Kriti

by Gaurav Raj

Submission date: 11-Apr-2018 03:13PM (UTC+0530)
Submission ID: 556160803
File name: Kriti_report_plagcheck.docx (1.82M)
Word count: 2424
Character count: 12634

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

The term 'cloud computing' originated in 1996. This technology was boosted by amazon.com when it released its product elastic cloud computing (EC2) in 2006. Since then, this technology has demand in almost every sector of IT industry, especially in management and analysis of on demand resources.

Cloud computing is a practice of deploying several virtual machines on the Internet so as to store, manage, analyze and process data anywhere around the world. It performs the delivery of pay per use of computing resources like, networking, database, software, storage, analytics, and much more resources online.

Cloud computing provides three types of services:

1. IaaS (Infrastructure as a Services): Vendors use this service to provide users with resources related to computing like servers, network and storage.
2. SaaS (Software as a Service): This makes use of internet in order to provide application to users that are being controlled by third party.
3. PaaS (Platform as a Service): A structure is provided to the developers in which they can build and use it to create

These services help in increasing elasticity, scalability and availability, load balancing in cloud computing.

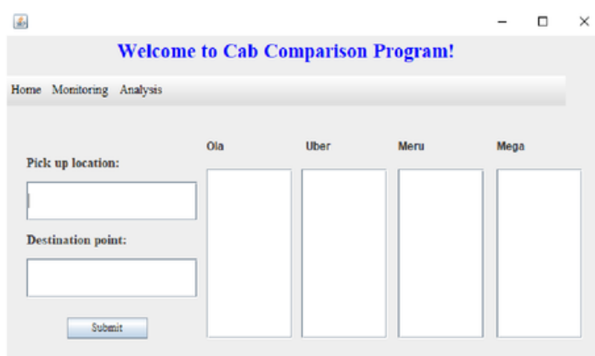

Figure 1: GUI Framework of the proposed model

1.2 APPLICATIONS OF WEB SERVICES

Web Services find a great variety of application domain like e-businesses, banking &finance etc. Coming under the purview of service-oriented architecture (SOA);

1.3 QOS OF WEB SERVICES

The quality of web service is highly competent. It is functional at all times. The quality is maintained by AWS standards. The web service will be non-functional only when the client does not have internet facilities near its vicinity. It is highly reliable model.

1.4 Purpose of Project

The purpose of this project is to deploy a web service in AWS and analyse the real-time results. The proposed mathematical model helps in predicting the best cab service to the customer based on its availability, cost and response time. The cost estimation calculators of four Indian based cab services OLA, UBER, MERU and MEGA are deployed on the AWS Elastic Beanstalk machine. The results are much precise and accurate as compared to existing methodologies.

1.5 Scope/ Phases of Project

The aim of this project is to develop a web service which is easily accessible to everyone. This web service helps in comparing the real time cab service providers and finding the best suitable service for the customer using Google maps for calculating distance and time.

1.6 Project Constraint

The main project constraints are defined as follows:

i. Risk

The risk in this project involves the policy management of cloud resources. This issue can be faced by careful usage of cloud policies and its algorithms.

ii. Resources

There is no constraint in resources as there are sufficient applications available for its implementation. All the applications and interfaces are easily available on the Internet. The team required for the project has sufficient knowledge of programing and cloud computing, therefore, no constraint is faced in this respect.

iii. Quality

The quality of the outcome of the project should be as desired. If the quality is not good the it can be worked upon by improving the minor errors in the program.

iv. Scope

The scope the project is to attain a task scheduling algorithm that is better than the exiting algorithms and gives a better performance in managing the load at risk situation.

v. Time

The time required to develop the complete project would be nearly 2 to 3 months. Within this time period the complete project interface can be easily developed, and results will be analyzed.

vi. Cost

The cost for this project is not a major factor because almost all the resources are free and only a tad amount of money which sums to about $0.062 per Hour utilization of EC2 resources.

