

Distributed Regression using Apache Spark

Neil Satra

History

MapReduce/Hadoop and extensions



General BSP – Pregel/Giraph



Spark



- ✓ Speed
- ✓ Ease of Use
- ✓ Generality
- ✓ Runs Everywhere

MapReduce WordCount in Spark

```
val textFile = sc.textFile("Input.txt")
val wordCounts =      textFile.flatMap(line => line.split(" "))
                        .map(word => (word, 1))
                        .reduceByKey((a, b) => a + b)
wordCounts.collect()
```

The Problem to Solve



How the application works

1. Write code in high-level scala, running transformations on the abstract collection (RDDs)
2. Package it into a jar file
3. Tell Spark the cluster to deploy the code on

Tasks

- Setup
 - Get Scala installed and running
 - Get Spark installed and running
 - Get a Mesos cluster running with Scala + Spark
- Implement regression
 - Manually
 - Using Mlib
 - Using Mahout
- Benchmark performance
 - On a single node
 - On clusters of various sizes
 - With each of the three implementations

Why?

Testing assumption that distributed = better