Journal of Theoretical and Applied Information Technology
15th May 2017. Vol.95. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN OPTIMIZED MAX-MIN SCHEDULING ALGORITHM IN CLOUD COMPUTING

[1]J. KOK KONJAANG, [2]FAHRUL HAKIM AYOB, [3]ABDULLAH MUHAMMED
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
[1]jameskonjangkok@bpoly.edu.gh, [2]fahrulhakim@gmail.com, [3]abdullah@upm.edu.my

ABSTRACT

The fundamental problems with cloud computing environment are resource allocation and cloudlets scheduling. When scheduling cloudlets in cloud environment, different cloudlets needs to be executed simultaneously by the available resources in order to meet consumers' expectations and to achieve better performances by minimizing makespan and balancing load effectively. To achieve this, we proposed a new noble mechanism called Modified Max-Min (MMax-Min) algorithm, inspired from Max-Min algorithm. The proposed algorithm finds a cloudlet with maximum completion time and minimum completion time and assigns either of the cloudlets for execution according to the specifications for the purpose of boosting up cloud scheduling processes and increasing throughput. From the results of the simulation using CloudSim, it shows that our proposed approach is able to produce good quality solutions, producing good values of makespan and balancing load effectively as compared to the standard Max-Min, and Round Robin algorithms.

Keywords: *Cloud Computing; Max-Min; MMax-Min, Makespan and Load Balancing.*

1. INTRODUCTION

The advancement of technology and how businesses and organizations runs in this cloud era are clear evidence that cloud computing is the future platform for businesses and individual computing needs. Cloud computing has gained massive attention and popularity [1] in the IT sector because majority of organizations, companies and individuals are unable to buy resources (hardware and software) due to the high cost involved in getting them [2]. Cloud computing is increasingly becoming a preferable platform for organizations and individuals because; it has been able to lower the barriers of the high costs associated with providing the requisite software and hardware [3]. Cloud computing has converted storage, for example, into a 'pay as you go service'. This has enabled individuals and organizations not to require the purchase of their own resources (hardware and software) or paying for maintenance cost, [4].

The impact of cloud computing can be felt in our daily activities. It has impacted our lives in a very broad range of innovations which have completely transformed how computing services are produced, priced and delivered [5]. Organizations and individuals are driving on minimizing cost, meeting customers' demands for resource sharing and data availability on demand. All these are supported by cloud computing [6]. This groundbreaking sophisticated platform is increasingly becoming the customer/ user choice in the Information Technology (IT) market because of its multi-tenacity, scalability, availability, security, efficient storage and resource sharing platform. Given the large market segment and targeted growth of cloud computing, it is expected to become a market leader in the storage industry in the near future.

The paper focuses on the fundamental problems with cloud environments which include; resource allocation and cloudlets scheduling. When scheduling cloudlet in cloud computing environment, different cloudlets need to be executed in parallel by the available resources in order to meet consumers' expectations and to achieve better performance by minimizing makespan and maximizing resource utilization [7]. Max-Min attempts to tackle these problems, but produces high makespan and poor resource utilization when number of tasks with high completion time are more than tasks with low

Journal of Theoretical and Applied Information Technology
15th May 2017. Vol.95. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

completion (i.e. it gives priority to tasks with maximum completion time).

In this research, we addressed the issues of Max-Min algorithm which include the production of higher makespan; poor resource utilization and unbalance load by developing a suitable approach call Modified Max-Min (MMax-Min) which is capable of selecting and assigning either task with maximum completion time or task with minimum completion time to a resource for improved efficiency. In addition, we presented a detailed algorithm which is a modification of the traditional Max-Min algorithm that 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. For the purpose of benchmarking the results, the study also implemented the proposed mechanism in a cloudsim, the results were compared, which shows that MMax-Min is efficient in producing less makespan and can balance load effectively.

The remaining sections of the paper are organized as follows: Section 2 introduces related works; Section 3 presents research methodology; Section 4 presents the proposed algorithm; implementation, experiments and results analysis are presented in section 5; and finally Section 6 presents conclusions and future work.

1.1 TASK SCHEDULING IN CLOUD

Task scheduling is NP-hard problem [8] in cloud computing platforms where there are potentially large numbers of users from different environments and with different needs and demands on thousands or millions of virtual machines waiting for their turn to be served [9], [10]. 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 [11].

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 consumers/user demands with the available resources under a given cloud standard [12], [13].

