Indian Journal of Science and Technology, *Vol 10(9), DOI: 10.17485/ijst/2017/v10i9/106576, March 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Locality-Load-Prediction Aware Multi-Objective Task Scheduling in the Heterogeneous Cloud Environment

Yadaiah Balagoni[1*] and R. Rajeswara Rao[2]

[1]Department of Computer Science and Engineering, MGIT, Gandipet Main Road, Kokapet, Hyderabad - 500075, Telangana, India; yadaiahbalagoni@mgit.ac.in
[2]Department of Computer Science and Engineering, JNTU Vizianagaram, Dwarapudi, Vizianagaram - 535003, Andhra Pradesh, India; rajaraob4u@gmail.com

Abstract

Objectives: Current state-of-the-art task scheduling algorithms were mainly focused on deadline, load and energy factors in centralized cloud context. So, the proposed research objective focuses on dynamic and decentralized context. **Methods/Statistical Analysis:** Multi-objective task scheduling has become an important criterion for the dynamic and decentralized nature of cloud environment. Moreover, existing research works assumes that the resource load, energy and task execution time are known due its homogeneous nature. In order to improve the cloud consumer's satisfaction, a novel Locality-Load-Prediction Aware Multi-objective Task Scheduling (LLPAMTS) algorithm is proposed to eventually distribute the tasks according to dynamic nature of cloud virtual machines. **Findings:** Proposed LLPAMTS algorithm will effectively schedule the tasks in an optimized manner by VM Scheduler component. This scheduling algorithm exploits the various monitoring parameters like locality, load and prediction parameters. It outperforms the existing deadline, load and energy aware scheduling algorithms in terms of task transfer time, task waiting time, task execution time, and task completion time. **Applications/Improvements:** The proposed LLPAMTS algorithm provides an average of 5 to 10% less task completion time compared to the existing deadline, load and energy aware scheduling algorithms.

Keywords: Cloud Environment, Heterogeneous Cloud, Locality-Load-Prediction Aware Scheduling, Multi-Objective, Task Scheduling

1. Introduction

Cloud computing offers the service like Infrastructure as a Service, Platform as a Service, Software as a Service, and Storage as a Service over the internet in the basis of pay-for-use utility model[1,2]. These services are offered to the user based on the Service Level Agreement (SLA) signed between service consumer and service provider. SLA is the contract made between provider and consumer to promise the vision of cloud computing Quality of Service (QoS) goals which clearly states the pricing and violation terms of cloud service delivery models[3]. Further, the SLA can be classified into provider predefined (static) SLA and negotiated (dynamic) SLA. In static case, a generic SLA template is provided to all the consumers but in the dynamic case, the consumer and provider undergo a series of negotiation processes to achieve a mutually agreed SLA template. Current cloud management system focuses on dynamic SLA to maximizing their revenue and to provide classified service provisioning for different types of consumers[4]. So, effective cloud management without violating SLA is identified as a major challenging issue in the today's SLA oriented cloud management system.

In order to maximize the cloud provider's revenue, an effective cloud management system is needed with appropriate task scheduling algorithm which can overcome the SLA violations happens during resource failure. In existing research work, runtime estimation

** Author for correspondence*

aware task scheduling algorithm is used to handle deadline based tasks by estimating the execution time of all waiting tasks present in resource queue[5]. In contrast, load aware scheduling algorithm and task prioritization mechanism is used in hierarchical manner for giving priority over the SLA or deadline based tasks[6]. This type of prioritization will overcome the SLA violations that occur due to resource failure by satisfying customer's business through quick task completion within stipulated time. It also improves the cloud management system throughput by uniformly migrating and distributing the tasks from overloaded and faulty resource to available dedicated resource. A novel dynamic forecast scheduling algorithm is used for future consumption forecasting by analyzing the historical memory consumption of virtual machine[7]. This approach will save the energy consumption by minimizing the number of physical machines running in the cloud environment.

All the above algorithms were developed in the context of handling any one the objectives like deadline, load, prediction, energy and so on. To further improve the performance of cloud management system, a multi-objective task scheduling algorithms are needed for handling the real time task scheduling problems. This can be achieved by heuristically combining some task scheduling objectives according to their problem requirement. Therefore, different combinations of objectives were used by the researchers for maximizing either throughput or consumer satisfactions in the cloud management system. So, the proposed research work focus on identifying a novel heuristic combinations of objective functions that can further maximize both throughput as well as the consumer satisfactions without any SLA violations.

