International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Enhanced Resource Allocation and Workload Management using Reinforcement Learning Method for Cloud Environment

P Suresh, P Keerthika, K Logeswaran, R Manjula Devi, M Sangeetha

Abstract*: Cloud computing is a delivery model of IT resources such as computing servers, storage, databases, networking and software over the Internet. It offers the resources as services based on demand with more faster, flexible and economies of scale. The major challenges in the cloud computing are resource allocation and workload management due to the scalability of the cloud users and the services deployed in it. Even though there are various approaches available to manage workload and resource allocation, unfortunately most of them fail to mange it properly. This paper proposes a Reinforcement Learning based Enhanced Resource Allocation and Workload Management (RL-ERAWM) approach to increase the performance of cloud with large number of tasks and users. It implements the Q-Learning approach which effectively considers arrival rate of the requests and workload of the virtual machine. Experimental results prove that the proposed method alleviates the performance of task scheduling and workload management process compared with other approaches in terms of response time, makespan and virtual machine utilization.

Keywords* : Cloud Computing, Task Scheduling, Workload management, Reinforcement Learning, Resource Allocation, Q-Learning.

I. INTRODUCTION

Cloud technology offers scalable IT resources and capacities as a service to multiple external customers over the Internet. It provides mechanisms for storing and accessing data and programs through the Internet instead of user's own computers. Cloud delivers both hardware and software resources as services [1]. With cloud technology, users can access files and applications from any device which can access the Internet. Nowadays Cloud computing is used by other related technologies such as big data computing, block-chain technology and wherever there is a need for processing large amount of data [3].

Cloud computing is the delivery model of computing services like servers, storages and more over the Internet. The companies that offer cloud services are called cloud providers on charge basis based on usage [2]. It is usually classified based on location, or on the service that the cloud is offering. Based on a cloud location, the cloud is classified as Public, Private, Hybrid and Community Cloud. There are three cloud service models namely IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-Service) and SaaS (Software-as-a-Service) [4].

Cloud offers its services to users with on demand access to shared pool of configurable, scalable and measurable computing resources. Since the cloud technology includes multiple web services and dynamicity, automation of mechanism to allocate resources becomes a necessity [5]. The resource allocation mechanism needs to take into account performance history, performance problems, Service Level Agreement (SLA) and resource conservation issues while scaling up or down. Such a mechanism can be implemented using reinforcement learning to allocate more appropriate resources to the service. In cloud, Virtualization technique allows to share a single physical instance of a resource or an application among multiple customers and organizations. It will assign a logical name to a physical resource and providing a pointer to it when demanded. Hence, it provides a feasible solution to improve node utilization. It creates virtual machines (VMs) which provides an environment that is logically separated from the underlying hardware [6].

The resource allocation can be done in two ways such as static allocation and dynamic allocation. Static resource allocation may cause many issues such as service consumer dissatisfaction, more idle time of resources, increasing operating costs and reducing profitability. A dynamic resource allocation method will be more proactive to the demand for a service. It will allocate resources closer to the true demand for the service. Usually in cloud environment, the claim for services change over a period, the resource allocation method will need to adapt the changes and increase or decrease resources allocated to the users [7]. The objectives of this research includes minimize the makespan, response time, dynamic overhead and maximize resource utilization.

Manuscript published on November 30, 2019.
* Correspondence Author
P Suresh*, Information Technology, Kongu Engineering College, Perundurai, Erode, TamilNadu, India. Email: sureshme@gmail.com
P Keerthika, Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, TamilNadu, India. Email: keerthikame@gmail.com
K Logeswaran, Information Technology, Kongu Engineering College, Perundurai, Erode, TamilNadu, India. Email: klogesbtech@gmail.com
R Manjula Devi, Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, TamilNadu, India. Email: rmanjuladevi.gem@gmail.com.
M Sangeetha, Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, TamilNadu, India. Email: sangeethcse@gmail.com.

II. RELATED WORKS

Nowadays the cloud computing area has attracted many researchers, but only a few have focused on machine learning approaches in the cloud.

