International Journal of Grid and Distributed Computing
Vol. 11, No. 11 (2018), pp.1-12
http://dx.doi.org/10.14257/ijgdc.2018.11.11.01

An Adaptive Load Balancing Algorithm using Categorization of Tasks on Virtual Machine based upon Queuing Policy in Cloud Environment

Neha Miglani[1] and Gaurav Sharma[2]

[1]Department of Computer Engineering, National Institute of Technology, Kurukshetra, Haryana-136119, India
[2]Department of Computer Science and Engineering, Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, Kurukshetra University, Kurukshetra, Haryana-136119, India
[1]neha.miglani27@gmail.com, [2]gauravsharma@jmit.ac.in

Abstract

Cloud computing is an emerging computing paradigm that allows pervasive access to shared resources and high level services over the Internet. As most of the companies are switching their network onto cloud, it is affecting the overall performance of cloud infrastructure in terms of task scheduling and load balancing. Scheduling the tasks efficiently reduces processing time, helps achieving better utilization of resources and hence provides profits to service providers. Though a lot of work has already been done in this field, yet the scope of betterment exists in terms of embedding the multiple objectives to enhance the performance. This research focuses on prioritizing tasks in such a way that the virtual machines can best utilise the resources and reduce turn-around time. The approach works on multi-objectives, i.e., time frames and job type such that the job with high computation should be assigned to VMs with high processing speed and communication-intensive jobs to VMs with high bandwidth.

Keywords: *Cloud Computing, Task Scheduling, Load Balancing, Virtual Machine, Resource Allocation*

1. Introduction

Cloud Computing is one of the latest technologies that has been considered in itself as the future of distributed on-demand computational work [1, 2]. This environment facilitates customers by providing services to them in a distant virtual platform by charging on a pay-per use basis. It provides an effective, economical, optimized platform as far as time and cost is concerned for the variety of businesses and services which are Internet-enabled. User can access the desired files at any terminal if it is Internet-enabled as well as one can utilise all the applications without deploying them [3]. It facilitates efficient approach for the deployment and scaling oriented applications [4]. This paradigm is based upon the architecture which follows the client-server model, comprising of front-end and back-end terminals networked together. A "Cloud" is the convergence of parallel and distributed system comprising number of inter-connected and virtualized nodes provisioned dynamically as well as described in the terms of computing resources based on service-level agreements which are established through negotiation between the consumers and service providers [5,16]. Cloud computing paradigm provides three types of services to the users based upon Infrastructure, Platform and Software, IaaS, PaaS and SaaS, respectively. As numerous facilities have been provided by the

Received (May 3, 2018), Review Result (July 16, 2018), Accepted (October 12, 2018)

ISSN: 2005-4262 IJGDC

Cloud Computing, henceforth; it has become the promising technology for the IT Companies as cost incurred is also less as far as cloud services are concerned in comparison to the installed services otherwise [6].

In Cloud Computing, virtualization is the backbone for improving the efficiency and optimization of cost along with the inspection and monitoring of all the nodes effectively and in parallel manner. Although it is the key factor yet it too has its limitations in the form of assigning large number of tasks to the available resources for distributed computing. Numerous factors can lead to the node in one of the two states, overloaded and underload. As such reasons for the same could be many; such as addition of new nodes over time, change in the needs and priorities of users, disrupted allocation of available resources to different tasks in hand, and occurrence of failure in the overloaded nodes [7-10].

For such reasons, the concept of scheduling is used to well utilize the resources. Scheduling aims at distributing the given load or tasks among different available resources by utilizing it to the maximum extent; eventually resulting in the reduced execution time [11]. The process of task scheduling can be done in two ways: Static task scheduling and Dynamic Task Scheduling [12]. The term static refers to allocating all the resources and data before initiating the actual manipulations whereas the dynamic scheduling signifies allocation for the task or job in hand is not done already rather it is done on the fly and information is not available in the beginning [13]. In this case, complexity gets increased due to dynamic nature of the problem.

