International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Index Copernicus Value (2015): 62.86 | Impact Factor (2015): 3.791

Scheduling Load over Cloud Computing: A Survey

Pragati Rajput[1], Ruchika Mishra[2]

[1]Computer Science, Adina Institute Science & Technology, Sagar, MP

[2]Assistant Professor, Computer Science, Adina Institute Science & Technology, Sagar, MP

Abstract: ***Here in this paper by analyzing the various issues and challenges of the existing Load Balancing Techniques during the Sharing and Scheduling of Resources over Cloud computing which makes Cloud Data Center's Load Imbalanced. Although these techniques provides an efficient Load Balancing Over Cloud Environment there may analyze some issues in the existing methodologies during Load Balancing such as Computational Overhead, Latency and efficient Utilization of Resources, hence by analyzing and complete survey of all such techniques their various advantages and limitations or issues a new and efficient technique for Load Balancing is implemented in future.***

Keywords: Load Balancing, Cloud Computing, Task Scheduling, Distributed System, Clustering, Virtual Machines

1. Introduction

Load balancing and task scheduling has close contacts with each other in the cloud environment, task scheduling mechanism responsible for the optimal matching of tasks and resources [1]. Because of the pertinency of task scheduling algorithm, load balancing become another important measure in the cloud. Since load balancing state level two loads in task scheduling under cloud computing environment: the first stage is the virtual machine load, the second one is the resource layer load [2].

2. Guidelines of Scheduling

Job Scheduling is used to allocate certain jobs to particular resources in particular time. In cloud computing, job- scheduling problem is a biggest and challenging issue. Hence the job scheduler should be dynamic. Job scheduling in cloud computing is mainly focuses to improve the efficient utilization of resource that is bandwidth, memory and reduction in completion time. An efficient job scheduling strategy must aim to yield less response time so that the execution of submitted jobs takes place within a possible minimum time and there will be an occurrence of in-time where resources are reallocated. Because of this, less rejection of jobs takes place and more number of jobs can be submitted to the cloud by the clients which ultimately show increasing results in accelerating the business performance of the cloud. There are different types of scheduling based on different criteria, such as static vs. Dynamic, centralized vs. Distributed, offline vs. Online etc are defined below:

1) **Static Scheduling:** Pre-Schedule jobs, all information are known about available resources and tasks and a task is assigned once to a resource, so it's easier to adapt based on scheduler's perspective [3].
2) **Dynamic Scheduling:** Jobs are dynamically available for scheduling over time by the scheduler. It is more flexible than static scheduling, to be able of determining run time in advance. It is more critical to include load balance as a main factor to obtain stable, accurate, and efficient scheduler algorithm [3].
3) **Centralized Scheduling:** As mentioned in dynamic scheduling, it's a responsibility of centralized / distributed scheduler to make global decision. The main benefits of centralized scheduling are ease of implementation; efficiency and more control and monitoring on resources. On the other hand; such scheduler lacks scalability, fault tolerance and efficient performance. Because of this disadvantage it's not recommended for large-scale grids [4].
4) **Distributed / Decentralized Scheduling:** This type of scheduling is more realistic for real cloud despite of its weak efficiency compared to centralized scheduling. There is no central control entity, so local schedulers' requests to manage and maintain state of jobs' queue [5].
5) **Preemptive Scheduling:** This type of scheduling allows each job to be interrupted during execution and a job can be migrated to another resource leaving its originally allocated resource, available for other jobs. If constraints such as priority are considered, this type of scheduling is more helpful [6].
6) **Non Preemptive Scheduling:** It is a scheduling process, in which resources are not being allowed to be re-allocated until the running and scheduled job finished its execution [6].
7) **Co-operative scheduling:** In this type of scheduling, system have already many schedulers, each one is responsible for performing certain activity in scheduling process towards common system wide range based on the cooperation of procedures, given rules and current system users [4].
8) **Immediate / Online Mode:** In this type of scheduling, scheduler schedules any recently arriving job as soon as it arrives with no waiting for next time interval on available resources at that moment [7].
9) **Batch / Offline Mode:** The scheduler stores arriving jobs as group of problems to be solved over successive time intervals, so that it is better to map a job for suitable resources depending on its characteristics [7].

