MapReduce Framework

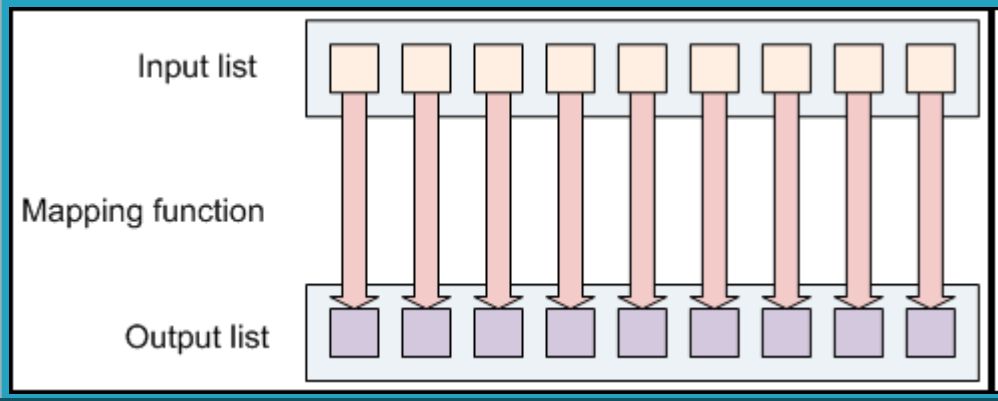
By Dinesh Amatya

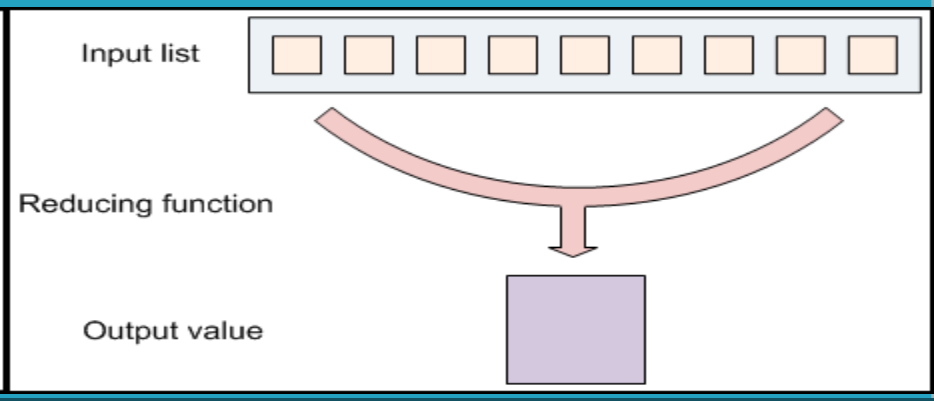
MapReduce : Introduction

- Most computations conceptually straightforward
- the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time
- most of the computations involved applying a map operation to each logical “record” in the input in order to compute a set of intermediate key/value pairs, and then applying a reduce operation to all the values that shared the same key, in order to combine the derived data appropriately

MapReduce: Programming Model

- Conceptually, MapReduce programs transform lists of input data elements into lists of output data elements
- MapReduce is a programming model for processing and generating large data sets
- A MapReduce program will do this twice, using two different list processing idioms: map, and reduce





MapReduce: Programming Model

```
map(String key, String value):
// key: document name
// value: document contents
for each word w in value:
EmitIntermediate(w, "1");
```

```
reduce(String key, Iterator values):
// key: a word
// values: a list of counts
int result = 0;
for each v in values:
result += ParseInt(v);
Emit(AsString(result));
```

MapReduce : Example

Data

0067011990999991950051507004...9999999N9+00001+99999999999...

0043011990999991950051512004...9999999N9+00221+99999999999...

0043011990999991950051518004...9999999N9-00111+99999999999...

0043012650999991949032412004...0500001N9+01111+99999999999...

0043012650999991949032418004...0500001N9+00781+99999999999...

MapReduce : Example

Input to map

(0, 006701199099999**1950**051507004...9999999N9+**0000**1+99999999999...)

(106, 004301199099999**1950**051512004...9999999N9+**0022**1+99999999999...)

(212, 004301199099999**1950**051518004...9999999N9-**0011**1+99999999999...)

