Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130317

Scheduling and Load Balancing using NAERR in Cloud Computing Environment

J. Arul Sindiya [1,*] *and R. Pushpalakshmi* [2]

[1] Department of Computer Science and Engineering, CARE Group of Institutions, Trichy,Tamil nadu India.
[2] Department of Information Technology, PSNA College of Engineering and Technology, Dindigul,Tamil nadu India.

Received: 6 Oct. 2018, Revised: 3 Feb. 2019, Accepted: 9 Feb. 2019
Published online: 1 May 2019

Abstract: Scheduling plays an important role in cloud computing to achieve effective load balancing by migrating tasks to partially utilized Virtual Machines (VMs). This sharing of resources provides effective scheduling in which non preemptive tasks are irretrievable constraints in cloud computing environment. Therefore, these non preemptive tasks should be initially allocated to the most suitable VMs itself. Basically, each jobs entering comprises of several interconnected tasks which may be executed by multiple VMs or different cores of a single VM. Moreover, the jobs are arrived during the server run time at random time intervals with different load conditions. In order to provide efficient cloud computing, static or dynamic scheduling techniques are used to allocate the tasks to the suitable resources and by which the involved heterogeneous resources are organized. Hence, the user satisfaction is improved. In this paper, a Novel and Adaptive Enhanced Round Robin (NAERR) algorithm is proposed which computes the size and length of all requesting jobs, the capabilities of all available VMs, and the interconnection among the tasks. The proposed and existing techniques are compared to prove the performance of the proposed algorithm.

Keywords: Scheduling, load balancing, cloud computing, NAERR algorithm, virtual machines.

1 Introduction

Cloud computing is defined as a typical computing model to manage and deliver services through the Internet. It is known as a pattern to enable on-demand, advantageous, and pervasive network access to the computing resources (for example, application, network, service, server and storage) which are in shared pool. These shared resources are frequently furnished and discharged with insignificant administration attempt or the interaction of service providers [1]. Both parallel and distributed computing concepts are integrated to produce cloud computing and it shares some of the computing resources such as software, hardware, data and other peripherals. A user can access the previously mentioned resource through cloud computing with internet facilities by paying for the period of utilization. In cloud computing technology, the virtual machines (VMs) act as a base in which executions are performed i.e. it is known as execution unit. Different applications and resources require the concept of virtualization that comprises of three phase such as formation, execution and host management. There are few resources that are shared among the VMs in the environment of cloud computing like buses, systems and processing units,etc. Each VM can constrain its available computing resources by calculating each of its overall processing power. In such environmental model, it is unpredictable for the job arrival pattern and generally each virtual machine has different capabilities. In order to achieve better performance and stability, it is necessary to provide load balancing which becomes a complex task. Hence, it is essential to propose a novel algorithm that can enhance the network performance by providing balanced workloads among the VMs. Some of the existing load balancing algorithms is ant colony algorithm, dynamic load balancing, Equally Spread Current Execution (ESCE) algorithm, First In First Out (FIFO), round robin, throttled algorithm, Weighted Round Robin (WRR) and so forth. Among which the FIFO and WRR are the most commonly employed scheduling algorithms for processing no preemptive tasks [2]. To perform research work on cloud computing CloudSim-3.0.3 is used as the simulation environment. The cloud computing components (such as, data ware house, VMs, hosts, and

[*] Corresponding author e-mail: sindiyaj6@gmail.com

some resource provisioning strategies) possess both behavior and system modeling which are supported by CloudSim-3.0.3 environment which supports designing and implementation of cloud computing comprising of both individual and inter-networked clouds. The implementation of scheduling and load balancing policies into virtual machines introduced custom interfaces. Under inter-networked cloud environment, the provisioning strategies are required to allocate VMs which can strengthen virtualization-based services even on Quality of Services (QoS) and workload requirements that vary with time.

