International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 11 • Issue 1 pp. 26-32 Sept 2019 - March 2020 www.csjournals.com

Comparative Analysis of Heuristic Based Task Scheduling Approach for Cloud Environment

Neha Dutta[1], Dr. Pardeep Cheema[2]
[1]Research Scholar, [2]Proffesor
ACET eternal university Baru sahib (H.P.)

Abstract: During last decade, remarkable growth has been experienced in the area of cloud computing. Its economic model is based on the satisfaction of service level agreement. Among the different factors working towards the satisfaction of service level agreement, load balancing plays a vital role. Load balancing can be achieved through task scheduling, task migration and resource allocation. In this paper authors have reviewed the literature on task scheduling in cloud computing. It has been followed by simulation analysis of min-min, max-min and sufferage heuristic approach for task scheduling in cloud computing. Simulation has been done using Cloudsim simulator and approaches has been analysed on the scale of response time, makespan and throughput parameter.
Key Words: Cloud computing, Heuristic technique, Load balancing, Task Scheduling, Min-min, Max-min, Sufferage.

1. INTRODUCTION

Cloud computing is a collection of homogeneous and heterogeneous interconnected systems, which may work in parallel or distributed mode. These group of resources are dynamically provided to user based on their demand and service level agreement. Figure 1 summarizes all the information about cloud computing in the form of diagram [1].

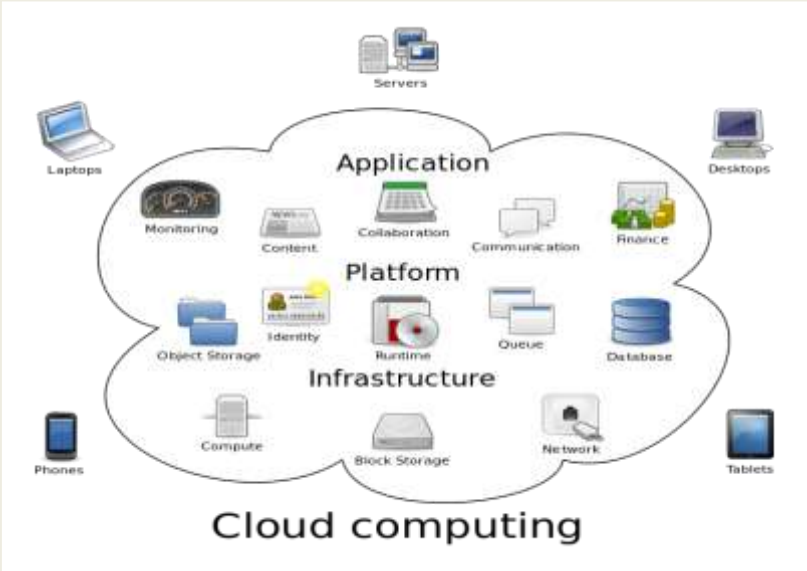

Figure 1. Outline of Cloud computing [2]

This service delivery model enables user to avail service any time anywhere as per their requirement by paying the subscription charges. Services can be provided in the form of application, environment or hardware. Based upon the service delivered, we can have Software as a service, platform as a service and infrastructure as a service cloud service delivery model. Also if cloud model is available to any user then it is called public cloud. If it is available only for employee of a company then it is called private cloud. If it's some part is open for all and some part is hidden, then it is called hybrid cloud. Similarly if cloud services are available only for a community then it is called community cloud [3].Figure 2 shows the model of working definition of cloud computing.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 11 • Issue 1 pp. 26-32 Sept 2019 - March 2020 www.csjournals.com

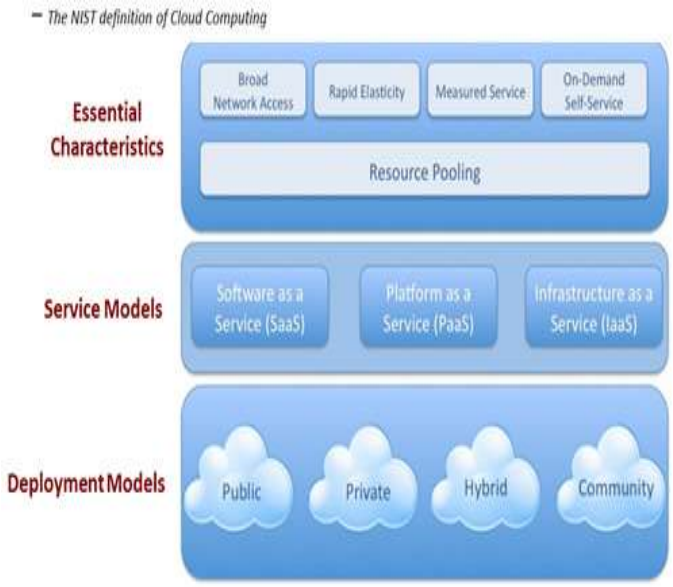

Figure 2. Model of Working Definition of Cloud Computing [4]

