National College of Ireland

Efficient resource optimization and scheduling of QoS in cloud content delivery network

MSc Research Project
Cloud Computing

Ateet Jayeshkumar Shah
Student ID: x17127785

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ateet Jayeshkumar Shah		
Student ID:	X17127785		
Programme:	MSc. Cloud Computing	**Year:**	2018-2019
Module:	MSc. Research Project		
Lecturer:	Victor Del Rosal		
Submission Due Date:	20/12/2018		
Project Title:	Efficient resource optimization and scheduling of QoS in cloud content delivery network		
Word Count:	5855	**Page Count:**	20

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: 20/12/2018

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):	

Contents

Efficient resource optimization and scheduling of QoS in cloud content delivery network

Ateet Jayeshkumar Shah
X17127785

Abstract

Cloud computing offers infrastructure platform which enables users to host different services in cloud. The benefits of virtualization, distributed nature of cloud and the scalability feature of cloud can be leveraged in content delivery network (CDN) which forms cloud content delivery network. CDN in cloud has become one of the significant services of the internet due to growing internet traffic. Server-side processing, streaming of content and delivering the content with better quality of service (QoS) at reduced cost is a key issue for content providers and end users. Many times, there is an uncertainty of delivering the content and achieving guaranteed quality of service (latency issue, real time issue, response and execution time issue) since the demand from the end user is dynamic. Thus, achieving guarantee quality of service becomes our primary objective in cloud content delivery network and forms the basis of our research. We propose an improved method on dynamic rate scheduling to improve QoS in cloud content delivery network. We are performing the experiment on cloud sim toolkit to create a distributed content delivery network environment and comparing it with different techniques like load balance and static mechanism. This research proposal is aimed towards the open problem of quality of service in cloud computing environment for content delivery network. This work is dedicated to researchers in cloud domain, content delivery network and for academic students.

Keywords – *Cloud computing, Content delivery network, dynamic rate scheduling, quality of service*

1 Introduction

Cloud computing consist of distributed homogeneous data centres which are virtualized in nature to host different services to the end users. The affordability and flexibility offered by the infrastructure platform lead to more and more usage of cloud computing technology by user across the globe. Content delivery network in cloud are used to deliver content produced by the content provider to the end users. Content may be in the form of images, audio and video which is streamed by the end user. This virtualization technology is leveraged in content delivery network to form cloud content delivery network taking the advantage of scalability and cost reduction.

A group of servers which are distributed in nature across the globe forms a content delivery network. The content is replicated from one server to another server which is closer to the end user for delivering better quality of service.

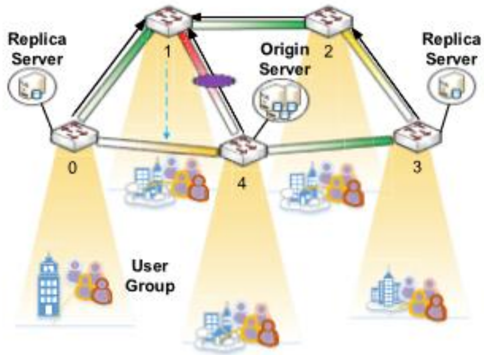


Figure 1: CDN example (Xu *et al.*, 2018)

Figure 1 shows a content delivery network (Xu *et al.*, 2018). There are five servers and content are hosted on the origin server and is replicated from one server to another depending on the requirement from the end user. Few of the parameters which affects quality of service are response and execution time, bandwidth, energy, utilization (Madni *et al.*, 2016). Also, the end user uses different platforms for content viewing like mobile phones, laptop, tablets. The rapid increase in adoption of broadband for watching of different sports and news events is done with the help of content delivery network. These traditional CDN are less scalable and robust in terms of varying user demand which results in poor quality of service.

71 % of the internet traffic will flow through content delivery network by 2021 as predicted by Cisco which is increase of 52 % as compared to traffic in 2016[1]. Thus, it becomes utmost important to deliver better quality of service at reduced price in content delivery network (Anjum *et al.*, 2017; Stocker *et al.*, 2017). Traditional CDN being more rigid for scaling demands results in more operating and capital costs to provide satisfactory quality of service.

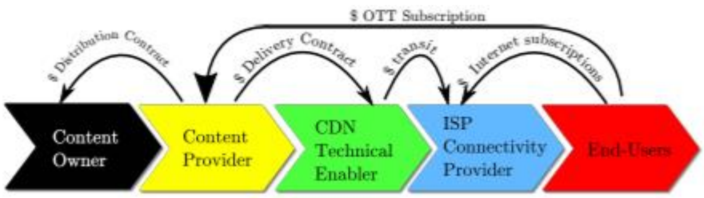


Figure 2: CDN chain (Herbaut *et al.*, 2016)

Figure 2 shows content delivery network chain which shows how content is delivered to the end user. Content is produced by the owner is delivered to the content provider. The end user subscribes to the content provider for accessing the content. There is a symbiotic relationship between CDN enabler and ISP provider to deliver the content to the end user (Herbaut *et al.*, 2016). Some of the facts presented by (Krishnan, Sitaraman and Systems, 2012) based on study of 6.7 million different users and 23 million views shows that a user stops watching a video

[1] Source: Cisco Visual Networking Index: Forecast and Methodology, 2013 - 2018

just after 2 seconds and more than 20 % of user moves to see new video just after 5 seconds. Thus, like quick response time becomes one of the important parameters in content delivery network. Also, as per Cisco, 78 % of the total consumer traffic is IP video traffic[2]. Considering the facts, there is lot of scope in properly scheduling of resources and optimizing it in cloud CDN. This brings us to a question in virtualization domain:

How do we achieve guaranteed QoS in cloud-based content delivery network using QoS aware resource scheduling and optimization technique?

The motivation behind the research question is to increase the placement and allocation technique with the added parameters required in service provider network like dynamic demand and quality of service penalty. We show how two stage approach of scheduling and optimization fares against default technique used in cloud sim toolkit as well as other technique like greedy and static QoS.
This report is organized as follows: In Section 2, related work and discussion are presented. In Section 3, we present method and details on how our research is carried out. Section 4 explains the specification of our two-stage algorithm that we are proposing. In Section 5, we present the output results by performing different simulation on cloud sim toolkit. In Section 6, we evaluate the numerical results obtained by comparing it with other techniques discussed in the section 2 of the paper and we summarize and conclude the paper with the future work in section 7.

2 Related Work

In this section we discuss the existing work done in scheduling and optimization of resources to draw out the key elements from the literature work and form the basis of our solution. There is plenty of research being done on virtual machine placement issue in cloud environment. This cloud environment is distributed in nature and forms the basis for content delivery network (Ben Jemaa, Pujolle and Pariente, 2016). The scheduling of resources and the optimization based on varying demand are key parameters for improving quality of service which are focused in the literature study. The challenge with scheduling is dynamic request from user, selection of resources and allocation of resources for quality of service (Kumar *et al.,* 2014).

2.1 Virtual machine placement in Cloud and CDN environment

(Kumar and Raza, 2015) presents a virtual machine allocation strategy based on swarm optimization approach. Parameters like wastage of resources and servers used are taken into consideration. The author did compare the output values with the best fit and first fit algorithm using the similar test bed by considering scheduling of resources on different host machines to avoid wastage of resources but failed to consider the dynamic nature of allocation required for cloud environment. Dynamic allocation of resources can improve the quality of service to great extent.

[2] Source: Cisco Visual Networking Index: Forecast and Methodology, 2013 - 2018

(Broberg, Buyya and Tari, 2009) presents an innovative approach of Meta-CDN for placing of web server in different locations. The key parameter evaluated in quality of service is responsiveness and throughput keeping cost and budget in mind. This approach is intended towards cloud storage strategy while failing to consider the global scenario for content delivery network.

On the contrary, the above problem was easily mitigated in two step approach proposed by (Chen *et al.*, 2012). The joint problem of large-scale network and quality of service requirement is solved using storage cloud-based content delivery network. The overall cost decreases with this approach but the quality of service is just satisfied and not improved drastically. The real power of virtualization used in cloud computing infrastructure of cloud provider is used in content delivery network called Active CDN as described by (Srinivasan *et al.*, no date). The dynamic placement of content using network virtualization partially serves the purpose of distributing and making content available over different locations to cater end user needs but it does not consider the key quality of service parameters like latency which should be minimum. (Wang, Liu and Chen, 2012) proposes Cloud assisted live media streaming (CALMS) framework. This framework takes care of the distributed users across the globe along with dynamic demand of the end user. The only drawback of this approach is that quality of service is just satisfactorily met. Also, this approach is specific for video content over peer to peer network and does not satisfy for distributed network. On the contrary, (Niu *et al.*, 2012) proposes a solution to cater to dynamic user demand by pre-booking bandwidth across multiple data center to deliver better quality of service to the end user but the approach is not scalable for content delivery network. (Kömer, Abraham and Snášel, 2014) proposes a combination of genetic and fuzzy theory hybrid approach to cater to the virtual machine placement problem in distributed environment. The author used cloudsim as the simulation platform and parameters like execution time and cost are evaluated. The average makespan was quite less with the new genetic approach as compared to Ant colony optimization. The pre-booking of bandwidth limits the power to scale in terms of dynamic demand.