Resource scheduling can be done in three main processes which include resource discovery and filtering, resource selection, and task submission [11], [7], [14]. 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 [15]. 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 as shown in Figure 1. Some of the algorithms used in scheduling cloudlets in cloud and grid computer environment are discussed below:

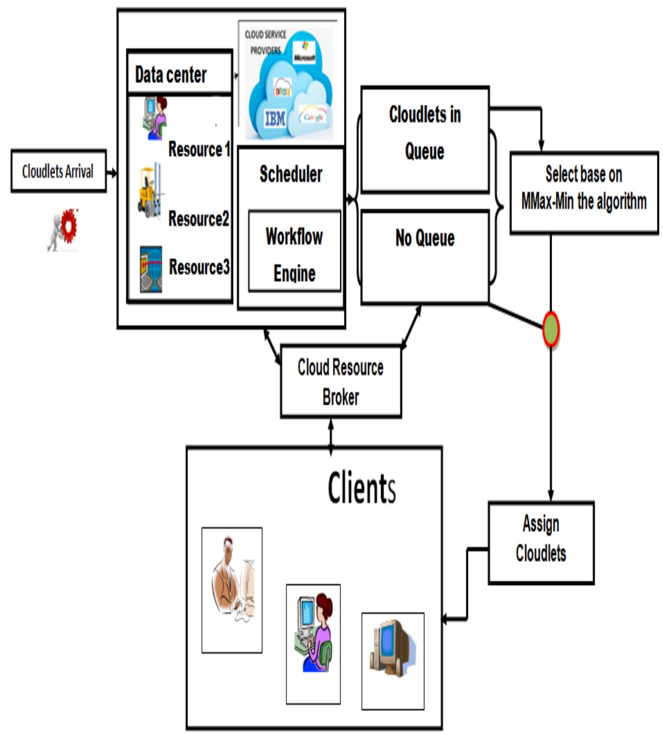

Figure 1: Task scheduling in cloud

1.1 Max-Min Algorithm

Max-Min is a resource allocation and scheduling algorithm used in cloud and in grid computing environments to minimize makespan, cost and maximize profit and resource utilization. This is done by selecting a task in the job list with the highest completion time on a resource that can execute it within a shorter period of time. The concern of this algorithm is to give priority to tasks with maximum completion time [16], [17] by executing them first before assigning other tasks that has minimum completion time [18]. However, the disadvantage of Max-Min algorithm is that, the execution of task with maximum completion time first might increase the total response time of the system [17] thus making it inefficient.

Previous researchers on scheduling techniques in cloud computing have endorsed Max-Min as the best method of scheduling resources in cloud and grid computing. These researchers have contributed tremendously by making Max-Min an efficient task scheduling algorithm. The efficiency of this algorithm has made cloud computing acceptable in both educational institutions and industries as a preferred platform for data storing and information dissemination.

1.2. Min-Min Algorithm

This algorithm works in the opposite direction of Max-Min algorithm. While Max-Min algorithm selects and assigns largest task to a resource with minimum completion time first before considering small tasks, Min-Min does the opposite by selecting and assigning small tasks to resources that can execute the job with a less makespan.

The purpose of Min-Min algorithm is to ensure that all tasks with minimum completion time are executed first in parallel and then same will be done to the long jobs after successfully executing short task [12]. However, its disadvantage is that, it sometimes leaves the long tasks unattended [19].

1.3. Round Robin

Round Robin is one of the most frequently used algorithms in resource allocation and task scheduling. In Round Robin, a dedicated time slot is allocated to each job waiting in a queue to be scheduled by the scheduler [20]. This means that, no task will be allocated to a resource for more than the allocated time and if a task is not able to complete scheduling within the allocated time, it will be preempted by another task and sent back in the queue to give way for other tasks.

1.4. First Come First Served (FCFS) Algorithm

Analysis made from various researchers on resource allocation and task scheduling algorithms has identified First-Come-First-Served (FCFS) as the simplest scheduling algorithm as far as task scheduling and resource allocation are concerned. FCFS is a non-preemptive scheduling algorithm that assigns tasks to resources based on their arrival time. It is non-preemptive in the sense that it does not release the resource to other tasks until it finishes scheduling [9]. One disadvantage of FCFS is that, its turnaround and response time is quite low because tasks are not preempted. The smaller tasks with the shortest execution time in the queue may have to wait for the larger task with large execution time ahead to finish [21].

2. RELATED WORKS

Cloud computing is becoming the most attractive platform for doing business because of its vast benefits which include multi-tenancy, scalability, security and on-demand applications that are made available to cloud users via the Internet. Task scheduling is however one fundamental concern in cloud and grid computing. Task scheduling is done by selecting a set of tasks from a list of tasks to be executed on available cloud resources that has the ability of executing the assigned task within a shortest period of time for effective and efficient utilization of computational power [22]. The main reason behind task scheduling in cloud and in grid computing is to increase system throughput by matching tasks with resources that has the capacity to execute them for effective cloud service delivering.

