A PROJECT REPORT ON
Prediction System for Proper Utilization of 911 Emergency Services

"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality"

SUBMITTED BY

Arjun Ravindran (40070983)
Ishaan Sharma (40053363)
Mohammed Hakeemuddin (40059007)
Santhosh Raviteja Gadadasu (40076463)
Sivaji Ganesh Chundu (40072826)

FALL 2018

COURSE NAME: PROGRAMMING ON THE CLOUD

COURSE NUMBER: COEN 6313

INSTRUCTOR: Prof. OMAR ABDUL WAHAB

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CONCORDIA UNIIVERSITY

DATE: 29th November, 2018

Prediction System for Proper Utilization of 911 Emergency Services

Arjun Ravindran: 40069977, Ishaan Sharma: 40053363, Mohammed Hakeemuddin: 40059007, Santosh Raviteja Gadadasu: 40076463, Sivaji Ganesh Chundu: 40072826

Submitted To: Professor Omar Abdul Wahab
Dept. of Electrical and Computer Engineering, Concordia University, Montréal, Canada

Abstract: As the population is growing day by day, people who encounter emergency situations like accidents because of traffic and fire are also increasing. Government is responsible to facilitate effective emergency response services when people suddenly encounter emergency situations. In this project, 911 emergency calls data set is analyzed to provide information for the government in the abstract form, which will be useful to manage the available human force and machine equipment in different places. So, according to the requirement response services can be increased or decreased which is beneficial to both the public and the government.

Index terms: Pre-processing, Analysis, Visualization, Predication, Web and Mobile Applications, URL to access source code.

INTRODUCTION

In this project, 911 Emergency calls data set of the city Montgomery of Pennsylvania Province is analyzed. The data set is taken from the Kaggle which contains around one lakh records with seven fields containing details about type of emergency, latitude and longitude, zip code, town ship, address, and so on.

Objectives:

- To analyze the 911 emergency situations which have occurred already, which in turn will help the government to decide whether to increase or decrease the services in a particular area.
- To provide some visualizations of the analyzed data for easy understanding of the situation.
- To implement a web and mobile application which displays the visualizations of the analyzed data.
- To predict the number of emergency situations that may occur on a particular day, so that government can be well prepared for the situations accordingly.

Problem Statement: In emergency situations each second is very crucial to save the lives of people. Currently the utilization of emergency services is not up to the mark in certain areas because of the unawareness about things like the number of much human force needed in certain area, the range of area allocated for an emergency service center. In this project, the existed problems in emergency response services can be fixed by making some analysis on the already available data set and estimating the requirement of the emergency resources to be maintained in the future by making some predictions.

Assumptions: It is always not possible to always have accurate data. In this case, an assumption is made that all the calls received are genuine. The data set using this project had some missing values in the field of zip code. Even though some zip codes are filled by using VLOOKUP in excel, still there were some zip codes which cannot be filled and those are filled with the next zip code values. And in this case, it is assumed that, it won't lead to any invalid analysis.

Methodology: The 911 emergency calls data set which is taken from the kaggle website is deployed in the google cloud. Pre-processing of the data set is done using Google dataprep. Then the processed data set is used for analysis using different technologies like Big Query, Hive, and PySpark and the performance of the technologies is observed. Then for better understanding, the analyzed data is visualized by using external web services like Tableau and Seaborn. By creating a REST API service, the visualizations are displayed on web application and mobile application. Then the analyzed data is used for training a model by machine learning using the gradient booster regression method. Machine Learning is done with scikit-learn library. Then finally the comparison and performance of different technologies used in this project are observed.

Schedule: The first four phases of the project,

- Getting familiar to the cloud computing platforms and its configuration (Used Cloud platform: Google Cloud).

- Scenario selection (Selected Scenario: Recommender Systems).
- Data set selection (Selected data set: 911 Emergency Calls Data Set from Kaggle Website).
- Preprocessing on the data set: Data cleaning and the data reduction are the two things which are done on the selected data set using "Goolge Cloud DataPrep". These two things are done in order to fill the missing values and to remove some unnecessary fields.

All the mentioned project phases till now are completed before October 4th, 2018.

The remaining phases of the project, programming and observations are scheduled as below.

- Analyzing the data using different technologies: October 25th, 2018
- Visualization of analyzed data by using external web services: November 5th, 2018.
- Creating a front end (web and mobile applications) to display the visualizations: November 20th, 2018.
- Training the analyzed data using machine learning concepts: November 25th, 208.
- Report and Presentation: November 27th, 2018.
- Demo: November 29th, 2018.

SYSTEM ARCHITECTURE

The main purpose of the service developed in this project is to provide a detailed analysis of the 911 emergency calls with the help of visualizations (Bar graphs, Maps). Then we are predicting the number of emergency calls that may occur in a certain period of time by training the model using the analyzed data with the concepts of Machine Learning.

The 911 dataset was taken from the Kaggle website which contains around 180000 records. The dataset consists of the following fields which are shown in table 1.

Field Name	Description
Latitude	The Latitude of the call location
Longitude	The Longitude of the call location
Description	Description of the call
Zip	Zip code of the call location
Call Type	Type of 911 call
Time Stamp	Time stamp of the call
Township	Area where the call originated
Address	Address of the call

Table 1: Fields of Data Set

The process begins with the preprocessing of the data set and continues with main parts, analysis of data set and visualizations using external web services. Then the visualizations are displayed on the front end using web application and mobile application. Finally a model is trained with concepts of machine learning by using the analyzed data for the purpose of future prediction. The system architecture is shown in figure 1.

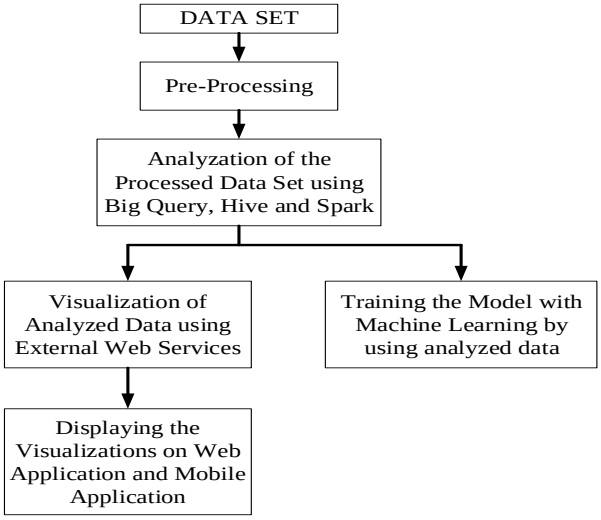

Figure 1: System Architecture

In this project, the major part of the project is done utilizing the services of the GOOGLE CLOUD PLATFORM. For data preprocessing CLOUD DATAPREP is used, while for data processing CLOUD DATAPROC is used. External Web Services Tableau and Seaborn libraries are used for visualization of analyzed data.

Configuration and Preprocessing Details: A new instance is created using the facilities providing by the Compute Engine. An example instance is created as shown in figure 1.

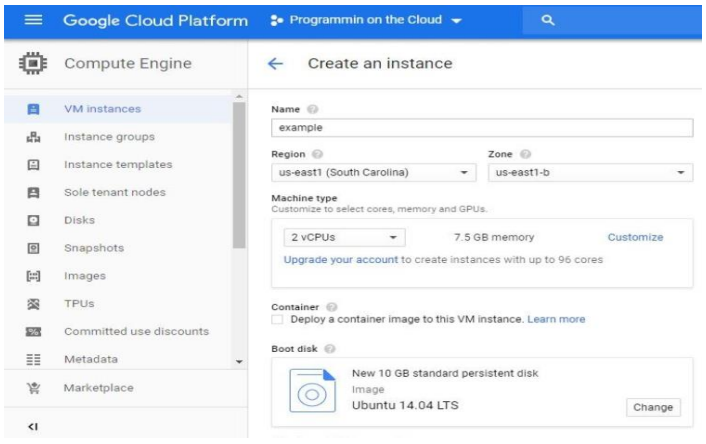

Figure 2: Creating a new instance

After creating the instance, the performance of the new VM Instance is observed which is as shown in figure 2. Actually the performance is observed after running a simple spark

job, if it is observed after running to many jobs the CPU utilization will be more.

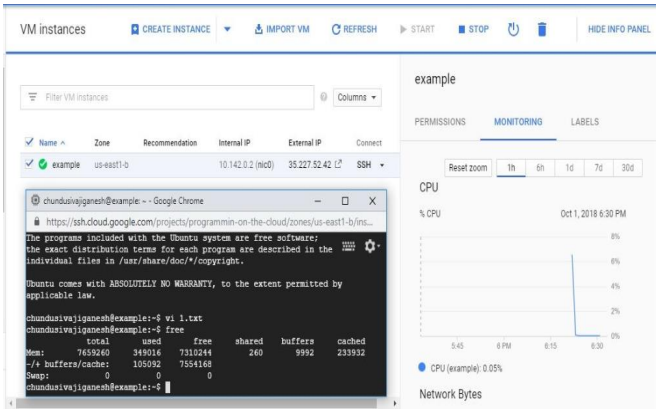

Figure 3: Performance of the created instance

After creating the new instance, a cluster is created according to the requirements. For example, a cluster with 2 nodes is created as shown in figure 3.