According to (Madni *et al.*, 2016), there are various scheduling technique for cloud computing. Following are the types of scheduling:

- Efficiency aware scheduling which includes response time, makespan, execution time, priority and bandwidth.
- QoS aware scheduling includes reliability, fault tolerance, throughput and recovery time.

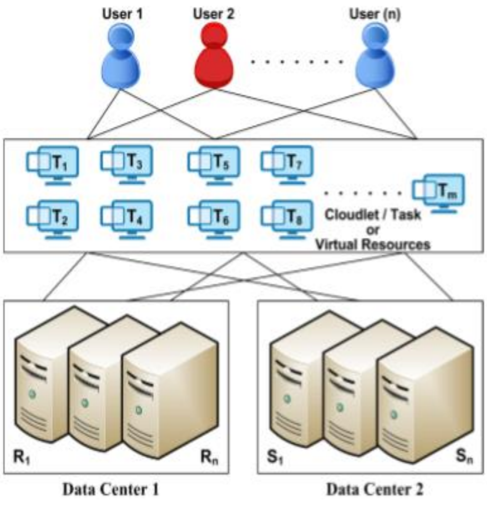


Figure 3: Placement in Cloud environment (Madni *et al.*, 2016)

Figure 3 shows the different user requirement for resource in different datacentre distributed across regions around the world. These requirements are called as cloudlet or virtual resources and needs to be allocated to the user depending on their requirement. (Madni *et al.*, 2016) presents different scheduling technique to cater cloudlet request in optimized way.

From the above related work, the virtual machine placement requires scheduling of resources in content delivery network. This scheduling involves allocation and de allocation of resources using optimized techniques. In the next section, we will discuss how resource scheduling affects quality of service in cloud environment.

2.2 QoS concerns for resource scheduling & optimization in cloud environment

(Son, Jung and Jun, 2013) presents a framework based on service level agreement considering distriuted data centers and allocation of resources. The authors approach focuses on increasing the profits for service providers using automated service level agreement. The author's approach shows improved results as compared to round robin, greedy and manual technique. But the author failed to consider dynamic demand from the end user which is of utmost importance for cloud environment.

(Alicherry, 2012) presents a network-based approach integrated with resource allocation technique for wide area network. The efficient resource allocation technique not only considers distance between bandwidth costs but also considers latency and communication cost. The proposed technique by the author does provide better results as compared to other simpler technique but the author fails to consider dynamic demand nature of the end user for distributed environment.

(Wei *et al.*, 2017) implemented a framework for video transcoding system based on real time workloads. Results from this framework shows improved results for predefined quality of service. The author uses queuing theory for transcoding workload to solve the optimization problem with QoS constraints in cloud environment. This QoS aware system for real time application can be implemented on the cloud-based content delivery network for streaming media services.

(Zhang *et al.*, 2013) proposes a game theory approach to solve the dynamic demand needs in cloud environment. The author tries to tackle main issue faced by service provider is to determine where the applications needs to be placed for optimized delivery such that the cost is minimum and quality of service (response time) is satisfied. The author performs simulation of their approach to optimize the cost for end user with improved results but failed to consider the scheduling parameter for distributed environment.

Recently, a lot of research is done in quality of service improvement in content delivery network. Instead of resource scheduling approach the author (Walkowiak, 2005) proposes a CDN enabled MPLS network to cater to dynamic content at reduced cost based on demand rejection ratio. The approach does satisfy the cost factor in content delivery network but does not improve the quality of service to great at extent as it is restricted to only service provider network which runs over MPLS. A two stage approach of resource provision and site assignment is proposed by (Haghighi, Shah Heydari and Shahbazpanahi, 2017). The author considers the large- and small-scale scenario for content assignment which is a must parameter for content delivery network, but the migration is not effective for variation in demand.

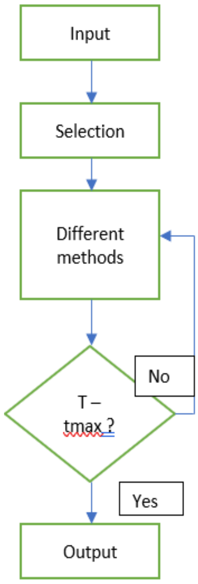


Figure 4:Hybrid heuristic method (Laha and Chakraborty, 2009)

An alternative technique called as hybrid heuristic is proposed by (Laha and Chakraborty, 2009) where combination of two different technique are used to get better results. This approach shows better results as compared to other traditional approach. But the author fails to consider dynamic environment.

From the above related work, it can be seen that the cloud environment is a distributed set of data center across the globe. This forms the basis of content delivery network which requires distributed environment. The problem of scheduling of resources in cloud environment can be extended to content delivery network. In the next section, we will discuss how quality of service is affected for scheduling of resources in content delivery network.

2.3 QoS concerns for resources scheduling in CDN environment

Content delivery networks have become very popular these days. The proposed technique by the (Sahoo *et al.*, 2017) ensures that the cost is minimized while delivery quality of service to the end user using server replication strategy. This technique of replica server placement at appropriate location is identified as key design issue. The results obtained is compared with traditional technique and thus providing better result. The only drawback of this approach is the dynamic request from the end user which is not considered in the design.

(Xu *et al.*, 2018) proposed a content delivery network framework focusing on maximising hit ratio with minimum latency. This approach considers placement of server replica along with content caching where performance of CDN changes with processing capacity. But the author fails to consider the optimization required for content delivery network. This issue was overcome by (Hu *et al.*, 2016) by studying video distribution in content delivery network considering parameters like cost and time delay. The author used Lyapunov optimization framework for development of dynamic technique which includes request routing and community-based replication of content. The results show that there is an improvement in terms of cost but failed to demonstrate any significant improvement in terms of quality of service.

(Jia *et al.*, 2017) presented a collaborative approach network infrastructure and content delivery network for content caching. This approach as suggested by the author overcomes the insufficient aspects of peer to peer network and CDN. This survey throws light on the new technology like software defined network integration with content delivery network to deliver better quality of service while considering the cost factor for service provider but still needs to be tested. In this approach, the delivery network utilizes characteristic and information from service providers to improve overall efficiency of content replication and optimization of network performance. The author compares traditional solutions with revolutionary solutions and presents challenges and future direction for content delivery networks.

(Jiang *et al.*, 2015) presents a novel concept of on demand dynamic resource allocation for distributed wireless network. This includes pre-determine and assigning fixed minimum bandwidth to each requirement using colouring algorithm and then reacting to new dynamic request based on current demand. The results indicate an improvement in quality of service as compared to other benchmarking schemes but this method needs to be further tested for content delivery network. (Lai, Hwang and Chao, 2017) proposes a QoS aware scheduling for 3D rendering in cloud environment. This approach is best suited when the bandwidth is not stable and need for dynamic request from end user is required. The proposed technique is a hybrid approach towards 3D rendering using client-side graphics along with cloud-based platform. This approach can be leveraged to content delivery network considering dynamic needs from the end user and host.

2.4 Discussion

The related work gives us indepth knowledge on the important elements needed to be considered for content delivery network in cloud. Each technique used in the related work is different from other one and have pros and cons in terms of quality of service and cost. Virtual machine placement in content delivery network in cloud considering quality of service as a key parameter is one of the key concerns. We highlight various scheduling and optimization technique in subsection 2.1, 2.2 and 2.3.

The problem of resource allocation in cloud can be extended to content delivery network along with optimization approach. Below table shows summarization of different techniques and missing parameters:

Literature title	Missing features
A PSO based VM resource scheduling model for cloud computing	dynamic scheduling
MetaCDN: Harnessing 'Storage Clouds' for high performance content delivery	scalability missing
Intra-cloud lightning: Building CDNs in the cloud	QoS missing
ActiveCDN: Cloud computing meets content delivery networks	Latency missing
CALMS: Cloud-assisted live media streaming for globalized demands with time/region diversities	QoS missing
Quality-assured cloud bandwidth auto-scaling for video-on-demand applications	not scalable for CDN
Hybrid job scheduling algorithm for cloud computing environment	dynamic scheduling
An SLA-based cloud computing that facilitates resource allocation in the distributed data centers of a cloud provider	dynamic scheduling
Network aware resource allocation in distributed clouds	scalability missing
QoS-aware resource allocation for video transcoding in clouds	not scalable for CDN
Dynamic service placement in geographically distributed clouds	Latency missing
QoS dynamic routing in content delivery networks	QoS missing
A constructive heuristic for minimizing makespan in no-wait flow shop scheduling	not scalable for CDN

Analyzing the missing parameters, it can be concluded that the techniques used in different related work doesn't exact solve the quality of service problem in content delivery network and thus following criteria should be considered for quality of service:

- Utilization of virtual machine for distributed regions
- Optimization factor
- QoS violation factor

The above parameters need to be considered for designing new technique to tackle this problem. These parameters consider the dynamic demand as well as response time for large distributed region. User's pay for content availability and the quality of service delivered to them. E.g. Netflix, Amazon, Facebook. In the next section, we will propose our approach and demonstrate the working of our model considering above parameters.

