A Genetic based Improved Load Balanced Min-Min Task Scheduling Algorithm for Load Balancing in Cloud Computing

Shyam Singh Rajput
Department of Computer Science & Engineering
Institute of Technology and Management
Gwalior, India
ssrshyam59@gmail.com

Virendra Singh Kushwah
Department of Computer Science & Engineering
Institute of Technology and Management
Gwalior, India
kushwah.virendra248@gmail.com

Abstract - **Cloud computing is evolving as a new model of big-scale distributed computing. It provides own services to online on-demand and pay-as-you-go basis. In cloud computing environment load balancing is a key issue which is required to distributing the dynamic workload over multiple machines to make certain that no single machine is overloaded. In order it helps in ideal use of resource and as a consequence enhancing the performance of the system, we need an efficient task scheduling algorithm. The Min-Min algorithm is simple that produce a schedule that minimize the makespan but this algorithm does not make use of resource effectively. In this paper, we proposed an Improved Load Balanced Min-Min (ILBMM) algorithm using genetic algorithm (GA) in order to minimize the makespan and increase the utilization of resource. The implementation of proposed algorithm has been completed using CloudSim simulator and simulation outcomes demonstration that the proposed algorithm outperforms to current algorithm on same objectives.**

Keywords- Cloud Computing, Load Balancing, LBMM Algorithm, Genetic Algorithm, Cloudlet, Virtual machine, makespan.

I INTRODUCTION

In the upcoming period the term Cloud Computing has turn out to be an emerging and quickest technology in the world [1]. Cloud computing technology set all the resources as a single point of access to the buyer and price is pay per consumption. In this technology all the resources are dynamically issue to private and public networks. In this cloud computing model individual can acquire computer resources and information from wherever that a network connection is available [2]. In the cloud environment, load balancing is a key challenge. It's a technique that improves the performance that applied in the space of network to distribute the workload similarly across all the available machines. This may avoid the problem the place some nodes are high loaded even as others are idle or doing slight work. It benefits to attain a consumer satisfaction and useful resource utilization ratio. It also confirms that each computing resource is distributed efficiently [3]. For better use of resource we need a well load balancing algorithm. In cloud computing environment there are a couple of load balancing algorithms is exists like: Min-Min, Max-Min, Load Balance Min-Min (LBMM), FCFS, SRTF, Round-Robin, Randomized algorithm etc. [4]. The Min-Min Task Scheduling Algorithm is a simple that initiate with a set of un-scheduling tasks. That minimizes the makespan but not make use of resource effectively. In the LBMM algorithm first the Min-Min algorithm is performed except all tasks are scheduled. Then the LBMM selects the resources with high load and reallocates the resources with low load. The LBMM technique minimizes the makespan and increases the utilization of resource [5]. Genetic Algorithm (GA) is a random search method situated on the recommendations of natural selection and genetics. It is used to resolve an optimization and search problems [6]. Generally GA is a four step process these are: Selection, Crossover, Mutation, and Termination [7].

In this paper, we proposed an improved load balanced min-min algorithm (ILBMM) using the genetic algorithm. Here we use a crossover, mutation and fitness function of GA.

The outlined of present paper as follows: The related work is mentioned in Section II, in Section III described a load balanced min-min algorithm, Section IV described a genetic algorithm, Section V and VI described an problem statement and proposed work, Section 7 simulation results and comparison and Section 8 gives conclusion.

II RELATED WORK

Load balancing task scheduling algorithms has at all times been a research area in cloud computing. In the related work, large numbers of task scheduling algorithms were proposed in the past. H. Chen et al. [8] studied the pre-defined Min-Min algorithm and considered it as a base algorithm to proposed Load Balance Improved Min-Min algorithm (LBIMM) and User-Priority Guided Load Balance Improved Min-Min scheduling algorithm (PALBIMM). To exam the outcomes a simulation basis is too provided. In [9] have proposed enhanced Max-Min task-scheduling algorithm for flexible cloud, that algorithm preserves a task position table to estimate the actual-time load of virtual machines and predictable tasks completion time, that can assign the work-load between nodes and comprehend the load balance and those proposed algorithm can enhance the resource utilization in addition to decrease the response time of tasks.

