 93
ABC (60, 50, 111, 100, 31)
PMN (23)
BRX (73, 103)
ABC - 111
PMN - 23
BRX - 103
Node X
Mapper
Reducer
STT - 63
ABC - 100
BRX - 73
ABC - 31
STT - 115
BRX - 103
Node B
ABC - 100
ABC - 31
BRX - 103
BRX - 73
STT - 63
STT - 115
BDD (123, 63)
STT (82, 93, 63, 115)
BDD - 123
STT - 93
Node Y

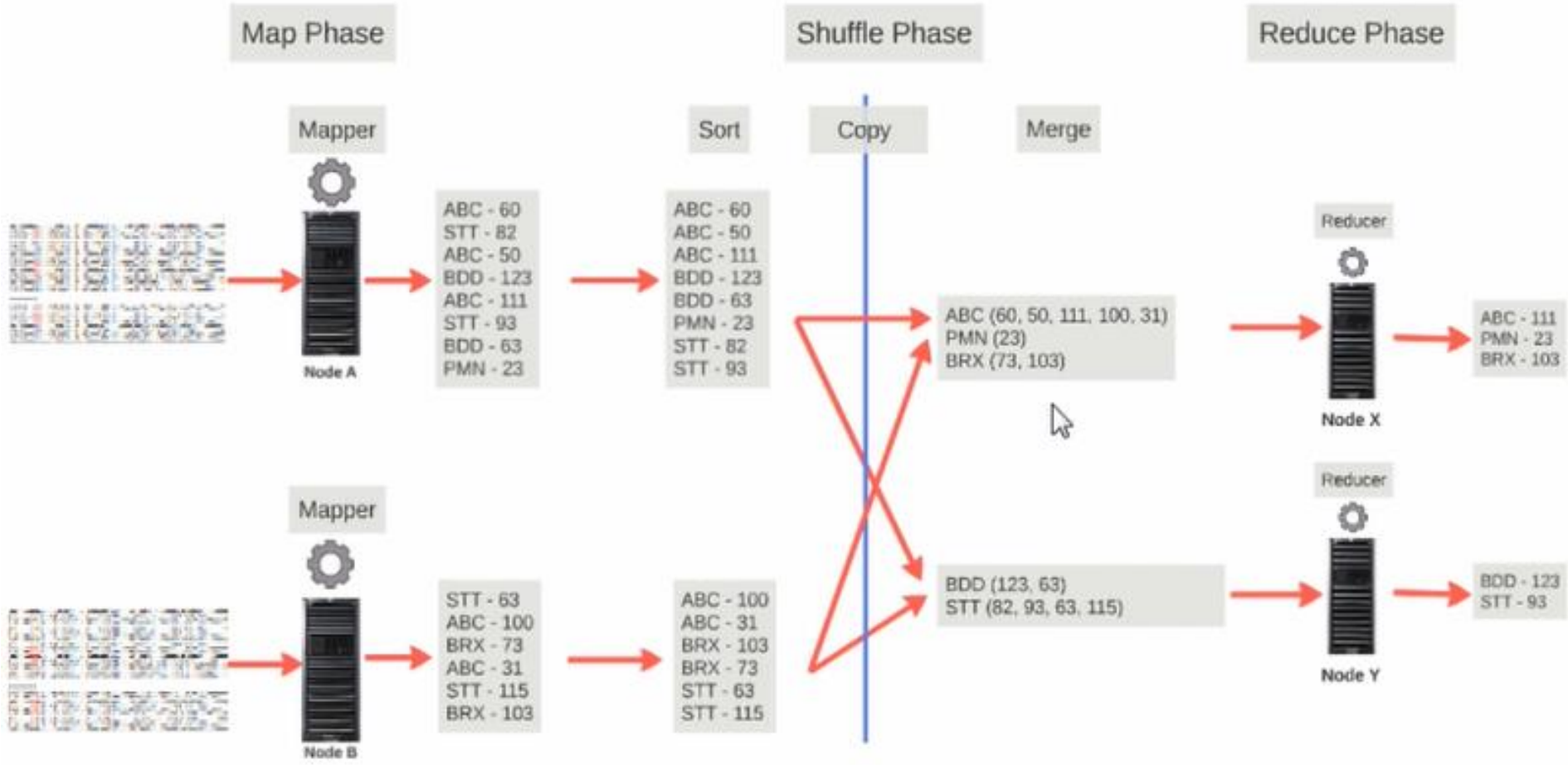
Shuffle

- Each mapper will process all the assigned records in its input split and output a key value pair
- Keys in one mapper can be found in multiple mappers

Shuffle Phase sort

- In the map phase, each key is assigned to **a partition** by a class called partitioner
  - One partition per mapper
- Within each partition, the key-value pairs will be **sorted by key**
- Once the key-value pairs are sorted, the key-value pairs are then copied to the appropriate reducers

