



sbt-spark-submit

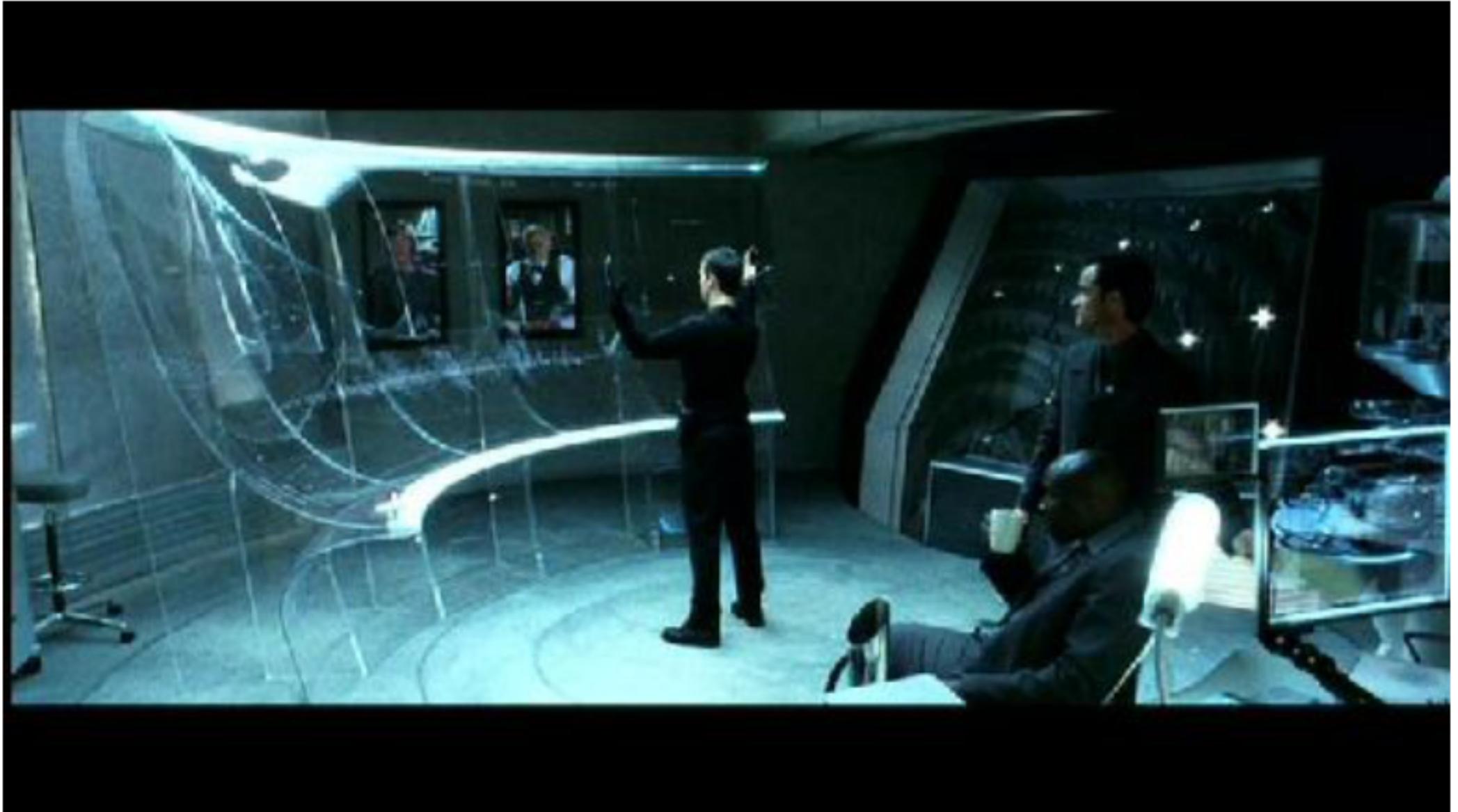
Straight-through Build to Spark Analytics