For overpowering the described problem, Load Balancing is the key concept which can be merged with the task scheduling to enhance the efficiency and effectiveness of the system. Load balancing comes into the effect ensuring the transparency of services regardless of the cloud location as well as physical implementation [10]. In the past years, various authors have diverted their focus to this area of load balancing and significant amount of work has already been done and many algorithms have been designed [14-22]. This paper aims at balancing the load while scheduling the tasks dynamically to make optimized use of available resources. The rest of the paper is structured as follows: Section 2 explains the related researches briefly and presents the analysis of scheduling schemes. Section 3 presents the methodologies utilized in the paper. Section 4 provides the experimental results and their discussions. Section 5 concludes the research.

2. Literature Review

In Cloud Computing framework, Scheduling and Load Balancing plays an essential role to enhance the efficiency and optimizing various resources by providing quality of service (QoS) guarantees. The concept of Load Balancing helps overcome an issue of overutilization and underutilization of virtual nodes resulting in minimized cost and response time and increased throughput [15, 23, 24]. The focus of this section is not to propose methodologies to fix the problems and issues of cloud computing but to have an in-depth study of existing work and techniques designed by various researchers to target the problem.

An active monitoring Load Balancer Algorithm was proposed based upon maintaining the database of all the virtual machines of how many requests have been allocated to all the VMs individually [24]. When new job arrived, the database was used to find the Virtual Machine which was least loaded and hence the new job was assigned to respective Virtual Machine.

Modified throttled algorithm was designed for balancing the load by distributing upcoming requests among different virtual machines [15]. Cloud Analyst simulator was used for calculating the performance and efficiency.

An algorithm for Virtual Machine Assignment was proposed [30], which further helped distributing all the coming requests to available virtual machines in an effective

manner. The proposed algorithm focussed to well- utilize the Virtual Machines or resources in comparison to already existing approaches for load balancing. CloudSim Simulator was used for the manipulations.

New Time Optimizing Probabilistic Load Balancing Algorithm was derived, targeting the load management and reduction of response time by fetching the resources based on better past history and least completion time [28].

The concept of burden adjustment in distributed computing environment was elaborated with the help of disseminated approach [25]. Few existing strategies for burden adjustment had also been discussed considering various parameters like adaptability, execution, asset usage, adaptation to non-critical failure, acquainted overhead.

Dynamic load management algorithm was derived which targeted the load distribution among available virtual machines uniformly and efficiently. Various parameters such as data processing time and response time were considered for measuring the performance of proposed approach [29].

The load balancing mechanism using ant colony optimization (ACO) was presented for balancing the work of job assignment in a dynamic manner. Two strategies, forward-backward ant mechanism and max-min rules were used in the proposed approach [10]. The technique focussed not only the computing time but also the network performance for medium and heavily loaded machines.

In order to optimize the performance of cloud storage by combining de-duplication and access point selection optimization techniques, a new data center management architecture named, INS (Index Name Server) was investigated [26]. Various other parameters were also embedded, like IP information and busy level index, resulting in attaining load balancing and avoidance of congestion problem.

Three different approaches based upon Honeybee Foraging Behaviour, Biased Random Sampling and Active Clustering were proposed to resolve an issue of load balancing and thus, provided three different distributed solutions. As job assignment to specific servers cannot be centralized, therefore, an effective distribution solution was required to overcome the complexity and scalability issues of the system [14].

From the literature review, it can be seen that many researchers have focussed on the domain of balancing the load. Various algorithms have been proposed for the distribution of the jobs or tasks to the Virtual Machines. Many other algorithms [5, 19, 27] have also been discussed in the previous researches. Though a lot of work has been done in the field, yet the concept of global queues is still untouched. This paper aims at balancing the load by handling the upcoming requests at global and local level resulting in optimized results and precise approach.

3. Proposed Model and Architecture

The proposed architecture targets the load balancing by assigning the new jobs in such a manner that it avoids the over-utilization and under-utilization of virtual machines so that the trade-off between the cost and performance can be made efficiently. Figure.1 illustrates and presents the proposed model and rest of the section describes the components and execution phase of the model presented.

