CSE 487/587Data-intensive Computing Fall 2010 Due: 11/6/10

PROJECT 2: DATA-STRUCTURES AND ALGORITHMS ENABLING DATA-INTENSIVE COMPUTING

Purpose:

1. To understand the data-structures and algorithms for **data-intensive computing**
2. To design and implement the solutions for a data-intensive problem using MapReduce and Hadoop DFS
3. Understand **virtualization** and deploy a single data-node **HDFS** using **VMware player** or any other virtual machine technology such as Oracle's VirtualBox or Microsoft's Virtual PC.
4. Implement **MapReduce** algorithm for a sample application using single node HDFS as described above and also on the workflow provided
5. To explore designing and implementing data-intensive computing solutions on a **cloud environment**: in this case on **Amazon Compute Cloud** (EC2)[1]. (In project 1 we explored Google App Engine (GAE)[2]).

Problem Statement:

Design and implement a data-intensive application using Hadoop-MapReduce framework using local infrastructure and the cloud (infrastructure as a service) offered by amazon.com's aws [3, 7].

Project 1 focused on content accumulation methods and the three tier web application (or regular application) infrastructure to enable data-intensive applications. This project will focus on parallel processing beyond multi-threading: management and parallel processing of ultra-scale data. MapReduce is a framework for processing large scale data sets. It exploits the write-once-read-many (WORM) characteristic typical of unstructured data to represent and process (i) as key-value pairs <key,value> and (ii) using a suite of operations such as map, partition, reduce, shuffle etc. A special data repository (like a file system used in a traditional operating system) is needed to efficiently and reliably store and deliver the large data set. Google file system (GFS)[4] is such a system, and Hadoop Distributed File System (HDFS)[5] is another GFS-like system available under Apache open source license.

Preparation before lab:

1. Review the foundations of MapReduce and Hadoop Distributed File System.
2. Read Chapter 4 from your Algorithms for the intelligent web: study the data in the table 4.1 and the clustering of this data by K-means clustering in section 4.1, and 4.4.
3. You will have to get an account on aws; If you cannot (for many reasons) we can set up an instance for you using a image preferred by you and you can work with that instance.
4. Read the text Chapter 4 (the data set and K-means clustering) and the references provided.

Assignment:

(i) You will work on a MapReduce application on a single node Hadoop Distributed File System on your laptop or desktop with a simple word count application. (ii) Then you will extend this to aws (amazon web services) Elastic Compute Cloud implementation of MapReduce, (iii) apply your knowledge of MapReduce to implement a solution for K-means clustering on your single node local HDFS/MapReduce environment, perfect the MapReduce

solution, and (iv) port it to EC2 environment. The data for the parts (i) and (ii) will come from your project 1 collection (saved) and parts (iii) and (iv) will be from Chapter 4, Table 4.1. Synthesize a corpus based on this data format. We are interested in clustering the data around age.

System Architecture

You will implement the MapReduce wordcount using the content you collected in Project 1. You may have to convert the relational collection you had in MYSQL to an unstructured content. This part of the project will be implemented on a local ***development environment***.

The system architecture for the assignment given is shown in Figure 1. In Figure 1, the development environment is shown on the left, that is a single-node (or multi-node) HDFS deployed in your local environment. It can be on your laptop; if you do not have a laptop or compute environment you can work on the "Greek Godesses" multi-node HDFS[1] available in the lab at Franzak 206 lab. You will implement the MR-wordcount using the data collected in project 1 and MR-K-means using the synthesized data similar to one given in Chapter 4 of your text. Debug the solutions.

