International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 2, April 2016, pp. 743~750
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i2.9060

A Simulation-based Approach to Optimize the Execution Time and Minimization of Average Waiting Time Using Queuing Model in Cloud Computing Environment

Souvik Pal, Prasant Kumar Pattnaik
School of Computer Engineering, KIIT University, Bhubaneswar, India

Article Info

Article history:

Received Sep 22, 2015
Revised Nov 28, 2015
Accepted Dec 20, 2015

Keyword:

Cloud broker
Cloud computing
Queuing model
Virtualization
Waiting time

ABSTRACT

Cloud computing is the emerging domain in academia and IT Industry. It is a business framework for delivering the services and computing power on-demand basis. Cloud users have to pay the service providers based on their usage. For enterprises, cloud computing is the worthy of consideration and they try to build business systems with lower costs, higher profits and quality-of-service. Considering cost optmization, service provider may initially try to use less number of CPU cores and data centers. For that reason, this paper deals with CloudSim simulation tool which has been utilized for evaluating the number of CPU cores and execution time. Minimization of waiting time is also a considerable issue. When a large number of jobs are requested, they have to wait for getting allocated to the servers which in turn may increase the queue length and also waiting time. This paper also deals with queuing model with multi-server and finite capacity to reduce the waiting time and queue length.

Corresponding Author:

Souvik Pal,
School of Computer Engineering,
Campus 15, KIIT University,
Patia, Bhubaneswar, Odisha-751024, India
Email: souvikpal22@gmail.com

1. INTRODUCTION

Cloud computing is the use of the Internet for the tasks the users performing on their computer. Cloud computing, also known as Internet computing, provides on-demand services which are concerned with shared resources, software, information, and other application specific services. Cloud Service Providers (CSPs) provide the service according to the clients' requirement with minimal effort at stipulated time [1]. Cloud computing has also the capability to deliver the services to the users by growing or shrinking in instantaneous requirements [2] [3]. Cloud infrastructure implies a business framework between cloud users and CSPs creating a relationship through cloud brokerage services [4] [5].

Cloud brokers [6], acting as negotiator or mediator, help out the users to maintain all the activities such as specific data centre required, execution time of the respective requests, number of CPU cores, waiting time of the respective user request. Cloud broker also helps the users for self-provisioning of the resources. In this manuscript, CloudSim Simulator has been used to find the minimum execution time with respect to the number of CPU cores. Cloud broker is there to decide the average waiting time in the system and also in queue using queuing model.

The paper is organized as follows:

Section 2 deals with the literature survey of the related work and objective of the study. In the section 3, we have discussed about the simulation workflow, sequence diagram of the flow content and simulation result using CloudSim Simulator V3.0. In the section 4, we have presented queuing model for

cloud computing environment. Section 5 deals with the numerical result analysis and graphical representation. At the end of the paper, conclusion section brings to a close of the work.

2. LITERATURE SURVEY OF THE RELATED WORK

In cloud computing environment, minimization of waiting time in the queue and in the system is the core research area in academia and also in industry [7]. It is also concerned with the number of servers or number of CPU cores to deliver the services as per the user request. Minimization of waiting time leads to customer satisfaction and usage of less number of CPU cores enhances the performance of the cost optimization. Execution time of a particular job request is also a considerable issue in cloud computing environment. In this manuscript, we have made an extensive survey that is related to work.

Buyya, Ranjan, and Calheiros, 2009 [8], have presented CloudSim toolkit which enables to model and simulate in cloud computing environments. In CloudSim simulator, they have given the provision to create the number of Virtual Machines (VMs) in a particular Data Center and to use different VM allocation and VM selection policies to model. Moreover, this paper allows the users to find the execution time of a job request, to do VM migration, and also regular scaling of applications in cloud computing environment.

Pu, Liu, et al., 2010 [9], have presented their performance analysis in parallel progression of CPU. They have also focused on monitoring workload on Xen Virtual Machine Monitors, which is concerned with network intensive workload. Their paper deals with the experiments for finding out the performance measurements on network I/O workload.

