ISSN:0975-9646

Study of Task Scheduling Algorithms in the Cloud Computing Environment: A Review

Ashwani Kumar Yadav[#1], Hardwari Lal Mandoria[*2]

[#] *Research Scholar, Department of Information Technology, College of Technology,*
Govind Ballabh Pant University Of Agriculture & Technology, Pantnagar, India

[*] *Professor,*
Department of Information Technology, College of Technology,
Govind Ballabh Pant University Of Agriculture & Technology, Pantnagar, India

***Abstract*— Cloud computing is a recent advancement in the internet world .The internet world has been revolutionized by this provision of shared resources. Cloud service providers compete for scalability of virtualized resources dynamically. The performance and efficiency of cloud computing services always depend upon the performance of the user tasks submitted to the cloud system. Cloud services performance can be significantly improved by scheduling the user tasks. The cost emerging from data transfers between resources as well as execution costs must also be taken into consideration while optimizing system efficiency in scheduling. Moving applications to a cloud computing environment trigger the need for scheduling as it enables the utilization of various cloud services to facilitate execution. Service provider's goal is to utilize the assets effectively and increase benefit. This makes task scheduling as a core and challenging issue in cloud computing. It is the process of mapping task to the available resource. This paper presents a detailed study of various task scheduling methods existing for the cloud environment.**

***Keywords*— Cloud computing; Virtualization; Task scheduling; Cloudsim; Makespan.**

I. INTRODUCTION

Cloud Computing is the newest trend in the field of computer science and is said to be the future of modern technology. It is popular in the internet world mostly for its special ability to utilize shared resources most efficiently. Hence presents a model for providing ubiquitous, sizeable, on-demand access to a shared network containing a pool of configurable processing assets that can be effortlessly furnished and discharged with negligible service provider interaction.[38] Currently, cloud computing provides dynamic services over the internet including applications, data, memory, bandwidth, and IT services. A substantial number of commercial cloud service providers (CSPs) have started to deliver various public cloud computing services. An ever increasing number of undertakings and organizations build their own cloud computing infrastructure or resort to a hybrid cloud.

Cloud task scheduling belongs to NP-complete problem. A task is a small portion of work that should be executed within a given duration of time. It is done on the basis of different parameters so that it enhances the overall cloud performance. A task may be related to entering data, processing, accessing software, or storage functions. The data center specifies tasks according to the service-level agreement and demanded services. In the process, the users submit their jobs to the cloud scheduler. The cloud scheduler probes the cloud information service for acquiring the status of available resources and their properties and hence allocating the different tasks onto diverse resources as per the task specifications. Cloud Scheduler will designate multiple user tasks to many virtual machines. A Good scheduler always selects the virtual machines in an optimal way. A good scheduling algorithm improves the CPU utilization, turnaround time and combined throughput. Task scheduling can be implemented based on various parameters in distinct ways. They can be allocated statically at compile time or allocated dynamically at runtime.

II. CATEGORIZATIONS OF TASK SCHEDULING ALGORITHMS

Based on the relevant works in literature [3],[4],[11],[23],[28],[39],[41],[45],[48], are categorizing scheduling methods in cloud environment generally into three groups: resource scheduling, workflow scheduling, and task scheduling.

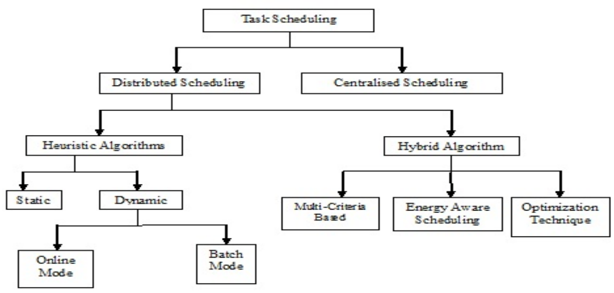

Fig1. Task Scheduling Algorithms(flowchart)

Our focus is mainly on the task cheduling as shown in fig.1 Resource scheduling does mapping of virtual machines on physical machines and workflow scheduling is done to schedule workflows composing an entire job in a proper order. Task scheduling techniques may be centralized or distributed. It can be implemented on dependent or independent tasks in an environment comprising homogeneous or heterogeneous resources. In centralized scheduling, a single scheduler performs all mappings whereas, in distributed scheduling, different schedulers are employed for this purpose.

