A Hybrid Method for Minimizing Makespan in Cloud Environment

Ayushi Agrawal[1], Ankur Mudgal

Computer Science Department
SRIST, RGPV-Bhopal, MP, India.

Abstract— **In cloud environment, the aim of using optimal resources can be achieved using a load-balancing technique. The load-balancing technique assigns a set of requests into a set of resources for distributing the load. It is one of the significant issues in cloud computing and known as an NP-hard problem. Therefore, many nature-inspired meta-heuristic techniques are proposed to provide high efficiency. However, despite the importance of the nature-inspired meta-heuristic techniques for solving the problem of the load-balancing in the cloud environment, there is not a complete and detailed paper about reviewing and studying the main important issues in this domain. Therefore, this paper presents comprehensive coverage of the nature-inspired meta-heuristic techniques applied in the area of the cloud load-balancing. In addition, to solve the load-balancing problem in the cloud environments, Hybrid Max-min algorithms have been proposed and their significant challenges are considered for proposing the techniques that are more effective in the cloud computing environment.**

Keywords— **Load balancing, Cloud computing, Max-min, Nature Inspired, Response Time, Cloud Analyst.**

I. INTRODUCTION

The basic idea behind the resource provisioning is to detect and select the best resources for users based upon upcoming application request (demands), so that upcoming demand can get an optimal resource i.e. Number of resources needed to serve the application should be minimum to maintain a desirable level of service quality (minimum execution time and maximum throughput). Resource provisioning map upcoming request with running virtual machines so that user gets the services in minimum cost and time while service provider get the maximum profit without affecting the violation of SLA [1]. Majorly there are three resource provisioning techniques available for the cloud users as shown in Fig 1.

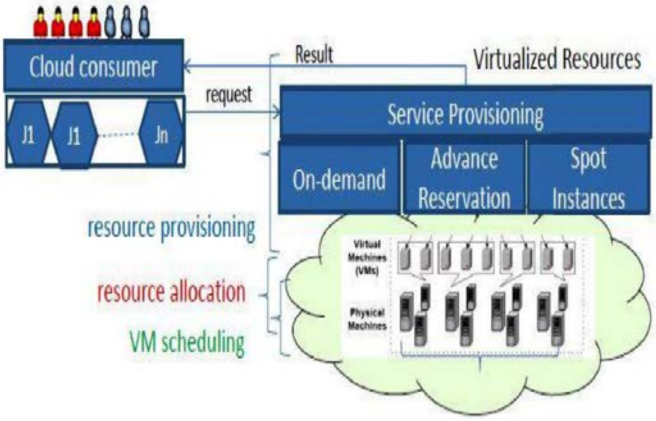

Fig.1: Resource provisioning plans.

A. On-demand provisioning

It is an intermediate level plan that allows users to pay per hour basis based upon the resources being used. If the demand for the cloud resources at a given time t exceeds the reserved value, then additional resources are required for on-demand resource provisioning. The under provisioning problem can be solved by provisioning more resources at higher cost with on-demand plan. Generally, on demand additional resources are allocated to the users at higher cost than advanced reservation resources.

B. Advanced reservation

This is a long-term plan that allows the users to reserve the resources in advance for a specific time period. This technique is very useful in federated cloud as well as elastic compute cloud (EC2). There are some drawbacks of this technique, such as prediction of future demand and prices of cloud resources is a challenging issue in this scheme. Overprovisioning and under-provisioning type of problem also occur in this technique.

C. Spot instances

It is a short-term plan that allows customers to bid on unused resources. Spot instances are Amazon's third plan that offer unused resources at a much lower cost in comparison with on-demand and advanced reservation. Major cloud service providers (AWS, Google, and Azure) provide the environment (facility) to use this scheme. Resources price rate vary frequently in spot instances based on supply and demand, this is the main limitation of spot instances scheme.

D. Advantages of cloud resource provisioning

Favours provided by cloud resource provisioning are mentioned below [2].

- Response time and makespan time of upcoming workload is reduced by efficient resource provisioning techniques.
- Better resource utilization can reduce the problem of Overprovisioning and under-provisioning.
- If virtual machine start-up delay is less, then it provides better resource provisioning in cloud environment.
- Effective cloud resource provisioning algorithm increases the robustness as well as fault tolerance capability.
- Resource provisioning algorithm reduces power consumption without affecting SLA violation.