The past years have witnessed an exciting movement for cloud task scheduling and resource allocation. It has maintained its growth in the computer networks, telecommunication and the storage market, and improved its concentration with the introduction of innovative and groundbreaking scheduling algorithms by various researches such as, Max-Min [17], [16], [20], Min-Min [9], [23], Min-Max [24], Game Theoretic [25], [26], FCFS [27] and other algorithms have made resource allocation and task scheduling in the cloud more efficient. Though cloud computing has gone through significant reforms and transformations, however, the rapid growth in information and the increasing demand for cloud services in recent times calls for a proactive, efficient and optimal scheduling techniques to provide better and cost effective cloud services to meet the increasing demands and requirements of cloud consumers/users.

For example, in [23], Enhanced Load Balanced Min-Min algorithm was studied. In their work, they categorized scheduling as "Static/Dynamic scheduling, Centralized/ Distributed scheduling, Preemptive/Non-Preemptive scheduling, Cooperative scheduling, Immediate/Online mode scheduling and Batch/Offline mode scheduling". In their approach, selection and assignment of task is based on task with maximum completion time first not minimum completion time. The results indicate that their proposed algorithm was able to produce minimum makespan and utilized resource better. However, its drawback is that, the analysis of results was only done theoretically which may not give the true picture of the performance of the algorithm.

In [18] Max-Min scheduling algorithm was presented. The idea was to implement Max-Min algorithm to allow multiple dependent and independent tasks with maximum execution time to be executed simultaneously with other independent tasks. The results show that the proposed approach is able to schedule multiple tasks on multiple machines for a reduced execution time. Also [28] studied improved Max-Min algorithm for load balancing and resource utilization. In that that study, a Max-Min task scheduling algorithm for load balancing in elastic cloud was proposed. The mandate of the proposed algorithm was to maintain task status tables for estimating real-time load of virtual machines and expected completing time of all tasks. The proposed algorithm when experimented was able to improve its efficiency by reducing responds time and making good used of the resources.

Bhoi and Ramanuj, [29] presented an Enhanced Max-Min task scheduling algorithm in cloud computing in which they considered selecting average or nearest greater than average task instead of selecting and assigning large task first as always in the case of the traditional Max-Min scheduling algorithm which saw a substantial reduction in makespan and also balancing load effectively when the results was implemented. Weighted Round Robin is another approached presented in [30] with the mandate of allocating all incoming task to idle VMs. The idea of Weighted Round Robin algorithm was conceived through the traditional Round Robin. The proposed weighted Round Robin allocates task to resources using Round Robin fashion but based on the weight of the income request instead of the present load of the virtual machine as in the case of the traditional Round Robin. Though, limited parameters were used in the analysis of result, but not withstanding, the experimented results saw weighted round robin putting a better performances with regards to time.

Though these researchers have contributed immensely in making Max-Min algorithm more effective in terms of load balancing, makespan and resource utilization, however, the issue of priority given to tasks with maximum execution time in the expense of tasks with minimum execution time has not been fully address. Thus, our proposed approach aims at addressing it by introducing a new dimension to make Max-Min more effective and proactive in selection and scheduling both task with maximum and minimum execution time simultaneously without given priority to tasks with maximum execution time.

3. RESEARCH METHODOLOGY

This research seeks to compare and evaluate the performance of the proposed Modified Max-Min algorithm with other existing cloud scheduling algorithms such as Max-Min and Round Robin. CloudSim toolkit, a simulation environment in cloud computing was used to experiment and ascertain the performance of these algorithms in relation to task scheduling in the cloud. The concern of this research is to measure the makespan and load balancing using these algorithms to schedule a given task to a resource.

The proposed steps are defined herein;

Phase 1: Creating workflows:

Scientific workflows such as Inspiral, Montages, Cyber Shake and Sipht are first created. This forms the basis and represents the dataset for simulation purposes.

Phase 2: Creating data center

A datacenter is created to serve as a storage platform to run applications, process data, and store data. The datacenter is a pool of servers that contain a collection of applications/software that give room for multiple instances of virtual server to be operated and accessed by users via the Internet [31]. The creation of a datacenter is to ensure that resources are made available on cloud for onward distribution to meet the demand of cloud users effectively and profitably.

Phase 3: Creating Cloud data center broker

A cloud data center broker is created. In real life, a cloud data center broker may be an individual, group of individuals or a business entity that acts as an intermediary between the cloud service provider and the cloud service consumer/user. The cloud data center broker will be responsible for discovering resources, selecting the discovered resources and submitting the resources to the network system and collecting status information related to the resource for scheduling.

Phase 4: Resource Discovering and Selecting

- In resource discovering, a list of all available resources in the cloud is obtained and listed by cloud data center broker who presents it in the network system and collects status information related to the resource for scheduling.
- During resource selection, information is collected on the available resources and then the best resource is selected to match the application requirements [32] for effective scheduling.

Phase 5: Task submission

Task submission is the final stage in scheduling a task to a resource. The cloud data center broker is responsible for submitting the selected tasks to the available or idle resources for scheduling.

