Analysis of USGS Earthquake Data

NANDITA SATHE[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: nsathe@iu.edu*

Project: S17-IO-3017, May 8, 2017

US Geological Survey's (USGS) Earthquake Hazards Program monitor and report earthquakes, assess earthquake impacts and hazards, and research the causes and effects of earthquakes [1]. The geo-spatial data it collects is available for free. Big Data Analytics tools are used to analyze this data. Machine learning algorithms are used for advanced data analysis and earthquakes prediction.

Keywords: I524, geospatial, MongoDB, Plotly, K-means clustering, DBSCAN, Python, pymongo, USGS, Ansible, Cherrypy

Code: https://github.com/cloudmesh/sp17-i524/tree/master/project/S17-IO-3017/code
Report: https://github.com/cloudmesh/sp17-i524/tree/master/project/S17-IO-3017/report/report.pdf

1. INTRODUCTION

USGS collects volumes of geospatial data pertaining to earthquakes and makes it available for analysis. The project obtains a chunk of the data using GeoJSON web service provided by USGS. The data is saved locally in MongoDB database. Data is analysed using machine learning algorithms. The output is plotted (rendered) on web browser. A light weight web server is used to respond to the web requests. Project is capable of running in cloud environment. Deployment is automated using Ansible.

2. TECHNOLOGIES USED

Technologies used for development and deployment of this project are listed below.

1. **Cloudmesh** - Cloudmesh provides Cloud Testbeds as a Service (CTaaS). It is a platform where one can manage all cloud accounts and local files enabling one to copy them between services. Project uses cloudmesh to connect to various cloud environments.

2. **Ansible** - Ansible is an IT automation tool that automates cloud provisioning, configuration management, and application deployment. Once Ansible gets installed on a control node, which is an agentless architecture, it connects to a managed node through the default OpenSSH connection type. Ptoject uses ansible for one-click deployment.

3. **Python** - Python is an object oriented, light weight programming language. Python is primary programming language used in this project.

4. **Mongo-DB** - MongoDB is an unstructured (NOSQL) database, which uses document-oriented data model. It stores data in flexible JSON-like documents. The project uses Mongo-DB to store GeoJSON data of earthquakes locally.

5. **Plotly** - Plotly is data analytics and visualization tool. One can create interactive graphs using plotly. It provides graphing libraries for Python. The plotly is used in the project as a visualization tool.

6. **Scikit-learn** - Scikit-learn provides tools for data analysis and data mining. Scikit-learn provides a wide range of learning algorithms. Scikits are the names given to the modules for SciPy, a fundamental library for scientific computing. As these modules provide different learning algorithms, the library is named as sciki-learn. In this project data classification is done using scikit learn.

7. **Cherrypy** - CherryPy is an object-oriented web application framework. It is designed for rapid development of web applications by wrapping the HTTP protocol. It is WSGI (Web Server Gateway Interface) thread-pooled web server. Cherrypy is used as a web server in the application.

3. DESIGN

Figure 1 shows main components and data flow of the application. There are four major components in the application.

3.1. WebServer

Purpose of web server is to listen to the user's HTTP GET request, call appropriate python method at the backend, get the response from the python method, and send it to the web client as a response. The server listens at port 8081.