3 Research Methodology

As discussed in the subsection 2.4, we are proposing a new technique to solve quality of service issue in cloud content delivery network. This technique is a combination of scheduling and optimization method. The two-stage method includes allocation and deallocation of resources based on cloudlet input and then performs the migration based on dynamic needs variation from the end user.
The method is displayed in figure 5 as following:

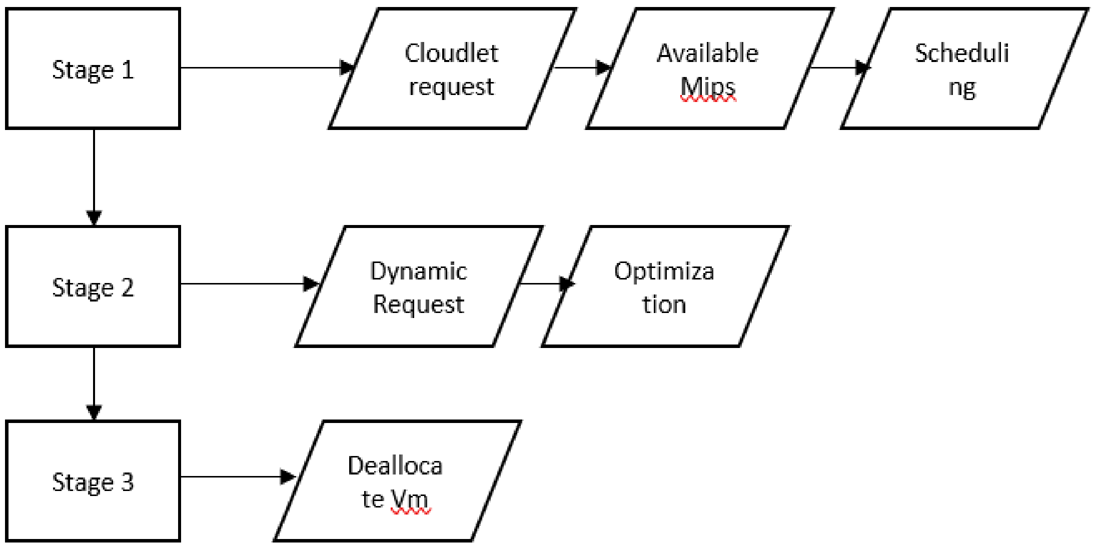


Figure 5: Process method

3.1 Stage 1 – Cloudlet request and allocation

Depending on the input cloudlet task to the broker, the broker checks for the available mips (instructions/second) in each virtual machine. Depending on the availability of mips, the cloudlet is assigned to either weak host or strong host to balance the system. Figure 6 shows the sequence flow diagram for stage 1.
In this diagram, each cloudlet is map to available size of virtual machine for proper load distribution in the network. A broker is first created for content delivery network operations. Cloudlet task are mapped to virtual machines depending on the element's availability. We sort the available hosts according to compare method and then assign each virtual machine to host which can handle the virtual machine.

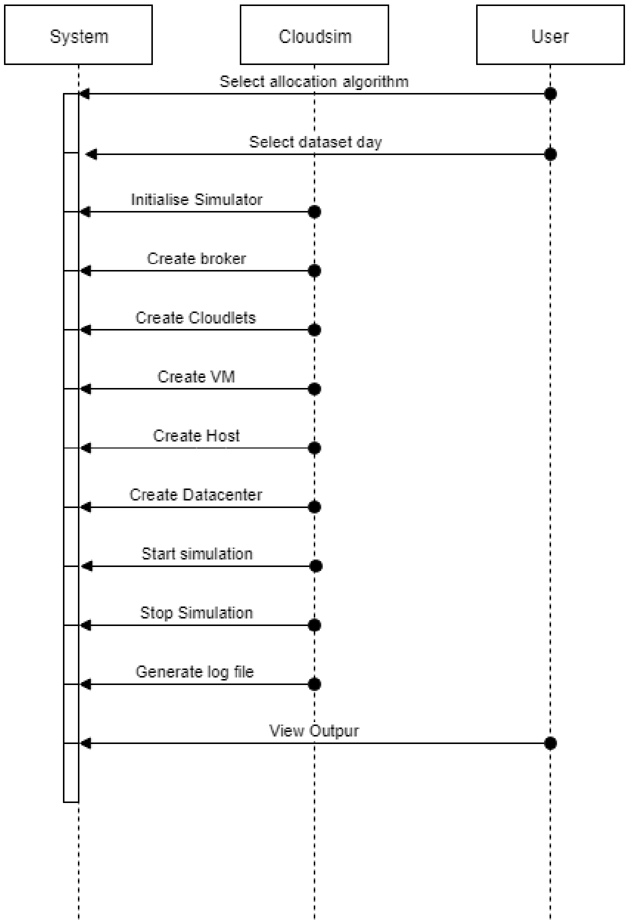


Figure 6: Sequence Diagram

3.2 Stage 2 – Optimization of dynamic request

In this section, we will explain the technique flow and process of optimization for dynamic request in detail. The new method of combination which is used in cloud sim with the help of Eclipse is further explained. Figure 7 shows a flowchart for the technique we have used. The process flow for stage 1 and stage 2 is explained as follows:

Step 1: *Cloudlet Task input to broker*

Step 2: *Initialize Broker based on cloudlet*

Step 3: *Create datacenter|broker*

Step 4: *Create virtual machine based on cloud information service*

Step 5: *Broker receive virtual machine detail*

Step 6: *Map cloudlet to virtual machine (scheduling)*

Step 7: *Start the simulation*

Step 8: *Migrate the virtual machine to nearest location based on dynamic demand (optimization)*

Step 9: *Complete the simulation*

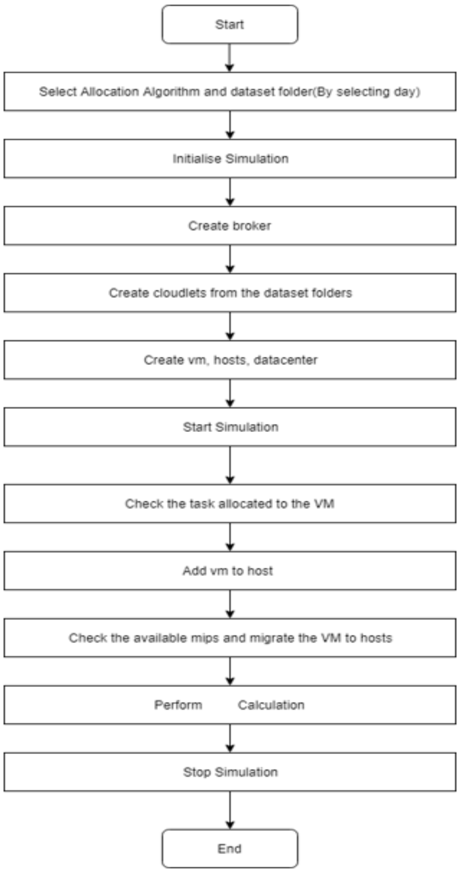


Figure 7: Flow chart

In this optimization process, after the mapping of cloudlet is done to the virtual machine, we check if there is any unallocated virtual machine and a list is created of such VMs. Then a list is created of virtual machine which are assigned to host as more utilized hosts category. Optimization is performed that is to increase the utilization of under-utilized host by reassignment to virtual machine from the category of underutilized hosts.

Formula –
QOS – optimization = total allocated mips - total missing mips divided by total allocated mips*100

3.3 Method Complexity

The optimization part of this method for dynamic movement of virtual machine is determined by linear integer equation. This complexity is due to the parameter of mips available in each host which is taken into account. (Ibn-Khedher *et al.*, 2017) presents similar approach to solve linear integer issue. In our research, optimization will happen based on number of mips available in each host.