Distributed scheduling suffers due to its high implementation complexity. Since the workload is distributed to partner nodes, therefore, processor cycles are saved. Centralized scheduling is easily implementable. Since it has a single scheduler, therefore, it always has a single point of failure. Centralized Scheduling is less scalable and fault tolerant. Distributed Scheduling methods can be of two types: heuristic and hybrid techniques. Heuristic methods are categorized into static as well as dynamic scheduling. Dynamic scheduling can be done in online mode or batch mode. In static scheduling, tasks details are known apriori to scheduling and they are statically designated to virtual machines. In dynamic scheduling, all the tasks are scheduled immediately, as they enter the system. Dynamic scheduling mechanism works much better compared to static. But the cost of dynamic algorithms is high as we want to determine the schedule and update the system information immediately.

The main purpose of job scheduling is to deliver a high performance in computing and best throughput of the system. Static scheduling is easily implementable from programmer's perspective whereas dynamic scheduling is fit for real world scenarios. Dynamic scheduling lessens the cost required to be spent for running the scheduler. Hybrid algorithms are of three types: multi-objective, minimization-maximization approach or energy aware methods.

III. SCHEDULING MODEL

A Scheduling process in the cloud computing scenario holds several structural elements as shown in the Fig. 2[5]. This model is given for the implementation of virtualization in the cloud computing framework. All the computing facilities like the operating system, software etc. are provided by the number of virtual machines in the cloud system that processes the submitted tasks.

1. Computing entity: They are designated by the computing capacity which is indicated by the number of instructions it can process in a second.

2. Job scheduler: It is the primary element of the scheduling process in the cloud computing environment. It computes the job sequence for execution for all the jobs waiting in the queue.

3. Job waiting queue: It is the queue in which the jobs are waiting to get assigned to an appropriate machine for execution.

4. Job arriving process: It is the method by which jobs enter in the scheduling system

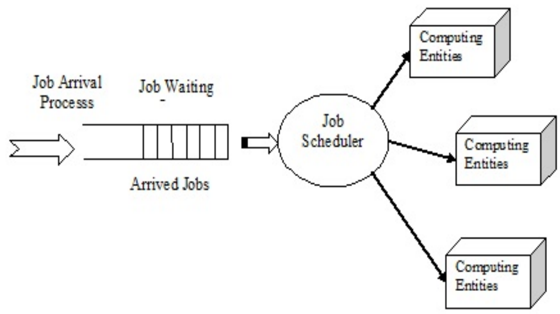

Fig. 2. Scheduling Model[5]

IV. EXISTING TASK SCHEDULING ALGORITHMS

In this paper we have presented three main categories of task scheduling that are heuristic, hybrid and energy efficient task scheduling. Each of these categories is described briefly with their further types:

A. *Heuristic Task Scheduling Approaches :*

1) *Static Scheduling Methods*: Static scheduling algorithms assume all tasks arrive at the same instant of time and they are independent of the system resource's States and their availability. The basic scheduling policies like First-Come-First-Serve and Round-Robin methods are implemented in static mode. FCFS methods receive the tasks and queue them until resources are available and once they become available the tasks are allotted to them depending on their arrival time. No other criteria for scheduling are considered in this technique this makes it less complex in nature. On the other hand, RR schedule [3] uses the similar technique but it grants a resource to a task for a particular time interval [41]. These tasks are then queued for the next execution. Yet Another heuristic method is Opportunistic load balancing which is based on their expected completion time. It schedules the tasks on the next available machines. It will bring about poor make-span because of the fact that it tries to use the resources making all machines busy at the same time.