According to the analysis of emerging research trends and past literature studies defined in the field of cloud task scheduling algorithm[8-13]. This research work gives an extensive form of new classifications in the cloud task scheduling schemes as shown in Figure 1. According to the recent research works, this figure shows the classification of cloud task scheduling in the context of credit[14], cost[15,16], deadline[17,18], fault tolerance[19,20], energy[21-24], normalization[25], latency[26,27], load[28,29], randomness[30], heuristic[31-34], optimization[35,36], prediction[37], scalability[38,39], QoS[40-43], SLA[44], trust[45,46] and utilization[47]. This new classification triggers to understand the existing schemes and helps to identify the emerging research issues and innovative techniques of handling the cloud task scheduling problems. It is evident from this literature study, much research works are not available in the optimization context of multi-objective task scheduling. Therefore, the proposed research work focuses in the design and development of novel multi-objective task scheduling algorithm.

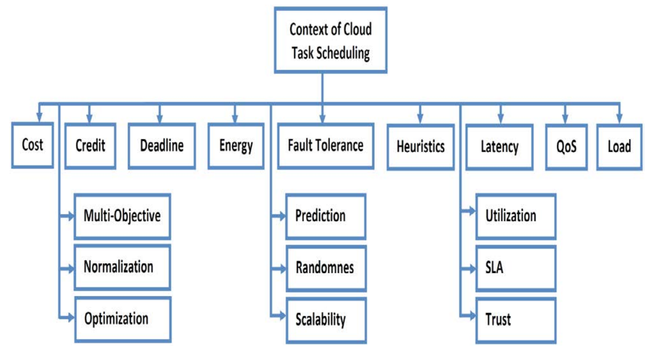

Figure 1. Taxonomy of Cloud Task Scheduling.

Current state of the art research work in the multi-objective task scheduling algorithm uses the objective functions like deadline, load, prediction, energy, cost, and other metrics. This kind of objective functions can improve the performance of scheduling algorithm by minimizing the task response time, task completion time, task energy consumption and maximizes the resource utilization and throughput more effectively without any SLA violation. In existing research work, the objectives like execution time, cost, and bandwidth of user task were considered to maximize the throughput and minimize the cost[48]. Next, the energy and processing time objectives were used by the researchers for maximizing provider's revenue and minimizing power consumptions[49]. Alternatively, the novel multi-objective evolutionary algorithms were proposed using response time and makespan objectives for minimizing cost and maximizing QoS[50]. From the literature study, this research work identifies that there is a need for novel multi-objective task scheduling algorithm with the heuristic combination of objectives to solve the real time cloud task scheduling problems.

The remainder of the paper is structured as follows: In Section 2, problem definition is described with the origination of multi-objective task scheduling problem. Section 3 describes about the proposed architecture of cloud scheduling mechanism with the novel Locality-Load-Prediction Aware Multi-objective Task Scheduling algorithm. In section 4, experimental evaluations are carried out by the comparison of various scheduling results and discussions. Final section will gives the conclusion and future directions of multi-objective task scheduling in the heterogeneous cloud environment.

2. Problem Definition

In cloud environment, the virtual machine resource instances available in the data centers are geographically distributed and dynamic in nature. This situation in turn affects the application performance like execution time and response time of user tasks due to improper way of matching the task to resource scheduling process. In order to improve the overall throughput of cloud environment, an effective and efficient task scheduling algorithms are fundamentally important for improving the application performance of cloud provider's. Here locality, load and prediction factors are identified as the essential objective functions which can improve the performance of cloud provider's at various levels. Therefore, this research work focus to consider the heuristic combination of objectives like locality, load and prediction as a basic characteristic functions for task scheduling decision making problem in cloud data centers. The list of acronyms used in the multi-objective task scheduling problem formulation is described in Table 1.