In [8], an effective resource management approach is proposed using adaptive reinforcement learning which mainly focuses on improving successful execution with low computational complexity. It uses an emerging methodology of Reinforcement Learning (RL) along with neural network which helps the scheduler to observe and adapt to dynamic changes in cloud environment. A novel multi-agent reinforcement learning method is proposed for job scheduling problems for realizing load balancing in computing environments [9]. It uses an ordinal distributed learning strategy to avoid the scalability problem.

The research work presented in [10] provides the insightful view about the task scheduling optimization problem for cloud environment. This work adapts queuing model comprised of three sub models to characterize the service process in cloud computing. It also theoretically analyzes the response time of each sub model and task scheduling method is implemented using reinforcement learning to formulate cloud computation model and minimize the response time under given resources in cloud computing.

In [11], a resource allocation strategy is introduced which performs virtual machine allocation by considering the failure rates, previous history of failure of resources and execution efficiency. It implements fault tolerant based scheduling algorithm to find suitable virtual machine considering the failure rate and also in view of minimized makespan. Manimala R and Suresh P proposed a task scheduling for computing environment by monitoring status of all resources and selecting the one with more computing power to execute the currently submitted task [12]. In [13], Greedy based algorithm in cloud computing which classifies the tasks based on QoS. Then, based on the tasks categories, it will select the appropriate resource.

In [14], an algorithm is proposed based on integer linear programming (ILP) to solve some common cases of the task scheduling problem. This ILP-based algorithm mainly focuses on higher heterogeneity loads in the cloud environment. Husamelddin et al. proposed a task scheduler which adapts the dynamic changes of cloud computing such as resource availability and other attributes to schedule user requests [15]. An adaptive action-selection method is developed that aims to control the action selection dynamically by considering the queue buffer size and uncertainty value function. In [16], reinforcement learning is proposed which adopts with environment conditions and responding to unsteady requests in the cloud environment.

Liu et al. proposed a reinforcement learning (RL) based resource management approach for managing hybrid workloads in distributed computing [17]. It utilizes neural networks to identify desired resource management model and uses ε-greedy methodology to extend exploration along the reinforcement learning process.

III. SYSTEM MODEL

As cloud computing is model of high dynamic resources and more complex platform in nature, a proper queuing model is required to organize the tasks which arrive dynamically. In this approach, the task queue is modeled based on the Poisson distribution model for task execution. Each cloud service provider can be modeled as a type of M/M/c queuing system model. The M/M/c queue is a multiple server queuing model. In Kendall's notation it describes a system where arrivals form a single queue and are governed by a Poisson process, there are c servers and job service times are exponentially distributed. The M/M/c queue is a stochastic process whose state space is the set {0, 1, 2, 3, ...} where the value corresponds to the number of tasks in the task queue including tasks currently in service. Arrivals occur at rate λ according to a Poisson process and move the process from state i to i+1. Service times of tasks have an exponential distribution with parameter μ. If there are less than c jobs, some of the virtual machines will be idle. If there are more than c jobs, the jobs queue in a buffer. Suppose the buffer is of infinite size, so there is no limit on the number of tasks or jobs it can contain. This model can be described as a continuous time Markov chain with transition rate matrix on the state space {0, 1, 2, 3, ...}.

$$Q = \begin{pmatrix} -\lambda & \lambda & & & & & & & & \\ \mu & -(\mu+\lambda) & \lambda & & & & & & & \\ & 2\mu & -(2\mu+\lambda) & \lambda & & & & & & \\ & & 3\mu & -(3\mu+\lambda) & \lambda & & & & & \\ & & & & \ddots & & & & & \\ & & & & c\mu & -(c\mu+\lambda) & \lambda & & & \\ & & & & & c\mu & -(c\mu+\lambda) & \lambda & & \\ & & & & & & c\mu & -(c\mu+\lambda) & \lambda & \\ & & & & & & & & \ddots & \end{pmatrix}$$

Fig. 1 Transition Rate Matrix

The model is a type of birth–death process. It can be represented as $\rho = \lambda/(c\ \mu)$ for the server utilization and require $\rho < 1$ for the queue to be stable. ρ represents the average proportion of time which each of the servers is occupied which means that jobs finding more than one idle virtual machine choose their suitable virtual machine randomly. The state space diagram is given in Fig.2.