Formula –
Profit for customer = cost incurred for VM – Optimization cost (penalty) – usage for each VM

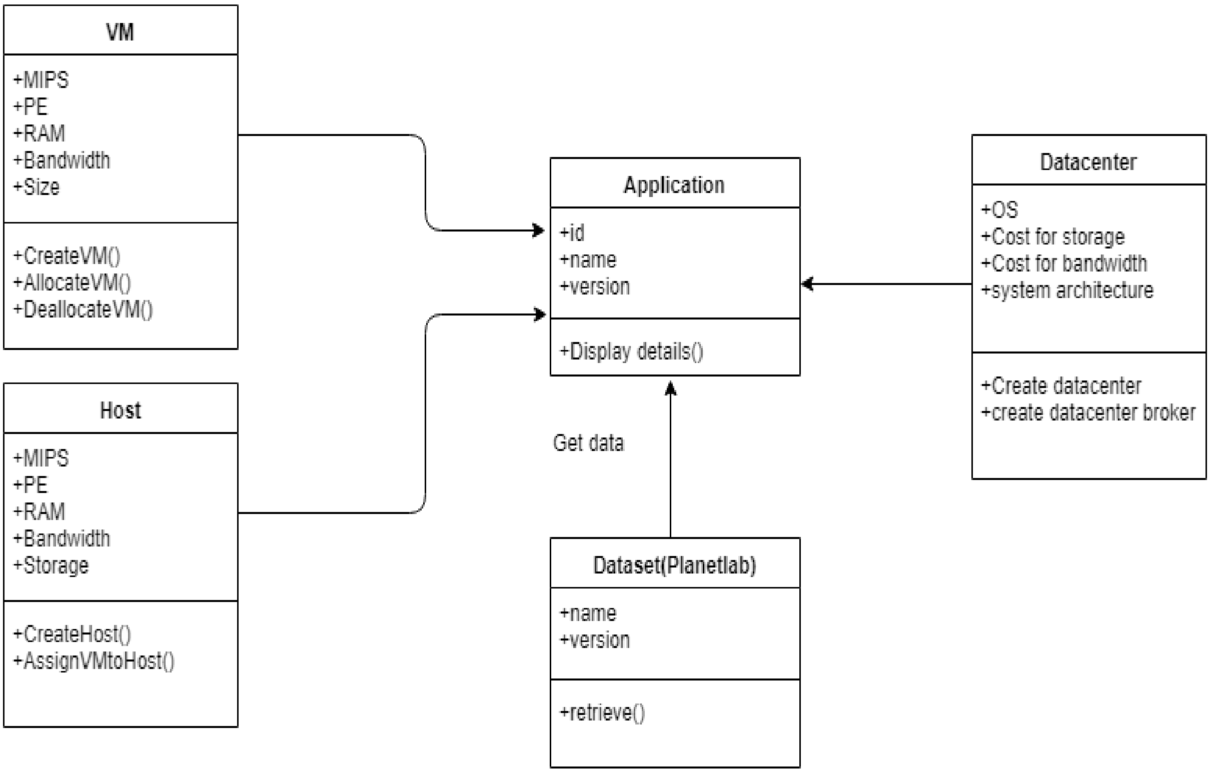


Figure 8: Object-Class Diagram

Figure 8 shows the object class diagram. Based on the important parameters, an object class diagram is made which is necessary for this method. This diagram depicts the different parameters for each element goes to the application where simulation is performed. Key elements are:

- Virtual Machine
- Host
- Dataset
- Data center
- Application

4 Design Specification

Our proposed technique is based on cloudsim framework coded in java language. We are installing cloudsim on Java Eclipse IDE.

4.1 Prototype architecture

We have created a prototype architecture which is integrated with eclipse and tomcat server.

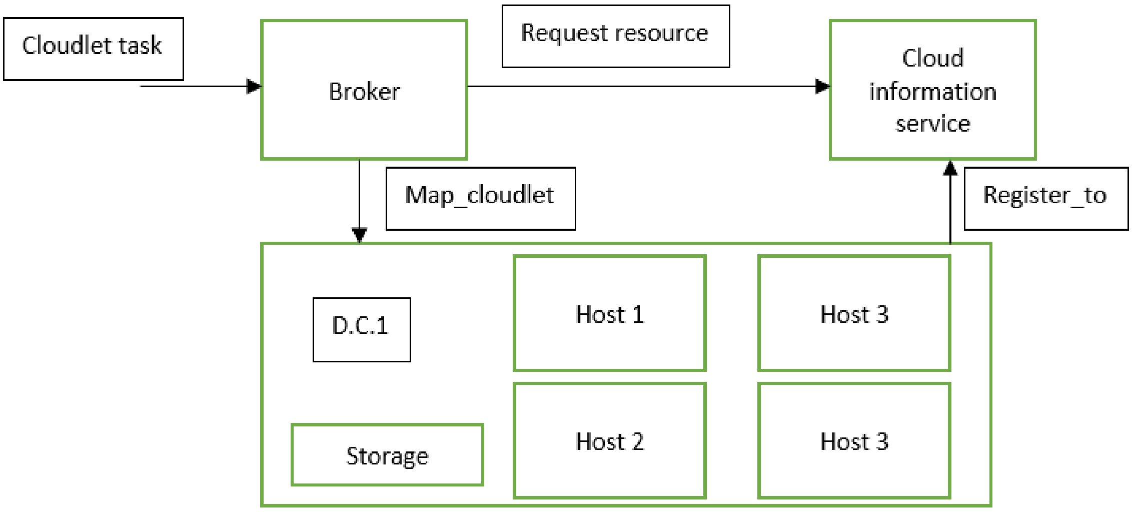


Figure 9: Cloudsim model for proposed method

Figure 9 shows cloudsim model for proposed method. In this model, there are input cloudlet to broker entity. The broker manages and maps all the cloudlet request to the virtual machine. This mapping is done based on available mips (processing power) in each host. All the host information like ram size, processing element size, bandwidth size is maintained in the cloud information service. These details are requested by the broker as and when required for mapping.

Allocation and deallocation of resources to virtual machine is done based on parameters in CIS. New virtual machines are instantiated based on dynamic request from the cloudlet user. Cloudsim allows us to create distributed data center in simulated manner to perform our propose technique feasibility and efficiency.

5 Implementation and Output

In this section, we will demonstrate the implementation part of the method described in section 3 is implemented with the help of following tools:

5.1 Cloudsim toolkit

We are using Cloudsim simulator to create a content delivery network infrastructure on local machine (Benali *et al.*, 2016). With the help of cloudsim we will be able to simulate a scheduler and then perform optimization.

5.2 Eclipse Photon

Eclipse Photon is an integrated developer java coding platform. Cloudsim binaries along with common math library is installed in eclipse photon. This helps us perform required calculation related to QoS penalties.

5.3 Java 1.7

Java 1.7 version is required to install for cloudsim to work properly. This version enables us to code in integrated developer platform(eclipse) and create our own technique which can be simulated on cloudsim platform. This java version is open source and can be downloaded from java website.

5.4 Planetlab cloudsim

Cloudsim provides default data set in its platform to simulate different technique. Also, data from the AWS site related to EC2 instance is fetched.

5.5 Tomcat Server

We are using tomcat server to output the results on localhost browser. Since cloudsim is non-graphical toolkit, we need to output the results to browser for better visualization. For this purpose, we are using tomcat version 9.0.0.M21.

5.6 Front end UI

We have created front end UI to simulate content delivery network environment. Figure below shows the front end of content delivery network in cloud. The number of cloudlet tasks changes randomly with 4 different virtual machines in different location is configured in cloud sim. The cost for each virtual machine is taken from AWS website micro/ small/ medium EC2 instance category.

VM specification (from aws site[3]) –

- Medium Instance: 2 EC2 Compute Units, 4 GB, 0.0416 $ per hour
- Small Instance: 2 EC2 Compute Unit, 2 GB, 0.02 $ per hour
- Micro Instance: 2 EC2 Compute Unit, 1 GB, 0.01 $ per hour

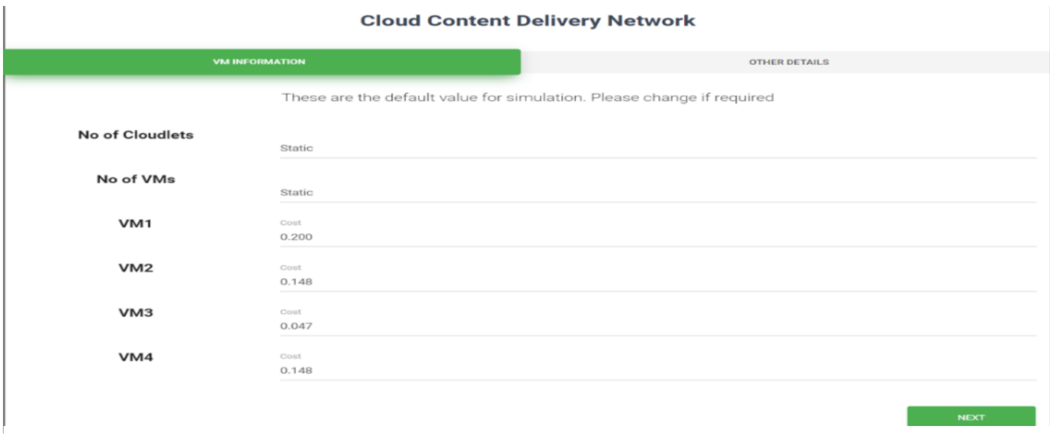


Figure 10: Front end UI

[3] Source: https://aws.amazon.com/ec2/pricing/on-demand/

The output obtained from the input parameters are the number of virtual machines allocated depending on the cloudlet request received to the broker and optimization performed depending on the availability of mips in virtual machine. Based on simulation performed, following table shows the output result:

N	**Simulation 1**	**Simulation 2**
No of hosts	800	800
No of Vm	1052	898
Utilization of energy	522.49 kWh	289kWh
Vm optimized	164524	125026
Deallocated VM	61195	45808

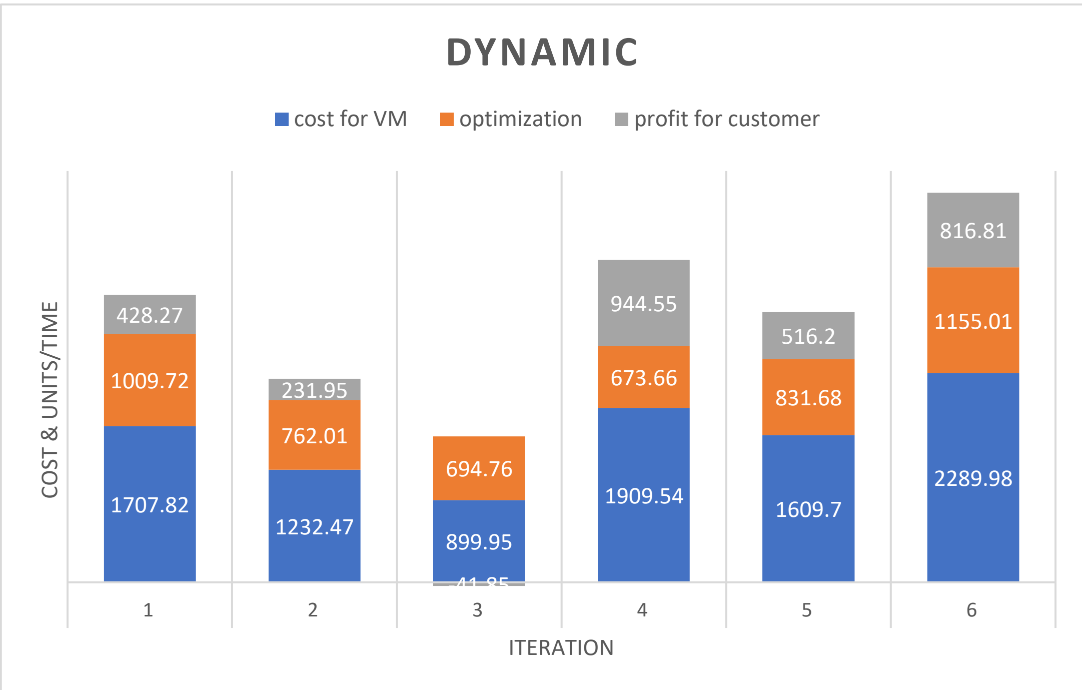
6 Evaluation and Result analysis

In this section, we will evaluate the results from the experiment performed and compare it with other traditional technique.

6.1 Experiment 1

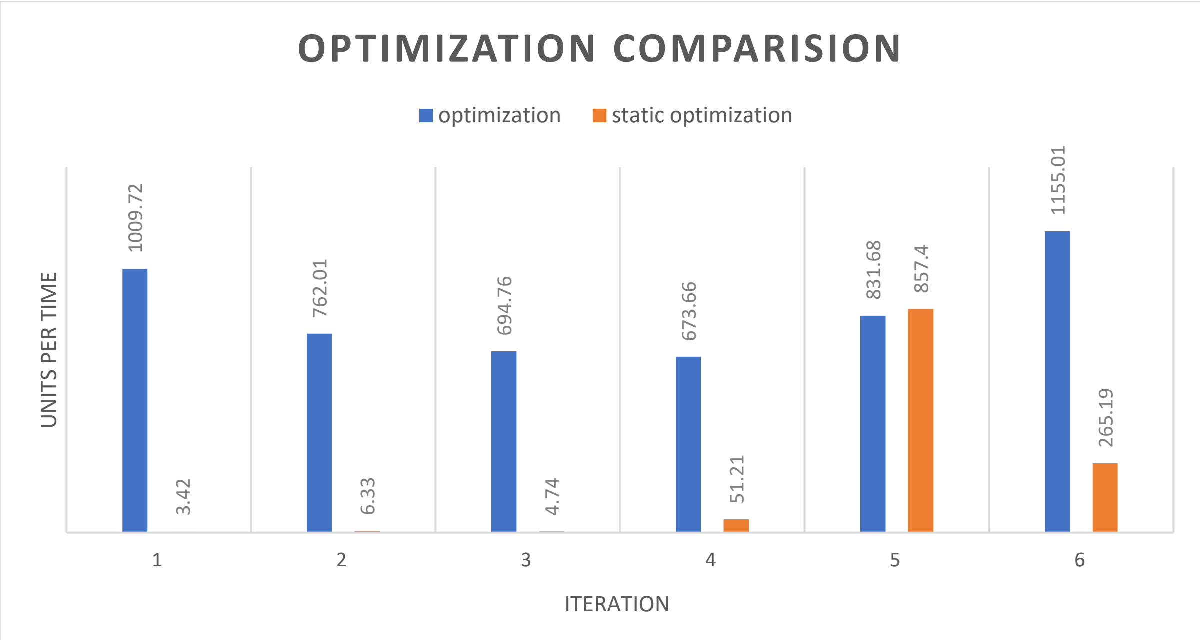
In this experiment, we are performing N = 6 number of iteration for dynamic workloads. As discussed in the subsection 2.4, 3 parameters are taken into consideration that is:

- Utilization of virtual machine
- Optimization factor
- QoS violation factor

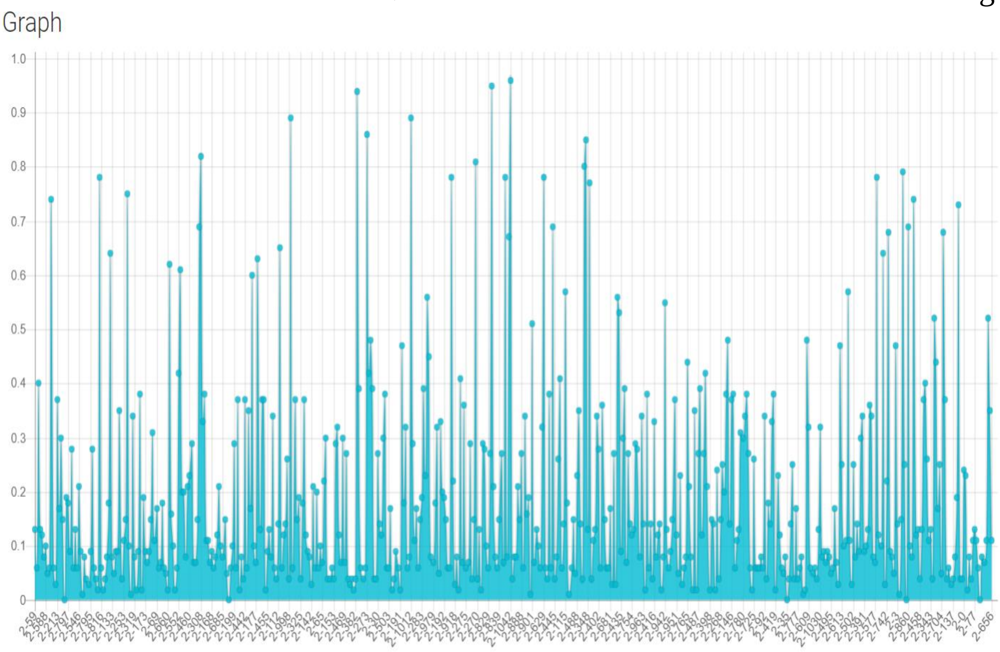


6.2 Experiment 2

In this experiment, we are comparing dynamic technique with the other traditional method like static technique. We observe that the optimization for dynamic exceeds and provides improved results as compared to other techniques. These results were obtained with the help of default dataset of planet lab from cloudsim toolkit. Table 1 shows the optimization results for each technique.