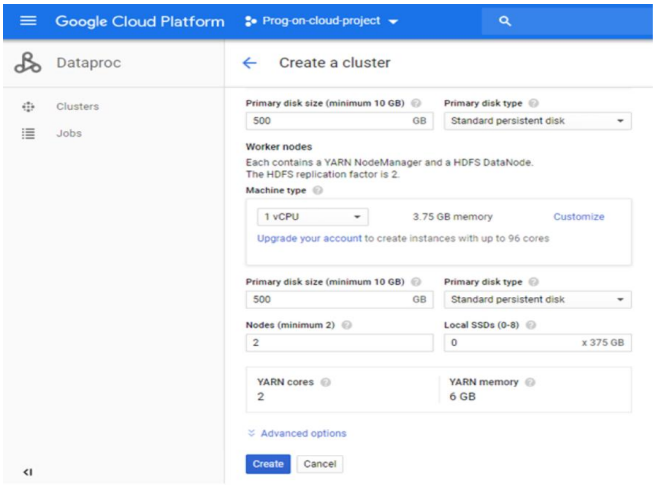

Figure 4 : Creation of Cluster

After creating the cluster, a simple spark job is developed which is shown in the documentation of spark and the developed job is submitted and make it run on the cluster as shown in the figure 4.

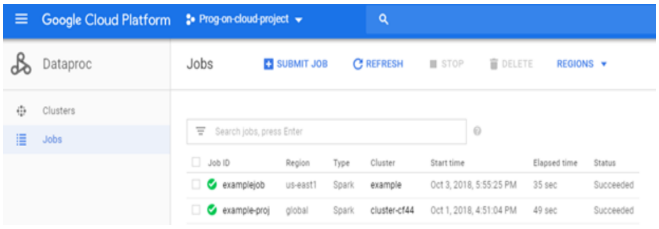

Figure 5 : Running a job on cluster

Then the data set taken form the kaggle web site is preprocessed using DataPrep is loaded to the google cloud BIG QUERY. While pre-processing, some unnecessary fields are removed and missing values in the zip codes field are replaced by relating them with the city names and zip codes and at the same time data filled in the same field is divided in to two fields. Some details about preprocessing are displayed in the figure 5.

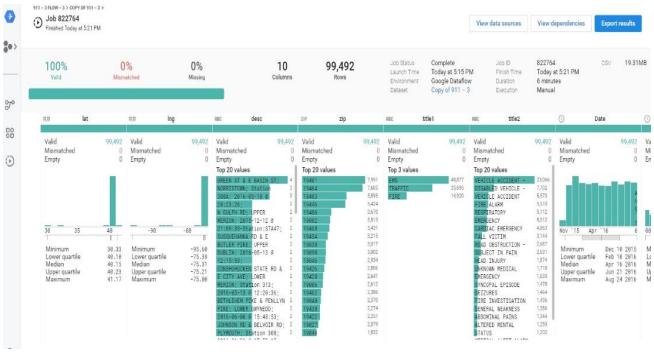

Figure 6 : Pre-Processing of data set

Analysis of Data Set using Big Query: Data sets usually are not 100% accurate and may not contain the information according to our requirements. In this project, it is required to have the data related to each city separately and at the same time it is required to have the data category wise. For this purpose, it is required to filter the data according to requirements. Querying the data and storing in the separate table is one of the most traditional technique which is done to achieve the above purpose. Google Cloud BIG QUERY is one of the available technology which allows to achieve the above mentioned purpose easily. Big Query is a fast, highly scalable, cost-effective, and fully managed cloud data warehouse which can be used for data analysis, with built-in machine learning library. For displaying the data in terms of city and category wise, the data related to month and EMS category are filtered from the preprocessed data set using necessary queries. Result of simple query is shown in figure 7 which gives the ems category wise details.

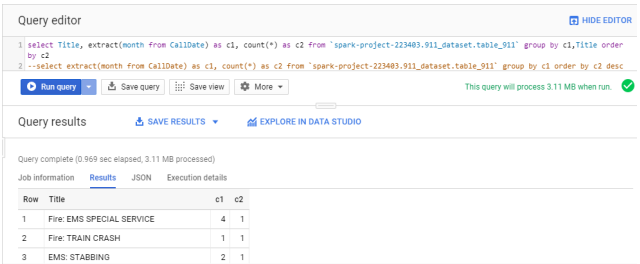

Figure 7 : Result of BIG QUERY

Analysis of data set using HIVE: Apache Hive is a data warehouse which is built on top of Apache Hadoop for providing data query and analysis. Hive gives an SQL-like interface to query data that can be stored in various databases and file systems which can be integrated with Hadoop. The same thing which is done by using big query

can be done using hive and a simple query which is applied on the data set is as shown in figure 8.

```
0: jdbc:hive2://localhost:10000/default> select category as c1, count(*) as c2 from 911_table group by category ord
er by c2 desc;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a diffe
rent execution engine (i.e. spark, tez) or using Hive 1.X releases.
+------------+--------+--+
|     c1     |   c2   |
+------------+--------+--+
| "EMS"      | 48853  |
| "Traffic"  | 35693  |
| "Fire"     | 14903  |
+------------+--------+--+
3 rows selected (66.205 seconds)
0: jdbc:hive2://localhost:10000/default>
```

Figure 8 : Result of HIVE Query

Analysis of data set using PYSPARK: Py4J is the popularly known library which is integrated with PySpark. PySpark allows python to interface dynamically with JVM objects (RDD's). Apache Spark comes with an interactive shell for python as it does for Scala.
First, a cluster has to be created with master and slave nodes as shown above. After developing a spark job using pyspark, the input pyspark file and data storage location is mentioned to run the job on the 911 calls data set. One of the main advantage of performing analysis using spark is that the developed job can be run on the data where it is located rather than extracting the data from its location and processing it. The result of spark job gives the details regarding city wise calls as shown in figure 9. The analyzed data will be stored in google cloud storage. Then a SQL instance is created in google cloud SQL with MYSQL environment. The analyzed data is imported in google cloud SQL for further process.

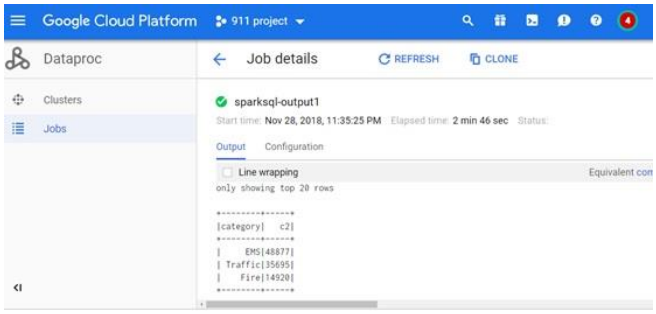

Figure 9 : Result of Spark Job

Visualization of analyzed data: The analyzed data which is imported to google cloud SQL database, is connected to Tableau platform by using IP address of the SQL instance for visualization. The data is visually represented in the form of maps and bar graphs for easy understanding using external web services like Tableau and Seaborn. Some of the visualizations of analyzed data using Tableau and Seaborn are shown below. The visualization shown in figure 10 depicts the total number of calls received in each month, that is in month of January total number of calls are around 12000 and which is highest when compared to other months. By this the government can be able to provide more number of emergency services for the public.

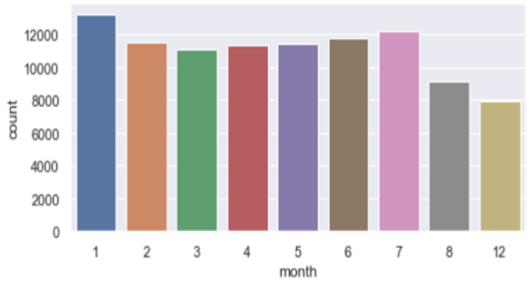

Figure 10 : Visualization based on monthly wise

The visualization shown in figure 11 depicts the total number of calls received on a particular week day. As seen in the figure, on Sunday the number of emergency calls are very low when compared to other days so that there will be a chance for the government to decrease the number of emergency services on Sunday.

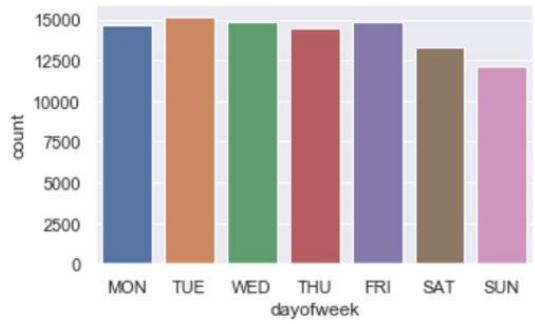

Figure 11: Visualization based on day week wise

Similarly, the visualization about category wise is shown in figure 12. Thus, by having this information government can allocate more services for EMS category by decreasing the resources for fire category.

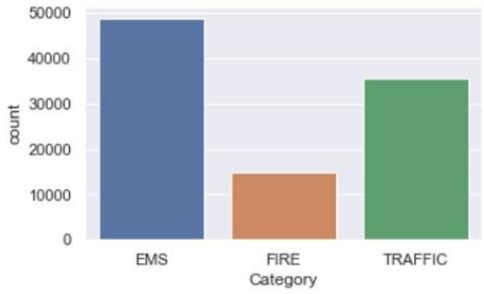

Figure 12: Visualization based on category wise

The below visualization depicts the number of calls in a particular week day and in a particular time slot, which allows to pay more attention during that particular time slots when compared to other time slots. This visualization is done using Seaborn.

