Introduction to MapReduce, Hadoop, Spark, Shark and Mahout

Christian Grant

cgrant@cise.ufl.edu
Outline

• First Period
  • What is map and reduce?
  • What is MapReduce?
  • What is Hadoop?
  • What is Spark?
  • What is Shark?
  • What is Mahout?
Outline

• First Period
  • What is map and reduce?
  • What is MapReduce?
  • What is Hadoop?

• What is Spark?
• What is Shark?
• What is Mahout?
What is **map** and **reduce**?

- Functional Languages and Functional Paradigms
- Haskell, Lisp, python, ...

- First-class/higher-order function → functions that take other functions as arguments or return them as results.

- Operators do not mutate collections
What is `map` and `reduce`?

- **map**
  
  ```
  map (+3) [1,2,3,4,5] => [4,5,6,7,8]
  ```

- **reduce**
  
  ```
  reduce' (+) [1,2,3,4,5] => 15
  ```

- **any**
  
  ```
  any (even) [1..10] => True
  ```

- **all**
  
  ```
  all (even) [1..10] => False
  ```

- **filter**
  
  ```
  filter (even) [1..10] => [2,4,6,8,10]
  ```

- **fold**
  
  ```
  fold' (*) (1) [1,2,3,4,5] => 120
  ```
What is **map** and **reduce**?

- These functional operators allow the chaining of several operations.

- For example in unix:
  ```
  cat input.txt |
  tr -c -s '[:alpha:]' '[:\n*:]' |
  sort |
  uniq -c |
  sort -n -r -k 1,1 |
  sed 20q
  ```

- This approach scales pretty well!
Outline

• First Period
  • What is map and reduce?
  • What is MapReduce?
  • What is Hadoop?

• What is Spark?
• What is Shark?
• What is Mahout?
What is MapReduce?

- Google engineers saw these benefits and developed the **MapReduce** ecosystem.
- This includes
  1) MapReduce programming model
  2) Google File System
  3) Job scheduling system
What is **MapReduce**? » Programming model

- Inspired by functions in Haskell, Lisp, python etc.

  \[
  \text{map} \ (k1, \ v1) \rightarrow [(k2, v2)] \\
  \text{reduce} \ (k2, \ [v2]) \rightarrow [v2]
  \]
What is **MapReduce**? » Programming model

```java
// Word Count Example
map(String key, String value):
    // key: document name
    // value: document contents
    for each word w in value:
        EmitIntermediate(w, "1");

reduce(String key, Iterator values):
    // key: a word
    // values: a list of counts
    int result = 0;
    for each v in values:
        result += ParseInt(v);
    Emit(AsString(result));
```
What is **MapReduce**? » Google File System

- A Large, distributed, highly fault-tolerant file system.
- A master process maintains metadata, data is stored in chunks.
What is **MapReduce**? » Google File System
What is MapReduce? » Google File System

• What is a Chunk?
  Analogous to a block in filesystems.
  64 MB size
  64-bit chunk handle (~ filename)
  Chunks are replicated across servers
What is **MapReduce**? » Google File System

• **What is a Master?**

  A single process on a machine.
  Stores all chunk metadata.
  Communicates instructions and status messages with chunk servers.

  * More info in paper
What is **MapReduce**? » Execution Model
Outline

• First Period
  • What is map and reduce?
  • What is MapReduce?
  • What is Hadoop?

• What is Spark?
• What is Shark?
• What is Mahout?
What is Hadoop?

• Open Source Apache project
• Championed by Yahoo!
• Written in Java
• Clone the MapReduce ecosystem from Google
What is **Hadoop**? » Word Count Mapper

```java
public static class MapClass extends MapReduceBase
    implements Mapper<LongWritable, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value,
                    OutputCollector<Text, IntWritable> output,
                    Reporter reporter) throws IOException {
        String line = value.toString();
        StringTokenizer itr = new StringTokenizer(line);
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            output.collect(word, one);
        }
    }
}
```
What is **Hadoop**? » Word Count Reducer

```java
public static class Reduce extends MapReduceBase
    implements Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values,
            OutputCollector<Text, IntWritable> output,
            Reporter reporter) throws IOException {
        int sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }
}
```
What is Hadoop? » Word Count Main

```java
public class WordCount {
    ......
    public static void main(String[] args) throws IOException {
        JobConf conf = new JobConf(WordCount.class);
        conf.setJobName("wordcount");

        // the keys are words (strings)
        conf.setOutputKeyClass(Text.class);
        // the values are counts (ints)
        conf.setOutputValueClass(IntWritable.class);

        conf.setMapperClass(MapClass.class);
        conf.setReducerClass(Reduce.class);
        conf.setInputPath(new Path(args[0]));
        conf.setOutputPath(new Path(args[1]));
        JobClient.runJob(conf);
    }
```
What is **Hadoop**? » **Ecosystem**

- **HDFS**: A clone of google file system. Storage Layer.

