An Enhanced Approach to QoS-Aware Resource Provisioning and Job Scheduling in Cloud Computing

by

Md. Modabbir Rahman Khan

ID: 12101132

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BRAC UNIVERSITY

April 2017

Declaration

I, hereby, declare that the work presented in this project is the outcome of the investigation performed by me under the supervision of Md. Shamsul Kaonain, Lecturer, Department of Computer Science and Engineering, BRAC University, and Tamal Adhikary, Former Lecturer, Department of Computer Science and Engineering, BRAC University, and I also declare that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

. . . .

(Md. Shamsul Kaonain)

Supervisor

. . . .

(Tamal Adhikary)

Co-Supervisor

Signature

. . . .

(Md. Modabbir Rahman Khan)

Acknowledgements

First of all, I am thankful and express our gratefulness to the Almighty who bestowed upon me His divine blessings, patience, mental and psychical strength to complete this thesis work.

I am deeply indebted to my thesis supervisor Md. Shamsul Kaonain, Lecturer, Department of Computer Science and Engineering, BRAC University and co-supervisor Tamal Adhikary, Lecturer, Department of Computer Science and Engineering, BRAC University. Their scholarly guidance, important suggestions, endless patience, constant supervision, valuable criticism, and enormous amount of work for going through the drafts and correcting them, and generating courage from the beginning to the end of the research work has made the completion of the thesis possible.

Last but not the least; I am highly grateful to my parents and family members for their support and constant encouragement, which have always been a source of great inspiration for me.

Md. Modabbir Rahman Khan

April, 2017

Table of Contents

List of Figures

Abstract

Cloud computing is an emerging technology provided over the internet in which resource provisioning (RP) and job scheduling (JS) are two of the most important issues. The goal of RS and JS is to accomplish better QoS with highest possible use of assets in one cloud domain. To achieve better QoS in SLA, an enhanced algorithm has been proposed here which considers all parameters of a certain job and maximize the resource utilization with proper scheduling of the jobs given.

Chapter 1

Introduction

1.1 Introduction to Cloud Computing

Matching paces with the advancement of time, the technological progression has escalated even more. The novelty of the usage of personal computer in business purposes has long since been outdated by the ever-proliferating demand of more and more computing power. The skyrocketing demand has made it absolutely necessary for a new system called 'cloud computing' to be established. Cloud computing, basically, indicates to the usage of computational power which is not situated in the immediate vicinity of the person using it, neither does the said person have any proprietary right over it. In simpler terms, a person or party, when in need of a certain amount of computational power, is either unable or unwilling to directly invest and purchase it, and decides to resort to using shared assets instead, uses cloud computing. The concept of cloud computing has lately become immensely popular all over the world. To both individuals and collective interests, it has proved itself to be of enormous help. By providing clients access to computational power online that can be used in any way the clients wish, it has gained much and rapid popularity. One of the greatest advantages of cloud computing is that it does not require its client to worry about the infrastructure or the maintenance. Especially for clients who frequently need different types of hardware configuration for the computation, cloud computing saves a massive amount of expense, as by resorting to this system the clients can easily avoid the purchase of mechanical equipment and the additional costs that invariably accompany it.

The fact that the popularity of cloud computing is travelling an ever-increasing curve can be witnessed in the amount of services that are being requested to the cloud servers. Since cloud computing has made it possible for multiple users to perform their tasks simultaneously, clients from all over the planet have started requesting for cloud resources to be allotted to them. This upsurge in the number of client requests has forced the service providers to consider the question of the proper distribution of viable resources. As the number of resource allotment request has gone up, users have experienced numerous troublesome aspects of this system. With the increasing load of users on the cloud servers, it has become quite cumbersome for them to maintain instant, continuous and secure access at the same time, even at a significant expense.

