IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Special Issue-10, May 2019 E-ISSN: 2347-2693

Load Balancing Algorithm Based On Task Transfer and Information Policy

Aakriti Pathak[1*], Anshul Khurana[2]

[1,2] CSE, SRIT, RGPV, Jabalpur, India

**Corresponding Author: pathakakriti92@gmaill.com, Tel.: +91-7000162139*

DOI: https://doi.org/10.26438/ijcse/v7si10.127130 | Available online at: www.ijcseonline.org

Abstract— Cloud computing is a modern paradigm to provide services through the Internet. Nowadays, cloud computing is emerging field in information technology, next generation of computing. It provides very extensive measure of computing and storage Service gave to users through the internet which follows pay-as-you-go model. Major problems faced in the cloud are resource discovery, fault tolerance, load balancing and security. Load balancing is one of the main challenges, important technique, and critical issue and plays an important role which is required to distribute workload or task equally across the nodes or servers. Load balancing is a key aspect of cloud computing and avoids the situation in which some nodes become overloaded while the others are idle or have little work to do. Load balancing can improve the Quality of Service (QoS) metrics, including response time, cost, throughput, performance and resource utilization.

Scheduling in Cloud computing infrastructure contain several challenging issues like computation time, budget, load balancing etc. Out of them, load balancing is one the major challenges for Cloud platform. Load balancing basically balances the load to achieve higher throughput and better resource utilization. Since scheduling task is NP-complete problem, so heuristic and meta heuristic approaches are preferred options to solve the same. In this dissertation, we chose a meta-heuristic approach to solve the task scheduling problem in cloud environment focussing mainly on two objectives, i.e., minimizing the makespan/ computation time and better load balancing.

Keywords— Load balancing, Execution time, response time, Task Transfer

I. INTRODUCTION

Cloud is just the Internet with all the standards and associated protocols which provide us with whole set of web services. When we have the Internet as cloud, then we are representing abstraction, which is one of the most important and essential property of cloud computing. In the Internet, which is called as cloud, resources are pooled and partitioned as needed and communications are based on some standards [1]. In general, cloud computing is engaged to deliver a totally different situation to make available a computing resource as associate degree overhaul over the network of networks. Thus, cloud computing depends on using and sharing a pool of virtual and/or physical resources, instead of activating the native or personal hardware system and software system. Cloud computing improves the supply and utilization of resources. Users can use the services of cloud on pay-per-use on demand mode and who provides this kind of services are called as cloud service providers. Microsoft, Google, Amazon etc., are the cloud service providers.

In IT industries the word 'cloud' is commonly used in some networks which setup is not commonly known to us. For example, Internet Service Provider's (ISP) network. The user of the internet is not known to the network setup, but they use it easily. Generally in networking diagrams ISP's networks are represented by cloud symbol. So the computing, which is done over the internet without knowing the resource's location can be called as cloud computing.

Cloud computing delivers a computing environment where different resources are delivered as a service to the customer or multiple tenants over the internet. Task scheduling is an essential and most important part in a cloud computing environment. The task scheduling mainly focuses to enhance the efficient utilization of resources and hence reduction in task completion time. A new type of computer model i.e. Cloud provides the user with convenient and ever-demanding access of network for various computing resources over the Internet. Cloud computing which is a huge distributed computing environment contains a large amount of virtualized computing resources available for individual or an organization. Reason behind development or introduction of cloud computing is to provide the guarantee of Quality of Service (QoS) which is quite challenging. Scheduling of job and maintaining load is becoming a main issue in cloud environment. This can be achieved by adopting appropriate task scheduling algorithm.

By considering parameters such as throughput, resource utilization, cost, computational time, priority, performance, bandwidth, resource availability and many more, Scheduling algorithms are implemented [2]. In order to provide better Quality of Service (QoS), we need to implement an appropriate task scheduling algorithm which maintains a good balance between resource utilization to enable efficiency, lesser makespan, concurrent task scheduling, proper resource utilization and management.