1.1 Our Contribution

In cloud computing, the computational tasks are assigned to the most appropriate VMs which are present in the dynamic pool by computing the loads of virtual machines and specifications of every task. Each user requests are forwarded to any of the servers or data centers in the cloud environment. Based on the policies of cloud management, the received user requests are sent to the most appropriate VMs by the servers or data centers depending on the load of the virtual machines. There are two major task scheduling techniques which are available in no-preemptive scheduling such as round robin scheduling and the Weighted Round Robin (WRR) scheduling techniques. The round robin scheduling never considers some of the parameters like resource capability, task priority and task length. Therefore, the lengthy task with higher priority is completed with higher reaction time. Rather, the Weighted Round Robin scheduling has considered the resource capability of each virtual machine and allocates more number of tasks to the high capacity VM depending on their weight age assigned to all VMs. In order to choose suitable VMs it fails to consider the task length. To overcome these drawbacks, we propose Novel Adaptive Enhanced Round Robin (NAERR) Algorithm which considers all three parameters such as resource capability, task priority and task length and it chooses the suitable virtual machine to perform the obtained tasks with lower reaction time. Our aim is to perform performance optimization of VMs by applying dynamic load balancing techniques by computing job lengths, interdependencies of available tasks, capacity of resources, predicting lower utilized virtual machines and excluding higher utilized VMs. The tasks are effectively scheduled to the perfect VMs by considering the task length as an additional parameter and it can able to provide the reaction in a minimal run time. This effective task scheduling of the proposed algorithm can reduce overloads to VMs and the task relocations are reduced. The analysis of the proposed algorithm is used to observe the performance which is then compared with the conventional RR and WRR task scheduling algorithms. Let us consider that each job is comprised of multiple tasks which are have interdependencies among them. To finish the complete processing instructions, each job can utilize several VMs for its multiple tasks. Depending on the availability and configuration of VMs, the tasks can utilize their multiple processing elements.

2 Related work

In cloud environments, no preemptive tasks need more concern since the load balancing of such tasks on VMs is considered as the essential characteristics of task scheduling. Optimal resource utilization is achieved by sharing the additional load of overloaded VMs with under loaded VMs for the minimum task completion time. In addition, the overhead effect on recognizing the resource usage and task relocation must be taken into account in the load balancing algorithms. Extensively, there are two separate task execution techniques used by the VMs such as space and time sharing techniques. The execution of tasks follows one by one manner in space shared technique which infers that only a single task on CPU per core is executed at a time in its CPU. The other tasks are waited in the waiting queue of the VM which is assigned to them. Thus, it can be known that in load balancing, the task migration are effortless on such space sharing technique by detecting a task waiting in the queue of an overloaded VM and then it is assigned to some under loaded VMs.

In time sharing technique, the tasks are performed in a time slicing manner simultaneously that resembles the parallel mode task execution techniques. Because of this time sliced implementation of entire tasks, the task migration will be highly complex in load balancing. Thus, a specific amount of is tasks only completed nearly 90the time using time slicing technique. The choice of detecting the task relocation from the overloaded to under loaded VMs is much expensive because the formerly completed segments are lost in the overloaded VM and the priori execution of jobs create impacts on the execution of other jobs in the overloaded VM. Therefore, the optimal or minimal task relocation should be attained by the scheduling and load balancing algorithms with uniform distribution of load among the resources according to its capacity without any jobless time of any resources at any time in the net consolidated resource run time. This technique achieves the optimal or minimal run time in the environment of cloud computing. Also, the algorithm has to observe the unpredictable job's arrival nature and its job assignment to the appropriate VMs by taking into account of multitask jobs and the interdependencies among them. The algorithm must be appropriate for both homogenous and heterogeneous network environments with different job lengths. In this section, some literatures are analyzed to reach the objective of the proposed technique.

M. Kumar et.al [9] has integrated two concepts of IBA and EASY algorithms to propose a novel scheduling algorithm for cloud computing. All of the previously proposed scheduling algorithms have its own advantages and disadvantages, for example, a single scheduling algorithm cannot able to improve all the essential parameters. Several authors select several parameters to improve the performance by maximizing/minimizing them, such as, reaction time, energy consumption, ratio of resource utilization, makespan time, and so on. In distributed systems, the task scheduling produces a non-deterministic complete issue which cannot be tackled at polynomial time. Consequently, a few researchers utilize the soft computing strategies to discover the sub optimal output of such issue.

