Resource Management and Scheduling in Cloud Environment

Vignesh V, Sendhil Kumar KS, Jaisankar N

School of Computing Science and Engineering, VIT University Vellore, Tamil, Nadu, India – 632 014

Abstract- In Cloud Environment, the process of execution requires Resource Management due to the high process to the resource ratio. Resource Scheduling is a complicated task in cloud computing environment because there are many alternative computers with varying capacities. The goal of this paper is to propose a model for job-oriented resource scheduling in a cloud computing environment. Resource allocation task is scheduled for the Process which gives the available resources and user preferences. The computing resources can be allocated according to the rank of job .This paper constructs the analysis of resource scheduling algorithms. The time parameters of three algorithms, viz. Round Robin, Pre-emptive Priority and Shortest Remaining Time First have been taken into consideration. From this, it has been computed that SRTF has the lowest time parameters in all respects and is the most efficient algorithm for resource scheduling.

Index Terms- Resource management, Cloud Computing Environment, Resource Scheduling, Round Robin, Preemptive Priority, Shortest Remaining Time First.

I. Introduction

A Cloud is a type of parallel and distributed system which consists of a collection of interconnected and virtualized computers. These computers are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements, which are established through negotiation between the service provider and consumers. The computing resources can be allocated dynamically upon the requirements and preferences of user. The consumers may access applications and data of the Cloud from anywhere at any time, it is difficult for the cloud service providers to allocate the cloud resources dynamically and efficiently [1]. Physical resource are Computer, Processor, disk, database, network, Bandwidth, scientific instruments and the logical resources are Execution, monitoring, communicate application and etc.

Dynamic allocation of tasks to computers is complicated in the cloud computing environment due to the complicated process of assigning multiple copies of the same task to different computers. Likewise, the resource allocation is also a big challenge in cloud computing [2]. Cloud computing not only enables users to migrate their data and computation to a remote location with minimal impact on system performance, but also ensures easy access to a cloud computing environment to visit their data and obtain the computation at anytime and anywhere. Cloud computing is attempting to provide cheap and easy access to measurable and billable computational resources when compared with other paradigms such as Distribute Computing, Grid Computing, etc. In a cloud computing environment, the tasks are distributed across distinct computational nodes. In order to allocate cloud computing resources, nodes with spare computing power are detected and network bandwidth, line quality, response time, task costs, and reliability of resource allocation are analyzed [6]. Hence, the quality of cloud computing service can be described by resources such as network bandwidth, complete time, task costs, and reliability, etc. [7].

There have been various types of scheduling algorithm that exists in distributed computing system. Most of them can be applied in the cloud environment with suitable verifications. The main advantage of job scheduling algorithm is to achieve a high performance computing and excellent system throughput. Traditional job scheduling algorithms are not able to provide scheduling in the cloud environments. According to a simple classification, job scheduling algorithms in cloud computing can be categorized into two main groups are Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms. In BMHA, Jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time.

The main examples of BMHA based algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Min–Min algorithm and Max–Min algorithm. By On-line mode heuristic scheduling algorithm, Jobs are scheduled when they arrive in the system. Since the cloud environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for a cloud environment [15].

