Docker Hadoop

What is Docker?

- Lightweight VM (sort of correct)
- Good vehicle for delivering an application
- Runs on Linux, Mac and Windows
- Images versus containers

Running Docker Hadoop

```
$ docker pull sequenceiq/hadoop-docker:2.3.0
```

```
/var/lib/docker
~/Library/Containers/com.docker.docker
~/.docker/machine/machines/default
```

```
$ docker run -it sequenceiq/hadoop-docker:2.3.0 /etc/bootstrap.sh -bash
```

```
/
Starting sshd:                                [  OK  ]
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-b524750d0773.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-b524750d0773.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-b524750d0773.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn--resourcemanager-b524750d0773.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-b524750d0773.out
bash-4.1#
```

```
$ cd $HADOOP_PREFIX
$ export JAR=share/hadoop/mapreduce/hadoop-mapreduce-examples-2.3.0.jar
$ bin/hadoop jar $JAR wordcount input output
```

```
$ bin/hdfs dfs –ls
```

```
Found 2 items
drwxr-xr-x   - root supergroup          0 2014-10-23 08:14 input
drwxr-xr-x   - root supergroup          0 2016-09-13 01:29 output
```

```
$ bin/hdfs dfs -ls output
```

```
Found 2 items
-rw-r--r--   1 root supergroup          0 2016-09-13 01:29 output/_SUCCESS
-rw-r--r--   1 root supergroup      30038 2016-09-13 01:29 output/part-r-00000
```

```
$ bin/hdfs dfs -cat output/*
```

```
!=      3
""      6
"$HADOOP_CLASSPATH"         1
```

WordCount.java

```
package org.apache.hadoop.examples;

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;
import org.apache.hadoop.util.GenericOptionsParser;
```

WordCount.java

```
public class WordCount {

    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);
            }
        }
    }
```

WordCount.java

```
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);
    }
}
```

WordCount.java

```
    public static void main(String[] args) throws Exception {

        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        if (otherArgs.length != 2) {
            System.err.println("Usage: wordcount <in> <out>");
            System.exit(2);
        }
        Job job = new Job(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(IntSumReducer.class);
        job.setReducerClass(IntSumReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
        FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Assignments