1.7 Project Timeline

Table 1: Project timeline table

S. No.	Project Phase	Duration (in days)	Start Date	End Date
1.	Requirement Analysis	03	18/12/2017	21/12/2017
2.	Feasibility Study	03	21/12/2017	24/12/2017
3.	Project Design	03	24/12/2017	27/12/2017
4.	Coding	84	27/12/2017	28/03/2018
5.	Testing	10	28/03/2018	07/04/2018
6.	Implementation	05	07/04/2018	12/04/2018
7.	Delivery	01	12/04/2018	06/04/2018

1.8 Project Risk

Table 2: Risk table for the project

Risk	**Risk Level L/M/H**	**Likelihood of Event**
Project Size	**M**	**Less likely**
Identification of the right tools based on the criteria of effectiveness, low-time consumption	**L**	**Highly likely**
Inherent errors in model and in its application	**M**	**Moderately Likely**
Data Mining Work	**L**	**Less likely**

CHAPTER 2: LITERATURE REVIEW

2.1 WEB SERVICE MONITORING TOOLS

Amazon Web Services (AWS) provide a plethora of ways to monitor the web services. In this project the web services are deployed on the AWS and it is used for monitoring the database.

2.2 Literature Survey of Existing Research

As data is increasing rapidly the necessity to manage and analyze such large volume of data is also increasing. Cloud computing has been observed as an ultimate approach to solve the problem of analyzing the increasing data globally. Kumbhare et al. [1] have proposed a model which displays the concept of "dynamic dataflows". They have proposed two greedy heuristics algorithm: shared and centralized which depends upon bin packing algorithm with size of a variable, then draw its comparison against the Genetic Algorithm which gives a near optimal solution. Thus, by alternate implementation of dataflow tasks users gain flexibility in areas of application composition and provide an additional dimension to the scheduler in order to overcome the constraints of the application. [2] He et al. have adopted the Linear Temporal Logic (LTL) to consider the multiple competing challenges faced at datacenters. Their paper deals with job scheduling of 200 MapReduce jobs on the AWS cloud. The results prove that their method is much better in balancing the multiple conflicting objectives as compared to the traditional methods.

Yassin et al. discuss about the security issues faced by the users of cloud [3]. As a client uses pay as you go service provided by the SaaS (Software as a Service) provider, the user must give their information to the internet which makes them vulnerable to threat. One such threat presented in their paper is about SQL Injection which paves way for the attackers to violate the confidentiality, availability and vulnerability of the cloud users. Therefore, the authors proposed a detection framework solution to resolve the security issues related to SaaS customers.

Resource scheduling is among one of the critical problems in the areas of cloud computing. [4] Lee et al. have presented DeepSpotCloud in their paper. It is implemented with the help of AWS cloud computing services in order to serve real deep learning tasks. Their proposed model of Billing Policy – Hourly migration heuristic achieves 13% more cost gain as compared to interrupt driven scheduling policy. Another scheduling method that is PLAStiCC: Predictive Look-Ahead Scheduling [5] has been proposed by Kumbhare et al. They have proposed a solution to the excessive stream of continuous flowing data through scheduling strategies. With the help of PLAStiCC methodology which is a predictive based lookahead model they could depict an improvement of 20% in the comprehensive profit in comparison with the reactive adaptation algorithm.

[6] Zinno et al. have proposed a solution to increase the efficiency of Parallel Small Baseline Subset(P-SBAS) algorithm that is nothing but advanced Differential Interferometric Synthetic Aperture Radar (DInSAR) technique. They automated the technique of generation of Earth surface

displacement time series by using cloud computing techniues. The results prove that with the aid of cloud computing and appropriate scheduling of parallel jobs they were able to achieve DInSAR results on a large scale of 150,000 km2 in a very short span of time of about 9 hours. Another research carried out by Khodadadi et al. [7] shows an application of cloud computing in the field of Internet of Things (IoT). In their paper they have proposed a framework taking data as the central element through which IoT applications such as sensors can communicate with each other with the help of a prototype which is executed on AWS that is placed on top of the Aneka Cloud Application Platform.

For the purpose of management of virtual machines (VMs) in AWS EC2 public cloud Grimaldi et al. have suggested a feedback mechanism [8] in which the policy of gain scheduling is evaluated at different workloads and it is compared to the robustness of algorithm at the time of VM failures. Thus, the results show high performance at the time of both constant and time-varying workloads. Arabnejad et al. proposed a Deadline Distribution Ratio (DDR) scheduling algorithm [9] for minimizing the cost across majority of deadlines along with maintenance of a high scheduling success rate. The results show that their methodology has successful outcome in yielding the lowest cost. CloudAnalyst [10] is a tool based on CloudSim which is adopted for analysis and modelling of computing environments at massive scale. Wickremasinghe et al. have proposed CloudAnalyst in their paper and have discussed the various features of the simulator. It can implement and showing results of large scale cloud computing environment but is incapable of taking real-time data centers as inputs. Therefore, this paper proposes an algorithm that is capable of taking input of real-time data sets and then implementing task scheduling on it.