(318, 004301265099999**1949**032412004...0500001N9+**0111**1+99999999999...)

(424, 004301265099999**1949**032418004...0500001N9+**0078**1+99999999999...)

MapReduce : Example

Output from map

(1950,0)

(1950,22)

(1950,−11)

(1949,111)

(1949,78)

MapReduce : Example

Output before sending to reduce function (processed by MapReduce Framework)

(1949, [111, 78])

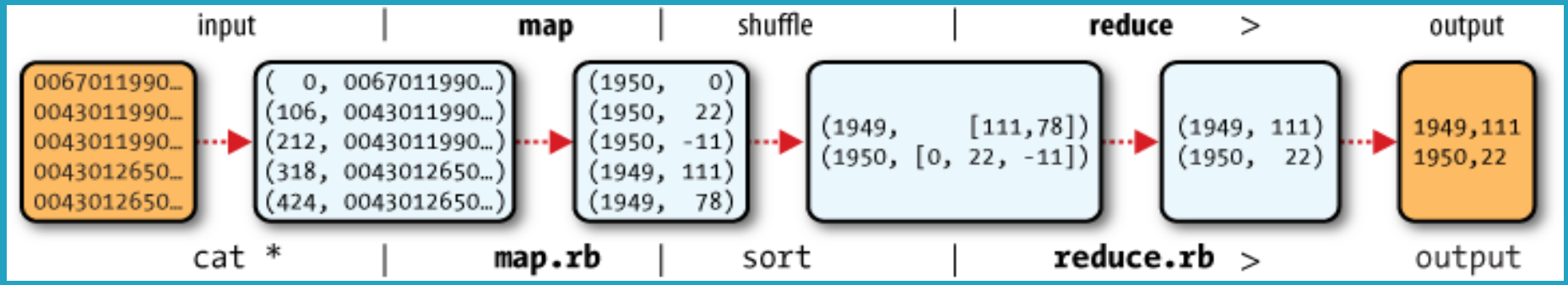
(1950, [0, 22, −11])

Output from reducer

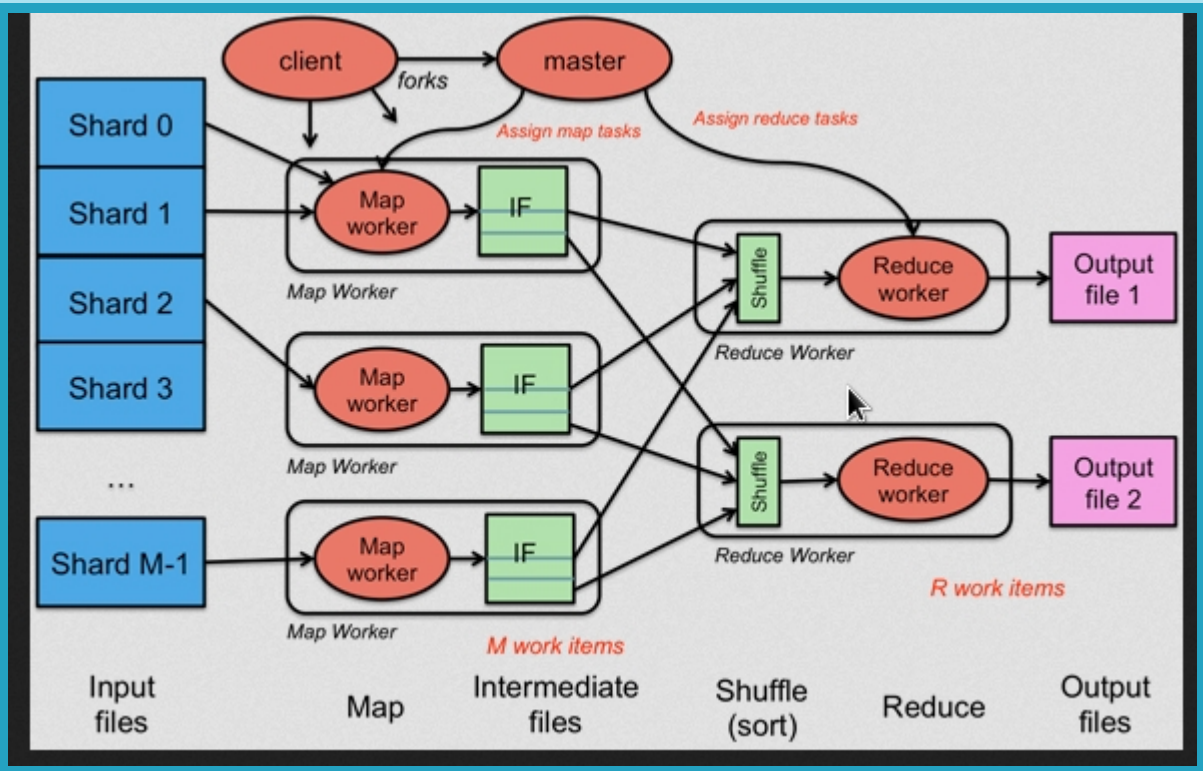
(1949, 111)

