Big Data Programming Project 3: Twitter User Analysis

Krish Moodbidri
University of Alabama at Birmingham
krish94@uab.edu

www.krishm.com

ABSTRACT

This is a project based report for the course Big Data programming (FA2017 CS 716-7R / 616-7R / 416-7R) taught by Dr. Jeremy Blackburn at University of Alabama, Birmingham.

The tasks were to analyze and summarize the patterns of the social network of people on Twitter, a popular social media platform. The dataset contained a text file of size 25 GB.

The summary of the tools used, the data analysis performed, the code and executing the jobs, is mentioned in this paper. Finally the analysis to the tasks given are performed and the results summarized.

1 INTRODUCTION

Twitter is an online news and social networking service where users post and interact with messages, called "tweets." These messages were originally restricted to 140 characters, but on November 7, 2017, the limit was doubled to 280 characters for all languages except Japanese, Korean and Chinese.[11] Registered users can post tweets, but those who are unregistered can only read them. Users access Twitter through its website interface, Short Message Service (SMS) or mobile device application software ("app"). Twitter, Inc. is based in San Francisco, California, United States, and has more than 25 offices around the world.[13]

Twitter was created in March 2006 by Jack Dorsey, Noah Glass, Biz Stone, and Evan Williams and launched in July of that year. The service rapidly gained worldwide popularity. In 2012, more than 100 million users posted 340 million tweets a day, and the service handled an average of 1.6 billion search queries per day. In 2013, it was one of the ten most-visited websites and has been described as "the SMS of the Internet". As of 2016, Twitter had more than 319 million monthly active users. On the day of the 2016 U.S. presidential election, Twitter proved to be the largest source of breaking news, with 40 million election-related tweets sent by 10 p.m. (Eastern Time) that day. *(ref. Wikipedia)*

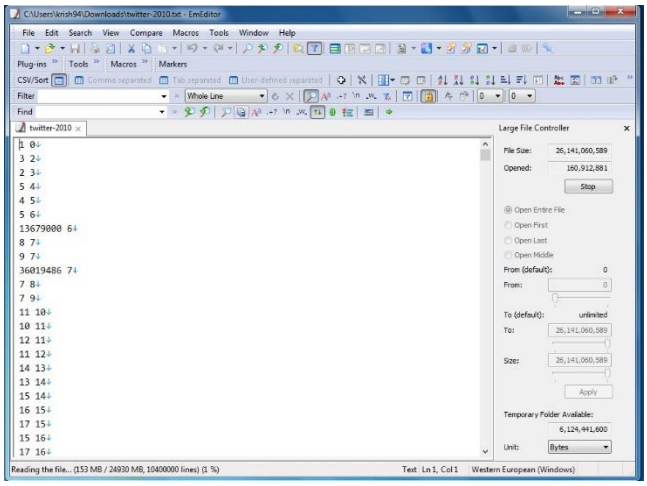
About the Data:

I planned on using the dataset used in Gary Warner's research on Twitter ISIS network infiltration, however I had to write for authorization emails and that would have taken quite some time. Thus I searched online and found a dataset that very closely matched my requirements.

https://snap.stanford.edu/data/twitter-2010.html - Directed follower network

The data provided was a text file having 2 fields associated with the user – userID /t FollowerID.

The dataset was too large to open in a default editor, thus I had to download EmEditor- EmEditor is a lightweight extensible commercial text editor for Microsoft Windows. It was developed by Yutaka Emura of Emurasoft, Inc. It includes full Unicode support, 32-bit and 64-bit builds, syntax highlighting, find and replace with regular expressions, vertical selection editing, editing of large files (up to 248 GB or 2.1 billion lines), and is extensible via plugins and scripts.



Dataset snippet

Sample of the data:

- userID – the Twitter ID of the user
- folowerID - a Twitter ID of the person following the user.

2 DESIGN AND ARCHITECTURE

2.1 Platform Introduction

Apache Hadoop is an open-source software framework used for distributed storage and processing of dataset of big data using the MapReduce programming model. It consists of computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.