Table 1. List of Acronyms

Symbol	Description
n	Number of task arrived at any time interval
m	Number of available VM resources
J	Set of task $J_1, J_2, \dots, J_n$ arrived at any time interval
R	Set of available VM resource $R_1, R_2, \dots, R_m$
d_i	Scheduling decision taken at time period T_i
$\text{Ụ}(R_i)$	Utilization of resource R_i
$\hat{\text{U}}(R_i)$	Utility value of the resource R_i
$TTT(J_i)$	J_i Task transfer time function
$TET(J_i)$	J_i Task execution time function
$TEC(J_i)$	J_i Task completion time function
$TEC(J_i)$	J_i Task energy consumption function
$LO(R_i)$	Locality function of resource R_i
$L(R_i)$	Load function of resource R_i
$P(R_i)$	Prediction function of resource R_i

To solve this research problem, a set of independent cloud tasks $J = \{J_1, J_2, \dots, J_n\}$ from different users are considered to map on set of heterogeneous cloud resources $R = \{R_1, R_2, \dots, R_m\}$. Assume, there are $'n'$ number of task arrives into cloud management system and the number of task arrives at each time period T denotes the task arrival rate λ. Then the task scheduling algorithm is initiated in the cloud request handler to make scheduling decision d_i over the time period T_i. Assume the task J_i contains M_i units of workload for its execution in virtual resource R_i at any time stamp T_i may leads to performance degradation such as maximization of resource utilization and throughput. This is maximized by the existing scheduler algorithm through the estimation of resource load and task priority during the scheduling decisions. To further maximize the resource utilization and throughput, resource locality estimation is planned to incorporate in the proposed research work. This can maximize the resource utilization, and minimize the task transfer time, task waiting time, task execution time, and task completion time as defined in equation (1) and (2) respectively.

$$Max_{i\in(1,m)}\ \text{Ụ}(R_i) \tag{1}$$

$$Min_{i\in(1,n)}\ TTT(J_i)\ ||\ TWT(J_i)\ ||\ TET(J_i)||\ TCT(J_i) \tag{2}$$

This maximization objective can be achieved by choosing the minimum utility value of the resource R_i as shown in equation (3). This utility value of the resource $\hat{\text{U}}(R_i)$ can be represented as the multi-objective task scheduling optimization problem by making the heuristic combination of all the objectives like locality, load and prediction functions as defined in equation (4).

$$Max_{i\in(1,m)}\ \text{Ụ}(R_i) = Min\ [\hat{\text{U}}(R_1), \hat{\text{U}}(R_2), \dots, \hat{\text{U}}(R_m)] \tag{3}$$

$$\hat{\text{U}}(R_i) = W_{lo} * LO(R_i) + W_l * L(R_i) + W_p * P(R_i) \tag{4}$$

Were W_{lo}, W_l, and W_p denotes the weight assigned to locality, load and prediction function of the resource R_i respectively such that $W_{lo} + W_l + W_p = 1$. The locality function of the resource R_i can be described as show in equation (5). Here, $Task\_Read\_From\_User$ denotes the time taken to read the task J_i from user, $Task\_Written\_To\_VM$ denotes the time taken to write the task J_i to the virtual machine, and $Transfer\_Duration(J_i)$ denotes the task transfer time from user environment to virtual machine environment.

$$LO(R_i) = Min_{R_i}\left(\frac{Task\_Read\_From\_User + Task\_Written\_To\_VM}{Transfer\_Duration(J_i)}\right) \quad (5)$$

The load function of the resource R_i can be calculated by estimating the expected average task completion time of all waiting task as represented in equation (6). Here, the load value is represented in the normalized form as $0 \leq L(R_i) \leq 1$.

$$L(R_i) = \frac{\sum_{i=1}^{n} TCT(J_i)}{n} \quad (6)$$

Prediction function includes the time series and qualitative forecasting of resource R_i allocation in the distributed cloud environment. This function can forecast the future memory demands of cloud datacenter resource as a probability distribution. The predicted value of resource R_i memory consumption over the time series $T = \{t_1, t_2, \ldots, t_k\}$ can be characterized as show in equation (7).

$$P(R_i) = \{P_{R_i}(t_1), P_{R_i}(t_2), \ldots, P_{R_i}(t_k)\} \quad (7)$$

Based on the time series predicted, an average predicted value of memory consumption in the specific time period can be defined as auto regression model as expressed in equation (8).

$$P_{R_i}(t_i + 1) = \Phi_1 \times P_{R_i}(t_i) + \Phi_2 \times P_{R_i}(t_i) + \delta_{t_i+1} \quad (8)$$

Let the parameter Φ_1 and Φ_2 denotes error residuals which can be computed from the static data, δ_{t_i+1} is the vector refers to error residuals. Here, the value of $P(R_i)$ is normalized as $0 \leq P(R_i) \leq 1$.