Contemplation of the various issues the cloud computing system, however, has revealed that the issue which stands out the most and poses to be the biggest obstacle today is the dilemma of Job Scheduling. Job Scheduling, in short, means the system that determines how to distribute the service based on priority, cost and such other issues. It is constantly observed that choosing which of the client ought to be allotted sooner than others and how much cost and how much benefit is being created from that specific client or occupation proves a tremendously difficult task. There could be numerous occupations with same or distinctively unique needs, diverse or same entry times, less or more costs, and, furthermore, the extent that needs to be gone to in order to complete the employment. Presently, in a scenario which contains just one single line of employment but various sorts of occupations—which, individually, require distinctive asset allotment—the VM provisioning is noted to encounter a number of complications. On the off chance when we offer altered asset for each sort of occupation, it may transpire that some employment will end up getting a lot of asset for working. This way a few assets would be squandered, and, in addition, a few occupations would be unable to lay their hands upon legitimate assets to finish the errands properly. Considering the following components — entry times, due dates set by the clients, costs by assets, boosting advantage for the CSP (Cloud Service Provider) — we are generating another calculation which will figure out the best occupation for any particular minute, and will upgrade the Job booking line each time another employment request shows up.

Figure 1.1.1: Cloud data center

1.2 Thesis Contributions

In our work, we proposed an efficient algorithm which will serve the jobs properly maintaining all the constraints and also Quality of Service(QoS). We have implemented the algorithm on the CloudSim toolkit which has given the output of the best job sequence to be served properly by CSP. In order that the requirements of the users could be fulfilled, while propelling the utilization of the resources towards the utmost without losing the sustenance of the quality of service, the Job Scheduling Algorithm was invented. Till date, a number of job scheduling algorithms have been devised to better the results.

1.3 Thesis Orientation

The rest of the report is organized as follows:

- Chapter 2 discusses on our motivation and background studies of our proposed approach algorithm.
- Chapter 3 discusses about the main problem definition on which basis we have built our algorithm.
- Chapter 4 has overviewed the whole proposed model with necessary diagrams.
- Chapter 5 describes about the performance evaluation and experimental result analysis to determine the proposed model's efficiency.
- Chapter 6 will conclude this paper with some word on the future scope regarding this research.

Chapter 2

Motivation and Background Studies

2.1 Introduction

The discovery of cloud computing, as can easily be inferred, has given birth to numerous difficulties as it has spewed out infinite possibilities, the most crucial of which being the impediment of Job Scheduling. Determining which user to serve first based on their requirements, as well as the available resources, had demanded immediate attention. In satisfying the clients of cloud computing it was essential that the demands of the clients were met. But this obviously could not be done if the entity serving lacked proper resources. Also, if these issues were given too much priority, the standard of the service being given went down significantly, which, in turn, discouraged the clients to depend on the usage of cloud computing technology. So, what followed was the proposition of an algorithm that took into account all the three issues mentioned — client requirements, maximum resource utilization, and maintenance of the quality of service. Thus, the Resource Provisioning and Job Scheduling Algorithm came to being.

2.2 Related Works

Developing this field, much has been done on, especially, the matter of waiting time variance. For long, extensive research [6] has gone on, largely maintaining the focus on the minimization of the waiting time variance on parallel machines. In achieving this, efforts were fundamentally directed towards determining the optimal value of the time variance of jobs. One method that has been tried and has met much success is a Heuristic Algorithm. This algorithm, the functions of which correspond to the job scheduling time variance, is basically a FCFS (First Come First Serve) system. Although appearing as a flawless method to be incorporated, the FCFS system has proven to disregard the priority of the job, hence spending resources after insignificant ones while the crucial jobs are put on queue. This intends to introduce a procedure that lacks the said limitation.

Another approach that has been attempted was a prototype that handled the operation mostly based on the Service Level Agreement (SLA). This prototype [8] , in fact, reached the results it produced taking decisions solely on the basis of SLA. The method which it operated in was almost entirely devoid of consideration of the matter of Quality of Service (QoS). As the quality maintained to provide the service is of immense importance, a good approach should definitely keep this in consideration and pay sufficient attention to it. In this approach, the focus will be directed to alluding to it so that the quality of service is efficiently maintained.

Another one among the more remarkable approaches made use of the Harmony Search algorithm. The HM algorithm [9] has found a rather popularity and is being used in various systems to find the optimal solution. The process of the HM algorithm is quite simple: it initially generates an HM memory with random solutions, creates a new harmony from the previous HM and checks if the newly generated one is better; if it is, it overrides the previous one. This, although quite unique and interesting,. has been proven a method a little too expensive to resort to. By generating harmonies repeatedly, a vast amount of resource is expended, which violates both the Service level Agreement and the Quality of Service.

Furthermore, it takes too much time, making the method virtually invalid for many scenarios. In this proposed model, this limitation has been surpassed so that the clients can be served with more efficiency in a notable short amount of time.