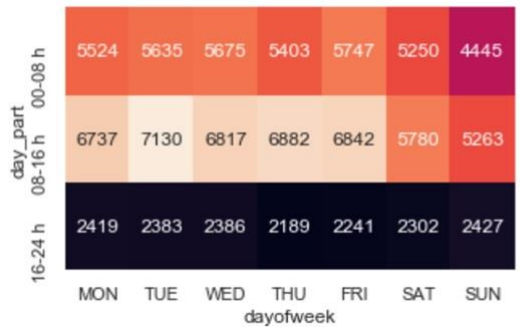

Figure 13: Visualization based on week day and time

The visualisation depicts the information on maps, so that when cursor is placed on map it shows the number of calls received in that area, along with the information related to zipcode, latitude and loongitude values.

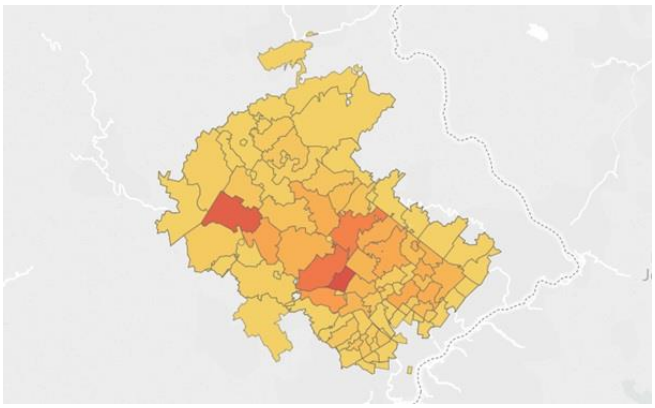

Figure 14: Visualization on maps

Development of web application: A web application is developed which displays the visualizations on front end when the necessary inputs are given. The web application is developed using dot net frame work. The visualizations are displayed on front end by creating a REST API. This user interface will play a crucial role to access the services provided by the system developed in this project from anywhere in the world.

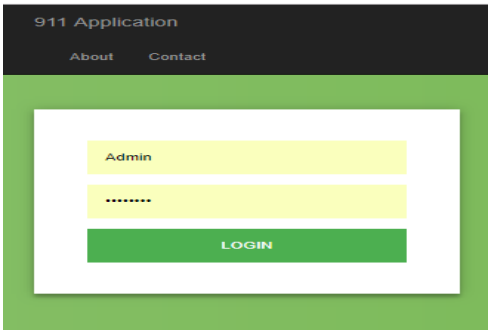

Figure 15: Login page of web application

The webpage displayed in figure 16, allows to select what kind visualization required like if all cities category is selected it will display the image shown in figure 14.

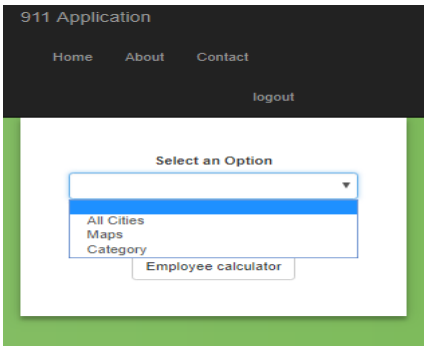

Figure 16: Home Page of web application

The calculation web page shown in figure 17, will calculate the number of employees required for a category when number of emergency calls received for respective category is mentioned.

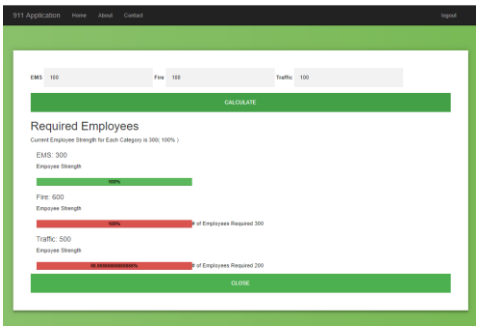

Figure 17 : Displaying no. of employees required on web page

Development of Mobile Application: A mobile application is developed for the service using Framework 7 and Phone Gap. The login page and drop-down menu are shown in figure 18 and figure 19 respectively.

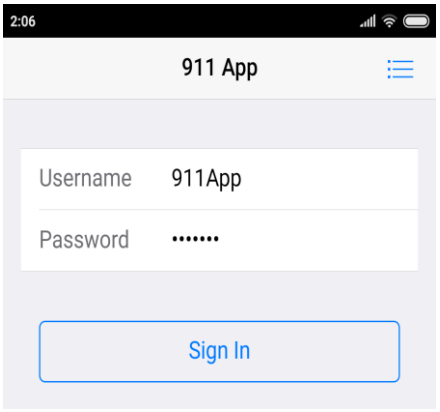

Figure 18 : Login page on mobile application

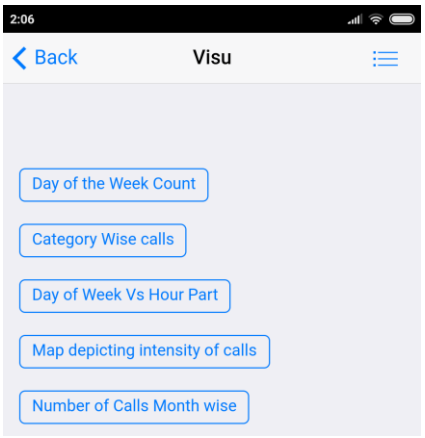

Figure 19: Drop down menu on mobile application

Prediction using Machine Learning: The number of calls will happen on a certain day can be predicted by training the model using already existed data. The model is trained and created by using the sci-kit learning. Python language is used for programming and for numerical analysis pandas and numpy has been used. The text data is converted 0, 1, 2, for easy analysis and understanding for machine learning model. Similarly, latitude and longitude values are converted to easily operable geo-grid. Gradient booster algorithm is used to train the model and data set is divided as test and train data sets to 30% and 70% respectively. The number of events or calls per day can be predicted on a given day when that day of a particular month is given as input. This will help for better allocation of resources as whole on any day, since the overall prediction rate is approximately 85% true. The prediction about number of calls on certain day is shown in figure 20.

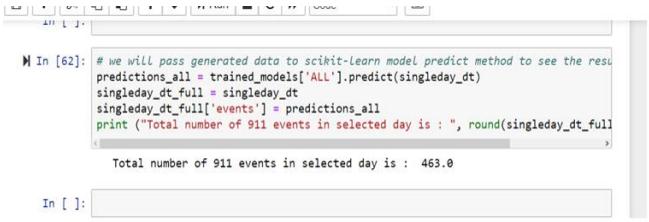

Figure 20 : Predication using machine learning

URL to access source code in Git Hub:
https://github.com/raja27ravi/911-App-code

OBSERVATIONS

Execution time: The time taken for the execution of the pyspark job run in google cloud takes around 140 seconds. The execution time varies depending on the size of the input data of the pyspark job.

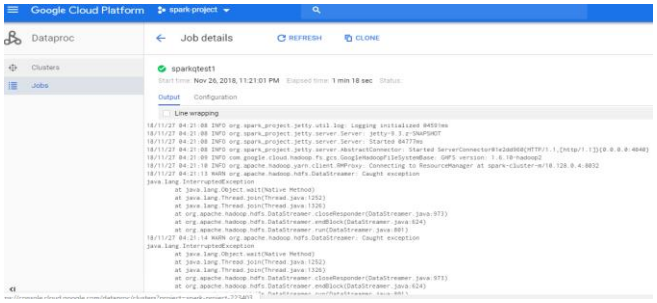

Figure 21: Execution of Pyspark job

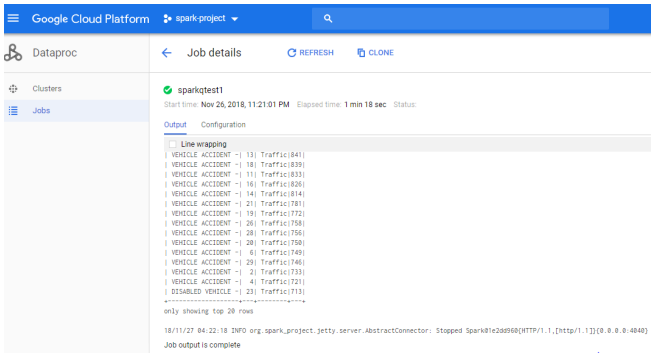

Figure 22: Completion of Pyspark Job

Scalability: Autoscaling option is turned on in google cloud in which the auto scaler increases the number of virtual machines by a minimum of instances or extra 50% when the CPU utilization reaches close to 100% or a user defined value. Another type of autoscaling is based on the HTTP load on the instances. New virtual machines are created when the load goes above a user specified limit. These are shown in the below figures.

