JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

LOAD BALANCE IN CLOUD COMPUTING USING MIN MIN ALGORITHM

[1]Barot Hinal, [2]Prof.Riddhi Patel

Dept. of Computer Engineering LDRP Institute of Technology & Research, Gandhinagar, India

e-mail:hinalbarot09@gmail.com

Dept. of Computer Engineering LDRP Institute of Technology & Research, Gandhinagar, India

ABSTRACT

Cloud computing is emerging as a new paradigm of large-scale distributed computing. In order to utilize the power of cloud computing completely, we need an efficient task scheduling algorithm. The Min-Min algorithm is a simple, efficient algorithm that produces a better schedule that minimizes the total completion time of tasks than other algorithms in the literature. However the biggest drawback of it is load imbalanced, which is one of the central issues for cloud providers. In this paper, an improved load balanced algorithm is introduced on the ground of Min-Min algorithm in order to reduce the makespan and increase the resource utilization (LBIMM). At the same time, Cloud providers offer computer resources to users on a pay-per-use base. In order to accommodate the demands of different users, they may offer different levels of quality for services. Then the cost per resource unit depends on the services selected by the user. In return, the user receives guarantees regarding the provided resources.

Keywords- *Cloud computing, Load Balancing Algorithm, Min-min Algorithm,*

I. INTRODUCTION

Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Workload of a machine means the total processing time it requires to execute all the tasks assigned to the machine. Balancing the load of virtual machines uniformly means that anyone of the available machine is not idle or partially loaded while others are heavily loaded. Load balancing is one of the important factors to heighten the working performance of the cloud service provider. The benefits of distributing the workload includes increased resource utilization ratio which further leads to enhancing the overall performance thereby achieving maximum client satisfaction[1]

In cloud computing, if users are increasing load will also be increased, the increase in the number of users will lead to poor performance in terms of resource usage, if the cloud provider is not configured with any good mechanism for load balancing and also the capacity of cloud servers would not be utilized properly. This will confiscate or seize the performance of heavy loaded node. If some good load balancing technique is implemented, it will equally divide the load (here term equally defines

low load on heavy loaded node and more load on node with less load now) and thereby we can maximize resource utilization. One of the crucial issue of cloud computing is to divide the workload dynamically.[1]

Classification of Load Balancing Algorithm

Based on process orientation they are classified as:

a) Sender Initiated: In this sender initiates the process; the client sends request until a receiver is assigned to him to receive his workload

b) Receiver Initiated: The receiver initiates the process; the receiver sends a request to acknowledge a sender who is ready to share the workload

c) Symmetric: It is a combination of both sender and receiver initiated type of load balancing algorithm.

II. Min-Min Algorithm

Min-Min algorithm starts with a set of unmapped / unscheduled jobs. Min-Min Task Scheduling Algorithm is a static scheduling algorithm. This algorithm first identifies the jobs having minimum execution time and these tasks are scheduled first in this algorithm. Then it will calculate the expected completion time for each tasks according to available virtual machines and the resource that has the minimum completion time for selected task is scheduled on that resource. The ready time of that resource is updated and the process is repeated until all the unexecuted tasks are scheduled. Hence Min-Min algorithm chooses the smallest size tasks first and assigned these tasks to fastest resource. So it leaves the some resource overloaded and other remains underutilized or idle. In addition it does not provide load balanced in the system. It will provide a better makespan and resource utilization when the number of the large task is more than the number of the small task in meta-task. Main disadvantage of this algorithm is that it selects small tasks to be executed firstly, which in turn large task delays for long time. Min=Min algorithm is failed to utilize resources efficiently which lead to a load imbalance. [2]

Expected completion time of each task to the available resources is calculated using this equation:

Ctij = Etij + rtj

Where, rtj = Ready Time of resource Rj

Etij = Execution Time of task Ti on resource Rj

Ctij = Completion Time of task Ti on resource Rj

III. BACKGROUND & RELATED WORK

Today, researchers attempt to build job scheduling algorithms that are compatible and applicable in Cloud Computing environment.