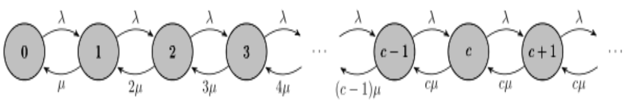

Fig. 2 State Space diagram

The condition which is followed in this proposed work is that tasks (Cloudlets) will not arrive faster than the virtual machines can process them. It is formulated as follows.

$$\sum_{j=1}^{n} \lambda_j < \sum_{k=1}^{m} \mu_k \qquad (1)$$

In equation (1), λ_j is the average arrival rate of tasks to the task queue i.e. scheduler and μ_k is the average completion or execution rate of tasks at the processing element j.

The tasks either experiences an immediate exponential service, or must wait for k tasks to be executed before their own execution, thus experiencing an Erlang distribution with shape parameter k + 1.The task completion time includes data transfer time, waiting time in the task queue and the task execution time at the virtual machine. The mean response time can be calculated as given in Equation 2.

$$R = \frac{1}{(\mu - \lambda)} \tag{2}$$

where μ is the service rate and λ is the arrival rate.

IV. ENHANCED RESOURCE ALLOCATION AND WORKLOAD MANAGEMENT USING REINFORCEMENT LEARNING (RL-ERAWM)

In cloud environment, the scheduling is done by the cloud scheduler. Fig. 3 shows that the cloud scheduling architecture. Clients submit jobs to the cloud scheduler which has all information about the cloud resources/virtual machines (VM). The scheduler selects suitable resource for the task and allocates the resource to the task. The proposed reinforcement based resource allocation and workload management approach is a dynamic approach to allocate resources to the tasks efficiently in the cloud environment. The scheduling model which is followed is based on the Markov Decision Process (MDP) model.

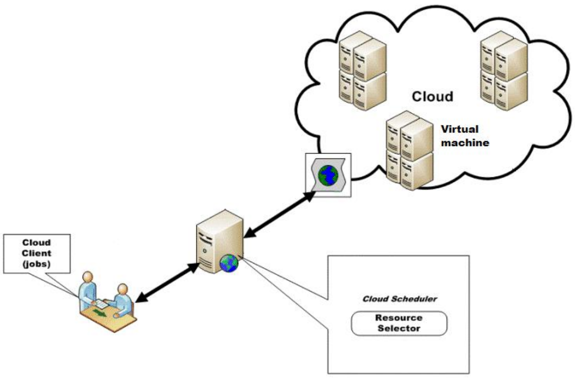

Fig. 3 Cloud Scheduling Architecture

It is a discrete time stochastic control process which provides a mathematical framework for modeling decision making in situations where outcome are partly random and partly under the control of decision maker. This model is mainly useful in solving optimization problems via dynamic programming and reinforcement learning. In the MDP model, state space is represented by S, action space is represented by A and the immediate reward function is represented by $r(s_t, a_t)$.

The Fig. 4 represents a simple MDP with three states, two actions and two rewards. MDP is represented as$(S, A, P_a, R_a)(S, A, P_a, R_a)$ where S is a finite set of states, A A is a finite set of actions,

$$P_a(s, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$$

is the probability that action a in state s at time t will lead to state s' s' at time $t+1$ $t+1$ and $R_a(s, s')$ is the immediate reward received after transitioning from state s s to state $s's'$, due to action a.

