CSCE 5300 Introduction to Big Data and Data Science

Lecture 3

Zeenat Tariq, Ph.D.

Hadoop MapReduce and Hadoop Distributed File System (HDFS)

WordCount Main Class

```java
// Create configuration
Configuration conf = new Configuration(true);

// Create job
Job job = new Job(conf, "WordCount");
job.setJarByClass(WordCountMapper.class);

// Setup MapReduce
job.setMapperClass(WordCountMapper.class);
job.setReducerClass(WordCountReducer.class);
job.setNumReduceTasks(1);

// Specify key / value
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);

// Input
FileInputFormat.addInputPath(job, inputPath);
job.setInputFormatClass(TextInputFormat.class);

// Output
FileOutputFormat.setOutputPath(job, outputDir);
job.setOutputFormatClass(TextOutputFormat.class);

// Delete output if exists
FileSystem hdfs = FileSystem.get(conf);
if (hdfs.exists(outputDir))
    hdfs.delete(outputDir, true);

// Execute job
int code = job.waitForCompletion(true) ? 0 : 1;
System.exit(code);
```

WordCount Mapper

I am a student
I like reading books

hadoop_wordcount × | WordCount.java × | WordCountMapper.java × | WordCountReducer.java ×

```java
/**
 * Created by Mayanka on 03-Sep-15.
 */
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class WordCountMapper extends
        Mapper<Object, Text, Text, IntWritable> {

    private final IntWritable ONE = new IntWritable(1);
    private Text word = new Text();

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

        String[] csv = value.toString().split(" ");
        for (String str : csv) {
            word.set(str);
            context.write(word, ONE);
        }
    }
}
```

I: 1
am: 1
a: 1
student: 1

I: 1
like: 1
reading: 1
books: 1

WordCount Reducer

hadoop_wordcount × | WordCount.java × | WordCountMapper.java × | WordCountReducer.java ×

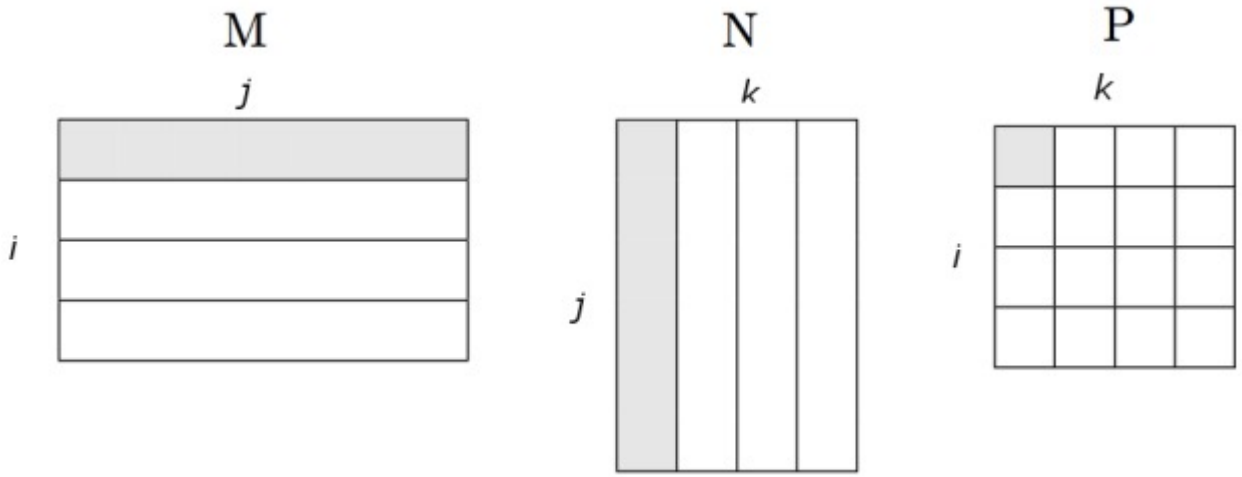
```java
/**
 * Created by Mayanka on 03-Sep-15.
 */
import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class WordCountReducer extends
        Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text text, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable value : values) {
            sum += value.get();
        }
        context.write(text, new IntWritable(sum));
    }
}
```