Chapter 3
Problem Definition

3.1 Introduction

As much as might have been done in the sector of cloud computing, there have always been difficulties or limitations. Of course, various ideas and solutions have followed those obstacles in no time. In this paper, describing the proposed model, the specific problem of making job scheduling more efficient has been resolved through the usage of resource provisioning, which, it should be noted, has never been done before.

3.2 Problem Definition

A standout amongst the most widely recognized and basic issue in cloud benefit giving is Job Scheduling. It is constantly hard to choose which of the client solicitations ought to be served sooner than others and how much cost and benefit are being created from that specific client or occupation. There could be numerous employments with same or diverse need, shifting entry time of solicitations, less or more cost furthermore to what extent it will take to complete the occupation. Presently if there is just a single line of occupations and there are various sorts of employments which requests distinctive asset prerequisites, there cloud benefit suppliers confront issues with VM provisioning. In the event that we offer settled asset for each sort of employment, it happens that some occupation is getting an excessive amount of asset than prerequisite for working along these lines a few assets would be squandered and additionally a few employments would not get legitimate assets to finish the undertakings properly. Considering components entry time, client given due date, cost by asset, augmenting advantage for the CSP(Cloud Service Provider) we are giving an algorithm which will pick the best employment for that specific time and it will upgrade the Job booking line each time another occupation asks for showing up.

Chapter 4

Proposed System Model

4.1 Introduction

The purpose of job scheduling is to determine which job should be given priority from among a ques full of jobs of varying aspects. In this proposed model, that particular predication is taken care of by using a simulation toolkit called 'CloudSim'. The fundamental idea of this model is to figure out the priority of a job by performing calculation using certain parameters of the requested jobs. Data centers containing multiple hosts comprised of multiple virtual machines will be utilized for this purpose to execute the task.

4.2 What Is CloudSim?

Although cloud computing has proved much useful a system, it has also presented the users with a number of limitations. The most prominent problems in the way of getting the most out of cloud computing system appear to be those faced in development — that is to say, the obstacles that are encountered while testing, debugging, and in such experimentations. For example, someone attempting to perform the simplest of operation using cloud computing will have to go through the entire process from acquisition of resources to the maintenance of the system, taking care of the installation and configuration and such tasks in between. Although these have been made much easier for the users, there are still traces of troubles left which make it almost impossible for the user to perform repetitive operations without spending a huge amount of monetary and other resources.

Predicaments of such nature are resolved using virtual simulations of the cloud computing system in which an user can very easily perform any task that was supposed to be performed in real life, but without all the trouble concerning the infrastructure and services of the cloud system. By providing a virtual setup where executing the same task is just as easy yet is free of all the additional and unnecessary inconveniences, the idea of simulation for cloud computing has become immensely popular among various entities these days. CloudSim, such a simulation framework, provides a generalized and extensible framework to achieve this. For seamless modelling and app performance simulation, CloudSim provides a much useful platform that lets the developers work on specific system design issues and work gracefully.

4.3 Proposed System Model

1. For the proposed model, a cloud toolkit will be used that would help make the tasks execute more easily. Known by the name CloudSim, this toolkit, or framework, will contain several built-in packages and classes. For each data center, there will be multiple hosts, which, in turn, will contain multiple virtual machines. The source code will be accessible to all (being open source), so it will become possible to implement algorithms mentioned in this paper.
2. Once the proposed system receives multiple job requests, it will put them in a priority queue.
3. Each job will have a number of certain parameters.
4. The jobs listed in the priority queue will be sorted according to the level of priority they are assigned.

5. After that, the resource need calculation will be done in order to finish the scheduling of the jobs.
6. Then all the jobs would be checked if they are compatible with the Vm in a certain Datacenter, if yes then it will be assigned to a Vm and if not then it will pass them to another datacenter.

$$R_j = \partial \frac{Lj}{Max(Lj)} + \partial \frac{Mj}{Max(Mj)} + \partial \frac{Pj}{Max(Pj)} + \partial \frac{Bj}{Max(Bj)}$$

Return *Rj*

Figure 4.3.1

```
For each j in List(j):
    If (CheckCompatability() == true):
        PQ.Add(j)
    Else:
        Send to other dataCenter
    End If
```