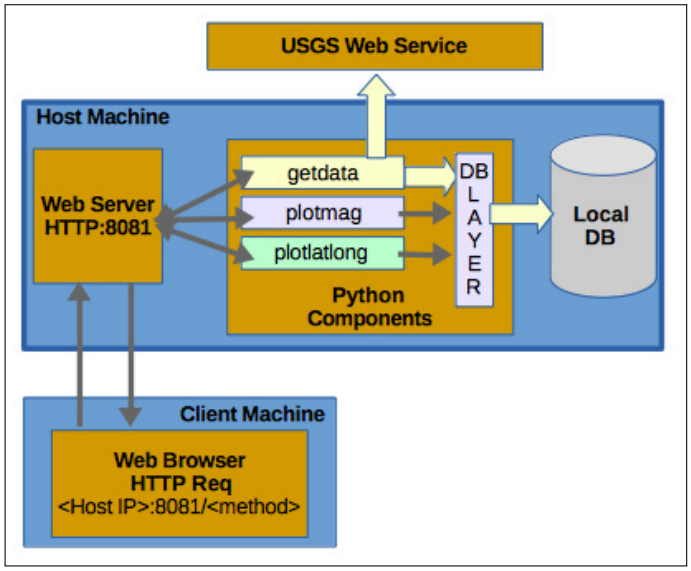

Fig. 1. Application Components

3.2. getusgsdata - getdata

getusgsdata component is responsible for fetching the data from USGS and storing it in local database. It uses dblayer common component to communicate with the local database. Running this component at least once is necessary as it downloads the data on which other components are dependent.

3.3. kmeansplot - plotmag

kmeansplot component implements the K-Means clustering algorithm and plots it using plotly library. It reads the Magnitude and Depth data from local database using dblayer common component. The algorithm classifies the data in 3 clusters based on the earthquakes magnitudes. Number 3 is selected by trial and error basis. Output of plotly.plot() function is set as *d*iv so that it returns the *d*iv with scatter plot and data, which has later been embedded into a simple HTML. This HTML is rendered on web browser as output.

3.4. dbscanplot - plotlatlong

dbscanplot implements DBSCAN algorithm for clustering the latitude-longitude data. It reads the data from local database using dblayer common component. Clusters are plotted on a globe using plotly's Scattergeo() function. Following programming statements ensure showing North and South America region on globe.

$$geo = dict(scope = (\prime northamerica\prime, \prime southamerica\prime)$$

$$projection = dict(type =\prime orthographic\prime)$$

For clustering lat long data DBSCAN algorithm is used instead of K-Means because, (1) for clustering lat long data it needs an algorithm that can handle arbitrary distance function, (2) K-Means works well with linear data. Lat long data is not linear, (3) Unlike K-Means, number of clusters are not required to mention in DBSCAN. It is hard to predict clusters when locations are spread across world.

4. IMPLEMENTATION

4.1. USGS Web Service and Data Set

USGS earthquake data is fetched using the USGS web service and passing relevant parameters to it. Out of the voluminous world-wide data of decades this project uses data of earthquakes appeared in North and South America for duration of 2 years from 2015-1-1 till 2017-1-1, having magnitude over 4. To select North and South America region data, its Latitude and Longitude are taken from Search Earth Catalog tool provided by USGS. [2]. Web service returns data in JSON format.

4.2. Data Processing and Visualization

Severity of earthquakes. Severity of earthquakes is analysed by their magnitude and depth. Data is classified into clusters using K-Means clustering algorithm (Section 4.1). Clusters are plotted on a interactive scatter plot.

Region affected the most by earthquakes. Analysing Latitude-Longitude of epicenter of the earthquake shows the regions where earthquakes are frequent. Lat-long data is clustered using DBSCAN algorithm (Section 4.2). Clusters are plotted on a geo-scatter plot on a world map.

4.3. Deployment

Project is deployed using ansible. Ansible jobs are collected in a playbook and run on virtual clusters provided by Chameleon and Jetstream cloud. The tasks include cloning the git repository, installing software stack, installing dependencies and installing MongoDB.

4.4. Local Data Storage

To avoid frequent web service calls and unnecessary traffic, data once downloaded, is stored locally in MongoDB for further usage by other components. Data is fetched using pymongo library.

4.5. Steps to Execute

Steps to execute the project are explained in detail in README.md file [3].

5. CLUSTERING ALGORITHMS

5.1. K-Means Clustering

Given a target number, k, of clusters to find, K-means algorithm locates the centers of each of those k clusters and the boundaries between them. It does this using the following algorithm [4].

- Step 1: Start with a randomly selected set of k centroids
- Step 2: Determine which observation is in which cluster, based on which centroid it is closest to (using the squared Euclidean distance.

$$\sum_{j=1}^{p}(x_{ij} - X_{i^1j})^2$$

  where p is the number of dimensions)
- Re-calculate the centroids of each cluster by minimizing the squared Euclidean distance to each observation in the cluster
- Repeat 2. and 3. until the members of the clusters (and hence the positions of the centroids) no longer change.