Forest Fang

Financial Modeling Group - Advanced Data Analytics

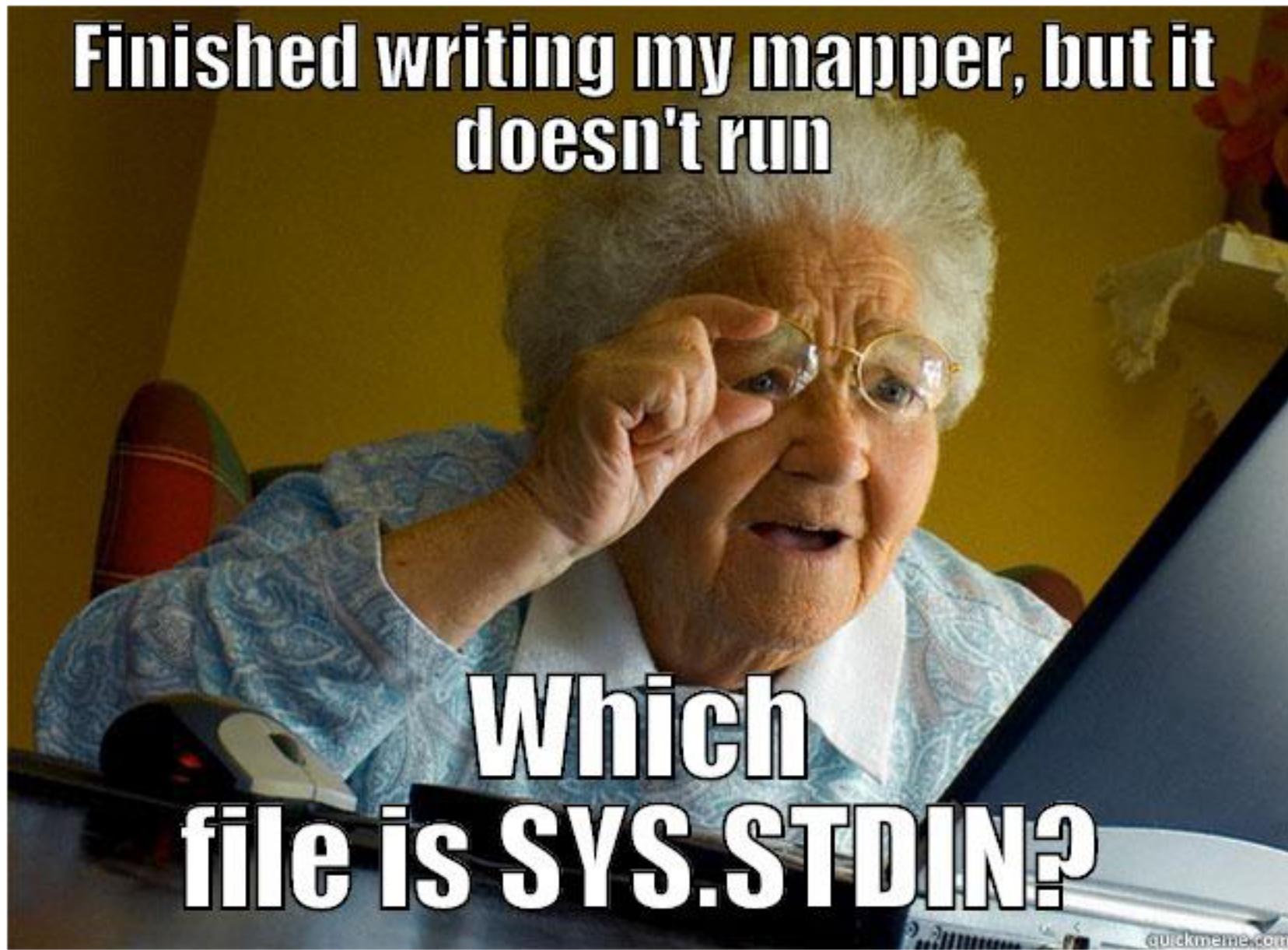
2015-09-24

What People Think We Do



Source: <https://99designs.com/designer-blog/2013/07/19/fantasy-user-interfaces/>

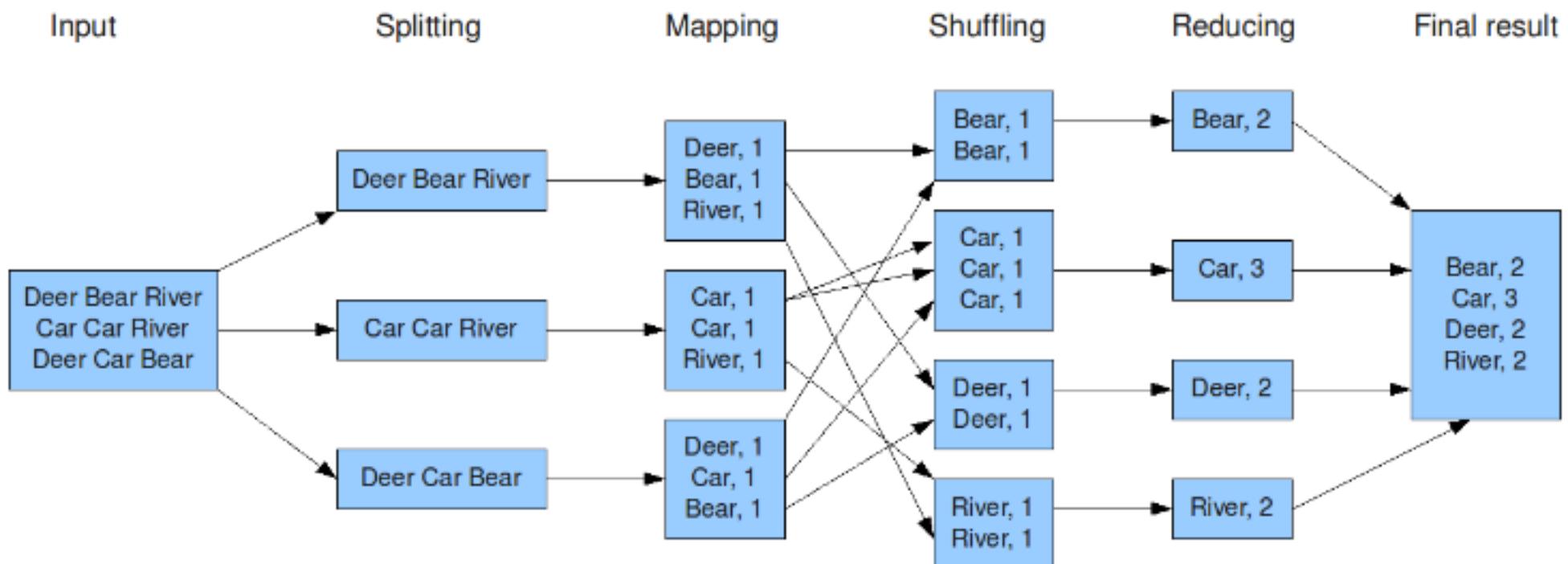
What We Really Do



Source: <http://www.quickmeme.com/p/3vovpz>

Word Count

The overall MapReduce word count process



"

Source: <http://www.milanor.net/blog/?p=853>

Word Count So Easy!