Khazaei, Misic, et al., 2011 [10], have discussed about the techniques of resource provisioning, the procedures of delivering different types of services such as infrastructure-based, platform-based, and software-based. They have focused to calculate the performance measurement while provisioning the resources to realize the Service Level Agreements (SLAs). In this paper, they have also described an analytical representation for evaluation of different issues such as response time, server farms, and number of tasks for sufficient accuracy.

Rodrigo, Ranjan, et al., 2011 [11], have described about the simulation strategy of CloudSim in their paper. CloudSim toolkit supports the modeling and simulation in cloud environment, which is also concerned with resource provisioning, creation of VMs, modeling of data centers, cloud broker policies, different SLAs etc. CloudSim toolkit also works in both single cloud and federation of clouds. This paper also represents the improvement of the application Quality of Service (QoS) requirements.

Spillner, Brito, et al., 2012 [12], have presented an economically compensation concept to raise the granularity and efficacy of reserved computation. This paper enables highly virtualized resource broker in the business-oriented market place, which facilitates the consumer with configurable VMs for resource sharing. This paper supports on-demand resource provisioning with the help of scalability.

Khazaei, Misic, et al., 2012 [13], have discussed about the modeling of cloud centers. They have proposed a performance measurement model to evaluate the cloud farms and found out the solution to get the estimation of probability distribution. Their model helps the CPSs to decide the number of servers, input size, and number of tasks in the system.

Pal and Pattnaik, 2013 [14], have presented virtualization classification in cloud computing environment. Virtualization technology manages and coordinates the accesses from the resource pool. Virtualization helps the CSPs to overcome composite workloads, and different software architecture. They have discussed about the virtualization classification and their working principle in the paper.

Xiao, Song, an Chen, 2013 [15], have described the technique of allocating data center resources through virtualization technology. They have introduced the idea of “skewness” for measurement of the unevenness of resource utilization of a server in multidimensional way. They have also developed a way by which the overall consumption of server resources can be improved.

Karthick, Ramaraj, and Subramanian, 2014 [16], have proposed MQS (Multi Queue Scheduling) algorithm which aims to minimize the cost of both on-demand requirements and reserved plans with the help of global scheduler. Global scheduler intends to share the physical resources to its maximum level. The proposed algorithm uses the technique of clustering the tasks depending upon the burst time. This paper also reduces the chances of fragmentation and also minimizes the starvation problem.

Yang, Kwon, et al., 2014 [17], have introduced the techniques which requires compiler code analysis. This procedure minimizes the transferred data size with the help of changing the heap objects. They have discussed the procedure of cost cutting techniques for dynamic execution in cloud. Their result shows that reduced size affects both the transfer time and execution offloading in an efficient manner.

Pal and Pattnaik, 2015 [18], have presented the minimization of average waiting time using Johnson sequencing algorithm. When a huge number of requests arrive, they have to wait for allocation. This situation

increases waiting time and queue length. They have minimized average waiting time in the system and in the queue by means of queuing model with finite capacity and multi-server capability.

Calero and Aguado, 2015 [19], have presented a monitoring architecture concerned to the CSP and cloud user. This architecture allows the user to customize the metrics. The cloud providers can easily track the services used by the users. CSPs have used Adaptive distributed monitoring technique which is implemented in cloud infrastructure.

2.1. Objective of the Study

In the previous section, we have discussed the CloudSim simulator, Queuing model, Virtualization, average waiting time. When a huge number of requests have come, the user requests have to wait in the queue and in the system. In this paper, we have used CloudSim simulator version 3.0 to find out the minimum execution time with respect to number of CPUs. After getting the minimum number of CPU and execution time, queuing model has been implemented to reduce the average waiting time. In this manuscript, M/M/c and M/M/c/K queuing system has been used to compare the average waiting time in the queue and also in the system.