Phase 6: Applying Max-Min, Round Robin and the MMax-Min algorithms

After submitting a task to a resource, the proposed Modified Max-Min algorithm (MMax-Min) is used to schedule a task, and thereafter, the existing Max-Min and Round Robin algorithms are applied to benchmark the performance of the proposed MMax-Min algorithm.

Phase 7: Simulation development/comparison of result

The purpose of the simulation excise is to evaluate the performance of the proposed MMax-Min algorithm so as to benchmark it against the existing algorithms. To achieve this, the researchers simulated the proposed algorithm in a simulation environment known as cloudsim toolkit. The results were compared and benchmarked with the existing algorithms such as Max-Min and Round Robin algorithms.

All of the phases involve in this research method are simplified below:

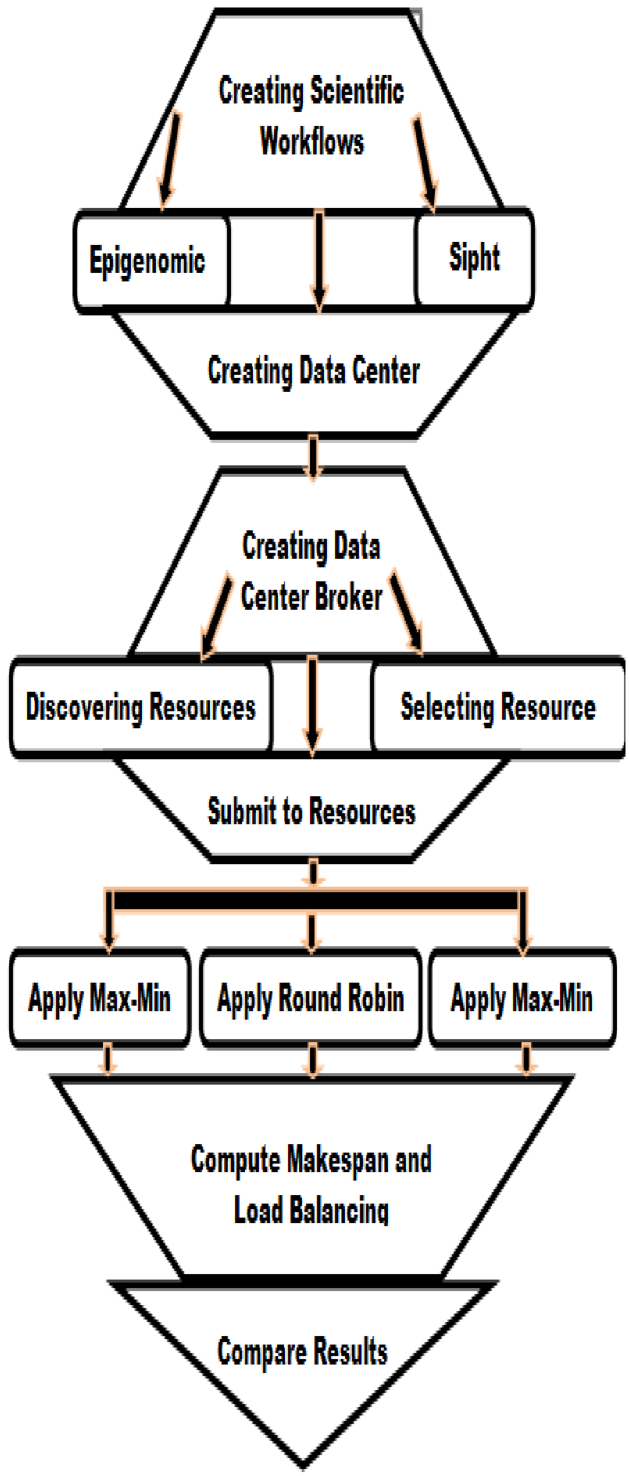

Figure 2: Simplified Research Method.

Journal of Theoretical and Applied Information Technology
15th May 2017. Vol.95. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. MODIFIED MAX-MIN (MMAX-MIN) ALGORITHM

Modified Max-Min algorithm works on the basis of multiple parameters that consider both large cloudlets and small cloudlets. It is the combination of both Max-Min and Min-Min algorithm, unlike the traditional Max-Min algorithm. For all submitted jobs in the CloudletList, it calculates completion time (CTij), $CT_{ij}) = et_{ij} + rt_j$; for each cloudlet to determine the cloudlet with minimum completing time and maximum completion time. Then, it selects a cloudlet (c_i) in the CloudletList and compare with MaxClt, if the c_i has maximum execution time, then it will assign the c_i to the resource that produces its minimum completion time for execution, else it will assign MinClt to the resource that has the capability to execute it within a short time. The assigned task will be removed from the cloudletList and the ready time will be updated. The flowchart and the pseudo Code of the Modified Max-Min Algorithm are displayed in Figure 2 and Figure 3.