1.2. Resource scheduling

Scheduling is the way to determine, which activity should be performed based upon the required quality of service (QoS) parameters. Scheduling is responsible to select optimal virtual machines for execution of tasks using either heuristic or meta-heuristic algorithm and responsible to examine that QoS constraint are met. Resource scheduling can be done in two ways; first one is on demand scheduling in which cloud service provider provides the resources quickly to random workload. This approach has a problem of unequal distribution of workload i.e. there is possibility of executing more tasks at a single virtual machine (VM), therefore performance start to degrade, and over provisioning type of problem can occur. Second is long term reservation in which large numbers of virtual machines are in ideal condition due to which under provisioning type of problem occurs. Over provisioning and under provisioning type of problem increase the cost of services due to unnecessary wastage of resources and time. To handle these types of problems, we need an efficient resource provisioning algorithm that analyze and schedule the upcoming workload in an efficient way.

Objective of resource provisioning with scheduling (RPS) is to provision the virtual machines to users without violation of SLA and fulfill the users demand. Also, to understand the expectation and requirement of the cloud users at the starting based on upcoming workload (applications). Service level agreement (SLA) commitment is defined between users and service provider after workload is properly analyzed. Fitness function (FFQoS) is calculated based upon the required QoS parameters for each workload and it is compared with the value calculated without considering QoS parameters (FFnon_QoS). We check the condition if value of FFQoS is less than the value of FFnon_QoS then it will provision; otherwise it analyses the workload again after resubmission of SLA by the cloud consumer through re-negotiation. If resource provisioning is completed successfully then scheduling algorithm is chosen to process the tasks in specified budget and deadline with the help of scheduler. Before allocation of workload or tasks at the virtual machines (resources), cloud running resources are monitored and load is calculated at each resource. If any virtual machine is in over utilized phase, then task is not allocated to such type of resources. Further upcoming workload is map with the available resources and checks the condition that running virtual machine is enough or not to execute the workload is checked. If running’s resources are not enough then the resources are increased using the horizontal scalability concept otherwise allocated to the workload and the required QoS parameters are calculated. We have collected and reviewed various research paper (year wise) based upon the concept of resource provisioning and scheduling in cloud, most of them belong to scheduling (task scheduling, resource allocation, load balancing etc.) techniques. There are various types of scheduling algorithm in cloud computing based upon: static and dynamic, online v/s batch mode, preemptive and non-preemptive scheduling algorithm etc. Scheduling algorithm can be categorized in two parts: static and dynamic scheduling.

1.2.1. Static scheduling algorithm

Static scheduling algorithms need the information about the task (length of task, number of tasks and deadline of tasks) and resource (node processing capacity, processing power, memory etc.) in advance. Static algorithms work well when variation in workload is very less and behavior of the system is not varying frequently, but load fluctuates instantaneously in cloud environment, so static algorithms are not a suitable choice for cloud computing. It is very easy to implement static algorithm, but these algorithms don't optimize the quality of service parameters and doesn't provide the good performance in real environment. Therefore, we need dynamic task scheduling algorithm for cloud environment. Example of static algorithms are first in first out (FIFO), round robin (RR), shortest job first (SJF) etc.

1.2.2. Dynamic task scheduling algorithm

Advance information about the task and node is not needed in dynamic algorithm but it needs to monitor the node continuously. These algorithms are more efficient and accurate for cloud environment because if any node is in overloaded condition then they transfers the task from overloaded node to under loaded node i.e., algorithm condition change frequently with load changes (increase or decrease) at a node. Example of dynamic algorithms are dynamic round robin, heterogeneous earliest finish time (HEFT), clustering based heterogeneous with duplication (CBHD), weighted least connection (WLC), particle swarm optimization (PSO), ant colony optimization (ACO) etc. Both the algorithms (static and dynamic) have their advantages and disadvantages as shown in Table 1.1.