Mayur S. Pilavare et al. [7], improvement the performance of Load Balancing with the aid of Genetic Algorithm by allotting the priority to the VMs and giving the prioritized input to the genetic algorithm can rise the response time of the system and the enhanced load balancing can be attained in cloud computing. That used Logarithmic Least Square Matrix

978-1-5090-1144-5/16 $31.00 © 2016 IEEE
DOI 10.1109/CICN.2016.139

Technique for allocating the priority to the VM's. In [10], Gaurang patel have proposed improved Load balanced Min-Min (ILBMM) algorithm for task scheduling and describe a kind of task scheduling algorithm. This ELBMM algorithm decreases the makespan and increases make use of resource effectively. The proposed algorithm has two-segments. In the first segment the pre-defined Min-Min algorithm is performed and in next segment the task are reschedule to use unutilized resources properly.

Mahesh B. Nagpure et al. [2] proposed a dynamic resource allocation system which allotted resources to cloud user. The Skewness Algorithm used for measure irregular use of several resources of all VMs and in accordance skew value load balance throughout VMs. Those easily combine dissimilar several resources and increase utilization of resources. That archive improves performance in terms of server resources on the server between VMs and checks obtainable server resources and guess future load to avoid over-load on server.

S. G. Domanal and Mohana Reddy et al. [11] proposed an algorithm VM-assign which responsible for distribution of whole incoming requests to the available virtual machine. And the proposed technique underutilization of the VM is improved significantly. In [12], a new enhancement of the load balancing proposed through the algorithm "estimated the completion time load balancer" that allows the cloud services providers to growth the performance and maximize the use of VMs of their data center and the algorithm calculates and estimates the completion time of task processing.

Xiaoming Dai et al. [13] have proposed job spanning time and load balancing genetic algorithm (JLGA) based on double-fitness adaptive algorithm to achieve task scheduling with minimum makespan and load balancing. And that adopts greedy algorithm to initialize the population and described the multi-fitness function. Then that evaluated the performance of JLGA with 30 jobs. In [14], a heuristic based load balanced scheduling model for efficient performance of tasks. The model balances the loads coming from several users among datacenters and hence it offers better resource utilization and excessive availability in the form of better response and turnaround time.

III Load Balanced Min-Min Algorithm

The Min-Min algorithm is simple that produce a schedule that minimized the makespan but this algorithm doesn't utilize resource efficiently. Load Balanced Min-Min (LBMM) algorithm minimized the makespan and also increases the resource utilization. The LBMM algorithm works in two stages.

- In the first stage the Min –Min algorithm is performed.
- In the second stage the tasks are rescheduled to use the resources efficiently.

The Min-Min algorithm is first determines the minimum execution time of all tasks. Then the algorithm selects the task with the least execution time amongst all the tasks. The algorithm progress by allocating the task to the resource that produces the minimum completion time. The similar procedure is repeated until all the available tasks are scheduled. Main weakness of this technique it's selects small tasks to be completed initially, which in turn lengthy delay for very long time.

To overcome this weakness, Max-min algorithm schedules larger tasks first. But in some cases, the makespan may increase due to the execution of larger tasks first. The waiting time of smaller tasks is too high in Max-min.

The LBMM algorithm outperforms those algorithms both in phrases of makespan and load balancing. Therefore a well load balancing is attained and response time of the procedure is improved [10].

- So, the LBMM algorithm first executed Min-Min algorithm.
- Then, the LBMM selects the resources with heavy load and reallocates the resources with light load.

LBMM recognizes the resources with heavy load by selecting the resource with high makespan in the schedule produced by Min-Min. Then LBMM considers the tasks assigned in that resource and selects the task with least execution time on that resource. The completion time of the task is calculated for all resources in the present schedule. Then the maximum completion time of that task is compared by the makespan produced by Min-Min. If it not up to makespan then the task are rescheduled, and the prepared time of each resources is updated.

Otherwise the next maximum completion time of that task is selected and the steps are repeated again. The process stops if all resources and all tasks assigned in them have been considered for rescheduling. Consequently the likely resources are rescheduled within the resources which are idle or have minimal load [5].

IV Genetic Algorithm

The Genetic Algorithm (GA) is an optimization algorithm which uses the method of computerized search based on natural selection and genetics. This algorithm was published (1975) by Prof. John Holland. The essential thought behind of this procedure is to commence with randomly generated chromosomes and carry out the "survival of the fittest" scheme with a purpose to develop upgraded solutions [15].

A solution generated by means of genetic algorithm is known as a chromosome, even as collection of chromosome is denoted as a population. A chromosome is consists from genes. These chromosomes will suffer a procedure named fitness function to measure the appropriateness of solution generated by GA with problem. Several chromosomes in population will mate via procedure named crossover accordingly generating new chromosomes called offspring which its genes composition are the combination of their parent.