3.1 Pre-execution Phase

The discussion and description of actual execution demands some pre-requisites in terms of what components should be tuned into before starting the executable phase.

i) Classification of Virtual Machines: This component is responsible for describing the configuration of Virtual Machine based on the previous historic data available in context with that virtual node. Virtual machines are classified in accordance with the upcoming jobs in the queue. Virtual machines cluster the jobs on the basis of MIPS and Bandwidth and hence, there are two categories: communication oriented virtual

machines depending upon the bandwidth requirement and computation oriented virtual machines which are based upon the processing speed, i.e. MIPS. Identifying similar type of jobs and assigning it to the node respective to the category reduces the time and effort required to provide services to the new task.

ii) Job Type: Whenever new job arrives, it is beneficial if the nature of job can be predicted because in such case, resources can be allocated in accordance with it. There are numerous factors which should be considered for the upcoming jobs such as quality, processing speed, bandwidth, CPU time, memory requirements and length of the upcoming tasks. Based on such parameters, two categories have been defined-communication intensive tasks and computation based tasks. Computation based tasks considers the processing speed *i.e.*, Million Instructions per Second (MIPS) and communication intensive tasks takes bandwidth into consideration. Once the upcoming task is categorized, it is forwarded to the Local Queues for further processing.

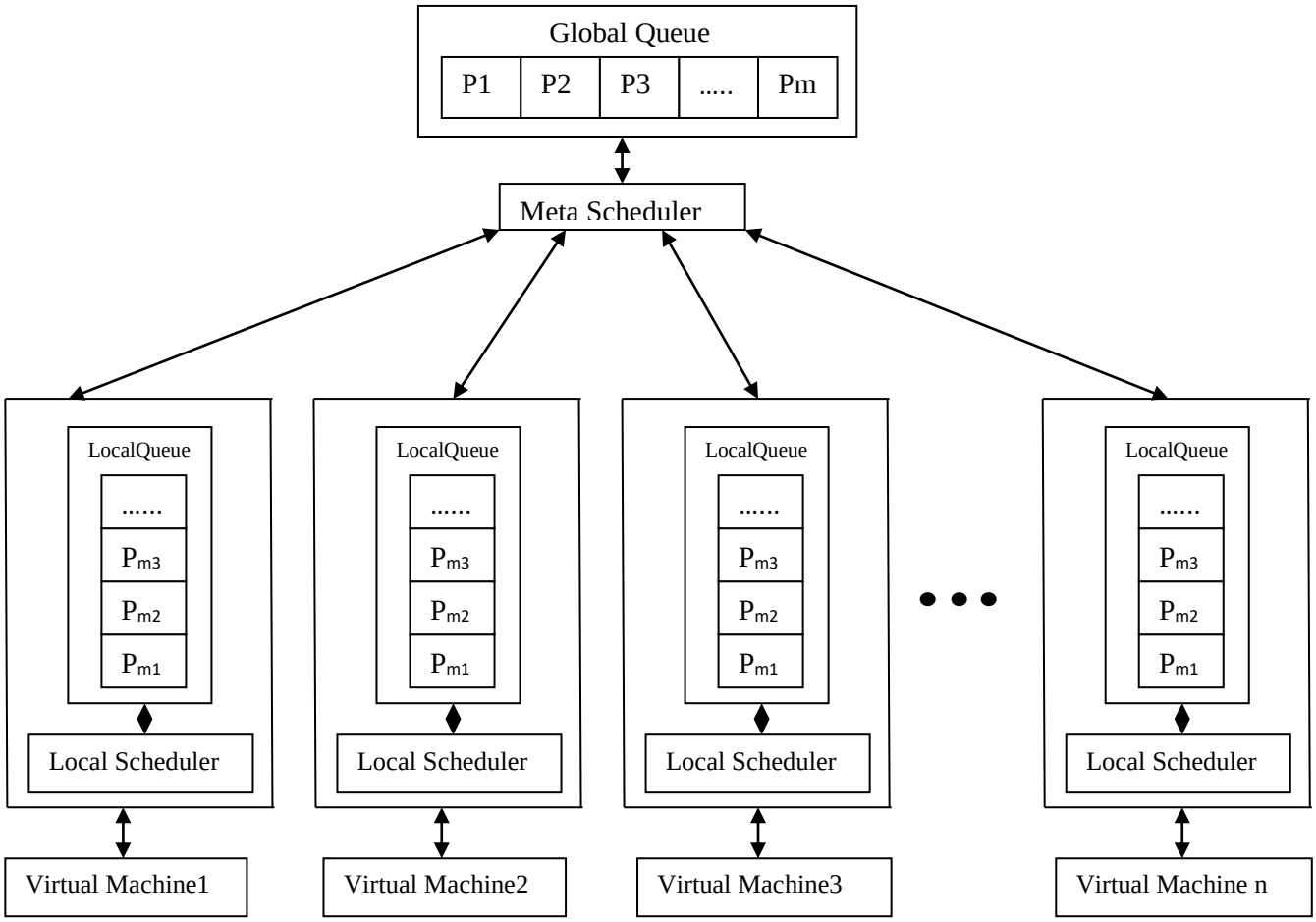

Figure 1. Proposed System Architecture for Task Scheduling

3.2. Implementation Phase

The proposed model targets the utilization of resources by sharing the jobs into available Virtual Machines. This approach focuses on balancing the load by making the use of feedback mechanism on the basis of time frames and job types, which is discussed in detailed in the rest of the section. It comprises of the following components:

i) Meta-scheduler: Categorization of new tasks is the focus of meta-scheduler.

ii) Local Scheduler: It further divides the task on the basis of min-min algorithm.