Static Load Balancing	Dynamic Load Balancing
Need the advanced information about the upcoming jobs/ requests.	There is no need of advance information about the jobs and resources.
Scheduling decision is taken at compile time	Scheduling decision is taken at run time
Easy to implement i.e. complexity is low	It is not easy to implement, complexity is high
Static algorithms can't deliver optimal results for large computational problem.	Dynamic algorithm is useful for large computational problem.
Only traditional algorithm comes under static algorithm	Meta-heuristic algorithms come under dynamic algorithm
Static algorithms take more time to solve computational problem.	Dynamic algorithm solves the computational problem in less time.
It is difficult to find optimal result of multi-objective problem by static algorithms.	We can find the optimal results of multi-objective problem using dynamic algorithms.
Static algorithm works well when workload does not change frequently.	Dynamic algorithms work well when workload varies frequently
These algorithms do not monitor the node continuously	Dynamic algorithm monitors the node continuously either on event basis or time interval
Static algorithms do not balance the workload properly at the running virtual machines (node).	Dynamic algorithms balance the workload in efficient way at the nodes.

Dynamic task scheduling algorithm is recommended in the literature due to its highly dynamic workload and system behavior in the cloud environment. We can find the approximate solution of NP-hard problem using dynamic algorithm. Further two techniques of scheduling can be followed as: online and offline mode scheduling, preemptive and non-preemptive scheduling.

II. Related Work

2.1 Classification of load balancing Algorithm

In this section the existing load balancing algorithms, in cloud computing environment, are presented. The main focus of load balancing is the efficient utilization of the virtual machines and balancing the incoming requests to various virtual machines [3]. A large public cloud consists of many nodes which may spread over different geographical locations. Hence partitioning of cloud is done to manage a large cloud. In Cloud Analyst [4], three different algorithms are there for load balancing:

1) Round Robin algorithm is a very simple load balancing algorithm that allocates the new cloudlets on the available virtual machines in a circular order. This algorithm is very simple and can be implemented easily. It is static in nature i.e., prior information of user tasks and system resources is required.

2) Equally Spread Current Execution (ESCE) algorithm is active VM Load balancing algorithm. It distributes the load equally on each virtual machine in a cloud environment. ESCE VM Load Balancer maintains a list of virtual machines. It continuously checks the VM list and the task queue. If a VM is found free, then cloudlet request is allotted to that VM. Meanwhile, VM Load balancer also checks for the overloaded VM so as to reduce its load by moving some load to an idle or under loaded virtual machine.

3) Throttled load balancing algorithm determines the appropriate virtual machine which can handle the assigned load with greater ease. It is dynamic in nature as it maintains the present state of the all VMs in a cloud environment. If an appropriate VM is found, then throttled VM Load balancers accept the cloudlet request and allocate it to that virtual machine.

Ant Colony Optimization algorithm proposed by Linan Zhu et al., [5] is based on the behavior of ants to solve travelling sales man problem. Ants drop pheromone liquid substance while following a path for the search of food source. Other ants follow the path based on high pheromone strength. This concept is used by Ant colony algorithm to choose optimal path from source to destination that results into reduction of response time and distribute the work load of network. Karaboga, proposed a foraging behavior of honey bee swarm in 2005 [6]. Honey bee search for their food and informs other bees in beehive about the quantity and quality of food by performing waggle dance. There are three types of bees in the algorithm:

1) Scout bees: Arbitrarily Search for food source. Perform waggle dance to show the quality of food.

2) Employed bees: Collect all the information about food source and exchange the information obtained with onlooker bees.

3) Onlooker bees: Calculate the fitness value to find the best food source.

In respect of load balancing of incoming requests, tasks from overloaded machines are referred as honey bees, these tasks are transferred from overloaded machines to under loaded machines. Dynamic nature of the algorithm makes the changes in the status of the load to be reflected. Updated load on that particular machine is taken into account for other waiting tasks.

Zeng Zeng and Bharadwaj proposed a request balancing strategy, also known as optimal metadata replication and request balancing strategy [7]. This paper focuses on choosing appropriate metadata server when data retrieval request is initiated. Chosen metadata server gives command to raw data server for actual data retrieval. There may be replicated copies of the object on multiple metadata servers. For distributing the data to respective metadata servers, Zipf law of distribution is applied and aim at achieving minimum mean response time in a highly loaded cloud environment.

Nader Mohamed et al., designed a load balancing technique to handle download of large files called as dual direction technique [8]. This approach makes use of two Dual Direction FTP servers (DDFTP) to download a large single file. One of the server download half portion of a file from left to right while another server download another half from right to left direction. When both the servers find a common middle point of file, download operation terminates. A reliable, ordered delivery feature of TCP allows file blocks to be downloaded in a sequential manner hence reducing the overhead of coordination and data loss. This unique method of parallelizing the download enhances effective bandwidth utilization and reduces response time to give better performance.