The proposed system develops a new algorithm for scheduling task over the available resources with comparison from the existing one. Our focus is to distribute the tasks over the resources in an appropriate manner to achieve lesser makespan as compare with other existing algorithms and increasing the profit of Cloud service Provider.

Section II contains related work carried out previously. Section III describe proposed method, Section IV represents result and performance evaluation. Section V contains conclusion and future work.

II. RELATED WORK

There are several task scheduling parameters used in different task scheduling algorithms. In this thesis, we will discuss some of these parameters such as execution time, makespan, resource utilization; load balancing, fault tolerance, energy consumption, performance, response time and QoS used in cloud task scheduling techniques.

A. Genetic-Based Task Scheduling

Safwat A Hamad et al. [3] proposed a scheduling technique on the basis of genetic algorithm i.e TS-GA to allocate and execute the application's task. When iteration starts, a solution that needs a good fitness function is selected and included in a population and the best solution is generated. The aim is to minimize makespan, cost and increase utilization of resources. The results show that the cost, completion time, and resource utilization of the algorithm is minimized and speed up is improved as compared to default GA and RR algorithms. Bahman Keshanchi et al. [4] proposed an improved genetic algorithm (N-GA) to minimize makespan and execution time. The algorithm proposed uses heuristic techniques with the genetic algorithm. A behavioral model is constructed on the basis of checking approaches for checking the correctness. The correctness is examined based on results verified according to certain anticipated specifications, deadlock-free, fairness and reach ability. The result is compared with other heuristic algorithms.

B. Multi-task Scheduling

Yongkui Liu et al. [5] presented a model based on multi-task scheduling that integrates workload modelling of the task with several service-related qualities. A model is created first and then the examination of different work-load effects according to task scheduling procedures such as random scheduling and work-based scheduling is done. Results show that scheduling higher priority workload is preferable for obtaining an efficient performance of the system like minimum makespan and high service rate.

C. Cloud Service scheduling model

Hongyam Cui et al. [6] presented a model of cloud service scheduling known as Task Scheduling System (TSS). Here, Genetic algorithm-Chaotic Ant Colony Optimization (GA-CACO) is proposed where weighted sum of makespan and flow time is used as the objective function. It is found out that the proposed algorithm has optimal performance than ACO and GA.

D. Task Scheduling Algorithm based on Cloud Pricing Model

Elhossiny Ibrahim et al. [7] proposed an enhancement task scheduling algorithm to reduce makespan and decrease the price of executing independent tasks. In this, the processing power of all virtual machines and user's task requests are calculated and then allocate a group of user task to each virtual machines based on the ratio of the power needed to the total processing power of all virtual machines.

Using Amazon EC2 and Google Pricing models, the power of virtual machines are determined. According to their strategy, there are few improvements like minimization of makespan and decrease of the price compared to FCFS, PSO, and GA algorithms.

E. Task scheduling based on Discrete Symbiotic Organism search algorithm.

Mohammad Abdullahi et al. [8] proposed a Discrete Symbiotic Organism Search algorithm (DSOS). It uses three symbiotic associations in the biological community for enhancement of the fitness function. It consists of three stages: population generation, iteration stage, and termination. Makespan, the degree of imbalance and response time were found to be better than self-adaptive particle swarm optimization (SAPSO).

F. An Extended Intelligent Water Drops algorithm based cloud scheduling

Shaymma Elsherbiny et al. [9] proposed an intelligent water drops (IWD) based cloud scheduling (IWDC) algorithm. It consists of three phases: initialization and preparation of parameters then path construction and task assignment phase. In the last phase, the task is assigned level by- level according to the best-discovered IWD path in order. IWDC has better makespan, cost and more economic than other algorithms such as Min-Min, Max-Min, and FCFS. It also shows better performance than PSO.

G. Hybrid multi-objective Particle Swarm Optimization

Amandeep Verma et al. [10] introduced a Hybrid Particle swarm optimization (HPSO) algorithm for managing workflow scheduling. It is a hybrid of Budget and Deadline constrained Heterogeneous Earliest Finish Time algorithm and multi-objective PSO. HPSO optimize makespan and cost under deadline and budget constraints. The results show that HPSO is better than meta-objective meta-heuristics algorithms.