iii) Global Queue: It comprises of all the upcoming tasks.

iv) Local Queue: It has a collection of tasks specific to the type of Virtual Node it has been connected to.

All these components are inter-connected as depicted in the Figure 1. The manipulation, addition and deletion of tasks is done in co-ordination and synchronized manner.

Algorithm-TaskSchedule(): Assignment of Tasks in Optimum Fashion to available Virtual Machines
Input: RunningTask(), MIPS, Virtual Machines(vm_i), Local Queue(lq_i), Global Queue(gq_i), NewTask()
Output: Tasks Scheduled in an optimized and efficient manner.

```
BEGIN
Initialise all Local Queues and Global Queues to empty.
   Logically categorize VMs on the basis of MIPS and Bandwidth.
   For upcoming newtask() in global queue do
         Check criticality of newtask()
         If task-criticality(bandwidth)> task-criticality(MIPS)
         Update local scheduler of VMs with High Bandwidth
         Call LoadBalance()
         Else
         Update local scheduler of VMs with High MIPS
         Call LoadBalance()
For RunningTask() of VMs do
      Select the NewTask() from Local Queue (lqi) based on Min-min algorithm and assign it to
VM for execution.
   For each RunningTask() do
      vmi.update()
       lqi.update()
      gqi.update()
   task() killed
END
```

User submits the task along with the desired parameters on the priority basis, namely quality of service, processing speed, bandwidth *etc*. As soon as new task arrives, it is added into a global queue.

Algorithm-LoadBalance(): Assignment of new task to the Virtual Machine which has lesser load
Input: Virtual Machines(vm_i), remaining processing time (tp_i), NewTask()
Output: Task Assigned to VM satisfying the Load Balancing property.

```
BEGIN
For each Virtual machine vmi
 Check remaining processing time tpi to complete the assigned tasks.
 Min=infinity
 For i=1 to n
       If tpi<min
       Min=tpi
Assign the NewTask() to VM with minimum remaining processing time, min
END
```

the task to the local queues based upon the nature and priority of virtual machines, that is, if VM has high computation power then submit computational intensive job to corresponding VM Local Queue and if it requires high bandwidth then submit the upcoming job to local queue corresponding to high bandwidth Virtual Machine.