Structure of the paper is as follows: Section 2 covers the details of load balancing, its importance along with the goal of load balancing. In section 3, author have discussed some of the heuristic based load balancing approaches along with the algorithm. Information about the simulator has been covered in section 4. Section 5 covers the result and analysis. It has been followed by conclusion in the section 6.

2. LOAD BALANCING IN CLOUD COMPUTING

Load balancing is define as the distribution of resources, simultaneous working of the schedulers, efficiency enhancement, and minimization of response time via a suitable matching of job to the available resource. Simultaneous working of the schedulers involves the distribution of load in equal manner among the processors. To restore the balance dynamic load balancing also known as load sharing or load migration is employed [5].
It is done by distributing the entire load to the individual processors of the complete structure for obtaining efficient resource mapping and concurrently removing the possibility of overloading or under loading of the nodes in the network. It is done to achieve for better ratio of user realization and resource utilization, thereby enhancing the throughput of the complete system. If done in proper manner the load management can limit the consumption of the available. It also helps in executing failures, making the system scalable, and over-burdening, minimizing response time etc. [6]. The main goals of the load balancing algorithms are listed below:

- Rate Efficiency: Load balancing support should execute with lesser rate in the given framework. To make certain the excessive efficiency, reduced response time, and confine the overload.
- Scalability: The structure for which load balancing methods are executed should be capable of being altered in elements in later time.
- Elasticity: The practical requirements should be able to manage with circumstances; they should be elastic and adaptable. To compel prospect variations in the system.
- Priority: The resources and the tasks should be arranged as per priority. So higher works show signs of upgrading and execution.
- Resource utilization: To make sure that the available resources are being utilized in a perfect manner.
- Backup: In situation of failure of the structure, load balancing algorithms must have a backup schedule.
- Homogeneous nature: To treat all tasks in the system homogeneously irrespective of their source [7, 8].

3. HEURISTIC BASED LOAD BALANCING ALGORITHM

In distributed computing, managing the load is essential so that all the virtual machines are provided with equal number of tasks. It supports in achieving client satisfaction and optimum utilization of resources by assuring an efficient and unbiased sharing of resources among the VMs. Main features of balancing the load includes the reduction in using resources and faults. It also allows in evading jams, etc. The underlying section deals with detailed analysis of some of the heuristic based task scheduling approaches for cloud computing environment.

Min-Min Scheduling Approach [9, 10, 11]: This is a two phase scheduling approach. First of all a batch of unscheduled task is created. Their execution time on each machine is recorded and they are sorted on the basis of minimum completion time. So every time task having minimum completion time is selected first for second phase. In the second phase, task is mapped to that virtual machine which may complete its processing in minimum time. Due to involvement of two minimum criterions, it is called min-min task scheduling approach.

```
Algorithm min-min()
{
        S= receive the task till a batch of task is created.
        For each task t_i∈S
        {
                Calculate the completion time of task t_i on machine m_j.
                C_ij = E_ij + R_j
                // C_ij=completion time
                // E_ij=execution time
                // R_j=ready time
        }
        Sort the task of set S on the basis of their completion time.
        While (S!= Null)
        {
                Map the task t_i with machine m_j for which C_ij have the minimum value
                Remove the task t_i from set S.
        }
}
```

Disadvantage of this approach is that it always select the small task first. This will result in bias task selection.