In a generation, a few chromosomes will also mutation in their gene. The number of chromosomes which will suffer crossover and mutation is managed by crossover rate and mutation rate value. Chromosome in the population to be able to keep for the next generation will be chosen, the chromosome which has high fitness value will have larger probability of being chosen again in the next generation. After a number of generations, the chromosome value will converges to a certain value which is the finest solution for the problem [16].

In the genetic algorithm procedure is as follows [17]:
Step 1: Determine the number of chromosomes, generation, and crossover rate and mutation rate value.
Step 2: Produce chromosome-chromosome quantity of population, and the initialization value of the genes chromosome-chromosome with an arbitrary value
Step 3: Process steps 4-8 till the quantity of generations is met
Step 4: Assessment of fitness value of chromosomes by using calculating objective function
Step 5: Chromosomes selection
Step 6: Perform Crossover
Step 7: Perform Mutation
Step 8: Offspring (New Chromosomes)
Step 9: Solution (Finest Chromosomes)

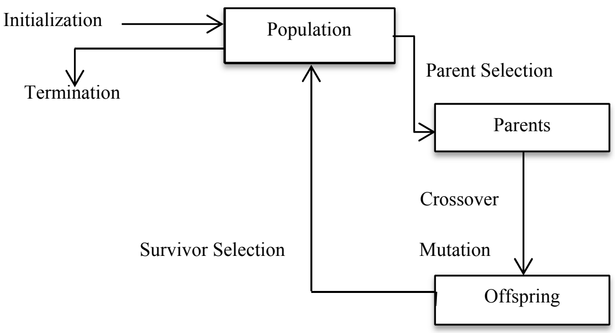

Figure 1: Flow chart of Genetic Algorithm

V PROBLEM STATEMENT

In existing Load Balanced Min-Min procedure all short process assign to VM and long process wait for completion of these short process, overcome this problem we propose a genetic based Improved LBMM approach to schedule all process on VM's and produce a better makespan and increase the utilization of resource then existing LBMM.

VI PROPOSED WORK

Cloud computing is a growing field of research, load balancing is among the predominant area of cloud computing where we maintain the execution of task timely manner. There are lots of works already done in load balancing but they have some issues. In Load Balanced Min-Min (LBMM) algorithm have the drawback that large task have wait for execution. Overcome this problem we propose a genetic based Improved Load Balanced Min-Min (ILBMM) algorithm. In our proposed work first we calculate the execution time of task on the virtual machine. Now we have the minimum or maximum time of task on virtual machine (VM), for better execution we apply genetic based approach on million instructions (MI) of task and million instruction per second (MIPS) of VM.

Crossover- In this phase we convert the MI of task and MIPS of VM into binary form, than we take last four bit of MI and MIPS for crossover.

Procedure for crossover:
A[] it contain 4-bits of VM, B[] it contain 4-bits of task.

```
For (i=0; i<length.A[i]; i++)
    If (A[i]>B[i]) Then
        C[i] =A[i] // C[i] contain bits of A[i]
    Else If
        C[i] =B[i] // C[i]   contain bits of B[i]
    Else
        C[i] = A[i] // C[i] contain bits of A[i]
End For
```

Now C[i] contain crossover bits for crossover we take highest bits of among two bits.

Mutation- In mutation we mutate one LSB bit of crossover.
Procedure for mutation

```
For (int i =0; i<4; i++)
    If (Ch[i] == '0') Then // 'i' contain 0 then mutate the bit.
        Ch[i] = '1'
    End If
    Else    // 'i' contain 1 then mutate the bit.
        Ch[i] ='0'
End For
```

This process continue for the entire task, now convert the mutate bit into decimal form and compare the value highest decimal value declare that which task execute in which VM. Than we convert binary bits into string values, so that we compare all the values and find out which task execute in which VM.

Fitness Function- In this function we get string value of our new generated offspring. Compare the fitness of new offspring and took highest fitted offspring to execute.

```
For (int i =0; i<string.length; i++)
    If (string[i]>maxvalue) Then
    // find the highest fitted offspring for execution
        Maxvalue = string[i];
        Loc = i;
    End If
End For
```

Where max value initial first task string value.