Klaithem Al Nuaimi et al., [9] presented a simple algorithm to tackle the issue of balancing the load in giving Data as a Service (DaaS) in the Cloud. The algorithm has a basis of some prior approach for efficient data download in dual direction. Main objective of this paper is to solve the problem of the requirement of high volumes of storage when data is replicated in multiple cloud servers. Instead of storing full replicas of file, this algorithm gives a technique to store partial replicas of the data file on Cloud servers. An efficient method is provided, to download the information from various distributed servers and organise them in proper order.

There is lot of research in growing phase and some research challenges are still over looked like load balancing, resource provisioning, scheduling of applications, energy consumption etc. in cloud environment. Research related to resource scheduling is still in infancy phase and needs the improvement. Here, we will discuss some review papers based upon resource provisioning and scheduling techniques that are related to our research and useful for the present survey. M. Amiriand L. Khanli [10] presented a comprehensive survey regarding to the prediction of future demand of applications in different aspects. J. Zhang et al. [11] discussed about the resource provision techniques and algorithm design. Further this is observed that results of one phase can be used for another phase, finally it focuses at virtual machine migration, availability etc. parameters but both surveys do not discuss about the over provisioning and under-provisioning problem profoundly. S. Smanchat and K. Viriyapant [12] have divided the taxonomy of workflow scheduling into two parts: scheduling criteria and scheduling generation in cloud computing. Several resource scheduling algorithms have been discussed in this paper to improve the research as well as development of scheduling algorithms. P. Dave et al. [13] have presented a comparative study based upon different scheduling techniques to measure the QoS parameters reliability, scalability, resource utilization, throughput, execution cost etc. K. Radha et al. [14] have discussed problem of resource allocation (mapping upcoming requests with available resources) and management in the field of cloud. To solve the mentioned issues, capacity allocation algorithm is proposed for multi-tier system. C. Nandakumar and K. Ranjithprabhu [15] analyzed and compared the performance of various heuristic and meta-meta-heuristic algorithm based upon QoS parameters in cloud environment. Metrics satisfied by the existing algorithms are depicted in tabulated form but Most of the algorithms did not considered energy efficiency parameter that is important for green computing and this survey is limited only for some heuristic and meta-heuristic algorithm.

M. Kalra and S. Singh [16] have discussed five meta-heuristic scheduling algorithms for grid as well as cloud environment. They also discussed about the Pareto optimal theory to find the solution of the multi-objective problem. The main limitation of their survey, it discussed about the meta-heuristic techniques only. M. Masdari et al. [17] have presented a solution of workflow scheduling problem (NP-Complete) using meta-heuristic algorithm named as particle swarm optimization (PSO). They have discussed various variants of PSO based scheduling algorithm like multi-objective PSO (MOPSO), bi-objective PSO, jumping PSO, learning PSO etc. To improve QoS parameters such as reliability, time, cost etc. in the field of cloud computing. S. Madni et al. have reviewed 91 papers based upon scheduling algorithm in which 23 studies are based upon meta-heuristic techniques. They have discussed various issues like static and dynamic allocation strategy; QoS parameters based allocation techniques etc. based upon the meta-heuristic algorithm. Further comprehensive review and systematically comparison between meta-heuristic algorithms are helpful for future research direction. Cloud resource broker works as an intermediate between cloud service users and cloud service provider. It closely addresses the challenges of interoperability and portability in cloud computing which makes it difficult to share resources among the interconnected cloud. S. Chauhan et al. [18], presents a comprehensive survey about the existing resource broker techniques and discussed about the research challenges, advantages as well as limitations of resource broker. L. Bittencourtet al., [19], has discussed various scheduling model, techniques and challenges in computer system such as grid, cloud etc. Further, authors analyzed the scheduling problem that are represented by taxonomy and proposed future research direction in the field of cloud.