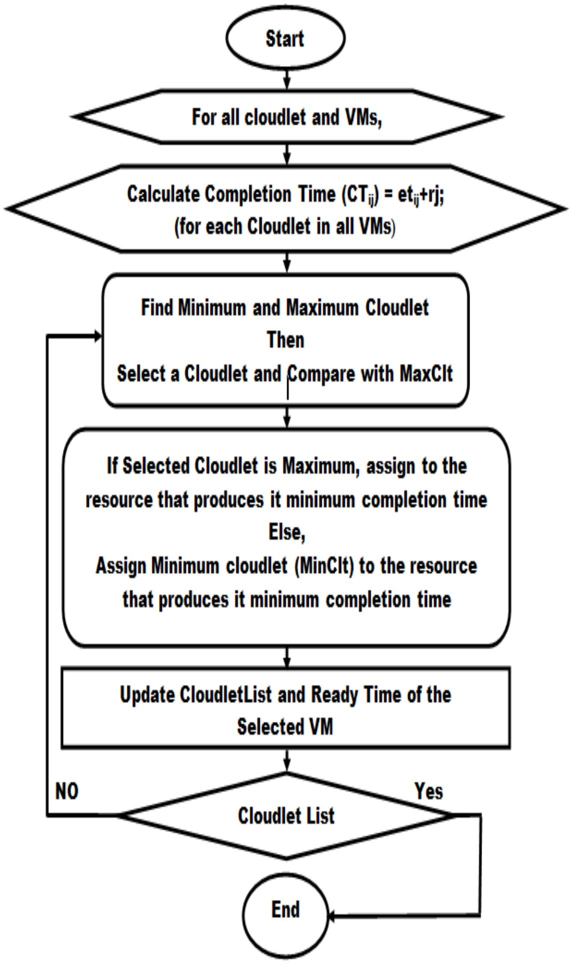

Figure 2: Flowchart - Modified Max-Min Algorithm

Pseudo Code of MMax-Min algorithm

1. While there are cloudlets in CloudletList
2. for all submitted cloudlets in the set; c_i
3. for all VMs; vm_j
   - Calculate completion time $(CT_{ij}) = et_{ij} + rt_j$; (for each cloudlet in all VMs)
   - find the cloudlet with minimum execution time (MinClt)
   - find the cloudlet with maximum execution time (MaxClt)
   - Select a cloudlet (c_i) in the CloudletList and compare with MaxClt
   - If the c_i has maximum execution time, assign the c_i to the resource that produce it minimum completion time for execution
2. Else assign MinClt to the resource that produce it minimum execution time
3. End if
4. Remove the cloudlet from the CloudletList
5. Update ready time (rt_j) of the selected vmR_j
6. Update ct_{ij} for all c_i
7. End While

FIGURE 3: Mmax-Min Algorithm

6. IMPLEMENTATION, EXPERIMENTS AND RESULTS ANALYSIS

For the purpose of comparison and benchmarking the proposed algorithm (MMax-Min) with other previously used cloud task scheduling algorithms such as Max-Min and Round Robin, a simulation environment called CloudSim toolkit was used. The simulation was run on Intel® core i3, 500GB HDD and 4GB Ram on 64 bit Windows 8 operating system. We considered two workflows, Epigenomics and Sipht which differ from each other in terms of cloudlet size. The procession speed of each cloudlet is measured in Million Instructions Per Second (MIPS). Table 1 and 2 gives detail descriptions of the four cases used in each of two workflows used for the simulation.

Table 1: Detail settings for Epigenomics workflow

Cases	Circumstances for using Epigenomics Cloudlets		NODE-DAX
1	EpiSma	Epigenomics with Small Cloudlets	24
2	EpiMe	Epigenomics with Median Cloudlets	46
3	EpiLa	Epigenomics with Large Cloudlets	100
4	EpiHe	Epigenomics with Heavy Cloudlets	997

Table 2: Detail settings for Sipht Workflow

cases	Circumstances for using Sipht Cloudlet		NODE-DAX
1	SiSma	Sipht with Small Cloudlets	30
2	SiMe	Sipht with Median Cloudlets	60
3	SiLa	Sipht with Large Cloudlets	100
4	SiHe	Sipht with Heavy Cloudlets	1000

Performance matrix:

In analyzing the performance of the proposed algorithm (MMax-Min), we used the following performance matrix to measure, compare and evaluate the performance of the proposed algorithm with the standard Max-Min and Round Robin.

i. Makespan: It is the total time taking to get a cloudlet scheduled. This is mostly used to measure the efficiency of scheduling algorithms in respect to time. It is measured by using equation 1 and 2.