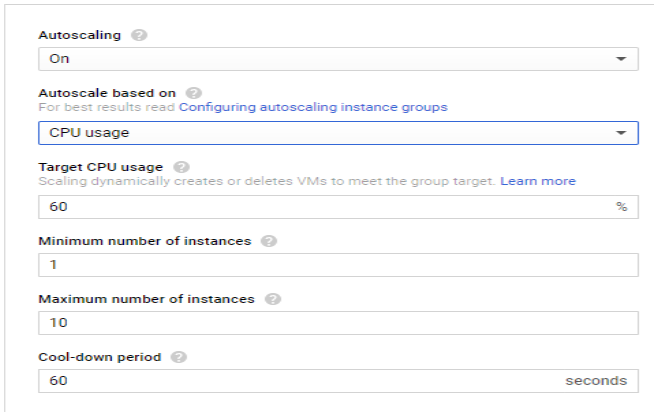

Figure 23: Autoscaling based on CPU utilization

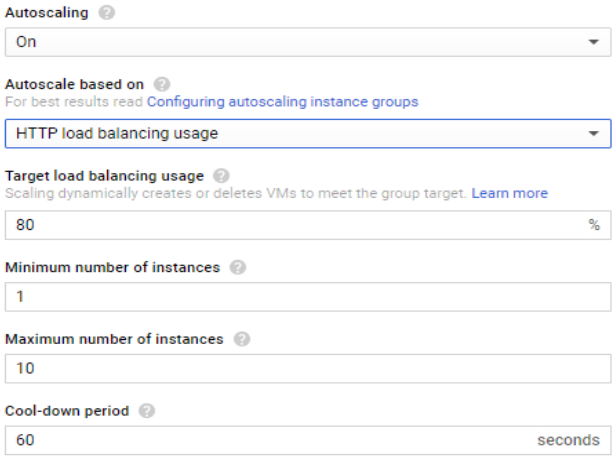

Figure 24: Autoscaling based on HTTP load

CPU and memory utilization: CPU utilization peaked to a maximum of 44.92 % when the pyspark job was run. The disk I/O operations and memory usage also peaked when the pyspark job was run. The values are displayed in the below figures.

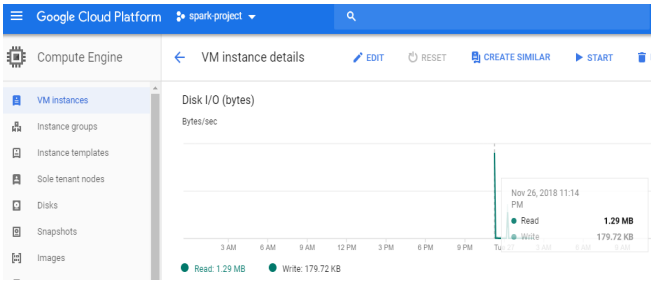

Figure 25: Disk I/O memory usage

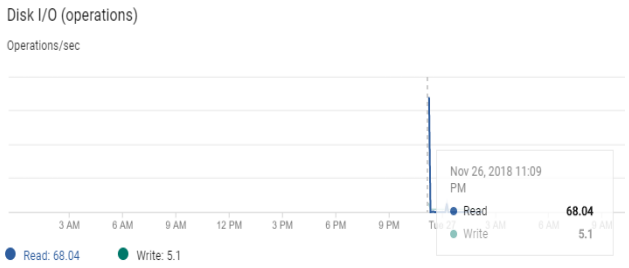

Figure 26: Disk I/O operations

Reliability and consistency: Our output consists of maps and graphs which stay the same irrespective of the size of the input data. Only the values in the maps and graphs get updated.

Security: Since our project is run on google cloud, it is highly secure as claimed by google. Our data is stored on google cloud storage which can be accessed only by the user. The user may give share access to others. And the data stored is encrypted by google cloud using Advanced Encryption Standard (AES) algorithm. The data also have a redundancy factor of 3. The web and mobile application can be accessed by the admin or users created by the admin.

FUTURE WORK

- Currently the web application is not deployed in the cloud because of compatibility issues with dot net frame work. If the web application is developed using Java framework it can be deployed easily on google cloud.
- Modifications to the system to work with real time streaming data.
- Predicting the number of calls along with the specific reason, like fire accident occurred because of thunderstorms and not because of short circuit. And this can be done if the data set contains the reason for the accident.
- Making the system more autonomous, like make the system to perform analysis on upload of the data set. This part can be done writing google cloud functions
- Hitting machine learning model with more data to increase correctness of prediction, like data about time taken to clear the emergency.

CHALLENGES FACED

- Resolving configuration and environment problems in Pyspark (This is resolved by first running the Pyspark job on local and then deploying it on google cloud).
- Connecting Google Cloud SQL instance with Tableau (This is resolved by installing ODBC and JDBC connectors).
- Replacing the values of missing and mismatching zip codes in the data set (This is resolved by using VLOOKUP in excel).
- Tableau integration in web application (This is resolved by connecting with google cloud SQL).
- Connecting the individual parts of the project, like giving the output of Pyspark job to cloud SQL.

CONCLUSION

The analysis of the 911 emergency calls data set is done using different technologies (Hive, BigQuery and spark) after some preprocessing of the data set using DataPrep. The analyzed data is visualized by using the external web services like Tableau and Seaborn. Web and Mobile applications are developed to display the visualizations by creating a REST API. Finally, prediction of emergency situation on a particular day is done by training the model with machine learning. By using the service provided by the system developed in this project, the management of 911 EMS services can be done effectively to provide services to the public.

REFERENCES

- *Presentation slides of course Programming on the Cloud: Prof. Omar Abdul Wahab.*
- *https://www.kaggle.com/mchirico/montcoalert.*
- *https://cloud.google.com/docs/*
- *https://www.tableau.com/support/help*
- *https://www.datacamp.com/community/tutorials/seaborn-python-tutorial*
- *https://www.codementor.io/kunaldhawan93/big-data-analysis-using-pyspark-8y5yobb44*
- *https://dzone.com/articles/data-analysis-using-apache-hive-and-apache-pig*
- *https://scikit-learn.org/stable/tutorial/basic/tutorial.html*
- *https://docs.microsoft.com/en-us/visualstudio/?view=vs-2017*

APPENDIX

Technical Contribution of Each Member

	Data Preprocessing	Analysis of Data using Hive, Big Query and Pyspark	Using external webservices for visualizations	Mobile and Web Application Development	Prediction using Machine Learning
Arjun Ravindran	Participant	Lead	Lead	Participant	Participant
Ishaan Sharma	Lead		Participant		Participant
Mohammed Hakeemuddin	Participant	Participant	Participant	Lead	Participant
Santhosh Raviteja Gadadasu	Participant	Lead	Participant	Participant	Lead
Sivaji Ganesh Chundu	Participant	Participant	Lead	Participant	Lead