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.net.URI;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
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.mapreduce.Counter;
import org.apache.hadoop.util.GenericOptionsParser;
import org.apache.hadoop.util.StringUtils;
```

Source: http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Example:_WordCount_v2.0

Word Count So Easy!

```
public class WordCount2 {  
  
    public static class TokenizerMapper  
        extends Mapper<Object, Text, Text, IntWritable> {  
  
        static enum CountersEnum { INPUT_WORDS }  
  
        private final static IntWritable one = new IntWritable(1);  
        private Text word = new Text();  
  
        private boolean caseSensitive;  
        private Set<String> patternsToSkip = new HashSet<String>();  
  
        private Configuration conf;  
        private BufferedReader fis;  
  
        @Override  
        public void setup(Context context) throws IOException,  
            InterruptedException {  
            conf = context.getConfiguration();  
            caseSensitive = conf.getBoolean("wordcount.case.sensitive", true);  
            if (conf.getBoolean("wordcount.skip.patterns", true)) {  
                URI[] patternsURIs = Job.getInstance(conf).getCacheFiles();  
                for (URI patternsURI : patternsURIs) {  
                    Path patternsPath = new Path(patternsURI.getPath());  
                    String patternsFileName = patternsPath.getName().toString();  
                    parseSkipFile(patternsFileName);  
                }  
            }  
        }  
    }  
}
```

Word Count So Easy!

```
private void parseSkipFile(String fileName) {
    try {
        fis = new BufferedReader(new FileReader(fileName));
        String pattern = null;
        while ((pattern = fis.readLine()) != null) {
            patternsToSkip.add(pattern);
        }
    } catch (IOException ice) {
        System.err.println("Caught exception while parsing the cached file "
            + StringUtils.stringifyException(ice));
    }
}

@Override
public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException {
    String line = (caseSensitive) ?
        value.toString() : value.toString().toLowerCase();
    for (String pattern : patternsToSkip) {
        line = line.replaceAll(pattern, "");
    }
    StringTokenizer itr = new StringTokenizer(line);
    while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
        Counter counter = context.getCounter(CountersEnum.class.getName(),
            CountersEnum.INPUT_WORDS.toString());
        counter.increment(1);
    }
}
}
```

Word Count So Easy!

```
public static class TrivialReduce {
    private Trivial<Text, Trivialable, Text, Trivialable> {
        private Trivialable result = new Trivialable();
    }

    public void reduce(Text key, Iterable<Trivialable> values,
                      Context context)
        throws IOException, InterruptedException {
        int sum = 0;
        for (Trivialable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}

public static void main(String[] args) throws IOException {
    Configuration conf = new Configuration();
    DeclarativeProcessor<Text> optimizer = new DeclarativeProcessor<Text>(conf, args);
    String[] remainingArgs = optimizer.getRemainingArgs();
    if ((remainingArgs.length <= 2) || (remainingArgs.length >= 4)) {
        System.out.println("Usage: wordcount <in> <out> [<skip whitespace>]");
        System.exit(2);
    }
    Job job = Job.getJob(conf, "word count");
    job.setMapperClass(MapReduceJobMapper.class);
    job.setReducerClass(TrivialReduce.class);
    job.setPartitionerClass(TextPartitioner.class);
    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(Trivialable.class);