Makespan = max(CT of j(ti, mj)) Eq 1

CT = the job end time – the job start time Eq 2

Figure 4 and Figure 5 illustrate the comparison of makespan between the four cases (EpiSma, EpiMe, EpiLa and EpiHe of Epigenomics and SiSma, SiMe, SiLa and SiHe of Sipht workflows). It is clear from the graphs that at every execution of cloudlets, the makespans of the proposed algorithm remains lower than Max-Min and Round Robin. Moreover, the comparison also shows that, Max-min algorithm has produced good values of makespan in all the four cases than Round Robin. Therefore, our proposed approach is more efficient in selecting and assigning cloudlets to cloud resource and producing good values of makespan than the other two algorithms.

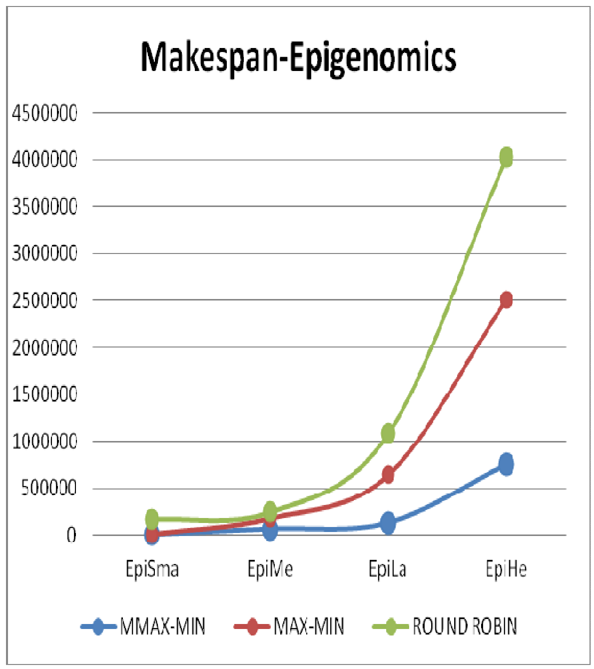

Figure. 4 Comparisons of Makespan of Different Algorithms using Epigenomics Dataset

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

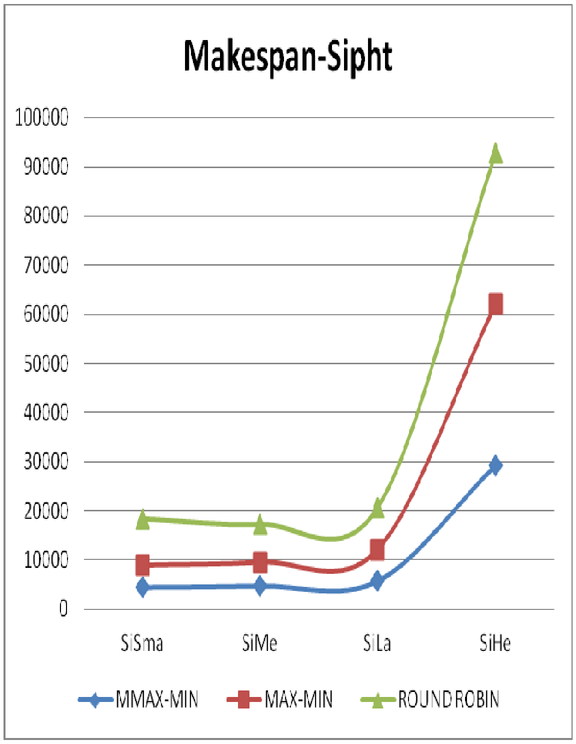

Figure 5: Comparison of Makespan of Different Algorithms using Sipht Dataset

ii. Load balancing. In cloud computing, load balancing is a process of distributing cloud load on multiple cloud resources evenly for the purpose of increation througput and optimizing the use of cloud resource. In [33], overload node is defined in equation 3, under load node in equation 4 and load balancing in equation 5.

$$p_{ij} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[EV_{ij}^{a_{k+1}}(t)]^{\beta}[\eta_{ij}(t)]^{\gamma}[D_{ij}(t)]^{\kappa}}{\sum_{j\in N_n}\{[\tau_{ij}(t)]^{\alpha}[EV_{ij}^{a_{k+1}}(t)]^{\beta}[\eta_{ij}(t)]^{\gamma}[D_{ij}(t)]^{\kappa}\}} & if\ j \in N_n \\ 0 & \end{cases} \qquad \textbf{eq 3}$$

$$p_{ij} = \begin{cases} \dfrac{[\eta_{ij}(t)]^{\gamma}[D_{ij}(t)]^{\kappa}}{\sum_{j\in N_n}[\eta_{ij}(t)]^{\gamma}[D_{ij}(t)]^{\kappa}} & if\ j \in N_n \\ 0 & \end{cases} \qquad \textbf{eq 4}$$

$$LB = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Load_i - Load_{avg})^2} \qquad \textbf{eq 5}$$