Minimum Execution Time and Minimum Completion Time[48][41] are other two heuristic approaches: in which MET maps tasks to machine depending on which machine takes less execution time and assigns it on the machines. This approach suffers from load imbalance as it selects the best machine for execution but avoids considering the availability of resources at the time of scheduling. Minimum Completion Time Algorithm chooses machines with minimum expected completion time of tasks among all the available machines. Machine examined for load before scheduling of task on that machine. Any two tasks cannot have minimum execution time on the same machine. Completion time of a task on a machine can be characterized as the aggregate of the execution time of the task on that machine and the ready time of that specific machine.

Min-Min and Max-Min[3][48] heuristic approaches. Min-min heuristic firstly selects the smallest task from among the available tasks. The smallest task selected is run on a machine with minimum completion time for that task. This method has a overall increase in completion time of task and consequently makespan also increases But does not consider load of the machines before scheduling. Here the expected completion time and execution time for a task are considered to be nearly same values or close values. The long tasks need to sit to complete the execution of smaller ones. Although this method improves the system's overall throughput. Max-Min is same as min-min with the exception of that it chooses the longest task(with most maximum completion time) first to map on the machine that has minimum completion time. Here the smaller tasks starve and load balancing is not considered. Since the longest task decides the makespan of all the accessible tasks in the framework. It tends to increases the makespan and system throughput than the min-min strategy. Thus in max-min, first the longer tasks will be executed in faster machines and also smaller tasks can be executed in parallel on other conceivable machines which brings about better makespan and adjusted load than the former technique.

Genetic Algorithm[46] and Simulated Annealing[30] are two other common schemes in heuristic approach which is used to perform proximal optimal scheduling. In Genetic Algorithm approach four conventional operations evaluation, selection, crossover and mutation are performed. The initial population represents the potential mappings of the given task list on the available machines. Each job is represented as a vector in which each position of that vector represents a task in the task list. The value in each position denotes the machine to which the task is mapped. Each job represents a chromosome. Fitness value for a chromosome indicates the overall execution time of all the tasks that are formed from mapping constituting that chromosome and it is chosen such that it lessens makespan. This method uses earlier results with existing results to get better conceivable mappings and survival of the fittest takes place.

Simulated Annealing is an iterative technique which can be represented comparable to genetic algorithm in which it starts with a single mapping selected from a random distribution. The underlying variant of SA is assessed to get a better form. Following mutation the new makespan is examined. If it is less than the preceding one then replace the old one with the new makespan. Simulated Annealing provides poorer mapping solutions than that of Genetic Algorithm. The aggregated features of both can give a better scheduling solution[30][43].

2) Dynamic Scheduling Methods : In dynamic scheduling method [3][41]is developed in view of dynamic nature of tasks. It is dependent on the system machine's state and tasks arrive at the varying point of time.

It is mainly categorized into two types: (1) online mode and (2) batch mode. In online mode tasks are assigned immediately once they arrive in the framework like most-fit task scheduling algorithm whereas in batch mode tasks are assembled in a group and scheduled at their predefined time. The popular examples in batch mode are Min-min, max-min and round robin. Similarly online modes have MCT, MET and OLB.

Switching algorithm is an algorithm that switches between MET and MCT according to the load of the system. K-Percent Best is another heuristic of same kind in which a subset of k computationally higher positioning machines is first chosen amid the scheduling procedure. A decent estimation of schedule demonstrates that it generally doles out a task to a machine from this list only. This method reaches a better makespan compared to MCT. It preserves machines that are more fit for yetto- arrive tasks.

In batch mode[48] another heuristic is called sufferage heuristic schedules tasks according to sufferage values[41]. It is calculated from the first and second earliest completion time of a task. The task with higher sufferage is selected for scheduling on a same resource.

B. Energy Efficient Task Scheduling Approach

The power management of a data center relies on different determinants and task scheduling is a critical one among them. The brief overview to different task scheduling algorithms that predominantly concentrate on the increasing energy efficiency, power consumption reduction, and cost reduction and performance improvement is as follows.

