National College of Ireland

A Novel Cloud-Based Task Scheduling in Fog Computing by Using Heuristic Approach

M.Sc Research Project
Cloud Computing

Rajesh Kavuri
Student ID: X17108250

School of Computing
National College of Ireland

Supervisor: Mr.Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Rajesh Kavuri
Student ID:	X17108250
Programme:	Cloud Computing
Year:	2019
Module:	M.Sc Research Project
Supervisor:	Mr.Vikas Sahni
Submission Due Date:	18/04/2019
Project Title:	A Novel Cloud-Based Task Scheduling in Fog Computing by Using Heuristic Approach
Word Count:	XXX
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	18th May 2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Novel Cloud-Based Task Scheduling in Fog Computing by Using Heuristic Approach

Rajesh Kavuri
X17108250

Abstract

With the rapid development of the Internet of Things (IoT), user load is increasing day by day on Cloud infrastructure. Fog computing similar to Edge computing, was introduced between Client and Cloud layer. Fog Computing is attractive due to its real-time services, reduction of transmission delay, response time and processing times. To support such computing there will be a need for very efficient task scheduling like Prioritized scheduling algorithm. This paper presents a Priority-based task scheduling for computing by carrying different parameters like response time, Overall cost and transfer cost. Performance evaluation of Prioritized scheduling shows it performs much efficiently than other algorithms by reducing the response time by 83.46% and Virtual machine cost by 46%.

Keywords: Internet of Things, Task scheduling, Fog Computing, Prioritized Algorithm, Cloud-analyst Simulator

1 Introduction

Cloud computing provides us a means for uploading data and use applications over the internet. Cloud computing provides three different service models which are - Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). These services are offered a pay-per-use model. Cloud computing provides many features like scalability, flexibility, performance-cost efficiency, ease of test and deploying new technologies. Choudhari et al. (2018) Even with these features cloud has few drawbacks like Cloud and users are physically apart that cause processing delays and shortage of resources while executing the tasks. Fog computing termed as edge computing came into existence as an extension to Cloud Computing. It works as a virtualized layer between cloud and end user. It became more important with the increase in Internet of Things system. Atlam et al. (2018) With the exponential increase in data at the nodes, fog computing is in demand. These data can be from smart cities, navigation devices, Sensors, and Mobile devices .

The main purpose of fog computing is to reduce the data load on Cloud,increase efficiency and performance by offering Edge Computing. Data is stored temporarily in the fog sent by cloud for further usage to avoid late response and network traffic.

Task scheduling is one of the essential technique in Cloud Computing for assigning tasks properly to the resource and utilizing the Overall system performance. Here task scheduling refers to a set of techniques Saleh.H et al. (2019) used to assign tasks to the Virtual

machines. The main aim of task scheduling mechanism is to increase resource utilization without affecting Quality of Services (QoS).

1.1 Motivation and Background

According to IHS Analysis number of Connected IoT devices will surge to 125 Billion by 2030 with a jump of 12 percent annually with an average of 27 Billion in 2017 to 125 Billion in 2018. There will be constant interaction of human with machines, generating humongous data. Howell (2017) The four Cs of IoT are Create, Collect, Compute and Connect offer a pathway to create and manage IoT Revolution
However data generated by IoT sensors will be huge called a Big Data. K.Naha et al. (2018) These problems can be overcome by using Fog computing paradigm at the edge layer which harnesses the processing of the data, storage, and Networking at the edge.

1.2 Research Question

The current issues in Internet of things are response time., Data transfer cost and Mobility and Latency. To address these issues the following Research question is specified:
RQ:*"Can Prioritized task scheduling algorithm in fog computing help to reduce response time,data transfer cost and Data center processing time by using Heuristic approach?"*

1.3 Research Objectives and contributions

The major Objective in this research paper are Designing, Developing and Implementing Prioritized Scheduling algorithm by Heuristic Approach The major contribution from this objective is Prioritized Task Scheduling algorithm.
The minor Objective in this research paper is Developing a fog computing Environment to perform the test the Proposed Prioritized Scheduling algorithm The Minor contribution from the above objective is assigning the tasks to the Scheduling algorithm.
The final objective in the research paper is Getting the results by assigning the tasks The Final contribution in this paper is Examining and Analyzing the results and proving the Prioritized Scheduling algorithm is successful in Test.
Further, This paper is structured as follows:
Chapter 2 presents the reason for proposing prioritized scheduling algorithm by examining different scheduling algorithms, Heuristic approaches and Review about Cloud Analyst. Chapter 3 describes the methods applied and the algorithm. Chapter 4 explains about the architectural design of the model. Chapter 5 explains about the Implementation strategies and Cases to Implement. Chapter 6 determines the results and make a comparison of the cases. Chapter 7 concludes the research paper.

2 Related Work

Task Scheduling in cloud computing mainly focuses on assigning the requested tasks to the available resource is the key factor here. The main characteristics of the task scheduling will be response time, Execution time and data size of communication between

tasks. Task scheduling in Fog computing is a vital issue in IoT devices. Fog layer serves between IoT devices and Cloud data center.

2.1 A Critical Review of Task Scheduling

2.2 Task Scheduling

Task scheduling algorithms are categorized into Deterministic and metaheuristic algorithm. Deterministic algorithms include Round Robin algorithm, Shortest job First and Load balance over Slow Resource. These algorithms cannot find an optimal solution in finding a reasonable time when complexity occurs. Metaheuristic algorithm includes Hill climbing, Ant Colony optimization, Cuckoo search, and particle swarm optimization.Saleh.H et al. (2019) These techniques are helpful when the solution is too costly to obtain using Deterministic algorithms.
The idea of Fog computing is to bring the cloud services to the IoT devices which generate data. Fog nodes will process the data without forcing the data on the Cloud. Adding Fog layer between Client and Cloud layer will perform low latency while large scale applications.Pham.X and Huh.E (2016) In Addition to this Cloud Scaling will help to provide the services while fog resource busy on peak demands. Many applications need interchange of data between fog and cloud. From this view Fog computing aimed to satisfy the increasing need for services demanded by users.

2.2.1 Task Scheduling in Fog Computing

(Bonomi et al.; 2012) explained about the importance of fog computing in IoT bu using different characteristics like a large number of nodes, mobility, location awareness, the role of wireless access and presence of streaming and real-time applications. Authors finally concluded that fog is the rightful platform for IoT services and applications like a connected vehicle, smart cities, wireless sensors, smart grid and Actuators Networks(WSANS).
Olaniyan.R and Maheswaran.M (2018) proposed Synchronous Scheduling algorithm based on comparison results between three algorithms. static, Dynamic and synchronous scheduling algorithm.While performing the test on simulator they finally concluded that the Synchronous scheduling algorithm is best for the internet of things.
Islam.T and Hashem.M (2018) proposed fork-Fulkerson scheduling algorithm for edge computing networks for fast processing of a large amount of big data. Here authors mainly considered latency of delicate applications from the fog layer.Finally, the authors concluded that the proposed algorithm was good in dealing with Big data.
Mohamed et al. (2017) proposed a UAV(Universal aerial vehicle) technology based fog computing for the Internet of things.UAVFog enables fast deployment of fog capabilities.UAV provides fast, safe deployment of civil, military actions and cost-effective.Authors have concluded that UAVFog is capable of achieving many positive results.
Krishnan et al. (2015) proposed fog computing and network-based cloud computing move the process from cloud to network devices present along with the node with the cloud. They conclude that the proposed method will give better response time.
Abreu et al. (2018) proposed Rank scheduling for fog computing environments for Low latency and move the services to the network. Even Fog computing provides mobility support, Geographical distribution. Authors concluded that the Proposed algorithm was successful in most simulated results than Round robin scheduling algorithm.