Dhinesh Babu and Venkata Krishna [2] have presented a load balancing algorithm which is inspired by the behavior of honey bee that focuses to accomplish effectively balanced loads throughout the VMs to increase the network throughput and the task priorities have to be balanced o the VMs. Subsequently, it achieves minimal waiting time of the tasks in the task waiting queue. With the help of this proposed algorithm reduced execution time and minimal waiting time of the tasks on the queue were enhanced. This algorithm is proposed only for the heterogeneous networks to balance non preemptive independent tasks.

Ramezani and Khadeer hussain [11] proposed Particle Swarm Optimization (PSO) based load balancing algorithm which does not relocate the VMs but it relocates the tasks of higher loaded VMs to under loaded VMs. This algorithm has achieved reduced response time and task relocation time by the way it provides better system performance.

J. Cao et al [3] presented the importance of minimum power consumption and performance optimization in cloud data centers to design waiting queue for a set of multi-core heterogeneous servers which have various sizes and different speeds were analyzed. Specifically, it dealt the uniform load distribution and optimal power assignment problems for multiple multi-cores heterogeneous servers throughout the cloud data centers. In any case, it is just a practicality consider for power modeling.

R. Naha, M. Othman [17], and Somasundaram et al [18] suggested a task scheduling based broker architecture for load balancing in cloud environment by employing eucalyptus and cloud analyst tools to perform implementation. A major drawback of this proposed technique is that the load balancer cannot able to sustain the sessions among different virtual occasions since it doesn't possess the feature of session affinity.

Ghanbari and Othman [**?**] have presented a job scheduling technique for cloud computing which consider the job's priority as the major parameter for QoS. Besides, this technique takes three essential problems such as complexity, consistency, and makes span.

3 Proposed Technique

The proposed techniques involves the following three phases such as,

1. Static and dynamic scheduling 2. Task scheduling and resource monitoring 3. Load balancing with minimum processing time

3.1 Static and Dynamic Scheduling

The static schedulers have to perform an operation to discover the most appropriate VMs to allocate arrived tasks to them in accordance with the algorithms implemented in the schedulers such as Simple Round Robin (SRR), Weighted Round Robin (WRR), and Improved Weighted Round Robin (IWRR). The dynamic schedulers have to perform an operation to assign the run-time arrival tasks to the most appropriate VMs by choosing the under-utilized VMs at the time of the specific task arrival. With the assistance of the resource monitor information, the scheduler or load balancer determines the task relocation from highly loaded VMs to the unoccupied or empty VMs or under-loaded VMs at runtime when it observes empty or under-loaded VMs. Resource monitor makes communication with all the participating resource probers of VMs and gathers some information such as capacity of VMs, current workload on every VM, number of executing jobs and number of waiting tasks in queues in every VM to determine the suitable VMs for the tasks. The length of each task is computed by the task requirement estimator and transmits the computed results to the scheduler or the load balancer to make its operational decisions.

3.2 Task Scheduling and Resource Monitoring

In order to attain minimal run time and mean resources utilization, all the tasks must be scheduled to execute number of virtual machine to limit the cost and make span time when the requirement of the CSP is to use the resources of the cloud to the extent. In cloud environment, 'N' numbers of task requests (i.e. T1, T2, T3 ... TN) reach the task scheduler which are on the basis of non priority and independent nature, that is, each task does not depend on anyone for its execution. This technique does not have any preemptive types of services and when all the tasks are assigned to VMs, it runs the entire tasks before it begins to run the following tasks. Moreover, the task deadlines are not considered in this paper. The length of each task is expressed in Million Instructions (MI) and the processing speed, number of Central Processing Unit (CPU) and main memory size are demonstrated as P, Q, and R respectively. The bandwidth required between two virtual machines a and b is Ba,b in Mbps. If a task scheduler is comprised of information of heterogeneous

VMs, then they possess various processing capabilities in terms of bandwidth, memory, speed and so on. At first, the capacity of all participating virtual machines is calculated. The Dalgaard and Strulik model shares the technical properties with the Solow model. In particular, there exists a unique globally stable steady-state to which the economy adjusts.