3. SIMULATION WORKFLOW

In this section, we have briefly discussed our simulation work-flow as shown as figure [1], and we are going to describe our sequence diagram stepwise as follows:

STEP 1: Cloud user sends the job request to the User Interface
STEP 2: User Interface analyzes the request according to the Service Level Agreement (SLA).
STEP 3: User Interface assigns the tasks to cloud broker.
STEP 4: Cloud broker divide the tasks into same sized cloudlets.
STEP 5: Cloud broker sends the cloudlets to Virtual Machine Manager (VMM).
STEP 6: Each data center entity registers with the Cloud Information Service (CIS) registry.
STEP 7: Cloud broker sends the resource request to Cloud Information service (CIS). Then Cloud broker consults the CIS to obtain the list of resources which can offer infrastructure services that matches user's hardware and software requirements.
STEP 8: The cloud broker gets information about the availability of the datacenter and resources from the CIS.
STEP 9: Virtual machine manager (VMM) creates the virtual machine.
STEP 10: Data Center entity invokes Update VM Processing for every host that manages it as processing of task units is handled by respective VMs. So their progress must be continuously updated and monitored.
STEP 11: At the host level, invocation of Update VM Processing triggers an Update Cloudlet Processing method that directs every Virtual Machine (VM) to update its task unit status (finish, suspend, executing) with the Data center entity.
STEP 12: VM analyze the approximate execution time and sends to host machine.
STEP 13: Host machine analyzes smallest time to next event.
STEP 14: Data center provides the information about the execution time and different resources such as Operating System, VMM used, RAM size, MIPS, number of cloudlets, number of CPU, storage capacity etc.
STEP 15: Request for execution of the cloudlet is sent to the virtual machine by VMM.
STEP 16: Cloudlet is being executed in the VM.
STEP 17: VM sends the executed cloudlets to the VMM.
STEP 18: After completing the execution, VM releases the resources for further use.
STEP 19: CIS updates the registry according to the information sent by data center.
STEP 20: VMM further passes the executed cloudlets to cloud broker.
STEP 21: Cloud broker combines all the executed cloudlets together to form the task.
STEP 22: Cloud broker sends the completed task to the User Interface.
STEP 23: After completion of the task, User Interface can either expire the session or make another renewal request.
STEP 24: If session is expired, then User Interface sends the executed task to the cloud user.