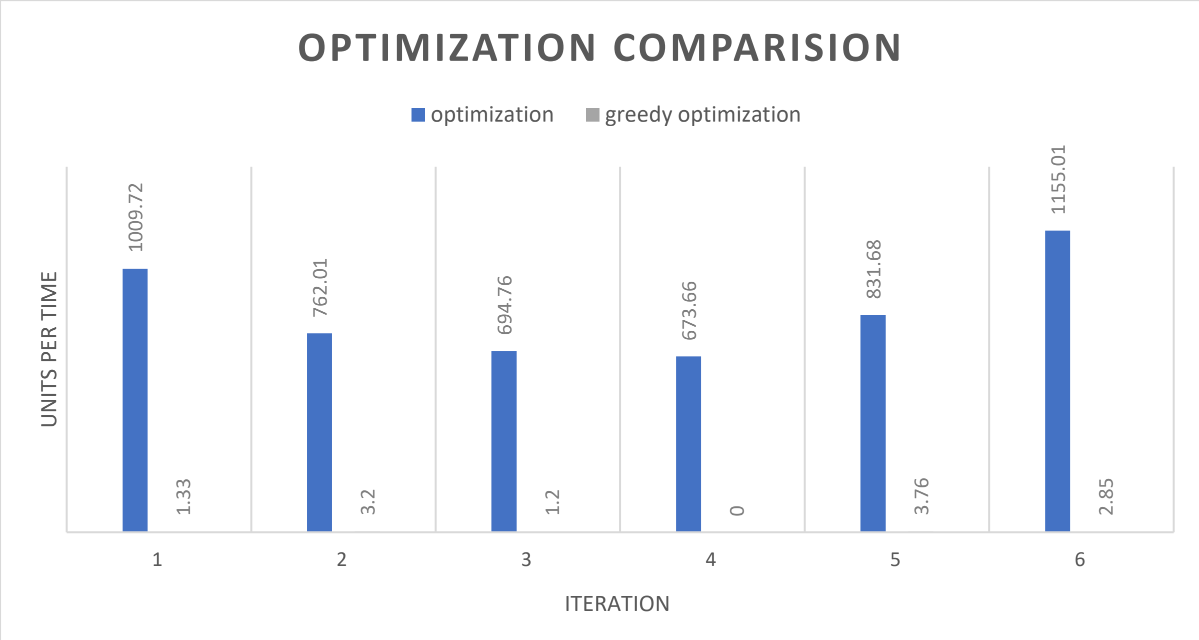


Below graph represents total utilization of resources in each virtual machine. There are total of 1052 elements which are simulated. Each virtual machine utilization is shown in graph.

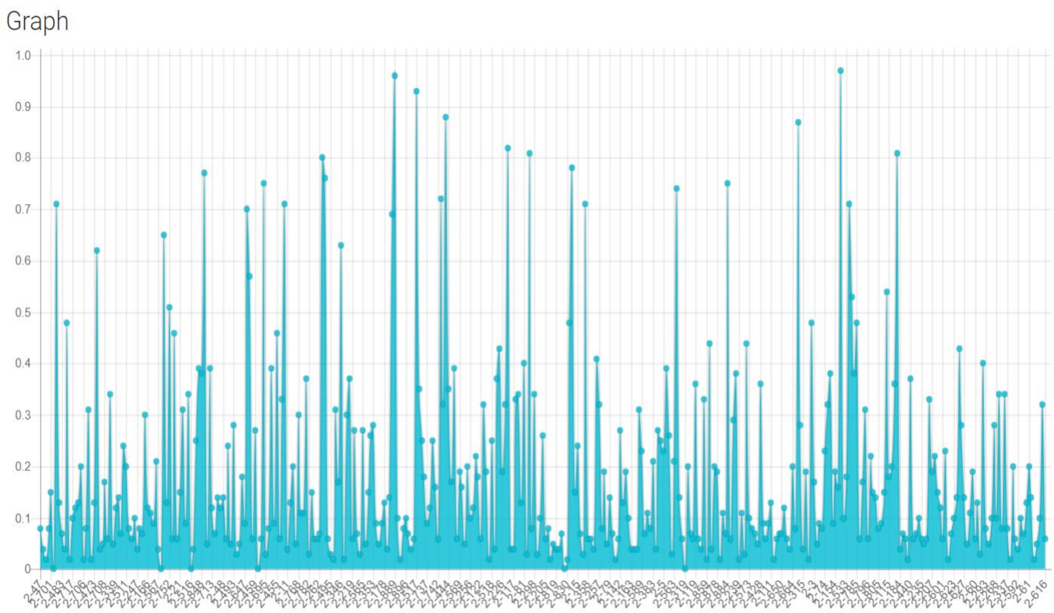


6.3 Experiment 3

We also performed and compared our approach with greedy technique. The results obtained were better than greedy technique. As the dynamic request increases, dynamic technique was able to move more of virtual machine as compared to greedy technique.

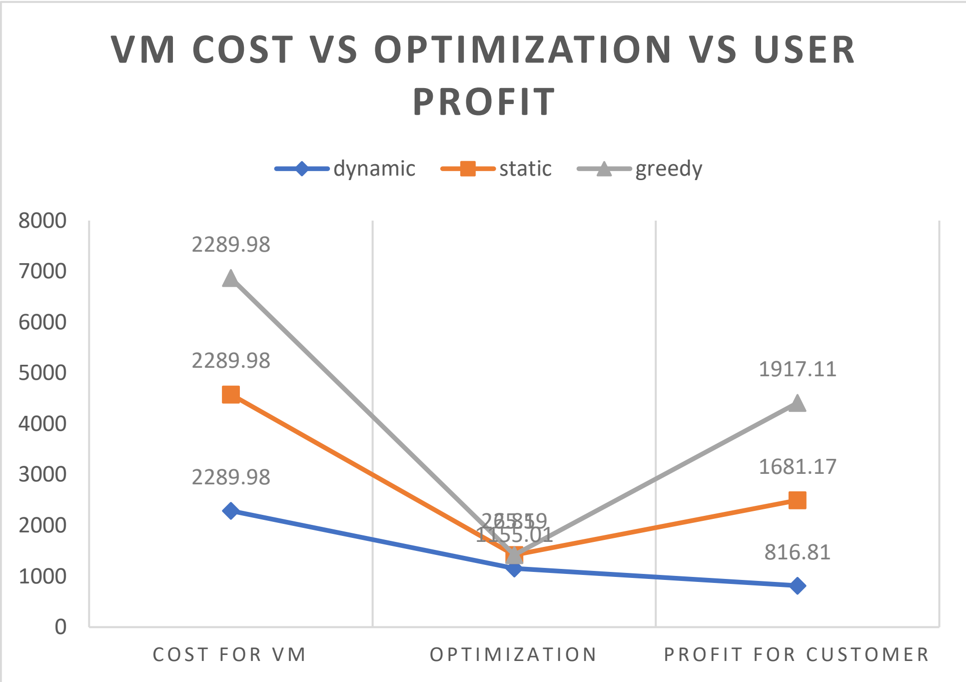


Below graph represents total utilization of resources in each virtual machine. There are total of 898 elements which are simulated.



6.4 Experiment 4

In this experiment, we are evaluating cost for each virtual machine compared to optimization compared to profit for end user. It is observed that the profit for end user in dynamic technique is far less than the greedy and static technique but the optimization for dynamic is better compared to others. This means the quality of service delivered to end user is better with dynamic technique.



6.5 Discussion

With the chosen technique, we were able to create content delivery network in simulated environment and optimize the technique for achieving better quality of service. The results for quality of service using our proposed technique are better as compared to other traditional techniques. This can be seen from the output results in section 6. The design used to implement this work in terms of previous related work was performed and carried out for each technique using similar testbed. Further, private cloud environment like open nebula can be integrated with cloudsim to achieve more realistic results.

7 Conclusion and Future Work

This research is focused on solving the quality of service issue in content delivery network with the help of improved scheduling and optimization techniques. The mixture of different technique used one after another in a single flow can be termed as hybrid approach. The new method not only considers the dynamic demand of the cloudlet user but also considers the distributed environment required for content delivery network. Though profit for customer calculated output in this method is less as compared to other method, still quality of service is achieved to greater extent. This experiment is conducted on cloudsim toolkit and tested on eclipse java IDE. The future work may be done by integration of private cloud like open nebula or openstack with cloud sim to simulate the process in more realistic way. Also cost for service broker can be integrated going forward.

8 Acknowledgement

I express my sincere gratitude and thanks to Prof. Victor Del Rosal for giving constant feedback on my theses topic. He helped me achieve better results by pointing out minute details which helped me complete my proposed idea within the given timeline. I also want to thank Mrs. Krina Shah who helped me with the coding part of the project. I want to thank my family and friends who motivated me and stood with me in tough times. Finally, I want to thank National College of Ireland for providing platform and giving wonderful opportunity to accomplish my dream.

References

Alicherry, M. (2012) 'Film & Theory - Genre.pdf', pp. 963–971. doi: 10.1109/INFCOM.2012.6195847.