5.2. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Application with Noise) is a clustering algorithm. Given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), and marks points as outliers if they lie alone in low-density regions. Unlike K-Means Clustering, DBSCAN doesn't need to specify the number of clusters as it finds all the clusters that satisfy the requirement. Following points summarize the algorithm [5].

- Step 1: For each point in the data set, an n-dimensional sphere of radius *e*psilon is drawn around the point (if you have n-dimensional data).
- Step 2: If the number of points inside the sphere is larger than *m*in-samples, the center of the sphere is set as a cluster, and all the points within the sphere belong to this cluster.
- Step 3: Loop through all the points within the sphere with the above 2 steps, and expand the cluster whenever it satisfy the 2 rules.
- Step 4: For the points, which do not belong to any cluster, are treated as outliers.

6. BENCHMARKING

The performance of application was tested on the speed and latency while data input/output and data processing. Two different sized datasets were used for testing. They are given in Table 'Datasets'. The Table 'VM Configuration' shows details of VMs used for performance testing. Results are shown in Table 'Benchmark Results'.

Table 1. VM Configuration

Cloud	Chameleon	Jetstream
Image	Ubuntu-Server-14.04-LTS	ubuntu-14.04-trusty-server-cloudimg
OS	Ubuntu 14.04	Ubuntu 14.04
RAM CPU (Cores)	2 GB	2 GB
HDD Size	20 GB	20 GB
Flavour	m1.small	m1.small
Group	pearth	pearth
Assign Floating IP	True	True

Figure 2 shows time taken in seconds for data read and data processing with small dataset. Figure 3 shows the result with large dataset.

Performance tests are included in the python scripts themselves. As the scripts are executed, results are written in text files in 'benchmark' folder at project directory.

7. RESULT AND ANALYSIS

Table 3 shows the time taken to run ansible playbook successfully on various environments. It took maximum time to deploy the

Table 2. Datasets

Parameter	Dataset1	Dataset2
Region	North, South America	North, South America
Duration	2 years	2 years
Min Magnitude	4	3
Data Size	Small	Large

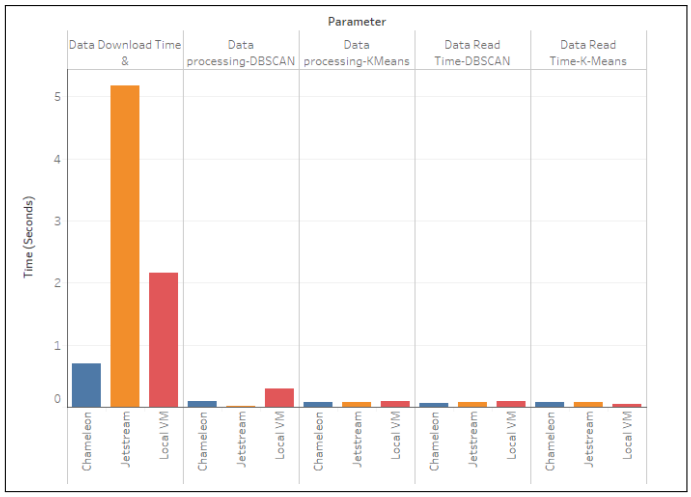

Fig. 2. Time taken for small dataset

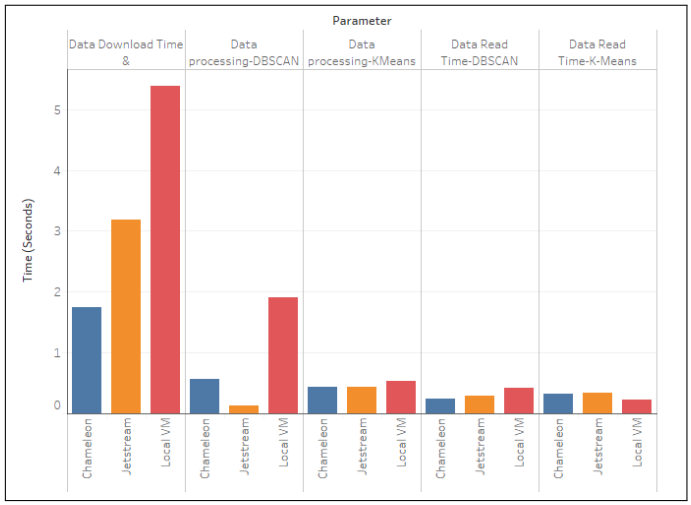

Fig. 3. Time taken for large dataset

application on jetstream. Possible reason could be slow network connectivity.