3. Task Scheduling Model

Task scheduling model [8] is based on the concept of value. Value function includes deadline, reward, decay, bottom line and penalty. Deadline represents time bottom line of the task to reflect time demand of user expecting to complete the task before this time and when the service provider fails to meet the needs of the user time, provider should pay compensation. Reward represents the cost that users pay for service of service provider and when provider meets the demands of user, provider gets the reward. Decay represents the ratio of compensation paid by provider. Bottom line represents time corresponding to the highest compensation offered by provider and with the time going on, compensation increases gradually, at this moment, compensation get to the maximum value and does not increase any more. Penalty represents the highest compensation offered by provider and with the time going on; compensation increases gradually, at this moment, compensation get to penalty and does not increase any more.

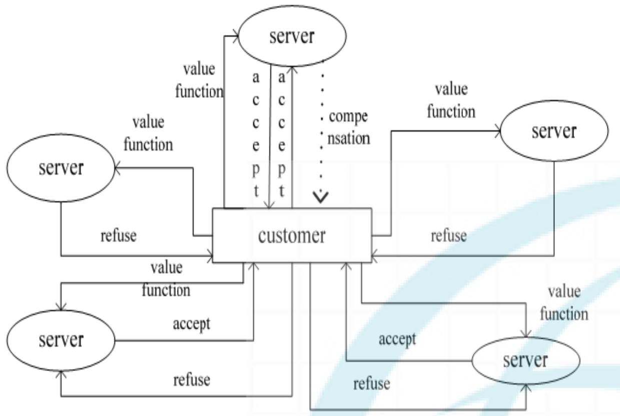

Figure 1: Task scheduling model

The figure 1 shows the basic task scheduling model. Here in this model is given a number of servers which can accept and refuse the values of jobs from the customer. The various jobs to be accessed by the user or customer is queried to the server, where the server checks the validity of the job in the scheduling pool, if the job is allotted accept function calls otherwise refuse.

The task scheduling algorithm consists of some functions so that the task is scheduled by the servers which are request by the user.

1. Accept Function: The accept function is used here between user request to the server. The various request or task given by the user is send to the server where server checks the scheduling of the task and on the basis accept the requests.
2. Refuse Function: This function gets call when a user is requested for some task but server refuses due to time or already allotted resources. The server once refuses the request of the user will be requested again by the user.

Value Function: This function is used to send the value to the server by the user once the request is accepted.

4. Literature Survey

Jio Zhao et. al's implemented a heuristic Clustering based Load Balancing in Cloud using Bayes Theorem. Bhupendra Panchal Et. al's uses a K-means Clustering approach for the Scheduling of Resources such as Virtual machines in cloud computing environment [9, 15]. In this paper a new dynamic VM allocation policy is introduced that takes VM`s as per user requirement and allocate them in cluster form to the available data centers. These clusters of VM`s are formed by using K-Means clustering algorithm. So before moving to any datacenters, sets of VM`s are created and then passed to the nearest datacenter Id. It allows fast accessing of servers and also efficient utilization of available resources. This reallocation of VM`s improves the performance of CPU, memory and network operations by reducing the load on datacenters. The implementation of proposed algorithm is performed by using CloudSim3.0.1 simulator. Since most of the existing load balancing approaches have relatively high complexity, this paper has focused on the selection problem of physical hosts for deploying requested tasks and proposed a novel heuristic approach called Load Balancing based on Bayes and Clustering (LB-BC). LB-BC introduces the concept of achieving the overall load balancing in a long-term process in contrast to the immediate load balancing approaches in the current literature. LB BC makes a limited constraint about all physical hosts aiming to achieve a task deployment approach with global search capability in terms of the performance function of computing resource [9, 15]. The Bayes theorem is combined with the clustering process to obtain the optimal clustering set of physical hosts finally. Simulation results show that compared with the existing works, the proposed approach has reduced the failure number of task deployment events obviously, improved the throughput, and optimized the external services performance of cloud data centers. This paper has proposed a task deployment approach LBBC for the long-term load balancing effect and it has employed a heuristic idea based on Bayes theorem and the clustering process. LB-BC first has narrowed down the search scope by comparing performance values. Then, LBBC has utilized Bayes theorem to obtain the posteriori probability values of all candidate physical hosts. Finally, LB-BC has combined probability theorem and the clustering idea to pick out the optimal hosts set, where these physical hosts have the most remaining computing power currently, for deploying and executing tasks by selecting the physical host with the maximum posteriori probability value as the clustering center and thus to achieve the load balancing effect from the long-term perspective.