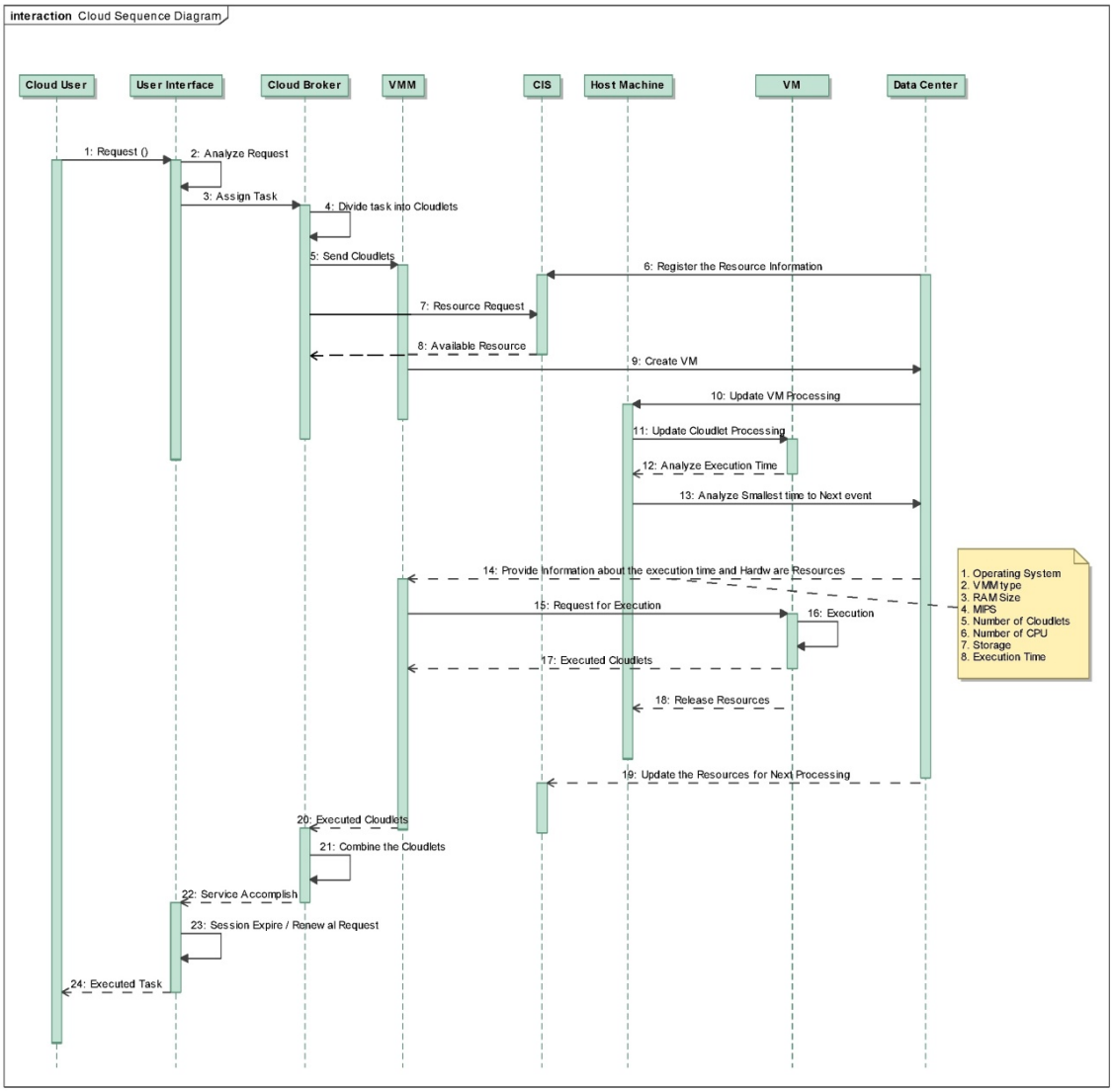

Figure 1. Sequence Diagram of the workflow

3.1. Simulation Result

In this section, we are going to test the execution time with respect to the number of CPUs. The tests were conducted on a 32-bit Intel Core i5 machine having 2.60 GHz and 3 GB RAM running windows 7 Professional and JDK 1.6. The main goal of our tests is to evaluate the execution time when the numbers of CPU cores as VM parameters which varies from 1 to 8. According to that variation, how the execution time is changing and we are going to find in an optimized situation where execution time is less.

We have used Eclipse Java EE IDE for Web Developers, Version: Juno Service Release 2 and CloudSim version 3.0 for simulation purpose. In our experimental set up, this simulation works only when simulation is paused for 5 sec and this simulation creates a datacenter Broker dynamically and also subject to other constraints this Simulation is done.

The simulation environment consists of two hosts; each host has been modeled to have 1000 MIPS, 16 GB of RAM memory, 1 TB of storage and 10 numbers of VMs each of which has been modeled to have 500 MIPS, 1 GB of RAM, and 10 GB of image size. A datacenter is created, which has the characteristics like x86 of architecture, Linux as operating system, Xen as VMM. Simulation uses VM Allocation Simple as VM Allocation Policy, which chooses, as the host for VM, the host with less processing elements in use.

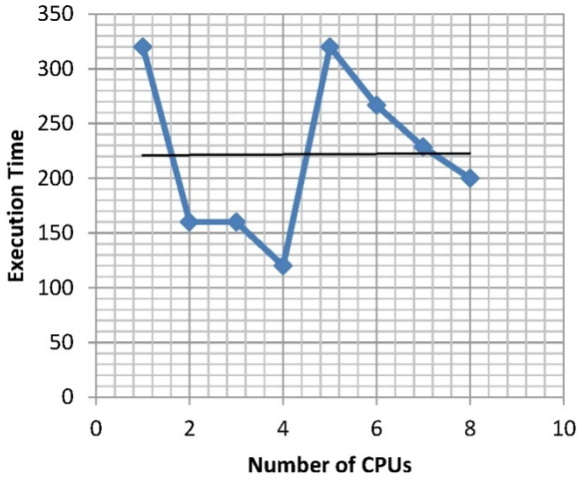

Figure 2. Evaluation of Execution Time

In the figure [2], we want to show the execution time in our experimental situation. While changing the Number of CPUs, the execution time for the user-request varies in a random order. In this case, we can say that if we use 4 numbers of CPUs, the execution time will be minimal. Therefore, cloud user can get the services with minimal time.