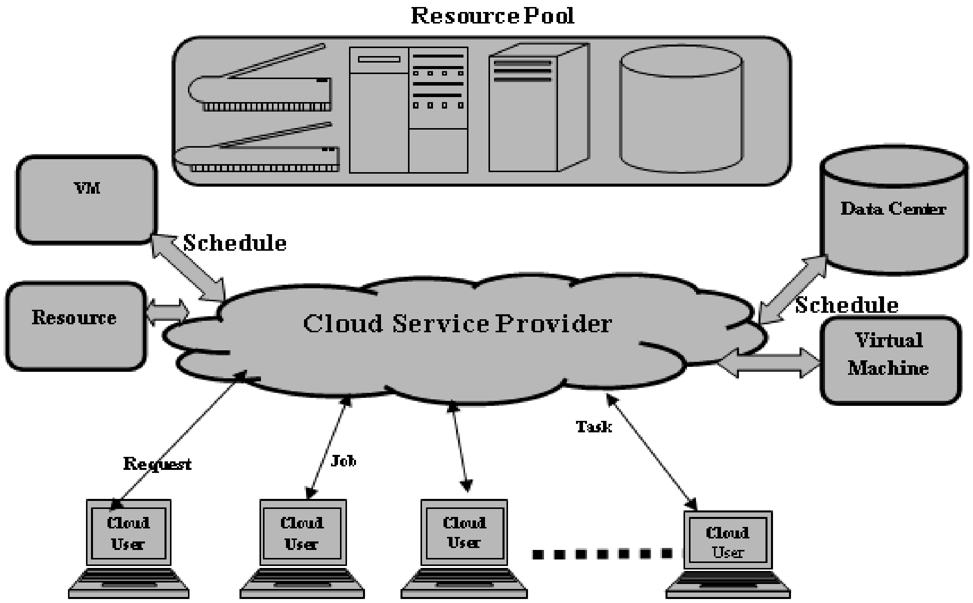

Figure1: Basic Resource Management & Scheduling in Cloud Environment

II. RELATED WORKS

Mayank Mishra et al. [2] in his paper has told that, the users of cloud services pay only for the amount of resources (a pay-as-use model) used by them. This model is quite different from earlier infrastructure models, where enterprises would invest huge amounts of money in building their own computing infrastructure. Typically, traditional data centers are provisioned to meet the peak demand, which results in wastage of resources during non-peak periods. To alleviate the above problem, modern-day data centers are shifting to the cloud. The important characteristics of cloud-based data centers are making resources available on demand. The operation and maintenance of the data center lies with the cloud provider. Thus, the cloud model enables the users to have a computing environment without investing a huge amount of money to build a computing infrastructure. This provides ability to dynamically scale or shrink the provisioned resources as per the dynamic requirements. Fine-grained metering. This enables the pay as-use model, that is, users pay only for the services used and hence do not need to be locked into long-term commitments. As a result, a cloud-based solution is an attractive provisioning alternative to exploit the computing- as-service model.

Anton Beloglazov and Rajkumar Buyya [5] have proposed the plan for the future research work that consists of several steps presented in Table. Once the algorithms for all of the proposed optimization stages are developed, they will be combined in an overall solution and implemented as a part of a real-world Cloud platform, such as Aneka. In this contemporary world, Internet has been a predominant, which has presented a great opportunity for providing real-time services over the Internet.

Venkatesa Kumar. V and S. Palaniswami [6], in their paper, have proposed the overall resource utilization and, consequently, reduce the processing cost. Our experimental results clearly show that our proposed preemptive scheduling algorithm is effective in this regard. In this study, we present a novel Turnaround time utility scheduling approach which focuses on both the high priority and the low priority takes that arrive for scheduling. Vijindra and Sudhir shenai. A [8] in their paper, have presented an algorithm for a cloud computing environment that could automatically allocate resources based on energy optimization methods. Then, we prove the effectiveness of our algorithm. In the experiments and results analysis, we find that in a practical Cloud Computing Environment, using one whole Cloud node to calculate a single task or job will waste a lot of energy, even when the structure of cloud framework naturally support paralleled process. We need to deploy an automatic process to find the appropriate CPU frequency, main memory's mode or disk's mode or speed. We have also deployed scalable distributed monitoring software for the cloud clusters.

Liang Luo et al.[10] in their paper, have discussed about, a new VM Load Balancing Algorithm is proposed and then implemented in Cloud Computing environment using CloudSim toolkit, in java language. In this algorithm, the VM assigns a varying (different) amount of the available processing power to the individual application services. These VMs of different processing powers, the tasks/requests (application services) are assigned or allocated to the most powerful VM and then to the lowest and so on. we have optimized the given performance parameters such as response time and data processing time, giving an efficient VM Load Balancing algorithm i.e. Weighted Active Load Balancing Algorithm in the Cloud Computing environment.