The three algorithms are given which predominantly concentrates on instructions to deal with a request from the clients in heterogeneous system. The first one is a benefit driven algorithm in which the tasks are doled out on the best server machines based on benefit value calculated. This method works for heterogeneous systems. For homogeneous systems here they are proposing two strategies: power best-fit algorithm in which they consider the machine with minimum power utilization increase as its decision for scheduling the task. And the other one is load balancing approach in which load balancing is done based on the power frequency ratio of each resource. Power frequency ratio indicates the computing capacity of the server.

In [12]; this energy efficient job scheduling method mainly concentrates on traffic load balancing in cloud data centers. They look on the traffic specifications requirements of the cloud applications. In turn, it limits congestion and communication delays in the network.

In [11]; Network information and Energy efficiency are combined and scheduling is based on this combination.. It satisfies QoS requirements and enhances job performance. Feedback channels are employed to obtain network awareness from the main network switches. It decreases the number of computing servers and avoids hotspots. This method has less computational and memory overhead.

In [36]; an optimized scheduling strategy is implemented to lower power consumption simultaneously satisfying task response time constraints during scheduling. Its greedy nature chooses a minimum number of most efficient servers for scheduling. The tasks are heterogeneous in nature so that they constitute distinctive different energy consumption levels and have different task response times. Optimal assignment is based on minimum energy dissipation and minimum completion time of a task on a particular machine.

In [9]; A green energy efficient way of scheduling is proposed using DVFS technique. The power utilization of infrastructure is reduced by the using Dynamic Voltage Frequency Scaling method. Limiting number of processing servers and time decreases energy utilization and can enhance resource usage. The servers are operated at various frequencies and voltages combinations. This method efficiently maps the tasks to resources without compromising the performance of the system meeting the SLA requirements and saving energy[8][2].

C. *Hybrid Scheduling Algorithms*

Large numbers of these algorithms are novel or are produced on the top of some current techniques combining more scheduling parameters to enhance the execution. Some of the existing works under this category are given below:

In [35]; they schedule tasks based on their cost against different resources. The cost of services fluctuates for various tasks in light of their complexities. The algorithm considers resource cost and processing capability of resources. Tasks are grouped based on the processing capacity and chooses some best resources to plan them in such an approach to lessen cost. This algorithm decreases the makespan and the processing cost when looked at to other scheduling algorithms like Activity Based Costing.[40] In QoS driven task scheduling algorithm; task's priority is added for scheduling them on various resources[51]. In light of the distinctive attributes of the tasks, priorities are ascertained for the tasks and they are sorted on that basis. At that point, they are allocated on the machine which delivers the best completion time. Thus this algorithm enhances execution by having better completion time.

In [34]; the tasks are partitioned into various groups and they are replicated to local middleware of the system. It makes the system fault tolerant and load balancing improves response time and resource utilization. Lexi search method is utilized here to map the tasks to different resources alongside decreasing the cost. The task is doled out according to a probabilistic estimation which is dependent on accessibility of the resource and execution time of the task. Load balancing decreases the overhead made at the scheduler in each resources.

In [33] they build up an algorithm in light of customary min-min algorithm which incorporates planning in view of load of the servers and considering the user priority. The users are categorized into two levels as VIP and ordinary users. Load is adjusted in view of the maximum loaded resource and the makespan of the system. The method demonstrates a decent pick up in user satisfaction, makespan and resource utilization ratio.

In [28]; an enhancement over the previously existing weighted least connection algorithm is Dual weighted least-connection algorithm. In this technique, the weights (processing capacity) of the servers are ascertained progressively and loads of the servers are calculated based on the properties of the tasks allocated to that server. The calculation gives better balancing of load and system efficiency compared with WLC technique.

In [33]; an algorithm is proposed in view of the divisible load theory that aims to diminish the overall processing time of the tasks. Homogeneous processors are utilized here for which the load fractions and processing time for each task are calculated. The divisible load is partitioned leading to reduced completion time of tasks. This method benefits cloud providers and also increases quality of service. The performance, total cost, delay cost, efficiency quality are better when compared to other random methods.

