Core Hadoop Syllabus

Preamble

The MapReduce Algorithm is used in Big Data to scale computations. Running in parallel the map reduce algorithms load a manageable chunk of data into RAM, perform some intermediate calculations, load the next chunk and keep going until all of the data has been processed. In its simplest representation it can be broken down into a Map step that often takes data set we can think of as 'unstructured' the a Reduce step that outputs a 'structured' data set often smaller.
In its simplest sense Hadoop is an implementation of the MapReduce Algorithm.
It's a convenient shorthand when we use the term Hadoop. There is the Hadoop project at a high level, then there is a core selection of tools the Hadoop refers to such as the Hadoop Distributed File System(HDFS), the HDFS shell and the HDFS protocol 'hdfs://'. Then there is a bigger stack of tools that are becoming central to the use of Hadoop often referred to as the 'Hadoop Ecosystem'. These tools consist of but are not limited to Hbase, Pig, Hive, Crunch, Mahout and Avro. Then there is the new Hadoop 2.2.x version that implements a new architecture for MapReduce and allows for efficient workflows using a 'DAG' of jobs, a significant evolution of the classic MapReduce job.
Finally Hadoop is written in Java. In Hadoop we see Java's significant contribution to the evolution of the distributed space as it is represented by Hadoop 2.2 and the Hadoop Ecosystem.

Prerequisites

1. A familiarity of programming in Java.
2. A familiarity of Linux.
3 Have access to a Amazon EMR account.
4. Have Oracle Virtualbox or VMware installed and functioning.

Textbooks

The focus is for online learning, for each subsection of the course a number of links are provided in the 'TopicLearningLinks.pdf' document to online learning resources that directly relate to the core concepts of that topic.

What Will I Learn?

In this course you will learn key concepts in Hadoop and learn how to write your own Hadoop Jobs and MapReduce programs.

The course will specifically facilitate the following High Level outcomes

1. Become literate in Big Data terminology and Hadoop.
2. Given a big data scenario, understand the role of Hadoop in overcoming the challenges posed by the scenario.
3. How Hadoop functions both in data storage and processing Big Data.
4. Understand the difference between MapReduce version 1 in Hadoop version 1.x.x and MapReduce version 2 in Hadoop version 2.2.x.
5. Understand the Distributed File Systems architecture and any implementation such as Hadoop Distributed File System or Google File System.
6. Analyze and Implement a Mapreduce workflow and how to design java classes for ETL(extract transform and load) and UDF (user defined functions) for this workflow.
7. Data Mining and filtering

The course will specifically facilitate the following Practical outcomes

1. Use the HDFS shell
2. Use the Cloudera, Hortonworks and Apache Bigtop virtual machines for Hadoop code development and testing.
3. Configure, execute and monitor a Hadoop Job.
4. Use Hadoop data types, readers, writers and splitters.
5. Write ETL and UDF classes for hadoop workflows with PIG and Hive
6. Write filters for Data mining and processing with Mahout , Crunch and Arvo.
7. Test Hadoop code on HortonWorks Sandbox.
8. Run Hadoop code on Amazon EMR.

1 Introduction to Big Data

a) Why Hadoop, Big Data and Map Reduce.
b) Architecture of Clusters.
c) Virtual Machine (VM), Provisioning a VM with vagrant and puppet

2. Hadoop Architecture

a) Set up a single Node Hadoop pseudo cluster VM with vagrant and puppet
b) Clusters and Nodes, Hadoop Cluster
c) NameNode, Secondary Name Node, Data Nodes
d) Running Multi node clusters on Amazons EMR

3. Distributed file systems

a) HDFS vs GFS a comparison.
b) Run hadoop on Cloudera, Web Administration
c) Run hadoop on Hortonworks Sandbox
d) File system operations with the HDFS shell
e) Advanced hadoop development with Apache Bigtop

4. Mapreduce Version 1

a) MapReduce Concepts in detail.
b) Jobs definition, Job configuration, submission, execution and monitoring.
c) Hadoop Data Types, Paths, FileSystem, Splitters, Readers and Writers.
d) The ETL class, Definition, Extract, Transform, and Load
e) The UDF class, Definition, User Defined Functions

5. Mapreduce with Hive (Data warehousing)

a) Schema design for a Data warehouse
b) Hive Configuration
c) Hive Query Patterns
d) Example Hive ETL class

6. Mapreduce with Pig (Parallel processing)

a) Introduction to Apache Pig
b) Pig LoadFunc and EvalFunc classes
c) Example Pig ETL class

7. The Hadoop Ecosystem

a) Introduction to Crunch
b) Introduction to Arvo
c) Introduction to Mahout

8. Mapreduce Version 2

a) Apache Hadoop 2 and YARN
b) Yarn Examples

9. Putting it all together

a) Amazon EMR example
b) Apache Bigtop example