The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality,[3] where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking. *(ref. Wikipedia)*

2.2 Platform Architecture

Hadoop employs a master/slave architecture for both distributed storage and distributed computation".In the distributed storage, the NameNode is the master and the DataNodes are the slaves. In the distributed computation, the Jobtracker is the master and the Tasktrackers are the slaves which are explained in the following sections.

MapReduce Job Processing

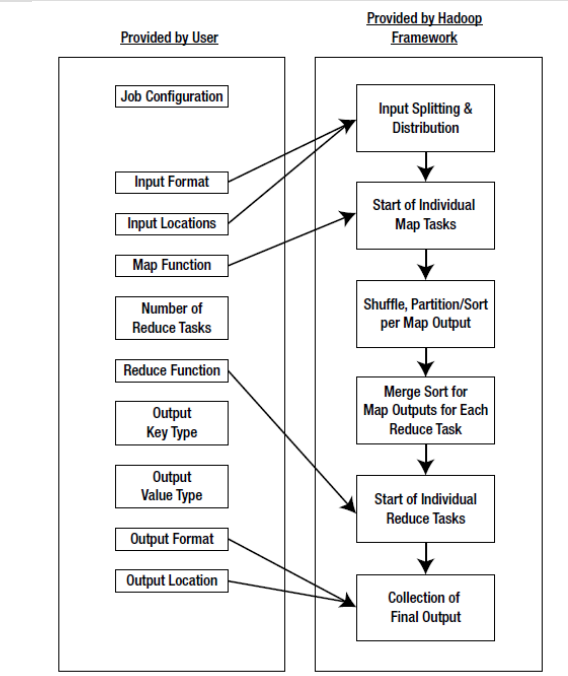
An entire Hadoop execution of a client request is called a job. Users can submit job requests to the Hadoop framework, and the framework processes the jobs. Before the framework can process a job, the user must specify the following:

- The location of the input and output files in the distributed file system
- The input and output formats
- The classes containing the map and reduce functions

Hadoop has four entities involved in the processing of a job:[1]

- The user, who submits the job and specifies the configuration.
- Hadoop architecture The JobTracker, a program which coordinates and manages the jobs. It accepts job submissions from users, provides job monitoring and control, and manages the distribution of tasks in a job to the TaskTracker nodes.[2] Usually there is one JobTracker per cluster.
- The TaskTrackers manage the tasks in the process, such as the map task, the reduce task, etc. There can be one or more TaskTracker processes per node in a cluster.
- The distributed file system, such as HDFS.

The user specifies the job configuration by setting different parameters specific to the job. The user also specifies the number of reducer tasks and the reduce function. The user also has to specify the format of the input, and the locations of the input. The Hadoop framework uses this information to split of the input into several pieces. Each input piece is fed into a user-defined map function. The map tasks process the input data and emit intermediate data. The output of the map phase is sorted and a default or custom partitioning may be applied on the intermediate data. Accordingly, the reduce function processes the data in each partition and merges the intermediate values or performs a user-specified function. The user is expected to specify the types of the output key and the output value of the map and reduce functions. The output of the reduce function is collected to the output files on the disk by the Hadoop framework. *(ref.* https://hadooptutorial.wikispaces.com/Hadoop+architecture *)*



Parts of a MapReduce job

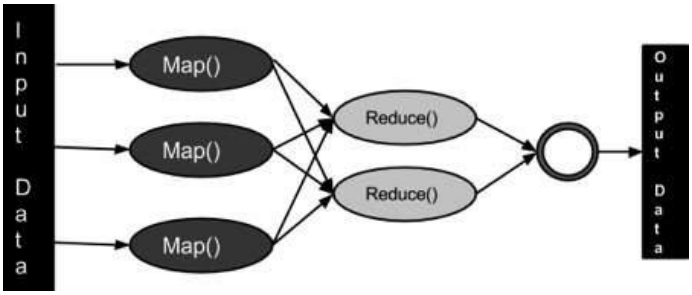
Map-Reduce:

- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

Map stage : The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

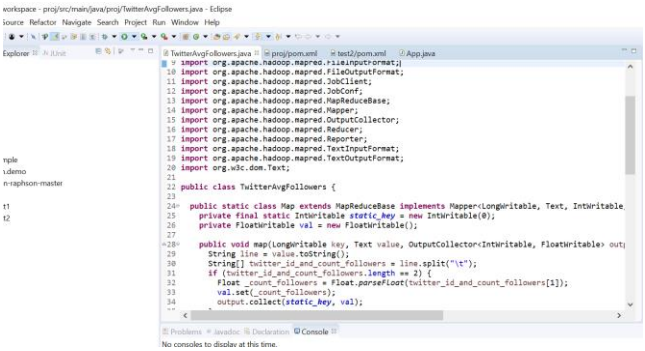
Reduce stage : This stage is the combination of the Shufflestage and the Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.