Where:

P_{ij} is the probability of moving from node *i* to node *j*.
N n is the set of yet to be visited nodes.
τ_{ij} (t) is the pheromone of node *j*.
EV_{ij}^{ak+1} (*t*) is the speed of task execution *j*. η_{ij} (t) is the cost between node *i* and node *j*,
1/*dij*. D_{ij} (*t*) is the degree of the proportion of node j

In Table 3 and Figure 6, we presented results of the proposed algorithm (MMax-Min), Max-Min and Round Robin with respect to load balancing using Epigenomics and Sipht dataset. As shown in both Table 3 and Figure 6, the proposed algorithm has consistently proven to be efficient in balancing load in all the four cases than the standard Max-Min and Round Robin. This is because MMax-Min algorithm was able to effectively distribute the available cloudlets in the clouldletlist evenly, therefore making it more efficient in terms of load balancing.

Table 3: Load Balancing by Different Algorithms using Epigenomics Dataset

Cases	Mmax-min	Max-Min	Round Robin
EpiSma	17723.35	5584.29	5584.37
EpiMe	41405.17	7731.46	7731.46
EpiLa	403411.32	53343.92	53343.93
EpiHe	3854811.9	301802.2	301802.42

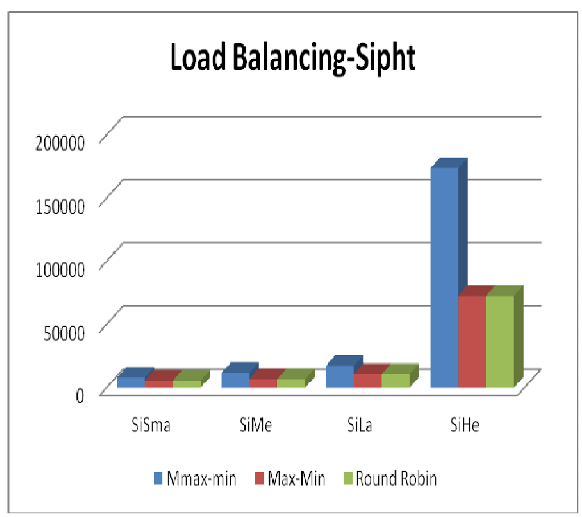

Figure 6: Load Balancing Comparison Using Sight

4. CONCLUSION AND FUTURE WORK

To overcome the major challenges in task scheduling and resource allocation using Max-Min algorithm in a cloud computing environment, a Modified Max-Min (MMax-Min) algorithm was proposed.

We simulated our approach using cloudsim toolkit which immensely saw a frequent reduction in waiting time leading to less makespan as more and more cloudlets arrived to be scheduled. This is because our approach is able to select and assign either cloudlet with maximum or minimum execution time to a resource.

Performance analysis, experimentation and benchmarking of the results indicates that our approached (MMax-Min) is able to produce good quality solutions, producing good values of makespan and balancing load effectively as compared to the standard Max-Min and Round Robin algorithms.

In future work, we will consider lengthening the scope by looking at the precedence between cloudlets and resource utilization and how efficient the algorithm can be in terms of energy consumption.

REFERENCES

[1] Zhang, J., Huang, H., & Wang, X. (2016). Resource provision algorithms in cloud computing: A survey. *Journal of Network and Computer Applications, 64*, 23-42.

[2] Kaur, G., & Sharma, E. S. (2014). Optimized utilization of resources using improved particle swarm Optimization based task scheduling algorithms in cloud computing. International Journal of Emerging Technology in Advanced Engineering, 4(6), 110-115.

[3] Pasupuleti, S. K., Ramalingam, S., & Buyya, R. (2016). An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing. *Journal of Network and Computer Applications, 64*, 12-22.

[4] Lin, Y., Thai, M., Wang, C., & Lai, Y. (2015). Two-tier project and job scheduling for SaaS cloud service providers. Journal of Network and Computer Applications, 52, 26-36.

[5] Fei Teng. Ressource allocation and scehlduling models for cloud computing. Other. Ecole Centrale Paris, 2011. English.< NNT : 2011ECAP0043 >.< tel-00659303>

[6] Rajan, A. P. (2013). Evolution of cloud storage as cloud computing infrastructure service. ArXiv Preprint arXiv:1308.1303.

[7] Duan, R., Prodan, R., & Li, X. (2014). Multi-objective game theoretic schedulingof bag-of-tasks workflows on hybrid clouds. Cloud Computing, IEEE Transactions on, 2(1), 29-42.

[8] Masdari, M., ValiKardan, S., Shahi, Z., & Azar, S. I. (2016). Towards workflow scheduling in cloud computing: A comprehensive analysis. *Journal of Network and Computer Applications, 66*, 64-82.