Algorithm: Proposed NAERR algorithm

```
1) Generate number of tasks 1 T , 2 T 3 T ,. . . n T
2) Sort the task in decreasing order based upon their length
3) Arrange the VMs based on short execution time to long execution time; VM Map = VMs // VM Map Monitoring VMs performance.
4) VM Map indicates “A0” and “B1” “AB”
(i) A0 = under load short execution time (ULSET);
(ii) B1 = over load long execution time (OLLET);
(iii) both are grouped calculate the threshold ;
(IV) VM load performance (Queue) Process
5) After the task transfer check the status of each VM,
6) For (VMs Total Load less than Data center);
7) end process.
```

3.3 Load balancing with minimum processing time

In this paper, We presented a Novel Adaptive Enhanced Round Robin (NAERR) Algorithm which uses dynamic load balancing. The aim of the proposed algorithm is to minimize the execution time and maximize the average utilization ratio of resources in cloud computing environment. Here ‘N’ number of tasks is created and length of these tasks is produced between 20000MI and 400000 MI in a random manner and then ‘M’ number of heterogeneous VMs is created. Every VM possess different execution power based on the processor speed in RAM,MIPS and so forth. By applying the bubble sort algorithm, all the tasks and VMs are sorted in descending order of their processing speed and task length. Then the task allocation to virtual machines is initiated that follows First In First Out (FIFO) manner until completion of all tasks in the queue. We have computed the size of each VM and have create the array with the same size which comprises the assigned task IDs of all VMs. When a task is assigned to a virtual machine then it starts the operation of load balancing and discovers the status of all task assigned VMs, that is, to begin to monitor the VMs. Then each VM workloads and total loads are calculated in the data servers at specific time by applying the equation and discovering the capacity of all VMs and data servers. After that,one initiates to check the given condition whether the forthcoming load on each VM is less than the capacity of VM and forthcoming workload at the data server is less than the data server capacity. The load balancing is performed only when the above said condition is true in traditional cloud environment. The VMs are divided into three types based on their load conditions such as under-loaded VMs, balanced VMs, and over-loaded VMs. For each VM a threshold value is set and if the utilization of a VM is less than 25capacity, then it is defined as an under-loaded VM. Likewise, if the utilization of a VM is more than 80capacity, then it is set to be an over-loaded VM. From this, two different variables UL and OL are defined to represent 25VMs and over-loaded VMs. (After computing ULSET and OLSET, the ULSET is sorted in an ascending order and the OLSET is sorted in a descending order and it begins to transfer the tasks of over-loaded VMs to under-loaded VMs. The time required to transfer tasks from a virtual machine to another virtual machine is estimated which is then integrated to the task run time and task make-span time. The task transfer time is defined as the ratio between task lengths to the bandwidth between two VMs that involve task relocation. The results are obtained by applying the proposed NAERR algorithm for load balancing. In traditional VMs, if such condition is not correct, then it is not possible to perform load balancing, that is, a new VM is booted by the cloud resource broker using the concept of horizontal scalability.If the capacity of the data server is less than the upcoming workload, then the number of VMs is increased by the cloud resource broker which is a condition given in our proposed algorithm.

4 Experimental results

The proposed NAERR algorithm is implemented in the simulation environment of CloudSim simulator platform for analyzing its performance. Testing of novel techniques in real world cloud computing environment is not possible realistically since a few simulations compromise the QoS of the end users. Thus, the Cloudsim simulator is used for simulation. The newly proposed algorithm can be utilized by extending or overriding the classes of the CloudSim simulator. Through the simulation some of the parameters like number of task relocation, task execution time, makespan time and average delay task are observed and compared with some existing techniques to prove the performance of the proposed technique.