The expected task transfer time of any task J_i is computed by equation (9).

$$TTT(J_i) = \sum_{l=1}^{p}\left(latency_l + \frac{J_i^{Size}}{bandwidth_l}\right) \quad (9)$$

Let p represents the number of link l to reach the available resource for execution, J_i^{Size} determines the size of task J_i in bytes, $latency_l$ and $bandwidth_l$ denotes the latency and bandwidth taken for each link l in the network route. Here, the locality value is represented as the normalized value as $0 \leq LO(R_i) \leq 1$. The average task waiting time of any task J_i in the resource R_i is computed by equation (10), where $n(J_i)$ denotes the number of task waiting in resource R_i.

$$TWT(J_i) = \frac{\sum_{i=1}^{n} StartTime(J_i) - SubmitTime(J_i)}{n(J_i)} \quad (10)$$

The expected task execution time of any task J_i is computed by equation (11).

$$TET(J_i) = CompletionTime(J_i) - StartTime(J_i) \quad (11)$$

Finally, the task completion time of any task J_i can be estimated by summing all the values of task transfer time, task waiting time, task execution time as shown in equation (12).

$$TCT(J_i) = TTT(J_i) + TWT(J_i) + TET(J_i) \quad (12)$$

3. Architecture SLA Based Cloud Management System

The conceptual architecture of SLA based cloud management system is shown in Figure 2. It consists of various components like request handler, VM provisioner, VM monitor, VM Scheduler and Resource Information Database. Cloud consumer submits the task to cloud management system along with their QoS requirements to be satisfied by the cloud providers. User tasks are received by the request handler component and follow the task scheduling activity by mapping the user tasks with appropriate resources. This mapping function is enabled by using the policy mapper and policy selector functions present in the VM provisioner component. Based on the user task requirement, the VMs are provisioned to the cloud users from the set of resources available in the under cloud environment. The VMs provisioned by the VM provisioner are scheduled in the underlying cloud resources by using the VM scheduler component. This component consists of dispatcher function which will dispatch the VMs for user task execution. It also monitors the VMs, and monitors the QoS and SLA metrics of user task through the VM monitor, QoS monitor, and SLA monitor functions respectively.