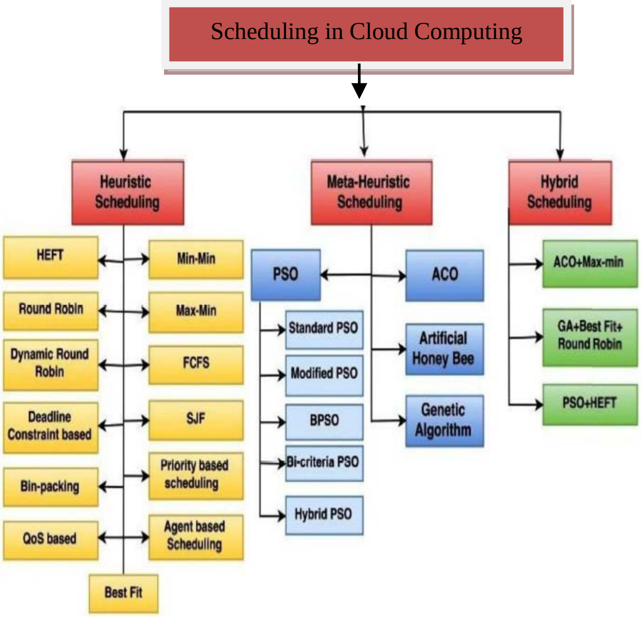

Fig. 2: Scheduling in Cloud Computing.

2.2 Min-min & max-min algorithm: To overcome the limitations of min-min algorithm, proposed improved load balancing algorithm that reduce the time and increase the resource utilization ratio. Priority of the user is also considered in this algorithm to assure guarantee of the service. Y. Mao et al. proposed min-max scheduling algorithm that assigned the larger tasks to best resources at the starting to improve the makespan time, response time as well as resource utilization ratio of user request but algorithm face the problem of overutilization and underutilization of resources and failed to improve required parameters. To cover the limitation of max-min algorithm, modified max-min algorithm has been proposed by O. Elzeki et al. that was based upon the concept of excepted execution time instead of complete processing time and improves makespan time but unable to improve other parameters like cost, energy etc. There are several improved versions of max-min algorithm that have been proposed to optimize the QoS parameters in cloud computing though proposed algorithms failed to improve the key performance indicator parameters. These are static algorithms and don't work well in dynamic environment (cloud computing).

III. Proposed Work

The proposed work focuses on the fundamental problems with resource allocation and cloudlets scheduling in cloud computing. When scheduling cloudlet, to achieve better performance by minimizing makespan and maximizing resource utilization, different cloudlets need to be executed in parallel by the available resources in order to meet consumer’s expectations. Min-Min algorithm is able to tackle these problems, but it produces high makespan and poor resource utilization when number of tasks with high completion time is more than tasks with low completion. In proposed work we develop improved Min-Min algorithm to resolve issues of higher makespan, poor resource utilization and unbalance load.

Proposed algorithm has the ability to assign both task with maximum and minimum completion time for the purpose of optimizing task scheduling and resource allocation in a dynamic cloud environment. The idea of task scheduling is to accomplish a high level of system throughput by dispatching a job to a resource that has the highest capacity to execute that job within a shorter period of time and also to match application needs by user demands with the available resources under a given cloud standard . In scheduling, it is the wish of every cloud service provider to ensure that every available resource is fully utilized to avoid resources being idle. Resource scheduling can be done in three main processes which include resource discovery and filtering, resource selection, and task submission. In resource discovery, a list of all the available resources in the cloud is discovered, arranged and listed by a cloud data center broker who presents it in the network system and collects the status of all the information related to the resource for scheduling. Resource selection, involves collecting information on the available resources and selecting the best set to match the application requirements for effective scheduling. Task submission is the final stage in scheduling whereby the selected task is submitted to the available or idle resources by the cloud datacenter broker for scheduling. Figure below represents proposed system with used components.

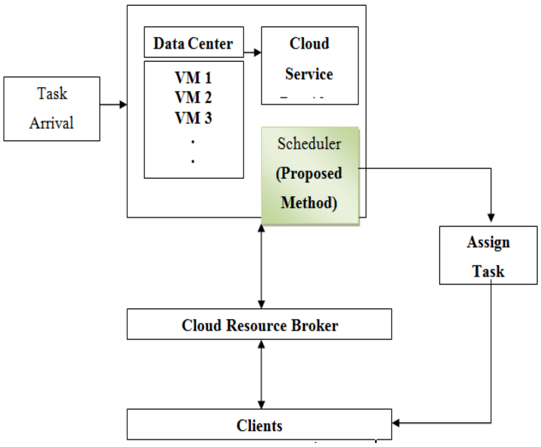

Fig. 3: System architecture.