(1950, 22)

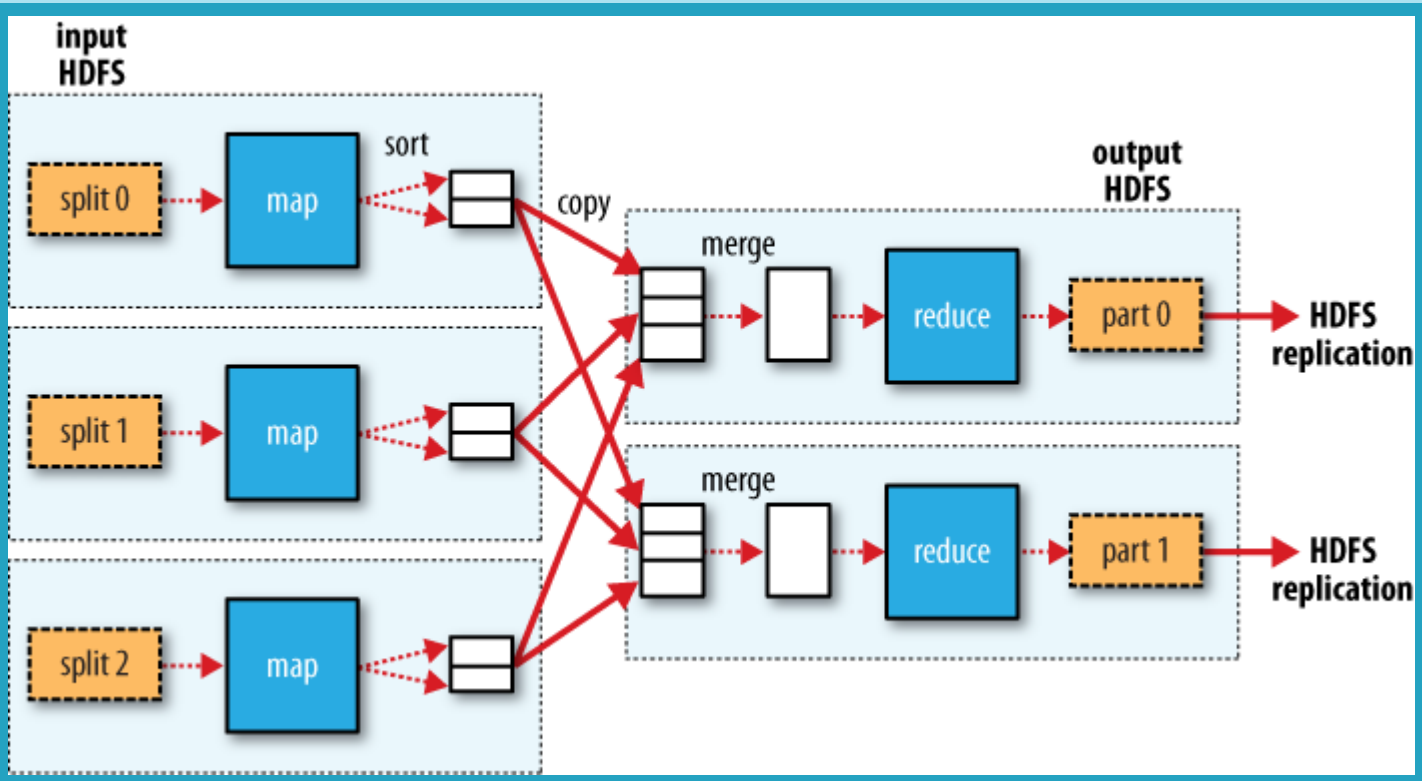
MapReduce : Example



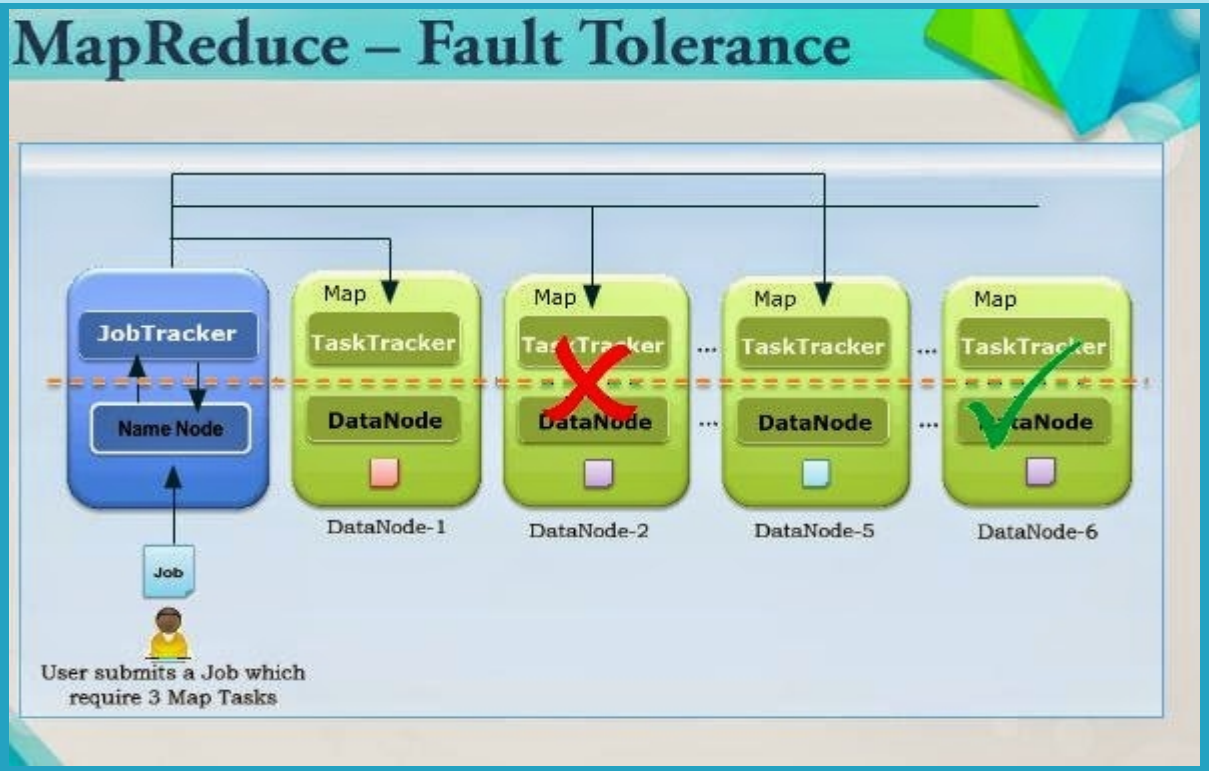
MapReduce : Execution overview



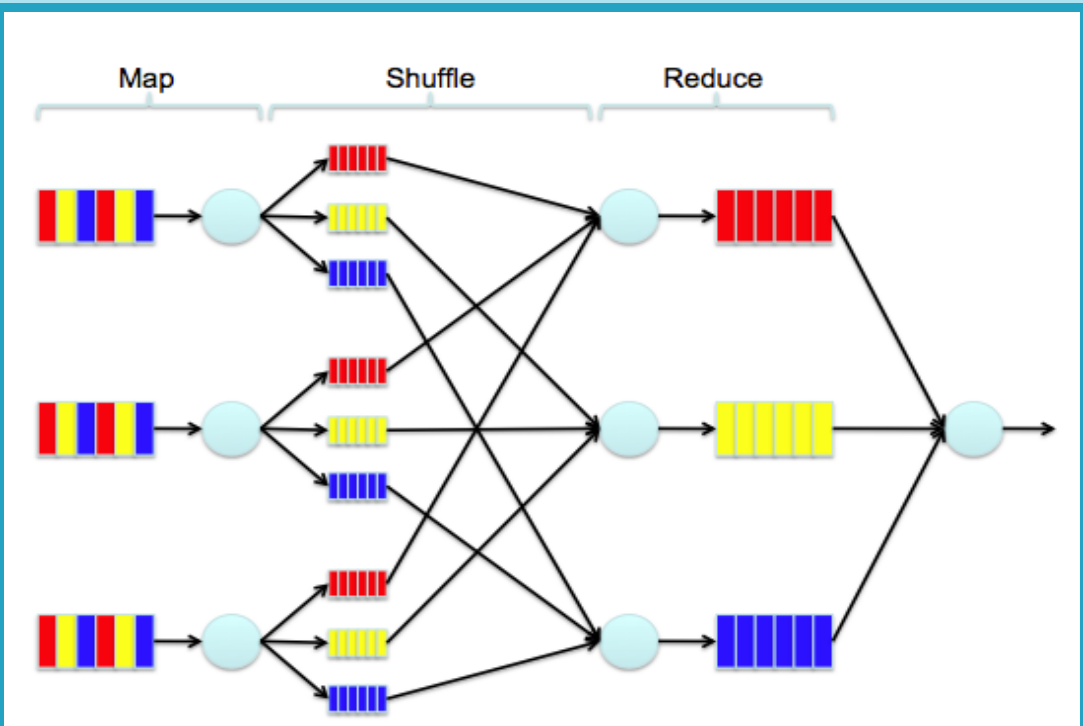
