Writing Standalone Spark Programs

Matei Zaharia
UC Berkeley

www.spark-project.org
Outline

Setting up for Spark development

Example: PageRank

PageRank in Java

Testing and debugging
Building Spark

Requires: Java 6+, Scala 2.9.1+

git clone git://github.com/mesos/spark
cd spark
sbt/sbt compile

# Build Spark + dependencies into single JAR
# (gives core/target/spark*assembly*.jar)
sbt/sbt assembly

# Publish Spark to local Maven cache
sbt/sbt publish-local
Adding it to Your Project

Either include the Spark assembly JAR, or add a Maven dependency on:

groupId:  org.spark-project
artifactId: spark-core_2.9.1
version:  0.5.1-SNAPSHOT
Creating a SparkContext

```scala
import spark.SparkContext
import spark.SparkContext._

val sc = new SparkContext(
  "masterUrl", "name", "sparkHome", Seq("job.jar"))
```

Important to get some implicit conversions

- Mesos cluster URL, or local / local[N]
- Job name
- Spark install path on cluster
- List of JARs with your code (to ship)
import spark.SparkContext
import spark.SparkContext._

object WordCount {
    def main(args: Array[String]) {
        val sc = new SparkContext(
            “local”, “WordCount”, args(0), Seq(args(1)))
        val file = sc.textFile(args(2))
        file.map(_.split(" "))
            .flatMap(word => (word, 1))
            .reduceByKey(_ + _)
            .saveAsTextFile(args(3))
    }
}
Outline

Setting up for Spark development

Example: PageRank

PageRank in Java

Testing and debugging
Why PageRank?

Good example of a more complex algorithm
  » Multiple stages of map & reduce

Benefits from Spark’s in-memory caching
  » Multiple iterations over the same data
Basic Idea

Give pages ranks (scores) based on links to them

» Links from many pages $\rightarrow$ high rank
» Link from a high-rank page $\rightarrow$ high rank
Algorithm

1. Start each page at a rank of 1
2. On each iteration, have page $p$ contribute $\frac{\text{rank}_p}{|\text{neighbors}_p|}$ to its neighbors
3. Set each page’s rank to $0.15 + 0.85 \times \text{contribs}$
Algorithm

1. Start each page at a rank of 1
2. On each iteration, have page $p$ contribute $\frac{\text{rank}_p}{|\text{neighbors}_p|}$ to its neighbors
3. Set each page’s rank to $0.15 + 0.85 \times \text{contribs}$
Algorithm

1. Start each page at a rank of 1
2. On each iteration, have page \( p \) contribute \( \frac{\text{rank}_p}{|\text{neighbors}_p|} \) to its neighbors
3. Set each page’s rank to \( 0.15 + 0.85 \times \text{contribs} \)
Algorithm

1. Start each page at a rank of 1
2. On each iteration, have page $p$ contribute $\frac{\text{rank}_p}{|\text{neighbors}_p|}$ to its neighbors
3. Set each page’s rank to $0.15 + 0.85 \times \text{contribs}$
Algorithm

1. Start each page at a rank of 1
2. On each iteration, have page $p$ contribute $\frac{\text{rank}_p}{|\text{neighbors}_p|}$ to its neighbors
3. Set each page’s rank to $0.15 + 0.85 \times \text{contribs}$
Algorithm

1. Start each page at a rank of 1
2. On each iteration, have page \( p \) contribute \( \frac{\text{rank}_p}{|\text{neighbors}_p|} \) to its neighbors
3. Set each page’s rank to \( 0.15 + 0.85 \times \text{contribs} \)

Final state:
val links = // RDD of (url, neighbors) pairs
var ranks = // RDD of (url, rank) pairs

for (i <- 1 to ITERATIONS) {
    val contribs = links.join(ranks).flatMap {
        case (url, (links, rank)) =>
            links.map(dest => (dest, rank/links.size))
    }
    ranks = contribs.reduceByKey(_ + _)
        .mapValues(0.15 + 0.85 * _)
}

ranks.saveAsTextFile(...)

Coding It Up
PageRank Performance

![Bar chart showing PageRank performance comparison between Hadoop and Spark for different numbers of machines.

