International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

An Approach of Analysis of Data Centers in Cloud by Dividing the Response Time

Preeti Kamble[1], Amar Buchade[2]

[1]Master of Engineering Student, Department of Computer Engineering, Pune Institute of Computer Technology, Pune 411043, Maharashtra, India

[2]Assistant Professor, Department of Computer Engineering, Pune Institute of Computer Technology, Pune 411043, Maharashtra, India

Abstract: ***Cloud computing is one of the most emerging fields in computer technology. Cloud computing is the delivery of computing services and resources over the Internet. The performance and availability of cloud applications are very important factors for user adoption the revenue of the cloud for the cloud service provider. The latest work on performance analysis of the cloud computing gives approximated analytical model which provides relationship between the input buffer size and number of servers available. It also gives the performance parameters like task blocking probability, immediate service probability, departure probabilityetc. In current system, the analysis is done only at arrival of requests and departure of responses. We can analyze the processing of the request in various stages of request processing like setup, execution, return & clean up. The analysis of traffic in cloud data centers while processing the requests this much deep level is not done yet. In this paper, we are proposing the idea, need and benefit and results for this much deep level analysis.***

Keywords: Cloud computing, hypervisor, performance analysis, queuing theory, response time, virtual machines

1. Introduction

Cloud computing delivers infrastructure, platform, and software as services, which are made available as subscription-based services in a pay-as-you-go model to consumers. These services in industry are respectively referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The importance of these services is highlighted in a recent report from Berkeley as: "Cloud computing, the long-held dream of computing as a utility has the potential to transform a large part of the IT industry, making software even more attractive as a service" [5].

According to National Institute of Standards and Technology (NIST), the cloud computing is defined as "A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort. In Section II, the related work is discussed. In Section III, the problem is modeled mathematically. In Section IV, we have proposed system architecture with block diagram. In Section V, we have discussed the performance parameters, experimental setup and graphical results regarding this system. In Section VI, we have conclusion and future work. Further, we are mentioning the references.

2. Related Work

Now days, many researchers and industries are now showing lots of interest in the field of cloud computing. The cloud computing fields like security, infrastructure of clouds, applications of clouds etc. have attracted many researchers to contribute towards cloud. However, the field of performance analysis of the cloud computing centers is quite unacknowledged.

The performance analysis of the cloud computing centers is mainly regarding the statistics and the mathematical concepts of the output of the project or the product. The arrangement and behavior of the arrival of the requests and the departure of the responses plays important role in performance analysis of cloud computing centers. The queuing system is involved in this part of the cloud.

In [8], assuming inter arrival and service times as exponential, the cloud is modeled as the classical open network, and its response time distribution was obtained. Using this distribution, the relationship between maximum number of tasks, highest level of service and minimum number of service resources was found. This is an approximate solution for steady state service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models [2].

Cloud services allow individuals and businesses to use software and hardware that are managed by third parties atremote locations. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications [2].

The probability theory parameters like coefficient of variance, standard deviation are illustrated in current researches. The issues like diversity of user requests, large number of server nodes were addressed separately. There was no work that addresses all of them simultaneously. To fill this gap, there is a modeling of cloud center as an M/G/m/m+rqueuing system [1]. It is with a single task

arrivals and a task buffer of finite capacity. If we want to extend the above defined model then we have to break down the response time in the setup, execution, return and clean up time. In this paper, we are going to give the idea about breaking down the response time in the mentioned parts.

The remainder of the paper is organized as follows. Section II gives an overview of existing work on performance evaluation of cloud and characterization of M/G/m/m+r queuing system. Section III discusses the design distribution of queue length in M/G/m queuing system assuming finite waiting space was described in [12].

An approximate solution for steady state distribution of queue length in M/G/m queuing system assuming finite waiting space was described in [12]. As the approximations were given in explicit form, its numerical computation is easier than that of given in [13], [14].

A similar approach with respect to M/G/m queues was explained in [15]. In this approach, the extension of the analysis is done to approximate the task blocking probability and, determines the smallest buffer size i.e. the rate of lost tasks remains below a predefined level. The results showthat the optimal buffer size is directly proportional to the order of convexity of service time.

In [16], another approximation technique was explained for blocking probability. The approach is based on the exact solution for the finite capacity M/G/m/m+r queuing system. Estimation of the task blocking probability helps to guide the buffer allocation. In [17], it was explained that average delay in queuing in M/G/m/m+r queuing system was based on the relationship of joint distribution of remaining service time to the equilibrium service distribution.