[9] El-Sisi, A. B., & Tawfeek, M. A. (2014). Cloud task scheduling for load balancing based on intelligent strategy. International Journal of Intelligent Systems and Applications (IJISA), 6(5), 25

[10] Huang, Q., & Huang, T. (2010). An optimistic job scheduling strategy based on QoS for cloud computing. Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on, 673-675.

[11] Choudhary, M., & Peddoju, S. K. (2012). A dynamic optimization algorithm for task scheduling in cloud environment. International Journal of Engineering Research and Applications (IJERA), 2(3), 2564-2568.

[12] Fazel Mohammadi, D., Jamali, S., & Bekravi, M. (2014) Survey on job scheduling algorithms in cloud computing. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) 2278-6856, Volume 3, (2)

[13] He, X., Sun, X., & Von Laszewski, G. (2003). QoS guided min-min heuristic for grid task scheduling. Journal of Computer Science and Technology, 18(4), 442-451.

[14] Wang, S., Yan, K., Wang, S., & Chen, C. (2011). A three-phases scheduling in a hierarchical cloud computing network. Communications and Mobile Computing (CMC), Third International Conference on, 114-117.

[15] Anousha, S., & Ahmadi, M. (2013). An improved min-min task scheduling algorithm in grid computing. Grid and pervasive computing (pp. 103-113) Springer.

[16] Elzeki, O., Rashad, M., & Elsoud, M. (2012). Overview of scheduling tasks in distributed computing systems. International Journal of Soft Computing and Engineering (IJSCE), 2(3), 470-475.

[17] Santhosh, B., & Manjaiah, D. (2014). An improved task scheduling algorithm based on max-min for cloud computing. 7. Patel, G., & Mehta, APR A Survey on various Task Scheduling Algorithm in Cloud Computing.

[18] Brar, S. S., & Rao, S. (2015). Optimizing workflow scheduling using max-min algorithm in cloud environment. International Journal of Computer Applications, 124(4).

[19] Liu, G., Li, J., & Xu, J. (2013). An improved min-min algorithm in cloud computing. Proceedings of the 2012 International Conference of Modern Computer Science and Applications, 47-52.

[20] Kaur, N., & Kaur, K. Improved max-min scheduling algorithm. OSR Journal of Computer Engineering (IOSR-JCE) e - ISSN: 2278 - 0661,p - ISSN: 2278 - 8727, Volume 17, Issue 3, Ver.1 (May –Jun. 2015), PP 42–49.

[21] Agarwal, D., & Jain, S. (2014). Efficient optimal algorithm of task scheduling in cloud computing environment. ArXiv Preprint arXiv:1404.2076.

[22] Rana, M., Bilgaiyan, S., & Kar, U. (2014). A study on load balancing in cloud computing environment using evolutionary and swarm based algorithms. Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference

[23] Patel, G., Mehta, R., & Bhoi, U. (2015). Enhanced load balanced min-min algorithm for static Meta task scheduling in cloud computing. Procedia Computer Science, 57, 545-553.

[24] Zhou, Z., & HU, Z. (2014). Task scheduling algorithm based on greedy strategy in cloud computing. Open Cybernetics.

[25] Duan, R., Prodan, R., & Li, X. (2014). Multi-objective game theoretic schedulingof bag-of-tasks workflows on hybrid clouds. Cloud Computing, IEEE Transactions on, 2(1), 29-42.

[26] Wei, G., Vasilakos, A. V., Zheng, Y., & Xiong, N. (2010). A game-theoretic method of fair resource allocation for cloud computing services. The Journal of Supercomputing, 54(2), 252-269.

[27] Marphatia, A., Muhnot, A., Sachdeva, T., Shukla, E., & Kurup, L. (2013). Optimization of FCFS based resource provisioning algorithm for cloud computing. Journal of Computer Engineering (IOSR-JCE), 2278-0661, p-ISSN: 2278-8727 Volume 10 (5).

[28] Mao, Y., Chen, X., & Li, X. (2014). Max–min task scheduling algorithm for load balance in cloud computing. Proceedings of International Conference on Computer Science and Information Technology, 457-465

[29] Bhoi, U., & Ramanuj, P. N. (2013). Enhanced max-min task scheduling algorithm in cloud computing. *International Journal of Application or Innovation in Engineering and Management*, *2*(4), 259-64.

[30] Supreeth, S., & Biradar, S. (2013). Scheduling virtual machines for load balancing in cloud computing platform. *International Journal of Science and Research (IJSR), India Online ISSN*, 2319-7064.

[31] Velte, T., Velte, A., & Elsenpeter, R. (2009). Cloud computing, a practical approach McGraw-Hill, Inc.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[32] Arshjot Kaur & Supriya Kinge (2014) A Survey of Resource Scheduling Algorithms in Green Computing. International

[33] Gao, R., & Wu, J. (2015). Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization. Future Internet 7, no. 4 (2015): 465-483