CHAPTER 3: DESIGN AND METHODOLOGY

This project proposes an algorithm which can take real time data sets as input and processing it to give an analysis of the data through a simulator. Amazon Web Services (AWS) provides the platform to host website and private cloud on virtual machines with the help of instances. Services in Amazon Web Services consists of EC2 (Elastic Compute Cloud), S3 (Simple Storage Service), etc. For the same, firstly an account is made in Amazon Web Services and then EC2 instance is launched. The following steps are performed in the deploying of webservices and predicting the best cab:

Step1: Take Input from the user about source and destination.

Step 2: Calculate estimated distance and time from Google API

Step 3: Select Elastic Beanstalk service from AWS console

Step 4: select create new application

Step 5: Enter application name and description

Step 6: Select ‘create web server’ option.

Step 7: Select ‘Tomcat’ in predefined configuration and click on next.

Step 8: select source of application version and click next

Step 9: Fill in environment information, ‘environment name’ and ‘environment URL’. Click on Next.

Step 10: do not select any option from additional resources. Click next.

Step 11: Review Configuration details and click next.

Step 12: Leave environment tags as it is and click next.

Step 13: Manage permissions and then click next

Step 14: Review all the details and click launch.

Step 15: After uploading the ‘.war’ file, wait till the health status reaches to ok.

Step 16: review the monitoring status.

Step 17: Use the estimate distance and time to calculate the cab fares.

Step 18: The prediction is done based on the results.

3.1 WEB SERVICE MONITORING TOOL DESIGN

The proposed model calculates the cost, availability and response time and then compares it with a service prediction model to propose the best cab service that should be taken by the customer. The workflow of the proposed methodology is diagrammatically represented by the following figure:

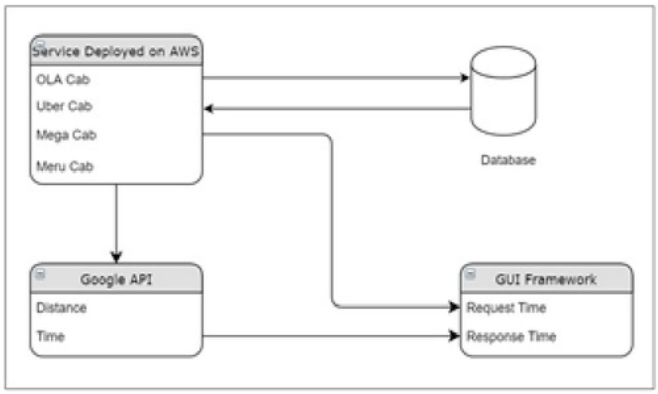

Figure 2: Architectural Design of Web Service Monitoring Tool

3.1.2. SOFTWARE / HARDWARE REQUIREMENTS:

- Amazon Web Service (AWS)
- OS: Windows 8.1 and above, 64-bit
- Web Browser: Google Chrome (Version 57), Mozilla Firefox
- Java Eclipse

3.2 WEB SERVICES CLUSTERING PROCESS DESIGN

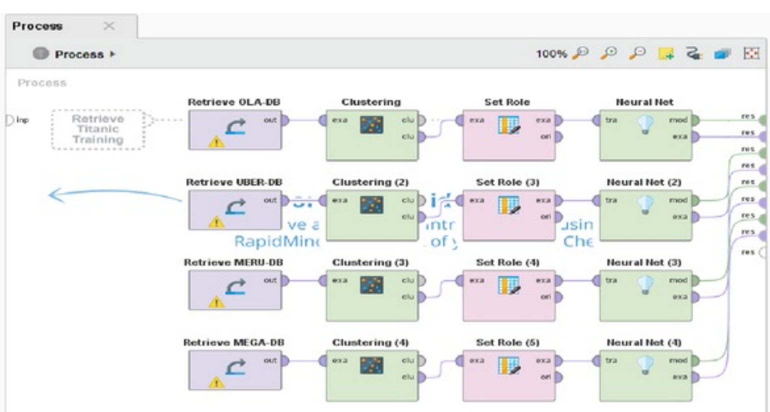

Figure 3: Clustering process on RapidMiner

The clustering process involves k-means and is then followed by prediction of web service using ANN (Artificial Neural Networks).