Similar QoS priority grouping scheduling algorithm considers deadline and acceptation rate of the tasks and make span of the whole system as major factors for task scheduling. It achieves better acceptance rate and completion time for submitted tasks compared with Min-min and QoS guided Min-min.

IV. LITERATURE SURVEY

Our paper focuses on the various load balancing algorithms and their applicability in cloud computing environment.

- We first categorized the algorithms as static and dynamic. Then we analyzed the various algorithms which can be applied in static environments. After that we described the various dynamic load balancing algorithms. For solving any particular problem some special conditions need to be applied. So we have discussed some additional algorithms which can help in solving some sub-problems in load balancing which are applicable to cloud computing. In our future work we will analyze the algorithms with numerical analysis and simulation.[3]

- This research process, it is concluded that there are a lot of issues still open in load balancing process which can be bridged in future by applying an efficient and sophisticated load balancing algorithm most importantly along dimensions of additional QoS metrics and algorithm complexity evaluation. The survey also presents some algorithms in taxonomy which can guide the future researchers to deal with load unbalancing problem effectively like nature inspired algorithms, machine learning and mathematical derived algorithms (Markov chain, game theory based)[4]

- This paper focused on the comparative analysis of different load balancing algorithms. Load balancing is one of the major problems in cloud computing which is needed to distribute the load of work on every node for the better utilization of resources. This paper gave brief detailed knowledge about cloud computing, load balancing, load balancing algorithms and we discussed some predefined parameters based on which we compared the different algorithms. These metrics such as throughput, response time, fault tolerance, etc. helps in calculating the efficiency of algorithms and give us the idea of improving their effectiveness. By increasing the effectiveness of algorithms, the process of distribution of workload can be improved. We also discussed various load balancing algorithms with their advantages and downsides. In the future, various predefined algorithms can be combined to get a better result in these parameters so as everyone will be benefitted from the load balancing procedure whether it be a client or a cloud provider.[5]

- This article is focused on cloud computing problems and its major challenges. Cloud computing is state of-the-art computer technology which delivers customer support at all times. LB is one of the biggest problems with CC, as overloading a device will lead to terrible results that could create technology obsolete. So there is always a need for an effective LB algorithm for efficient use of resources. The main goal of LB is to meet user needs by distributing the workload across multiple network nodes & maximizing resource usage & growing device efficiency. Consequently, effective load management is critical for system efficiency, resource usage, reliability, throughput optimization and response time minimization. This research described the numerous algorithms for LB & their static load balancing algorithm, dynamic load balancing algorithm & dynamic nature inspired load balancing algorithm types. In the future, the need to build fully autonomous new dynamic LB algorithms will allow better use of resources, minimum make-span, and an improved degree of mismatch, effective task migrations, and minimum time span. CC itself is a technology that can last for years. It's one of the main innovations and we can use it to perform a critical part of the company. The above-mentioned innovations will make CC in the long term completely better.[6]

- In the literature, large numbers of task scheduling algorithm were proposed in the past. Braun et al [7] have studied the relative performance of eleven heuristic algorithms for task scheduling such as Opportunistic Load Balancing (OLB), Minimum Execution Time (MET), Minimum Completion Time (MCT), Min-Min, Max-Min, Duplex, Genetic Algorithm (GA), etc. They have also provided a simulation basis for researchers to test the algorithms. Their results show that the simple Min-Min algorithm produces a better schedule that minimizes the make span than the other algorithms and performs next to GA which the rate of improvement is also very small in most of the scenarios.[7]

V. OVERVIEW

Min - Min algorithm [13] minimizes work completion time in each node, but the algorithm does not yet consider the workload of each resource. The traditional Min-Min algorithm is the basis for the current scheduling improvement algorithm (LBIMM) in cloud computing and overcome the disadvantages of the traditional Mix-Mix algorithm. After providing an improved algorithm for considering the workload of each resource, the algorithm improved (LBIMM) for better test results

VI. Metrics of Load balancing

Researchers need to make sure that all the factors are managed effectively to increase the whole system performance.

1 Performance: System effectiveness are verified after successfully implementing the respective technique.

2 Makespan: This measures the total amount of time taken to allot resources to its users.

3 Throughput: This measures the rate at which the submitted requests are processed per unit time. The more the throughput rate, the greater the system performance.