S.No	Parameter	Qty
1	Data center	1
2	VM RAM Size	2GB
3	VM Manager	Xgen
4	Task length (or) instructions	500000 to 100000000
5	PE processing capacity	174/247/355 MIPS

A few VM and task parameters and their quantities are tabulated in Table 1.

Fig 1 compares the number of task relocation against number of virtual machines and this comparison is done

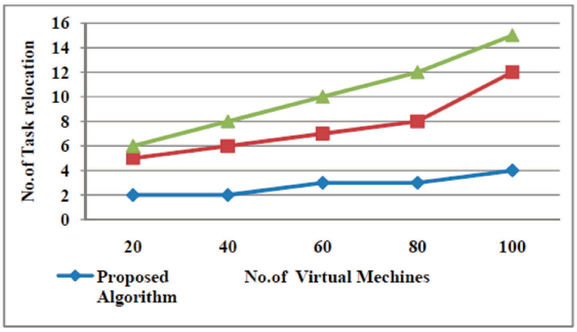

Fig. 1: No.of virtual mechines vs no.of task relocation

for existing LBWRR and LBA algorithms and the proposed NAERR algorithm. In X-axis, the number of virtual machines is plotted and the number of tasks is plotted in Y-axis. In the graph, the green, red and blue lines show the performance of the LBA, LBWRR and proposed NAERR algorithms. By comparing the performance of the all three techniques, the proposed technique provides better performance in terms of minimum or reduced number of task relocations because the NAERR algorithm discovers the most suitable virtual machine for each task. Thus, the load balancer has not discovered the additional optimization to finish the execution of tasks with shortest time period. But the LBWRR and LBA algorithms have not considered the length of the jobs. Rather it only considers the list of arrived jobs and the capability of resources. Therefore, the load balancer has discovered the additional optimization during run time and then it transfers the tasks from overloaded VMs to lower loaded VMs. This consequence has produced the increased number of task relocation in LBWRR and LBA techniques.

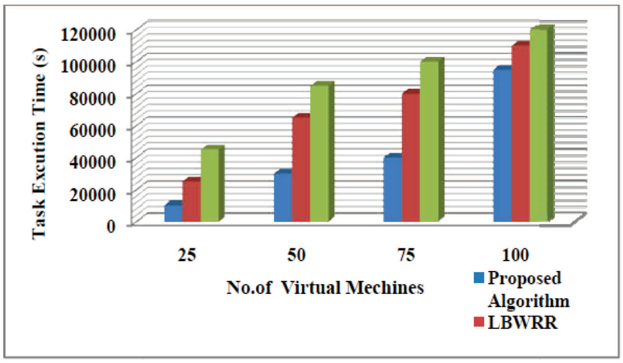

Fig. 2: No.of virtual vechines vs task excution time (s)

From the above Fig-2, it is observed that the proposed novel and adaptive enhanced round robin algorithm by task length has provided a faster execution time compared to other existing LBWRR and LBA load balancing algorithms with heterogeneous tasks and heterogeneous resources. In order to allocate tasks, the proposed NAERR algorithm has computed the job length and processing ability of the heterogeneous virtual machines. Hence, the lengthy jobs are allocated to the VMs having higher capacity in the heterogeneous environment which helps to execute the tasks in a shorter period of time. The proposed scheduler has considered the workload of its all configured virtual machines and its uncertain execution time of ongoing workload has been discovered. Then, the estimated execution time of arrived tasks are calculated by the scheduler in every configured VM. It includes this computed time with the execution time of existing loads on every VM. From this calculation, the least possible execution time is chosen to implement a specific task in one of the VMs. Then, the task has been allocated to that VM. Thus, the proposed NAERR algorithm is most appropriate for the heterogeneous cloud environment. From the above Fig-3 after task assignment, the status of each VM is discovered, that is, the number of under-loaded and over-loaded VMs is computed. These under-loaded and over-loaded VMs have been discovered by computing the VM capacity and the load at the specific time to the VM. If a VM possesses the load of 80