Figure 4.3.2

```
Def CheckCompatibility(Job j):

        If (Size(V_m)>=Size_j && memory(V_m)>=M_j
&&
processing(v)>=P_j && Bandwidth(V_m)>=B_j):

                If (Vs = Φ):
                        Return true
                Else:
                        Return false
                End if
        End if
```

Figure 4.3.3

```
Def  VmAllocation(List<Vm>):
     for each j in PQ:
             Calculate Wj ← Wj / Max (Wj)
              Normalize (Rj)
              Rj ← Rj X Wj
     End for
For each V in Vs:
     If (V.isEmpty() == true):
     For each j in PQ:
          If (CheckCompatability(j) == true):
                         j.AllocateToVm (V)
         end if
       end for
end for
```

Figure 4.3.4

Chapter 5

Performance Evaluation

5.1 Introduction

This chapter contains the results that were obtained after several virtual machines were set up inside the CloudSim simulation and, by creating a number of jobs, the proposed system was tested. For the fabrication in the simulation, a few values were assumed, and the experimentation went on accordingly.

5.2 Simulation Environment Setup

A CloudSim simulation was set up in an Eclipse IDE and was run. Five random jobs were initialized, each having different time values and different arrival times. Every job requires some particular resources. Also, each job contains different parameters.

Jobs	PE Required / / Resource Needed	Priority	Waiting time Minutes	Memory requirement MB	Bandwidth Requirement	MIPS
J1	1	5	20	2000	200	250
J2	2	4	30	2000	200	300
J3	3	3	30	2000	200	250
J4	4	2	40	2000	200	300
J5	5	1	40	2000	200	250
J6	6	6	10	2000	200	300
J7	7	7	20	2000	200	200
J8	8	2	10	2000	200	200
J9	9	1	20	2000	200	200
J10	10	10	50	2000	200	200

Figure 5.2.1

5.3. Simulation Result

From the simulation, we have found that by using our proposed broker model a significant performance improvement is achieved . The comparative study using graph is discussed below.

```
<terminated> Scheduling [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Apr 17, 2017, 1:01:09 PM)
Starting Scheduling...
Initialising...
cloudlest :org.cloudbus.cloudsim.Cloudlet@5c647e05cloudlets :org.cloudbus.cloudsim.Cloudlet@339097525tarting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.0: Broker: Trying to Create VM #4 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #1
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: VM #3 has been created in Datacenter #2, Host #1
0.1: Broker: VM #4 has been created in Datacenter #2, Host #0
0.1: Broker: VM #5 has been created in Datacenter #2, Host #1
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
82.1: Broker: Cloudlet 1 received
160.1: Broker: Cloudlet 0 received
160.1: Broker: All Cloudlets executed. Finishing...
160.1: Broker: Destroying VM #0
160.1: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all cloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0        SUCCESS        2            1        80       0.1        80.1
    1        SUCCESS        2            0        160       0.1        160.2
    2        SUCCESS        2            3        50       10.1        60.1
    3        SUCCESS        2            4        40       20.2        60.2
    4        SUCCESS        2            5        60       30.2        90.2
    5        SUCCESS        2            2        70       30.4        100.4
Scheduling finished!
```

Figure 5.3.1

Cloudlet ID	Incoming Time	Data Center	VM ID	Burst Time	Start Time	Finish time
1	0	2	1	80	0.1	80.1
2	0	2	0	160	0.1	160.2
3	10	2	3	50	10.1	60.1
4	10	2	4	40	20.1	60.1
5	20	2	5	60	30.2	90.2