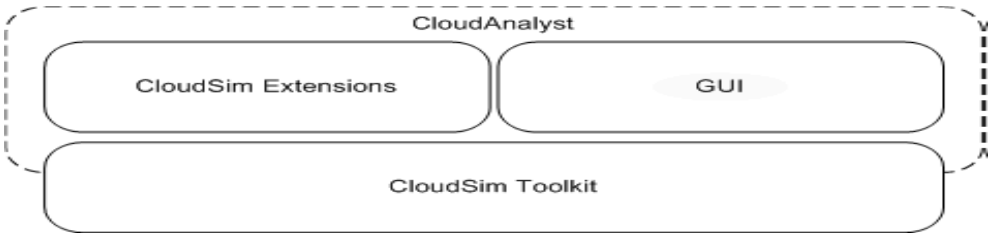

Figure 1: Cloud Analyst (Wickremasinghe et al. (2010))

2.2.2 Task Scheduling By using Heuristic approaches

Mohammed.D and Nawwaf.K (2011) proposed a hyper heuristic-genetic algorithm for task scheduling for heterogeneous networks which is a combination of heuristic GA and Dynamic level scheduling (DLS). After deep evaluation of results, the authors concluded proposed Hyper-Heuristic algorithm is successful in achieving good results in Every case in two-phased H2GS.
Rahbari et al. (2017) proposed Hyper-Heuristic algorithm in fog computing by Secure aware Scheduling for the Internet of things to Low latency and energy consumption. Authors considered three parameters they are integrity, authentication, and Confidentiality for maintaining security in fog devices.
Choudhari et al. (2018) proposed a priority based task scheduling for fog computing. The proposed algorithm contain queuing, Priority models and priority based assignment module. Authors conclude that the proposed algorithm was successful in reducing overall response time and total cost while comparing with existing task scheduling algorithms. The proposed algorithm is capable of prioritizing tasks based on their task delays.
Rashid.S and Sharifian.S (2017) proposed heuristic Queue based task scheduling in a mobile cloud environment. The proposed algorithm mainly focuses on reducing offload duration time when balancing the cloudlets load. Successful simulation authors concluded that proposed algorithm succeeded in decreasing mean completion time and total energy consumption of mobile computing devices.

2.3 Simulation tool

Regular advancement of cloud technologies brings developers more pressure to handle the load capacity ,application peak demand and configuration of application. Several cloud providers are offering pay-per-use and flexible elastic infrastructure. However different cloud providers prices varies from geographical distribution which is a new concern in selecting data centers.There is no efficient tool to evaluate the requirements of large scale applications based on Geo graphical distribution of servers and workloads. Wickremasinghe et al. (2010) proposed cloud analyst tool to evaluate the behavior of large scale applications in different various deployment configurations. Figure 2 describes the architecture of cloud analyst. Cloud analyst is a mature simulation frameworks of SimJava and CloudSim.

2.4 Table of comparison

Table 1 shows the comparison of methods used for task scheduling in Fog computing and using Heuristic approaches

Comparison of task scheduling at various levels			
Methods used	**Evaluation Metrices**	**Authors**	**Critic**
Synchronous Scheduling Algorithm	Latency and Geographical Distribution	Olaniyan.R and Maheswaran.M (2018)	Proposed Synchronous Scheduling algorithm for Internet of Things was succeeded in Achieving mobility.
Ford-Fulkerson scheduling algorithm	location awareness and Reducing latency	Islam.T and Hashem.M (2018)	Proposed algorithm was successful in achieving in Location awareness and mobility in real time services.
Rank Scheduling algorithm	Peak time and Heavy transfer of data	Abreu et al. (2018)	Scheduling algorithm was successful in working in Critical time in fog environment.
Prioritized Scheduling algorithm	Response time and Cost	Choudhari et al. (2018)	Algorithm was successful in achieved in getting desirable results but failed to cover latency .
Hyper-Heuristic algorithm	Reduce Offload duration time	Rashid.S and Sharifian.S (2017)	Heuristic approach uses Ant Colony Optimization to minimize mean time for whole system. Their approach finally proves it reduces users battery consumption and Completion time.

Table 1: Comparison of different methods used for task scheduling

2.5 Conclusion

Many task scheduling algorithms have been proposed by many authors in fog computing. These algorithms are useful in Achieving low latency and reduce energy cost. The proposed algorithm includes prioritized scheduling algorithm using a heuristic approach.

3 Methodology

3.1 Cloud Analyst

Cloud Analyst is a tool used to simulate the proposed algorithm by using creating virtual environment.It provides many services like user requirements, VM Management, Cloud services and Load balancing.figure 2 describes the architecture of Cloud analyst simulator.It communicates between hosts, data center, Virtual machines. The major advantages of Cloud analyst are description of geographical distribution of user base and data base and number of users generating traffic, number of resources available. By using above parameters we can generate processing time,response time and other information.Wickremasinghe et al. (2010) By using these results developers can prepare a strategy for allocating resources among available resources in the data center and reducing costs related to data flow.
The main key components of the Cloud Analyst are Cloudlets ,Datacenter Characteristics, Fog data center broker,Host, Network Topology and Virtual machine characteristics.

1. Cloudlets:Cloudlet is an extension of the gridlet . It stores all the information of VM.The following are the parameters of Cloud analyst :
    - Cloudlet length
    - Input
    - Output
2. Data center Characteristics:It will stores all the information about the resource in the data center. VMscheduler and VMProvisioning will take care processing of Gridlets and processing of Vitual machines.
3. Fog data center: It acts as a broker between user and provider by storing information about the user for providing better service in fog layer..
4. Host:Hosts allocates the resources to the user like Memory, CPU and VM.
5. Network Topology contains information of network topology used to simulate the latency in traffic flow in Cloud analyst.
6. Virtual machine Characteristics:This class is responsible for assigning available resources for sharing to VM.

Priority assignment Algorithm:
Following the design model of queuing and priority model described above, the priority assignment is presented:

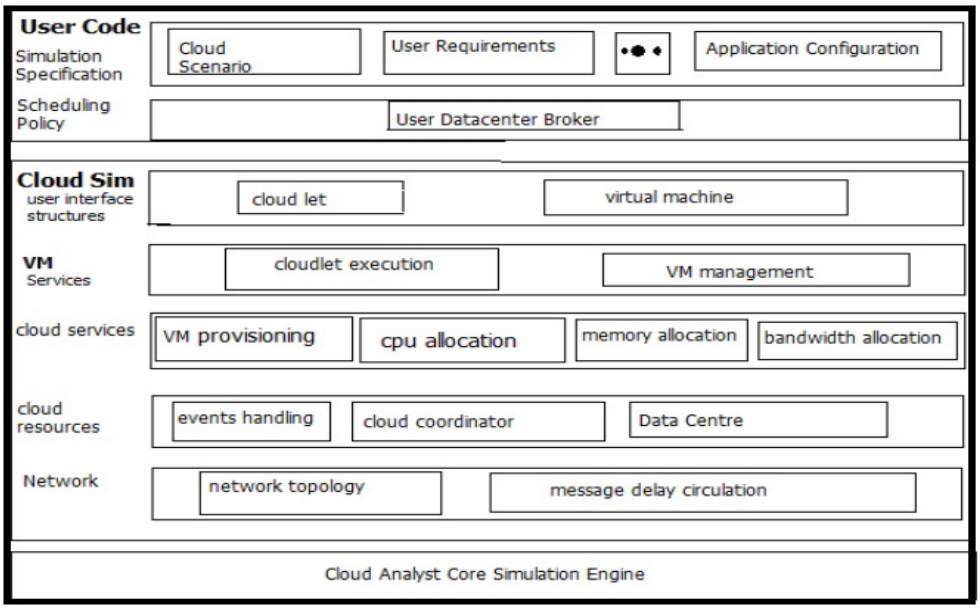

Figure 2: Cloud Analyst architecture Talasila et al. (2016)

$Priority(delay_i^T, ST_est, i)$
1.Maximum allowed delay no greater than estimated service time.
$if delay_i^T == ST_est, i$
Return(H) 2.Maximum allowed delay between threshold T1 and T2
else if$T_1 < delay_i^T <= T_2$
if $SB_CAT == 1$
Return (H)
else if $SB == 2$
Return (M)
else if $SB_CAT == 3$
Return(L)
3.Maximum Allowed delay greater than Threshold T_2
if $SB_CAT == 1$
if $Q_H isnot full$
Return (H)
else
Return(M) else if$SB_CAT == 2$
if $Q_M isnot full$
Return(M)
else
Return(L)
else if $SB_CAT == 3$
Return(L)

(Source:Kavuri (2018))

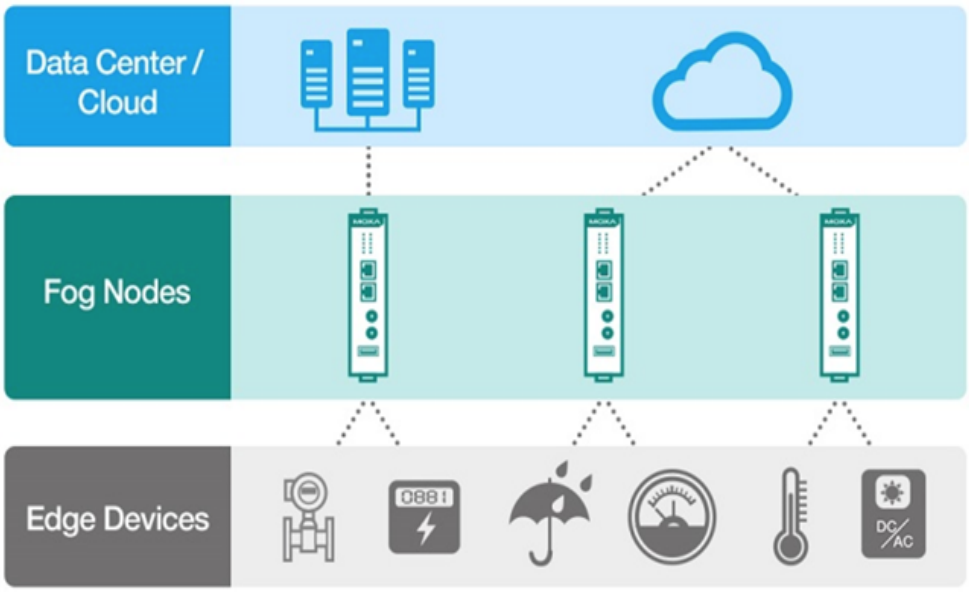

Figure 3: Fog computing Moxa (2018)