4 Scalability: This measures the ability of the node to carry out consistent load balancing even in the case when the needed number of nodes shoots up.

5 Fault Tolerance: This measures the ability of the load balancing method to function unvaryingly even in the case of any failure of any node or link.

6 Response Time: This measures the total duration of time taken to respond to a submitted request for a service.

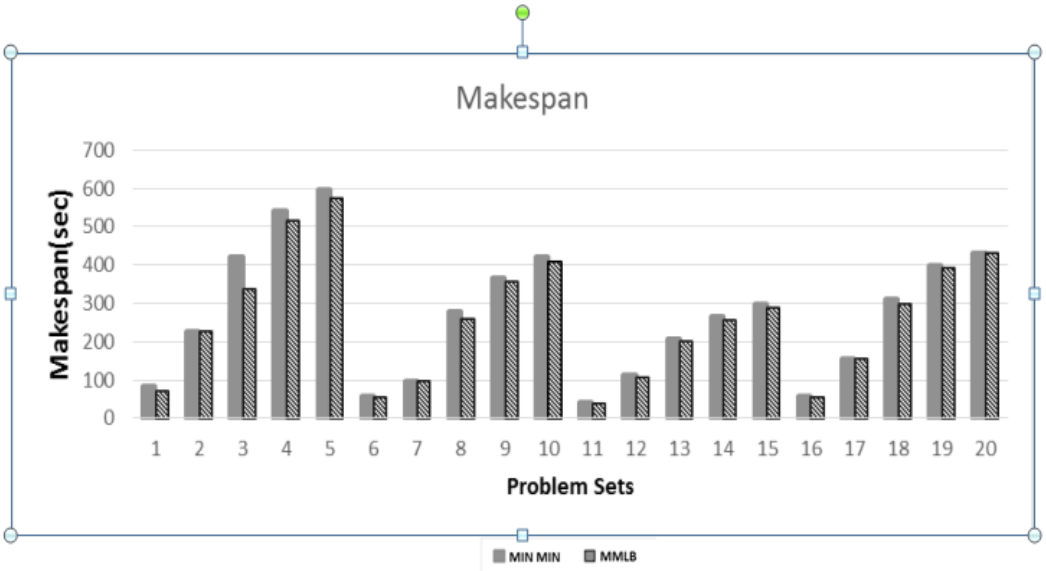
7 Migration Time: This measures the total period of time taken to shift a request from an overburdened machine to an under-burdened machine. The lesser the migration time, the greater the system performance.

8 Resource Use: This measures that all the available resources are appropriately utilized in cloud environment. The greater the IT resource usage, the lesser the overall cost, energy expenditure and carbon emission.

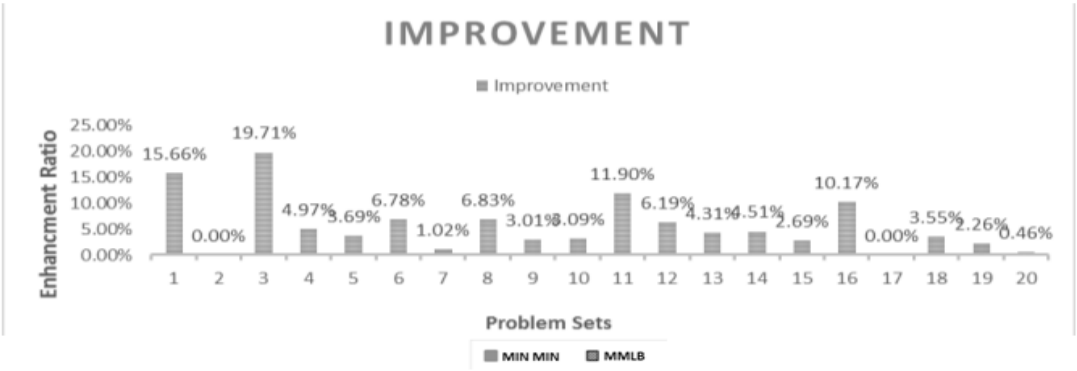
VII. TEST TASK USING PROPOSED ALGORITHM:

VIII.