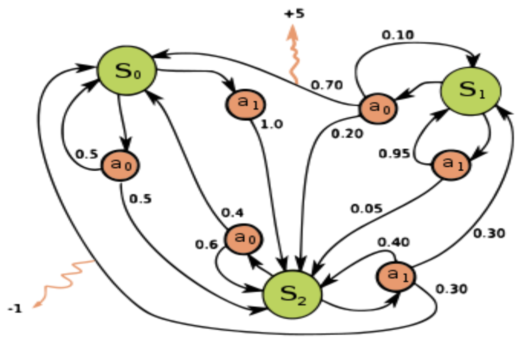

Fig.4 Simple MDP model

The proposed resource allocation approach uses Q-Learning method which is a model free reinforcement learning algorithm. The ultimate goal of this method is to learn a policy which directs an agent what action to take under what conditions. A simple Q-learning structure is depicted in Fig.5. Q-learning identifies a policy which is optimal in the sense of maximizing the expected values of the total reward over successive steps from the current state. It can identify an optimal resource allocation policy for the cloud environment which is represented as Finite Markov Decision Process (FMDP).

Fig.5 Simple Q-Learning Structure

The structure of reinforcement learning is shown in Fig. 6. The reinforcement learning has three components such as state space (S), action space (A) and immediate reward (r(s,a)). State space is represented based on the total number of virtual machines available to execute the tasks in the cloud environment. Assume that totally n virtual machines are available and m tasks are in the task queue. The state space is represented as a vector as $s_i = (s_1, s_2, s_3, \dots \dots \dots, s_n)$, action space is represented as a vector as $a_i = (0,0, \dots 1, \dots 0)_i^k$ which means that i[th] task is allocated to k[th] virtual machine. The immediate reward is used to describe the current running state and the efficiency of resource allocation. The workload of each virtual machine is calculated by using the following formula in Equation 3.

$$L(vm_i) = \frac{\sum_{k=1}^{n} Length_k}{speed_i} \tag{3}$$

where $L(vm_i)$ is the current load at virtual machine i at time t, n is the number of tasks assigned to the virtual machine i, $Length_k$ is the total number of instructions in the task k represented as Million Instructions (MI) and $speed_i$ indicates the speed of the virtual machine i represented as Million Instructions Per Second (MIPS). Finally load will be calculated as below in Equation 4.

$$wl_i = 1 - \frac{1}{L(vm_i)} \quad (4)$$

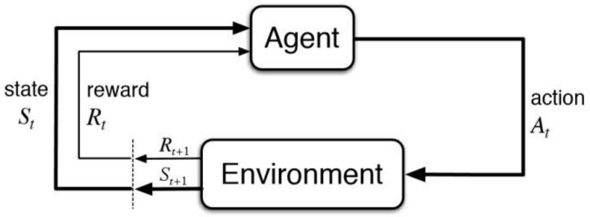

Fig. 6 Basic structure of reinforcement learning

It executes an action $a \in A$ in a specific state which provides the agent with a reward value. The main goal of the agent is to maximize the future reward.

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a)) \quad (5)$$

In Equation (5) s is the current state of the agent, a is the current action picked according to policy, s' is the next state where the agent ends up, a' is the next best action to be picked using current Q-value estimation, r is the current reward observed from the environment in response of current action, γ (>0 and <=1) is the discounting factor for future rewards and α is the step length taken to update the estimation of $Q(s,a)$.

1: Initialize $Q(s,a)$ arbitrarily
2: Repeat (for each episode):
3: Initialize s
4: Repeat (for each step of episode):
5: Choose a from s using policy derived from Q (Eg. ε -greedy)
6: Take action a and observe r, s'
7: Calculate
$Q(s,a) = Q(s,a) + \alpha(r + \gamma\, max_{a'} Q(s',a') - Q(s,a))$
8: $s = s'$
9: End for
10: End for

Fig. 7 Algorithm for Resource allocation and Task scheduling

The proposed algorithm for resource allocation and scheduling is given in Fig.7. The agent receives the reward value r at decision time t given in Equation 6.

$$r = \begin{cases} 1 \ \ if\ wl < \overline{wl} \quad and\ s_i = \max(s_i) \\ 0 \ \ if\ wl < \overline{wl} \\ -1 \ \ otherwise \end{cases} \quad (6)$$

where wl is the workload of the virtual machine and $\overline{wl}$ is the mean workload of all the virtual machines. For a submitted users task, task dispatcher will receive a positive reward 1 if the current task is scheduled to the VM which has the minimum workload among workloads at different virtual machines and the workload is less than the mean workload of virtual machines; 0 if only the condition of workload is less than the mean workload; otherwise task dispatcher will receive a reward value −1.