3.2 Creating a Fog environment

The research paper focuses on implementing the prioritized algorithm in fog computing for completing tasks more efficiently. The research paper tries to replicate the work proposed by Choudhari et al. (2018). The research paper includes reducing overall response time and waiting time of tasks in IoT by creating fog environment figure 2 describes the configuration of data center required in the simulation.figure 3 explains about the boundaries of regions of the user base and their location.

Data center configuration	
Parameters	**Specifications**
Virtual machine monitor	Xen.
Operating system	Linux.
System Architecture	X86.
Virtual Machines	5.
No. of processors	4.
Storage	1000000 GB.
RAM	2048 MB.
Bandwidth speed	1000.

Table 2: Data center configuration

3.3 Conclusion

This section had explained about the methodologies applied and requirements of the simulation. the following section describes the implementation strategies of the research project.

Region configuration	
User base	**Region**
UB 1	0-N.America
UB 2	1-S.America
UB 3	2-Europe
UB 4	3-Asia
UB 5	4-Africa
UB 6	5-Oceania

Table 3: List of regions

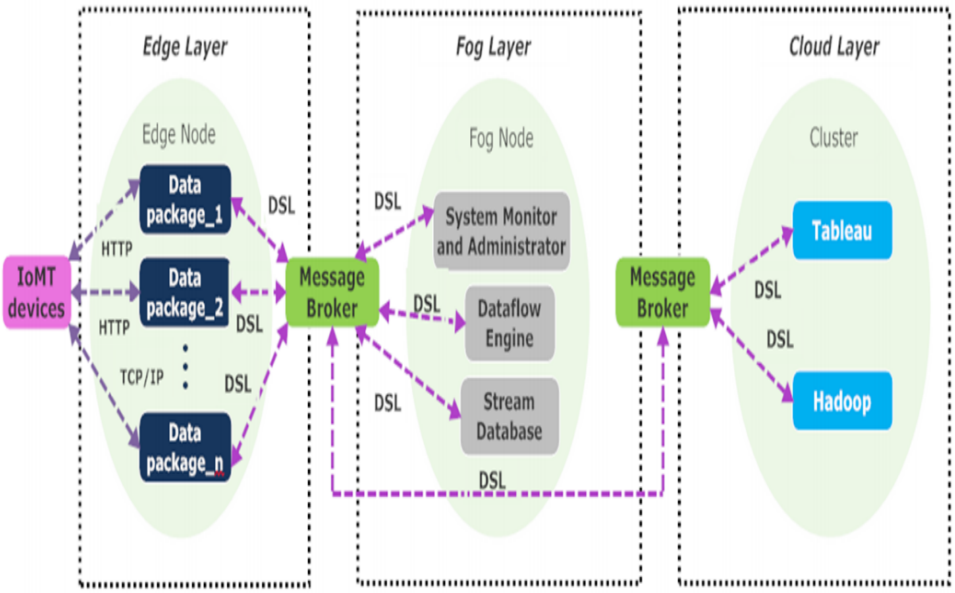

Figure 4: Three tier architecture (Hernandez.L et al. (2017))

4 Design

The research design includes three layers namely Cloud layer, Fog layer, Edge layer.. The proposed paper includes Edge, fog and Cloud Computing for providing means of data life cycle. Fog computing is an extension of Cloud Computing but not replace of Cloud because fog computing process the data stream before sending it to Cloud.Various experiments have been conducted on Fog three layer architecture in smart cities , factories and diary farming shows the optimization of streaming of huge amount of data and cost minimization in Geographically distributed. Figure 4 shows the edge layer which consists of edge nodes which acquires from the IoT devices.Hernandez.L et al. (2017)Fog layer includes fog nodes where streaming of data cycle model is executed. Finally, cloud layer where data center is located.The communication between these three layers are performed by two major components.Message broker and Distributed service links(DSL).

```
public class PrioritizedTaskSchedulingInFog implements CloudAppServiceBroker {
    protected Map<Integer, List<String>> regionalDataCenterIndex = null;

    public PrioritizedTaskSchedulingInFog(){
        regionalDataCenterIndex = new HashMap<Integer, List<String>>();

        init();
    }

    protected void init(){
        List<GeoLocatable> allInternetEntities = InternetCharacteristics.getInstance().getAllEntities();
        int region;

        for (GeoLocatable entity : allInternetEntities){
            if (entity instanceof DataCenterController){
                region = entity.getRegion();
                List<String> l = regionalDataCenterIndex.get(region);
                if (l == null){
                    l = new ArrayList<String>();
                    regionalDataCenterIndex.put(region, l);
                }
                l.add(entity.get_name());
            }
        }
    }
```

Figure 5: Implementing Prioritized Scheduling Algorithm

5 Implementation

5.1 Cloud Analyst Setup

The proposed research paper uses Cloud Analyst version 1.0 beta to test the algorithm and helpful in performing tests on the various algorithm for load balancing, task scheduling, and Monitoring, etc. The major advantage of Cloud Analyst is to get prior knowledge on virtual machines by testing before performing real-time implementation in the Internet of Things(IoT).
The proposed experiment was performed on Intel i5 processor, Windows operating system. The main parameters used in the proposed research project are Cloudlets, Virtual Machine, Host and Datacenter.The parameters for Datacenter are explained in figure 2.

5.2 Experiment 1: Scheduling Tasks using Prioritized Algorithm

This research paper evaluates the technique of "Prioritized Scheduling Algorithm". The aim of this algorithm is to reduce the response time and waiting time of tasks allocated in the Internet of Things. The experiment was conducted on Cloud Analyst 1.0 by creating 1 DC and 5 VM and scalable UB. The prioritized scheduling algorithm will execute the tasks based on their priority level. The following are the results obtained from the experiment.

5.3 Fog Environment setup

This section explains about setting up fog environment in Cloud analyst for making quick processing of requests. It involves communication between the fog layer and cloud layer. Here cloud layer becomes a data center to provide the required resource and computational resources. In this step, we create only one Data Base in Region 0 and