In [49]; they propose an algorithm which is an adjustment done on the improved max-min algorithm[3]. Having the knowledge of expected execution time it assigns a task with execution time equivalent to average execution time on a machine that gives minimum completion time. Hence the largest task is the deciding factor for makespan of the system. There may be a condition of load imbalance if a task is too large. Therefore instead of choosing the largest task to schedule; they choose the task equivalent to an average largest task or nearest to average largest task. This method produces a load balancing better and makespan is also reduced compared to improved max-min method.

V. COMPARISON OF EXISTING SCHEDULING ALGORITHMS

A good scheduling algorithm always considers benefits of both the parties the cloud users and the service providers. The algorithms should try to reduce both the cost and power consumption as well as provide better performance. Scheduling algorithms must consider Load balancing and energy consumption as there two main parameters. Moreover, it should provide the user's fairness and security while providing services. A future enhancement in developing a suitable algorithm is by considering the combination of some important parameters together which can be deployed in a cloud environment for providing better cloud services to the users.[41][28]

The main scheduling parameters considered in the previously mentioned methods are listed below:

• Makespan: It is the aggregate consummation time of all tasks in the job queue. A good scheduling algorithm dependably tries to diminish the makespan.

• Deadline : It is characterized as the timeframe from presenting a task to the time by which it must be finished. A good scheduling algorithm dependably tries to keep the tasks executed with in the deadline constraint.

• Execution Time: This is the exact time taken to execute the given tasks. A good scheduling algorithm ultimately aims to minimize execution time.

• Completion Time: Completion time is the time taken to finish the whole execution of work. It incorporates the execution time and delay caused by the cloud system. A number of existing scheduling algorithms consider minimizing completion time of tasks.

• Energy Consumption: Energy utilization in cloud data centers is a present issue that ought to be considered with more care nowadays. Numerous scheduling algorithms were developed for diminishing power consumption and enhancing execution and consequently making the cloud services green.

• Performance: Performance shows the by and large productivity given by the scheduling algorithm all together to give good services to the clients according to their necessities. A good scheduling algorithm ought to consider the execution at the client end and in addition the cloud service provider end.

• Quality of Service: SLAs is defined as a contract document defined between the cloud user and cloud service provider. Input constraints such as meeting execution cost, deadline, performance, cost, makespan, etc enhances quality of service.

• Load balancing: It is the strategy for dissemination of the whole load in a cloud network crosswise over various nodes furthermore, connects so that at once no nodes and connections remain under loaded while a few nodes or connections are over-loaded. Most of the scheduling algorithms try to keep the load balanced in a cloud network in order to increase the efficiency of the system.

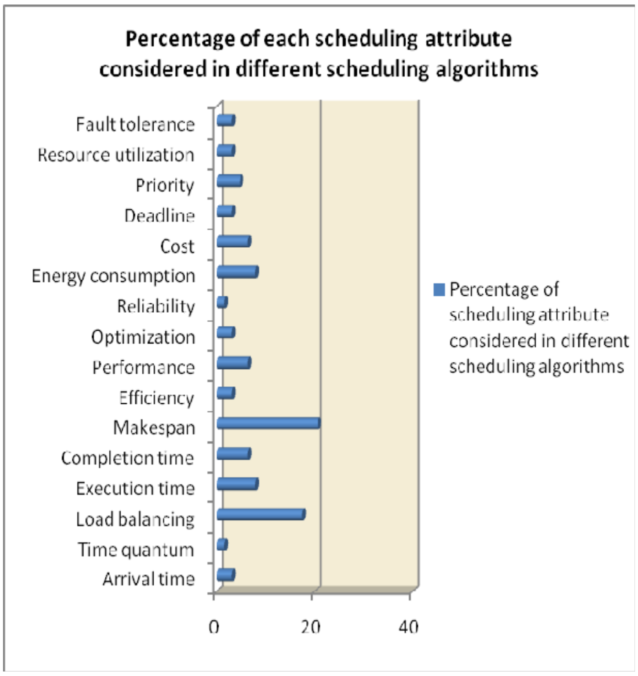

Fig. 3 Percentage of scheduling parameters considered in existing scheduling methods