H. Priority-based Performance improved algorithm for meta-task scheduling

D.I George Amalarethinam et al. [11] proposed a Meta-task Scheduling algorithm, Priority-based Performance Improved Algorithm. It uses the user priority of meta-tasks. The high priority meta-task set is scheduled based on the Min- Min algorithm and then the normal priority meta-task set is scheduled based on the Max-Min algorithm. The proposed algorithm gives minimum makespan and better resource utilization.

III. PROPOSED METHODOLOGY

The proposed algorithm is based on combination of two algorithms, task transfer policy and task information policy. The task transfer policy determines the conditions under which a task should be transferred from one node to another. Incoming tasks enter the transfer policy, which based on a rule determines the transfer of the task or processes it locally. This rule relies on the workload of each of the nodes. This policy includes task re-scheduling and task migration.

The task information policy collects all information regarding the nodes (VM) in the system and the other policies use it for making their decision. It also determines the time when the information should be gathered. If the transfer policy decides that a task should be transferred, the information is triggered in order to find a remote node for processing the task.

The relationships among different policies are as follows. Incoming tasks are intercepted by the transfer policy, which decides if they should be transferred to a node for the purpose of load balancing. If the task is not eligible for transferring, it will be processed locally. The proposed algorithm is given below:

Algorithm TTP ( )

Input: List of Tasks and Resources as VMs.

1. Initialization:

R= no. of available resources (VM)

T= Total number of incoming tasks.

$T_{L=}$ Assign incoming tasks to as per their arrival time.

2. While ($T_{L\ not}$ empty)
   If ($T_L > R$)
   Call LBP (T_L,R)
3. Else
   Assign all tasks from T_L to R.
4. End IF.
5. End While ( ).

The above algorithm will select scheduling only when the number of tasks is greater than total number of virtual machines. When it is less, the scheduling method will not called upon and the tasks will allocated to the virtual machines based on their arrival times.

Algorithm LBP (T_L ,R)

1. Do until all tasks are mapped.
2. For all task 1 to T_L
3. For all resources 1 to R
4. Arrange the tasks in order of their completion time.
   CT = start time+ ET
   Tarray[n] = {t1, t2, t3 ...tn}
5. Arrange the resources based on their non-decreasing order of MIPS.
   Rarray [m] = {R1, R2, R3........R_n}
6. Divide the resources into two groups.
   For R=1 to m/2
   Set Lowp[m]
   For R= m/2 to m
   Set Highp[m]
7. Allocate the task to VM based on completion time.
   If CT is high then
   Assign task to Highp[m]
8. Else
   Assign task to Lowp[m].
9. End if
10. End For
11. End For
12. Go to TTP ( )

13 End algorithm.

In this algorithm, processors are assigned tasks in such a way that it reduces makespan and performs load balancing thereby enhances the average resource utilization of the system. To achieve these targets, the best task mapping to the processor is found by applying LBP.

IV. RESULTS AND DISCUSSION

Simulation is one of the most popular evaluation methods in scientific workflow studies. However, existing cloud simulators fail to provide a framework that takes into consideration heterogeneous system overheads and failures. They also lack the support for widely used workflow optimization techniques such as task clustering. In this thesis, we introduce WorkflowSim1.0, which extends the existing

CloudSim simulator by providing a higher layer of workflow management.

We have conducted several experiments to test the performance of proposed algorithm in terms of response time and this section also shows performance comparison between the proposed algorithm and three existing algorithms named LBIMM, MCT and proposed Load Balancing algorithms. Proposed methods are implemented by using different workflows such as Montage50, Montage100, CyberShake30, CyberShake50, and Inspiral50. All the workflows are available into WorkowSim-1.0. Table below shows the comparisons of average execution time of three algorithms based on three workflows.