Presentation link: https://prezi.com/p/zbk4t_5cpmeb/

Demo link: https://www.youtube.com/watch?v=XYJwni3RLAY&feature=youtu.be

Github URL : https://github.com/raja27ravi/911-App-code

Figure 2: Creating a new instance

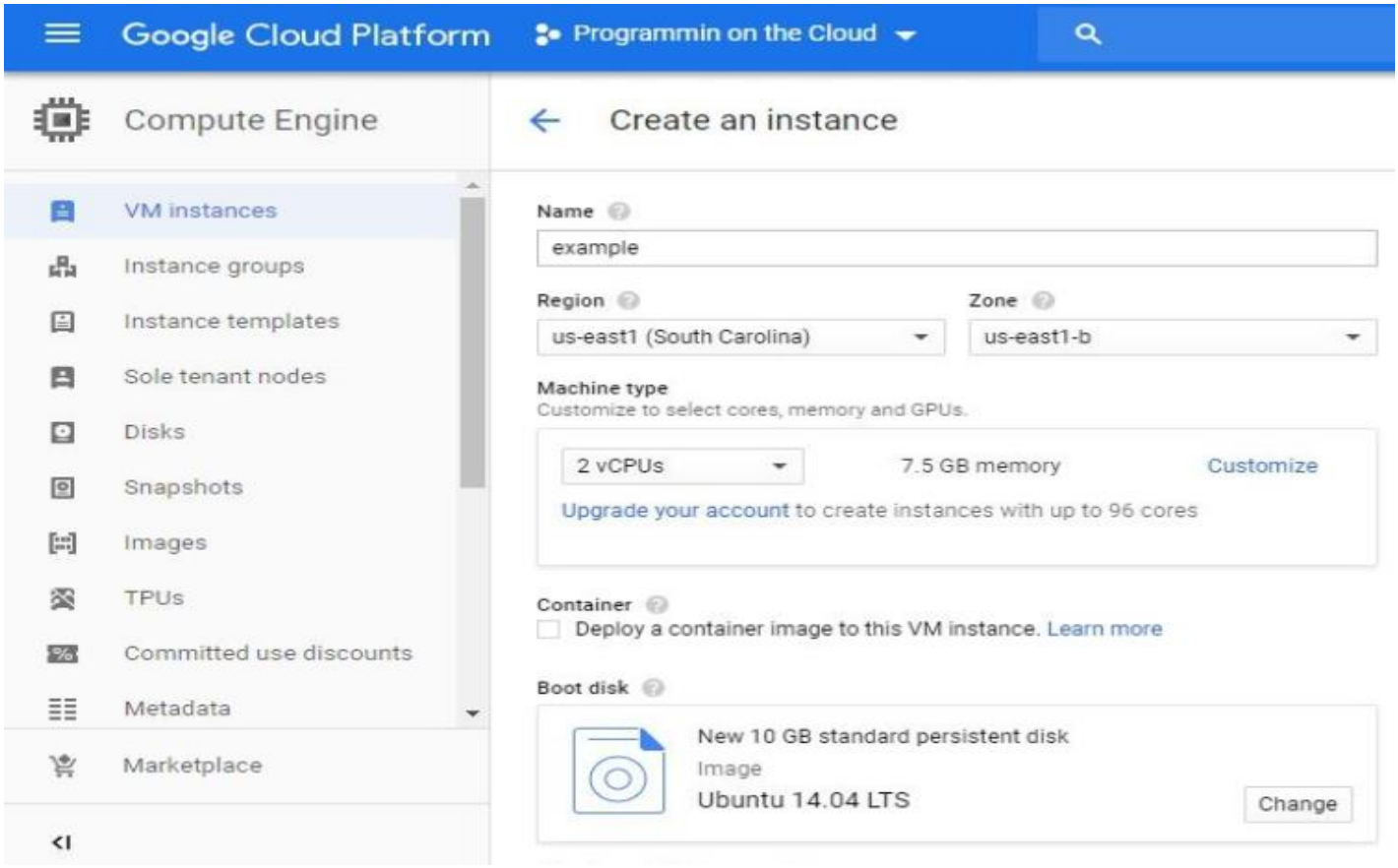

Figure 3: Performance of the created instance

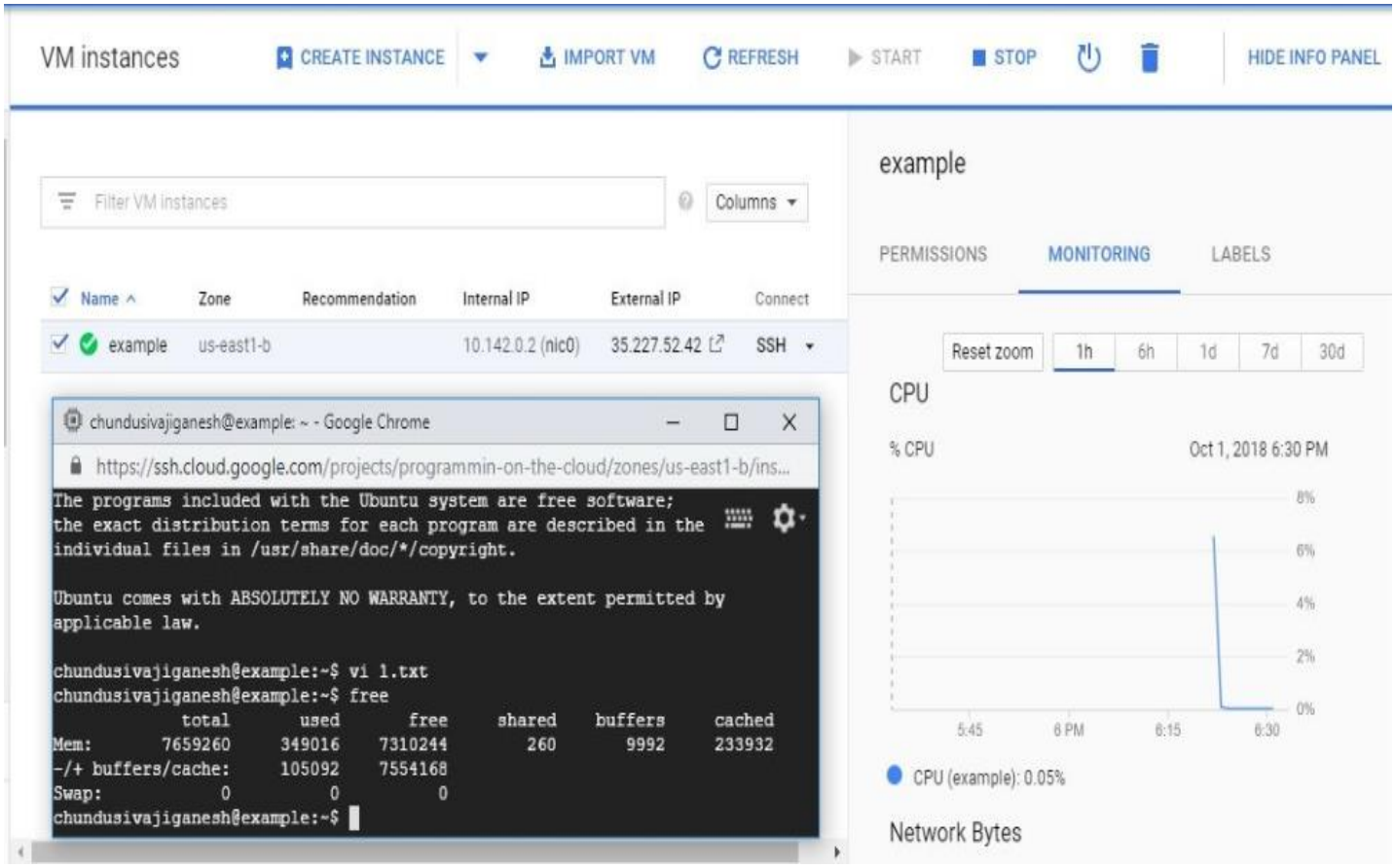

Figure 4 : Creation of cluster

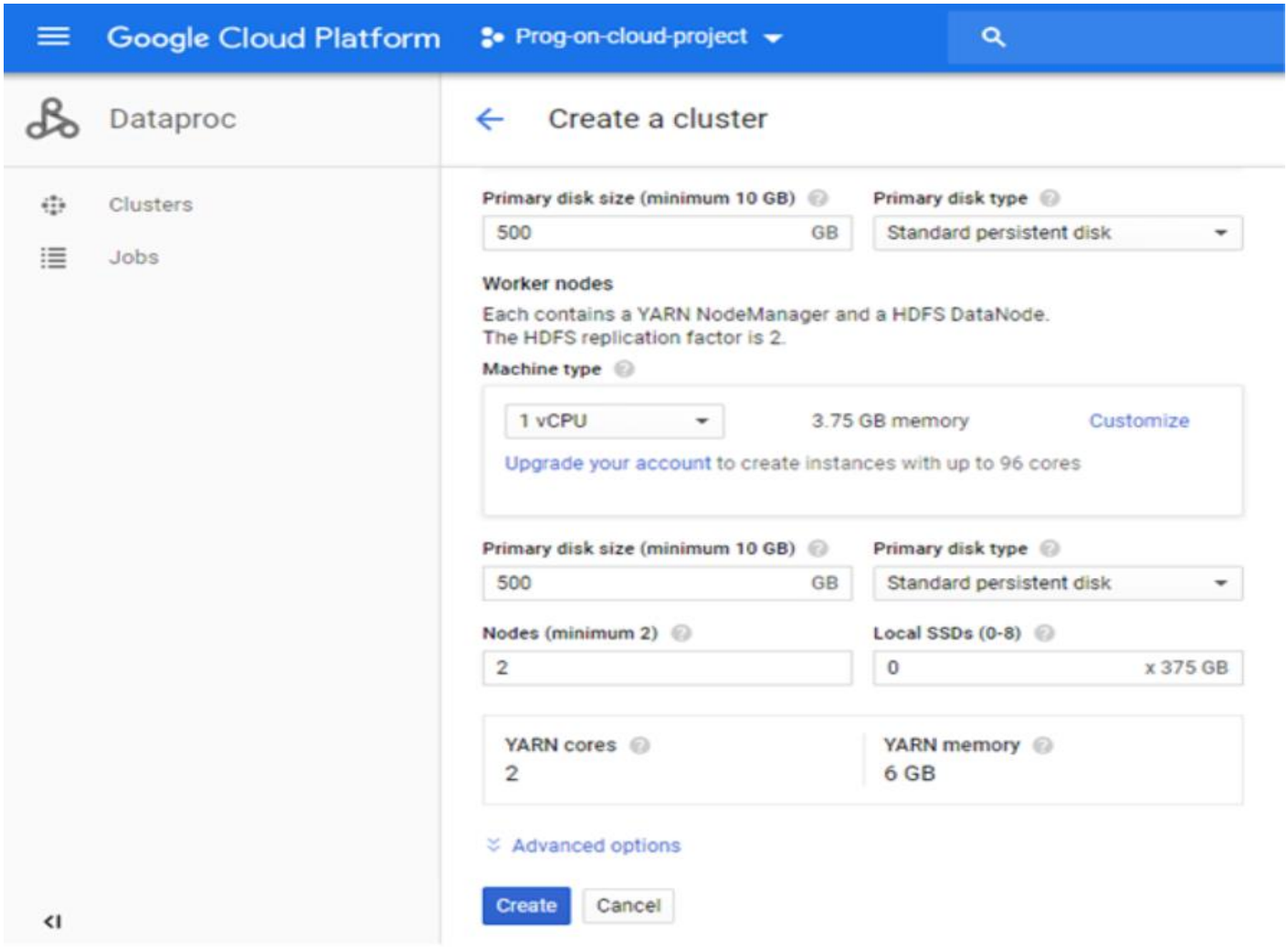

Figure 5 : Running the job on cluster

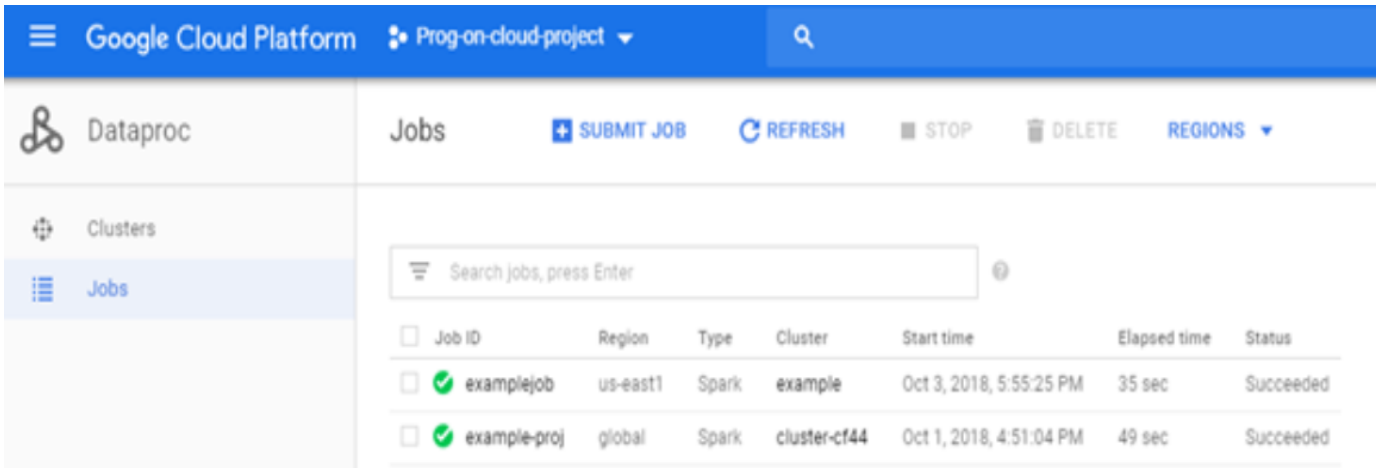

Figure 6 :Pre-processing of data set

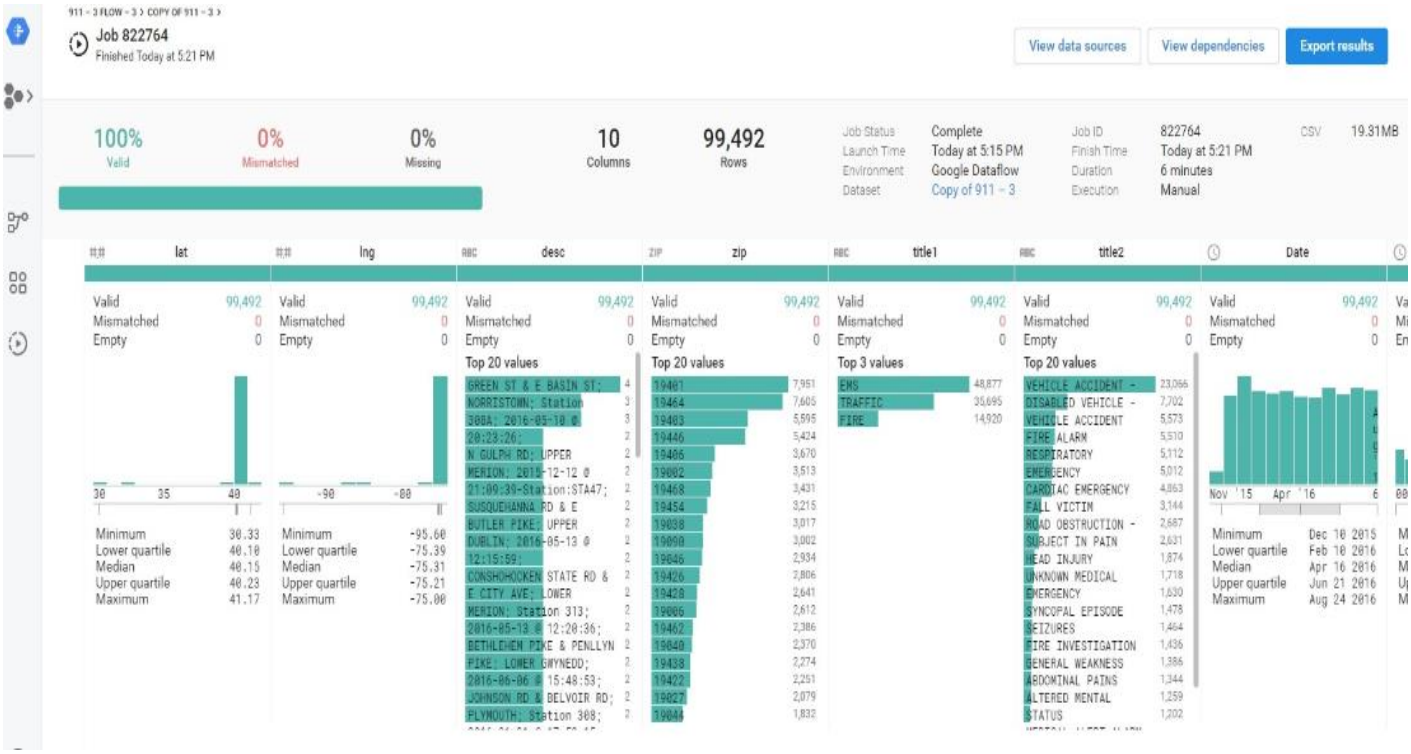

Figure 7: Result of Big Query