From the diverse task scheduling algorithms discussed in this paper, the general rate of each scheduling parameters considered in various techniques are combined in Fig. 3. This investigation is restricted to the strategies explained in this paper. It recognizes the scheduling traits which are considered most and which all are less noteworthy in various schedules so that better algorithms can be created by varying slightly considered parameters or joining them with different parameters in existing calculations to get a good general scheduling.

From Fig. 3, we would see be able to that makespan and load balancing are considered reasonably in numerous strategies for development. Completion time, execution time, execution and energy utilization are then considered in a direct rate in the above examined calculations. A portion of the strategies are scheduling in light of priority, deadline, resource utilization, cost, efficiency and so on. Adaptation to non-critical failure and unwavering quality are additionally considered in less sums in existing strategies. A few algorithms give fundamental concentration to diminishing the cost, and to have better asset use inside the cloud frameworks along with optimization. However the current strategies considered have neglected to bolster above parameters together as it prompts high intricacy and cost. Creating new planning calculations considering more parameters together and henceforth delivering better execution results can be considered as an imperative issue in current situation. Existing techniques and methodologies can likewise be enhanced by consolidating more characteristics along these lines fulfilling client SLAs necessities and giving better Quality of service.

VI. Conclusion And Future Work

Productive scheduling algorithms dependably play a critical part in the execution given by a cloud computing system. An investigation of existing task scheduling algorithms is done in this paper. we considered heuristic, energy efficient and hybrid strategies for study. A short investigation of every strategy is done and most calculations perform scheduling in view of one or two parameters. A superior scheduling algorithm can be created from the current techniques by including more number of parameters which would result be able to in good performance and outputs that can be conveyed for deployment in a cloud environment in future. The table made, combines all the distinctive scheduling parameters utilized as a part of the current scheduling algorithms. A decent scheduling algorithm must consider the necessities of clients fulfilling their necessities given in SLA and in the meantime advantageous to the cloud providers. Consolidating distinctive parameters with the end goal that to get a productive scheduling algorithm and enhance the overall execution of the cloud computing should be possible upgrade.

REFERENCES

[1] Ahuja, R., De, A., Gabrani, G., "SLA Based Scheduler for Cloud for Storage & Computational Services", IEEE, 2011.

[2] Anju Bala, Dr.Inderveer Chana,"A Survey of Various Workflow Scheduling Algorithms in Cloud Environment", 2nd National Conference on Information and Communication Technology (NCICT) 2011 Proceedings published in International Journal of Computer Applications® (IJCA)

[3] Archana mantri, Suman Nandi, Gaurav Kumar, Sandeep Kumar, High Performance Architecture and Grid Computing Computing, International Conference HPAGC 2011, Chandigarh, India, July 2011, Proceedings.

[4] Arya L.K, Verma A, Workflow scheduling algorithms in cloud environment - A survey, Recent Advances in Engineering and Computational Sciences, 2014, IEEE,1-4.

[5] Bo Yang, Xiaofei Xu, Feng Tan, Dong Ho, A Utility-Based Job Scheduling Algorithm for Cloud Computing Considering Reliability Factor, 2011 IEEE.

[6] Boloor, K., Chirkova, R., Salo, T., Viniotis, Y., "Heuristic-Based Request Scheduling Subject to a Percentile Response Time SLA in a Distributed Cloud". IEEE, 2011

[7] Celaya, J., Arronategui, U., "A Highly Scalable Decentralized Scheduler of Tasks with Deadlines", IEEE, 2011

[8] Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, Jian Xie, Jicheng Hu, "Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing", IEEE, 2009

[9] Chia-Ming Wu, Ruay-Shiung Chang, Hsin-Yu Chan received, "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters", Elsevier, 2013.