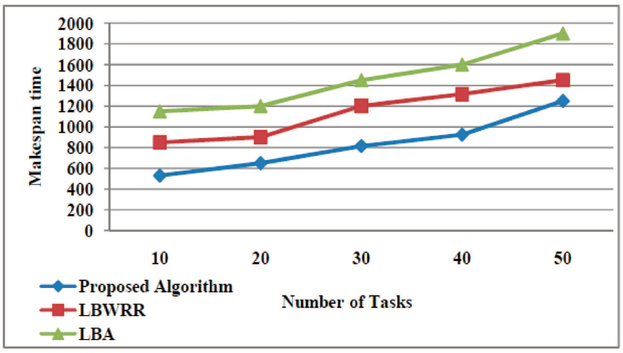

Fig. 3: number of tasks vs makespan time

From Fig 4, the number of delays of the proposed NAERR algorithm is less than the two existing LBWRR and LBA algorithms. This reduction is happens because of the assignment of more number of tasks to the higher-capacity virtual machines. At a time, only one task can execute even if the VM has higher capacity Processing Element (PE). Thus, if other tasks get assigned to that same virtual machine owing to its higher capacity, then those tasks have to be waited until the completion of currently executing task. Therefore, the

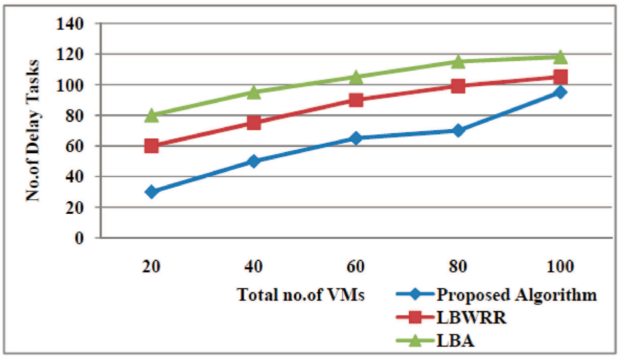

Fig. 4: Total no.of VMs vs No.of Delay Tasks

number of delay tasks is increased. In the existing algorithms, the accurate task execution time is not computed; so, it assigns the tasks to the lower capacity VMs which increase the number of delay tasks.

5 Conclusion

In this paper, the Novel and Adaptively-Enhanced Round Robin (NAERR) algorithm is proposed which allocates tasks based on the capacity of VMs, i.e. it chooses the most suitable VMs for the requesting tasks. In the proposed technique the load balancing is done within three phases. In static and dynamic scheduling, the initial placement of tasks and the parameter computation of VMs are accomplished. At initial task placement, arrived task requests are distributed to all VMs participating in the network and the VM parameter computation involves in computing some of the parameters like VM capacity and the accurate execution time of the currently executing tasks. Then, the VM having higher capacity with less execution time is discovered to which particular tasks are given. The load balancer of the NAERR technique executes at the completion of each task. This makes uniform distribution of workloads throughout all the virtual machines. From the experimental results, the performance of the proposed NAERR algorithm is observed which proved that it is the most suitable technique for heterogeneous resources with heterogeneous tasks compared to the existing LBWRR and LBA algorithms.

References

[1] Z. Xiao, W. Song, and Q. Chen, “Dynamic resource allocation using virtual machines for cloud computing environment,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, No. 6, pp. 1107-1117(2013).

[2] L. D. Dhinesh Babu, L. D. Dhinesh Babu and P.Venkata Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments,” Applied Soft Computing Journal, vol. 13, No.5, pp.2592-2303 (2013).

[3] Kokilavani T, George Amalarethinam DI. Load balanced min-min algorithm for static meta-task scheduling in grid computing. Int J Comput Appl. , vol.20, No 2, pp.43-49 (2011)

[4] Rewinin HE, Lewis TG, Ali HH. Task scheduling in parallel and distributed system Englewood Cliffs. New Jersey (NJ): Prentice Hall;, vol no pp.401-403 (1994).