VM scheduler performs the resource scheduling

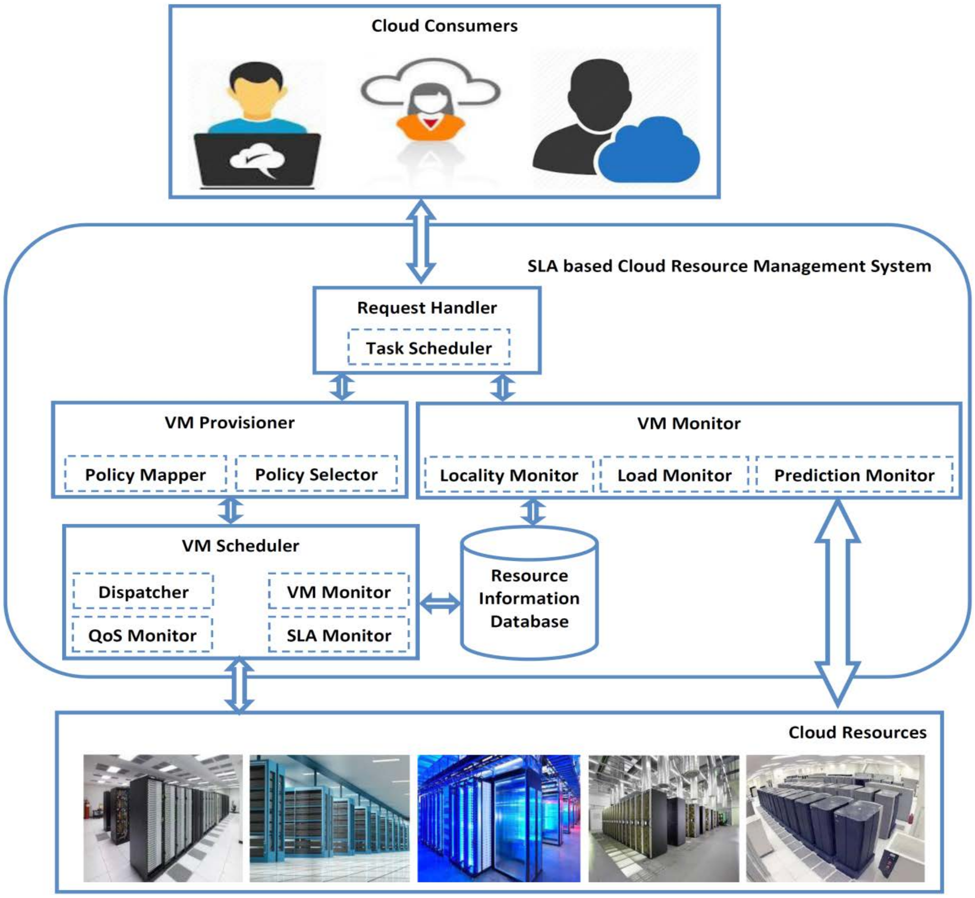

Figure 2. Architecture of SLA based Cloud Management System.

based on the updated information available in the resource information database. This database is frequently updated through various functions like locality monitor, load monitor and prediction monitor available in the VM monitor component which in turn get updated by the trigger function running in all cloud virtual machine resources. The hierarchical way of task-to-resource scheduling in the cloud management system can be focused from different perspectives like locality, load, prediction, security and so on. These sequences of decisions and computational operations used in the task scheduler component are generalized into Locality-Load-Prediction Aware Multi-objective Task Scheduling pseudo-code as shown in Algorithm 1.

4. Experimental Results and Discussion

The experimental evaluation of proposed LLPAMTS algorithm was implemented using java framework in CloudSim tool[51]. This experimental simulation consists of 30 datacenters and 100-to-300 real time tasks which are scheduled using the proposed LLPAMTS and other existing scheduling algorithms. Each task can take minimum of 60 seconds to execute the job in the allotted virtual machines present over the datacenter. Assume each task has different deadline to execute its operations in the virtual machines. The experimental results are observed with respect to task completion time by varying

Algorithm 1 Locality-Load-Prediction Aware Multi-objective Task Scheduling

Begin
Initialize the list of user tasks as $T = \{t_1, t_2, \dots, t_n\}$
Initialize the list of VM resources as $R = \{R_1, R_2, \dots, R_m\}$
for all $t_i \in T$
  Get the list of available resources $R_i \in R$ in cloud
  for all $R_i \in R$ **do**
    Get the list of tasks waiting in R_i resources queue
    Estimate task transfer time as $TTT(t_i)$
    Estimate the task waiting time as $TWT(J_i) = \dfrac{\sum_{i=1}^{n} StartTime(J_i) - SubmitTime(J_i)}{n(J_i)}$
    Estimate the task execution time as $TET(t_i) = CompletionTime(J_i) - StartTime(J_i)$
    Compute task completion time as $TCT(t_i) = TTT(t_i) + TWT(t_i) + TET(t_i)$
    if $[TCT(t_i)\ in\ R_i] < ECT(t_i)$
      Add R_i to Eligible Resource list ER
      for all $R_k \in ER$ **do**
      Estimate the locality function as $LO(R_k) \in ER$
      Estimate the load function as $L(R_k) \in ER$
      Estimate the prediction function as $P(R_k) \in ER$
      Compute resource utilization as $\underset{\cdot}{U}(R_k) = Min\left[\hat{U}(R_1), \hat{U}(R_1), \dots, \hat{U}(R_m)\right]$
      Estimate R_i utility value as $\hat{U}(R_k) = W_{lo} * LO(R_k) + W_l * L(R_k) + W_p * P(R_k)$
      if $\hat{U}(R_k) < \hat{U}(R_{k+1})$
        Prefer d_i to submit the task t_i to resource R_k
      else
        Prefer d_i to submit the task t_i to resource R_{k+1}
      end if
    else
      Reject the task t_i
    end if
  end for
end for
  End Algorithm

the number of tasks like 100, 200, and 300 tasks as shown in Figure 3. Then the performance of the proposed LLPAMTS algorithms is compared with the existing deadline aware scheduling, load aware scheduling and energy aware scheduling algorithms as shown in Figure 3.