Figure 5.3.2

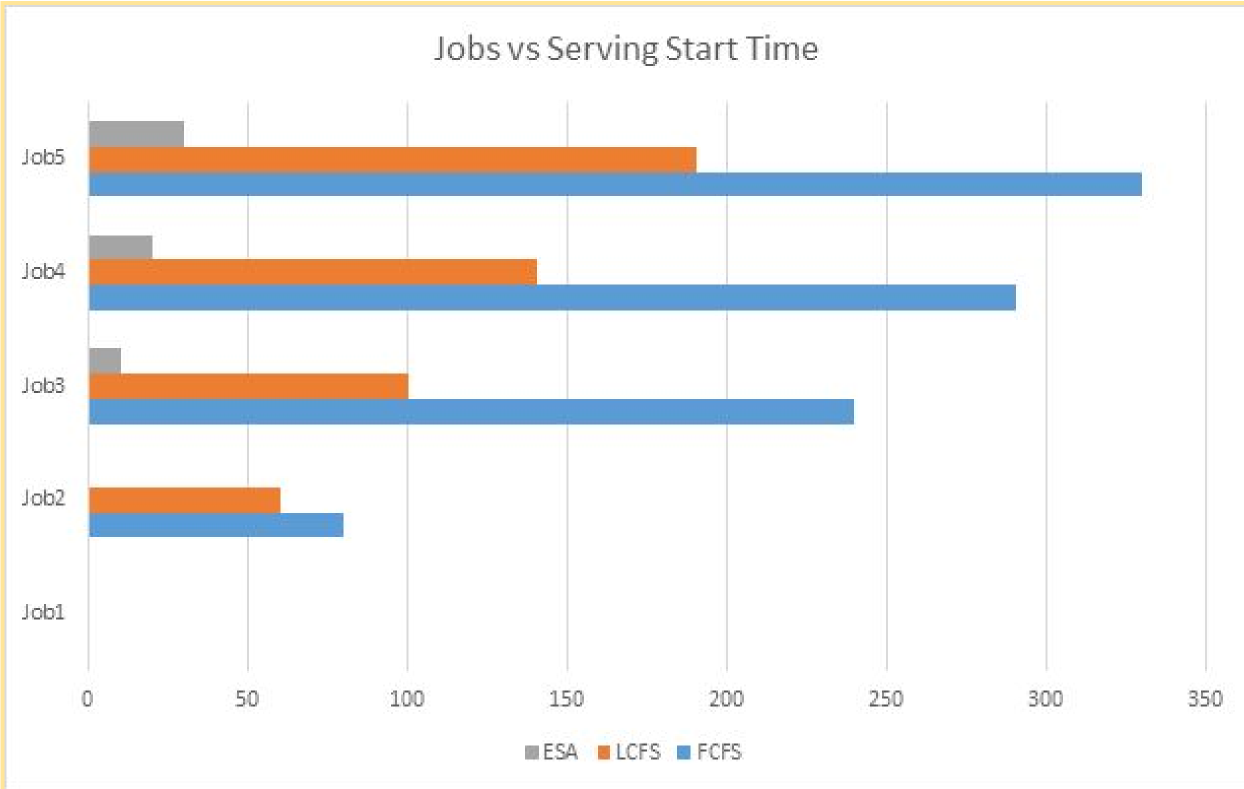
Jobs vs Serving Start Time
Job5
Job4
Job3
Job2
Job1
0
50
100
150
200
250
300
350
ESA
LCFS
FCFS

Figure 5.3.3

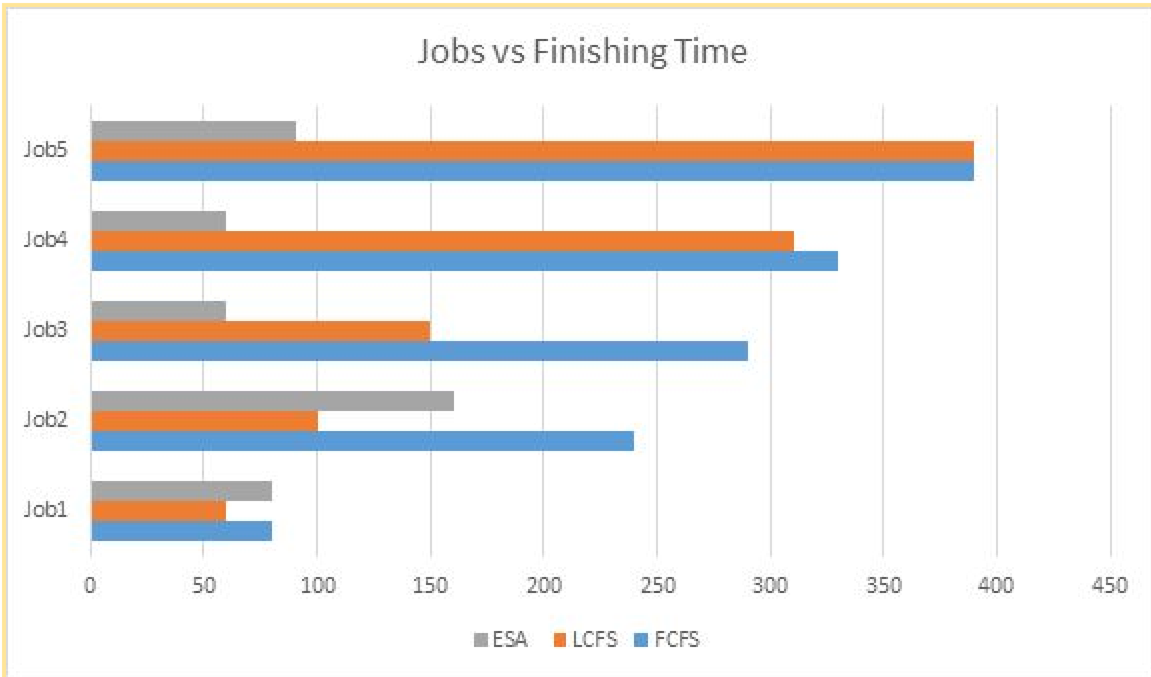
Jobs vs Finishing Time
Job5
Job4
Job3
Job2
Job1
0
50
100
150
200
250
300
350
400
450
ESA
LCFS
FCFS