    TextInputFormat<Text> inputFormat = new TextInputFormat();
    for (int i=0; i < remainingArgs.length; ++i) {
        if (!"-skip".equals(remainingArgs[i])) {
            job.addInputFile(new Path(remainingArgs[i+1]), i+1);
            job.getConfiguration().setBoolean("mapreduce.map.skip.punctuation", true);
        } else {
            optimizer.add(remainingArgs[i+1]);
        }
    }
    FileOutputFormat.setOutputPath(job, new Path(remainingArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(remainingArgs[1]));
    System.exit(optimizer.waitForCompletion(true) ? 0 : 1);
}
```

Word Count So Easy!



Source: <http://knowledgespeakswisdomlistens.blogspot.com/2013/01/am-i-sheldon-cooper.html>

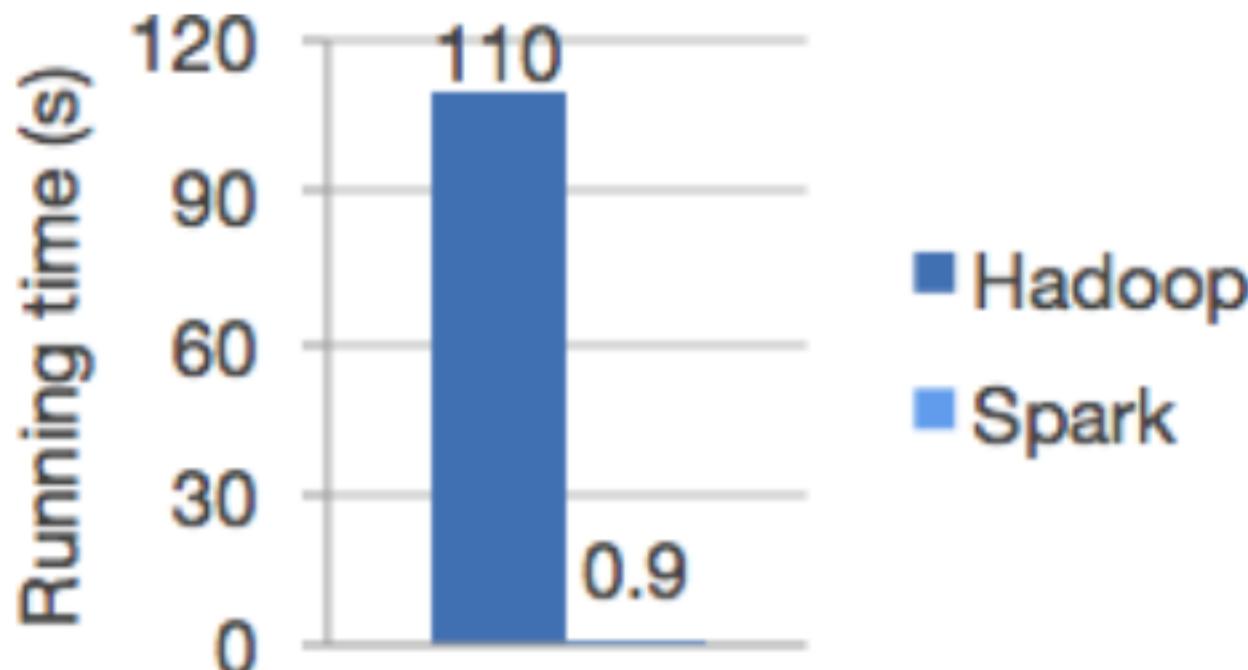
We Do Big Data



Source: <http://i.imgur.com/OF87Yiw.jpg>

Until Spark Came Along

```
val textFile= spark.textFile("hdfs://...")  
val counts= textFile.flatMap(line => line.split(" "))  
    .map(word => (word, 1))  
    .reduceByKey(_ + _)  
counts.saveAsTextFile("hdfs://...")
```



Source: http://www.disputcirce.nl/leden/img_2426/ <http://spark.apache.org/>

Spark First World Problem



Source: <https://i.imgur.com/r4xnp.jpg>

Submit Spark Application

Create application JAR

```
sbt vis/assembly
```

Upload JAR to cluster

```
scp vis/target/scala-2.10/vis-assembly-x.y.z-SNAPSHOT.jar my-awesome-spark-cluster:
```



Source: <https://xkcd.com/303/>

Submit Spark Application

The `spark-submit` script in Spark's bin directory is used to launch applications on a cluster.

```
# Run application locally on 8 cores
$SPARK_HOME/bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master local[8] \
/path/to/examples.jar \
100

# Run on a Spark Standalone cluster in client deploy mode
$SPARK_HOME/bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master spark://207.184.161.138:7077 \
--executor-memory 20G \
--total-executor-cores 100 \
/path/to/examples.jar \
1000
```

- Commands difficult to remember
- No easy way to store different pairs of arguments

Source: <http://spark.apache.org/docs/latest/submitting-applications.html>

Bash Scripting Rocks!

```
#!/bin/bash

PROJECT_VER=SNAPSHOT
PROJECT_ML_JAR=${PROJECT_DIR}/project-ml-assembly-${PROJECT_VER}.jar

PREFIX=kmeans
HBASE_KEYPREFIX=${PREFIX}_${SUFFIX}
HBASE_CLASSPATH=/etc/hbase/conf:/usr/lib/hbase/lib/hbase-client.jar:/usr/lib/hbase/lib/hbase-0