It is clear from the performance graph, the total completion time of task is minimized in the proposed LLPAMTS algorithms while comparing to the existing algorithms. This minimization is achieved due the consideration of equal preferences to multiple objectives of task scheduling. In addition, this research work can be extended with addition multi-objective parameters for further minimization of total completion time of task running in virtual machines.

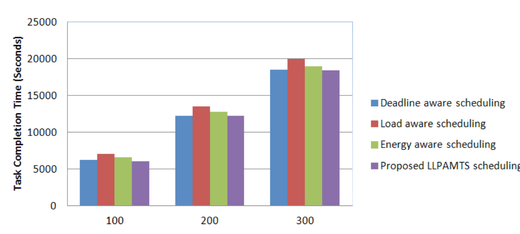

Figure 3. Performance of task scheduling algorithms with respect to total completion time of tasks.

5. Conclusion and Future Works

Proposed Locality-Load-Prediction Aware Multi-objective Task Scheduling algorithm for dynamic cloud environment is an optimal task scheduling algorithm which provides minimum task transfer time, task waiting time, task execution time, and task completion time than the existing algorithms. Thus the experimental results shows that the proposed LLPAMTS algorithm outperforms the existing deadline aware scheduling, load aware scheduling and energy aware scheduling algorithms in terms of total completion time of task. Further, this research work can be extended with additional objectives like bandwidth, foreground and background VM load balancing, and other QoS parameters to effectively reduce the energy and make-span through the implementation of robust cloud task scheduling algorithm.

6. References

1. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems. 2009; 25(6):599-616. Available from: Crossref.
2. Brindha T, Shaji RS. An Efficient Framework for Providing Secured Transaction of Data in Cloud Environment. Indian Journal of Science and Technology. 2016 Nov; 9(44):1-6. Available from: Crossref.
3. Emeakaroha VC, Brandic I, Maurer M, Dustdar S. Cloud resource provisioning and SLA enforcement via LoM2HiS framework. Concurrency and Computation: Practice and Experience. 2013; 25(10):1462-81. Available from: Crossref.
4. Macias M, Guitart J. SLA negotiation and enforcement policies for revenue maximization and client classification in cloud providers. Future Generation Computer Systems. 2014; 41:19-31. Available from: Crossref.
5. Rajkumar R, Mala T. Runtime Estimation Aware Scheduling Algorithm for Handling Deadline Based Tasks in Grid Environment. Advances in Intelligent and Soft Computing. 2012; 132:555-62. Available from: Crossref.
6. Rajkumar R, Mala T. Achieving Service Level Agreement in Cloud Environment using Task Prioritization in Hierarchical Scheduling. Advances in Intelligent and Soft Computing. 2012; 132:547-54. Available from: Crossref.
7. Zhuo T, Yanqing M, Kenli L, Keqin L. Dynamic forecast scheduling algorithm for virtual machine placement in cloud computing environment. Journal of Supercomputing. 2014; 70:1279-96.
8. Raju IRK, Varma PS, Sundari MVR, Moses GJ. Deadline Aware Two Stage Scheduling Algorithm in Cloud Computing. Indian Journal of Science and Technology. 2016 Jan; 9(4):1-10.
9. Diallo L, Hashim AHA, Olanrewaju RF, Islam S, Zarir AA. Two Objectives Big Data task Scheduling using Swarm Intelligence in Cloud Computing. Indian Journal of Science and Technology. 2016 Jul; 9(28):1-10. Available from: Crossref.
10. Usman MJ, Ismail AS, Chizari H, Gital AY, Aliyu A. A Conceptual Framework for Realizing Energy Efficient Resource Allocation in Cloud Data Centre. Indian Journal of Science and Technology. 2016; 9(46):1-8.
11. Komarasamy D, Muthuswamy V. A Novel Approach for Dynamic Load Balancing with Effective Bin Packing and VM Reconfiguration in Cloud. Indian Journal of Science and Technology. 2016; 9(11):1-6. Available from: Crossref.
12. Syed HHM, Muhammad SAL, Yahaya C, Shafi'Ima. Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities. Journal of Network and Computer Applications. 2016; 68:173-200. Available from: Crossref.
13. Akilandeswari P, Srimathi H, Srimathi H. Survey and analysis on Task scheduling in Cloud environment. Indian Journal of Science and Technology. 2016 Oct; 9(37):1-6. Available from: Crossref.
14. Antony T, Krishnalal G, Jagathy RVP. Credit Based Scheduling Algorithm in Cloud Computing Environment. Procedia Computer Science. 2015; 46:913-20. Available from: Crossref.
15. Rui Z, Kui W, Minming L, Jianping W. Online Resource Scheduling Under Concave Pricing for Cloud Computing. IEEE Transactions on Parallel and Distributed Systems. 2016; 27(4):1131-45. Available from: Crossref.
16. Nuttapong N, Booncharoen S, Tiranee A. Cost optimal scheduling in IaaS for dependent workload with particle swarm optimization. Journal of Supercomputing. 2014; 68:1579-603.
17. Florin P, Ciprian D, Valentin C, Nik B, Fatos X, Leonard B. Deadline scheduling for a periodic tasks in inter-Cloud environments: a new approach to resource management. Journal of Supercomputing. 2015; 71:1754-65. Available from: Crossref.
18. Sanjaya KP, Prasanta KJ. Efficient task scheduling algorithms for heterogeneous multi-cloud environment. Journal of Supercomputing. 2015; 71:1505-33. Available from: https://doi.org/10.1007/s11227-014-1376-6.
19. Shafi'Ima, Muhammad SAL, Syed HHM, Mohammed A. Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm. Neural Computing and Applications. 2016; p. 1-15.
20. Sook Kyong C, Kwang Sik C, Heonchang Y. Fault tolerance and QoS scheduling using CAN in mobile social cloud computing. Cluster Computer. 2014; 17:911-26. Available from: Crossref.