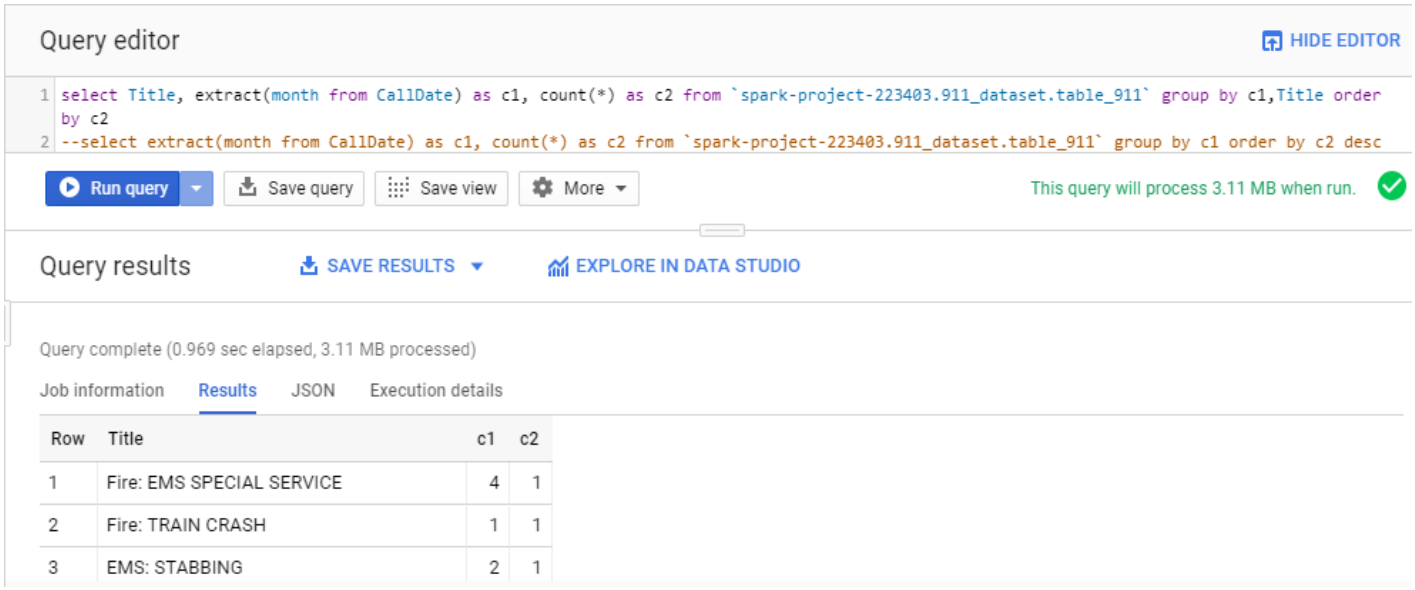

Figure 8: Result of Big Query

```
0: jdbc:hive2://localhost:10000/default> select category as c1, count(*) as c2 from 911_table group by category ord
er by c2 desc;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a diffe
rent execution engine (i.e. spark, tez) or using Hive 1.X releases.
+------------+--------+--+
|     c1     |   c2   |
+------------+--------+--+
| "EMS"      | 48853  |
| "Traffic"  | 35693  |
| "Fire"     | 14903  |
+------------+--------+--+
3 rows selected (66.205 seconds)
0: jdbc:hive2://localhost:10000/default>
```