OUTPUT_HDFS_DIR=${PREFIX}/${SUFFIX}

time \
  spark-submit \
  --master yarn-cluster \
  --num-executors 1000 \
  --class com.awesome.ml.KMClustering \
  --driver-class-path ${HBASE_CLASSPATH} \
  --driver-memory=3g --executor-memory=3g \
  --conf="spark.executor.extraClassPath=${HBASE_CLASSPATH}" \
  --conf="spark.yarn.jar=hdfs:///proj/share/spark/spark-assembly-1.5.0-hadoop2.6.0.jar" \
  ${PROJECT_ML_JAR} \
  -output=${OUTPUT_HDFS_DIR} -input=my_awesome_big_dataset \
  -k=10 -partitions=10 -runs=10 \
  -hbase=${HBASE_KEYPREFIX}
```

Bash Scripting Rocks!



Source: <http://memegenerator.net/instance/37544999>

Bash Scripting Rocks?

- Environment isolation difficult to achieve (repeatability)
- Weak abstraction and encapsulation (code reuse)
- Runtime exception only (reliability)



Source: <http://animatedmeme.blogspot.com/2013/06/peter-griffin-vs-blinds.html>

Spark Development Challenges

- Multiple non-instantaneous steps
 - Build, Upload, Submit
- Multiple main classes
 - Exploratory Analysis, Model Training, Model Evaluation and etc
- Multiple projects
 - Core, Batch ETL, ML, Streaming, Visualization and etc
- Multiple configurations
 - Different Datasets, Parameters and etc
- Multiple deployment locations
 - local, yarn, EC2 and etc

Scala!



Source: <http://memegenerator.net/instance/58038541>

sbt-spark-submR**ugin to Rescue**

Demo: SparkPi Submission for Local and YARN

What is sbt

The interactive build tool

Use Scala to define your tasks. Then run them in parallel from the shell.

- *build.sbt* uses an elegant Scala DSL for build definition
- *project/*.scala* use full power of Scala to configure the build and create custom tasks