Shuffle


Map Phase
Shuffle Phase
Reduce Phase
Mapper
Sort
Copy
Merge
Reducer
ABC - 60
STT - 82
ABC - 50
BDD - 123
ABC - 111
STT - 93
BDD - 63
PMN - 23
Node A
ABC - 60
ABC - 50
ABC - 111
BDD - 123
BDD - 63
PMN - 23
STT - 82
STT - 93
ABC (60, 50, 111, 100, 31)
PMN (23)
BRX (73, 103)
ABC - 111
PMN - 23
BRX - 103
Node X
Mapper
Reducer
STT - 63
ABC - 100
BRX - 73
ABC - 31
STT - 115
BRX - 103
Node B
ABC - 100
ABC - 31
BRX - 103
BRX - 73
STT - 63
STT - 115
BDD (123, 63)
STT (82, 93, 63, 115)
BDD - 123
STT - 93
Node Y

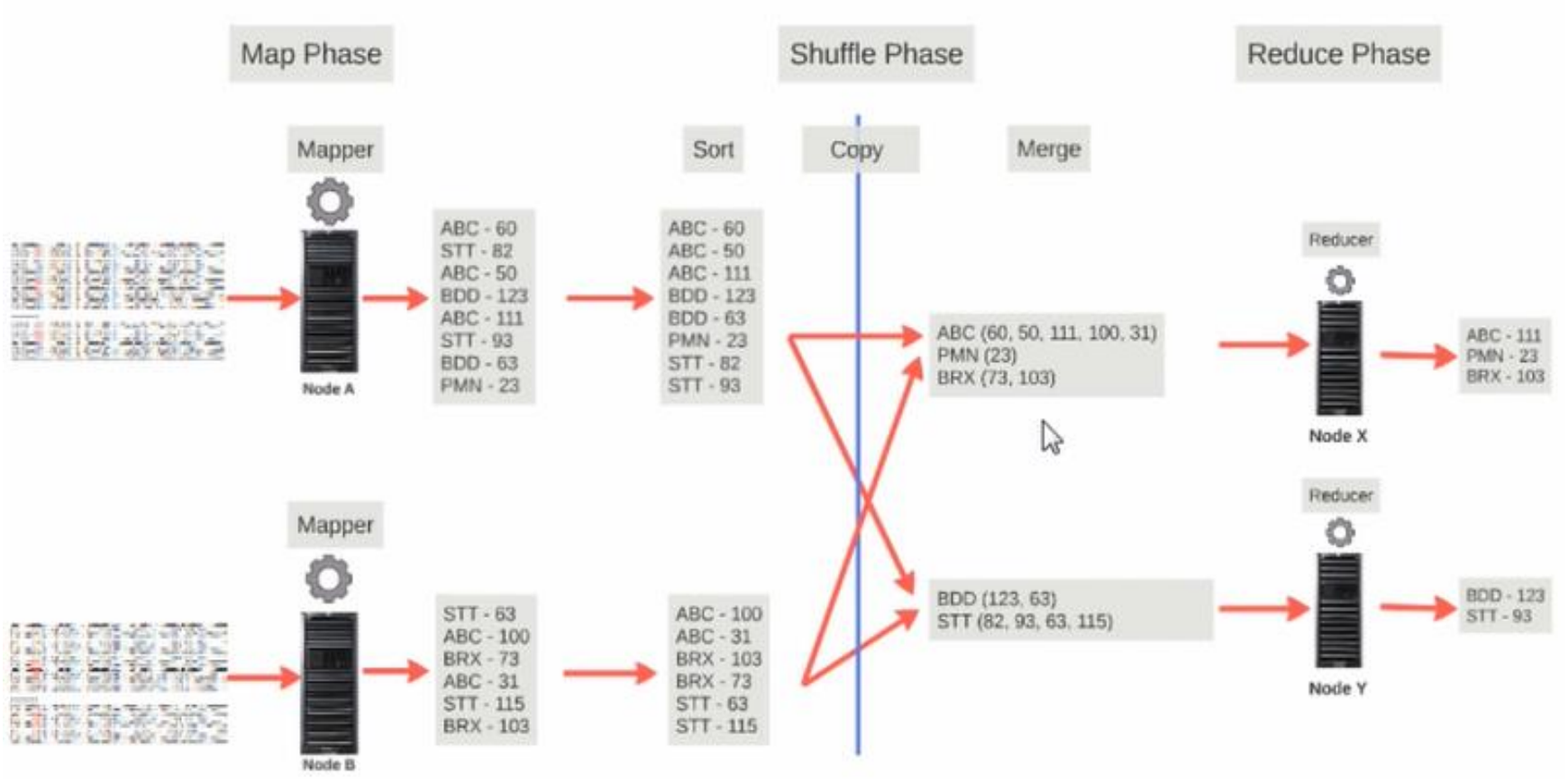
Shuffle Phase copy

- Once the key-value pairs are sorted, the key-value pairs are then copied to the appropriate reducers

Shuffle Merge

- Remember the data for partition one can come from many mappers
- The key-value pairs from different mappers will be merged maintaining the sort order
  - Hadoop framework ***guarantees*** that input the reducer is sorted by key
- Key's will be unique to each reducer