MapReduce: Data Flow



MapReduce : Fault Tolerance

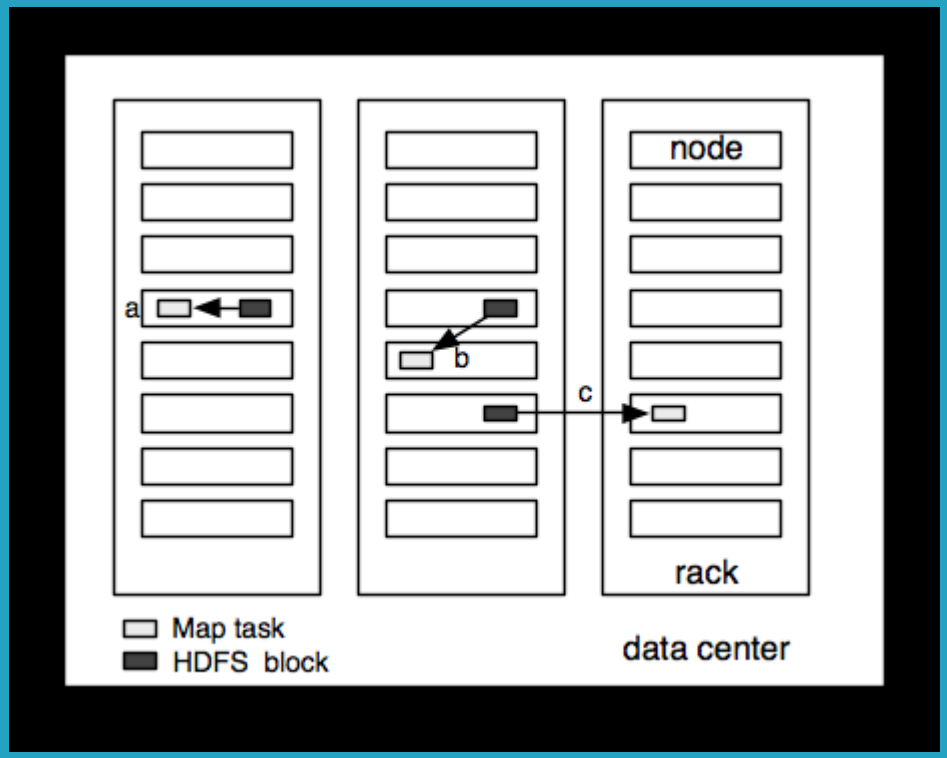


MapReduce: Locality



Data Locality Optimization

MapReduce: Locality



MapReduce: Backup Tasks

→ “straggler”: a machine that takes an unusually long time to complete one of the last few map or reduce tasks in the computation