Qiang Li and Yike Guo [11] have proposed a model for optimization of SLA-based resource schedule in cloud computing based on stochastic integer programming technique. The performance evaluation has been performed by numerical studies and simulation. The experimental result shows that the optimal solution is obtained in a reasonable\y short time. Xin Lu, Zilong GU [15], in their paper have discussed that, by monitoring performance parameters of virtual machines in real time, the overloaded is easily detected once these parameters exceeded the threshold. Quickly finding the nearest idle node by the ant colony algorithm from the resources and starting the virtual machine can bears part of the load and meets these performance and resource requirements of the load. This realizes the load adaptive dynamic resource scheduling in the cloud services platform and achieves the goal of load balancing.

Zhongni Zheng, Rui Wang [16] did the research of using GA to deal with scheduling problem in the cloud, we propose PGA to achieve the optimization or sub-optimization for cloud scheduling problems. Mathematically, we consider the scheduling problem as an Unbalanced Assignment Problem. Future work will include a more complete characterization of the constraints for scheduling in a cloud computing environment, improvements for the convergence with more complex problems. Lu Huang, Hai-shan Chen [17] also presented system architecture for users to make resource requests in a cost-effective manner, and discussed a scheduling scheme that provides good performance and fairness simultaneously in a heterogeneous cluster, by adopting progress share as a share metric. By considering various configurations possible in a heterogeneous environment, we could cut the cost of maintaining such a cluster by 28%. In addition, we proposed a scheduling algorithm that provides good performance and fairness simultaneously in a heterogeneous cluster. By adopting progress share as a share metric, we were able to improve the performance of a job that can utilize GPUs by 30% while ensuring fairness among multiple jobs.

III. PROPOSED WORK

In order to efficiently allocate computing resources; scheduling becomes a very complicated task in a cloud computing environment where many alternative computers with varying capacities are available. Efficient task scheduling mechanism can meet users' requirements and improve the resource utilization. [4]The cloud service providers often receive lots of computing requests with different requirements and preferences from users simultaneously. Some tasks need to be fulfilled at a lower cost and less computing resources, while some tasks require higher computing ability and take more bandwidth and computing resources. When the cloud computing service providers receive the tasks from users, the tasks can be pair wise compared using the comparison matrix technique. The cloud computing providers negotiate with the users on the requirements of tasks including network bandwidth, complete time, task costs, and reliability of task. [5] The computing resource or storage resource in a cloud computing environment can be assigned to the corresponding task according to the weight of each task once