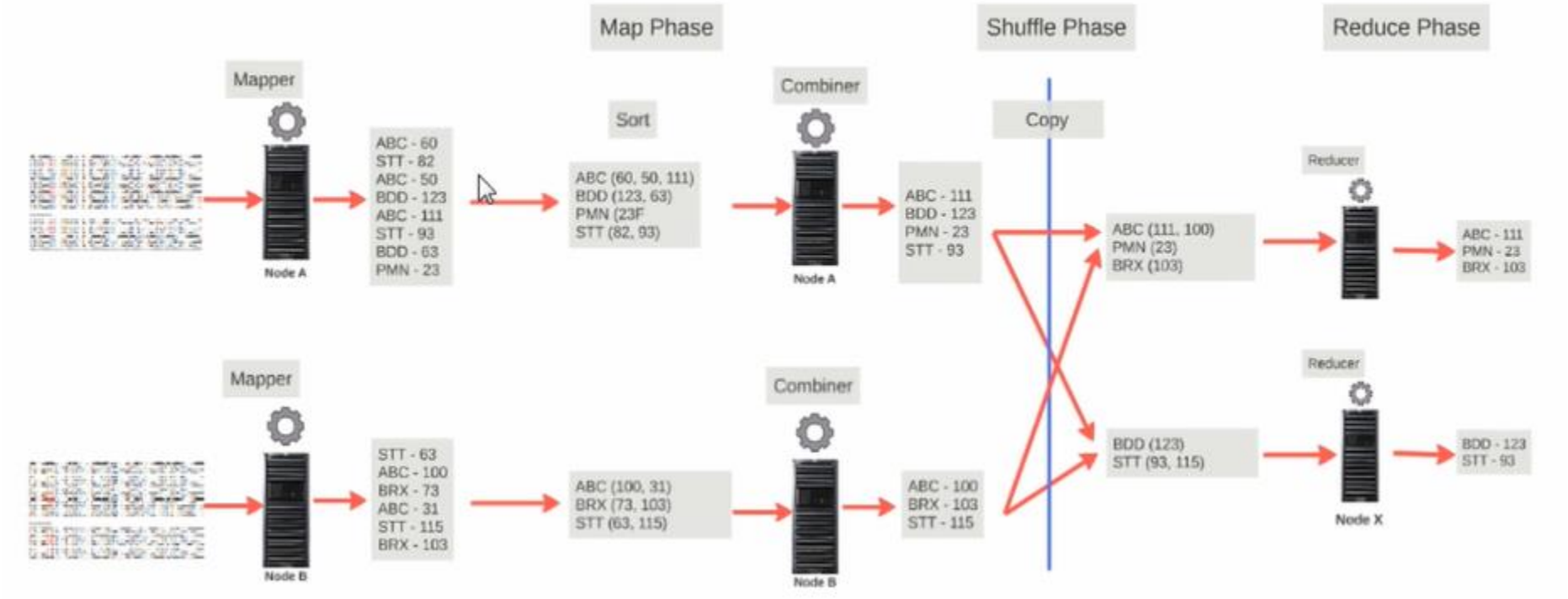
Shuffle


Map Phase
Shuffle Phase
Reduce Phase
Mapper
Sort
Copy
Merge
ABC - 60
STT - 82
ABC - 50
BDD - 123
ABC - 111
STT - 93
BDD - 63
PMN - 23
Node A
ABC - 60
ABC - 50
ABC - 111
BDD - 123
BDD - 63
PMN - 23
STT - 82
STT - 93
ABC (60, 50, 111, 100, 31)
PMN (23)
BRX (73, 103)
Reducer
ABC - 111
PMN - 23
BRX - 103
Node X
Mapper
STT - 63
ABC - 100
BRX - 73
ABC - 31
STT - 115
BRX - 103
Node B
ABC - 100
ABC - 31
BRX - 103
BRX - 73
STT - 63
STT - 115
BDD (123, 63)
STT (82, 93, 63, 115)
Reducer
BDD - 123
STT - 93
Node Y

Combiner

- We can also have a optional combiner at the map phase
- Combiner can be used to reduce the amount of data that is given to the reducer
- Combiner is like a mini-reducer that runs at the map phase

Combiner


Map Phase
Shuffle Phase
Reduce Phase
Mapper
Sort
Combiner
Copy
Reducer
ABC - 60
STT - 82
ABC - 50
BDD - 123
ABC - 111
STT - 93
BDD - 63
PMN - 23
Node A
ABC (60, 50, 111)
BDD (123, 63)
PMN (23F
STT (82, 93)
ABC - 111
BDD - 123
PMN - 23
STT - 93
ABC (111, 100)
PMN (23)
BRX (103)
ABC - 111
PMN - 23
BRX - 103
Mapper
Combiner
Reducer
STT - 63
ABC - 100
BRX - 73
ABC - 31
STT - 115
BRX - 103
Node B
ABC (100, 31)
BRX (73, 103)
STT (63, 115)
ABC - 100
BRX - 103
STT - 115
BDD (123)
STT (93, 115)
BDD - 123
STT - 93
Node X

Interview Question

- Can you use a combiner in all scenarios?

MapReduce Tutorial

- A "Hello World" example
- Take a word file, and find the number of times words appear in the file
- Follow tutorials on:

https://github.com/saberamini/MapReduceTutorial