In [1], HamzehKhazaei, JelenaMisic and Vojislav B. Misic, proposed the analytical technique based on approximate Markov chain model for performance analysis of cloud computing center. The assumption of general service time for request and large number of servers made this model flexible and scalable. This model can be extended by breaking down response time in setup, execution, return and response time. And this approach we are elaborating in our paper.

3. Mathematical Model

The problem can be modeled mathematically given below.

S = { s, e, X, Y, Fs, DD, NDD,ϕs, Sc, Fl }

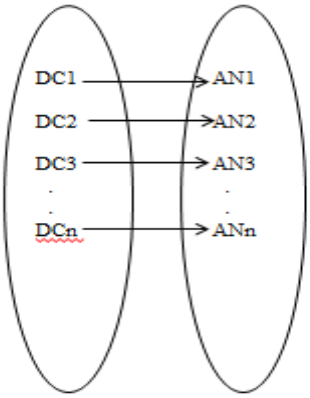
S = A Proposed System
s = Starting point
= c->DC
Where c = Set of client requests
& DC = Data Center
e = Ending Point
= Output displayed to cloud provider with performance parameters
X = Input of the system
= Requests from the clients or tasks for the servers
= (c1, c2, c3,…,cn)
Y = Output of the system
= Performance parameters to cloud provider
Fs = Friend functions for system
= {initialize_cloud(), start_datacenter(), send_cloudlet_to_vm(), recv_cloudlet_after_exec(),shutdown_datacenter() }
DD = Deterministic Data
= cR ϵ (AR Λ FR)
Where, cR = Required resources by client request
AR = Available Resources
FR = Free Resources
NDD = Non-Deterministic Data
= cR ϵ (AR V FR)
= Required resources are not available or free
Φs = Rules of S (Functions in System)
= { start_hypervisor(), create_vm_in_datacenter(), details_about_setup() , details_about_execution(), details_about_return(), details_about_cleanup(), destroy_vm(), calculate_performance_parameter s(), send_output_to_cloud_provider(), print_output_and_decison() }
Sc = Successful execution of user request
= c1->r1
where, c1 is client request
& r1 is response for c1
Fl = Failure in execution of user request
= Conditions:

- Resource unavailability
- Insufficient input
- Starvation of request

3.1 Function Mapping

1. Data Center and Analyzer:

Each data center can have one analyzer for itself.

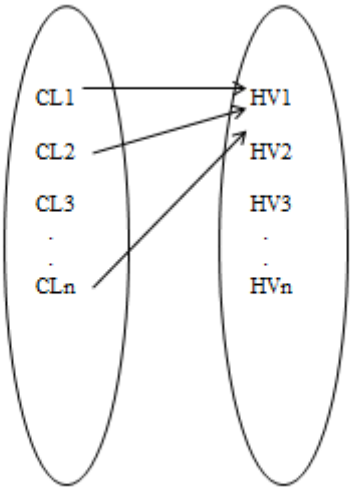


Data Center Analyzer

Figure 1: One-one onto mapping for Data center and analyzer

2. Client and Hypervisor

Many clients can send requests to one hypervisor.

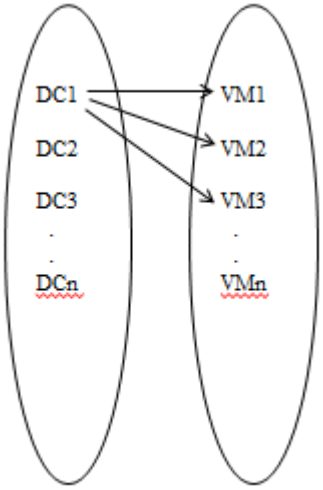


Client Hypervisor

Figure 2:.Many-one onto mapping for client and hypervisor

3. Data Center and Virtual machine

One data center can have many virtual machines.



Data Center Virtual machine

Figure 3: One-many mapping for data center and virtual machine

4. A Proposed System

The performance analysis of a cloud computing centers involves mainly the arrangement of the arrival of requests and departure of responses. The queuing theory is one of the most important factors of modeling a cloud and analyzing the performance of cloud computing centers. Till now, the performance analysis of the cloud computing centers is done only considering arrival time of requests and departure time of request.