For the second part of the assignment you will work with the same problem as above but in the ***production environment*** shown on the right. You will transfer the data for the two applications into 2 buckets of the amazon simple storage service (s3) and upload the map and reduce functions designed in the part1. You will then create a MR workflow

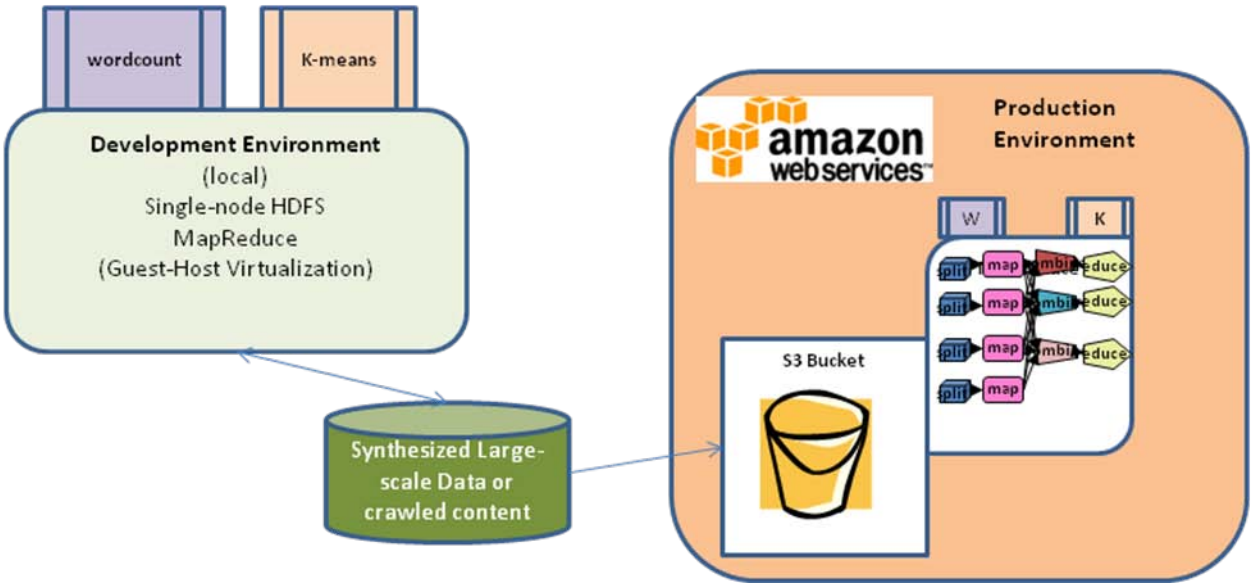

Figure 1: System Architecture for Data-intensive HDFS-MapReduce Application Development

[1] This infrastructure is partially supported by NSF-DUE-CCLI-0737243, 0920335

Project Implementation Details:
Most critical aspect of this project is converting K-means clustering into a MR parallel algorithm. In particular, what will be Map, what will be the Reduce function for K-means and what are input and output for the respective functions, and how to implement K-means using these functions? We will discuss this during lecture. See [6] which is intermittently available and I have provided a hardcopy for your review.

Project Deliverables:

1. A tar file of the single node implementation of the MR-wordcount including the data; a tar file of the single node implementation of the MR-K-means;
2. Demo of the implementation on the amazon cloud.
3. An experience report providing all the details of the project design, implementation and the performance evaluation report. This should have the user's manual, programmer's manual and any design diagrams.

Submission Details:
submit_cse487 files separated by space
submit_cse587 files separated by space

References:

1. Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/, last viewed October 11, 2010.
2. Google App Engine (GAE). http://code.google.com/appengine/, last viewed October 11, 2010.
3. MapReduce on Amazon. Advanced MapReduce Features. http://developer.yahoo.com/hadoop/tutorial/module5.html, last viewed October 11,2010.
4. Dean, J. and Ghemawat, S. 2008. **MapReduce: simplified data processing on large clusters.** *Communication of ACM* 51, 1 (Jan. 2008), 107-113.
5. Hadoop Distributed File System (HDFS). Apache Hadoop: http://hadoop.apache.org, last viewed September, 2010.
6. Linoff, G and Berry, M.Data miner's Blog. MapReduce and K-Means Clustering. http://blog.data-miners.com/2008/02/mapreduce-and-k-means-clsutering.html, last viewed September 2010.
7. White, T. Running Hadoop MapReduce on Amazon EC2 and Amazon S3. Amazon Web Services Developer Connection, July 2007.