3.3 WEB SERVICES SELECTION PROCESS DESIGN

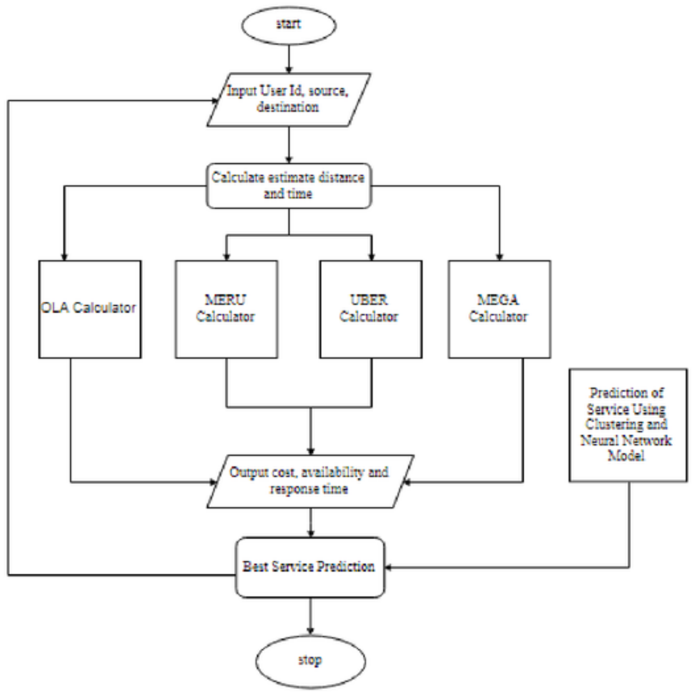

Figure 4: Web Services selection algorithm

CHAPTER 4: EXPERIMENTS AND RESULTS

4.1 SCREENSHOTS OF WEB SERVICE MONITORING TOOL

The results of the proposed methodology are shown by the following figures. The webservice is successfully made and deployed on AWS which is easily accessible from anywhere.

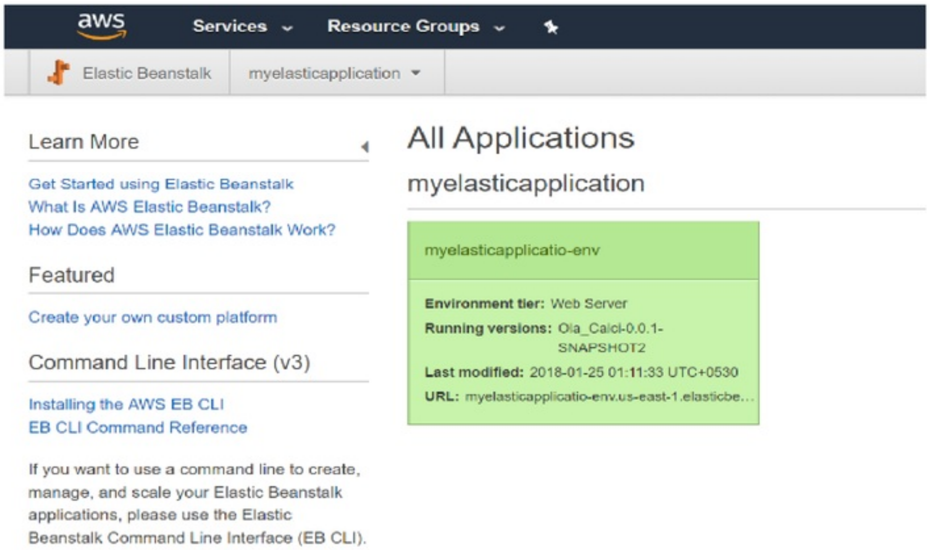

Figure 5: Deployment of Application

Once the application is deployed the successful status is shown by the following figure.