After the filtering of tasks to the respective queues on the basis of priority and job type, next is the challenge of prioritizing the tasks in local queues. For such purpose, Local Scheduler comes into role while applying min-min heuristic onto the TaskList and all the tasks are mapped to the respective Virtual Machines.

Besides allocating the tasks to the Local Queues, meta scheduler also performs the load balancing by using the principle of feedback, that is, it takes the feedback from local scheduler and predict the availability time of different Virtual Machines. While assigning the new task to any of the Virtual Machines, database of Virtual Machines is formed comprising of number of existing tasks and total time duration required to perform the assigned tasks. New task is assigned to the VM having less number of tasks and lesser completion time, that is, lesser remaining processing time, to balance the load.

4. Experimental Setup

The proposed approach has been implemented in simulator environment *i.e.*, CloudSim to simulate the task scheduling algorithm and hence, to verify the efficiency of the proposed work. The experimental environment is setup with the configurations as described: OS: Windows 7; CPU: Processor 3.20GHZ; Memory: 8G. DataCenterBroker Class has been included in the CloudSim platform for simulating the experiment. As the end user quality is the key factor, therefore, for such reasons, CloudSim has been used as a simulator because otherwise it would become a difficult task to experiment a new technique in real cloud environment. Results have been derived and are depicted in the figures below.

In order to perform the comparisons and check the efficiency of proposed approach, results obtained are compared with the existing approaches namely, FCFS, SJF, min-min approaches [17, 18] on the basis of two parameters makespan time and resource utilization.

Table 1. Parameters for Virtual Machines

VM Id	VM MIPS	Size	Memory	No. of CPU	VMM
0	1000	1000	512	1	Xen
1	580	1000	1024	1	Xen
2	700	1000	512	1	Xen
3	550	1000	1024	1	Xen
4	400	1000	256	1	Xen
5	250	1000	1024	1	Xen

Every single algorithm considered runs 10 times to remove the impact of random factors on the evaluated result and eventually, average value calculated is considered as the basis for comparison.

Simulation has been done on six heterogeneous virtual machines (having varying bandwidth and processing speed) by running for more than three hours (approximate 10 times) on varying number of tasks (task range 10 to 50) along with different length cloudlet to yield the final results.

Table 2 comprises task parameters in which 10 tasks of random length, which are independent of each-other, are generated with specified range.

Initially, 10 tasks on 6 Virtual machines are taken into consideration which are gradually increased to 20, 30 and so on uptil 50 of random length and makespan time of

virtual machines as well as resource utilization ratio is evaluated and compared with the existing techniques. Three different results are fetched to prove the efficiency of proposed approach. Results shown in Figure 2 assure that proposed algorithm is performing better in comparison to existing algorithms. In Figure 2, *x*-axis represents the number of tasks in increasing order and y axis represents the calculated makespan time of task using the proposed algorithm. As the general tendency is to have minimum makespan time in order to achieve optimized results; the Figure 2 is depicting the same as the makespan for varying values of tasks is lowest in comparison to the existing approaches.

Table 2. Task Parameters

Cloudlet Id	Length	No. of CPU
0	89424	1
1	32273	1
2	345002	1
3	388452	1
4	251227	1
5	65833	1
6	311431	1
7	234568	1
8	35677	1
9	15820	1

4.1 Resource Utilization Ratio

Expression to calculate Resource utilization ratio is:

$$\textit{Resource Utilization Ratio} = \textit{mean-time/makespan} \times 100 \quad (1)$$

where mean-time = $\sum$ Time taken by resource (VM_j) to finish all the allocated jobs where j = {1, 2, 3 ...n}, the valid range of Resource Utilization Ratio lies between 0 to 1, where 1 signifies 100% resource utilization and 0 depicts resource is in ideal condition. Resource Utilization Ratio has been calculated with the help of proposed algorithm using the Equation (1) and results are compared with the existing approaches as shown in Figure 3. Consider the Table 1 and Table 2 for finding the resource utilization of available cloud resources form proposed algorithm and others algorithms. Resource utilization is supposed to be high, if results so obtained are optimum. The proposed approach has successfully increased the ratio of resources utilized, when compared with the exiting techniques.

