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. Introduction

Nowadays the size of the dataset that people is using is increasing considerably. Besides, users need a swift analysis of the data. Therefore, finding new technologies that allow users to analyze big data in a fast way is extremely important. Now users need a platform that is more agile, and with which they can manipulate, distribute and storing the data, and at the same time be able to build efficient scalable applications

Amazon Web Services is a cloud service platform that allows users to have a high compute power, a large durable dataset storage, high performance databases, among other functionalities. AWS provides to the users a large amount of services that they can use together for building the applications that they need.

Therefore, users can manage their data in an accessible way. Users can integrate, import and export the data, manage Hadoop and built secure environments for the analysis of large datasets.

II. 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.

F. Amazon EMR

Amazon EMR is a web service that makes it easy to quickly and cost-effectively process vast amounts of data.Amazon EMR simplifies big data processing, providing a managed Hadoop framework that makes it easy, fast, and cost-effective for you to distribute and process vast amounts of your data across dynamically scalable Amazon EC2 instances. You can also run other popular distributed frameworks such as Apache Spark and Presto in Amazon EMR, and interact with data in other AWS data stores such as Amazon S3 and Amazon DynamoDB.Amazon EMR securely and reliably handles your big data use cases, including log analysis, web indexing, data warehousing, machine learning, financial analysis, scientific simulation, and bioinformatics.

G. Amazon S3

Amazon Simple Storage Service (Amazon S3), provides developers and IT teams with secure, durable, highly-scalable cloud storage. Amazon S3 is easy to use object storage, with a simple web service interface to store and retrieve any amount of data from anywhere on the web. With Amazon S3, you pay only for the storage you actually use. There is no minimum fee and no setup cost. Amazon S3 offers a range of storage classes designed for different use cases including Amazon S3 Standard for general-purpose storage of frequently accessed data, Amazon S3 Standard - Infrequent Access (Standard - IA) for long-lived, but less frequently accessed data, and Amazon Glacier for long-term archive. Amazon S3 also offers configurable lifecycle policies for managing your data throughout its lifecycle. Once a policy is set, your data will automatically migrate to the most appropriate storage class without any changes to your applications. Amazon S3 can be used alone or together with other AWS services such as Amazon Elastic Compute Cloud (Amazon EC2) and AWS Identity and Access Management (IAM), as well as data migration services and gateways for initial or ongoing data ingestion. Amazon S3 provides cost-effective object storage for a wide variety of use cases including backup and recovery, nearline archive, big data analytics, disaster recovery, cloud applications, and content distribution.

III. 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

The project implementation begins by using two customised scripts for the Hive and Pig process, the first script is for getting the data and the second is for storing and doing some computations.

Pig script: To add ...

Hive script: This script uses sql command for selecting the data form the Hadoop Distributed File System. The script only select the require information for doing the posterior computations for analyzing of the data.

```
SELECT ds562.time,
    ds562.one_day_precipitation,
    ds562.max_temp, ds562.min_temp
FROM ds562
```

C. Implementation

We setup a EMR cluster on Amazon Web Service. The cluster is consist of 3 nodes, 1 mater and 2 core(workersw). All nodes are running Lastest Ubuntu as its operating system while installed Hadoop, Hue, Hive and Pig by default. Hue is stalled on master node while the other 3 are achitectures involved both master and core nodes. So we do not need to do extra configuration or application installation for our data analysis. The raw climate data are uploaded onto S3. And all output and logs are also stored in S3. The archtecure of the system are shown in fig1 and fig 2. fig1 shows the data process path of Pig while fig2 shows the data process path of Hive. We use Pig to load raw data from S3, and then caculate the average values of each columns and stored them in HDFS format. After the processed data are in HDFS, we can use Hive which provides SQL executor to access the data. Hue is a web page that providing interface for Pig and Hive.

IV. Evaluation

A. Qualitatively

Easy to Use: You can launch an Amazon EMR cluster in minutes. You dont need to worry about node provisioning, cluster setup, Hadoop configuration, or cluster tuning. Amazon EMR takes care of these tasks so you can focus on analysis.

Predictable cost: Amazon EMR pricing is simple and predictable: You pay an hourly rate for every instance hour you use. You can launch a 10-node Hadoop cluster for as little as $0.15 per hour. Because Amazon EMR has native support for Amazon EC2 Spot and Reserved Instances, you can also save 50-80% on the cost of the underlying instances.