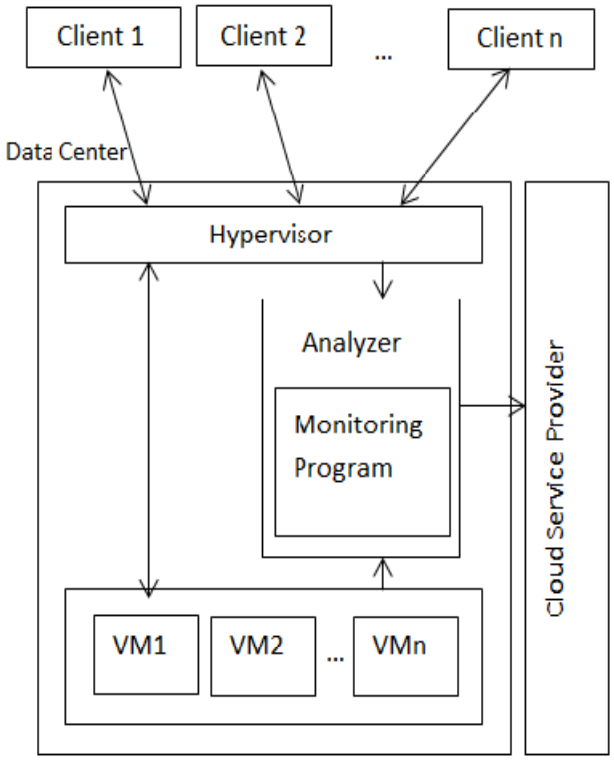


Figure 4: Architecture of Traffic Analyzer in Cloud Data Centers

The architecture of our system is shown in Fig. 4. This architecture explains how the flow of request processing in the cloud occurs and how we are going to analyze these stages with cloud performance parameters. First the client sends request to the data center i.e. server. This request is accepted by the hypervisor of data centers. Then accepted request is sent to the virtual machines to get served. Hypervisor and virtual machine will give every detail of executions to the analyzer. According to those observations, the monitoring program will do the traffic analysis in data center i.e. analysis of request processing in each stage. We will give this analysis to the cloud provider and decision making is also provided for the cloud provider to work and monitor cloud more wisely.

The request processing has four stages as following.

1. Setup: In this stage, the request is accepted from client and resource allocation is done for the request as per the request need. After resource allocation, it is sent to execution stage.

2. Execution: As the resources are allocated, the request proceeds for the execution where the response is created by execution of request. The response will be ready to send to the client at the end of this stage.

3. Return: In this stage, the response is taken from the execution stage. The returning of the response to the client is done is this stage. The response is provided to the client in this stage.

4. Cleanup: This is the last stage of request processing. In this stage, the resources which are allocated for the request

are deallocated as the response is given to the client. These resources are made available to the next requests.

We are proposing an idea of analyzing the traffic which is inside the data centers while processing the request. We are defining the approach to analyze traffic by dividing the response time in setup time, execution time, return time and cleanup time. The setup time is the time required for the allocation of resources to serve the request. The execution time is the time to serve the computations for request. The return time is the time to return the response to the client after executing request. The cleanup time is the time to deallocate the resources after sending the response to the respective client.

This analysis will have many benefits regarding the management and analysis of cloud computing centers. The analysis of cloud at different stages of request processing will give the clearer view of analyzing to the deep level. We will find the performance parameters like task blocking probability, immediate service probability, departure probability, regarding these stages of request processing.

These parameters regarding setup stage will give the suggestions about changing the resource allocation strategy and increasing or decreasing the buffer size in the system. The parameters regarding the execution stage will give the decisions about the increasing or decreasing the execution unites or memory. The parameters about the return time will guide regarding the size of buffer which is used to keep the responses before after executing the request and before sending that response to the respective client. This will give the situations for connections to send the responses to the clients. The analysis of cleanup stage will give the strategy to deallocate the resources as soon as the response is sent to the client.

This deep analysis will help the cloud provider to manage and monitor the cloud more efficiently. For example, if there is a scenario like most of the requests are starved before entering into the system. Then the cloud provider can look at the parameters which we have analyzed and can observe that task blocking probability is higher in setup stage. Hence, the curative actions like changing resource allocation policy, increasing resources, increasing buffer size etc. can be taken by the cloud provider. So, instead of changing the whole cloud, changing the part of the cloud which is performing low is more efficient, more cost saving and better. Thus, this analysis will help cloud provider to make decisions wisely and correct with the help of real-time statistical proofs.