Figure 9: Result of spark job

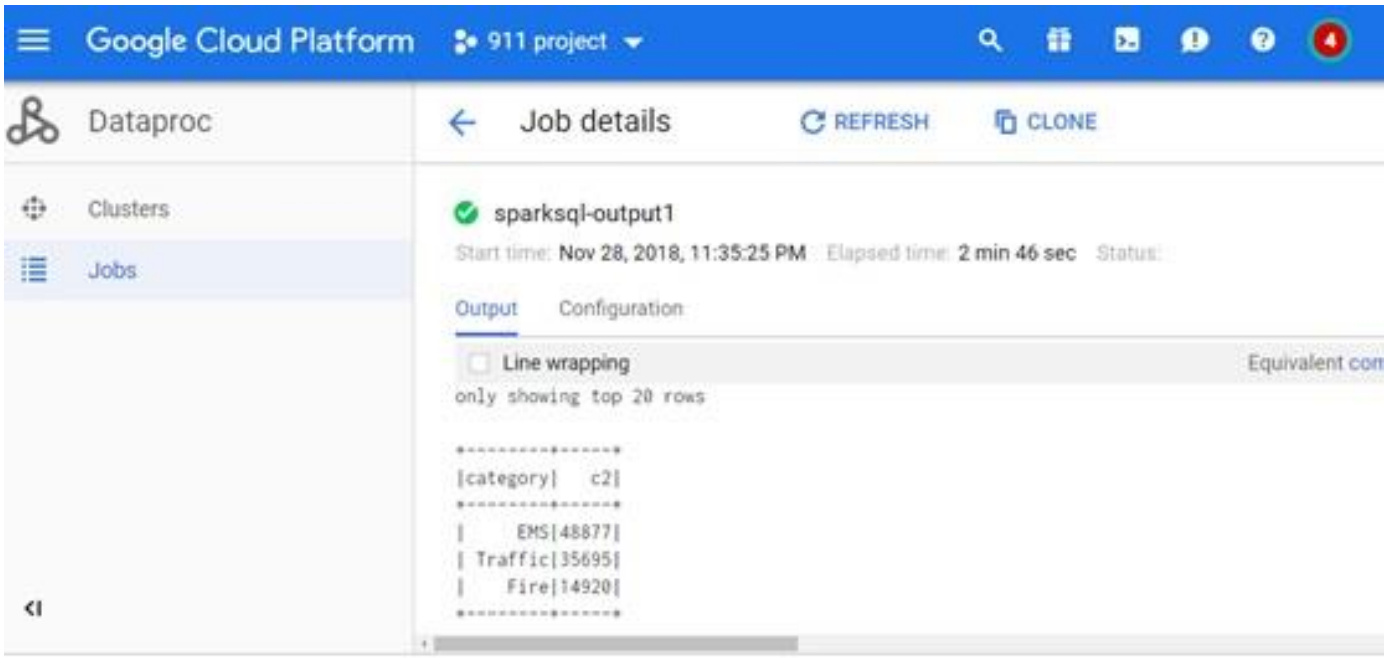

Figure 21 : Excution of Spark Job

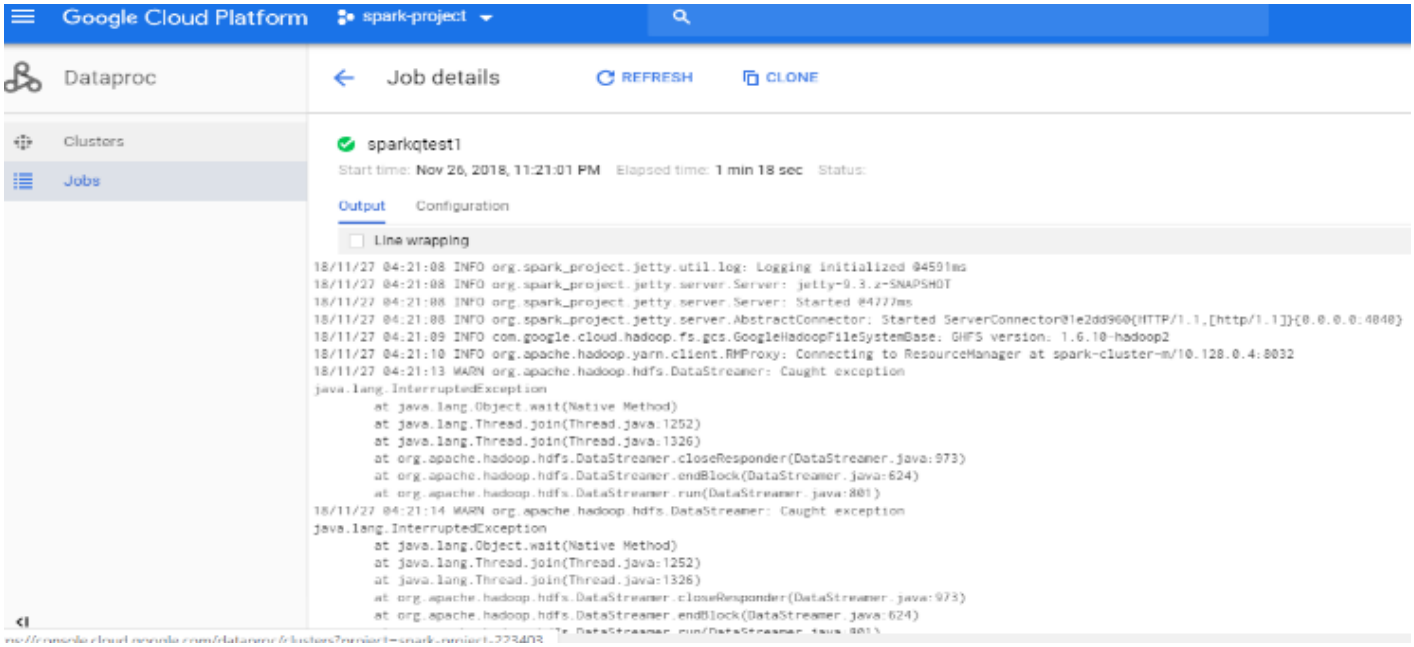

Figure 22: Completion of Spark Job

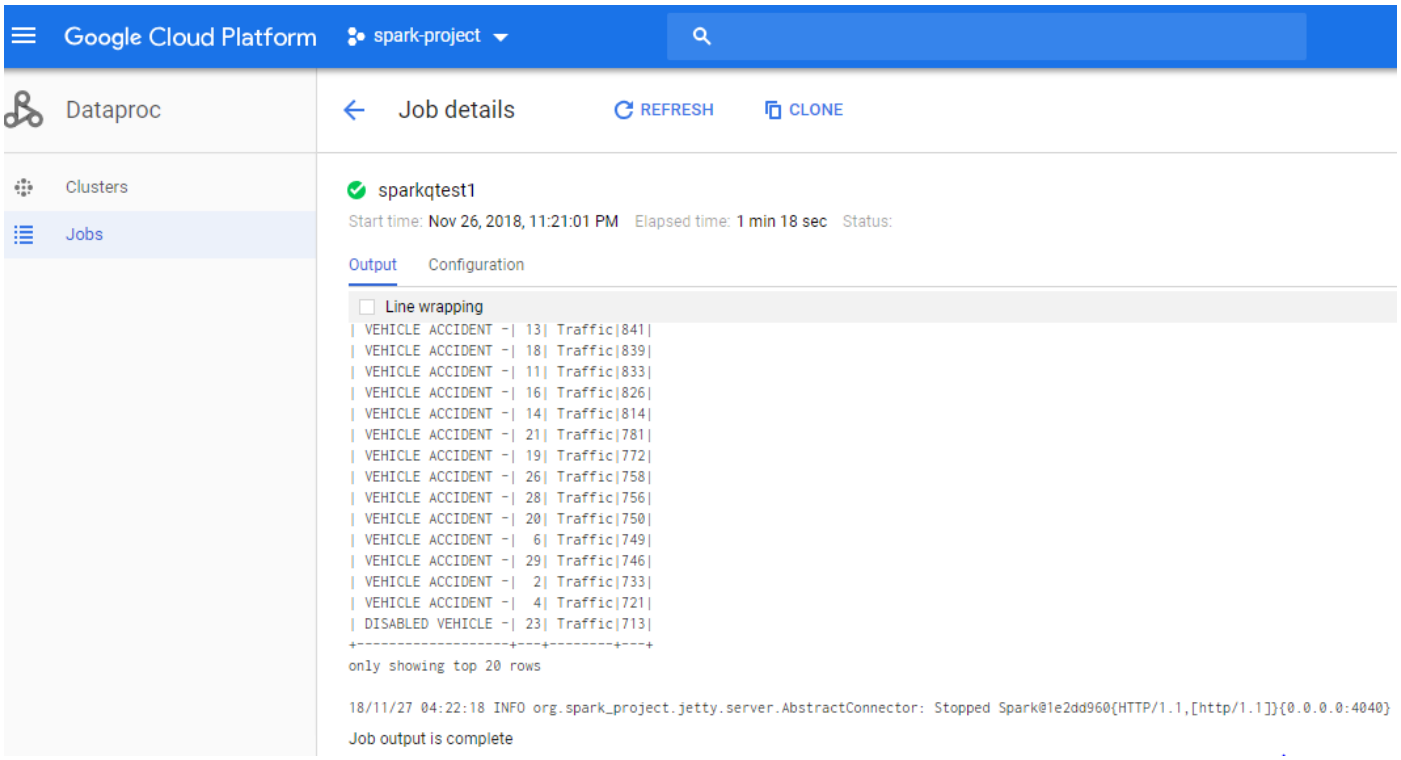

Figure 23: Auto scaling based on CPU utilization