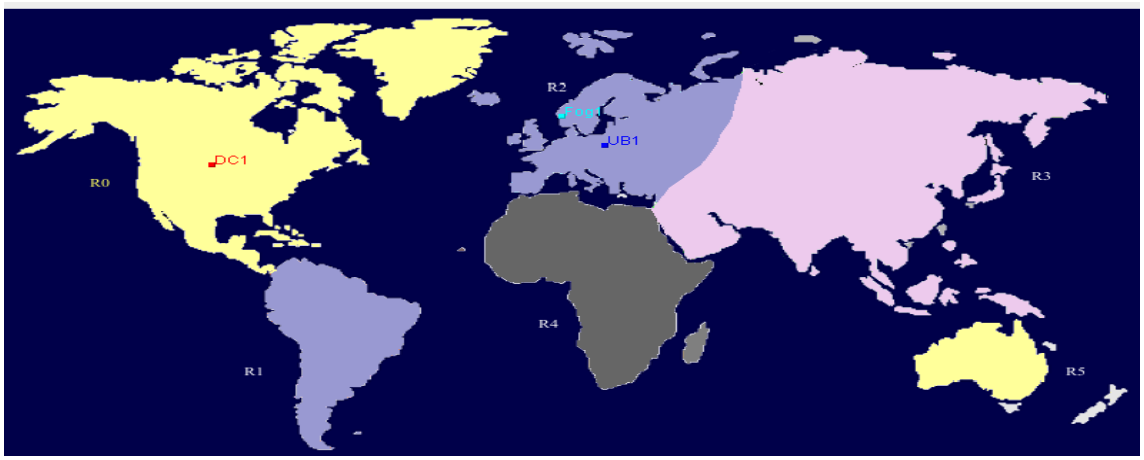

Figure 6: Region Boundaries

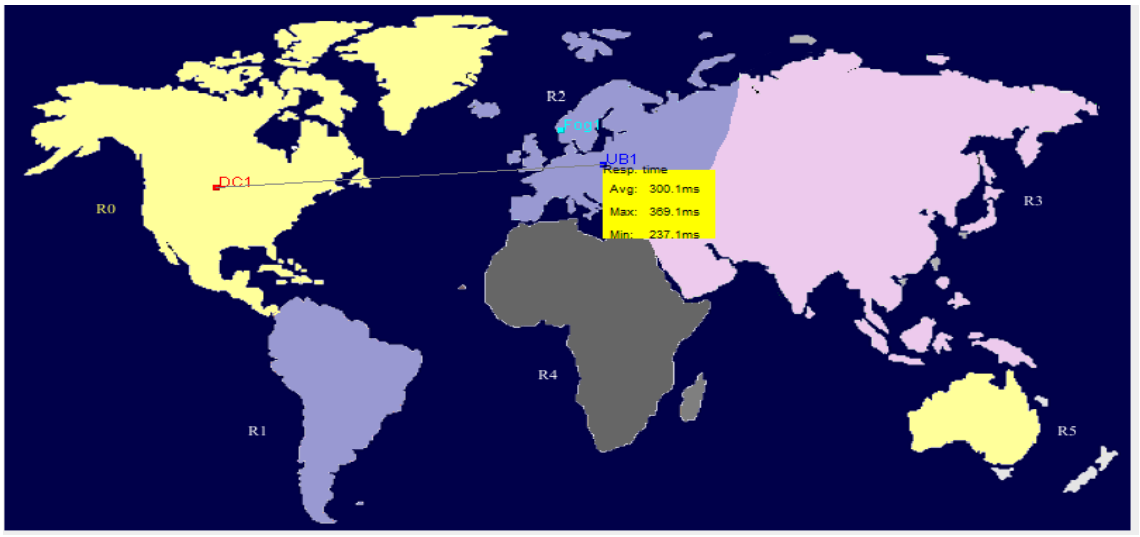

Figure 7: One user base and One Data base with Cloud only

one User base in Region 2 and Fog layer in Region 2. Figure 6 explains about region boundaries in Cloud Analyst simulator.

5.4 Experiment 2: One User Base and One Data Base with cloud only

In this experiment we create one user base in region 2 and one Data Center in region 0 to test how the Proposed prioritized algorithm impacts on the Internet of Things(IoT). In this experiment we use cloud only to perform the simulation and to observe the results.Figure 7 shows the simulation in Cloud analyst with Cloud only without Prioritized Algorithm.

5.5 Experiment 3: One User Base and One Data base with fog Enabled

In Experiment 3 we create One User base in Region 2, One Data Center in Region in Region 0 and Fog Data Center in Region 2 to evaluate the response time of the tasks. Figure 8 shows the simulation with above explained parameters.

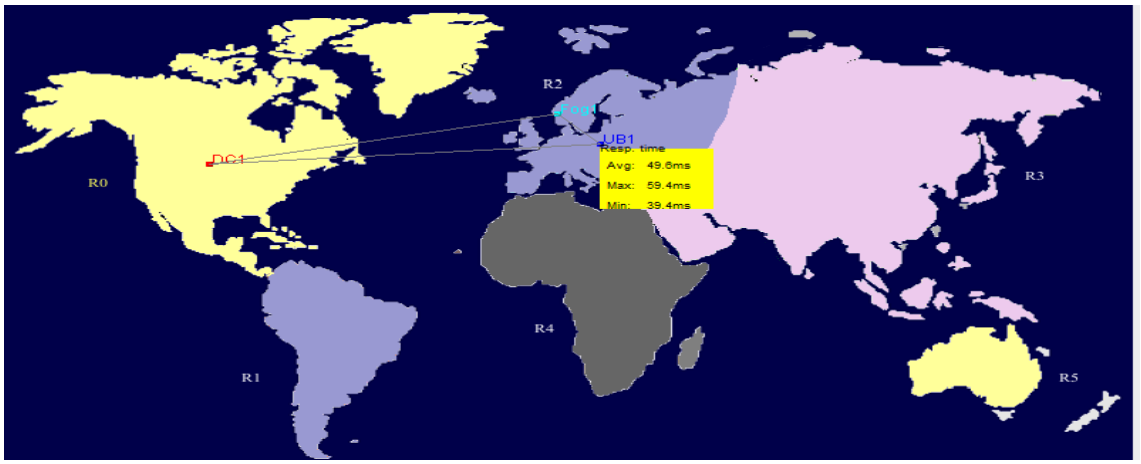

Figure 8: One user base and One Data base with Fog

5.6 Experiment 4:Two User Bases and One Data Center with fog Enabled

In this Experiment 4 we create Two user bases in two different regions one in Region 2 and another in Region 1 and one Data center in Region 0 and Fog Data center in Region 2.Figure 9 shows the simulation in Cloud Analyst with Two Data Bases with Fog enabled.

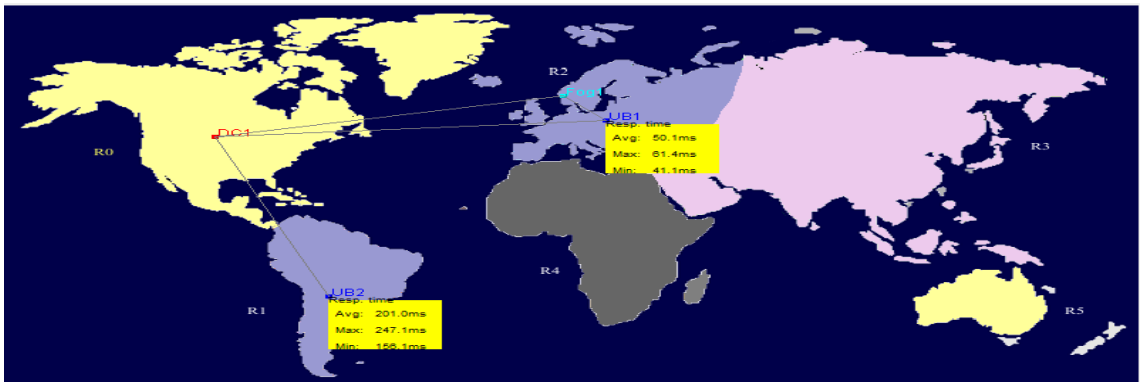

Figure 9: Two user Bases and One Data base with Fog

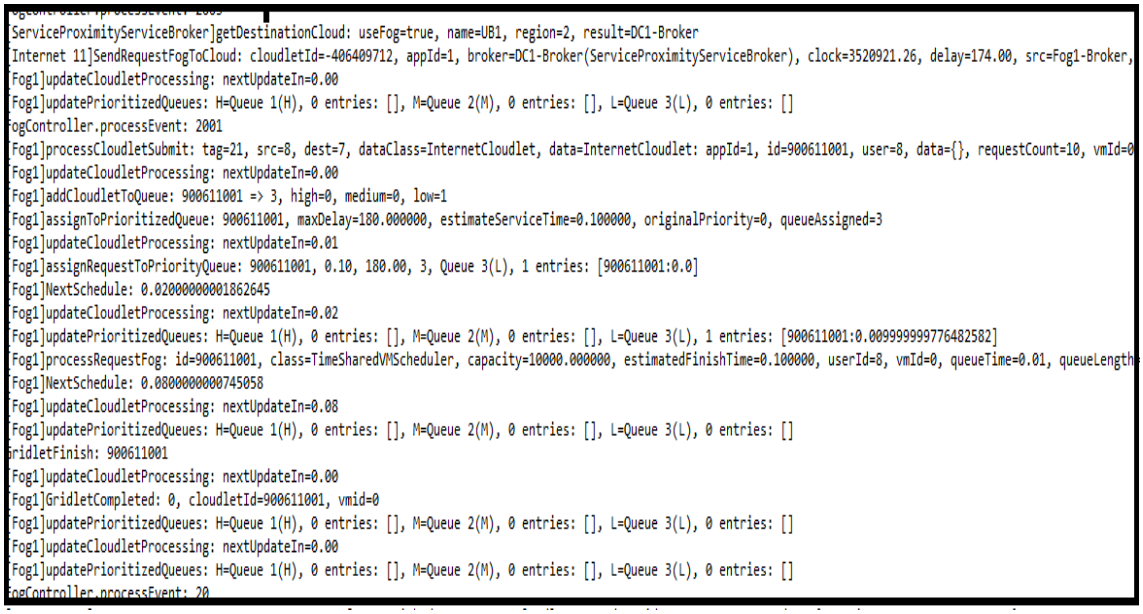

Figure 10: Scheduling tasks

5.7 Conclusion

This section Implemented Tasks and Fog computing and conducted various experiments by taking different cases to get the result. Following section evaluates the results and make a comparison.