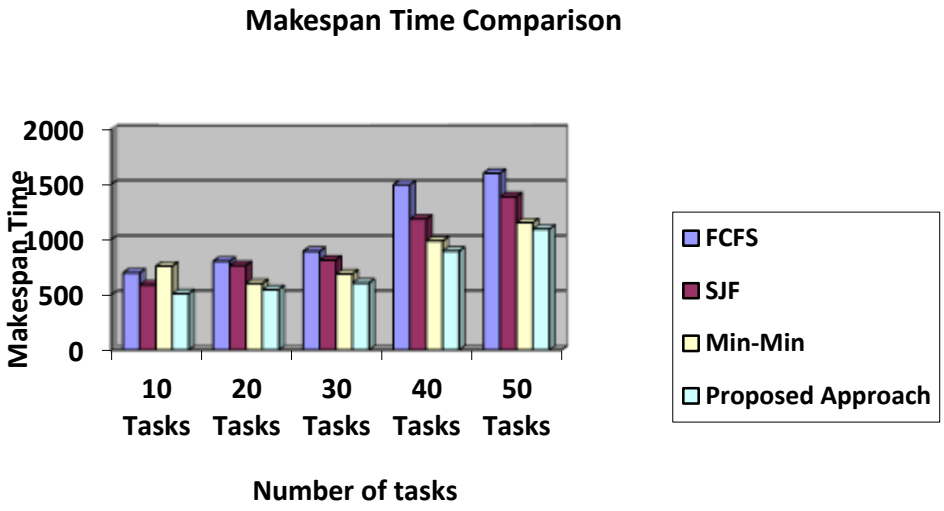

Figure 2. Makespan Comparison Chart

Table 3. Makespan Time for Varying Number of Tasks

Number of Tasks / Approach	10 Tasks	20 Tasks	30 Tasks	40 Tasks	50 Tasks
FCFS	699	805	895	1498	1602
SJF	590	765	815	1187	1390
Min-Min	760	602	689	988	1154
Proposed Approach	510	546	607	897	1096

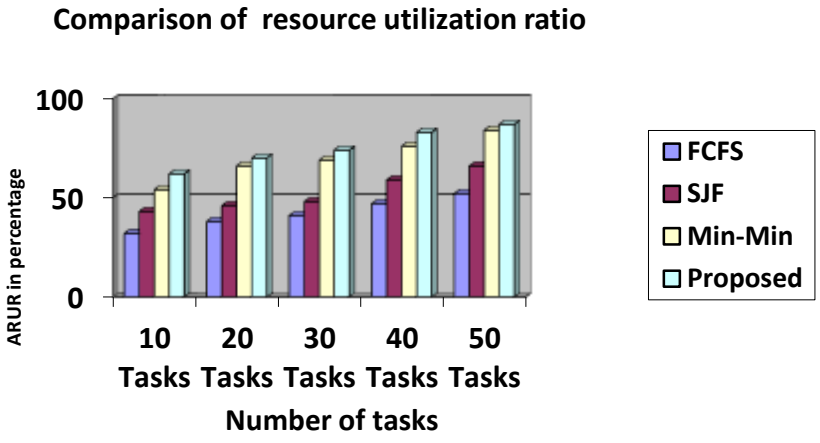

Figure.3 Resource Utilization Ratio Comparison Chart

Table 4. Resource Utilization Ratio for Varying Number of Tasks

Number of tasks / Approach	10 Tasks	20 Tasks	30 Tasks	40 Tasks	50 Tasks
FCFS	32	38	41	47	52
SJF	43	46	48	59	66
Min-Min	54	66	69	76	84
Proposed	62	70	74	83	87

Calculated results of Resource Utilization Ratio from the proposed algorithm shows that proposed algorithm utilizes the cloud resources 79.04% better than FCFS, and43.51% better than SJF, and 7.73% better than min-min algorithm.