- For all submitted tasks in the set; ECT ← **Call** Min-Min(T[1..N], R[1..K])
  // Create ESCT adds new columns to each row with total complete per task.
- For all resources; Rj
  ESCT ← **Call** Construct (ECT)
  // Find makespan of all resources.
- Calculate Ctij=Etij+rtj;
- Do while tasks set is not empty
- Find task Tk that cost minimum execution time.
  For i=1 to N do
  // Find Task Ti with min ESCT value
- Assign Tk to the resource Rj which gives minimum expected complete time
  For j=1to K do
  // Find a resource Rj with maxCT time
- Find the difference between MaxET and MinET
- If Diff <=MinET, assign MaxET to the resource that produces it minimum completion time
- Else assign MinET
- delete the task assigned from the set of tasks
  If Max-ECT < makespan then
  Remove(Task(Ti) from meta-Task)
- update ready time rti for the selected resuources R
- update Ctij for all Ti
- End if
  end do



Improvement the Makespan of MMLB over Min-Min



. Improvement the ratio achieved by using the proposed algorithm rather than using Min-Min algorithm

IX. CONCLUSION

Min-Min algorithm are common applicable in small scale distributed systems . When the number of small tasks is more than number of the large tasks in a meta-task. This study is only concerned with the number of the resources and the tasks. The study can be further extended by applying the proposed algorithm on actual cloud computing environment and considering many other factors such as scalability, availability, stability and others. Also, in future we can improve the presented algorithm to be optimized and produce more efficient .

X. REFERENCES

1) Foram F Kherani, Prof. Jignesh Vania ”Load balancing in cloud computing” International Journal of Engineering Development and Research, Volume 2, Issue 1,2014
2) Bhavisha Patel, Mr. Shreyas Patel” Various Load Balancing Algorithms in cloud Computing” *Vol-1 Issue-2 201, IJARIIE*
3) SHANTI SWAROOP MOHARANA, RAJADEEPAN D. RAMESH & DIGAMBER POWAR “ANALYSIS OF LOAD BALANCERS IN CLOUD COMPUTING” International Journal of Computer Science and Engineering, Vol. 2, Issue 2, May 2013
4) Shahbaz Afzal and G. Kavitha “Load Balancing in cloud computing-A hierarchical taxonomical classification ” Journal of Cloud Computing, 2019

5) Mohit Kumar, Bharat Bhushan "A Methodological Comparison of the most efficient Load Balancing algorithms in Cloud Computing
6) MUHAMMAD ASIM SHAHID, NOMAN ISLAM, MUHAMMAD MANSOOR ALAM, MAZLIHAM MOHD SU'UD AND SHAHRULNIZA MUSA "A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach" VOLUME 8, 2020 IEEE ACCESS
7) Huankai Chen, Professor Frank Wang, Dr Na Helian, Gbola Akanmu "User-Priority Guided Min Min Scheduling Algorithm For Load Balancing in Cloud Computing" PerComp Tech ,2013
8) B. Yagoubi, and Y. Slimani, "Task Load Balancing Strategy for Grid Computing," Journal of Computer Science, Vol. 3, No. 3, pp. 186-194, 2007
9) F. Dong, J. Luo, L. Gao, and L. Ge, "A Grid Task Scheduling Algorithm Based on QoS Priority Grouping," In the Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06), IEEE, 2006.
10) Pooja Arora, Anurag Dixit," An optimized Load Balancing Algorithm in Cloud Computing" (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-5, June 2020
11) Sajjan R.S, Biradar Rekha Yashwantrao" Load Balancing and its Algorithms in Cloud Computing" Volume-5, Issue-1 on 16 February 2017
12) A. A. Jaiswal, Dr. Sanjeev Jain," An Approach towards the Dynamic Load Management Techniques in Cloud Computing Environment", 2014
13) J. Rathore, ''Review of various load balancing techniques in cloud computing,'' Comput. Sci. Electron. J., vol. 7, no. 1, p. 5, 2015.