I: 2
am: 1
a: 1
student: 1

like: 1
reading: 1
books: 1

Advanced Map Reduce Algorithms

Matrix Multiplication



Map Function

Algorithm 1: The Map Function

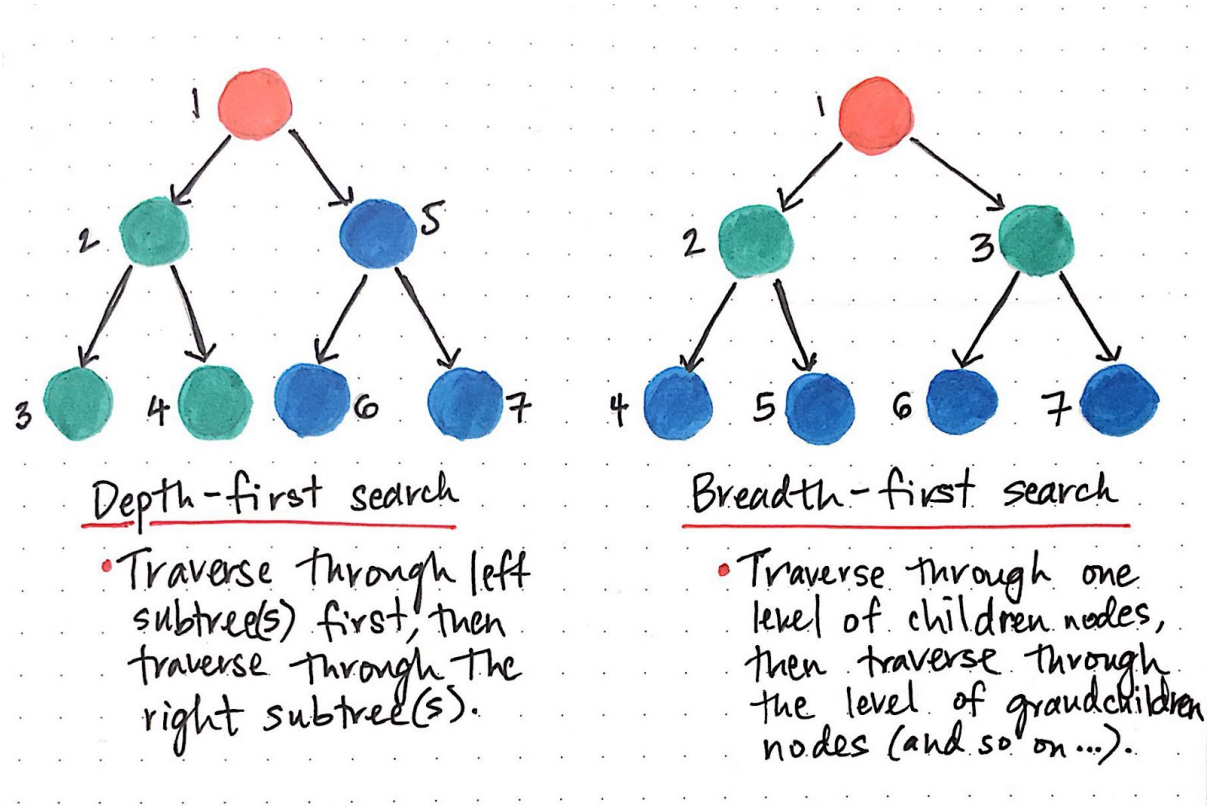
```
for each element m_ij of M do
    produce (key, value) pairs as ((i, k), (M, j, m_ij)), for k = 1, 2, 3, .. up
    to the number of columns of N
for each element n_jk of N do
    produce (key, value) pairs as ((i, k), (N, j, n_jk)), for i = 1, 2, 3, ... up
    to the number of rows of M
return Set of (key, value) pairs that each key, (i, k), has a list with
values (M, j, m_ij) and (N, j, n_jk) for all possible values of j
```

Reduce Function

Algorithm 2: The Reduce Function

```
for each key (i,k) do
    sort values begin with M by j in list_M
    sort values begin with N by j in list_N
    multiply m_ij and n_jk for j_th value of each list
    sum up m_ij * n_jk
return (i,k), Σ_{j=1} m_ij * n_jk
```

Breadth First Search / Depth First Search



References

- http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/CommandsManual.html
- https://hadoopi.wordpress.com/2013/05/25/setup-maven-project-for-hadoop-in-5mn/

References

- Map Reduce Patterns
  - https://highlyscalable.wordpress.com/2012/02/01/mapreduce-patterns/
- Matrix Multiplication
  - http://www.mathcs.emory.edu/~cheung/Courses/554/Syllabus/9-parallel/matrix-mult.html
  - https://lendap.wordpress.com/2015/02/16/matrix-multiplication-with-mapreduce/
- Breadth First Search
  - https://www.programiz.com/dsa/graph-bfs
  - https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/tutorial/
  - https://www.programiz.com/dsa/graph-dfs

Additional Resources

- Cloudera
  - https://archive.cloudera.com/cdh5/cdh/5/hadoop/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html
- Hadoop Commands Cheat sheet:
  - https://hadoop.apache.org/docs/r1.2.1/commands_manual.pdf
  - https://linoxide.com/images/hadoop-hdfs-commands-cheatsheet-900x1500.png
  - https://www.thegeekstuff.com/2015/02/hadoop-command-reference/#comments