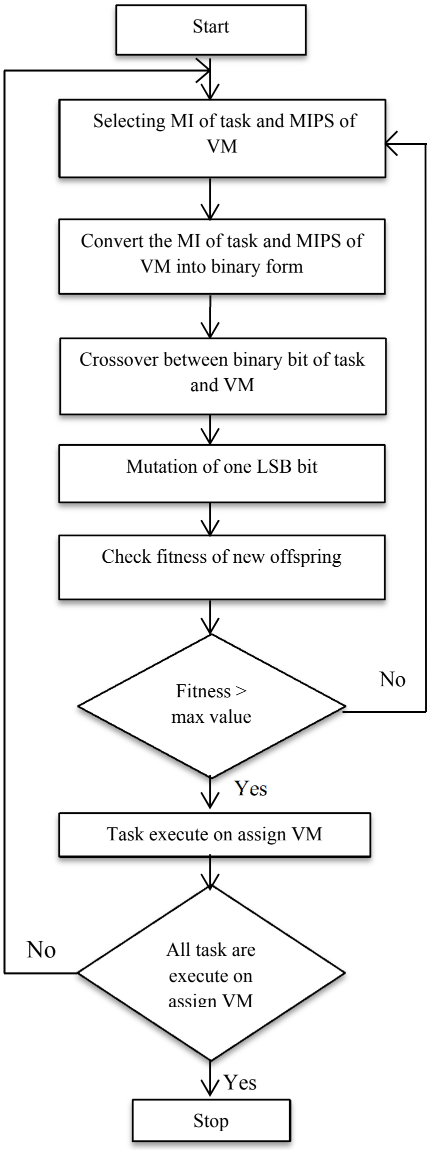
Start
Selecting MI of task and MIPS of VM
Convert the MI of task and MIPS of VM into binary form
Crossover between binary bit of task and VM
Mutation of one LSB bit
Check fitness of new offspring
Fitness > max value
No
Yes
Task execute on assign VM
All task are execute on assign VM
No
Yes
Stop

Figure 2: Flow chart of Genetic based Improved LBMM Algorithm

VII SIMULATION RESULTS AND COMPARISON

In a simulation conducted on Cloudsim, one data center is deployed, which has one physical host. There are take four task (cloudlet) T0, T1, T2 and T3 and also take two virtual machines VM0 and VM1. Table 1 represents the volume of instructions of tasks T0 to T3. Instruction volume is specified in MI (Million instructions) unit and Data volume is specified in Mb.

Table 1: Tasks Specification

Task	Instruction Volume (MI)	Data Volume (Mb)
T0	8178	137
T1	11295	258
T2	12109	182
T3	6107	137

Table 2 represents the processing speed and bandwidth of communication links for virtual machines where processing speed is specified in MIPS and bandwidth is specified in Mbps.

Table 2: Virtual Machine Specification

Virtual Machine	Processing Speed (MIPS)	Bandwidth (Mbps)
VM 0	100	70
VM 1	350	60

After that the tasks (cloudlets) want to be execute in data center for this we apply an ILBMM algorithm. After applying this algorithm these following tasks have been assigned onto virtual machine as shown below in Table 3.

Table 3: Assignment of Cloudlets onto Virtual Machines

Cloudlet	Virtual Machine
Cloudlet 0	VM 0
Cloudlet 1	VM 1
Cloudlet 2	VM 1
Cloudlet 3	VM 1

Using data given in Table 2 and Table 3, to calculate the finishing time of the cloudlets on assign virtual machine.

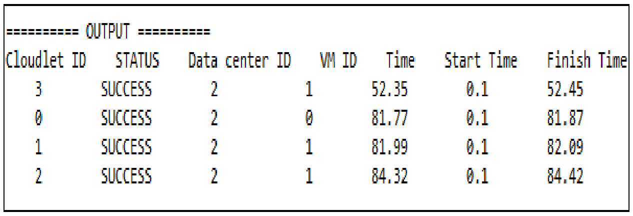
```
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    3          SUCCESS         2             1      52.35      0.1          52.45
    0          SUCCESS         2             0      81.77      0.1          81.87
    1          SUCCESS         2             1      81.99      0.1          82.09
    2          SUCCESS         2             1      84.32      0.1          84.42
```

Figure 3: Time taken to finish the tasks

Figure 3 demonstrates calculated finishing time of various cloudlets for Improved Load Balanced Min-Min (ILBMM) algorithm. The LBMM algorithm achieves makespan equals to 309.70 milliseconds and ILBMM algorithm achieves makespan equals to 300.83 milliseconds.

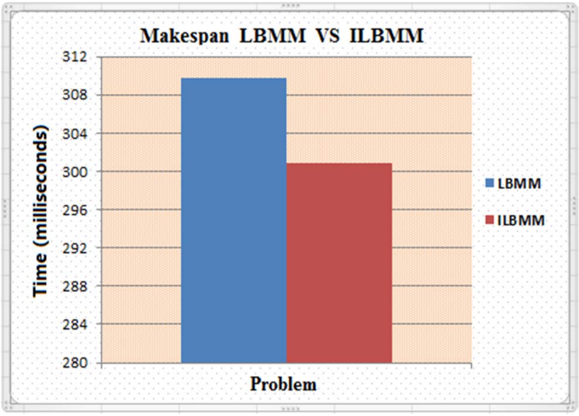

Figure 4: Makespan (Total Completion Time of all tasks)

Figure 4 represent that the ILBMM algorithm gives better Load Balance and Makespan than LBMM algorithm.