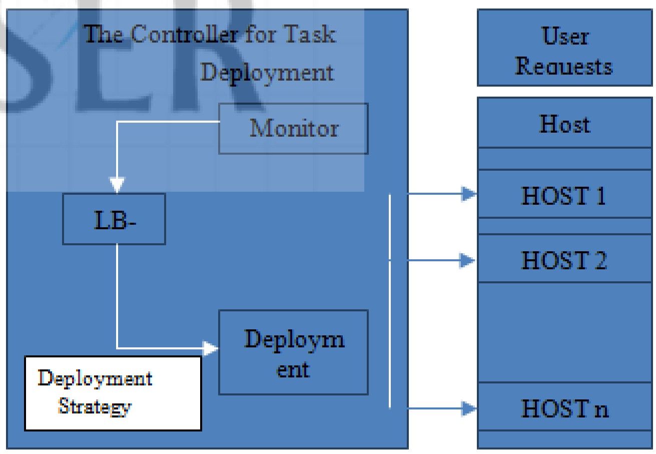

Figure 2: Existing LB-BC's Internal Architecture Model [9]

Hao Yuan et. al's also proposed an efficient Particle Swarm Optimization (PSO) based Virtual Machine Resource Scheduling in Cloud Computing Environment [10, 13]. Since Cloud Computing enable various Resources to be shared over distributed manner where Scheduling of Resources to be done efficiently and accurately. Here in this paper Particle Swarm Optimization (PSO) technique is implemented for the Sharing and Scheduling of Resources. Here the improved particle swarm optimization to obtain the optimal solution in reasonable time. The experiments show that the improved algorithms can provide effective solutions that the original algorithm cannot provide on cloud systems.

Shaoyi Song et. al's implemented an efficient Load Balancing methodology for Future Internet based on Game Theory [11]. During the survey of Future internet it is observed that Load balancing algorithms and job allocations are main research problems in areas of resource management of future internet. In this paper, we introduce a load balancing model for future internet. We formulate the static load balancing problem in the model proposed above as non cooperative game among users and cooperative game among processors. Based on this model, we derive a load balancing algorithm for computing center. Finally, execute the algorithm presented in this paper with another three algorithms for comparison purpose. The advantages of our algorithm are better scalability to the model, improving system performance, and low cost on maintaining system information.

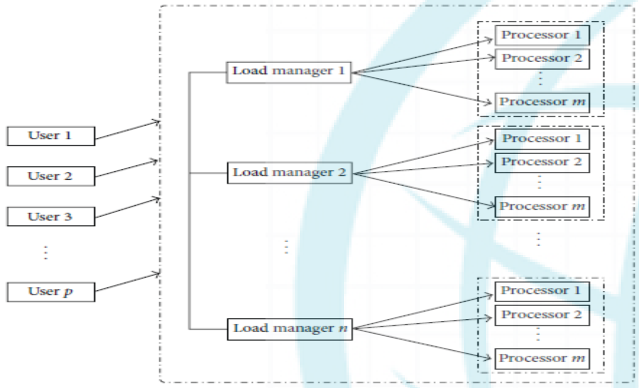

Figure 3: Load Balancing Model implemented for Future Internet [11]

Jacques M. Bahiet. al's proposed a new and efficient technique for Dynamic Load Balancing and load Estimator for Asynchronous Iterative Algorithms [12]. In this article, study the interest of coupling load balancing with asynchronism in such algorithms. After proposing a non-centralized version of dynamic load balancing which is best suited to asynchronism, we verify its efficiency by some experiments on a general Partial Differential Equation (PDE) problem. Finally, by giving some general conditions for the use of load balancing to obtain good results with this kind of algorithms and discuss the choice of the residual as an efficient load estimator.

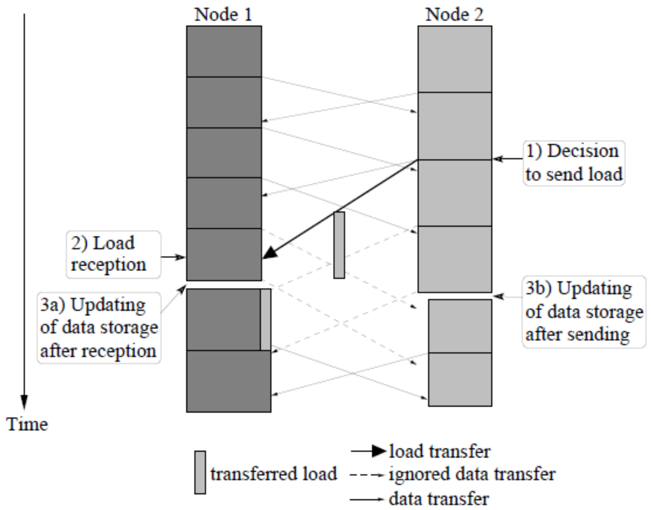

Figure 4: Flow Chart of the Working Methodology [12]