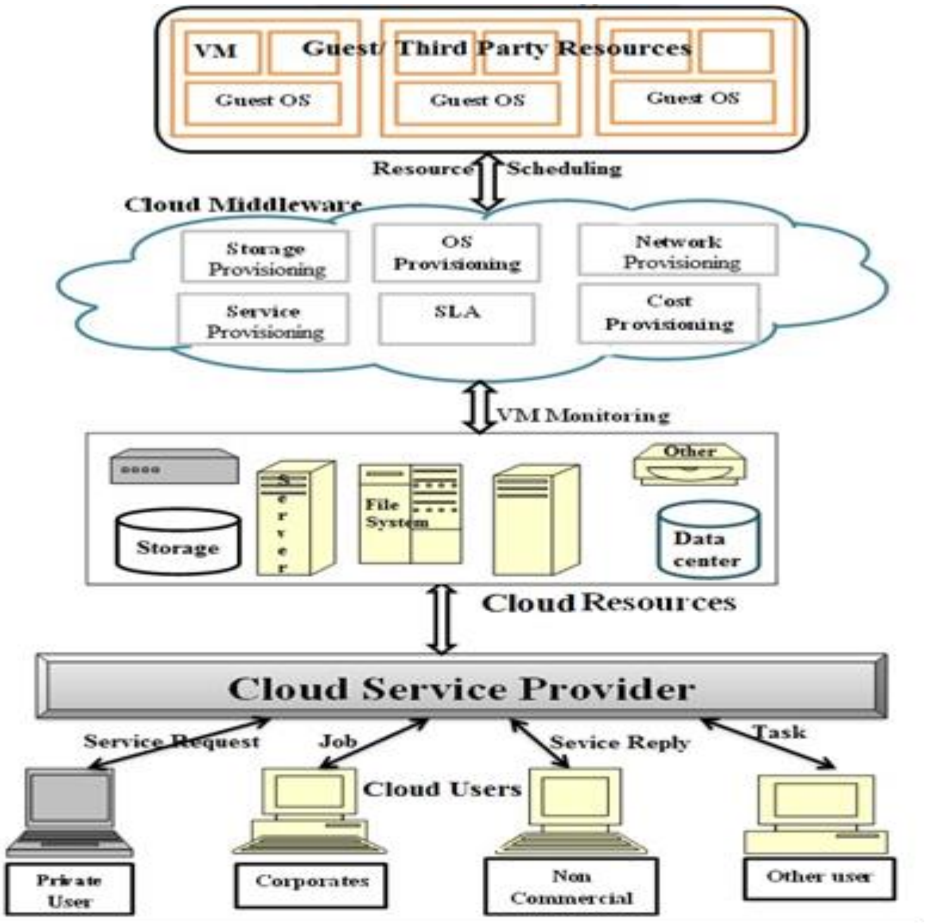

Figure 2: Resource Scheduling in Cloud Computing Environment

III. 1. Round robin

It is the simplest algorithm that uses the concept of time quantum or slices. Here the time is divided into multiple slices and each node is given a particular time quantum or time interval and in this quantum, the node will perform its operations. The resources of the service provider are provided to the client on the basis of this time quantum. In Round Robin Scheduling the time quantum play a very important role for scheduling, because if time quantum is very large then Round Robin Scheduling Algorithm is same as the FCFS Scheduling. If the time quantum is extremely too small then Round Robin Scheduling is called as Processor Sharing Algorithm and number of context switches are very high. It selects the load on random basis and leads to the situation where some nodes are heavily loaded and some are lightly loaded.

Though the algorithm is very simple, there is an additional load on the scheduler to decide the size of quantum and it has longer average waiting time, higher context switches higher turnaround time and low throughput. The Round Robin algorithm mainly focuses on distributing the load equally to all the nodes. Using this algorithm, the scheduler allocates one VM to a node in a cyclic manner. The round robin scheduling in the cloud is very similar to the round robin scheduling used in the process scheduling. The scheduler starts with a node and moves on to the next node, after a VM is assigned to that node. This is repeated until all the nodes have been allocated at least one VM and then the scheduler returns to the first node again. Hence, in this case, the scheduler does not wait for the exhaustion of the resources of a node before moving on to the next. As an example, if there are three nodes and three VMs are to be scheduled, each node would be allocated one VM, provided all the nodes have enough available resources to run the VMs.

The main advantage of this algorithm is that it utilizes all the resources in a balanced order. An equal number of VMs are allocated to all the nodes which ensure fairness. However, the major drawback of using this algorithm is that the power consumption will be high as many nodes will be kept turned-on for a long time. If three resources can be run on a single node, all the three nodes will be turned on when Round Robin is used which will consume a significant amount of power.

III. 2. Preemptive Priority

Priority of jobs is an important issue in scheduling because some jobs should be serviced earlier than other those jobs can't stay for a long time in a system. A suitable job scheduling algorithm must be considered priority of jobs. To address this problem some researchers have considered priority of jobs scheduling algorithm. Those researches have focused on a few criteria of jobs in scheduling. In cloud environments we always face a wide variety of attributes that should be considered. Priority is an important issue of job scheduling in cloud environments. In this paper we have proposed a priority based job scheduling algorithm which can be applied in cloud environments. Also we have provided a discussion about some issues related to the proposed algorithm such as complexity, consistency and finish time. Result of this paper indicates that the proposed algorithm has reasonable complexity. In addition, improving the proposed algorithm in order to gain less finish time is considered as future work. As a scheduling policy, preemption has wide applications in many areas (e.g. Process scheduling, bandwidth allocation, manufacturing scheduling).