[5] M, Gajski D. Hypertool, ‘a programming aid for message passing system’. IEEE Trans Parallel DistribSyst. vol. 1, No 3, pp,330-343 (1990)

[6] Hwang JJ, Chow YC, Anger FD. Scheduling precedence graphics in systems with inter-processor communication times. SIAM J Comput.vol 18,No 2, 244-257 (1989)

[7] Rewinin HE, Lewis TG. Scheduling parallel programs onto arbitrary target machines. J Parallel DistribComput.vol. 9 No. 2,pp.138-153 (1990)

[8] W.Kadri, B. Yagoubi, and M. Meddeber, “Efficient dependent tasks assignment algorithm for grid computing environment,” in Proceedings of the 2nd International Symposium onModelling and Implementation of Complex Systems (MISC ’12), Constantine, Algeria, May (2012).

[9] S. Ijaz, E. U.Munir,W. Anwar, andW. Nasir, “Efficient scheduling strategy for task graphs in heterogeneous computing environment,” The International Arab Journal of Information Technology,vol. 10, No. 5, (2013).

[10] Y.Xu,K. Li, L.He, and T. K. Truong, “ADAGscheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization,” Journal of Parallel andDistributed Computing, vol. 73,No. 9, pp. 1306-1322, (2013).

[11] L.-T. Lee, C.-W. Chen, H.-Y. Chang, C.-C. Tang, and K.-C. Pan, “A non-critical path earliest-finish algorithm for interdependent tasks in heterogeneous computing environments,” in Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications (HPCC ’09), pp. 603–608, Seoul, Republic of Korea, June 2009.

[12] B. Xu, C. Zhao, E. Hu, and B. Hu, “Job scheduling algorithm based on Berger model in cloud environment,” Advances in Engineering Software, . vol.42, no. 7, pp.419-425, (2011).

[13] Babu D, Venkata P. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl Soft Comput. vol.13, No.5, pp.2292-2303.(2013)

[14] M, Dubey K, Sharma SC. Job scheduling algorithm in cloud environment considering the priority and cost of job. Proceedings of Sixth International Conference on Soft Computing for Problem Solving; Thapar, India. Springer; (2017).

[15] Moniruzzaman, ABM., Kawser WN, Syed Akther H. An experimental study of load balancing of OpenNebula open-source cloud computing platform. International Conference on Informatics, Electronics & Vision (ICIEV). Dhaka, Bangladesh: IEEE; (2014).

[16] Singh Aarti, Juneja Dimple, Malhotra Manisha. Autonomous agent based load balancing algorithm in cloud computing. Procedia Comput Sci.vol.45 pp.832-841.(2015)

[17] Naha RK, Othman M. Brokering and load-balancing mechanism in the cloud – revisited. IETE Tech Rev.vol.31 No.4, pp.271-276.(2014)

[18] Somasundaram, TS, Govindarajan K, Rajagopalan MR, et al. A broker based architecture for adaptive load balancing and elastic resource provisioning and deprovisioning in multi-tenant based cloud environments. Proceedings of International Conference on Advances in Computing. Chennai, India: Springer; (2013).

J. Arul Sindiya is a Assistant Professor of Computer Science and Engineering at CARE Group of Institutions,Trichy (Anna University), TamilNadu, India.My research interest is in Cloud Computing, I have published Research article in International Journal of Engineering and Technology.

R. PushpaLakshmi is a Professor of Information Technology at PSNA College of Engineering & Technology (Anna University), TamilNadu, India. She received her PhD in Information and Communication Engineering from Anna University, Chennai, India, in 2014. She received the B.E. degree in Computer Science and Engineering from Madurai Kamaraj University, TamilNadu, India, in 2001, and the M.E. degree in Computer Science and Engineering from Anna University, Chennai, India, in 2004. Her main areas of research interest are Wireless Networks, Network Security, Soft Computing and Data Mining. She is a Life Member of the Indian Society for Technical Education (ISTE).