Max-Min Scheduling Approach [12, 13, 14]: Like min-min approach, this is also a two phase scheduling approach. In the first phase, this approach first make a batch of unscheduled task like min-min approach. Their execution time on each machine is recorded and they are sorted on the basis of maximum completion time. So every time task having maximum completion time is selected first for second phase. In the second phase, selected task is mapped to that virtual machine which may complete its processing in minimum time. Due to involvement of first maximum and second minimum selection criterion, this task scheduling approach is called max-min approach.

```
Algorithm max-min()
{
        S= receive the task till a batch of task is created.
        For each task t_i∈S
        {
                Calculate the completion time of task t_i on machine m_j.
                C_ij = E_ij + R_j
                 // C_ij=completion time
                // E_ij=execution time
                // R_j=ready time
        }
        Sort the task of set S on the basis of their decreasing completion time.
        While (S!= Null)
        {
                Map the task t_i with machine m_j for which C_ij have the maximum value
                Remove the task t_i from set S.
        }
}
```

Disadvantage of this approach is that it always select the task having highest completion time. This will also result in bias task selection.

Sufferage Approach [15, 16, 17]: Like previous two scheduling approach, this approach is also a two phase scheduling approach. In this approach first we identify the completion time of each task on each machine. We

then record the first two minimum completion time of each task and identify the difference among them. This difference is called the sufferage of that task. Now we sort the tasks on the basis of decreasing order of the sufferage, i.e. task having the highest sufferage will be available first. In the second phase, task having highest sufferage is mapped with that machine which will complete the task in the minimum time.

Algorithm Sufferage()
{
S= receive the task till a batch of task is created.
For each task $t_i \in S$
{
Calculate the smallest and second smallest completion time of task t_i on machine m_j.
$C_{ij}(1) = E_{ij} + R_j$
$C_{ij}(2) = E_{ij} + R_j$
$SV= C_{ij}(2)- C_{ij}(1)$
// $C_{ij}(1)$=minimum completion time
// $C_{ij}(2)$=second minimum
// completion time
// E_{ij}=execution time
// R_j=ready time
// SV=sufferage value
}
Sort the task of set S on the basis of their decreasing sufferage value.
While (S!= Null)
{
Map the task t_i with machine m_j for which C_{ij} have the highest sufferage value
Remove the task t_i from set S.
}
}

4. CLOUDSIM SIMULATOR [18, 19]

Cloudsim provides a simulation framework which enable researchers to develop cloud like environment. User will able to simulate data center, hosts, virtual machines, other resources and traffic. This is possible because of variety of classes available in Cloudsim. Due to its implementation in Java language, it is easy to simulate it on any platform having JVM. Figure 3 shows the relationship among different classes available in Cloudsim.

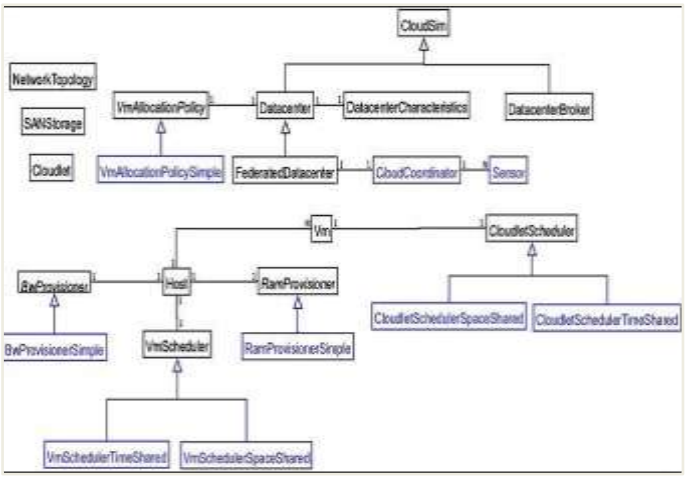

Figure 3. Cloudsim class diagram [19]

Following are the important components of Cloudsim simulator:

- **CloudSim:** This class manages event queue and execution of events related to simulation.
- **Cloudlet:** Functionality of this class is to simulate the application services which are based on cloud.
- **CloudletScheduler:** This is an abstract class which is extended for the implementation of different processing power policies.

International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 11 • Issue 1 pp. 26-32 Sept 2019 - March 2020 www.csjournals.com

- **Datacenter:** Functionality of this class is to simulate different hardware services provided by cloud.
- **DatacenterBroker or Cloud Broker:** Functionality of this class is to simulate correspondence between SaaS and cloud service provider.
- **Host:** Functionality of this class is to simulate physical resources.
- **Vm:** This class provides the functionality of virtual machine simulation.
- **VmmAllocationPolicy:** This is an abstract class which implements the VM allocation policy.
- **VmScheduler:** This is also an abstract class which provides the functionality of scheduling between virtual machine and task.