Most basically, preemption can be seen as a process that removes (un-schedules, suspends or aborts) one or more previously scheduled activities according to certain criteria and re-allocates freed resource capacity to a new activity. A preemption policy is normally used for scheduling high priority activities when a capacity shortage appears. Preemption has been investigated fairly extensively relative to scheduling single-capacity resources. CPU scheduling, which is central to operating system design, is preventative example, The CPU is single-capacity resource, which can be time-shared to accommodate multiple tasks by algorithms (such as round robin) that repeatedly allocate time slices to competing tasks. Here, a preemptive scheduling policy provides a means for reallocating time slices as new, more important jobs arrive for processing. Preemptive scheduling is much more complex in the context of cumulative or multi-capacity resources, and this problem has received much less attention in the literature. The principal complication concerns the selection of which activity (or activities) to preempt. In the case of multi-capacity resources, the number of candidate sets of activities increases exponentially with resource capacity size, while only a single activity must be identified in the single-capacity case.

III. 3 Shortest Response Time First

The basic idea is straightforward: each process is assigned a priority, and priority is allowed to run. Equal-Priority processes are scheduled in FCFS order. The shortest-Job-First (SJF) algorithm is a special case of general priority scheduling algorithm. An SJF algorithm is simply a priority algorithm where the priority is the inverse of the next CPU burst. That is, the longer the CPU burst, the lower the priority and vice versa. Priority can be defined either internally or externally. Internally defined priorities use some measurable quantities or qualities to compute priority of a process. Instead, we have to check the event clock at each cycle, to see whether pre-emption is taking place, and if so, update the process data accordingly. The main thrust of the changes is to refine the process data to include a variable to count how much time to the next wait, and to the event module to explicitly cycle the clock, and see if an event is due at the current time. SJF policy selects the job with the shortest (expected) processing time first. Shorter jobs areal ways executed before long jobs. One major difficulty with SJF is the need to know or estimate the processing time of each job (can only predict the future!)Also, long running jobs may starve; because the CPU has a steady supply of short jobs.

IV. RESULTS AND ANALYSIS

Based on the observation from the following five request and based on their submission time the priority has been assigned .The priority has been assigned based on the submission time or type of job request .The execution time has been determined by the cloud provider. Based on the five job request undergone by the above algorithms we attain the above values that show in Table1. For round robin the time quantum is assigned as 5.

Request Id	Request Submission time	Request Priority	Execution time
#R1	5	3	8
#R2	1	2	3
#R3	3	1	5
#R4	9	5	10
#R5	8	4	9