- **HBase**: BigTable clone. A scalable, distributed database that supports structured data storage for large tables.

- **Hive**: A data warehouse infrastructure that provides data summarization and ad hoc querying. SQL-like interface (HiveQL).

- **Mahout**: A scalable machine learning library.

- **Pig**: A high-level data-flow language and execution framework for parallel computation.
What is Hadoop? » Pros and Cons

- Hadoop is immature but under heavy development.
- Becoming the main tool large data processing
- Batch oriented framework
- Transactions vs. Eventual Consistency
- others?
Outline

• First Period
  • What is map and reduce?
  • What is MapReduce?
  • What is Hadoop?

• What is Spark?
  • What is Shark?
  • What is Mahout?
What is **Spark**?

- An open source cluster computing system that aims to make data analytics fast to write and execute.
- Excels when memory helps: Iterative algorithms (data mining + machine learning)
- Scala and Java APIs.
- Developed in UC Berkeley AMP lab.
What is **Spark**?  »  Setup

Requires: Java 6+, Scala 2.9.1+

```bash
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

./spark-shell
```
What is **Spark?** » Scala

Declaring variables:

```scala
var x: Int = 7
var x = 7  // type inferred
val y = "hi"  // read-only
```

Functions:

```scala
def square(x: Int): Int = x*x
def square(x: Int): Int = {
  x*x
}
def announce(text: String) {
  println(text)
}
```

Java equivalent:

```java
int x = 7;
final String y = "hi";
```

Java equivalent:

```java
int square(int x) {
  return x*x;
}
void announce(String text) {
  System.out.println(text);
}
```
What is **Spark?** » Scala (2)

**Generic Types:**

```scala
define
var arr = new Array[Int](8)
define
var lst = List(1, 2, 3)
```

**Java equivalent:**

```java
define
int[] arr = new int[8];
List<Integer> lst =
new ArrayList<Integer>();
lst.add(...)
```

**Indexing:**

```scala
define
arr(5) = 7
define
println(lst(1))
```

**Java equivalent:**

```java
define
arr[5] = 7;
define
System.out.println(lst.get(1));
```
What is **Spark?** » Scala (3)

Processing collections with functional programming:

```scala
val list = List(1, 2, 3)
list.foreach(x => println(x)) // prints 1, 2, 3
list.foreach(print(x))        // same

list.map(x => x + 2)          // => List(3, 4, 5)
list.map(_ + 2)               // same with place holder notation

list.filter(x => x % 2 == 1)  // => List (1, 3)
list.filter(_ % 2 == 1)       // => List(1, 3)

list.reduce((x,y) => x + y)  // => 6
list.reduce(_ + _)           // =6
```
What is Spark? » Scala (4) : Closure Syntax

(x: Int) => x + 2  // full version

x => x + 2  // type inferred

_ + 2  // when each argument is used exactly once

x => {
  // when body is a block of code
  val numberToAdd = 2
  x + numberToAdd
}

// If closure is too long, can always pass a function

def addTwo(x: Int): Int = x + 2

list.map(addTwo)
Scala collections provide many other functional methods; for example, Google for “Scala Seq”

<table>
<thead>
<tr>
<th>Method on Seq[T]</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>map(f: T =&gt; U): Seq[U]</td>
<td>Pass each element through f</td>
</tr>
<tr>
<td>flatMap(f: T =&gt; Seq[U]): Seq[U]</td>
<td>One-to-many map</td>
</tr>
<tr>
<td>filter(f: T =&gt; Boolean): Seq[T]</td>
<td>Keep elements passing f</td>
</tr>
<tr>
<td>exists(f: T =&gt; Boolean): Boolean</td>
<td>True if one element passes</td>
</tr>
<tr>
<td>forall(f: T =&gt; Boolean): Boolean</td>
<td>True if all elements pass</td>
</tr>
<tr>
<td>reduce(f: (T, T) =&gt; T): T</td>
<td>Merge elements using f</td>
</tr>
<tr>
<td>groupBy(f: T =&gt; K): Map[K, List[T]]</td>
<td>Group elements by f(element)</td>
</tr>
<tr>
<td>sortBy(f: T =&gt; K): Seq[T]</td>
<td>Sort elements by f(element)</td>
</tr>
</tbody>
</table>

...
What is **Spark?** » Word Count

```scala
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))
  }
}
```
What is **Spark?** » Word Count (2)

```scala
val lines = sc.textFile("hamlet.txt")
val counts = lines.flatMap(line => line.split(" "))
  .map(word => (word, 1))
  .reduceByKey(_ + _)
```

```
"to be or"
  "to"
  "be"
  "or"

"not to be"
  "not"
  "to"
  "be"
```

```
(to, 1)
(be, 1)
(or, 1)

(not, 1)
(to, 1)
(be, 1)
```

```
(be, 2)
(not, 1)
(or, 1)
(to, 2)
```
What is **Spark**? » Primitives

- Resilient distributed datasets (RDDs)
  - Immutable, partitioned collections of objects
- Lazy Transformations
  - Lazy operations to build RDDs from other RDDs.
- Actions (e.g. count, collect, save)
  - Return a result of write to storage
What is **Spark**? » Iterative Hadoop
What is **Spark**? » Iterative Spark
What is Spark? » Spark Cache
What is **Spark**? » Logistic Regression

![Graph showing running time vs. number of iterations for Hadoop and Spark.]