4. QUEUING MODEL FOR CLOUD COMPUTING ENVIRONMENT

Queuing system is a mathematical model for analysis of waiting line. Waiting lines or queues have occurred when service demand goes beyond the capacity of the service provider. The queuing model is basically discussed by specifying [20] [21] arrival and service process, number of servers and the maximum capacity of the system. In this paper, we concentrated on M/M/c queuing model and M/M/c/K queuing model. Assuming that the requests come to the server at Poisson distribution rate and the process time is taken as exponential distribution. It is also assumed that all the processes are non-pre emptive. In those queuing model, c denotes the number of servers and K indentifies the places for maximum capacity. So, (K-c) is the queue capacity. We have considered five places of maximum capacity.

According to the Kendal's notation [22], average arrival rate will be $\lambda = \frac{1}{E[\tau]}$, where τ = Inter-arrival time and E[τ] is denoted as the average or mean Inter-arrival time. Service rate will be $\mu = \frac{1}{E(S)}$, where S is denoted as service time of the customer and E(S) identifies as average service time. To make a stable system, an equilibrium condition is to be maintained where the utilization factor $\rho = \frac{\lambda}{\mu} \leq 1$.

5. NUMERICAL ANALYSIS

In this section, we have briefly discussed numerical analysis using queuing model. We have initially taken average arrival rate and average service rate as shown in the table [1]. Using M/M/c queuing model and M/M/c/K queuing model, we have a comparison study of waiting time.

Table 1. Initial Parameter (Average arrival rate and Service rate [23])

λ	μ
20	40
60	70
120	122

According to queuing system, it has been denoted average number of customers in the system, average number of customers in the queue, average waiting time in the system, and average waiting time in the queue as Ls, Lq, Ws, and Wq respectively. The tables [2-3] show the comparison study using different queuing model.

Table 2. Results with M/M/c Queing Model

	Lq	*Ls*	*Wq*	*Ws*
λ = 20	0.0003	0.5003	0.00001	0.0250
λ = 60	0.0033	0.8605	0.00006	0.0144
λ = 120	0.0063	0.9900	0.00005	0.0083

Table 3. Results with M/M/c/K Queing Model

	Lq	*Ls*	*Wq*	*Ws*
λ = 20	0.0002	0.5001	0.00000	0.0250
λ = 60	0.0021	0.8575	0.00001	0.0143
λ = 120	0.0036	0.9837	0.00000	0.0082

According to the numerical results we have discussed the comparison study regarding Lq, Ls, Wq, and Ws. The following Figures [3-6] show that the average number of customers and the average waiting time in the queue and in the system can be minimized using M/M/c/K rather than M/M/c model.

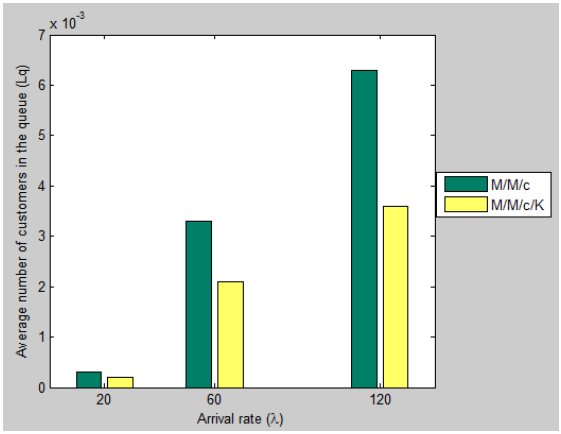

Figure 3. Graph Analyzing Average number of customer in the queue (L_q)

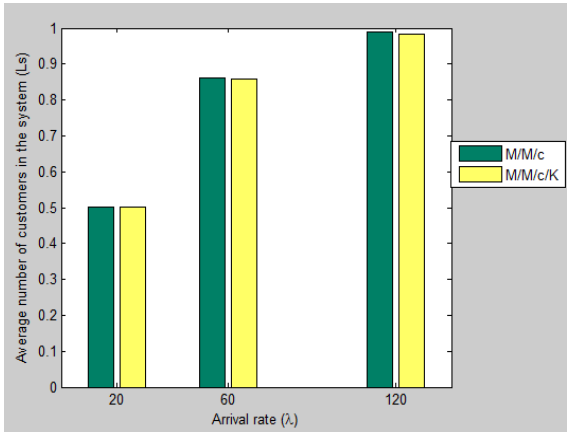

Figure 4. Graph Analyzing Average number of customer in the system (L_s)