In this optimization problem, the state space will grow exponentially due to increase of number of tasks and virtual machines dynamically. As a result, the learning process will suffer because of the issue of dimensionality and it leads to have a negative influence on convergence speed and optimized value. To overcome this problem, state aggregation method is employed to accelerate the learning progress.

The abstract function is defined in Equation 7 below.

$$S = 1 - \frac{1}{f}\ where\ f \begin{cases} 0 \quad if\ wl = 0 \\ 4 \quad if\ wl \in \left(0.1, \frac{1}{3}\right) \\ 3 \quad if\ wl \in \left(\frac{1}{3} + 0.1, \frac{2}{3}\right) \\ 2 \quad if\ wl \in \left(\frac{2}{3} + 0.1, 0.9\right) \\ 1 \quad if\ wl = 1 \end{cases} \quad (7)$$

The above formula is used to rank the virtual machine based on the workload at the virtual machine. If a virtual machine is idle or free which means workload is zero at the VM, then it will be provided with higher priority. Suppose if a virtual machine is fully loaded ($wl = 1$), then the virtual machine is not in the allocation scope. The task dispatcher schedules the task to the VM with high priority. In this approach, the learning algorithm evaluates the state based on the resource workload function, f. The given state then provides its reward to the agent. In the lack of prior knowledge about resource states, rewards are necessary for the agent to collect useful information to evaluate the action taken. The action refers to resource allocation and assigning tasks to the most suitable resource. The resource allocation policy is updated based on the given reward value which considers each action corresponds to one state. The agent saves state-action-reward and updates the learning policy and global utility value for better future decision making.

V. RESULTS AND PERFORMANCE EVALUATION

The performance of the proposed approach is evaluated using Cloudsim framework which is a generalized and extensible framework that enables seamless modeling and simulation of resource allocation and task scheduling. Simulation of cloud environments and applications to evaluate performance can provide useful insights to explore such dynamic, massively distributed, and scalable environments [19]. Specifically in cloud environment, where access to the infrastructure incurs payments in real currency, simulation-based approaches offer significant benefits, as it allows the researcher to test their services in repeatable and controllable environment free of cost, and to tune the performance bottlenecks before deploying on real clouds.

The performance of the proposed approach is evaluated based on two experimental setups. The proposed approach is compared and analyzed with two algorithms such as greedy algorithm and random scheduling algorithm. For the proposed approach, the reinforcement learning algorithm parameters values are assumed as $\gamma = 0.5$, $\alpha = 0.5, \varepsilon = 0.01\ and\ \lambda = 10$ for the Poisson distribution.

Table 1: Experimental Setup 1

Parameter	Value
Number of VMs	10
Arrival rate of the requests	5-20
VM memory capacity	512 -2048
Number of tasks	100-1000
Task Length (MI)	1000-5000
VM Speed (MIPS)	500-1000
VM bandwidth (bps)	5000-10000

Table 1 refers the experimental setup considered for the simulation. The performance of the proposed algorithm is evaluated using the parameters such as response time, makespan and VM utilization. Response time is calculates using the following Equation 8.

$$RT_i = CT_i - AT_i + delay_i \tag{8}$$

Makespan is defined as the completion time of the last job to leave the virtual machine [18]. It is formulated IN Equation 9.

$$\text{Makespan} = \max(\text{CT}_\text{j}) \tag{9}$$

where $j = 1, 2, 3 \dots \dots n$. Workload balance among the virtual machines can be evaluated based on the VM utilization. Suppose VM utilization is more, it shows that the workload is properly balanced among the available virtual machines. The utilization of the virtual machines is calculated using following formula.

$$VMU_j = CT_j / Makespan \tag{10}$$

The Average VM utilization is calculated as follows

$$\text{AU} = \frac{\sum_{\text{j}=1}^{\text{n}} \text{VMU}_\text{j}}{\text{n}} \times 100 \tag{11}$$

where

VMU_j – Utilization of the virtual machine j

CT_j – Completion time of the last task at virtual machine j

AU – Average VM utilization

n - Total number of virtual machines.