Writing code for this problem:

I coded in java, using Eclipse Oxygen as my IDE. I used the Maven project and used maven to convert the code to a jar file and execute the executable jar on the cluster.

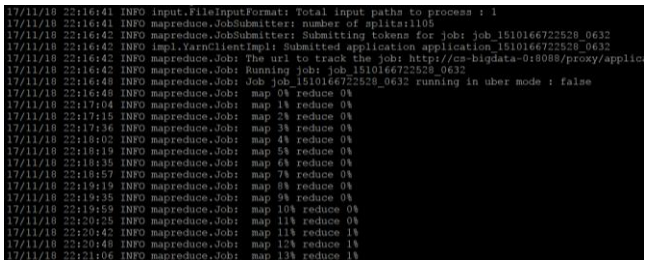


The above is a snippet of the code, IDE and platform.

Executing the Code:

I executed my code on the cs-bigdata-0.cs.uab.edu

This cluster was set up by Dr. Blackburn and Greg Bowersock for the course Big data Programming



Snippet of Program executing

3 EXECUTION STRATEGY

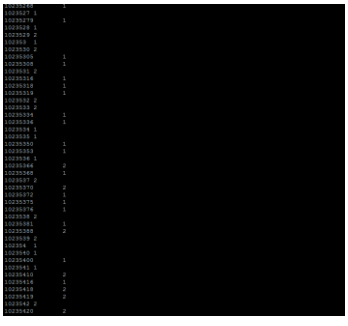
I started by planning the relations possible to determine with the dataset. Here are the tasks that I'm executing on the data

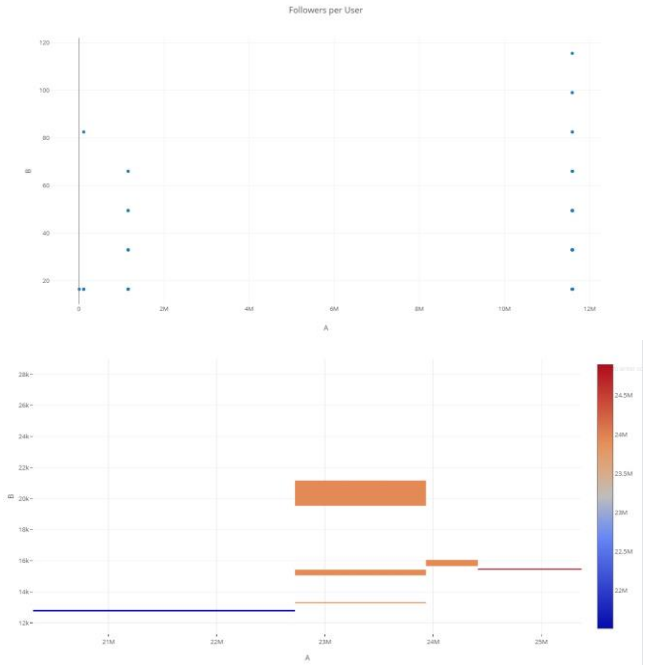
1. Follower Counter – I'm finding the number of followers per user
2. Average Followers – I'm finding the average number of followers per user of the dataset (2010)
3. Followers of people (eg. celebrities), grouped by ranges.
4. People with highest following – most influential people.

Initially I was unable to parse the entire text file since it was requiring more virtual memory than available. Thus I split the file into 25 parts and parsed it. I did seem to get fair results. Its difficult to analyse the results since the dataset is that of 2010, and a lot of followers have been removed, or deactivated or banned.