5. Conclusion and Future Scope

The Experimental result of Proposed Algorithm has led to some important conclusions. The research explored and considered more than one factors such as job type, time frames for existing tasks in Virtual Machines resulted in better utilization of resources in comparison to conventional approaches. Availability of Virtual Machines and length of tasks have been considered for the Load Balancing by using the feedback mechanism. Experimental results also show that the proposed task scheduling mechanism provided better performance in terms of execution time, computation cost, communication cost, and bandwidth and CPU utilization when compared with existing task scheduling approaches.

The proposed work may be elaborated and extended in future to enhance the performance and efficiency. Currently, the non-preemptive scheduling is considered. In order to aim and cover other domains as well, pre-emptive scheduling may be considered.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing", UCB/EECS-2009-28, **(2009)**.

[2] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters", Sixth Symposium on Operating System Design and Implementation (OSDI'04), **(2004)** December, pp. 1-13.

[3] Q. Duan, "Modeling and Performance Analysis on Network Virtualization for Composite Network-Cloud Service Provisioning", Proc. of SERVICES, **(2011)** July, pp. 548-555.

[4] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications", IEEE J. Sel. Areas, Commun. vol. 14, no. 7, **(1996)** September, pp. 1228-1234.

[5] R. Buyya, C. Shin Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility", Future Generation Computer Systems, Elsevier Science, Amsterdam, vol. 25, no. 6, **(2009)** June, pp. 599-616.

[6] M. A. S. Mosleh, G. Radhamani, M. A. G. Hazber and S. Hamid Hasan, "Adaptive Cost-Based Task Scheduling in Cloud Environment", Hindawi Publishing Corporation Scientific Programming, vol. 2016, Article ID 8239239, 9 pages http://dx.doi.org/10.1155/2016/8239239.

[7] Q. Zhang, L. Cheng and R. Boutaba, "Cloud computing: State-of-the-art and research challenges", J. Internet Serv. Appl., vol. 1, **(2009)** June, pp. 7-18.

[8] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems", J. Supercomput., vol. 60, **(2010)**, pp. 268-280.

[9] B. P. Rimal, E. Choi and I. Lumb, "A taxonomy and survey of cloud computing systems", In Proceedings of the Fifth International Joint Conference on INC, IMS and IDC, 2009. NCM'09, Seoul, Korea, **(2009)** August 25-27, pp. 44-51.

[10] R. Gao and J. Wu, "Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization", Future Internet, doi:10.3390/fi7040465, vol. 7, **(2015)**, pp. 465-483.

[11] A. Y. Zomaya and Y. The, "Observations on using genetic algorithms for dynamic load-balancing", IEEE Transaction on Parallel and Distributed Systems, vol. 12, no. 9, **(2001)**, pp. 899-911.