5. RESULT AND ANALYSIS

To analyze the performance of discussed approaches in section 3, makespan, average resource utilization and load balancing level has been considered.

Makespan is an indicator of total time taken to complete a set of jobs. Lower the value of makespan better will be the scheduling approach.

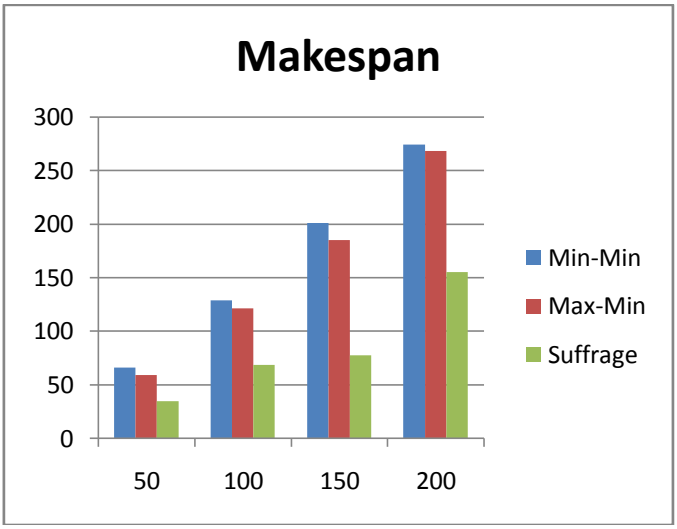

Figure 4. Comparison of heuristic approach on the basis of makespan

Figure 4 shows the comparison of three heuristic approach on the scale of makespan. It has been found that sufferage scheduling approach is showing better performance relative to min-min and max-min approach on the scale of makespan.

Resource utilization is another parameter used to judge the performance of scheduling approach. More is the value of this factor better will be scheduling approach in terms of response and waiting time.

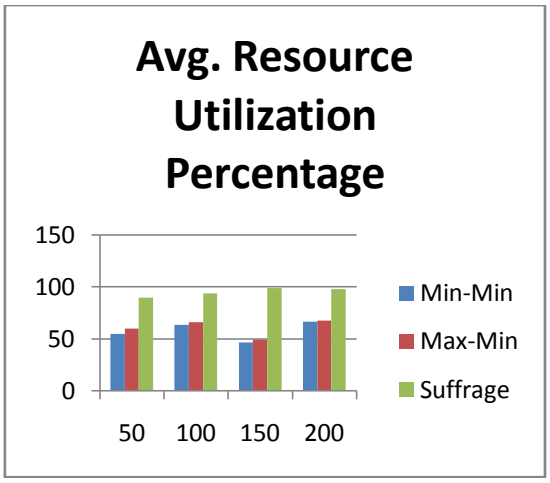

Figure 5. Comparison of heuristic approach on the basis of resource utilization

Figure 5 shows the comparison of three heuristic approach on the scale of resource utilization. It has been found that sufferage scheduling approach is showing better performance relative to min-min and max-min approach on the scale of average resource utilization.

Even distribution of task among all resources will result in better response and waiting time. So higher value of this parameter is also desirable.

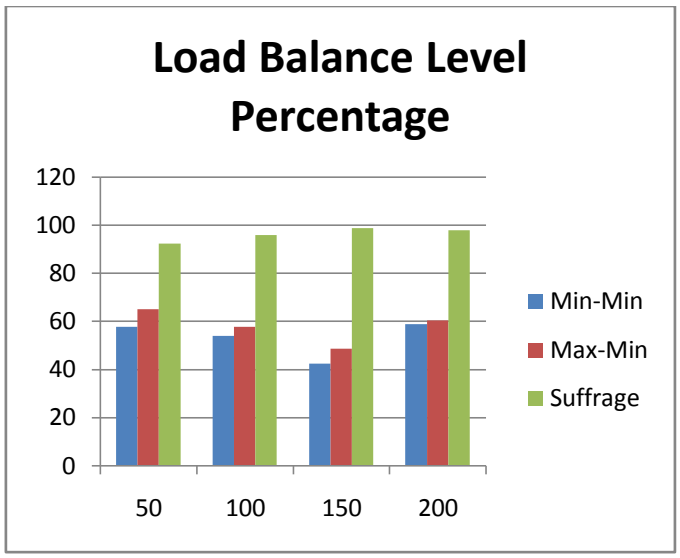

Figure 6. Comparison of heuristic approach on the basis of load distribution

Figure 6 shows the comparison of three heuristic approach on the scale of load distribution. It has been found that sufferage scheduling approach is showing better performance relative to min-min and max-min approach on the scale of load distribution.