21. Xiaoli W, Yuping W, Yue C. An energy-aware bi-level optimization model for multi-task scheduling problems under cloud computing. Soft Computer. 2016; 20:303-17. Available from: Crossref.
22. Tarandeep K, Inderveer C. Energy aware scheduling of deadline-constrained tasks in cloud computing. Cluster Computer. 2016; 19:679-98. Available from: Crossref.
23. Zhuo T, Ling Q, Zhenzhen C, Kenli L, Samee UK, Keqin L. An Energy-Efficient Task Scheduling Algorithm in DVFS-enabled Cloud Environment. Journal of Grid Computing. 2016; 14:55-74. Available from: Crossref.
24. Seyedmehdi H, Farshad K, Rashidaldin S. SEATS: smart energy-aware task scheduling in real-time cloud computing. Journal of Supercomputer. 2015; 71:45-66. Available from: Crossref.
25. Sanjaya KP, Prasanta KJ. Normalization-Based Task Scheduling Algorithms for Heterogeneous Multi-Cloud Environment. Information Systems Frontiers. 2016; p. 1-27.
26. Bei G, Jingzheng W, Yongji W, Samee UK. CIVSched: A Communication-Aware Inter-VM Scheduling Technique for Decreased Network Latency between Co-Located VMs. IEEE Transactions on Cloud Computing. 2014; 2(3):320-32. Available from: Crossref.
27. Dzmitry K, Johnatan EP, Andrei T, Pascal B, Samee UK, Albert YZ. CA-DAG: Modeling Communication-Aware Applications for Scheduling in Cloud Computing. Journal of Grid Computing. 2016; 14:23-39. Available from: Crossref.
28. Rajkumar R, Thamarai SS, Kannan G. Dynamic Load Balancer Algorithm for the Computational Grid Environment. Communications in Computer and Information Science. 2010; 101:223-7. Available from: Crossref.
29. Rajkumar R. Topology and Load Aware – Grid Scheduler for the Computational Grid Environment. India: Proceedings of IEEE International Conference on Communication Control and Computing Technologies. 2010; p. 431-36.
30. Zhiping P, Delong C, Jinglong Z, Qirui L, Bo X, Weiwei L. Random task scheduling scheme based on reinforcement learning in cloud computing. Cluster Computing. 2015; 18:1595-607. Available from: Crossref.
31. Chun-Wei T, Wei-Cheng H, Meng-Hsiu C, Ming-Chao C, Chu-Sing Y. A Hyper-Heuristic Scheduling Algorithm for Cloud. IEEE Transactions on Cloud Computing. 2014; 2(2):236-50. Available from: Crossref.
32. Keng-Mao C, Pang-Wei T, Chun-Wei T, Chu-Sing Y. A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing. Neural Computer and Application. 2015; 26:1297-309. Available from: Crossref.
33. Arash GD, Yalda A. HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. Cluster Computer. 2014; 17:129-37. Available from: Crossref.
34. Mohammad S, Saeed J, Saeid A, Nicola C. FUGE: A joint meta-heuristic approach to cloud task scheduling algorithm using fuzzy theory and a genetic method. Cluster Computer. 2015; 18:829-44. Available from: Crossref.
35. Fan Z, Junwei C, Keqin L, Samee UK, Kai H. Multi-objective scheduling of many tasks in cloud platforms. Future Generation Computer Systems. 2014; 37:309-20. Available from: Crossref.