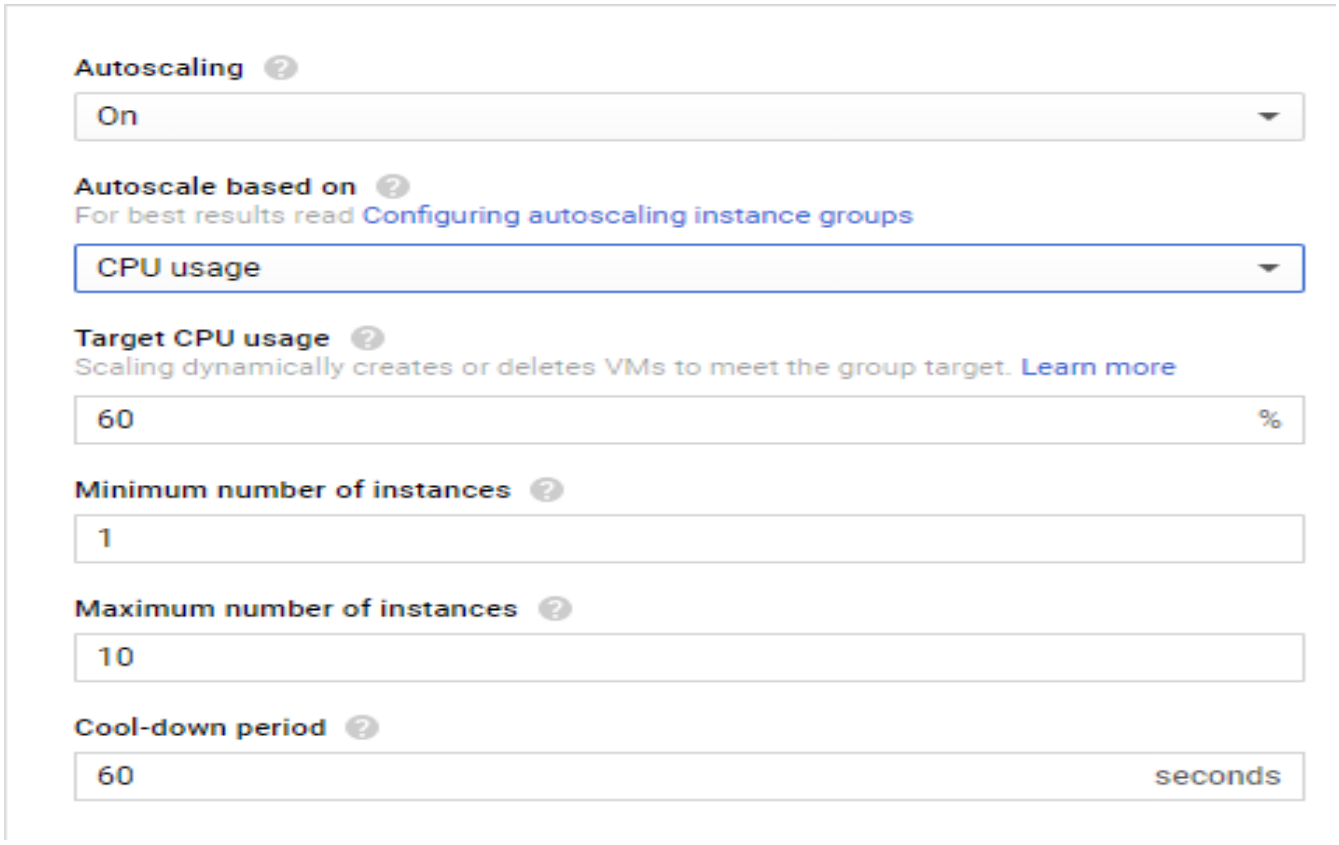

Figure 24 : Autoscaling based on HTTP load

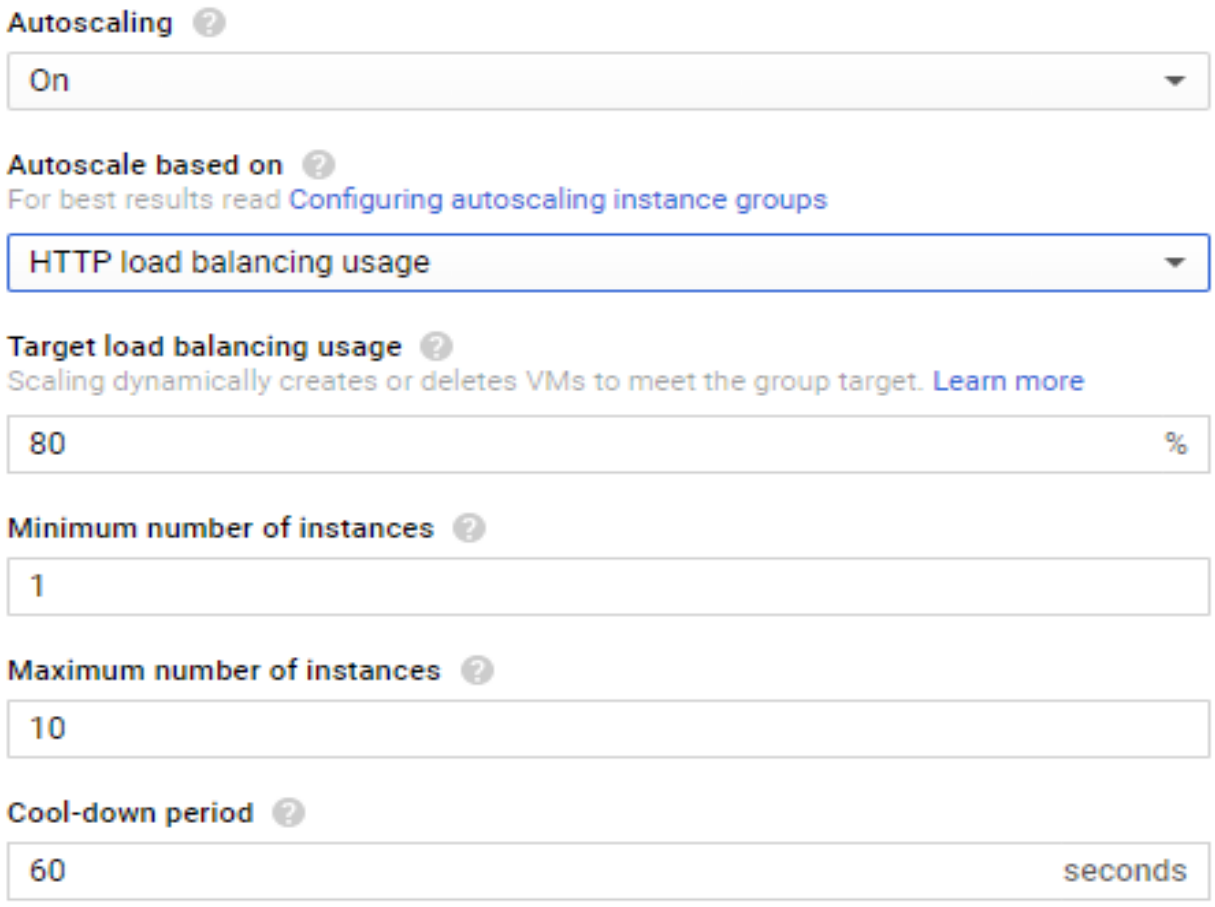

Figure 25 : Disk I/O Memory usage

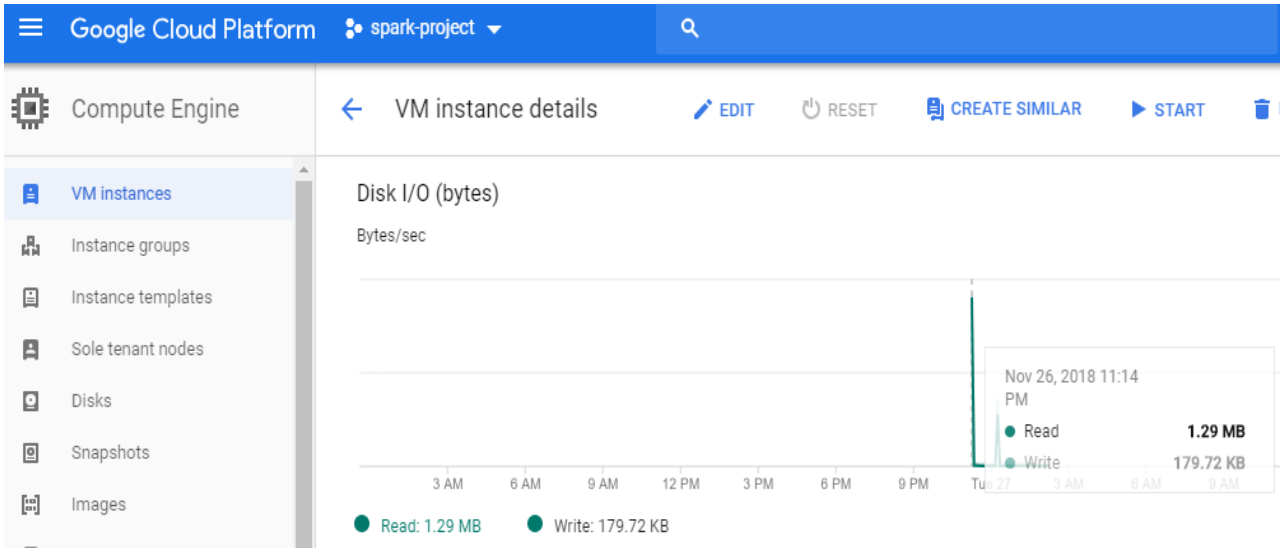

Figure 26 : Disk I/O operations