[12] E. Ibrahim, N. A. El-Bahnasawy and F. A. Omara, "Dynamic Task Scheduling in Cloud Computing Based on the Availability Level of Resources", International Journal of Grid and Distributed Computing, http://dx.doi.org/10.14257/ijgdc.2017.10.8.03, vol. 10, no. 8, **(2017)**, pp. 21-36.
[13] R. Kannan, R. U. Rasool, H. Jin and S. R. Balasundaram, "Managing and Processing Big Data in Cloud Computing", vol. i, 1AD.
[14] M. Randles, D. Lamb and A. Bendiab, "A comparative Study into distributed load balancing algorithms for cloud computing", IEEE 24th International Conference on Advanced Infonnation Networking and Applications Workshops (WALNA), **(2010)** April, pp. 551-556.
[15] S. G. Domanal and G. Ram Mohana Reddy, "Load Balancing in Cloud Computing Modified Throttled Algorithm", IEEE, International conference, CCEM, **(2013)**.
[16] R. K. Bawa and G. Sharma, "Reliable resource selection in grid environment", International Journal of Grid Computing & Applications, vol. 3, no. 1, **(2012)** March, pp. 1-10.
[17] R. K. Bawa and G. Sharma, "Modified Min –Min Heuristic for Job Scheduling based on QoS in Grid Environment", IEEE Conference on International Management in the Knowledge Economy, **(2013)** December 19-20.
[18] A. Bikramjit Singh, S. Bhat J., R. Raju and R. D'Souza, "A Comparative Study of Various Scheduling Algorithms in Cloud Computing", American Journal of Intelligent Systems, DOI: 10.5923/j.ajis.20170703.06, vol. 7, no. 3, **(2017)**, pp. 68-72.
[19] D. Singh Chawla and K. Singh Dhindsa, "A Load Balancing Based Improved Task Scheduling Algorithm in Cloud Computing", International Journal for Research in Applied Science & Engineering Technology, ISSN: 2321-9653, vol. 5, no. IX, **(2017)** September.
[20] V. Sesum-Cavic and E. Kuhn, "Applying swarm intelligence algorithms for dynamic load balancing to a cloud based call center", In Proceedings of the 2010 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, Budapest, Hungary, **(2010)** September 27, pp. 255-256.
[21] Z. Liu and X. Wang, "A PSO-Based Algorithm for Load Balancing in Virtual Machines of Cloud Computing Environment", In Advances in Swarm Intelligence; Lecture Notes in Computer Science; Tan, Y., Shi, Y., Ji, Z., Eds.; Springer: Berlin, Germany; Heidelberg, Germany, vol. 7331, **(2012)**, pp. 142-147.
[22] X. Feng and Y. Pan, "DPSO Resource Load Balancing in Cloud Computing", Comput. Eng. Appl., vol. 11, **(2011)**, pp. 1-8.
[23] R. Beri and V. Behal, "Cloud Computing: A Survey on Cloud Computing", International Journal of Computer Applications, vol. Ill, no. 16, **(2015)**.
[24] H. S. Mahalle, P. R. Kaveri and V. Chavan, "Load Balancing On Cloud Data Centers", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. I, **(2013)**.
[25] A. Vig, R. S. Kushwah and S. S. Kushwah, "An Efficient Distributed Approach for Load Balancing in Cloud Computing", International Conference on Computational Intelligence and Communication Networks, IEEE, **(2015)**.
[26] Y.-S. Lin, T.-Y. Wu, W.-T. Lee, Y.-S. Lin, H.-L. Chan and J.-S. Huang, "Dynamic load balancing mechanism based on cloud storage", Proceedings of the Computing, Communications and Applications Conference (ComComAp), Hong Kong, China, **(2012)** January 11-13, pp. 102-106.
[27] Z. Bo, G. Ji and A. Jieqing, "Cloud Loading Balance algorithm", In Proceedings of the 2010 2nd International Conference on Information Science and Engineering (ICISE), Hangzhou, China, **(2010)** December 4-6, pp. 5001-5004.
[28] M. Moradi, M. A. Dezfuli and M. H. Safavi, Department of Computer and IT, Engineering, Amirkabir University of Technology, Tehran, Iran", A New Time Optimizing Probabilistic Load Balancing Algorithm in Grid Computing, IEEE, **(2010)**, pp. 232-237.
[29] R. Panwar and B. Mallick, "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm", 2015 International Conference on Green Computing and Internet of Things.
[30] S. G. Domanal and G. Ram Mahana Reddy, "Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines Center, IEEE Conference, **(2014)**.

Authors

Neha Miglani, she has received her Master Degree in Computer Science from Kurukshetra University, India. Currently, she is working as an Assistant Professor in National Institute of Technology, Kurukshetra, India. Her research interest includes Cloud Computing, Software Reliability ranging from Cost Models, Software Reliability Growth Models, and Reliability metrics, etc.

Dr. Gaurav Sharma, he has received his Master Degree in Computer Science from Guru Jambeshwar University, India. He received Ph.D. degree from Punjabi University, Patiala, India. Currently, he is working as an Associate Professor in Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, India. His research interest covers Cloud Computing, Grid Computing, Software Engineering and Artificial Intelligence. He has published more than 35 Research Papers in various National and International Journals and Conferences.