VIII CONCLUSION

In this paper, an Improved Load Balanced Min-Min task scheduling algorithm (ILBMM) has been proposed using genetic algorithm (GA). The proposed algorithm minimizes the makespan and increase the utilization of resource. Here we use a crossover, mutation and fitness function of genetic algorithm and we apply a genetic based approach on MI of task and MIPS of VM. The improved LBMM algorithm has been implemented using CloudSim with four cloudlets and two virtual machines. The all the cloudlets are assign to the virtual machine and the results of this improved LBMM algorithm demonstrate that the algorithm is minimize the makespan than to the existing LBMM algorithm.

ACKNOWLEDGEMENT

We are thankful to department of computer science & engineering, Institute of Technology and Management, Gwalior for supporting and making our work in better way.

REFERENCES

[1] Reena Panwar and Bhawna Mallick, “Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm”, IEEE International Conference on Green Computing and Internet of Things, 2015, pp. 773-778.

[2] Mahesh B. Nagpure, Prashant Dahiwale and Punam Marbate “An Efficient Dynamic Resource Allocation Strategy for VM Environment in Cloud”, IEEE International Conference on Pervasive Computing (ICPC) 2015.

[3] Velagapudi Sreenivas, Prathap,M and Mohammed Kemal, “Load Balancing Techniques: Major Challenge In Cloud Computing – A Systematic Review”, IEEE International Conference on Electronics and Communication Systems (ICECS), 2014.

[4] Amandeep, Vandana Yadav and Faz Mohammad, “Different Strategies for Load Balancing in Cloud Computing Environment: a critical Study”, International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 3 Issue 1, April 2014, pp.85-90.

[5] T. Kokilavani, D.I. George Amalarethinam, “Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing”, International Journal of Computer Applications, Volume 20– No.2, April 2011.

[6] John McCall, “Genetic algorithms for modeling and optimization”, Journal of Computational and Applied Mathematics, 2005, pp. 205-222.

[7] Mayur S. Pilavare, Amish Desai, “A Novel Approach Towards Improving Performance of Load Balancing Using Genetic Algorithm in Cloud Computing”, IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIIECS, 2015.

[8] Huankai Chen, Frank Wang, Na Helian and Gbola Akanmu, “User-Priority Guided Min-Min Scheduling Algorithm for Load Balancing in Cloud Computing”, IEEE Conference on Parallel Computing Technologies (PARCOMPTECH), 2013.

[9] Xiaofang Li, Yingchi Mao, Xianjian Xiao and Yanbin Zhuang, “An Improved Max-Min Task-Scheduling Algorithm for Elastic Cloud”, IEEE International Symposium on Computer, Consumer and Control-2015, pp. 340-343.

[10] Gaurang Patel, Rutvik Mehta, Upendra Bhoi, “Enhanced Load Balanced Min-Min algorithm for Static Meta Task Scheduling in Cloud Computing”, 3rd Science Direct International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

[11] Shridhar G. Domanal, G. Ram Mohana Reddy, “Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines ", 6th IEEE International Conference on Communication Systems and Networks (COMSNETS) 2014.

[12] YoussefFAHIM, Elhabib BEN LAHMAR, El houssine LABRIJI, Ahmed EDDAOUI, “The load balancing based on the estimated finish timeof tasks in cloud computing”, 2nd IEEE International Conference.

[13] Tingting Wang, ZhaobinLiu, Yi Chen, Yujie Xu and Xiaoming Dai, “Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing”, IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, 2014, pp. 146-152.

[14] Raza Abbas Haidri, C. P. Katti and P. C. Saxena, “A Load Balancing Strategy for Cloud Computing Environment”, IEEE International Conference on Signal Propagation and Computer Technology (ICSPCT), 2014, pp. 636-641.

[15] M. Durairaj and P. Kannan, “Improvised Genetic Approach for an Effective Resource Allocation in Cloud Infrastructure”, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 6 (4), 2015.

[16] Krantikumar K. Bhosale and Vijay D. Jadhav, “A review of genetic algorithm for metal cutting processes and a research agenda”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 3 Issue V, 2015.

[17] Denny Hermawanto,” Genetic Algorithm for Solving Simple Mathematical Equality Problem”, https://arxiv.org.