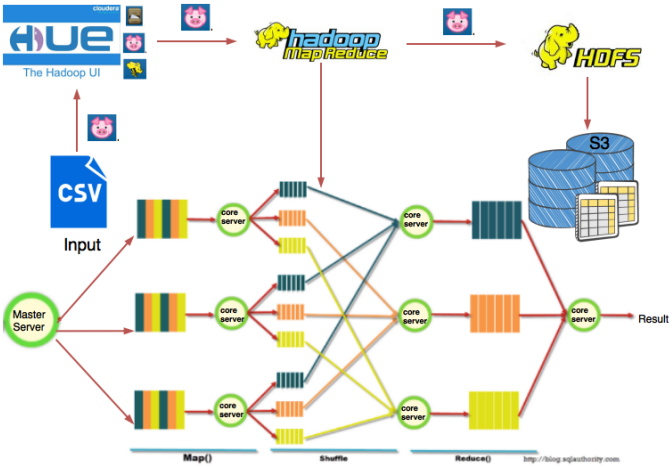

Fig. 1: Pig Diagram

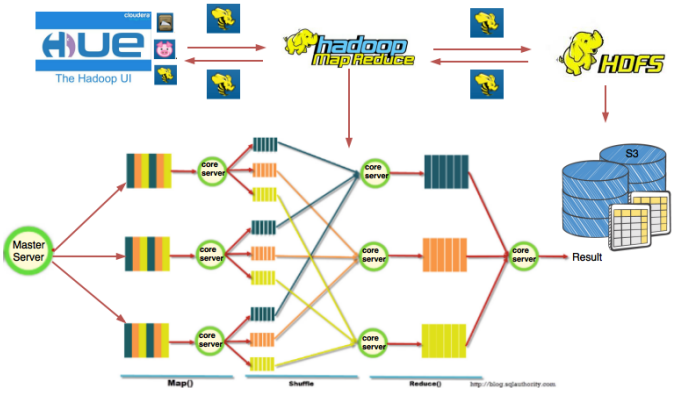

Fig. 2: Hive Diagram

Elastic: With Amazon EMR, you can provision one, hundreds, or thousands of compute instances to process data at any scale. You can easily increase or decrease the number of instances and you only pay for what you use.

Reliable: You can spend less time tuning and monitoring your cluster. Amazon EMR has tuned Hadoop for the cloud; it also monitors your cluster retrying failed tasks and automatically replacing poorly performing instances.

Flexible: You have complete control over your cluster. You have root access to every instance, you can easily install additional applications, and you can customize every cluster. Amazon EMR also supports multiple Hadoop distributions and applications.

You can enjoy lots of features and benefits when you are using AWS EMR, however, graphical interface provided by Hue is limit, there are only several simple charts that it can display base on your data. And it is not user configurable. If you want a new chart type, you may need to change the source code of Hue. The good thing is that Hue is an open source project which you can access it source code and contribute on it easily.

TABLE I: S3 Storage Pricing

/month	Standard Storage per GB	Standard-Infrequent Access Storage per GB	Glacier Storage per GB
First 1 TB	$0.0300	$0.0125	$0.007
Next 49 TB	$0.0295	$0.0125	$0.007
Next 450 TB	$0.0290	$0.0125	$0.007
Next 500 TB	$0.0285	$0.0125	$0.007
Next 4000 TB	$0.0280	$0.0125	$0.007
Over 5000 TB	$0.0275	$0.0125	$0.007

TABLE II: Pricing for Amazon EMR and Amazon EC2 General Purpose - Current Generation

	Amazon EC2 per Hour	Amazon Elastic MapReduce per Hour
m3.xlarge	$0.266	$0.070
m3.2xlarge	$0.532	$0.140
m4.large	$0.12	$0.030
m4.xlarge	$0.239	$0.060
m4.2xlarge	$0.479	$0.120
m4.4xlarge	$0.958	$0.240
m4.10xlarge	$2.394	$0.270

B. Quantitatively

This section provides some pricing statistic and performance measurement on AWS EMR.

Pricing: In our project the data is no more than 1GB. And we use 3 m3.xlarge nodes in our cluster each of which cost $0.336 per hour. The detail of pricing can be found in Table I and Table II

performance: Since Pig is the most time consuming process in our data analysis, we only evaluate processing time of Pig. Hive take almost no time when access the processed data by Pig. We configure 1 m3.xlarge as master and 2 m3.xlarge as core. And execute the Pig Script on 1 year, 3 years, 5 years and 10 years data. Table III demonstrate our measurement.

V. Conclusions

To add ...

VI. 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.

TABLE III: Hive Processing Time

Data Volume year(s)	Processing Time ms
1	37536
3	41908
5	42312
10	42054