Climate Data Analysis with Amazon Web Services

Nora Huang*, Maria Ferman†
Department of Computer Science, University of Victoria
Victoria, BC, Canada
Email: *norahuangsun@gmail.com, †tania.ferman@gmail.com

***Abstract*—Nowadays, it is extremely important to work with big datasets, and work with that large amount of data it is not always straightforward. For example, in the climate research it is more and more common to work with thousands of climate stations that describe the climate from a particular location. With this analysis, climate scientists can have an accurate understanding of the climate change. Specifically, scientists need to have the average of the data according to a range of time and location, in order to discover anomalies in the climate, thus they can find monthly anomalies base on periods of time. Besides, this calculation allows to analyze if the current year was colder or warmer, or if the temperature is higher or lower than the previous years. Therefore, scientists need to deal with the issue of analyze huge datasets. For this reason new technologies such as the MapReduce paradigm allows us to have large-scale data analysis. However, the MapReduce paradigm is not easy to code, thus systems like Amazon Web Services is using Hadoop framework, Hive, Pig and Hue for helping users. AWS allow to have an easier and faster analysis. In this project we present an evaluation of Amazon Web Services by using a large dataset of a climate department.**

I. Related work

This section we will provide background on the distributed system techniques and tools that will be used in the project.

A. MapReduce

Nowadays, we need processes that support a large amount of data, and in order to do that, we need a group of computers working together to process that large amount of data. However, there are several issues we need to solve in order to accomplish these processes. For example, how to distribute the data, how to deal with some parts of the process fail, such as some nodes stop working.

Besides, there are also a specify data set processing problems. This data set includes raw data such as crawled documents and web request logs, derived data such inverted indices, various representations of the graph structure of web documents, summaries of the number of pages crawled per host, the set of most frequent queries in a given day.

MapReduce is a functional model with user-specified map and reduce operations. It provides simple and powerful interface that enables automatic parallelization and distribution of large-scale computations, combined with an implementation of this interface that achieves high performance on large clusters of commodity PCs.

Specifically, the MapReduce solution is divided in two sections written by the user: Map and Reduce. On the Map section, it will be taken an input pair with which will be producing a key/value pairs. In this section, the MapReduce process will classify the values with their corresponded keys. Then, these classified pairs will be passed to the second section, the Reduce function.

In the Reduce section, it will receive the previously classified pairs, then, in this section it will be merged together the key with their corresponded values, in order to create a smaller set of values. In other words, the reduce phase collect intermediate results in order to have an assemble final result. [2]

B. Hadoop Distributed File System

Another solution for solving Big Data problems is Hadoop Distributed File System[5]. In which the system will create replicas of the Data and divide them between different nodes. In order to do this, Hadoop uses the MapReduce paradigm. The data can be distributed among hundreds or thousands of nodes, and use parallel computations for having a reliable and scalable system. Specifically, the block of data will be carried out in the MapReduce system, in order to accomplish the big data processing requirements.

The process used by Hadoop starts by breaking the Data into small sections, consecutively all these small blocks of data will be distributed among the clusters. The reason for doing that division is for allowing the MapReduce system to work in parallel, thus the process can solve the scalability issues in systems that require Big Data.

Due to in all Distributed systems the presence of failures is inevitable, the fault tolerance and fault compensation capabilities are well understood by the Hadoop system. The solution found by Hadoop Distributed File system (HDFS) consist in having multiple sections (blocks) of the data and send them to different servers in the Hadoop cluster. Therefore HDFS uses these divisions of workloads into different servers in order to have the benefit of Data locality, a critical property when systems requires large Datasets.

For the sake of having a system that always maintains availability in the presence of failures, HDSF replicates small blocks of data into different servers.

C. Pig

Pig [3] was developed by Yahoo to allow users to focus more on the analysis of the data, rather than worry to write MapReduce implementations. By using Pig in Hadoop users can load any kind of data format. In order to compile the programs Pig uses its own language named Pig Latin.

Pig will use as an input pig latin program, and it will divide the program into different sections that will be used as MapReduce jobs. Then, this job sections will be executed and coordinated in the Hadoop MapReduce environment. Therefore, all the pig programs will be loaded into the MapReduce job cluster.

By using the Hadoop engine, Pig can have a remarkable fault-tolerance and scalability properties. The first step is loading the data, Pig will use a pig latin program as an input, then a set of transformations will be carried out by ping in order to translate the data in sections of MapReduce tasks. Then, when all the process ends, the Pig results can be shown on the screen or stored in a file.

D. Hive

As a result of some of the issues of the MapReduce model such as the low level and the requirement of building custom program, Facebook developed a new component that allows to users to leverage the Hadoop platform. Users with a background in SQL can easily use the Hive platform developed on top of Hadoop, because the queries are similar to SQL statements [6].