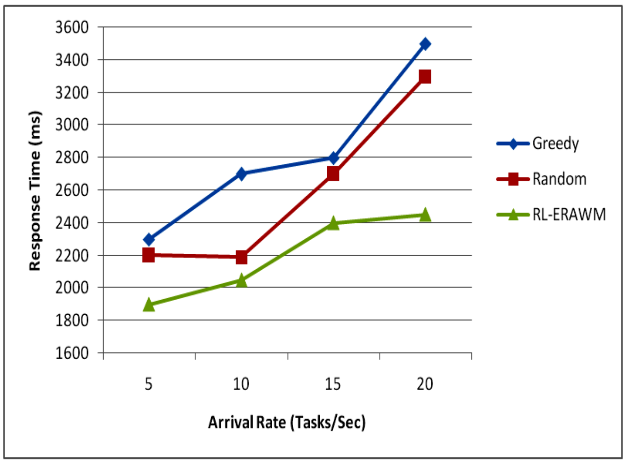

Fig.8 Performance evaluation based on Response Time

The Fig.8 shows that the performance analysis based on the response time and arrival rate of the users' tasks. The proposed RL-ERAWM algorithm shows minimal response time while comparing with Greedy algorithm and Random scheduling approach.

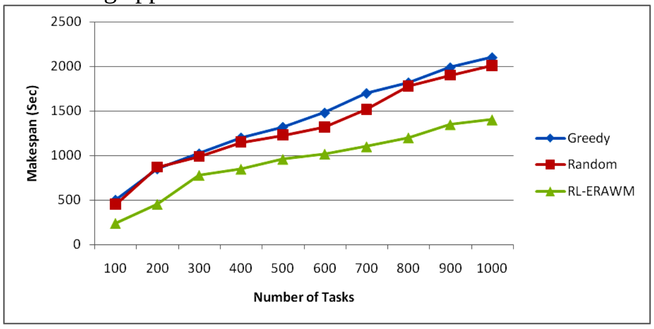

Fig. 9 Performance evaluation based on Makespan

Fig. 9 shows the performance analysis based on makespan. It shows that the proposed algorithm has minimized makespan when comparing with other algorithms.

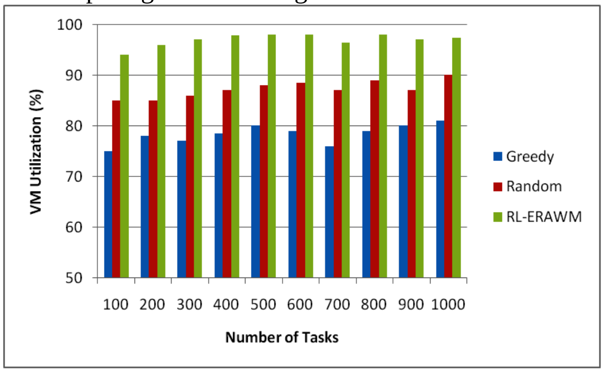

Fig. 10 Performance evaluation based on VM utilization

Fig. 10 shows that the performance based on the utilization of the virtual machines. The proposed RL-ERAWM algorithm has an improved VM utilization when compared to other benchmark algorithms.

VI. CONCLUSION AND FUTURE WORK

This research paper proposes an optimized resource allocation and workload management approach. It implements RL based resource allocation approach which effectively considers the workload of the virtual machines and allocates the tasks efficiently to the most appropriate virtual machines. The Q-Learning algorithm is adopted to maximize the performance the resource allocation approach by reducing dynamic overhead. The Proposed RL-ERAWM method is evaluated based on response time, makespan, resource utilization and workload balance and observed that it shows better performance. In future, multi-agent based reinforcement learning approach can be used to improve the performance of resource allocation. Also the other cloud considerations such as customer deadline, budget, virtual machine failures, and virtual machine migration will be other dimensions of extension.