Table 3. Time taken for Deployment

Environment	Time (hh:mm:ss)
Local VM	00:07:35
Chameleon	00:08:12
Jetstream	00:10:08

Figure 4 shows default page showing application usage. Output of K-Means clustering of magnitude and depth data is given in Figure 5. Figure 6 shows a closer look at scatter plot. Figure 7 is the output of DBSCAN clustering of earthquake locations. Both the graphs are interactive. On the 'Magnitude and Depth' graph one can choose one or more clusters. Mouse hover shows magnitude and its corresponding depth. The 'Earthquake locations' globe shows lat-long information on mouse hover. The user can rotate the globe with mouse key press.

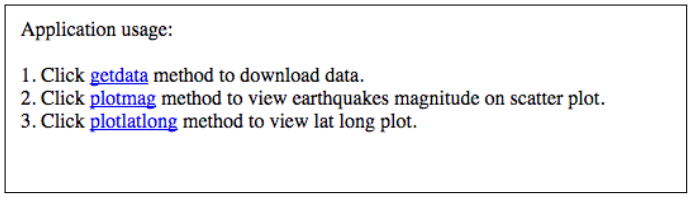

Fig. 4. Application usage

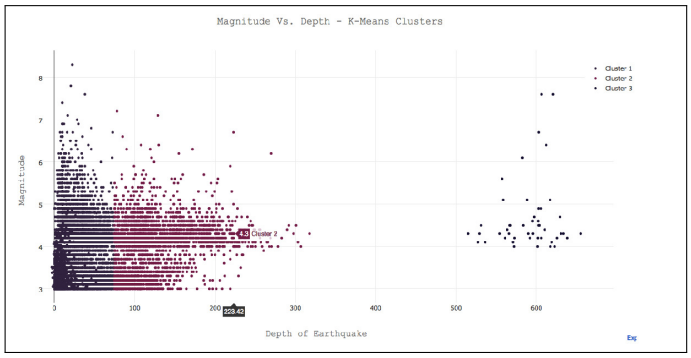

Fig. 5. Magnitude and Depth(Kms)

The focus of an earthquake is the actual point underground where rocks break. From the results it is clear that there were fewer earthquakes having the focus deep, greater than 300kms. Earthquakes having shallow focus (less than 70kms) are comparatively lesser than the ones having intermediate focus (70kms to 300km). Thus, maximum earthquakes occurred between 70kms to 300kms deep underground.

Analysis of earthquake locations shows that maximum earthquakes have happened in the Pacific coastal area of North and south America. We can confirm this result with USGS facts, which states "The majority of the earthquakes and volcanic eruptions occur along plate boundaries such as the boundary between the Pacific Plate and the North American plate. One of the most active plate boundaries where earthquakes and erup-