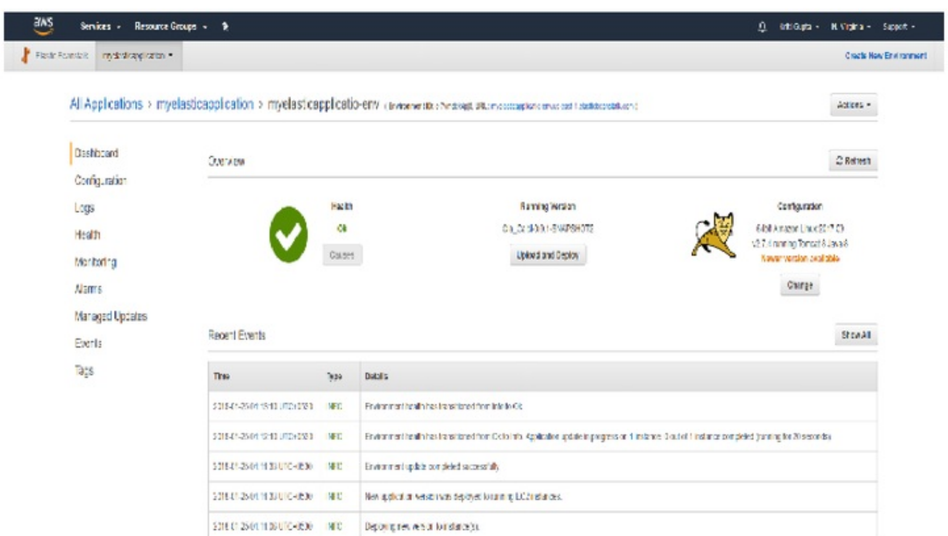

Figure 6: Status of the application deployed

The working model of the calculator is shown as below from where the outputs are gathered.

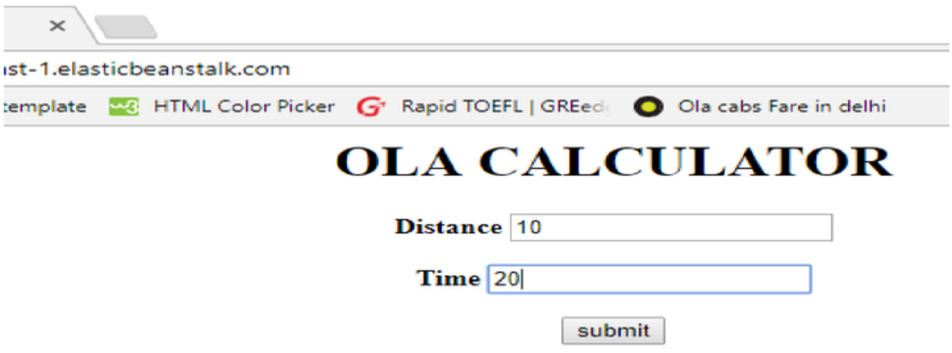

Figure 7: Ola Calculator using distance and time as input

Following figure shows the output which is further used as input to calculate estimated time and distance using Google API.

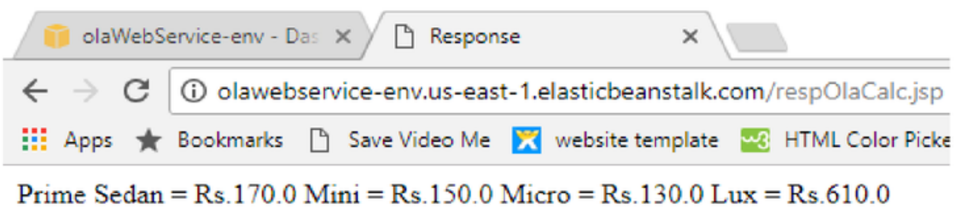

Figure 8: Output from OLA calculator

The analysis and monitoring can be carried out with the AWS analysis program. It shows response time, number of requests and various parametres that can be used for load balancing, increasing scalability and increasing elasticity of the model.

4.2 RESULTS OF WEB SERVICES MONITORING TOOL

These results help in generating the availability, cost and response time of the services and with the help of a prediction model it can predict the best service to the customer.