Mr. R. Gogulanet. al's implemented a Multiple Pheromone algorithm for the scheduling in Cloud Computing Environment for various QoS Requirements [13, 22]. This paper introduces a new algorithm called Multiple Pheromone Algorithm which is belongs to Ant Colony Optimization Algorithm. Xin Lu, Zilong GU [13, 22], 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.

The objective of MPA algorithm is to dynamically generate an optimal schedule so as to complete the task in minimum period of time as well as utilizing the resources in an efficient way. In this paper three different Quality of Service (QoS) make span, cost and reliability constraints are considered as performance measure for scheduling. This algorithm is compared with normal Ant colony algorithm, Genetic Algorithm. With the implementation of this approach, the Multiple Pheromone Algorithm (MPA) Algorithm reaches optimal solution as well as obtains the better QoS than ACO and GA.

B. Kruekaew et. al's implemented an Artificial Bee Colony based Scheduling Algorithm for Virtual Machines in Cloud Computing [14]. In this paper, Artificial Bee Colony (ABC) is applied to optimize the scheduling of Virtual Machine (VM) on Cloud computing. The main contribution of work is to analyze the difference of VM load balancing algorithm and to reduce the make span of data processing time. The scheduling strategy was simulated using CloudSim tools. Experimental results indicated that the combination of the proposed ABC algorithm, scheduling based on the size of tasks, and the Longest Job First (LJF) scheduling algorithm performed a good performance scheduling strategy in changing environment and balancing work load which can reduce the make span of data processing time.

In this paper [16, 17, 27] they have proposed a priority based job scheduling algorithm which can be applied in cloud environments they have named it "PJSC". Also we have provided a discussion about some issues related to the proposed algorithm such as complexity, consistency and finish time. 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 consider priority of jobs. It is a multi-criteria decision-making (MCDM) and multi-attribute decision-making (MCDM) model. Basically architecture of AHP is consisted of three levels which are objective level, attributes level and alternatives level respectively. Result of this paper indicates that the proposed algorithm has reasonable level of complexity. In this paper [16, 17, 27] we have proposed a novel scheduling heuristic by considering QoS factor in scheduling and have proposed some modifications using existing Sufferage heuristic and Min-min heuristic. Additionally, all of these scheduling approaches consider neither both the different levels of user tasks' Qos requests nor the resource properties of dynamic and heterogeneity in cloud computing environment. A QoS guided Sufferage-Min heuristic model, mainly inheriting from the Sufferage algorithm and Min-min algorithm, is presented in this paper after comparing and analyzing different heuristic algorithms. This model is composed of a Sufferage-Min heuristic algorithm and some Qos guided scheduling strategies considering the QoS requirements, low execution complexity, and the dynamic and heterogeneity resource properties in cloud computing environment. We have compared our proposed scheme to other scheme based on heuristic algorithm with a particular example and proved that the proposed scheduling heuristic had a significant performance gain in terms of reduced makespan. QoS is an extensive concept and varies from different research application.

Mayank Mishra et al. [18] 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.

Venkatesa Kumar. V and S. Palaniswami [19], 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 [20] 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.[21] 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.

Zhongni Zheng, Rui Wang [23] 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 [24] 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.

