Biomedical Engineering
The Second Workshop

MapReduce

Dr. Hisham Mohamed
Sensirion AG, Switzerland
Overview

• MapReduce programming model

• Hadoop framework

• Conclusion
MapReduce

A programming model for large-scale distributed data processing

– Simple
  • Building blocks!
  • Hide messy internals from users

– Powerful
  • High level parallelization
  • Efficient failures handling

– Extensible
Programming Model

Map\((k,v) \rightarrow (k', v')\)

\((k, v)\)  
\rightarrow \text{Mapper} \rightarrow (k', v')

GroupByKey()

\((k', v')\)

Reduce\((k', []) \rightarrow (k', v'')\)

\((k', v'')\)

Output

Reduce

\((k', [])\)
MapReduce Execution

User program

Master

Worker

Worker

Worker

Worker

Worker

Worker

Worker

Working with input (k, v)

Mapping (k', v')

Local write

Remote read

Remote read

Reducing

Output 0

Output 1

Input

Mapping

Local intermediate files (k',v')

Reducing

Output

Split 0

Split 1

Split 2

Split 3

Split 4

Split 5

Split 6

Split 7
Example: Word Count

@pharaoh tako · 19h
Nefertiti the legend of ancient Egyptian beauty 😍
#Egypt
#tourism

@ahramonline · 12m
World’s most expensive IKEA Billy bookcase sold in #Egypt: Bloomberg survey
english.ahram.org.eg/News/161218.as...

@ahramonline · 1h
Number of women voters so far “four times greater” than that of men in #Egypt's elections
english.ahram.org.eg/News/161179.as...

@RT_com · 4h
Apple ordered to pay $234 million in damages over patent violation on.rt.com/6u2s

@AlArabia_Eng · 5h
BREAKING: #Egypt begins voting to elect a new parliament ara.tv/5dtnw

@RelaxInVenice · Oct 3
RelaxInRome: nnnnmmmm454: RT samirasaleh157:
#Discover_Egypt_come_2Egypt
#travel #tourism #enjoy
#Egypt #PHOTOGR...

@NihadAhmed · Apr 6
White desert مصر #Egypt #Definitely_Warm_Egypt #discover_egypt_come2_egypt #togeeter #tourism #travel
#ENJOY
Example: Word Count

```python
def word_count(key, value):
    ""
    @key: Twitter id
    @value: Tweet
    ""
    count = {}
    line = value.split(" ")
    for word in line:
        if word not in count:
            count[word] = 1
        else:
            count[word] += 1
    return count
```
Word Count: Mapping

Input

Worker 1

Worker 2

Worker 3

Map(twitter_id, tweet) --> (word, 1)
Map(twitter_id, tweet) --> (word, 1)

\texttt{def map(key, value):}

\texttt{"""
@key: Twitter id
@value: Tweet
""""}

\texttt{line = value.split(" ")}

\texttt{for word in line:}

\texttt{emit(word, 1)}
Word Count

Input

Worker 1:
```python
def map(key, value):
    key: Twitter id
    value: Tweet
    line = value.split(' ')
    for word in line:
        emit(word, 1)
```

Worker 2:
```python
def map(key, value):
    key: Twitter id
    value: Tweet
    line = value.split(' ')
    for word in line:
        emit(word, 1)
```

Worker 5:
```python
def reduce(key, value):
    @key: word
    @value: list of ones; [1,1,1,1,..]
    emit(word, sum(value))
```

(ancient, 1)
(breaking, 2)
(begins, 2)
...
Overview

• MapReduce programming model

• Hadoop framework

• Conclusion
Hadoop

• First public implementation of MapReduce

  – Hadoop Distributed File system (HDFS) to handle input, intermediate and output data

  – Cluster management
    • Hardware failure
    • High throughput

  – MapReduce

HDFS Architecture

- Master/Slave architecture

[Diagram of HDFS Architecture]

More information can be found at:
https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html#NameNode+and+DataNodes
Hadoop Word Count

```java
public class WordCount {
    public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                word.set(tokenizer.nextToken());
                output.collect(word, one);
            }
        }
    }

    public static class Reduce extends MapReduceBase implementsReducer<Text, IntWritable, Text, IntWritable> {
        public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
            int sum = 0;
            while (values.hasNext()) {
                sum += values.next().get();
            }
            output.collect(key, new IntWritable(sum));
        }
    }

    public static void main(String[] args) throws Exception {
        JobConf conf = new JobConf(WordCount.class);
        conf.setJobName("wordcount");
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
        conf.setMapperClass(Map.class);
        conf.setCombinerClass(Combiner.class);
        conf.setReducerClass(Reducer.class);
        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);
        TextInputFormat.setInputPaths(conf, new Path(args[0]));
        TextOutputFormat.setOutputPath(conf, new Path(args[1]));

        JobClient.runJob(conf);
    }
}
```
Hadoop Word Count

- 190+ parameters in Hadoop
Does MapReduce fit all problems?

- MapReduce only fits *embarrassingly parallel problems*

- Mapreduce is not suitable for:
  - Iterative algorithms
  - Data sharing while processing
  - Online learning algorithms
  - Monte Carlo simulations
  - ....
Conclusion

• Data and Algorithms parallelization

• MPI, OpenMP and MapReduce

• Choosing the right programming model depends on the application
References

- https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html