Performance Parameters:

Performance is the accomplishment of a given task measured against preset known standards of accuracy, completeness, cost, and speed [2]. Performance monitoring is integral part of maintenance. Requirements for the monitoring solution for cloud are totally different from a legacy and virtualized environment monitoring solution.

There are many third party tools/solutions available for cloud monitoring. But, there are no standard model exist for such a solution as what all parameters needs to be covered in the solution so that an exhaustive performance report can be produced for the service provider [6]. The Cloud service provider needs to get an exhaustive view of the health of the entire cloud to assess the situation. Lot of decision making and determining SLAs are driven by the cloud performance.

There are three performance parameters as follows.

1. Task Blocking Probability(TBP)

The task blocking probability is defined as the probability of blocking of the task i.e. request from processing in each stage of request processing. It is denoted as P_B. As the request is blocked, it means the request is not served immediately. Hence, the task blocking probability is calculated as follows,

$$P_B = (T_{SA} - T_{SE}) / T_{SA}$$

where the terms T_A and T_E are same as in the immediate service probability.

2. Immediate Service Probability(ISP)

This is defined as the probability of getting the request served immediately in each stage of request processing. It is complementary for task blocking probability.

It is denoted as P_S.

$$P_S = 1 - [(T_{SA} - T_{SE}) / T_{SA}]$$

3. Departure Probability(DP)

The departure probability is the probability that the request is departed from each stage successfully. It is the probability that the request has completed all the stages of request processing. In other words, it is the probability that the request id not killed or suspended while the processing.

We consider the T_{DA} as the actual time when the request is departing for next stage of processing. Let T_{DE} be when the request is expected to be departed to the next stage of processing. Hence, P_D is calculated as follows,

$$P_D = (T_{DA} - T_{DE}) / T_{DA}$$

The departure probability will give us the prediction about request is processed or killed.

5. Experimental Setup

In the experimental setup for this system, we have used deployed a cloud in java and analyzed that.We have also built a cloud infrastructure using Openstack on Ubuntu 12.04 using DevStack module also. We have configured the Stacktach tool for the Openstack to invoke the parameters regarding analysis.

The particulars about platform and technology used:

- Base Operating System: Ubuntu 12.04, Windows 7
- Databases: MySQL

- Web Server: Apache
- Openstack Network: Nova
- Language: Python, Java
- Browser: IE8 & above, Mozilla Firefox, Google Chrome, Opera etc.
- Hypervisor: KVM hypervisor

According to these specifications and structure, the system is built up. The following is the list of some of performance parameters which we have analyzed in our system.

- Serial number of request
- Used VM ID
- Datacenter ID
- Setup time
- Execution time
- Return time
- Cleanup time
- Start time for setup
- End time for setup
- Start time for execution
- End time for execution
- Start time for return
- End time for return
- Start time for cleanup
- End time for cleanup

After completing the request and observing transaction metrics like above, we have found the following performance parameters for all the setup, execution, return and cleanup stage.

- Task Blocking Probability (TBP)
- Immediate Service Probability (ISP)
- Departure Probability (DP)

6. Results

We have analyzed the components which the previous system was unable to analyze. The following table shows the comparison of the analyzed parameters in our system and previous system.

Table 1 show that we have developed the system which can give deeper analysis than the previous system. Hence, our goal is accomplished as decided

Table 1: Comparison of previous system and our system

Parameters	Can it be calculated in the following systems?	
	Previous System	*Our System*
Request arriving time	✓	✓
Resource allocation start time	×	✓
Resource allocation end time	×	✓
Total resource allocation time	×	✓
Execution start time	×	✓
Execution end time	×	✓
Total execution time required	×	✓
Return start time	×	✓
Return end time	✓	✓
Total return time required	×	✓
Cleanup start time	×	✓
Cleanup end time	×	✓
TBP for overall system	✓	✓
ISP for overall system	✓	✓
Departure probability for overall system	✓	✓
TBP for each stage	×	✓
ISP for each stage	×	✓
Departure probability for each stage	×	✓