Task 1: Calculate follower counter

I did so by running a map-reduce on the data with the values being user and follower. Every time a follower is found for an existing user, counter is increased by 1. This way I get the total number of followers per person. The results are plotted on a graph using plot.ly





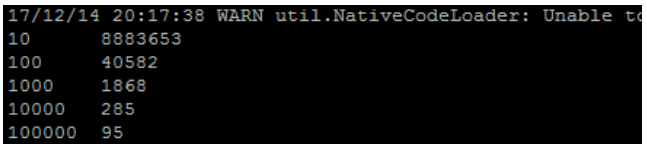
Task 2: Calculate average followers

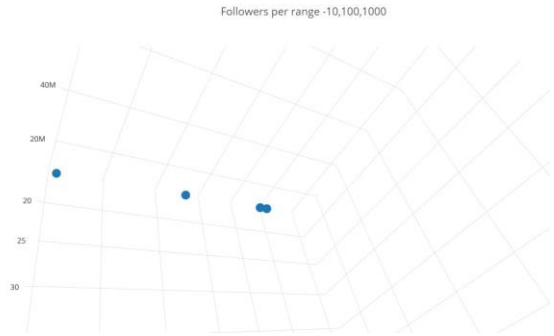
I found the average users between people wrt the follower ID.

The average no. of followers per user is estimated to be approx. 11.

Task 3: Followers of people grouped by ranges

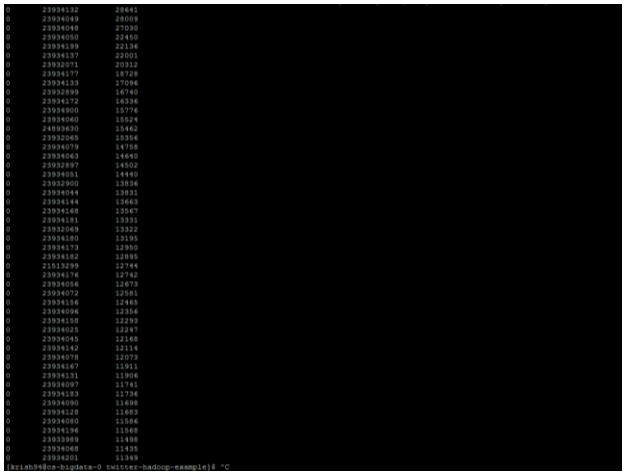
I did so by running a map-reduce on the data with the values being user and follower. Every time a follower is found for an existing user, counter is increased by 1. Only after the count hits beyond a range, it is written to the file. This way I get the total number of followers per person grouped by the range.

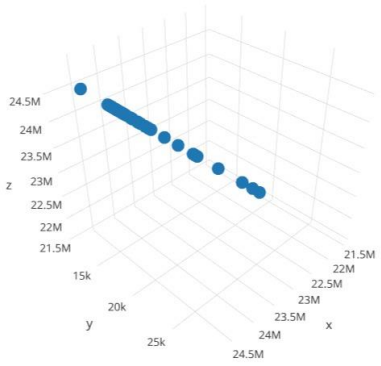


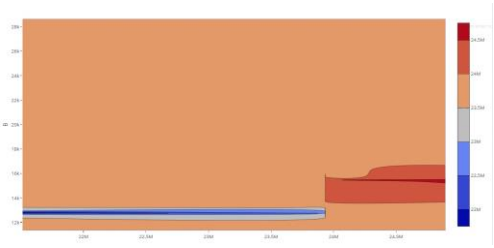


Task 4: Calculate highest follower users

I did so by running a map-reduce on the data with the values being user and follower. For the users having largest follower base, I ran a sort- Bubble sort algorithm on it.







5 CONCLUSIONS

I learnt a lot during the project. The data set seemed really interesting and the data association as well. I would specially thank 3 people – Dr. Blackburn, Amalee and Mashuir for their help and support. It took me quite some time to figure out the system and dataset. Though I did my best on the project, I would like to work on the error cases for each of the questions to determine a more accurate analysis. Overall, it was quite a fun and educational project.

REFERENCES

[1] http://jsoneditoronline.org/
[2] https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm
[3] https://hadoop.apache.org/docs/r2.8.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
[4] https://developer.twitter.com/en/docs
[5] https://developer.twitter.com/
[6] https://chandramanitiwary.wordpress.com/category/mapreduce/