Source: <http://www.scala-sbt.org/>

Why sbt

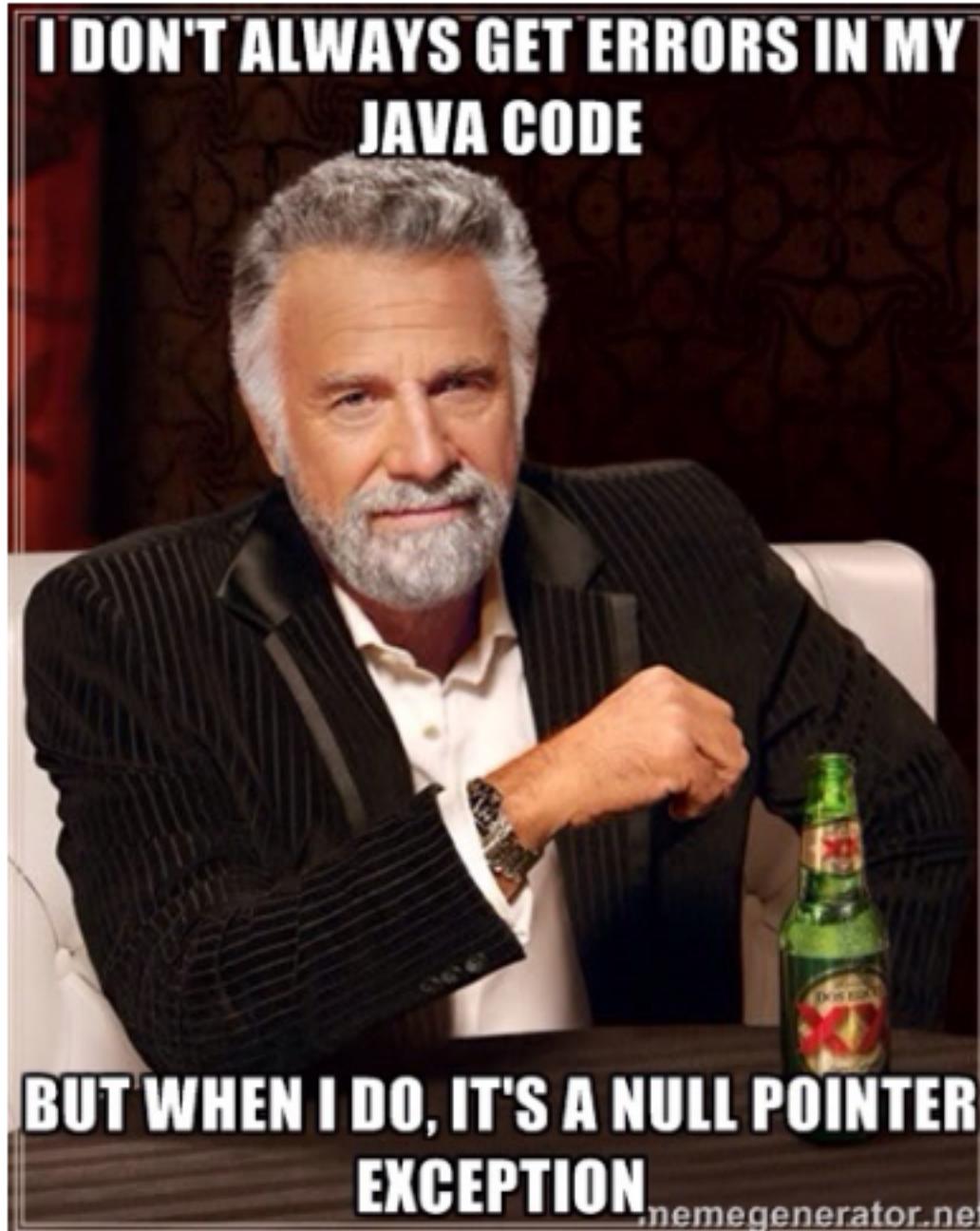
- Fast incremental build
- Continuous compilation/testing
- Concise and powerful configuration
- Built-in auto-complete
- IDE highlights compile time exception

Scala!



Source: <http://www.iamprogrammer.net/>

Scala!



Source: <http://imgur.com/r/programmingmemes/ju1CgAo>

Multi-project Build

Demo

sbt Multi-project Build

- Create tasks specific to each project
- Common settings/tasks are automatically deduplicated
- Tasks are executed in parallel

It's extensible!

Demo: Deploy on AWS EC2

sbt-spark-submit Recap

- Streamline build, upload and submit into single task
- Codify *spark-submit* command into settings
- Integrate well for multi-project
- Extensible

Thank you

Slides and code