REFERENCES

1. Buyya, R., Yeo, C., Venugopal, S., "Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities", Proceeding of the 10th IEEE International Conference on High Performance Computing and Communications (HPCC2008), pp. 5–13, 2008.
2. Delimitrou, C., Kozyrakis, C., "QoS-aware scheduling in heterogeneous datacenters with paragon", Journal ACM Transactions on Computer Systems, vol. 31, issue 4, December 2013 Article No. 12.
3. Keshk, A. E., El-Sisi, A. B., & Tawfeek, M. A., "Cloud task scheduling for load balancing based on intelligent strategy" , International Journal of Intelligent Systems and Applications, 6(5), 25-36, 2014.
4. Ankita Jain, Arun Kumar Yadav, Brijesh Kumar Chaurasia, "Proactive Approach for Resource Provisioning in Cloud Computing", International Journal of Recent Technology and Engineering, vol. 7, issue-5S3, February 2019.
5. Wei-Jen Wang, Yue-Shan Chang, Win-Tsung Lo, Yi-Kang Lee, "Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments", The Journal of Supercomputing , vol. 66, issue 2, pp. 783–811, 2013.
6. X. Liu, W. Tong, X. Zhi, F. ZhiRen, and L. WenZhao, "Performance analysis of cloud computing services considering resources sharing among virtual machines", The Journal of Supercomputing., vol. 69, no. 1, pp. 357– 374, 2014.
7. V.Vinothina, Dr.R.Sridaran and Dr.PadmavathiGanapathi, "A Survey on Resource Allocation Strategies in Cloud Computing", International Journal of Advanced Computer Science and Applications, Vol.3, No.6, June 2012.
8. M. Hussin, N. A. W. A. Hamid, and K. A. Kasmiran, "Improving reliability in resource management through adaptive reinforcement learning for distributed systems", Journal of Parallel and Distributed Computing., vol.75, pp. 93–100, 2015.
9. J. Wu, X. Xu, P. Zhang, and C. Liu, "A novel multi-agent reinforcement learning approach for job scheduling in Grid computing", Future Generation Computer Systems, vol. 27, no. 5, pp. 430–439, 2011.
10. Zhiping Peng, Delong Cui, Jinglong Zuo, Qirui Li, "Random task scheduling scheme based on reinforcement learning in cloud computing" , Journal of Cluster Computing, vol.18, issue 4, pp. 1595–1607, 2015.
11. Keerthika P, Suresh P, Manjula Devi R, Sangeetha M, and Sagana C, "Design of a Fault Tolerant Strategy for Resource Scheduling in Cloud Environment", International Journal of Engineering and Advanced Technology, vol-9, issue-1, October 2019.
12. R.Manimala, P. Suresh, "Load balanced job scheduling approach for grid environment", Information Communication and Embedded Systems (ICICES) 2013 International Conference on. IEEE, pp. 336-339, 2013.
13. Li, Ji & Feng, Longhua & Fang, Shenglong, "An Greedy-Based Job Scheduling Algorithm in Cloud Computing", Journal of Software, vol. 9, issue-10, pp. 921-925, 2014.
14. Mostafa Rezvani , Mohammad Kazem Akbari and Bahman Javadi, "Resource Allocation in Cloud Computing Environments Based on Integer Linear Programming", The Computer Journal , Vol. 58, issue. 2, pp. 300-314, 2015.
15. Husamelddin A.M.N. Balla, Guang Sheng Chen and Weipeng Jing, "Reliability Enhancement in Cloud Computing Via Optimized Job Scheduling Implementing Reinforcement Learning Algorithm and Queuing Theory", Proceeding of the 1st International Conference on Data Intelligence and Security, pp. 127-130. DOI 10.1109/ICDIS.2018.00027, 2018.
16. Seyedakbar Mostafavi, Fatemeh Ahmadi, Mehdi Agha Sarram, "Reinforcement Learning based Foresighted Task Scheduling in C loud Computing", arXiv:1810.04718v1, 2018.
17. Liu, Z, Zhang, H, Rao, B, and Wang, L., "A reinforcement learning based resource management approach for time-critical workloads in distributed computing environment", In Proc. Big Data, pp. 252–261. IEEE, 2018.
18. P. Keerthika, P. Suresh, "A multiconstrained grid scheduling algorithm with load balancing and fault tolerance", The Scientific World Journal, vol. 2015, 2015.
19. R. N. Calheiros, R. Ranjan, C. A. De Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services", ArXiv Prepr. ArXiv09032525, 2009.