→ when a MapReduce operation is close to completion, the master schedules backup executions of the remaining in-progress tasks

→ task is marked as completed whenever either the primary or the backup execution completes

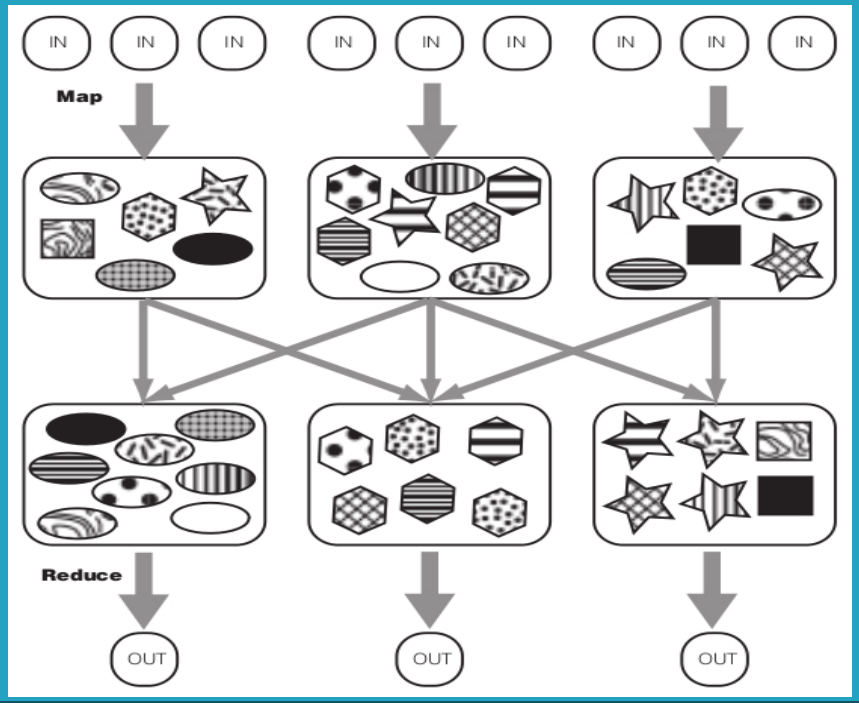
MapReduce: Partitioner

→ partitioning phase takes place after the map phase and before the reduce phase

→ number of partitions is equal

to the number of reducers

→ default is HashPartitioner

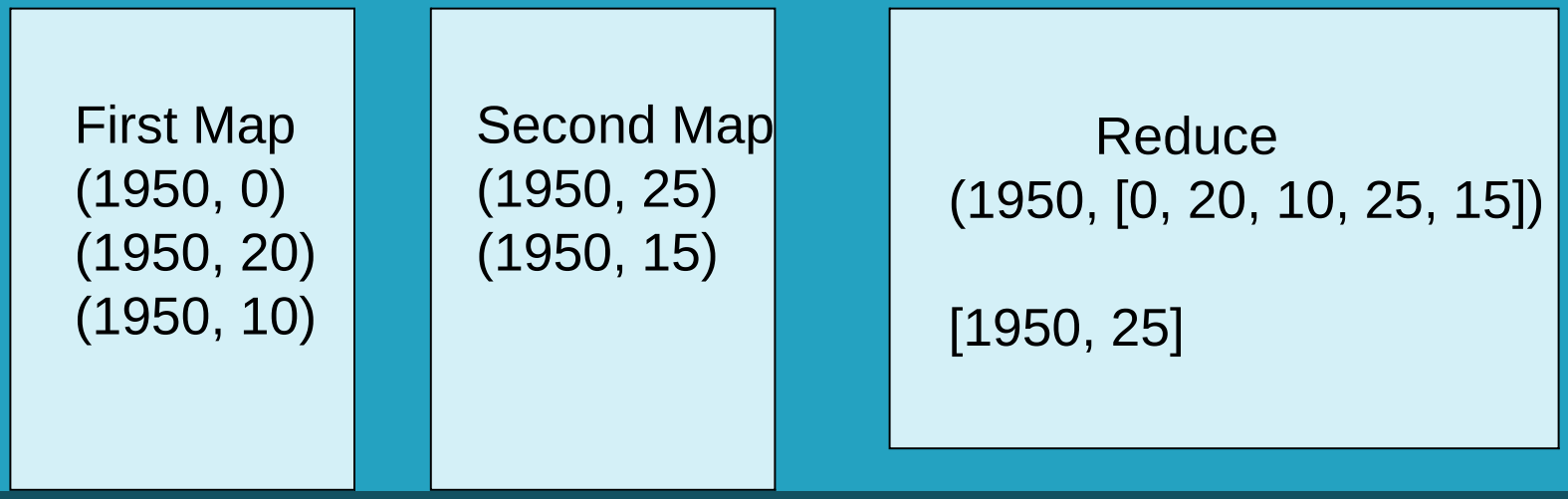


MapReduce: Combiner

→ one reducer overwhelmed due to uneven distribution of key

→ helper for reducer

→ reduce network traffic and increase performance



MapReduce: Combiner

→ local reducer

First Map
(1950, 0)
(1950, 20)
(1950, 10)

[1950,20]

Second Map
(1950, 25)
(1950, 15)

[1950,25]

Reduce
(1950, [20, 25])

[1950, 25]

Java MapReduce: Data Types

BooleanWritable Wrapper for a standard Boolean variable

ByteWritable Wrapper for a single byte

DoubleWritable Wrapper for a Double

FloatWritable Wrapper for a Float

IntWritable Wrapper for a Integer

LongWritable Wrapper for a Long

Text Wrapper to store text using the UTF8 format

Java Mapreduce: Map Function

```
public class MaxTemperatureMapper extends Mapper<LongWritable, Text, Text, IntWritable> {

@Override

public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

    String line = value.toString();

    String year = …….

    int airTemperature = ……..

    context.write(new Text(year), new IntWritable(airTemperature));

}
```

Java Mapreduce: Reduce Function

```
public class MaxTemperatureReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

@Override

public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException{

    int maxValue = ……

    // logic to get max value

    context.write(key, new IntWritable(maxValue));

}

}
```

Java Mapreduce: Main class to run the job