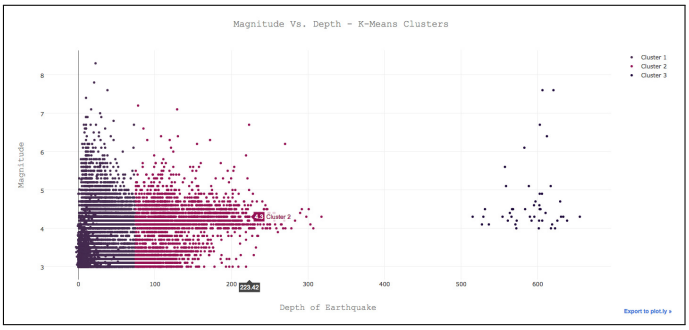

Fig. 6. Close look at the cluster

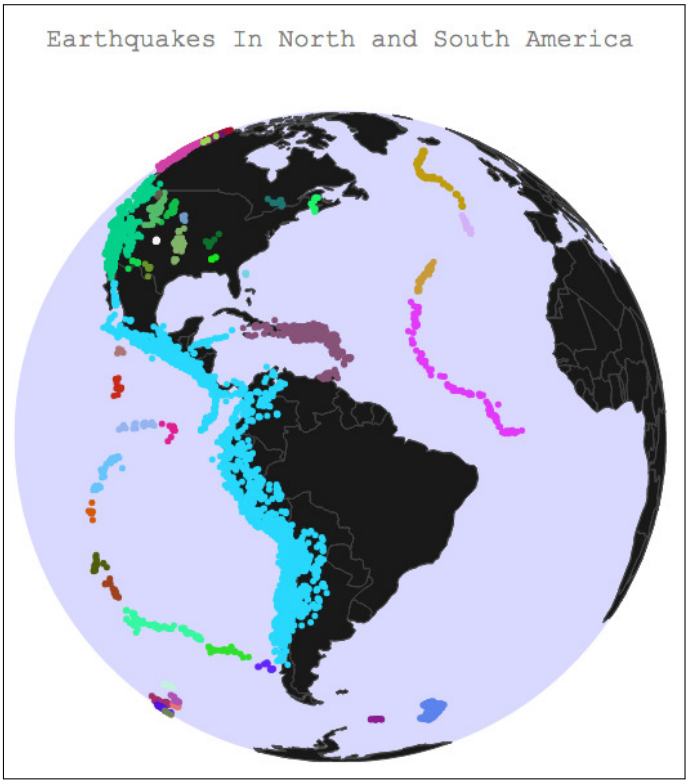

Fig. 7. Earthquake locations on globe

tions are frequent, for example, is around the massive Pacific Plate commonly referred to as the Pacific Ring of Fire [6]."

There is a subtle difference in output of K-Means and DBSCAN algorithms. K-Means worked with lesser number of clusters (3), while DBSCAN created greater number of clusters, more than 50 in this case. K-Means produced output linearly, cluster1 to cluster3. DBSCAN produced non-linear output. A cluster in Pacific ocean is Cluster28, whereas, the cluster adjacent to it is Cluster39.

8. CONCLUSION

This project was an opportunity to use Big Data open source software and projects. We can conclude that Ansible is a powerful tool for one click deployment and continuous integration. Cloudmesh client is a convenient tool to work with multiple cloud environments at the same time. Plotly is a powerful library that allows creating interactive visualization.

While researching for the algorithms we came to know that DBSCAN is better choice for classification if one doesn't want to specify number of clusters beforehand.

Analysis of earthquakes data showed interesting facts. In 2 years span from year 2015 till 2017 there were around 15,000 earthquakes of magnitude more than 3, and approx 7,100 earthquakes of magnitude more than 4.

9. ACKNOWLEDGEMENTS

This project is undertaken as part of the I524: Big Data and Open Source Software Projects course at Indiana University. The author would like to thank Prof. Gregor von Laszewski and his associates from the School of Informatics and Computing for providing all the technical support and assistance.

10. LICENSING

REFERENCES

[1] USGS, "Earthquake hazards program," Web Page. [Online]. Available: https://earthquake.usgs.gov/

[2] USGS, "Search earthquake catalog," Web Page. [Online]. Available: https://earthquake.usgs.gov/earthquakes/search/

[3] N. Sathe, "Readme," Code Repository, April 2017, accessed: 2017-4-25. [Online]. Available: https://github.com/cloudmesh/earth/blob/master/README.md

[4] B. Govan, "Clustering using scikit-learn," Web Page, May 2013. [Online]. Available: http://fromdatawithlove.thegovans.us/2013/05/clustering-using-scikit-learn.html

[5] Q. Kong, "Clustering with dbscan," Web Page, August 2016. [Online]. Available: http://qingkaikong.blogspot.in/2016/08/clustering-with-dbscan.html

[6] USGS, "Earthquake facts," Web Page. [Online]. Available: https://earthquake.usgs.gov/learn/facts.php

11. APPENDICES

Appendix A: The work on this project was distributed as follows between the authors:

Nandita Sathe. She completed all the work related to development of this application including research, testing and writing the project report.