AUTHORS PROFILE

Dr.P.Suresh is currently working as Senior Assistant Professor in the Department of Information Technology, Kongu Engineering College, Perundurai, Erode. He has a work experience of 13 years in teaching. He has completed his Ph.D in Anna University, Chennai. He is contributing towards research in the area of Grid and Cloud Computing for the past 9 years. He has published 24 research papers in reputed journals indexed by Scopus and SCI and presented 25 papers in National and International Conferences. His areas of interest are Networks, Cloud Computing and Big data. Currently he is working in the areas of Machine Learning and IoT. He is one of the recognized Supervisors in the Faculty of Information and Communication Engineering under Anna University, Chennai. He is currently guiding Ph.D scholars in the areas of Machine Learning and Cloud. He has received grants from various funding agencies like DRDO, ICMR, and CSIR towards the conduct of seminars and workshops. He has received Best Faculty Award from Kongu Engineering College. He is currently acting as Reviewer and Editorial Board member in reputed journals.

Dr.P.Keerthika is currently working as Senior Assistant Professor in the Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode. She has a work experience of 13 years in teaching. She has completed her Ph.D in Anna University, Chennai. She is contributing towards research in the area of Grid and Cloud Computing for the past 9 years. She has published around 20 research papers in reputed journals indexed by Scopus and SCI and presented 15 papers in National and International Conferences. Her areas of interest are Machine Learning, Cloud Computing and IoT. She is one of the recognized Supervisors in the Faculty of Information and Communication Engineering under Anna University, Chennai. She is currently guiding Ph.D scholars in the areas of Machine Learning and Cloud. She has received grants from various funding agencies like DRDO and DST. She is currently acting as Reviewer and Editorial Board member in reputed journals.

K. Logeswaran received the M.E.degree in computer science and engineering from Anna university, Chennai, India, in 2011 and B.Tech degree in information technology from Anna university, Chennai, India in 2009. He is presently pursuing his Ph.D. in Information and Communication Engineering under Anna University, Chennai. He is currently working as Assistant Professor in the department of Information Technology atKongu Engineering College, Perundurai, India. His research interests are in theareas of Data mining, Knowledge discovery and Networking. He has authored over 15 research papers in refereedjournals and international conferences. He has received grants from various funding agencies like DRDO, SERB, and CSIR towards the conduct of seminars and workshops.

Dr. R. Manjula Devi is working as a senior Assistant Professor in the Department of Computer Science and Engineering at Kongu Engineering College, Perundurai. . She has a work experience of 13 years in teaching. She completed her PhD (Information and Communication Engineering) from Anna University, Chennai (2015). Her research interests are Machine Learning, Soft Computing, Computer Graphics, and Artificial Intelligence. She has published over 25 technical papers on Neural Network, Intrusion Detection System, Data Mining and Cloud Computing. She is currently acting as Reviewer and Editorial Board member in reputed journals. She is authored more than 15 books edited by Charulatha Publications, Chennai. She has been honored with the various award such as Best Faculty Award, Shri P. K. Das Memorial Best Faculty Award & Life Time Achievement Award–2015, Best Author Award, GRABS Best Young Teacher Award, Young Woman Educator & Scholar Award, etc.,

Ms.M.Sangeetha is currently working as Assistant Professor in the Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode. She has a work experience of 9 years in teaching. She has pursuring her Ph.D in Anna University, Chennai. She is contributing towards research in the area of Machine Learning for the past 9 years. She has published around 3 research papers in reputed journals indexed by Scopus and presented 7 papers in National and International Conferences. Her areas of interest are Machine Learning and Graph theory.