Anjum, N. *et al.* (2017) 'Survey on peer-assisted content delivery networks', *Computer Networks*. Elsevier B.V., 116, pp. 1339–1351. doi: 10.1016/j.comnet.2017.02.008.

Benali, R. *et al.* (2016) 'Evaluation of traffic-aware VM placement policies in distributed cloud using Cloud Sim', *Proceedings - 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2016*, pp. 95–100. doi: 10.1109/WETICE.2016.29.

Broberg, J., Buyya, R. and Tari, Z. (2009) 'MetaCDN: Harnessing "Storage Clouds" for high performance content delivery', *Journal of Network and Computer Applications*. Elsevier, 32(5), pp. 1012–1022. doi: 10.1016/j.jnca.2009.03.004.

Chen, F. *et al.* (2012) 'Intra-cloud lightning: Building CDNs in the cloud', *Proceedings - IEEE INFOCOM*, pp. 433–441. doi: 10.1109/INFCOM.2012.6195782.

Haghighi, A. A., Shah Heydari, S. and Shahbazpanahi, S. (2017) 'Dynamic QoS-Aware Resource Assignment in Cloud-Based Content-Delivery Networks', *IEEE Access*, 6, pp. 2298–2309. doi: 10.1109/ACCESS.2017.2782776.

Herbaut, N. *et al.* (2016) 'Content delivery networks as a virtual network function: A win-win ISP-CDN collaboration', *2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings*. doi: 10.1109/GLOCOM.2016.7841689.

Hu, H. *et al.* (2016) 'Joint content replication and request routing for social video distribution over cloud CDN: A community clustering method', *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7), pp. 1320–1333. doi: 10.1109/TCSVT.2015.2455712.

Ibn-Khedher, H. *et al.* (2017) 'OPAC: An optimal placement algorithm for virtual CDN', *Computer Networks*. Elsevier B.V., 120, pp. 12–27. doi: 10.1016/j.comnet.2017.04.009.

Ben Jemaa, F., Pujolle, G. and Pariente, M. (2016) 'QoS-aware VNF placement optimization in edge-central carrier cloud architecture', *2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings*. doi: 10.1109/GLOCOM.2016.7842188.

Jia, Q. *et al.* (2017) 'The Collaboration for Content Delivery and Network Infrastructures: A Survey', *IEEE Access*, 5, pp. 18088–18106. doi: 10.1109/ACCESS.2017.2715824.

Jiang, W. *et al.* (2015) 'QoS-aware dynamic spectrum resource allocation scheme in C-RAN based dense femtocell networks', *2015 International Conference on Wireless Communications and Signal Processing, WCSP 2015*. IEEE, pp. 1–6. doi: 10.1109/WCSP.2015.7341016.

Kömer, P., Abraham, A. and Snášel, V. (2014) 'Hybrid Job Scheduling Algorithm for Cloud Computing Environment', *Advances in Intelligent Systems and Computing*, 303, pp. 43–52. doi: 10.1007/978-3-319-08156-4.

Krishnan, S. S., Sitaraman, R. K. and Systems, A. C. M. S. A. C. M. S. U. A. T. C. (2012) 'Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs', *2012 ACM Internet Measurement Conference, IMC 2012*, 21(6), pp. 211–224. doi: 10.1145/2398776.2398799.

Kumar, D. and Raza, Z. (2015) 'A PSO based VM resource scheduling model for cloud computing', *Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015*. IEEE, pp. 213–219. doi: 10.1109/CICT.2015.35.

Kumar, N. *et al.* (2014) 'Achieving quality of service (QoS) using resource allocation and adaptive scheduling in cloud computing with grid support', *Computer Journal*, 57(2), pp. 281–290. doi: 10.1093/comjnl/bxt024.

Laha, D. and Chakraborty, U. K. (2009) 'A constructive heuristic for minimizing makespan in no-wait flow shop scheduling', *International Journal of Advanced Manufacturing Technology*, 41(1–2), pp. 97–109. doi: 10.1007/s00170-008-1454-0.

Lai, C. F., Hwang, R. H. and Chao, H. C. (2017) 'A QoS Aware Resource Allocation Strategy for Mobile Graphics Rendering with Cloud Support', *IEEE Transactions on Circuits and Systems for Video Technology*, 27(1), pp. 110–124. doi: 10.1109/TCSVT.2016.2589740.

Madni, S. H. H. *et al.* (2016) 'Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities', *Journal of Network and Computer Applications*, 68, pp. 173–200. doi: 10.1016/j.jnca.2016.04.016.

Niu, D. *et al.* (2012) 'Quality-assured cloud bandwidth auto-scaling for video-on-demand applications', *Proceedings - IEEE INFOCOM*, (May 2014), pp. 460–468. doi: 10.1109/INFCOM.2012.6195785.

Sahoo, J. *et al.* (2017) 'A Survey on Replica Server Placement Algorithms for Content Delivery Networks', *IEEE Communications Surveys and Tutorials*, 19(2), pp. 1002–1026. doi: 10.1109/COMST.2016.2626384.

Son, S., Jung, G. and Jun, S. C. (2013) 'An SLA-based cloud computing that facilitates resource allocation in the distributed data centers of a cloud provider', *Journal of Supercomputing*, 64(2), pp. 606–637. doi: 10.1007/s11227-012-0861-z.

Srinivasan, S. *et al.* (no date) 'ActiveCDN : Cloud Computing Meets Content Delivery Networks', *IEEE Computer Society*.

Stocker, V. *et al.* (2017) 'The growing complexity of content delivery networks: Challenges and implications for the Internet ecosystem', *Telecommunications Policy*. Elsevier Ltd, 41(10), pp. 1003–1016. doi: 10.1016/j.telpol.2017.02.004.

Walkowiak, K. (2005) 'QoS Dynamic Routing in Content Delivery Networks', *Networking*, 346(Lectures in Computer Science), pp. 1120–1132.

Wang, F., Liu, J. and Chen, M. (2012) 'CALMS: Cloud-assisted live media streaming for globalized demands with time/region diversities', *Proceedings - IEEE INFOCOM*, pp. 199–207. doi: 10.1109/INFCOM.2012.6195578.

Wei, L. *et al.* (2017) 'QoS-Aware Resource Allocation for Video Transcoding in Clouds', *IEEE Transactions on Circuits and Systems for Video Technology*, 27(1), pp. 49–61. doi:

10.1109/TCSVT.2016.2589621.

Xu, K. *et al.* (2018) 'Joint Replica Server Placement, Content Caching, and Request Load Assignment in Content Delivery Networks', *IEEE Access*, 6, pp. 17968–17981. doi: 10.1109/ACCESS.2018.2817646.

Zhang, Q. *et al.* (2013) 'Dynamic service placement in geographically distributed clouds', *IEEE Journal on Selected Areas in Communications*, 31(12), pp. 762–772. doi: 10.1109/JSAC.2013.SUP2.1213008.

Configuration Manual

Ateet Jayeshkumar Shah
Student ID: x17127785

9 Introduction

The configuration manual will help academic students and other researchers understand the detailed method involved in implementing this research project. It is a manual guide with detailed steps from setting testbed, designing method, code for method.

10 Setting up the testbed

Pre-requisite software and hardware requirements are required.

10.1 Hardware specification

CPU	Intel 7700HQ 2.28Ghz
RAM	16 GB

10.2 Software specification

OS	Windows 10
Tomcat server	9.0.0.M21
Cloudsim	3.0.3
Java	1.7
Eclipse	photon
Common math library	Apache

10.3 Data set

We are using inbuilt data set – planet lab from cloudsim. This data set contains dynamic values which are necessary for our implementation.

10.4 Installing software

- Downloading cloudsim[4] and extracting it to a folder in D drive in windows operating system

[4] Source: https://github.com/Cloudslab/cloudsim/releases

- Downloading java eclipse juno[5] and installing it.
- Downloading common math library[6] and installing it in cloudsim folder.
- Download and install tomcat server[7] for localhost.

11 Designing new method

Our approach consists of mainly 2 stage.

Stage 1 is allocation and deallocaton and Stage 2 is optimization

Default VM allocator in cloudsim is static. We are changing this allocator with our own dynamic allocator.

Comparing the output of dynamic technique with other method like static and greedy. (GitHub, 2018).