Sunita Bansal, Bhavik Kothari, Chitranjan Hoda [25] proposed a novel grid scheduling heuristic that adaptively and dynamically schedules task without requiring any prior information on the workload of incoming tasks. This models the grid system in the form of a state – transition diagram with job replication to optimally schedule jobs. This algorithm uses prediction information on processor utilization. In this algorithm they uses concept of job replication that is, a job can be replicated to other resource if that resource completes execution of current job than the resource it is currently allocated. This algorithm uses two types of queue namely, Waiting Queue and Execution Queue. This approach is based on exploiting information on processing capability of individual grid resources and applying replication on tasks assigned to the slowest processors. The approach facilitates replication of tasks, and also assigned to execute on slower machines, on machines with higher processing capacity. In this approach the communication costs are ignored. Experimental results show the better performance of this approach compared to traditional round robin algorithm.
Li Yang, ChengSheng Pan, ErHan Zhang, HaiYan Liu [26] proposed one kind of weighted fair scheduling algorithm. It is based on strict rob priority class which adds an absolute priority queue based on the foundation of based class weighted fair scheduling algorithm (CBWFQ). This algorithm covers the disadvantage of traditional weighted fair scheduling algorithm. Weighted Fair Scheduling algorithm differentiates the services of all active queues on the basis of weight of each business flow. When a new job arrives the classifier classifies the jobs into categories. Then buffer is checked for each category and if buffer is not overloaded then job is stored in the buffer otherwise job is dropped. Each job enters a different virtual queue. Weight, Dispatch, Discard and Rob are four main rules of this algorithm. The main advantage of this algorithm is that it has introduced the rob rule together with dropping rule. Experiments are done on NS-2 software to simulate SRPQ-CBWFQ algorithm. This new algorithm combined buffer management and queue scheduling and only guarantees low delay of real time applications. It also gave consideration to fairness and better utilization of buffers. This algorithm has two great advantages of bandwidth allocation and delay without throughput reducibility.

Shamsollah Ghanbari, Mohamed Othman [16, 17, 27] presented a novel approach of job scheduling in cloud computing by using mathematical statistics. This algorithm considers the priority of jobs for scheduling and named as priority based job scheduling algorithm. It is based on multiple criteria decision making model. A pair wise comparison based on multiple criteria and multiple attributes method was first developed by Thomas Saaty [28] in 1980 and named as Analytical Hierarchy Process (AHP). Consistent Comparison Matrix is the foundation of AHP, so to use the concept of AHP comparison matrices are computed according to the attributes and criteria's accessibilities.

5. Conclusion

The Various Load Balancing and Scheduling techniques are discussed and analyzed here. Since Load Balancing techniques requires a lot of computational overhead lot of techniques are implemented to solve the issues. Here by analyzing these techniques their various advantaged and limitations an efficient technique for Load Balancing can be implemented in future.

S.No	*Author*	*Year*	*Technology Used*	*Advantages*	*Issues*
1.	En-Juichang et.al.	2010	ACO-based Cascaded Adaptive Routing (ACO CAR)	Concept About Ant Colony Optimization	More memory utilization
2.	Le Shanguo et.al.	2011	Pheromone And available Resources On The Link As A Aspect For Load Balancing.	Load Balancing Is Anticipated.	Traffic Problems
3.	Marco Dorigoa et.al	2011	Concept for simulated network models between arbitrary nodes.	Traffic Problems with Network load imbalance are removed	Lower throughput
4.	Rata Mishra et.al.	2012	Pseudo Boolean Solution is Implemented.	Efficient than existing Methodology	Only one Pheromone is updated.
5.	Preeti Kushwah et.al.	2014	Load Balancing for Minimum Spanning tree implemented.	Compared Heuristic And Evolutionary Approach	Performance Degraded.
6.	Jia Zhao Et.al.	2016	Heuristic Clustering based on Bayes Theorm Implemented.	A Very New Concept Implemented.	Highly Complex Methodology.
7.	Kamil Krynicki	2015	Non-Hybrid Ant Colony Optimization Heuristic for Convergence Quality	Increased Efficiency and Flexibility with Adaptability.	Doesn't support multiclass resource querying.
8.	Chia-Feng Juang	2014	Cooperative Continuous ant Colony Optimization (CCACO).	Useful for Fuzzy Controller and Design Problems.	CCACO can't be applied to multiobjective FS design problems for optimization.
9.	Gengbin Zheng	2010	an automatic dynamic hierarchical load balancing method that overcomes the scalability challenges of centralized schemes and poor solutions of traditional distributed schemes.	Scalable and provides reduces memory.	Doesn't provide interconnect topology aware strategies that map the communication graph on the processor topology to minimize network contention.

10.	Jun Wang	2009	Machine learning based load prediction and fuzzy logic based replica management for adaptive and flexible load balancing mechanism within the framework of distributed middleware.	Fluctuate load can be easily managed and flexible.	More number of iterations to be performed.

Algorithm	Static Environment	Dynamic Environment	Centralized Balancing	Distributed Balancing	Hierarchical Balancing
Round-robin	YES	NO	YES	NO	NO
Ant Colony	NO	YES	NO	YES	NO
Map Reduce	YES	NO	NO	YES	YES
Particle Swarm Optimization	NO	YES	NO	YES	NO
MAXMIN	YES	NO	YES	NO	NO
MINMIN	YES	NO	YES	NO	NO
Genetic Algorithm	NO	YES	YES	NO	NO

References

[1] Chunyan Zhao. Research and Implementation of Job Scheduling Algorithm Under Cloud Environment. Master's Thesis of Beijing jiaotong University. 2009.