Specifically, SQL queries will be broken into blocks using the Hive Service for being used in the MapReduce jobs and being executed in the Hadoop cluster. The language that Hive provides is named HiveQL which is base on the SQL language, and uses statements such as select, project, joint among others.

One of the main components of Hive is the Hive Thrift Server that is used to execute the HiveQL statements. The Hive Thrift Server is a simple client API used for allowing the communication with the Hive server.

E. Hue

Hue [4] stands for Hadoop User Experience, and it is an open source graphical interface that allows users to have an easier HDFS ecosystem experience by using a web basic application. Hue is a functional and intuitive tool that allows users to execute Hive jobs among other services on the web. Besides, Hue provides to the user with an automatic visualization at the end of the process, and users can change that graph by using simple operations.

Hue [1] is an application that allows users to upload, download and delete files in HDFS. After a file is uploaded, Hue will automatically extract the file to use. Therefore, Hue allows users to make a faster analysis of the data and HDFS file management in the web interface.

II. Project

A. Dataset

The dataset gathered for this project comes from the Pacific Climate Impacts Consortium (PCIC) of the University of Victoria. The dataset consist in a series of climate measurements that shows how the climate changes and varies over time. All the measurements were gathered at the same location in the Pacific and Yukon region of British Colombia, Canada. The dataset is divided by time and stations. The station is the exact location where the data were gathered. The dataset will allow us to understand the climate change among time and regions (stations). The resulting analysis of the dataset might help for trend analysis and climate change researches. This information can be used as an input for more complex climate models in order to have accurate climate predictions.

B. Scripts

To add ...

C. Implementation

To add ... In this section we present our implementation of the project.

Pig To add ...

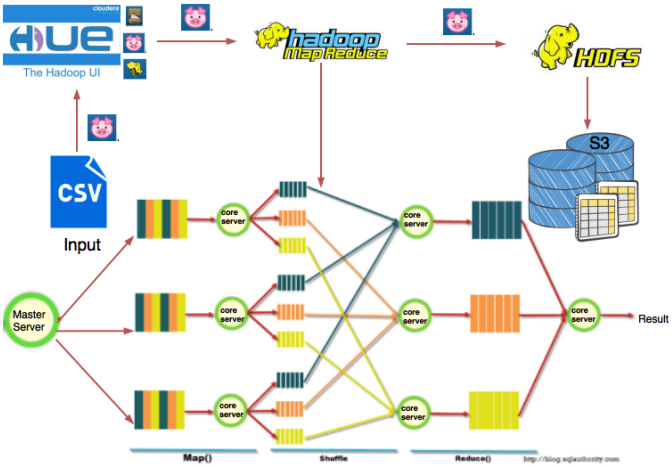

Fig. 1. Pig Diagram

Hive To add ...

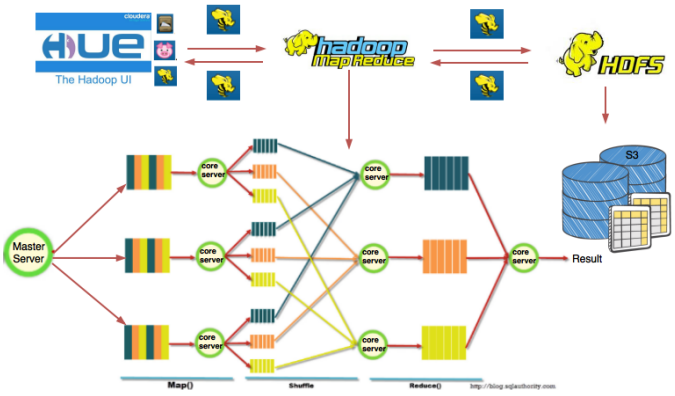

Fig. 2. Hive Diagram

III. Results and Analysis

To add ... The dataset allows us to select the period of time that we want to work. Specifically, we used from 2014 to 2015, because we believe that this range of time is representative for the dataset. The climate variables that we use are minimum and maximum temperature of a specific day.

IV. Conclusions

To add ...

V. Future Work

To add ...

References

[1] A. B. *Professional hadoop*. US: Wrox Press Ltd, 2016.

[2] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[3] A. F. Gates, O. Natkovich, S. Chopra, P. Kamath, S. M. Narayanamurthy, C. Olston, B. Reed, S. Srinivasan, and U. Srivastava. Building a high-level dataflow system on top of map-reduce: the pig experience. *Proceedings of the VLDB Endowment*, 2(2):1414–1425, 2009.

[4] A. Rasheed and M. Mohideen. Fedora commons with apache hadoop: A research study. 2013.

[5] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*. IEEE, 2010.

[6] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy. Hive: a warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment*, 2(2):1626–1629, 2009.