6 Evaluation

The main purpose of this research is to analyze of the performance of proposed task scheduling algorithm in fog computing . The results are examined by performing three experiments using Cloud Analyst Simulator. The results are below

6.1 Experiment 1:Scheduling Tasks using Prioritized Scheduling Algorithm

Cloud Analyst will schedule the tasks according to their priority. Scheduling will be carried out at Fog level. Figure 10 shows the working of prioritized scheduling algorithm in Fog Environment.

6.2 Experiment 2:One UB with Cloud only

This experiment was done by one user base with cloud only to show the results with the prioritized scheduling algorithm.Table 4 shows the response time by performing experiment.

User base	Avg(ms)	Min(ms)	Max(ms)
UB 1	300.06	237.06	369.12

Table 4: Response time of Cloud only

User base	Avg(ms)	Min(ms)	Max(ms)
UB 1	49.64	39.36	59.37

Table 5: Single user with fog enabled

6.3 Experiment 3:One UB with fog enabled

This experiment was carried out by taking one User Base and one data Center with fog enabled.Table 5 shows the results of simulation performed on Single UB with Fog enabled .

6.4 Experiment 4:Multiple users with fog enabled

This experiment was carried by creating two user bases in two different regions with fog enabled environment.Table 6 shows the results after experiment.

6.5 Discussion

From the above experiments, we can conclude that Prioritized Scheduling algorithm in fog computing is successful in processing tasks much faster than traditional algorithms. It even proves that it works much efficiently for large data applications.Figure 12 shows a comparison in response time of Cloud only and Single User Base and Two User bases .

Figure 11 shows us the comparison of virtual machines cost in data centers while processing data.

Finally results shows that response time was gradually reduced by 83.46% and virtual machines cost by 46 %. Prioritized Scheduling algorithm will be successful in achieving greater results in real time environment.Prioritized scheduling algorithm will achieve more positive results in using Internet of Things.

7 Conclusion and Future Work

The main aim of this research was to reduce the response time and Virtual machine cost while data transferring using Prioritized task scheduling in fog computing. The Cloud Analyst Simulator was used to test the Prioritized scheduling algorithm. Scheduling using Prioritized algorithm will reduce the response time and Virtual machine cost. Fog will effectively bring the cloud closer to the end devices.The collaboration of fog computing with Internet of Things will bring many advantages to IoT applications. Fog provides

User base	Avg(ms)	Min(ms)	Max(ms)
UB 1	50.03	40.62	61.11
UB 2	502.46	390.09	617.62

Table 6: Multiple users with fog enabled

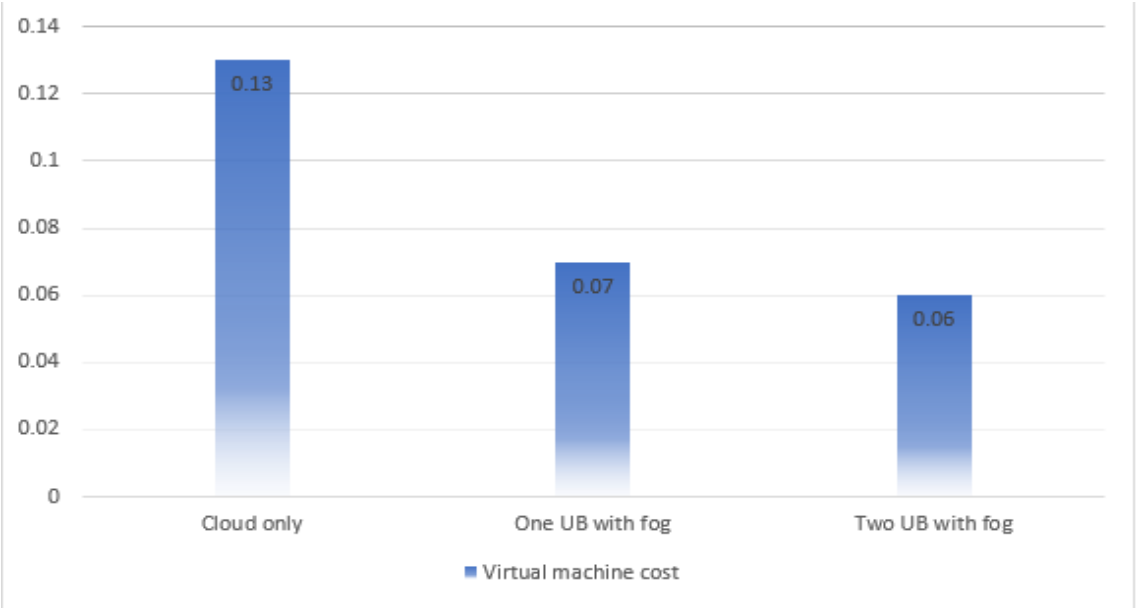

Figure 11: Comparison Graph of Virtual machine cost

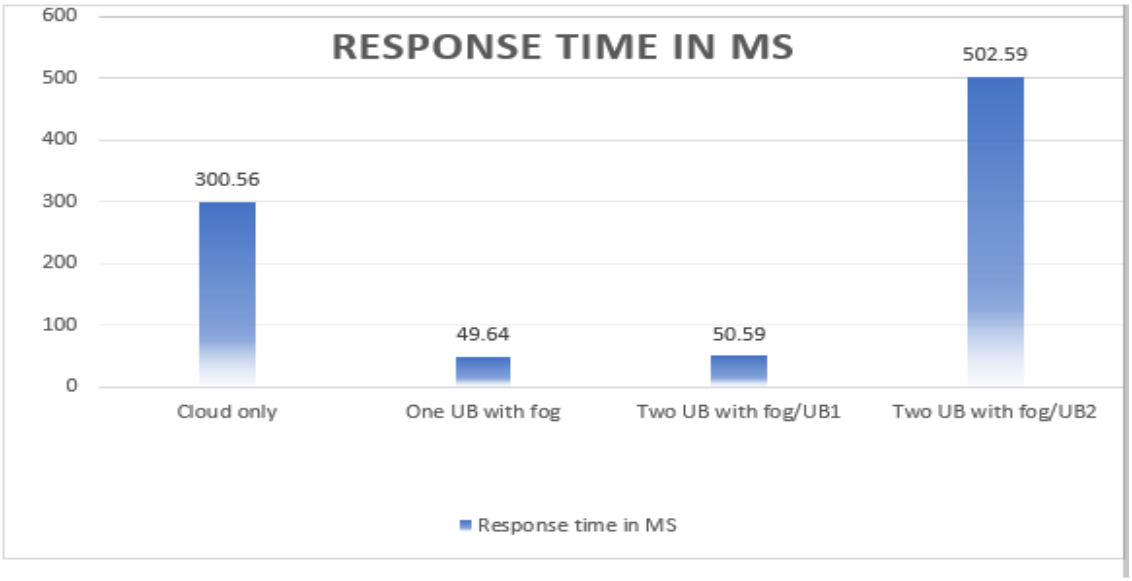

Figure 12: Comparison Graph of Response time

IoT data and stores them to avoid forwarding response to the cloud. The research paper has developed a Prioritized scheduling algorithm for the fog layer. Simulation results of Cloud Analyst shows that the Prioritized scheduling algorithm is successful in reducing the response time and decreased the overall cost. This is because of the Prioritized scheduling algorithm prioritize the tasks according to their delay time which eventually results in high throughput.
Future work of this paper includes scheduling algorithm by dynamically accordingly to the traffic flow in Cloud Environment, scheduling algorithm to reduce energy efficiency and implementing prioritized scheduling in other cloud environments for better results.

References

Abreu, D. P., Velasquez, K., Assis, M. R. M., Bittencourt, L. F., Curado, M., Monteiro, E. and Madeira, E. (2018). A rank scheduling mechanism for fog environments, *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)* pp. 363–369.

Atlam, H., Walters, R. and Wills, G. (2018). Fog computing and the internet of things: A review, *Big Data and Cognitive Computing* .

Bonomi, f., Militon, R., Zhu, J. and Addepalli, S. (2012). Fog computing and its role in the internet of things, *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* **4**: 13–16.

Choudhari, T., Moh, M. and Moh, T.-S. (2018). Prioritized task scheduling in fog computing, *Proceeding of the ACMSE 2018 certificate* **16**(22): 22:1–22:8.