After the combined analysis of figure 4, 5 and 6, it can be concluded that sufferageheuristic approach is best suited for scheduling of load in cloud environment among min-min and max-min.

6. CONCLUSIONS

In this paper authors have reviewed the literature on task scheduling in cloud computing. Authors have focused on heuristic approach. Authors have thoroughly analyzed min-min, max-min and sufferage approach for scheduling in cloud environment. It has been followed by simulation analysis of min-min, max-min and sufferage heuristic approach for task scheduling in cloud computing. Simulation has been done using Cloudsim simulator and approaches has been analyzed on the scale of response time, makespan and throughput parameter. Through simulation analysis, it can be concluded that sufferage heuristic approach is best suited for scheduling of load in cloud environment among min-min and max-min.

REFERENCES

[1] Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). Mastering cloud computing: foundations and applications programming. Newnes.
[2] Jain, A., & Kumar, R. (2014). A taxonomy of cloud computing. International journal of scientific and research publications, 4(7), 1-5.
[3] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
[4] Brunette, G., & Mogull, R. (2009). Security guidance for critical areas of focus in cloud computing v2. 1. Cloud Security Alliance, 1-76.
[5] Randles, M., Lamb, D., & Taleb-Bendiab, A. (2010, April). A comparative study into distributed load balancing algorithms for cloud computing. In Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on (pp. 551-556). IEEE.
[6] Al Nuaimi, K., Mohamed, N., Al Nuaimi, M., & Al-Jaroodi, J. (2012, December). A survey of load balancing in cloud computing: Challenges and algorithms. In Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on (pp. 137-142). IEEE.

[7] Jain, A., & Kumar, R. (2016, February). A multi stage load balancing technique for cloud environment. In Information Communication and Embedded Systems (ICICES), 2016 International Conference on (pp. 1-7). IEEE.
[8] Chaczko, Z., Mahadevan, V., Aslanzadeh, S., & Mcdermid, C. (2011, September). Availability and load balancing in cloud computing. In International Conference on Computer and Software Modeling, Singapore (Vol. 14).
[9] Chen, H., Wang, F., Helian, N., & Akanmu, G. (2013, February). User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In Parallel computing technologies (PARCOMPTECH), 2013 national conference on (pp. 1-8). IEEE.
[10] Moharana, S. S., Ramesh, R. D., & Powar, D. (2013). Analysis of load balancers in cloud computing. International Journal of Computer Science and Engineering, 2(2), 101-108.
[11] Wang, S. C., Yan, K. Q., Liao, W. P., & Wang, S. S. (2010, July). Towards a load balancing in a three-level cloud computing network. In Computer Science and information technology (ICCSIT), 2010 3rd IEEE International Conference on (Vol. 1, pp. 108-113). IEEE.
[12] Salot, P. (2013). A survey of various scheduling algorithm in cloud computing environment. International Journal of Research in Engineering and Technology, 2(2), 131-135.
[13] Kaur, R., & Luthra, P. (2012, December). Load balancing in cloud computing. In Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing, ITC.
[14] Jain, A., & Kumar, R. (2017). Critical analysis of load balancing strategies for cloud environment. International Journal of Communication Networks and Distributed Systems, 18(3-4), 213-234.
[15] Paul, M., Samanta, D., & Sanyal, G. (2011). Dynamic job scheduling in cloud computing based on horizontal load balancing. International Journal of Computer Technology and Applications (IJCTA), 2(5), 1552-1556.
[16] Lin, W., Liang, C., Wang, J. Z., & Buyya, R. (2014). Bandwidth- aware divisible task scheduling for cloud computing. Software: Practice and Experience, 44(2), 163-174.
[17] Paul, M., & Sanyal, G. (2011, December). Survey and analysis of optimal scheduling strategies in cloud environment. In Information and Communication Technologies (WICT), 2011 World Congress on (pp. 789-792). IEEE.
[18] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and experience, 41(1), 23-50.
[19] Buyya, R., Ranjan, R., & Calheiros, R. N. (2009, June). Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In High Performance Computing & Simulation, 2009. HPCS'09. International Conference on (pp. 1-11). IEEE.