12 Code for method

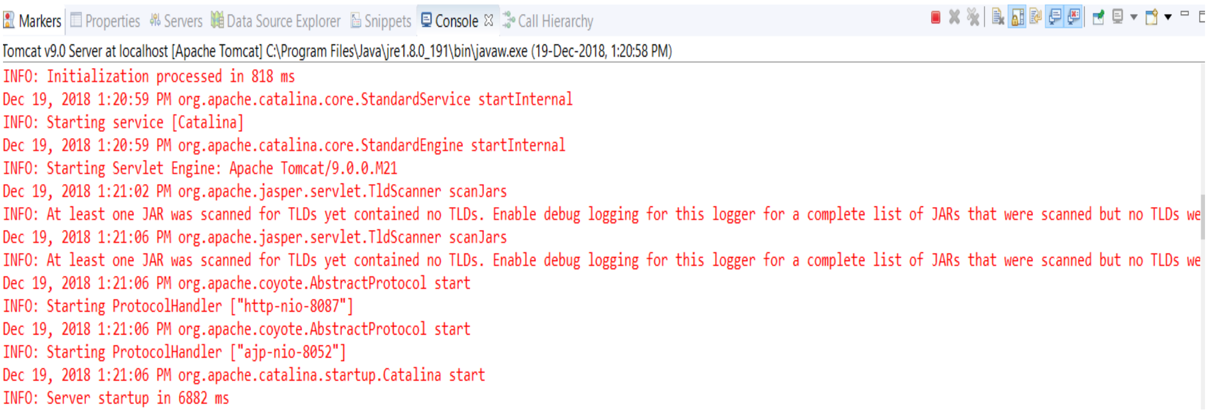
Below shows the code for dynamic VM allocation policy –

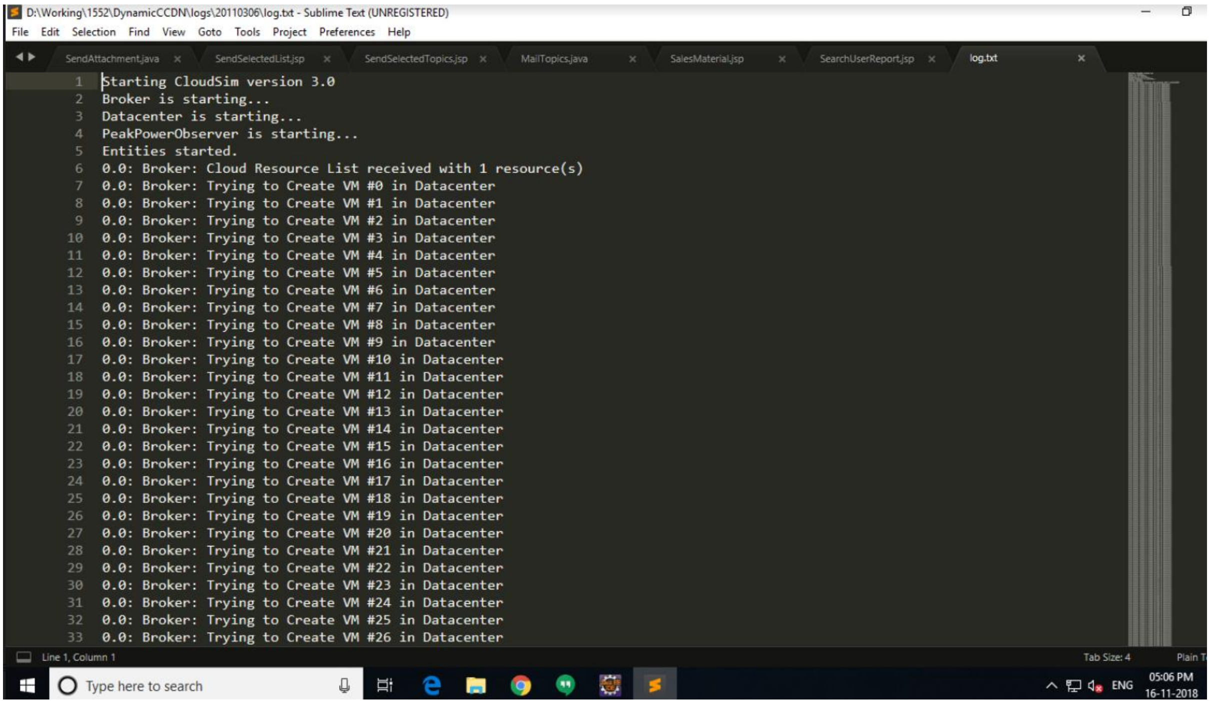
```
NewAllocationPolicy vmAllocationPolicy = new NewAllocationPolicy();

Datacenter dc = new Datacenter (String name, DatacenterCharacteristics characteristics, vmAllocationPolicy(hostList), List<Storage> storageList, double schedulingInterval)
```

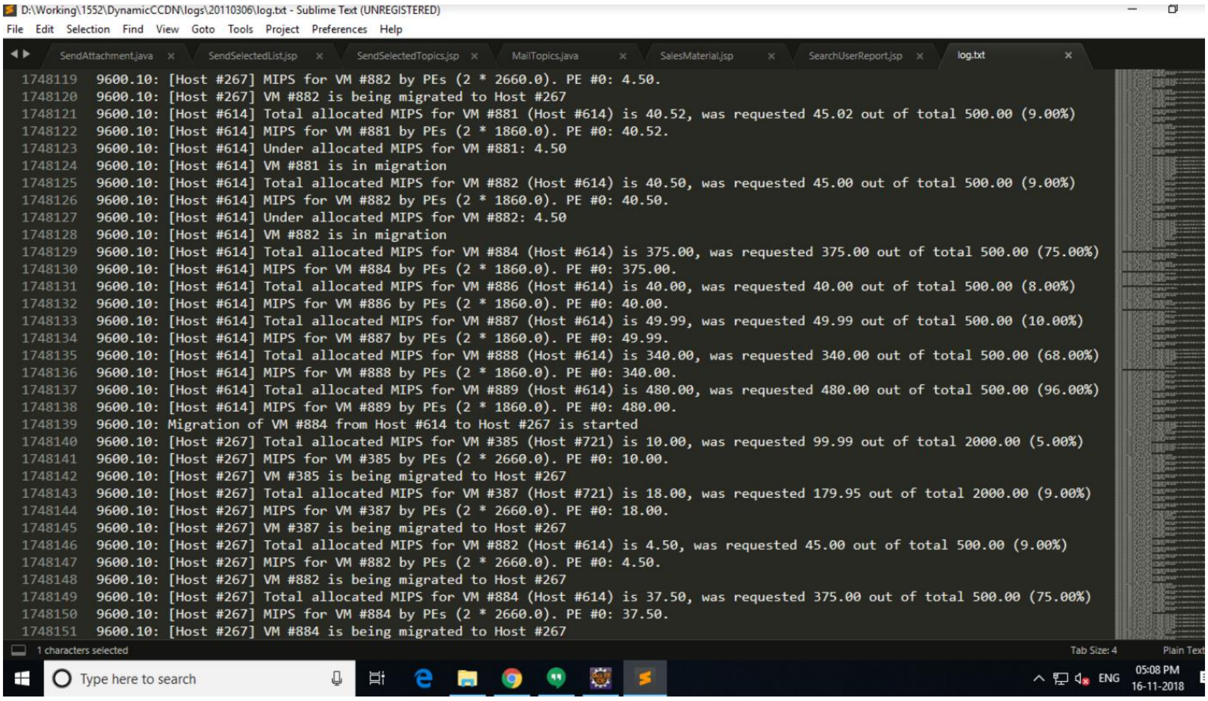
```
    public boolean allocateHostForVm(Vm vm, Host host) {
        if (host.vmCreate(vm)) {
            //the host is appropriate, we track it
            vmTable.put(vm.getUid(), host);
            return true;
        }
        return false;
    }

    public boolean allocateHostForVm(Vm vm) {
        //sort the hosts according to the available mips present
        Collections.sort(getHostList(), new Comparator<Host>() {
            @Override
            public int compare(Host h1, Host h2) {
                return (int)(h1.getAvailableMips() - h2.getAvailableMips());
            }
        });

        //allocate the vm to host
        for (Host h : getHostList()) {
            if (h.vmCreate(vm)) {
                //track the host
                vmTable.put(vm.getUid(), h);
                return true;
            }
        }
        return false;
    }

    public void deallocateHostForVm(Vm vm,Host host) {
        vmTable.remove(vm.getUid());
        host.vmDestroy(vm);
    }
```

[5] Source: https://www.eclipse.org/downloads/packages/release/luna/r/eclipse-ide-java-developers
[6] Source: http://commons.apache.org/proper/commons-math/
[7] Source: https://tomcat.apache.org/download-90.cgi

Below are the logs for reference –







References

GitHub. (2015). *mourjo/VMScheduling*. [online] Available at: https://github.com/mourjo/VMScheduling/blob/master/src/main/java/fr/unice/vicc/GreedyVmAllocationPolicy.java [Accessed 19 Dec. 2018].

Tomcat.apache.org. (2018). Apache Tomcat® - Apache Tomcat 9 Software Downloads. [online] Available at: https://tomcat.apache.org/download-90.cgi [Accessed 19 Dec. 2018].