Eclipse Documentation (2018).
URL: *https://www.eclipse.org/documentation/*

Hernandez.L, Cao.H and Wachowicz.M (2017). Implementing an edge-fog-cloud architecture for stream data management, *2017 IEEE Fog World Congress* pp. 1–6.

Howell, J. (2017). The internet of things is here and growing exponentially.
URL: *https://technology.ihs.com/596542/number-of-connected-iot-devices-will-surge-to-125-billion-by-2030-ihs-markit-says*

Islam.T and Hashem.M (2018). Task scheduling for big data management in fog infrastructure, *2018 21st International Conference of Computer and Information Technology* pp. 1–6.

Kavuri, R. (2018). A cloud-based task scheduling in fog computing by using heuristic approach: Iot.

K.Naha, R., S.Garg, D.Georgakopoulos, P.P.Jayaraman, L.Gao, Y.Xiang and R.Ranjan (2018). Fog computing: Survey of trends, architectures, requirements, and research directions, *IEEE Access* pp. 47980–48009.

Krishnan, Y. N., Bhagwat, C. N. and Utpat, A. P. (2015). Fog computing network based cloud computing, *2015 2nd International Conference on Electronics and Communication Systems* pp. 250–251.

Mohamed, N., Al-Jaroodi, J., Jawhar, I., Noura, H. and Mahmoud, S. (2017). Uavfog: A uav-based fog computing for internet of things, *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* pp. 1–8.

Mohammed.D and Nawwaf.K (2011). A hybrid heuristicgenetic algorithm for task scheduling in heterogeneous processor networks, *Journal of Parallel and Distributed Computing* pp. 1518–1531.

Moxa (2018). should you consider fog computing for your iot.
URL: *https://www.moxa.com/en/articles/should-you-consider-fog-computing-for-your-iiot*

Olaniyan.R and Maheswaran.M (2018). Synchronous scheduling algorithms for edge coordinated internet of things, *2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)* pp. 1–10.

Pham.X and Huh.E (2016). Towards task scheduling in a cloud-fog computing system, *18th Asia-Pacific Network Operations and Management Symposium* **1**: 1–4.

Priority Scheduling Algorithm Java Program (2018).
URL: *https://meetshrikantb.blogspot.com/2018/04/priority-scheduling-algorithm-java.html*

Rahbari, D., Kabirzadeh, S. and Nickray, M. (2017). A security aware scheduling in fog computing by hyper heuristic algorithm, *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)* .

Rashid.S and Sharifian.S (2017). A hybrid heuristic queue based algorithm for task assignment in mobile cloud, *Future Generation Computer Systems* pp. 331–345.

Saleh.H, H.Nashaat, W.Saber and H.M.Harb (2019). Ipso task scheduling algorithm for large scale data in cloud computing environment, *IEEE Access* pp. 5412–5420.

Talasila, S., Havisha, V., Koushik, S., Deep, M. and Reddy, V. (2016). Load balancing techniques for efficient traffic management in cloud environment, *International Journal of Electrical and Computer Engineering (IJECE)* **6**: 963.

The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne (2016).
URL: *The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne*

Wickremasinghe, B., Calheiros, R. N. and Buyya, R. (2010). Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications, *2010 24th IEEE International Conference on Advanced Information Networking and Applications* pp. 446–452.

8 Appendix

8.1 Specifications

This is an instruction manual which provides the instructions to install, Build and run the project.

8.2 Software tools used

The main software tools used in this project are
1.Eclipse IDE were used in this project to develop
2.Cloud Analyst Simulator
3.Windows Operating system

8.3 Installation Procedure

8.3.1 Installing Eclipse IDE in the Computer

1.Download Eclipse IDE 2018-12 from the following link
Url=https://www.eclipse.org/downloads
Eclipse Documentation (2018)
Figure 13 shows the installation procedure of IDE by selecting IDE for Java Developers.

Figure 13: Installing Eclipse IDE in personal computer

8.3.2 Importing Cloud Analyst into Eclipse

Figure 14 shows the importing procedure of Cloud Analyst .

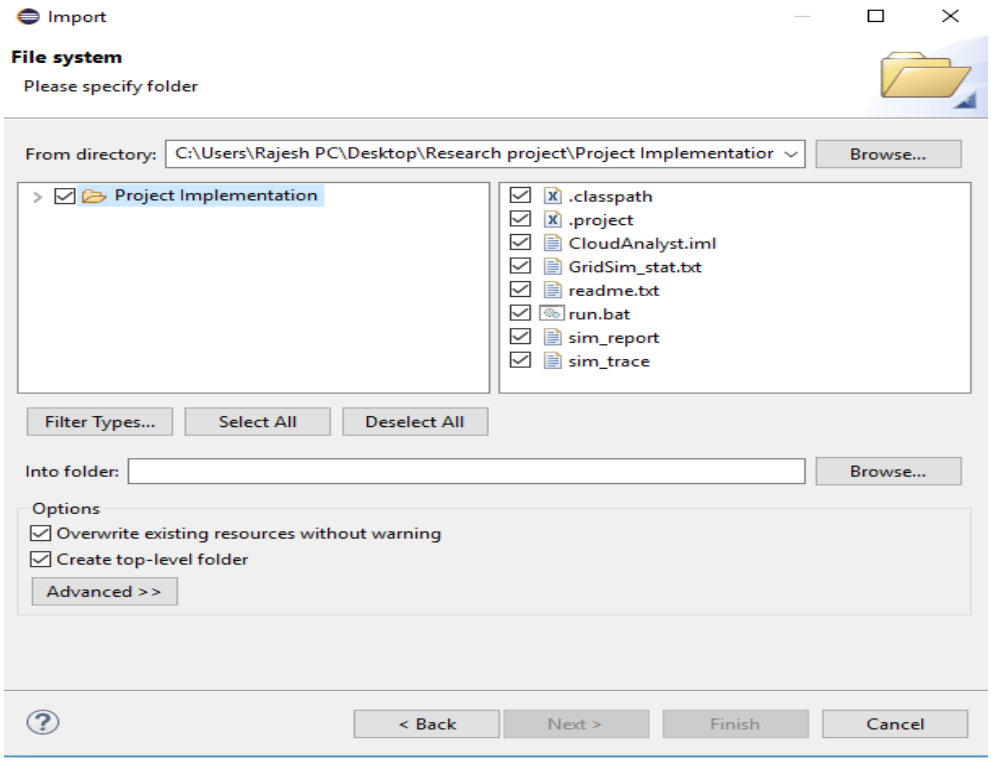

Figure 14: Importing Cloud Analyst into Eclipse IDE

8.3.3 Creating Prioritized Scheduling algorithm

```
public class PrioritizedTaskSchedulingInFog implements CloudAppServiceBroker {
    protected Map<Integer, List<String>> regionalDataCenterIndex = null;

    public PrioritizedTaskSchedulingInFog(){
        regionalDataCenterIndex = new HashMap<Integer, List<String>>();

        init();
    }

    protected void init(){
        List<GeoLocatable> allInternetEntities = InternetCharacteristics.getInstance().getAllEntities();
        int region;

        for (GeoLocatable entity : allInternetEntities){
            if (entity instanceof DataCenterController){
                region = entity.getRegion();
                List<String> l = regionalDataCenterIndex.get(region);
                if (l == null){
                    l = new ArrayList<String>();
                    regionalDataCenterIndex.put(region, l);
                }
                l.add(entity.get_name());
            }
        }
    }
```

Figure 15: Creating Prioritized Scheduling algorithm

Figure 15 shows the code for Developing Prioritized Scheduling algorithm.*Priority Scheduling Algorithm Java Program* (2018)

8.3.4 Creating a Fog environment

Figure 16 shows the code for creating Fog environment in Eclipse .

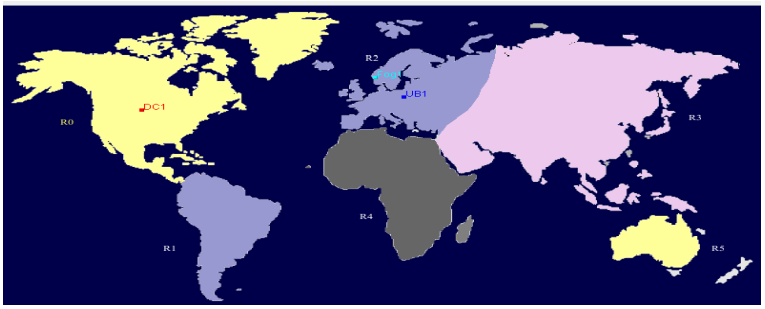

Figure 16: Creating Fog environment

8.3.5 Creating Sim Logger

Figure 17 shows the code for Sim Logger to implement a logging service. *The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne* (2016)

```
public class SimLogger {

   String delimiter;
   String prefix;
   public static final String DEFAULT_FIELD_DELIMITER = ",";
   /**
    * Constructor.
    *
    * @param sim
    * @param screenListener
    */
   private Logger logger;

   public SimLogger(String name, String prefix) {
       this.prefix = prefix;
       ///Init logger
       this.logger = Logger.getLogger(name);
       this.logger.setLevel(Level.ALL);
       ConsoleHandler handler = new ConsoleHandler();
       SimpleFormatter ft = new SimpleFormatter();
       handler.setFormatter(ft);
       this.logger.addHandler(handler);
       this.logger.fine("hello world");
       delimiter = DEFAULT_FIELD_DELIMITER;
   }
```

Figure 17: Creating Sim Logger

8.3.6 Eclipse Console

Figure 18 shows the console of fog computing in Eclipse IDE.