Figure 5.3.4

Jobs	Incoming Time	Burst Time	FCFS		LCFS		ESA	
			Start	Finish	Start	Finish	Start	Finish
J1	0	80	0.1	80.1	0.1	60.1	0.1	80.1
J2	0	160	80.2	240.2	60.2	100.1	0.1	160.2
J3	10	50	240.2	290.2	100.3	150.1	10.1	60.1
J4	20	40	290.3	330.2	140.4	310.1	20.1	60.1
J5	30	60	330.4	390.2	190.5	390.1	30.2	90.2

Figure 5.3.5

Chapter 6

Discussion and Conclusion

6.1 Summary of Research

In this proposed model, the idea of a system has been presented that will make use of an enhanced algorithm which implements the concepts of resource provisioning and job scheduling simultaneously in order to ascertain the priority of a requested job waiting in a queue. The arrangement set up to test the model, in which random jobs were given to the system inside a simulation made by the cloud toolkit CloudSim, delivered satisfactory results.

6.2 Discussion & Future Works

Individual usage of internet is facing an almost explosive increase lately. People are immensely interested to stay connected to others worldwide. Although, however apparent might the massive increase in usage of the internet for personal purposes be, the industrial side of it is all the more surprising. Business these days can simply not be done without using the internet. There are a number of websites or online services which now are crucial parts of the global economy, and without the very best of service from whom the whole world may encounter extreme economic and social disasters.

The system proposed in this paper is being planned to be put into use in order to provide a much better service to the clients of renowned cloud service providers. Those such as Google or Amazon are the best options to perform the real life implementation of the proposed system that will be both beneficial for the providers and a pleasant experience for the clients. Furthermore, along with the knowledge of the process being proposed, infrastructure will also be provided as a service.

We have utilized CloudSim toolbox [11], to simulate our proposed display. CloudSim contains classes which should be acquired per the prerequisites. To reenact our proposed, demonstrate, we acquired all the vital classes from bundles and construct our own crude code.

Execution of the proposed model is assessed on the premise of two measures:

- Resource utilization
- Processing time

We compared the simulation results for our proposed model with the results for other job scheduling strategies, i.e. FCFS, LCFS and other such methods. Using figure 5.3.3 and figure 5.3.4, with the obtained data showed in figure 5.3.5, a comparison has been performed between the proposed model and other established methods.

Reference

1. Google App Engine, "https://console.cloud.google.com/projectselector/app-engine?src=ac," accessed on 5th August, 2016

2. Microsoft Azure Services Platform, "https://azure.microsoft.com/en-us/overview/what-is-azure/," accessed on 5th August, 2016

3. Amazon Elastic Compute Cloud, "https://aws.amazon.com/ec2/," accessed on 5th August, 2016

4. A. K. Das, T. Adhikary, M. A. Razzaque and C. S. Hong, "An intelligent approach for virtual machine and QoS provisioning in cloud computing," The International Conference on Information Networking 2013 (ICOIN), Bangkok, 2013, pp. 462-467.

5. T. Zain, M. Aslam, M. R. Imran and A. M. Martinez-Enriquez, "Cloud service recommender system using clustering," Electrical Engineering, Computing Science and Automatic Control (CSRS), 2014 11th International Conference on, Campeche, 2014, pp. 1-6.