[10] Ching-Hsien Hsu, Tai-Lung Chen, "Adaptive Scheduling Based on Quality of Service in Heterogeneous Environments", IEEE, 2010

[11] Dzmitry Kliazovich, Pascal Bouvry, Samee Ullah Khan, "DENS: Data Center Energy-Efficient Network-Aware Scheduling, Cluster Computing", special issue on Green Networks, vol. 16, no. 1, pp. 65-75, 2013.

[12] Dzmitry Kliazovich1, Sisay T. Arzo, Fabrizio Granelli, Pascal Bouvry and Samee Ullah Khan, "e-STAB: Energy-Efficient Scheduling for Cloud Computing Applications with Traffic Load Balancing", IEEE International Conference on Green Computing and Communications and

[13] Emeakaroha, V.C., Brandic, I., Maurer, M.and Breskovic, I., "SLA-Aware Application Deployment and Resource Allocation in Clouds", IEEE, 2011

[14] Fei Teng, "Resource allocation and scheduling models for cloud computing", Paris, 20111

[15] Gan Guo-ning, Huang Ting-lei, Gao Shuai, "Genetic simulated annealing algorithm for task scheduling based on cloud computing environment", IEEE, 2010

[16] Gao Ming and Hao Li, "An Improved Algorithm Based on Max-Min for Cloud Task Scheduling", Yunnam University, China, 2011

[17] H. Izakian, A. Abraham, V. Snasel, "Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments," in Proc. of the International Joint Conference on Computational Sciences and Optimization, IEEE, vol. 1, 2009, pp. 8-12. 664 2014

[18] Huankai Chen, Frank Wang, Na Helian, Gbola Akanmu, "User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing", IEEE, 2013.

[19] J. Cao, D. P. Spooner, S. A. Jarvis, G. R. Nudd, "Grid Load Balancing Using Intelligent Agents", Future Generation Computer Systems, Vol. 21, No. 1, 2005, pp. 135-149.

[20] J. Yu and R. Buyya, "Workflow Scheduling Algorithms for Grid Computing", Technical Report, GRIDS-TR-2007-10, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, May 2007.

[21] Jiayin Li, Meikang Qiu, Jianwei Niu, Wenzhong Gao, Ziliang Zong, Xiao Qin, "Feedback Dynamic Algorithms for Preemptable Job Scheduling in Cloud Systems", IEEE, 2010

[22] Kai Zhu, Huaguang Song, Lijing Liu, Jinzhu Gao, Guojian Cheng, "Hybrid Genetic Algorithm for Cloud Computing Applications", IEEE, 2012

[23] Karan Dipakbhai Prajapati, Comparison of Virtual Machine Scheduling Algorithms in Cloud Computing, International Journal of Computer Applications12/2013;Volume83(973-93-80879-15-9):15.

[24] Kessaci, Y., Melab, N., Talbi, E.-G., "A pareto-based GA for scheduling HPC applications on distributed cloud infrastructures", IEEE, 2011

[25] Kuan-Rong Lee, Meng-Hsuan Fu, Yau-Hwang Kuo, "A hierarchical scheduling strategy for the composition services architecture based on cloud computing", IEEE,2011

[26] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Wang, D., "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization", IEEE, 2011

[27] Laiping Zhao, Yizhi Ren, Sakurai, K.,"A Resource Minimizing Scheduling Algorithm with Ensuring the Deadline and Reliability in Heterogeneous Systems", IEEE, 2011

[28] Lianying ZHOU, Xingping CUI, Shuyue WU, An Optimized Loadbalancing Scheduling Method Based on the WLC Algorithm for Cloud Data Centers, Journal of Computational Information Systems, 2013.

[29] Luqun Li, "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers", IEEE, 2009

[30] M. Coli, P. Palazzari, "Real Time Pipelined System Design through Simulated Annealing," Journal of Systems Architecture, vol.42, no. 6-7, 1996, pp. 465-475.