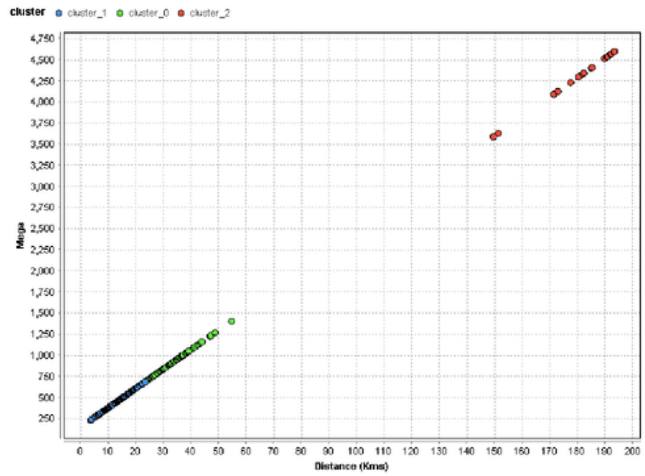

Figure 9: mega k-means

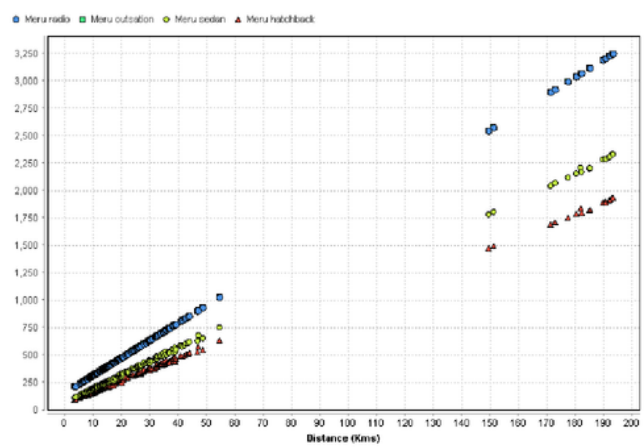

Figure 10: meru k- means

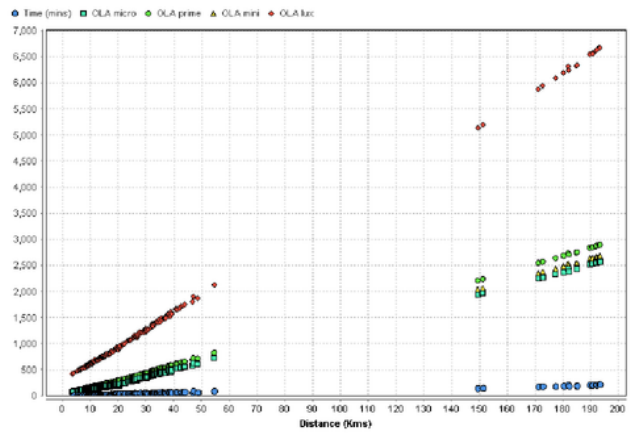

Figure 11: ola k-means

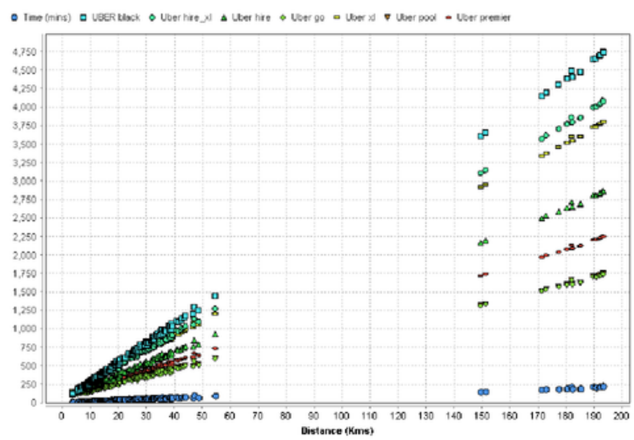

Figure 12: uber k-means

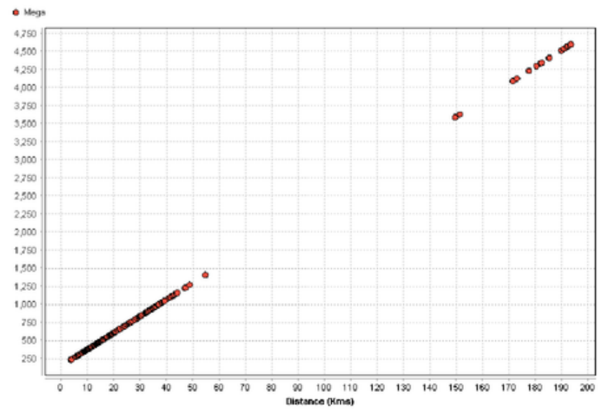

Figure 13: mega svc

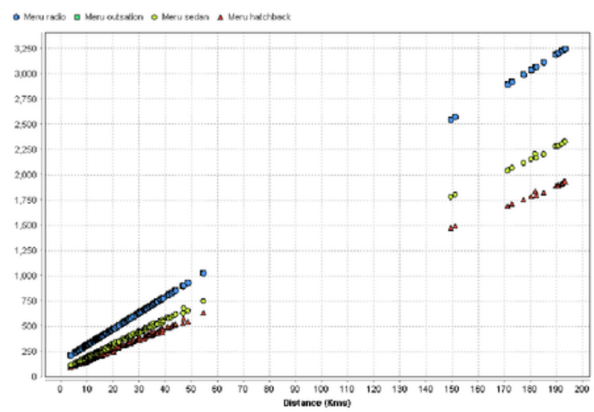

Figure 14: meru svc

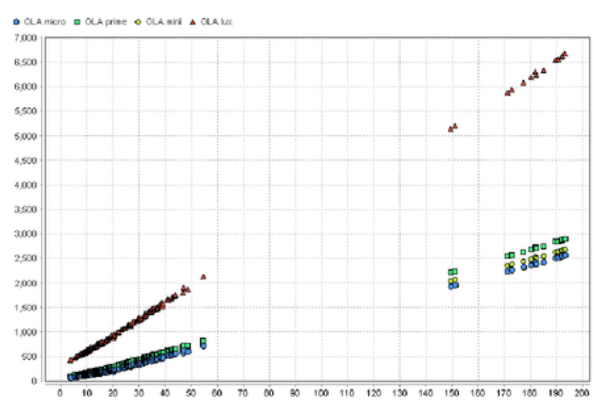

Figure 15: ola svc

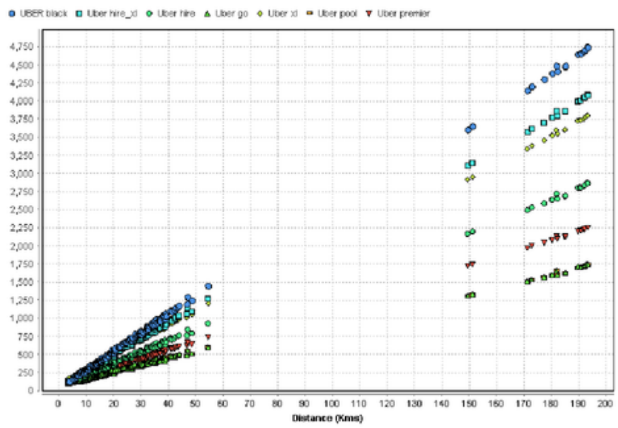

Figure 16: uber svc

4.4 SUMMARY OF RESULTS OF CLUSTERING

The clustering process yielded three clusters as per the k-means (k=3) clustering algorithm. A total of 2507 services have been successfully clustered as per this process into 3 different clusters. It is noteworthy to observe that there is an underlying pattern in the grouping of these web services into these three meaningful

clusters.

Index	Nominal value	Absolute count	Fraction
1	cluster_0	111	0.550
2	cluster_1	71	0.351
3	cluster_2	20	0.099

Figure 17: Cluster formation of services in AWS

These results help in generating the availability, cost and response time of the services and with the help of a prediction model it can predict the best service to the customer.