Table1: Job Request Information

AVG TT- Average Turnaround Time
AVG WT- Average Waiting Time

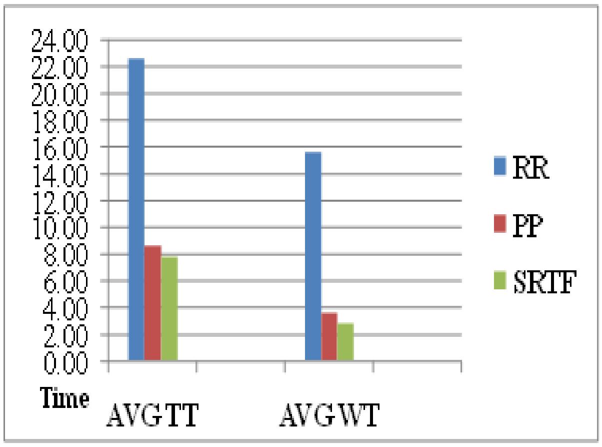

Figure3: Average Waiting & Turnaround Time

Total TT- Average Turnaround Time
Total WT- Average Waiting Time

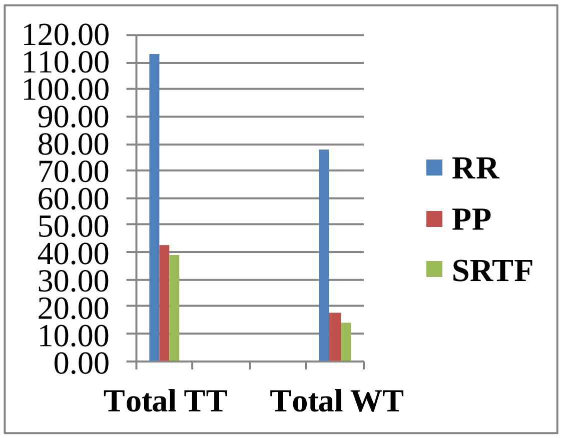

Figure4: Total Waiting & Turnaround Time

Based on the above observation that have been put under the above scheduling algorithm we attain the average waiting and average turnaround time as show in graph 1.Eventually graph2 which show the total turnaround time and waiting time . After all the performance analysis we have been through shortest remaining time first has efficient performance compare with the round robin and preemptive priority.

The Average and Total times have been calculated and presented for the three resource scheduling algorithms in Table 2. From the analysis, we can conclude that, Shortest Remaining Time First algorithm shows excellent performance and computes the resource scheduling with relatively less waiting time and Turnaround time.

Job Execution Description	Round robin	Preemptive Priority	Shortest Remaining Time First
Average Waiting time	15.6	3.6	2.8
Average Turnaround time	22.6	8.6	7.8
Total Turnaround time	113	43	39
Total Waiting time	78	18	14

Table 2: Result of Job waiting and Turnaround time in Execution

V. CONCLUTION

Scheduling is one of the most important tasks in cloud computing environment. In this paper, we have analyzed various scheduling algorithm and tabulated various parameter. We have noticed that disk space management is critical issue in virtual environment. Existing scheduling algorithm gives high throughput and cost effective but they do not consider reliability and availability. So we need algorithm that improves availability and reliability in cloud computing environment. In future enhancement will propose a new algorithm for resource scheduling and comparative with existing algorithms. The efficiency of the user request first may be optimized the processor and execute the request. In future enhancement will propose a new algorithm for resource scheduling and comparative with existing algorithms. The efficiency of the user request first may be optimized the processor and execute the request.

REFERENCES

[1] Hitoshi Matsumoto, Yutaka Ezaki," Dynamic Resource Management in Cloud Environment", July 2011, FUJITSU science & Tech journal, Volume 47, No: 3, page no: 270-276.

[2] Mayank Mishra, Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo, "Dynamic Resource Management Using Virtual Machine Migrations", Sep 2012, 0163-6804/12, IEEE Communications Magazine, page no: 34-40.

[3] Fetahi Wuhib and Rolf Stadler, "Distributed Monitoring and Resource Management for Large Cloud Environments", 2011, 12th IFIP/IEEE 1M 2011: Dissertation Digest, 978-1-4244-9221-31111, IEEE, page no: 970-975.

[4] Ghalem Belalem, Samah Bouamama and Larbi Sekhri, "An Effective Economic Management of Resources in Cloud Computing", March 2011, JOURNAL OF COMPUTERS, Vol. 6, No. 3, page no: 404-411.

[5] Anton Beloglazov and Rajkumar Buyya," Energy Efficient Resource Management in Virtualized Cloud Data Centers", 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 978-0-7695-4039-9/10,IEEE, DOI 10.1109/CCGRID.2010.46, page no: 826-831.

[6] Venkatesa Kumar. V and S. Palaniswami," A Dynamic Resource Allocation Method for Parallel Data Processing in Cloud Computing", 2012, Journal of Computer Science 8 (5), ISSN 1549-3636, Science Publications, page no: 780-788.