<https://github.com/saurfang/nyc-spark-meetup>

sbt-spark-submit

<https://github.com/saurfang/sbt-spark-submit>

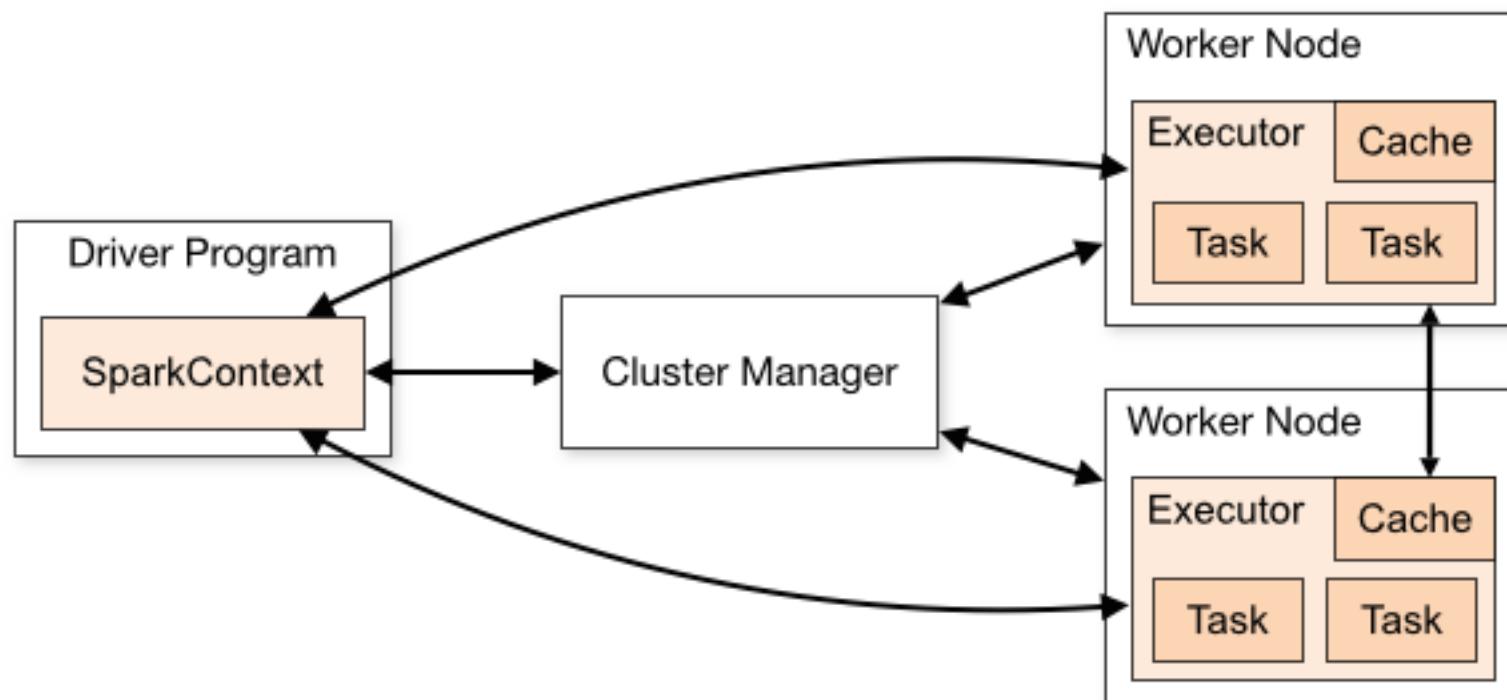


Source: <http://www.blankchapters.com/2012/12/22/big-data-is-not-always-the-solution-to-all-our-problems>

Appendix

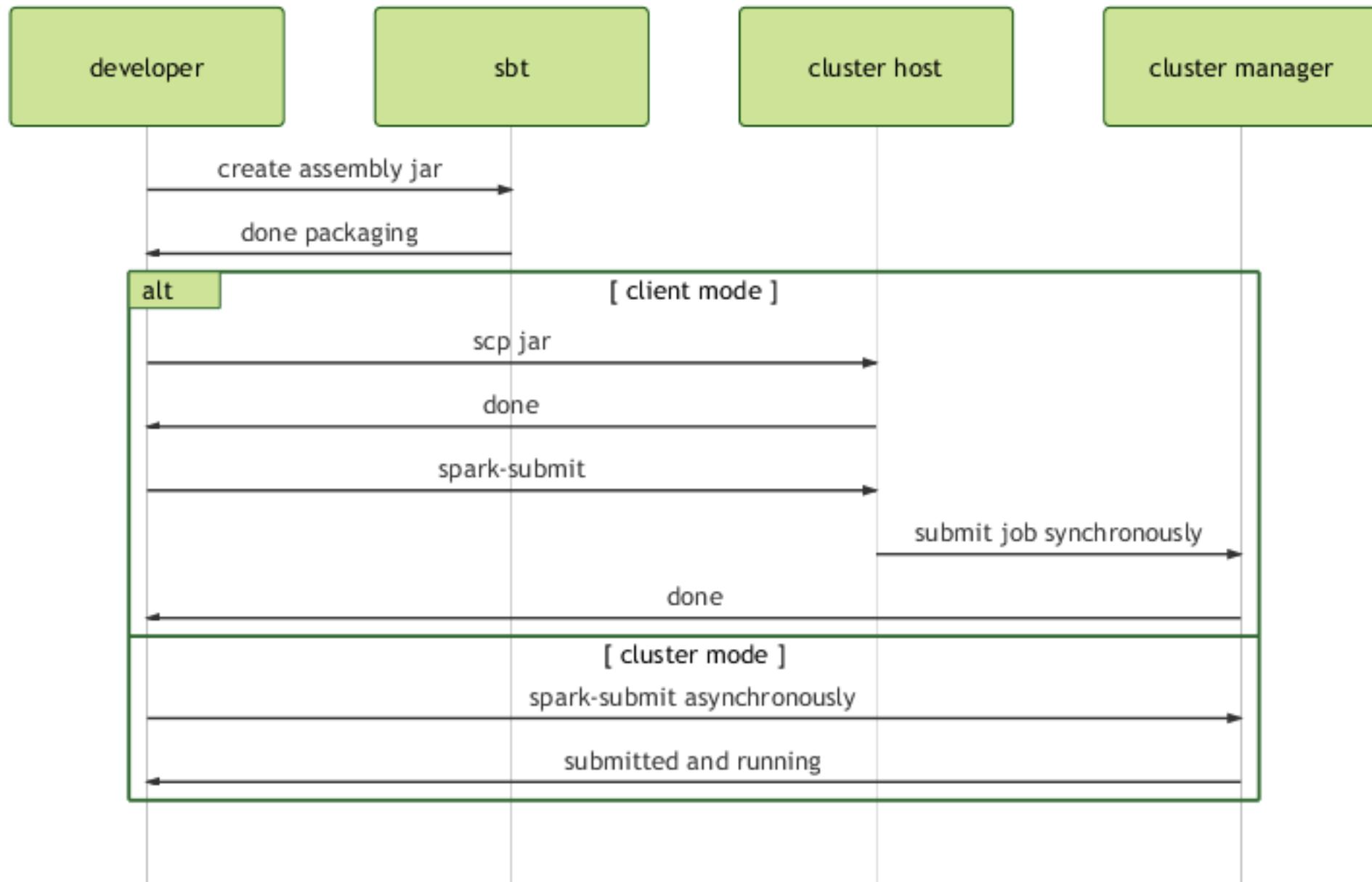
Spark Cluster Mode Overview

- Driver Program
- Executors (worker, slave, container)

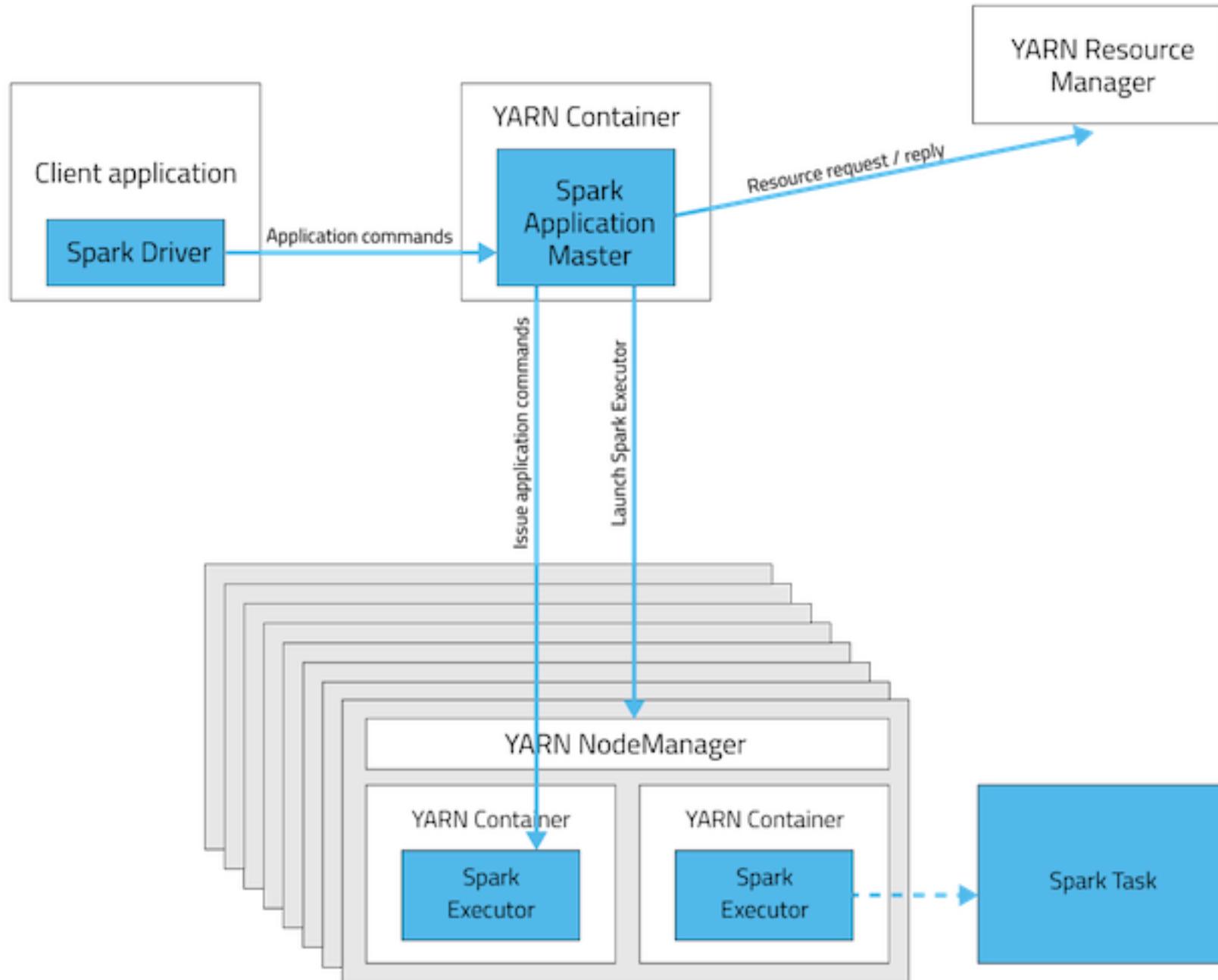


Source: <http://spark.apache.org/docs/latest/cluster-overview.html>

Submit Spark Application

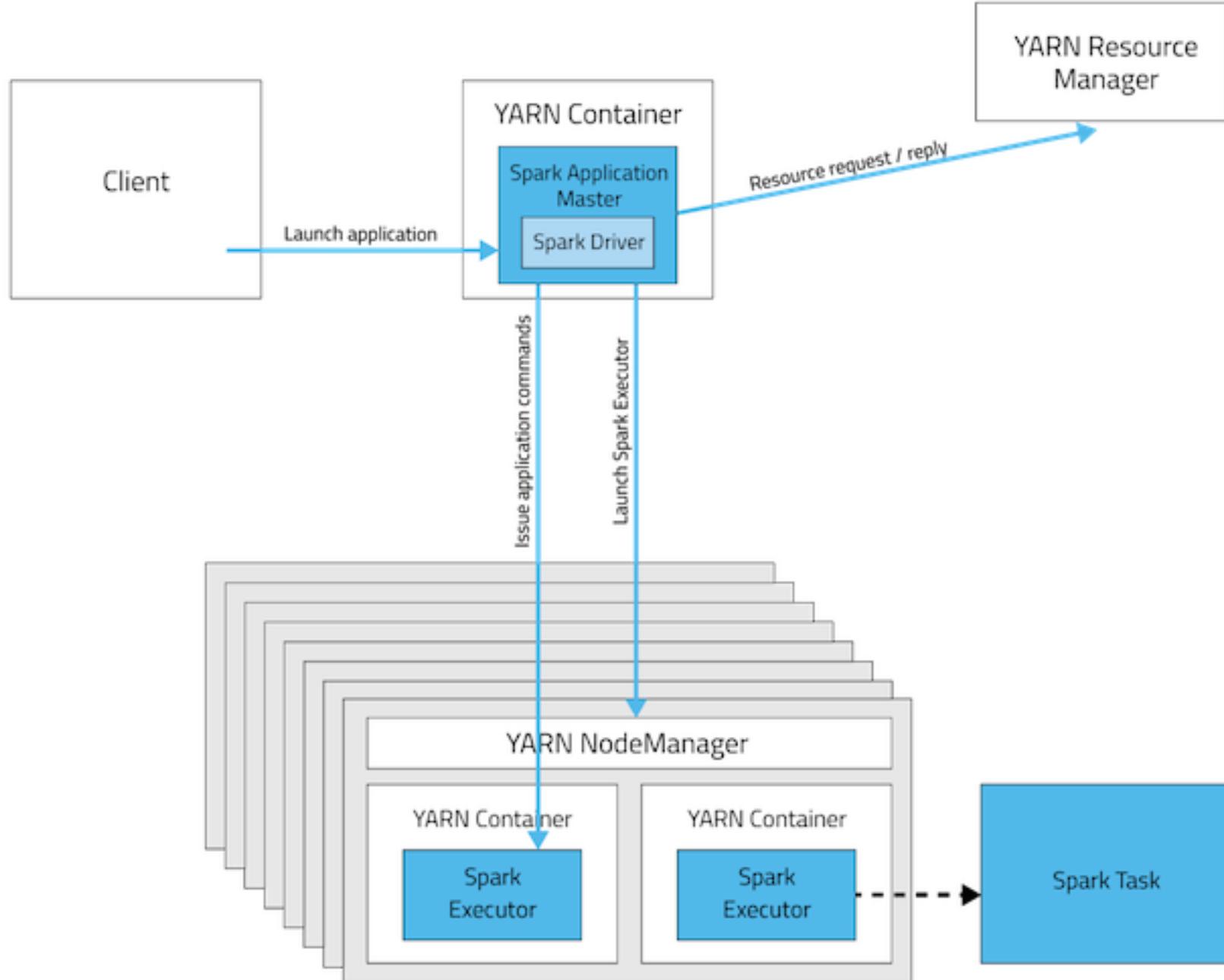


Client Deployment Mode



Source: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cdh_ig_running_spark_on_yarn.html

Cluster Deployment Mode



Source: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cdh_ig_running_spark_on_yarn.html