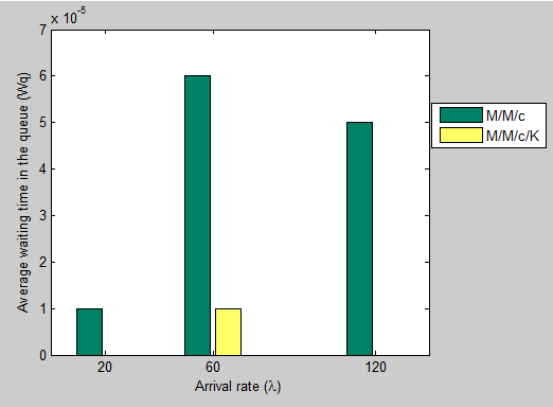

Figure 5. Graph Analyzing Average waiting time in the queue (W_q)

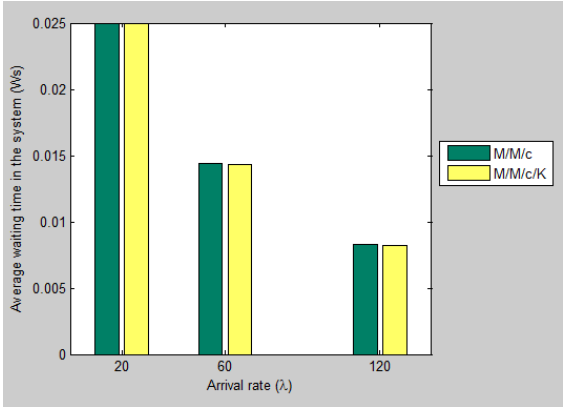

Figure 6. Graph Analyzing Average waiting time in the system (W_s)

6. CONCLUSION

Rapid usage of Internet over the globe, Cloud Computing has placed itself in every field of IT industry. The recent efforts to make cloud computing technologies better, which includes executing time. Therefore, we have concentrated on simulation-based approaches which help the cloud developers to test performance of their service delivery policies and also their execution time and average waiting time so that the cloud service providers can provide better quality services with minimum execution time. At the end of our work, we can conclude that our sequence diagram and our simulation results may help to grow in cloud infrastructure in surge of fast-growing usage of internet among the people.

REFERENCES

[1] F.B. Shaikh and S Haider, "Security threats in cloud computing", "*6th International IEEE conference on Internet Technology and secured transaction"*, 11-14 December 2012, pp. 214-219.

[2] V. Sarathy, P. Narayan and R. Mikkilineni, "Next generation cloud computing architecture-enabling real-time dynamism for shared distributed physical infrastructure", *in the Proceedings of 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'10)*, Larissa, Greece, 2010, pp. 48-53.

[3] S. Pal and P.K. Pattnaik, "Efficient architectural Framework of Cloud Computing", in "*International Journal of Cloud Computing and Services Science (IJ-CLOSER)"*, Vol. 1, No. 2, 2012, pp. 66-73.

[4] V. I. Munteanu, C. Mindruta and T. Fortis, "Service brokering in cloud governance", *in the Proceedings of 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, IEEE, 2012, pp. 497-504.

[5] S. Sundareswaran, A. Squicciarini and D. Lin, "A Brokerage-Based Approach for Cloud Service Selection", *in the Proceedings of 2012 IEEE Fifth International Conference on Cloud Computing*, IEEE Computer Society, 2012, pp. 558-565.

[6] Stella Gatziu Grivas, Tripathi Uttam Kumar, Holger Wache, "Cloud Broker: Bringing Intelligence into the Cloud", *IEEE 3rd International Conference on Cloud Computing*, IEEE, 2010, pp.544~545.

[7] T. Sowjanya, *et al.*, "The Queuing Theory in cloud Computing to Reduce the Waiting Time", *International Journal of Computer Science and Engineering Technology (IJCSET)*, Vol. 1, No. 3, 2011, pp. 110-112.

[8] R. Buyya, R. Ranjan and R.N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities", *in the Proceedings of International Conference on High Performance Computing & Simulation (HPCS '09)*, 21-24 June 2009, pp. 1-11.