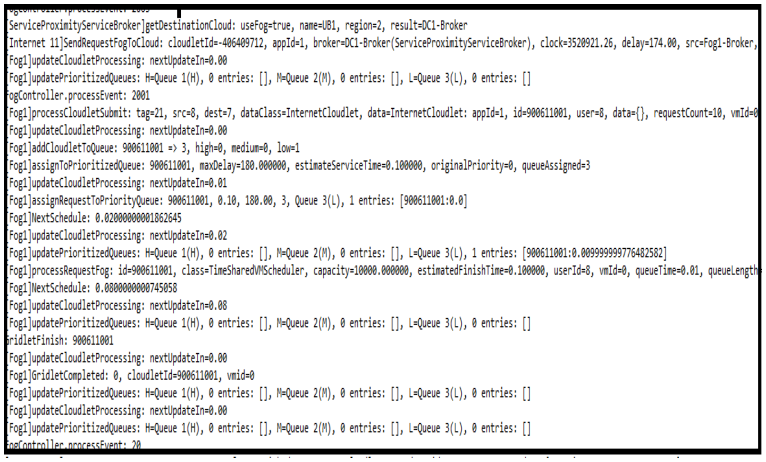

Figure 18: Console

8.4 Creating the Users in Cloud Analyst

After successfully writing the code in the Eclipse IDE when we run the program the Cloud Analyst will open .Figure 19 shows the Cloud Analyst Screenshot . Click on Configure Simulation to configure the users, fog and Data bases. figure 19 shows the screenshot of Cloud Analyst.

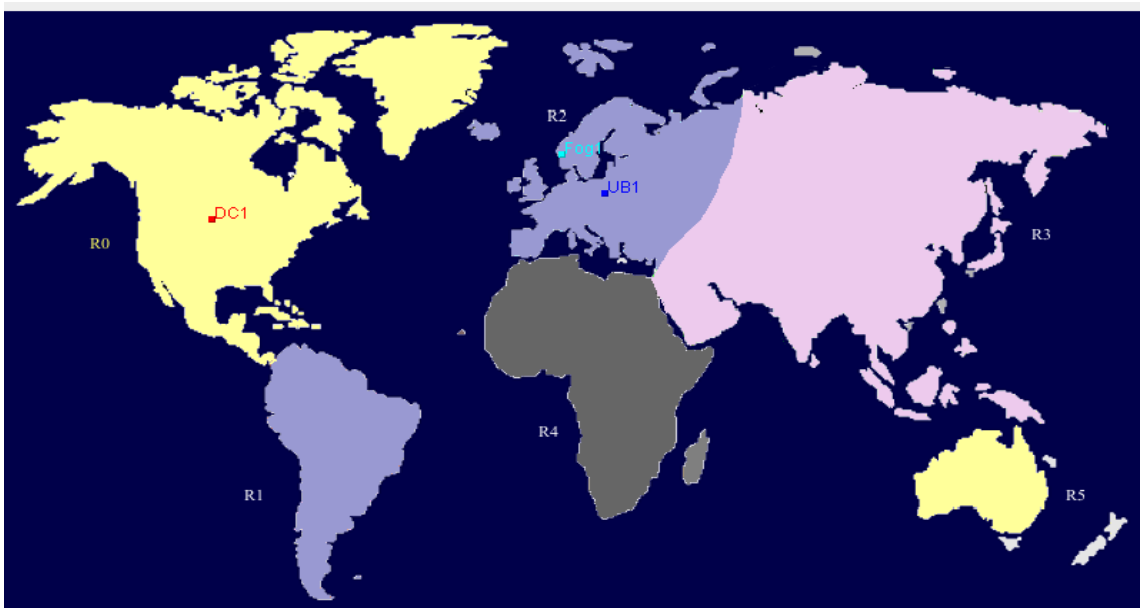

Figure 19: Cloud Analyst

8.4.1 Experiment 1:One User base and One Data base with Cloud only.

In this Experiment we create One User base in region 2 and Data base in region 0. 20 shows the configuring the user base using Cloud Analyst.
Figure 21 shows the result after the simulation .

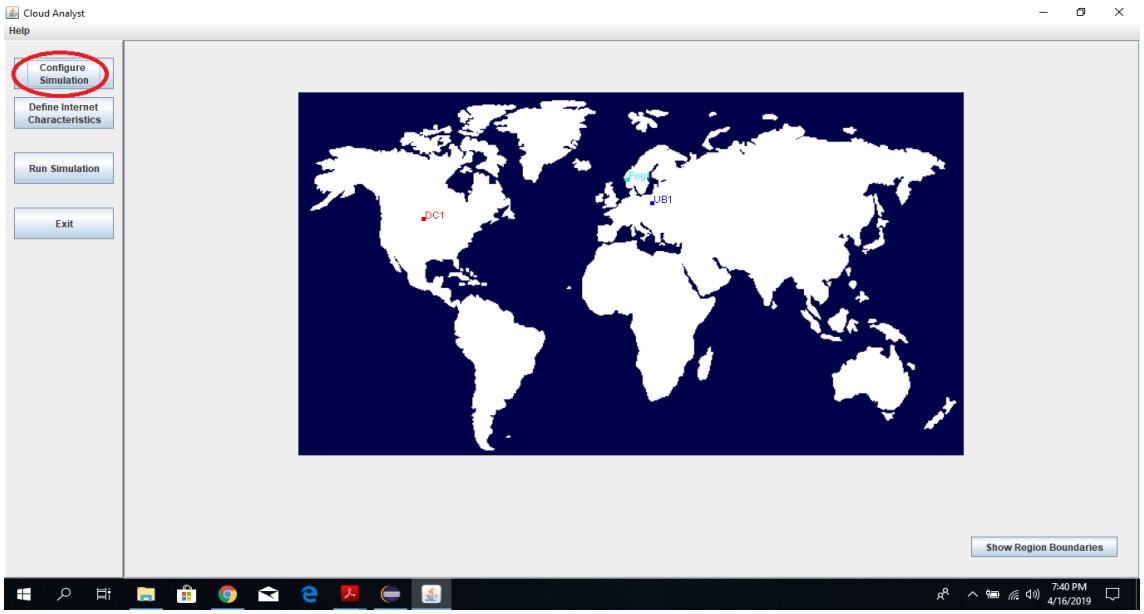

Figure 20: Cloud Analyst with Cloud only

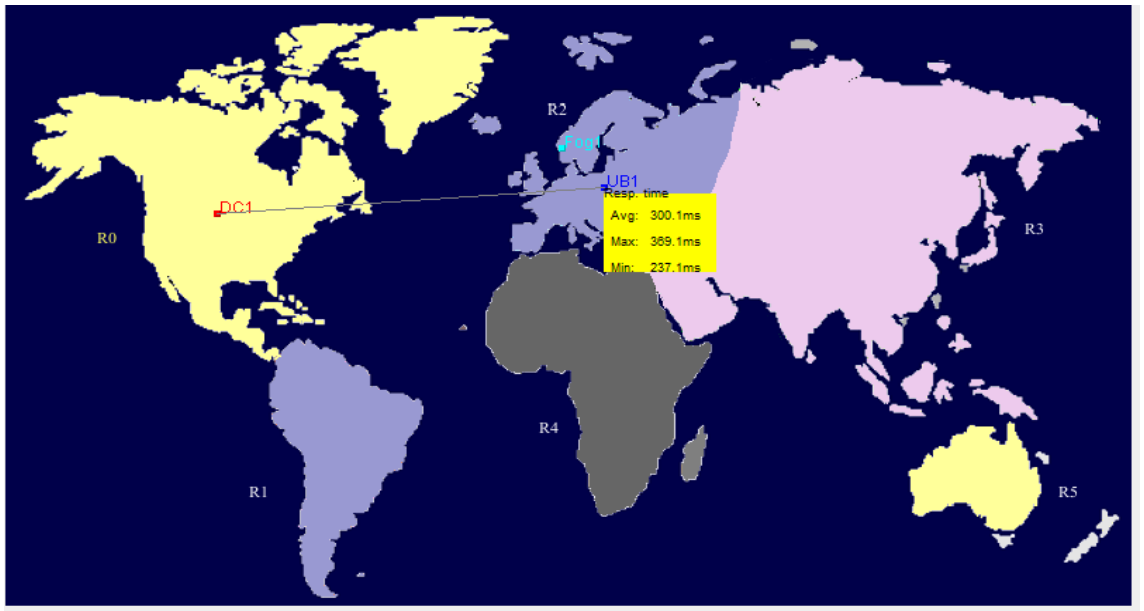

Figure 21: Users with Cloud only

8.4.2 Experiment 2:One User base and One Data base with fog enabled.

In this experiment we create one user base and one data base with fog enabled .Figure 23 shows the simulation results after configuring single user with fog enabled.
22 shows the screenshot of enabling Prioritized Scheduling algorithm in Cloud Analyst Simulator.