- **Number of machines**: 30 and 60.
- **Time per iteration (s)**: The comparison is shown in two bars for each number of machines.
  - For 30 machines:
    - Hadoop: 72s
    - Spark: 28s
  - For 60 machines:
    - Hadoop: 171s
    - Spark: 80s

- **Legend**:
  - Blue: Hadoop
  - Red: Spark
Other Iterative Algorithms

Logistic Regression

K-Means Clustering

Time per Iteration (s)

Hadoop  Spark
Outline

Setting up for Spark development

Example: PageRank

PageRank in Java

Testing and debugging
Differences in Java API

Implement functions by extending classes
  » spark.api.java.function.Function, Function2, etc

Use special Java RDDs in spark.api.java
  » Same methods as Scala RDDs, but take Java Functions

Special PairFunction and JavaPairRDD provide operations on key-value pairs
  » To maintain type safety as in Scala
import spark.api.java.*;
import spark.api.java.function.*;

JavaSparkContext sc = new JavaSparkContext(...);
JavaRDD<String> lines = ctx.textFile("hdfs://...");

JavaRDD<String> words = lines.flatMap(
    new FlatMapFunction<String, String>() {
        public Iterable<String> call(String s) {
            return Arrays.asList(s.split(" "));
        }
    });

System.out.println(words.count());
Examples

```java
import spark.api.java.*;
import spark.api.java.function.*;

JavaSparkContext sc = new JavaSparkContext(...);
JavaRDD<String> lines = ctx.textFile(args[1], 1);

class Split extends FlatMapFunction<String, String> {
    public Iterable<String> call(String s) {
        return Arrays.asList(s.split(" "));
    }
}

JavaRDD<String> words = lines.flatMap(new Split());

System.out.println(words.count());
```
Key-Value Pairs

```java
import scala.Tuple2;

JavaPairRDD<String, Integer> ones = words.map(
    new PairFunction<String, String, Integer>() {
        public Tuple2<String, Integer> call(String s) {
            return new Tuple2(s, 1);
        }
    }
);

JavaPairRDD<String, Integer> counts = ones.reduceByKey(
    new Function2<Integer, Integer, Integer>() {
        public Integer call(Integer i1, Integer i2) {
            return i1 + i2;
        }
    }
);
```
Java PageRank
Outline

Setting up for Spark development

Example: PageRank

PageRank in Java

Testing and debugging
Developing in Local Mode

Just pass `local` or `local[k]` as master URL

Still serializes tasks to catch marshaling errors

Debug in any Java/Scala debugger
Running on a Cluster

Set up Mesos as per Spark wiki
  » github.com/mesos/spark/wiki/Running-spark-on-mesos

Basically requires building Mesos and creating config files with locations of slaves

Pass master:port as URL (default port is 5050)
Running on EC2

Easiest way to launch a Spark cluster

git clone git://github.com/mesos/spark.git

cd spark/ec2

./spark-ec2 -k keypair -i id_rsa.pem -s slaves \
[launch|stop|start|destroy] clusterName

Details: tinyurl.com/spark-ec2
Viewing Logs

Click through the web UI at `master:8080`

Or, look at stdout and stderr files in the Mesos “work” directories for your program, such as:

/\tmp/mesos/slaves/<SlaveID>/frameworks/<FrameworkID>/executors/0/runs/0/stderr

FrameworkID is printed when Spark connects, SlaveID is printed when a task starts
Common Problems

Exceptions in tasks: will be reported at master

17:57:00 INFO TaskSetManager: Lost TID 1 (task 0.0:1)
17:57:00 INFO TaskSetManager: Loss was due to java.lang.ArithmeticException: / by zero
   at BadJob$$anonfun$1.apply$mcII$sp(BadJob.scala:10)
   at BadJob$$anonfun$1.apply(BadJob.scala:10)
   at ...

Fetch failure: couldn’t communicate with a node
   » Most likely, it crashed earlier
   » Always look at first problem in log
Common Problems

NotSerializableException:

» Set `sun.io.serialization.extendedDebugInfo=true` to get a detailed trace (in SPARK_JAVA_OPTS)

» Beware of closures using fields/methods of outer object (these will reference the whole object)
For More Help

Join the Spark Users mailing list:

groups.google.com/group/spark-users

Come to the Bay Area meetup:

www.meetup.com/spark-users