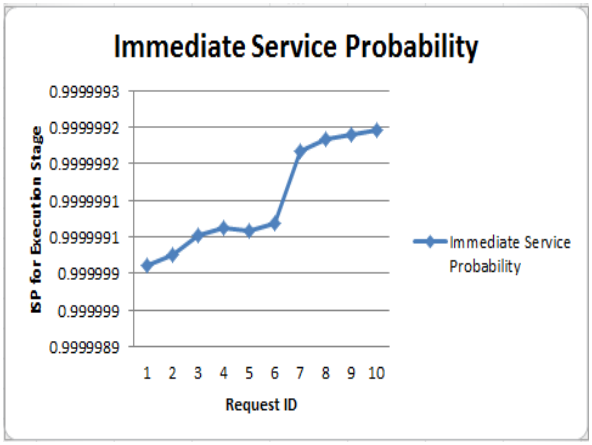


Figure 5: Immediate Service Probability for execution stage of request processing

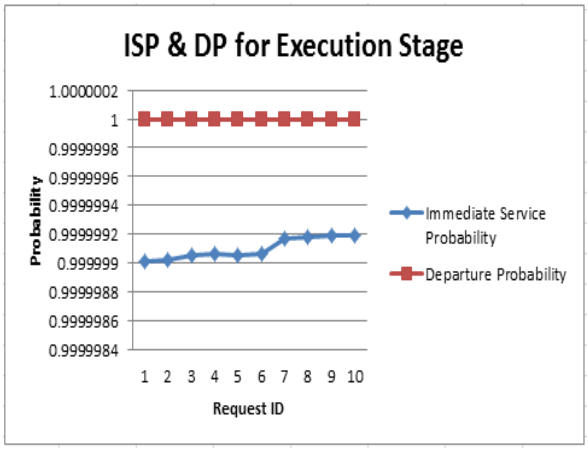


Figure 6: Comparison of ISP and DP for execution stage of request processing

Figure 5 gives the immediate service probability graph of the execution stage of request processing. We can see it that it is very near to 1. It is increasing because the initial starting up of the cloud takes very little fraction of time to start the cloud as request come and to create virtual machines. We can see that the immediate service probability is increasing as the

requests come. It is improvement in the system that the requests are less waiting for processing. The task blocking probability for each stage is calculated as per the above formula as explained in the proposed system. We have calculated the departure probability for 10 requests for the numerical validation regarding the execution stage of request processing. We have compared the immediate service probability and departure probability and found the following analysis.

Table 2: The benefits and analysis of the parameters ISP and DP to improve the cloud

Parameters	*Result*	*Analysis*
Departure probability	1	No request is killed or suspended. All processing is running smoothly
Immediate Service Probability	Near to 1, but not exactly 1	The requests are servers as soon as possible, but still some of the requests need to wait. Increasing the virtual machines can serve the requests by the cloud provider more immediately and the performance can be improved.

According to this analysis we can easily analyze that where our system is lacking and where we need to improve our system. We have obtained that the performance parameters like immediate service probability, task blocking probability and departure probability for each stage of request processing like setup, execution, return and cleanup. We can analyze the exact area for improvement instead of changing the whole system.

Fig. 7 shows the CPU utilization of the cloud in Openstack. This is regarding all the requests in the cloud. From this we can analyze our system that how our CPU is utilized for the cloud.

Figure 7: CPU utilization of the cloud in Open stack

7. Conclusion

The cloud computing field has opened many doors for the inventions. To increase the scope of the cloud, the better and deep level analysis is the need of time. The state-of-the-art methods and strategies are required to give a detailed analysis regarding the results and working of the cloud. The cloud provider plays important role in making the cloud more adaptable and popular. Hence, better analysis will give better results. The extensive analysis of the performance of the cloud computing centers can be given by monitoring the stages of the request processing. The request processing stages are setup, execution, return and cleanup. Analyzer which we have kept in the data centers keeps track of each transaction in the hypervisor and virtual machine.

In future, we are planning to extend the infrastructure of our system. We are planning to have many data centers and many virtual machines in that. This system can be extended by giving burst arrival of requests and analyze them precisely.