3.2 Proposed Method for Scheduling within sub-groups: Proposed Modified Min-Min algorithm is multiple parameters based algorithm that considers both big cloudlets and small cloudlets. It is developed by combining both Max-Min and Min-Min algorithm. Proposed approach is able to select and assign either cloudlet with maximum or minimum execution time to a resource. For all submitted tasks in the Cloudlet list, it calculates completion time (CETij) based on following formula:

$CET_{ij} = et_{ij} + rt_j$…………………………………… (1)

Where et is expected time and rt is ready time.

For each cloudlet, system will determine the cloudlet with minimum completing time and maximum completion time. Then, it selects a cloudlet (c_i) from the Cloudlet List and compare with cloudlet having maximum completion time (MaxCET). If the c_i has maximum execution time, then it will assign the c_i to the resource that produces maximum completion time for execution, else it will assign cloudlet having minimum completion time (MinCET) to the resource that has the capability to execute it within a short time. The assigned task will be removed from the cloudlet List and the ready time will be updated.

The system stability improved by balancing the load across the available portioned virtualized resources. To have a better load balancing approach within sub-programs, the system requires a better scheduler. There are n input tasks and N number of virtual machines within each sub groups. The mapping of these n tasks to N VMs affects various system performance parameters. The finite set of user requests or tasks is {T1, T2, ..., Tn}.We have used Expected Time to Compute (ETC) matrix as tasks model on heterogeneous resource environment.

The value for ETCij is Li/Pj, where Li is the length of ith task in terms of Million instructions (MI), and Pj is the processing speed of jth VM in terms of MIPS. There are two important performance parameters in cloud system: (1) makespan (MS) and (2) energy consumption (EC). The execution time of different VMs in the cloud system is different. The maximum time taken by any VM to execute all input tasks by the system is referred to as makespan of the system. The minimal makespan results in a better balancing of the load. The execution time of jth VM (ETj) is based on the decision variable Xij, where:

Xij = 1 if Ti is allocated to VMj
0 if Ti is not allocated to VMj.

IV. Result And Analysis

There are different sizes virtual machines are used for this evaluation. Table 1 shows the performance comparison between the proposed algorithm, Existing algorithm and Round Robin algorithm with respect to makespan using different CSP. Figure below shows the graphical representation of the results where CU = 5 and CSP=50.

Results show that the performance of proposed algorithm is better.

Number of CU and CSP	RR	Existing	Proposed
50/6	79.34	79.01	77.06
20/3	150.23	149.95	147.50

Table 4.1: Overall Response Time Comparison of load balancing Algorithms.

CASE I: Result Chart for 50 Cloud users and six Cloud Service providers.

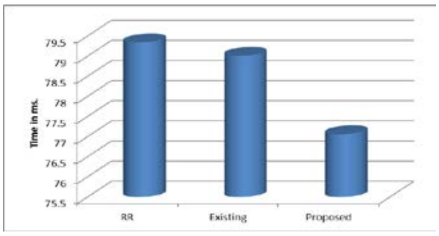

Fig. 4: Graph showing result of Proposed Algorithm.

CASE II: Result Chart for 20 Cloud users and 3 Cloud Service providers.

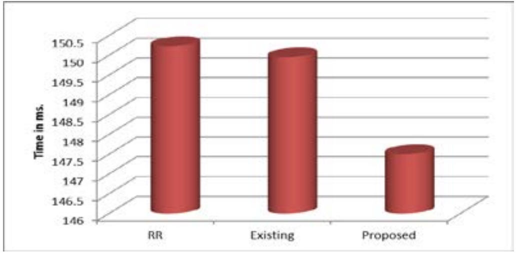

Fig. 5: Graph showing result of Proposed Algorithm.

4.1 Results & Evaluation

Results of above evaluations show that proposed algorithm completes user allocation with lower response time and higher performance as compared to existing Load Balancing algorithms. Performance of proposed algorithm is better than existing algorithms. Results shows that proposed algorithm behaves better in terms of response time after testing it with Cloud Analyst Simulator.