[2] Xindong You, Guiran Chang, Xueyao Deng, et. Grid Task Scheduling Algorithm Based on Merit Function. Computer Science. 2006; 33(6).

[3] Thomas A. Hen zinger, Anmol V. Singh, Vasu Singh, Thomas Wies, "Static Scheduling in Clouds".

[4] T.Casavant and J.Kuhl, "A Taxonomy of Scheduling in General Purpose Distributed Computing Systems", "IEEE Trans. On Software Engineering", vol.14, no.3, February 1988, pp.141-154

[5] M.Arora, S.K.Das, R.Biswas, "A Decentralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments", "Proc. Of International Conference on Parallel Processing Workshop (ICPPW'02)", Vancouver, British Columbia Canada, August 2002, pp.400-505

[6] Fatos Xhafa, Ajith Abraham, "Computational models and heuristic methods for Grid scheduling problems", "Future Generation Computer Systems 26", 2010, pp.608-621.

[7] Yun-Han Lee et al, Improving Job Scheduling Algorithms in a Grid Environment, Future Generation Computer Systems, 27 (2011) 991–998.

[8] Wang Juan Li Fei Chen Aidong, "An Improved PSO based Task Scheduling Algorithm for Cloud Storage System", AISS: Advances in Information Sciences and Service Sciences, Vol. 4, No. 18, pp. 465-471, 2012.

[9] Jia Zhao, Kun Yang," A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment", *IEEE Transaction on Parallel and Distributed Systems*, Volume 27, Issue 2, February 2016.

[10] Hao Yuan, Changbing Li, Maokang Du," Optimal Virtual Machine Resources Scheduling Based on Improved Particle Swarm Optimization in Cloud Computing", *Journal of Software*, Volume 9, Issue 3, March 2014.

[11] Shaoyi Song, TingjieLv, Xia Chen," Load Balancing for Future internet: An Approach Based on Game Theory", *Journal of Applied Mathematics*, 2014.

[12] Jacques M. Bahi, Sylvain Contassot-Vivier, Raphael Couturier," Dynamic Load Balancing and Efficient Load Estimators for Asynchronous Iterative Algorithms", *IEEE Transactions on Parallel and Distributed Systems*, 2005.

[13] Mr.R.Gogulan, Ms.A.Kavitha, Mr.U.Karthick Kumar," An Multiple Pheromone Algorithm for Cloud Scheduling with Various QoS Requirements", *International Journal of Computer Science*, Volume 9, Issue 1, May 2012.

[14] B. Kruekaew and W. Kimpan," Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony", *Proceedings of the International MultiConference of Engineers and Computer Scientists* 2014 Vol I, IMECS 2014, March 12 - 14, 2014, Hong Kong.

[15] Bhupendra Panchal , Prof. R. K. Kapoor," Dynamic VM Allocation Algorithm using Clustering in Cloud Computing", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 9, September 2013.

[16] Shamsollah Ghanbari,Mohamed Othman." A Priority based Job Scheduling Algorithm in Cloud Computing" International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012).

[17] Haiwen Han, Qi Deyu "A Qos Guided task Scheduling Model in cloud computing environment" Fourth International Conference on Emerging Intelligent Data and Web Technologies, 2013.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] Xin Lu, Zilong Gu, "A Load-Adaptive Cloud Resource Scheduling Model Based on Ant Ccolony Algorithm", 2011, 978-1-61284-204-2/11, Proceedings of IEEE CCIS2011, Page no: 296-300.

[23] 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.

[24] 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.

[25] Sunita Bansal et al, Dynamic Task-Scheduling in Grid Computing Using Prioritized Round Robin Algorithm, IJCSI International Journal of Computer Science Issues, 8(2)(2011) 472-477.

[26] Li Yang et al, A new Class of Priority-based Weighted Fair Scheduling Algorithm, Physics Procedia, 33 (2012) 942 – 948.

[27] Ghanbari, Shamsollah, and Mohamed Othman. "A Priority based Job Scheduling Algorithm in Cloud Computing." Procedia Engineering 50 (2012): 778-785.

[28] T.L. Saaty, The Analytic Hierarchy Process,(New York 1980).