References

[1] HamzehKhazaei, JelenaMisic and Vojislav B. Misic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queuing Systems", In IEEE Transaction on Parallel and Distributed System, Vol. 23,No. 5, May 2012.
[2] Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-145, Sept. 2011.
[3] PreetiKamble, HemlataChanne, "A Survey Paper on Performance Analysis of Cloud Computing Centers", In Proc. Of "International Conference on Computer Science and Computational Mathematics 2013", ISBN:978-93-81693-88-10 , Feb 2013.
[4] PreetiKamble, HemlataChanne, "Performance Analysis Of Cloud Computing Centers By Breaking-Down Response Time", In International Journal of Advanced Computational Engineering and Networking, ISSN(p): 2320-2106, ISSN(e) 2321-2063 Volume- 1, Issue- 8, Nov 2013.
[5] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. *Above the Clouds: A Berkeley View of Cloud computing*. Technical Report No. UCB/EECS-2009-28, University of California at Berkley, USA, Feb. 10, 2009.
[6] Vineetha V., "Performance Monitoring in Cloud", Infosys View-Point 2012.
[7] itl.nist.gov/div898/handbook/eda/ section3/eda35b.html
[8] K. Xiong and H. Perros, "Service Performance and Analysis in Cloud Computing", Proc. IEEE World Conf. Services, pp. 693-700, 2009.
[9] G. Grimmett and D. Stirzaker, Probability and Random Processes, third ed. Oxford Univ. Press, July 2010.
[10] M. Miyazawa, "Approximation of the Queue-Length Distribution of an M/G/s Queue by the Basic Equations," J. Applied Probability, vol. 23, pp. 443-458, 1986.

[11] D.D. Yao, "Refining the Diffusion Approximation for the M/G/m Queue," Operations Research, vol. 33, pp. 1266-1277, 1985.
[12] T. Kimura, "A Transform-Free Approximation for the Finite Capacity M/G/s Queue," Operations Research, vol. 44, no. 6, pp. 984-988, 1996.
[13] P. Hokstad, "Approximations for the M/G/m Queues," Operations Research, vol. 26, pp. 510-523, 1978.
[14] H.C. Tijms, "Heuristics for Finite-Buffer Queues," Probability in the Eng. And Informational Sciences, vol. 6, pp. 277-285, 1992.
[15] T. Kimura, "Optimal Buffer Design of an M/G/s Queue with Finite Capacity," Comm. in Statistics Stochastic Models, vol. 12, no. 6, pp. 165-180, 1996.
[16] J.M. Smith, "M/G/c/K Blocking Probability Models and System Performance," Performance Evaluation, vol. 52, pp. 237- 267, May 2003.
[17] S.A. Nozaki and S.M. Ross, "Approximations in Finite-Capacity Multi- Server Queues with Poisson Arrivals," J. Applied Probability, vol. 15, pp. 826-834, 1978.
[18] O. J. Boxma, J. W. Cohen, and N. Huffel, "Approximations of the Mean Waiting Time in an M/G/s Queueing System", Operations Research, vol. 27, pp. 1115-1127, 1979.
[19] T. Kimura, "Diffusion Approximation for an M/G/m Queue," Operations Research, vol. 31, pp. 304-321, 1983.
[20] H.C. Tijms, M.H.V. Hoorn, and A. Federgru, "Approximations for the Steady- State Probabilities in the M/G/c Queue," Advances in Applied Probability, vol. 13, pp. 186-206, 1981.
[21] J. Virtamo, "Queuing Systems", http://www.netlab.tkk.fi/opetus/s383143 /kalvot/E_jonojarj.pdf
[22] J.Virtamo, "Poisson Process", http://www.netlab.tkk.fi/opetus/ s383143/kalvot/E_poisson.pdf
[23] B.N.W. Ma and J.W. Mark, "Approximation of the Mean Queue Length of an M/G/c Queuing system", Operations Research, vol. 43, pp. 158-165998.

Author Profile

Preeti Kamble received the B.E. in Computer Engineering in 2012 from BRACT's Vishwakarma Institute of Technology, (Affiliated to University of Pune), Pune, India. She is pursuing Master of Engineering (M.E. Computer Engineering) from SCTR's Pune Institute of Computer Technology, (Affiliated to University of Pune), Pune, India. Her research interests include cloud computing, modeling, and performance evaluation of computing systems and queuing theory. She has published 3 papers in international journals and conferences.

Amar Buchade is assistant professor in Department of Computer Engineering at SCTR's Pune Institute of Computer Technology, (Affiliated to University of Pune), Pune, India. He has received his degree B.E. and M.E. from Walchand College of Engineering, Sangali in 2002 and 2005 respectively. His research area is Distributed System, Cloud computing and Security. He has published more than 8 papers in national and international conferences.