Table 1: Makespan comparison for VM=10

Workflow	Algorithm		
	LBIMM (ms)	***MCT (ms)***	***Proposed (ms)***
MONTAGE_50	41.78	14.94	12.88
CYBERSHAKE_50	129.86	101.23	41.59
INSPIRAL_50	1160.84	1132.51	725.96

Table 2: Makespan comparison for VM=20

Workflow	Algorithm		
	LBIMM (ms)	***MCT (ms)***	***Proposed (ms)***
MONTAGE_50	61.94	24.79	18.79
CYBERSHAKE_50	334.37	107.56	53.33
INSPIRAL_50	893.68	281.45	264.24

Performance of proposed algorithm is better than LBIMM and MCT for different workflows.

V. CONCLUSION AND FUTURE SCOPE

In this thesis we analysed the combinations of three load balancing algorithms and compare the three different policies in cloud computing environments. We proposed a simulation scenario for evaluating the performance of these load balancing approaches. By these combinations, we generate the different possible load balancing approaches which simulated each one with different workloads. Finally we achieve different simulated results. Through these results we compare the performance of load balancing in cloud computing in terms of average response time.

We analysed the performance of these approaches by simulating on WorkFlowSim simulator. The simulation results shows that proposed task transfer and resource selection algorithm have a better performance than other load balancing algorithms, because it uses a threshold and available VM list for preventing serve the workload by overloaded VMs. In addition we analysed and offered the best combinations of each VM load balancer with different datasets. As the future works we will expand these experimental results by evaluating the more VM load balancers in cloud computing and under the different scenarios by considering the more evaluation factors and parameters.

REFERENCES

[1] Peter Mell, Timothy Grance, "The NIST definition of Cloud Computing (September, 2011)", Accessed on May, 2014.

[2] Maryam Masoudi Khorsand, Mehdi Effatparvar and Mahdi Kanani, "A survey of Scheduling Algorithms in Grid Computing," International Journal of Research in Computer Applications and Robotics, ISSN 2320-7345, Volume 2, Issue 2, Pg. 118-126,February 2014.

[3] Mahajan, Komal, Ansuyia Makroo, and Deepak Dahiya. "Round Robin with server affinity: a VM load balancing algorithm for cloud based infrastructure. "Journal of information processing systems 9.3 (2013): 379-39.

[4] Liang, Po-Huei, and Jiann-Min Yang. "Evaluation of Cloud Hybrid Load Balancer (CHLB)." International Journal of E-Business Development (2013).

[5] Moharana, Shanti Swaroop, Rajadeepan D. Ramesh, and Digamber Powar. "Analysis of load balancers in cloud computing." International Journal of Computer Science and Engineering 2.2 (2013): 101-108.

[6] Domanal, Shridhar G., and G. Ram Mohana Reddy. "Load Balancing in Cloud Computing using Modified Throttled Algorithm." Cloud Computing in Emerging Markets (CCEM), 2013 IEEE International Conference on. IEEE, 2013.

[7] Nakrani, Sunil, and Craig Tovey. "On honey bees and dynamic server allocation in internet hosting centers." Adaptive Behavior 12.3-4 (2004): 223-240.

[8] Nishant, Kumar, et al. "Load balancing of nodes in cloud using ant colony optimization." Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on. IEEE, 2012.

[9] R.Achar, Optimal Scheduling of Computational task in Cloud using Virtual Machine Tree, Third International Conference on Emerging Applications of Information Technology(EAIT), IEEE, 2012.

[10] Tushar Desai, Jignesh Prajapati "A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing" IJSTR, 2013.

[11] Reena Panwar , Prof. Dr. Bhawna Mallick "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm" , IEEE 2015.

AUTHORS PROFILE

Ms. Aakriti Pathak is pursuing Master of Engineering in CSE branch from Shri Ram Institute Of Technology, Jabalpur, M.P.

Prof. Anshul Khurana is currently working as a Asst. Prof. In CSE deptt. SRIT ,Jabalpur,M.P.
He has done B.Tech and M.tech. in CSE branch.