[31] M. Malathi, "Cloud Computing Concepts", IEEE,2011

[32] Mehdi, N.A., Mamat, A.; Amer, A., Abdul-Mehdi, Z.T.,"Minimum Completion Time for Power-Aware Scheduling in Cloud Computing", IEEE, 2012

[33] Monir Abdullaha, Mohamed Othmanb, Cost Based Multi QoS Job Scheduling using Divisible Load Theory in Cloud Computing, International Conference on Computational Science, ICCS 2013.

[34] Mousumi Paul, Debabrata Samanta, Goutam Sanyal, "Dynamic Job Scheduling in Cloud Computing Based on Horizontal Load Balancing", IJCTA, 2011.

[35] Mrs.S.Selvarani, Dr.G.Sudha Sadhasivam, "Improved Cost-Based Algorithm For Task Scheduling in Cloud Computing", IEEE, 2010.

[36] Ning Liu, Ziqian Dong, Roberto Rojas-Cessa, "Task Scheduling and Server Provisioning for Energy-Efficient Cloud-Computing Data Centers", IEEE 33rd International Conference on Distributed Computing Systems Workshops Task, 2013.

[37] Paul, M., Sanyal, G., "Survey and analysis of optimal scheduling strategies in cloud environment", IEEE, 2012

[38] Peter Mell, Timothy Grance, "The NIST definition of Cloud Computing (September, 2011)", Accessed on May, 2014.

[39] Pinal Salot, A Survey of Various Scheduling Algorithm in Cloud Computing Environment, IJRET, February 2013, Available @ http://www.ijret.org/.

[40] Qi Cao, Zhi-Bo Wei, Wen-Mao Gong, "An Optimized Algorithm for Task Scheduling Based on Activity Based Costing in Cloud Computing", IEEE,2009

[41] S.Nagadevi1, K.Satyapriya2, Dr.D.Malathy3, "A Survey On Economic Cloud Schedulers For Optimized Task Scheduling", International Journal of Advanced Engineering Technology, 2013.

[42] Selvarani, S., Sadhasivam, G.S., "Improved cost based algorithm for task scheduling in cloud computing ", IEEE, 2011

[43] Shenai Sudhir, Survey on Scheduling Issues in Cloud Computing, Procedia Engineering, Elsevier, 2012.

[44] Shu-Ching Wang, Kuo-Qin Yan, Shun-Sheng Wang, Ching-Wei Chen, "A Three-Phases Scheduling in a Hierarchical Cloud Computing Network", IEEE, 2011

[45] Simsy Xavier, S.P.Jeno Lovesum, A Survey of Various Workflow Scheduling Algorithms in Cloud Environment, International Journal of

[46] Sung Ho Jang, Tae Young Kim, Jae Kwon Kim , Jong Sik Lee School, "The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing", International Journal of Control and Automation Vol. 5, No. 4, December, 2012.

[47] Thomas A. Henzinger, Anmol V. Singh, Vasu Singh, Thomas Wies, Damien Zufferey, "Static Schedling in Clouds", Austria

[48] Tracy D. Braun, Howard Jay Siegel, Noah Beck, "A Comparison of Eleven Static Heuristics or Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing 61, pp. 810-837, 2001.

[49] Upendra Bhoi, Purvi N. Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing", International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 4, April 2013, pp 259-264.

[50] Van den Bossche, R., Vanmechelen, K., Broeckhove, J., "Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds", IEEE, 2012
[51] Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, Shengyuan Zhou, “A Task Scheduling Algorithm based on QoS driven in Cloud Computing”, Information Technology and Quantitative management, 2013.
[52] Yuanyan Gu, Yujia Ge, "A Real-Time Workload Driven Approach for the Cloud", IEEE, 2010
[53] Yujia Ge, Guiyi Wei, "GA-Based Task Scheduler for the Cloud Computing Systems", IEEE, 2010
[54] H. Izakian, A. Abraham, V. Snasel, “Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments,” in Proc. of the International Joint Conference on Computational Sciences and Optimization, IEEE, vol. 1, 2009,