[9] P. Xing, *et al.*, "Understanding Performance Interference of I/O Workload in Virtualized Cloud Environments", *in the Proceedings of IEEE 3rd International Conference on Cloud Computing (CLOUD),* 5-10 July *2010*, pp.51-58.

[10] H. Khazaei, J. Misic and V.B. Misic, "Modelling of Cloud Computing Centers Using M/G/m Queues", *in the Proceedings of 31st IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW),* 20-24 June, 2011, pp. 87-92.

[11] R.N. Calheiros, *et al.*, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and evaluation of Resource Provisioning Algorithms", *Software: Practice and Experience (SPE)*, January 2011, Vol. 41, No. 1, pp. 23-50.

[12] J. Spillner, *et al.*, "A Highly-Virtualising Cloud Resource Broker", *in the Proceedings of IEEE Fifth International Conference on Utility and Cloud Computing (UCC),* 5-8 Nov., 2012, pp. 233-234.

[13] H. Khazaei, J. Misic and V.B. Misic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queuing Systems", *IEEE Transactions on Parallel and Distributed Systems*, May 2012, Vol. 23, No. 5, pp. 936-943.

[14] S. Pal and P.K. Pattnaik, "Classification of Virtualization Environment for Cloud Computing", in *Indian Journal of Science and Technology (IJST)"*, Vol. 6, Issue 1, January 2013, pp. 3965~3971.
[15] Z. Xiao, W. Song and Q. Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment", *IEEE Transaction on Parallel and Distributed Systems*, Vol. 24, No. 6, June 2013, pp. 1107-1117.
[16] A.V. Karthick, *et al.*, "An Efficient Multi Queue Job Scheduling for Cloud Computing", *in the Proceedings of World Congress on Computing and Communication Technologies (WCCCT),* IEEE, Feb. 27- March 1, 2014, pp. 164-166.
[17] S. Yang, *et al.*, "Techniques to Minimize State Transfer Costs for Dynamic Execution Offloading in Mobile Cloud Computing", *IEEE Transactions Mobile Computing*, Vol. 13, No. 11, November 2014, pp. 2648-2659.
[18] S. Pal and P.K. Pattnaik, "Adaptation of Johnson Sequencing for Job Scheduling to Minimize the Average Waiting Time in Cloud Computing Environment", in "*Journal of Engineering Science and Technology (JESTEC)"*, Taylor's University. (Article in Press).
[19] J.M.A. Calero and J.G. Aguado, "MonPaaS: An Adaptive Monitoring Platform as a Service for Cloud Computing Infrastructures and Services", *IEEE Transactions Service Computing*, Vol. 8, No. 1, January 2015, pp. 65-78.
[20] L. Tadj, "Waiting in line", *Potential, IEEE*, Vol. 14, No. 5, 1996, pp. 11-13.
[21] C. Cheng, J. Li and Y. Wang, "An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing", *Tsinghua Science and Technology*, Vol. 20, No. 1,2015, pp. 28-39.
[22] D.G. Kendall. "Some Problems in Theory of Queues", *J. Roy. Stat. Soc., Series B*, Vol. 13, No. 1, 1951, pp. 151-185.
[23] L. Li, "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers, "*Third International Conference on Multimedia and Ubiquitous Engineering (MUE '09)"*, 4-6 June 2009, pp. 295-299.

BIOGRAPHIES OF AUTHORS

Souvik Pal, Member of *CSTA/ACM*, USA, Member of *IAENG*, Hong Kong, Member of *IACSIT*, Singapore, is Assistant Professor at the Department of Computer Science and Engineering, Elitte College of Engineering, Kolkata and Ph.D. Research Scholar at KIIT University, Bhubaneswar. He has published various Research Papers in peer-reviewed International Journals and Conferences. His research area includes Cloud Computing.

Dr. Prasant Kumar Pattnaik, Senior Member IEEE (USA), Fellow IETE, is Professor at the School of Computer Engineering, KIIT University, Bhubaneswar. He has more than a decade of teaching and research experience. Dr. Pattnaik has published numbers of Research Papers in peer-reviewed International Journals and conferences. His areas of interest include Mobile Computing and Cloud Computing.