- **Hadoop:**
  - First iteration: 174 s
  - Further iterations: 6 s

- **Spark:**
  - 127 s / iteration
Outline

• First Period
  • What is map and reduce?
  • What is MapReduce?
  • What is Hadoop?

• What is Spark?
• What is Shark?
• What is Mahout?
What is Shark?

- Hive on Spark
- Runs Standard HiveQL including UDFs
- Exploit temporal locality: working set of data can often fit in memory to be reused between queries
- Provides low latency for small queries
What is **Shark**? » Benchmark Query

```
SELECT * FROM grep WHERE field LIKE '%%XYZ%%';
```
What is **Shark**? » Benchmark Query (2)

```sql
SELECT sourceIP, AVG(pageRank), SUM(adRevenue) AS earnings
FROM rankings AS R, userVisits AS V
ON R.pageURL = V.destURL
WHERE V.visitDate BETWEEN '1999-01-01' AND '2000-01-01'
GROUP BY V.sourceIP
ORDER BY earnings DESC
LIMIT 1;
```
Outline

- First Period
  - What is *map* and *reduce*?
  - What is *MapReduce*?
  - What is *Hadoop*?

- What is *Spark*?
- What is *Shark*?
- What is *Mahout*?
What is Mahout?

• A machine learning library to run algorithms over hadoop.

• Open source and open to contributions
What is **Mahout**? » Includes

- Collaborative Filtering
- User and Item based recommendations
- K-Means, Fuzzy K-Means clustering
- Mean Shift clustering
- Dirichlet process clustering
- Latent Dirichlet Allocation
- Singular Value decomposition
- Parallel Frequent Pattern mining
- Complementary Naive aByes classifier
- Random forest decision tree classifier
- ... Others
What is **Mahout**? » K-Means overview
What is **Mahout**? » K-Means overview
What is Mahout? » K-Means code

```java
public static void main(String[] args) {
    List<Vector> sampleData = new ArrayList<Vector>();
    generateSamples(sampleData, 400, 1, 1, 3);
    generateSamples(sampleData, 300, 1, 0, 0.5);
    generateSamples(sampleData, 300, 0, 2, 0.1);

    int k = 3;

    List<Vector> randomPoints = RandomPointsUtil.chooseRandomPoints(
                           sampleData, k);
    List<Cluster> clusters = new ArrayList<Cluster>();
    int clusterId = 0;
    for (Vector v : randomPoints) {
        clusters.add(new Cluster(v, clusterId++,
                                  new EuclideanDistanceMeasure()));
    }

    List<List<Cluster>> finalClusters =
        KMeansClusterer.clusterPoints(sampleData, clusters,
                                        new EuclideanDistanceMeasure(), 3, 0.01);
    for(Cluster cluster : finalClusters.get(finalClusters.size() - 1)) {
        System.out.println("Cluster id: " + cluster.getId() + " center: " +
                           cluster.getCenter().asFormatString());
    }
```
What is Mahout? » Means execution code

$ bin/mahout kmeans -i reuters-vectors/tfidf-vectors/ \
-c reuters-initial-clusters \
-o reuters-kmeans-clusters \
-dm org.apache.mahout.common.distance.SquaredEuclideanDistanceMeasure \
-cd 1.0 -k 20 -x 20 -cl
Further Reading

- MapReduce: http://research.google.com/archive/mapreduce.html
- Apache Hadoop: http://hadoop.apache.org/
- Spark: http://spark-project.org/
- Shark: http://shark.cs.berkeley.edu/

• MapReduce is Good Enough? If All You Have is a Hammer, Throw Away Everything That's Not a Nail! [http://arxiv.org/abs/1209.2191](http://arxiv.org/abs/1209.2191)

• Big Data: Hadoop, Business Analytics and Beyond [http://wikibon.org/wiki/v/Big_Data:_Hadoop,_Business_Analytics_and_Beyond](http://wikibon.org/wiki/v/Big_Data:_Hadoop,_Business_Analytics_and_Beyond)

• Mahout In Action [http://manning.com/owen/](http://manning.com/owen/)
• Questions?