4.2 Conclusion

This thesis focuses on the problem of load balancing in cloud computing environment. In this work, a better approach has been designed for the load balancing of data requests received at the large scale cloud data centers. In such data centers thousands of servers are connected by interconnection network. For high priority tasks, servers that are currently having less number of tasks are selected so that such task gets executed faster. Proposed algorithm is compared with existing load balancing algorithms and it is observed that both response time and processing time are improved in the proposed strategy.

4.3 Future work

Load balancing is aimed to distribute the load properly among all nodes in order to achieve sufficient resource utilization without wastage of time and resource to improve overall performance of the system. In the future work, the algorithm will be extended to handle partial replicas, thereby reducing the storage space requirement in the cloud.

REFERENCES

[1] S. Javadi, B., et al., 2012. Hybrid cloud resource provisioning policy in the presence of resource failures. In: 4th International Conference on Cloud Computing Technology and Science. CloudCom, pp. 10–17.

[2] Singh, S., Chana, I., 2016. QoS-aware autonomic resource management in cloud computing: a systematic review. ACM Comput. Surv. 48 (3) article 42.

[3] Kavitha, K V and Suthan, Vinza V, “Dynamic Load Balancing in Cloud Based Multimedia System with Genetic Algorithm”, International Conference on Inventive Computation Technologies (ICICT), pp. 1–4, 2016.

[4] Wickremasinghe, Bhathiya and Calheiros, Rodrigo N and Buyya, Rajkumar “Cloud Analyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications”, 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 446-452, 2010.

[5] Zhu, Linan and Li, Qingshui and He, Lingna, “Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm”, IJCSI International Journal of Computer Science Issues, Vol. 9, No. 5, pp. 54–58, 2012.

[6] Karaboga, Dervis “An Idea Based on Honey Bee Swarm for Numerical Optimization”, Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, October 2005.

[7] Sheeja, YS and Jayalekshmi, S, “Cost Effective Load Balancing Based on Honey Bee Behavior in Cloud Environment”, First International Conference on Computational Systems and Communications (ICCSC), pp 214–219, 2014.

[8] Zeng, Zeng and Bharadwaj, Veeravalli, “Optimal Metadata Replications and Request Balancing Strategy on Cloud Data Centers”, Journal of Parallel and Distributed Computing, Elsevier, Vol. 74, No. 10, pp. 2934–2940, 2014.

[9] Al-Jaroodi, Jameela and Mohamed, Nader, “DDFTP: Dual-direction FTP”, Proceedings of the 2011, 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE Computer Society, pp. 504–513, 2011.

[10] Amiri, M., Khanli, L., 2017. Survey on prediction models of applications for resource provisioning in cloud. J. Netw. Comput. Appl. 82, 93–113.

[11] Zhang, J., et al., 2016. Resource provision algorithms in cloud computing. A survey” Journal of Network and Computer Applications 64, 23–42.

[12] Smanchat, S., Viriyapant, K., 2015. Taxonomies of workflow scheduling problem and techniques in the cloud. J. Netw. Comput. Appl. 52, 1–12.

[13] Dave, P., et al., 2014. Various job scheduling algorithms in cloud computing: a survey. International Conference on Information Communication and Embedded Systems, pp. 1–5.

[14] Radha, K., et al., 2014. Allocation of resources and scheduling in cloud computing with cloud migration. Int. J. Appl. Eng. Res. 9 (19), 5827–5837.

[15] Nandhakumar, C., Ranjithprabhu, K., 2015. Heuristic and meta-heuristic workflow scheduling algorithms in multi-cloud environments—a survey. In: International Conference on Advanced Computing and Communication Systems, pp. 1–5.

[16] Kalra, M., Singh, S., 2015. A review of metaheuristics scheduling techniques in cloud computing. Egyptian informatics journal 16 (3), 275–295.

[17] Masdari, M., et al., 2017. A survey of PSO-based scheduling algorithms in cloud computing. J. Netw. Syst. Manag. 25 (1), 122–158.

[18] Chauhan, S., et al., Aug. 2018. Brokering in interconnected cloud computing environments: a survey. J. Parallel Distrib. Comput.

[19] Bittencourt, L., et al., 2018. Scheduling in distributed systems: A cloud computing perspective. Comput. Sci. Rev. 30, 31–54.