[7] Weiwei Lina, James Z. Wangb, Chen Liangc and Deyu Qia, "A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing", 2011, 1877-7058, Elsevier Ltd, PEEA 2011 Doi:10.1016/j.proeng.2011.11.2568, page no: 695 – 703.

[8] Vijindra and Sudhir Shenai. A, "Survey of Scheduling Issues in Cloud Computing", 2012, ICMOC-2012, 1877-7058, Elsevier Ltd, Doi: 10.1016/j.proeng.2012.06.337, page no: 2881 – 2888.

[9] Jasmin James, and Dr. Bhupendra Verma," EFFICIENT VM LOAD BALANCING ALGORITHM FOR A CLOUD COMPUTING ENVIRONMENT ", Sep 2012, IJCSE, ISSN: 0975-3397 Vol. 4, No. 09, page no: 1658 – 1663.

[10] Liang Luo, Wenjun Wu, Dichen Di, Fei Zhang, Yizhou Yan, Yaokuan Mao, "A Resource Scheduling Algorithm of Cloud Computing based on Energy Efficient Optimization Methods", 2012, 978-1-4673-2154-9/12, IEEE.

[11] Qiang Li and Yike Guo, "Optimization of Resource Scheduling in Cloud Computing", 2010, 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 978-0-7695-4324-6/10, IEEE, DOI 10.1109/SYNASC.2010.8, page no: 315 – 320.

[12] Sivadon Chaisiri, Bu-Sung Lee and Dusit Niyato, "Optimization of Resource Provisioning Cost in Cloud Computing", January 31 2011, DRAFT Digital Object Identifier 10.1109/TSC.2011.7 1939-1374/11, IEEE, Pages: 32.

[13] Chandrashekhar S. Pawar and R.B.Wagh, "A review of resource allocation policies in cloud computing", April 21 2012, World Journal of Science and Technology 2012, 2(3):165-167, ISSN: 2231 – 2587, www.worldjournalofscience.com, Page no: 165-167.

[14] Hongbin Liang, Lin X. Cai, Dijiang Huang, Xuemin (Sherman) Shen, and Daiyuan Peng, "An SMDP-Based Service Model for Interdomain Resource Allocation in Mobile Cloud Networks", JUNE 2012, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 61, NO. 5, 0018-9545/IEEE, Page no: 2222-2232.

[15] Xin Lu, Zilong Gu, "A LOAD-ADAPATIVE CLOUD RESOURCE SCHEDULING MODEL BASED ON ANT COLONY ALGORITHM", 2011, 978-1-61284-204-2/11, Proceedings of IEEE CCIS2011, Page no: 296-300.

[16] Zhongni Zheng, Rui Wang, Hai Zhong, Xuejie Zhang, "An Approach for Cloud Resource Scheduling Based on Parallel Genetic Algorithm", 2011, 978-1-61284-840-2/11, IEEE, Page no: 444-447.

[17] Lu Huang, Hai-shan Chen, Ting-ting Hu, "Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing", JOURNAL OF SOFTWARE, VOL. 8, NO. 2, FEBRUARY 2013, ACADEMY PUBLISHER,doi:10.4304/jsw.8.2.480-487, Page no: 480-487.

[18] Gunho Lee, "Resource Allocation and Scheduling in Heterogeneous Cloud Environments", Thesis, Technical Report No. UCB/EECS-2012-78, http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/ EECS-2012-78.html, May 10, 2012, Pages: 113.

AUTHORS

First Author – Vignesh V, School of Computing Science and Engineering, VIT University Vellore, Tamil, Nadu, India – 632 014, Email: vicky.varath@yahoo.com
Second Author – Sendhil Kumar KS, School of Computing Science and Engineering, VIT University Vellore, Tamil, Nadu, India – 632 014, Email: sendhilkumar.ks@vit.ac.in
Third Author – Jaisankar N, School of Computing Science and Engineering, VIT University Vellore, Tamil, Nadu, India – 632 014, Email: njaisankar@vit.ac.in