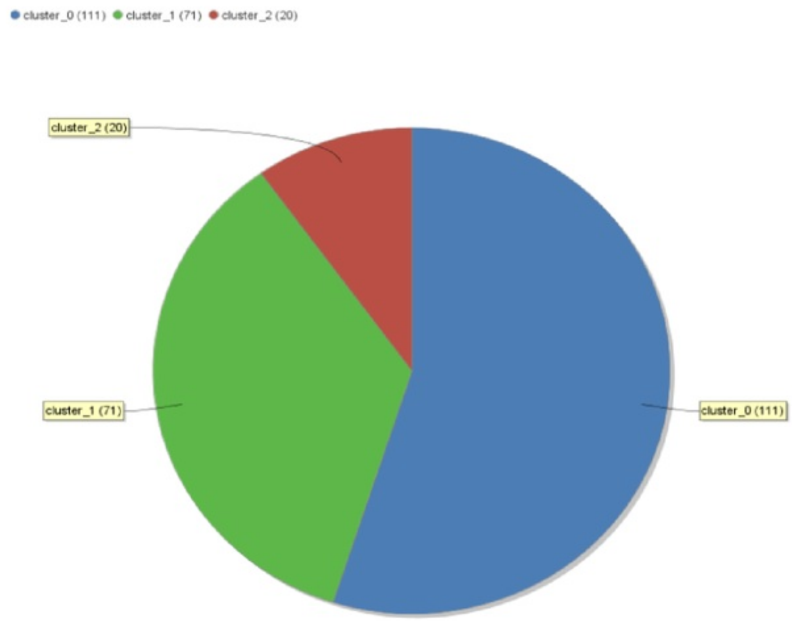

Figure 18: Distribution of the web services across three clusters

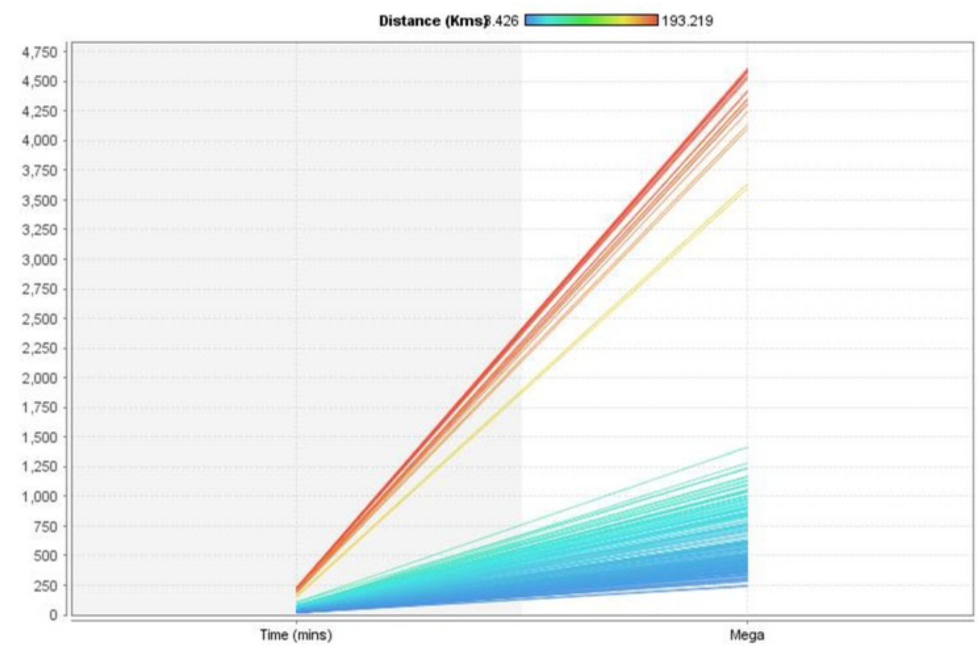

Figure 19: Plot of the cluster centroids for each cluster

4.5 SUMMARY OF RESULTS OF WEB SERVICE SELECTION

The results above show the cluster formation from the data input of the 4-web services OLA, UBER, MEGA and MERU. The clusters and the plot of the clusters has been depicted by the above figures. It shows that the for different time zones and distances which cab would be more suitable and predicts which cab should be taken at a certain moment.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

The methodology in the project is highly competent and precise. It provides faster and better results as compared to existing methodologies. It can provide best service with inputs from any predictive model. In future, work can be done to improve the model with the help of increase in the number of policies and the load applied on the system.

Kriti

ORIGINALITY REPORT

2%	2%	1%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ieeexplore.ieee.org Internet Source	1%
2	ivythesis.typepad.com Internet Source	1%
3	www.takadu.com Internet Source	<1%
4	Submitted to 87988 Student Paper	<1%

Exclude quotes	Off	Exclude matches	< 4 words
Exclude bibliography	Off		