6. X. Xu and N. Ye, "Minimization of Job Waiting Time Variance on Identical Parallel Machines," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 5, pp. 917-927, Sept. 2007.

7. R. Buyya, S. K. Garg and R. N. Calheiros, "SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions,"

2011 International Conference on Cloud and Service Computing, Hong Kong, 2011, pp. 1-10.

8. C. Malik, S. Jain and S. Randhawa, "Resource scheduling in cloud using harmony search," *2016 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, 2016, pp. 1-6.

9. D. Komarasamy and V. Muthuswamy, "Adaptive Deadline Based Dependent Job Scheduling algorithm in cloud computing," *2015 Seventh International Conference on Advanced Computing (ICoAC)*, Chennai, 2015, pp. 1-5.

10. J. Son, A. V. Dastjerdi, R. N. Calheiros, X. Ji, Y. Yoon and R. Buyya, "CloudSimSDN: Modeling and Simulation of Software-Defined Cloud Data Centers," *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Shenzhen, 2015, pp. 475-484.

11. R. Buyya, R. Ranjan and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," *2009 International Conference on High Performance Computing & Simulation*, Leipzig, 2009, pp. 1-11.

12. I. S. Moreno and J. Xu, "Customer-aware resource overallocation to improve energy efficiency in realtime Cloud Computing data centers," *2011 IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, Irvine, CA, 2011, pp. 1-8.

13. F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, "NIST Cloud Computing Reference Architecture Recommendations of the National Institute of Standards and Technology," Cloud Comput. Program, Inf. Technol. Lab., 2011.

14. B. Kantarci and H. T. Mouftah, "Optimal Reconfiguration of the Cloud Network for Maximum Energy Savings," 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), IEEE Computer Society, Washington, DC, USA, pp. 835-840.

15. M. Usha, J. Akilandeswari and A. S. S. Fiaz, "An Efficient QoS Framework for Cloud Brokerage Services," Cloud and Services Computing (ISCOS), 2012 International Symposium on, Mangalore, 2012, pp. 76-79.

16. http://www.cloudbus.org/applications.html

17. G. Kaur and S. Midha, "A Preemptive Priority Based Job Scheduling Algorithm in Green Cloud Computing," *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, Noida, 2016, pp. 152-156.

18. B. S. Rajeshwari and M. Dakshayini, "Optimized service level agreement based workload balancing strategy for cloud environment," *2015 IEEE International Advance Computing Conference (IACC)*, Banglore, 2015, pp. 160-165.

19. S. Selvarani, A. Julian and E. I. Nehru, "Enhanced two stage heuristics algorithm for VM scheduling," *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, Ramanathapuram, 2016, pp. 726-729

20. M. Alhamad, T. Dillon and E. Chang, "Service Level Agreement for Distributed Services: A Review," *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, Sydney, NSW, 2011, pp. 1051-1054.

21. R. d. R. Righi, V. F. Rodrigues, C. A. da Costa, G. Galante, L. C. E. de Bona and T. Ferreto, "AutoElastic: Automatic Resource Elasticity for High Performance Applications in the Cloud," in *IEEE Transactions on Cloud Computing*, vol. 4, no. 1, pp. 6-19, Jan.-March 1 2016.

22. K. Dubey, M. Kumar and M. A. Chandra, "A priority based job scheduling algorithm using IBA and EASY algorithm for cloud metaschedular," *2015 International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, 2015, pp. 66-70.

List of Acronyms

- CSP- Cloud Service Provider
- CSC- Cloud Service Consumer
- QoS- Quality of Service
- SLA- Service Level Agreement
- PaaS- Platform as a Service
- SaaS- Software as a Service
- IaaS- Infrastructure as a Service
- CA- Cloud Agent
- CDC- Cloud Data Center
- FCFS - First Come First Served
- LCFS - Last COme First Served

List of Notations

Variable Name	Identifier
Resource for per job	R_j
Length of per job	Lj
MIPS needed for each job	Mj
Pe number for per job (Processing Element)	Pj
Bandwidth of per Job	B_j
Job	J
Virtual Machine Id	V_m
Particular weight value for each job	W_j
Virtual Machine Allocated	V

Table: List of notations