8.4.3 Experiment 3:Two user base and One Data base with fog enabled.

In this experiment we create Two user bases and One data base with fog enabled

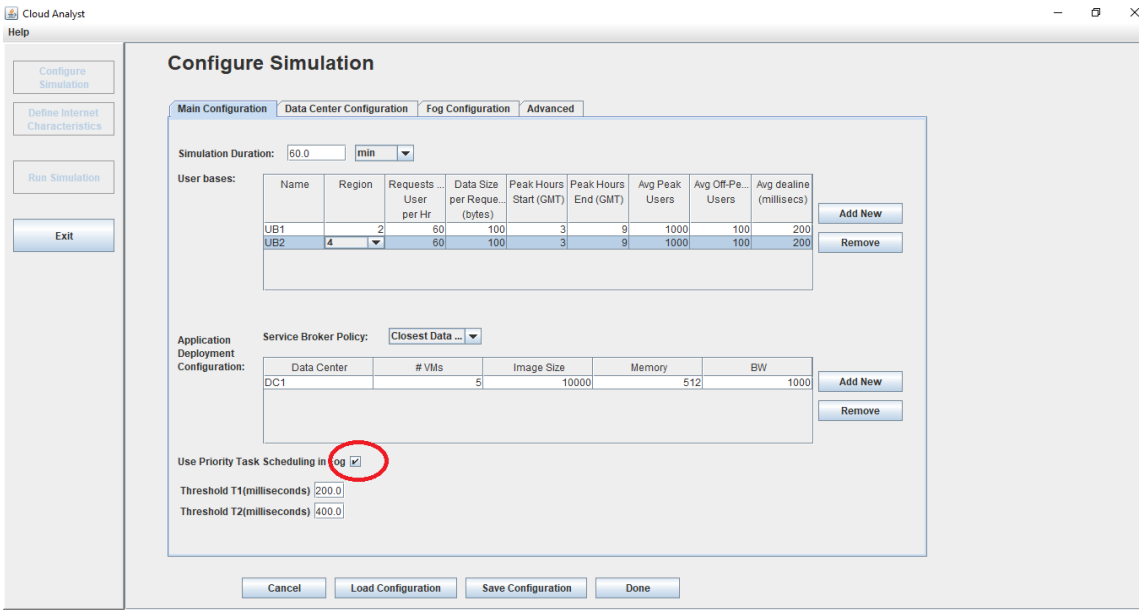

Figure 22: Enabling Prioritized algorithm with fog

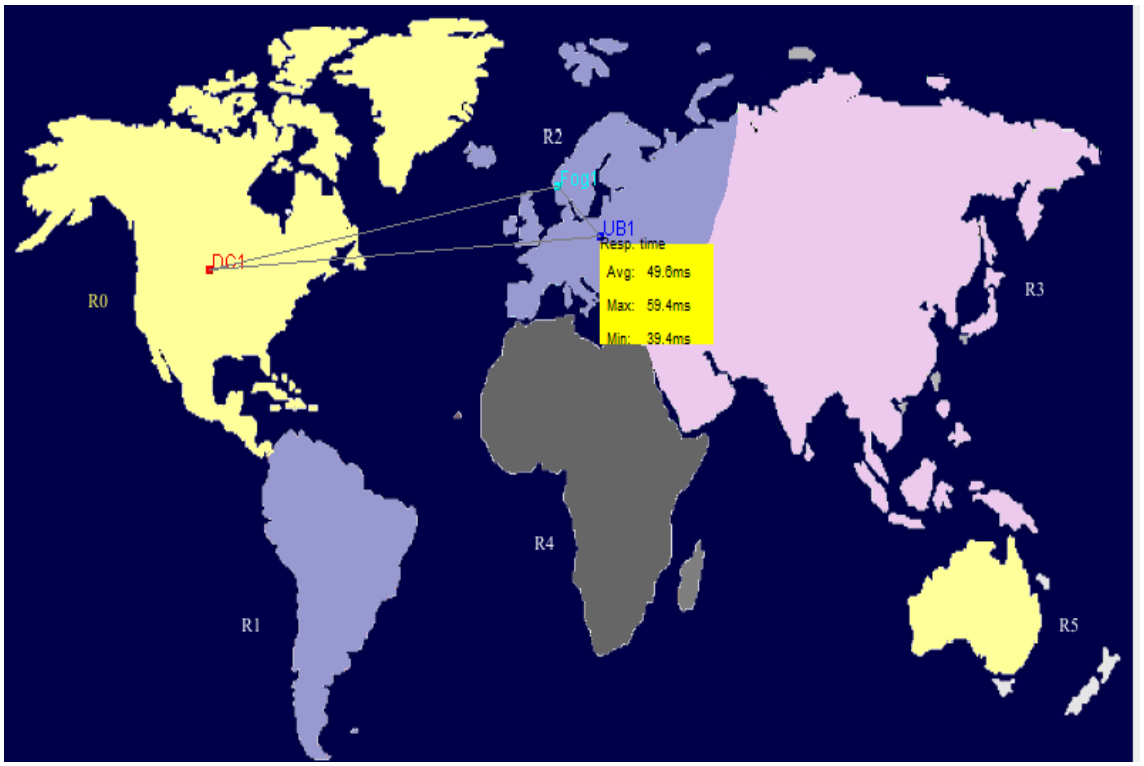

Figure 23: Single user with fog enabled

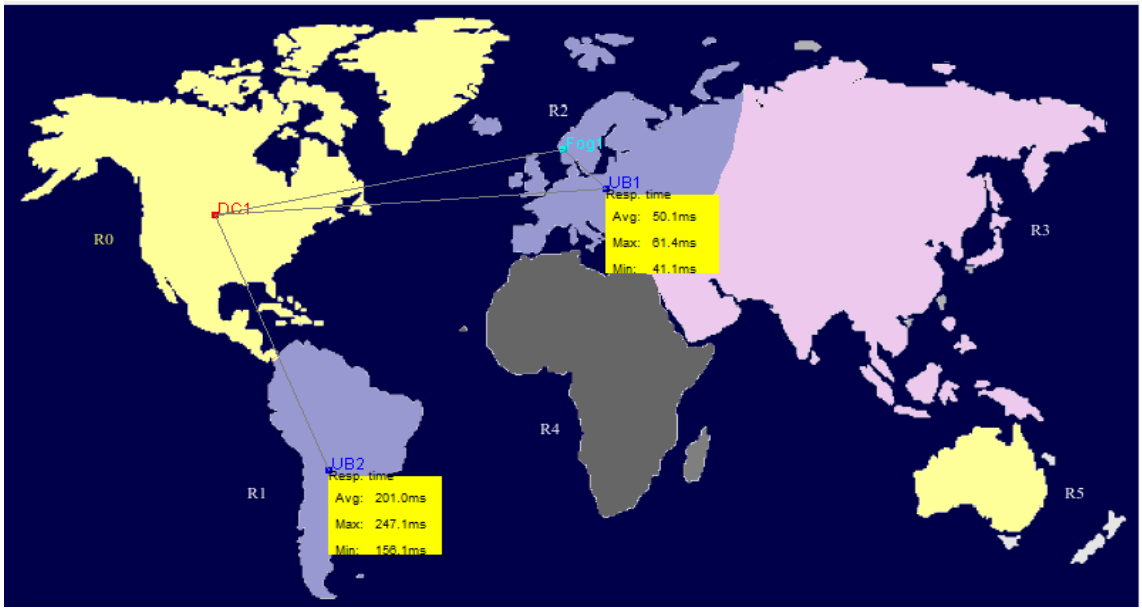

Figure 24: Multiple users in two different regions with fog enabled