36. Jinglian W, Bin G, Hong L, Shaohui L. Multidisciplinary approaches to artificial swarm intelligence for heterogeneous computing and cloud scheduling. Applied Intelligence. 2015; 43:662-75. Available from: Crossref.
37. Ellendula M, Thirumalaisamy R. Efficient Scheduling Algorithm for Cloud. Procedia Computer Science. 2015; 50:353-56. Available from: Crossref.
38. Yiming H, Anthony TC. Scalable Loop Self-Scheduling Schemes for Large-Scale Clusters and Cloud Systems. International Journal of Parallel Programming. 2016; p. 1-17.
39. Haohao Z, Su D, Hongbin H. Stability property of clouds and cooperative scheduling policies on multiple types of resources in cloud computing. Journal of Supercomputing. 2016; 72:2417-36. Available from: Crossref.
40. Rajinder S, Sandeep KS. Scheduling of big data applications on distributed cloud based on QoS parameters. Cluster Computer. 2015; 18:817-28. Crossref.
41. Esmail A, Azadeh A, Mostafa D, Michel GK, Mohsen S, Sayed VA. Kani: a QoS-aware hypervisor-level scheduler for cloud computing environments. Cluster Computer. 2016; 19:567-83. Available from: Crossref.
42. Andrei T, Luz L, Uwe S, Pascal B, Johnatan EP, Sergio N, Alexander YD. Online Bi-Objective Scheduling for IaaS Clouds Ensuring Quality of Service. Journal of Grid Computing. 2016; 14:5-22. Available from: Crossref.
43. Tamal A, Amit KD, Md AR, Ahmad A, Majed A, Mohammad MH. Quality of Service Aware Reliable Task Scheduling in Vehicular Cloud Computing. Mobile Networks and Applications. 2016; 21:482-93. Available from: Crossref.
44. Rajkumar R, Mala T. SLAOCMS: A Layered Architecture of SLA Oriented Cloud Management System for Achieving Agreement during Resource Failure. Advances in Intelligent Systems and Computing. 2014; 236:801-09. Available from: Crossref.
45. Navendu J, Ishai M, Joseph SN, Jonathan Y. A Truthful Mechanism for Value-Based Scheduling in Cloud Computing. Theory of Computing Systems. 2014; 54:388-406. Available from: Crossref.
46. Wei W, Guosun Z. Bayesian Cognitive Model in Scheduling Algorithm for Data Intensive Computing. Journal of Grid Computing. 2012; 10:173-84. Available from: Crossref.
47. Sanjaya KP, Indrajeet G, Prasanta KJ. Allocation-Aware Task Scheduling for Heterogeneous Multi-Cloud Systems. Procedia Computer Science. 2015; 50:176-84. Available from: Crossref.
48. Atul VL, Dharmendra KY. Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization. Procedia Computer Science. 2015; 48:107-13. Crossref.
49. Jena RK. Multi objective Task Scheduling in Cloud Environment Using Nested PSO Framework. Procedia Computer Science. 2015; 57:1219-27. Available from: Crossref.

50. Fahimeh R, Jie L, Javid T, Farookh KH. Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments. World Wide Web. 2015; 18:1737-57. Available from: Crossref
51. Rodrigo NC, Rajiv R, Anton B, C'esar AFDR, Rajkumar B. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software Practice and Experience. 2011; 41:23-50. Available from: Crossref.