```
public class MaxTemperature {
    public static void main(String[] args) throws Exception{
        Job job = new Job();
        job.setJarByClass(MaxTemperature.class);
        job.setJobName("max temperature");

         FileInputFormat.addInputPath(job, new Path("guide/temperature.txt"));
        FileOutputFormat.setOutputPath(job,new Path("guide/max"));

        job.setInputFormatClass(TextInputFormat.class);
```

Java Mapreduce: Main class to run the job

```
    job.setMapperClass(MaxTemperatureMapper.class);
    job.setReducerClass(MaxTemperatureReducer.class);

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    System.exit(job.waitForCompletion(true)?0:1);


  }
}
```

Java Mapreduce: Command to run job

hadoop jar target/hadoop-first-1.0-SNAPSHOT.jar first.loader.MaxTemperature

Java Mapreduce: Partitioner

```
public class TemperaturePartitioner extends Partitioner<LongWritable, Text> {
@Override
public int getPartition(LongWritable key, Text value, int numPartitions) {
    // If year less than 1980 return 0
    // else return 1
}
}
```

→ *job.setPartitionerClass(TemperaturePartitioner.class);*

→ *job.setNumReduceTasks(2);*

Java Mapreduce: Combiner

job.setCombinerClass(MaxTemperatureReducer.class);

Figures

- Slide 9 (Hadoop: The definitive guide , pg 22)
- Slide 11 (Hadoop: The definitive guide , pg 57)
- Slide 16 (Hadoop in Action, pg 50)

References

- Hadoop : The definitive Guide
- Hadoop in action
- MapReduce Paper
- http://www.quora.com/Any-real-life-use-case-of-Apache-Hadoop
- https://www.cs.rutgers.edu/~pxk/417/notes/content/mapreduce.html
- http://bcuz-my-data-is-big.blogspot.com/2014/10/introduction-to-apache-hadoop-and-its.html
- http://hadooptutorial.wikispaces.com/Custom+partitioner
- http://dailyhadoopsoup.blogspot.com/2014/01/partitioning-in-mapreduce.html
- http://man7.org/linux/man-pages/man2/fork.2.html