"Task Scheduling of Load Balancing in Cloud Computing Environment for Increasing Response Time and Reducing Cost"

Kanika Patel
Guide: Dr. Suresh Chand Gupta
College: Panipat Institute of Engineering and Technology

Abstract- Cloud computing is a developing technology in today's Internet world which offers the users with on demand access to resources through different service models. In spite of providing many advantages over the traditional computing, there are some critical issues in cloud computing. Load balancing is a crucial issue in cloud computing that distributes the user's requests to the nodes in such a manner to balance the load on nodes. A proper load balancing algorithm is required to execute the process and manage the resources. The paper proposes a dynamic load balancing algorithm for cloud computing environment and compares with the existing algorithm. The results show that proposed algorithm outperforms existing algorithm in terms of average response time, turnaround time, and total cost.

Keywords- Cloud computing, Load balancing, Virtual machine, CloudSim.

I. INTRODUCTION

Cloud computing is a great revolution in the IT industry and in its development trends. Cloud computing provides heterogeneous services such as application, servers, and storage to different peoples through the Internet. Cloud computing services are used by individuals and businesses to access the application from anywhere in the world on demand. Cloud term is used for the cloud service provider that holds the resources for storing and accessing the data [2]. Cloud computing is an Internet-based computing and very successful because of its characteristics like on demand service, pool of resources, ubiquitous network access, measured service, and rapid elasticity. Cloud provides the services such as IaaS, PaaS, SaaS to clients. The IaaS delivers the cloud computing infrastructure storage, server and operating system to users on rent basis for various purposes. In case of platform as a Service (PaaS), it provides the environment for users to develop, test, host, and maintain their applications. In case of software as a service, software is hosted by cloud service provider and made available to clients over the Internet Cloud computing is adopted by many industries such as social networking websites and online applications so number of users is increasing to reach cloud. Load balancing is a main requirement for managing the performance of cloud and properly utilizing the resources. Load balancing is an optimization technique that is used to allocate the workload across the different nodes. The load is assigned in an efficient manner to enhance the response time of task and to achieve the efficient resource utilization [2, 3]. In this study, we aim to develop an optimized load balancing algorithm which gives the maximum benefit to the cloud service provider.

Load Balancing in Cloud Computing: Cloud computing is composed of several different resources, interconnected to each other to form a network or a grid. These resources should be flexible and dynamic in terms of usage and allocations. In cloud computing, load balancing is one of the major techniques that has a dramatic impact on resource availability. The term availability, was always a main concern in cloud-computing. Fundamentally, availability explains the ubiquitousness of the network information in case of resource scaling. Load balancing could be illustrated, as proper strategy for task scheduling that will lead to balanced load distribution in cloud networks. It is an important key to improve the network performance. Moreover load balancing algorithm can minimize the response time while utilizing the resource usage. The lack of proper load management can create traffics due to the long waiting time for accessing the resources. Today most of the cloud vendors are trying, to use automated load balancer to enable the users, scale the numbers of their resources automatically. Promoting the availability and performance of the cloud system highlights the main goal of the automated load balancers. To design an effective load balancing algorithm, Dillon [5] suggested the following strategies:

• Load balancing algorithm should be smart enough to make load balancing decisions in a right time; • Depends on behaviour of the application load balancer should be able to gather the information locally and globally;

• Load balancer should be designed in a centralized or distributed pattern. If the load balancer is centralized then there is a less opportunity for scalability purposes;

• Local load balancers are costing less, but the information provided by global load balancer is more accurate.

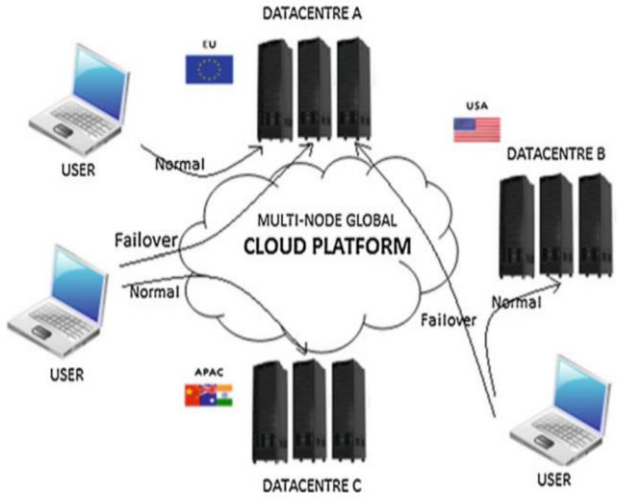


Fig.1: Diagram for load balancing

II. RELATED WORK

The paper [3] proposed a task-based load balancing algorithm based on QoS driven in cloud computing. This algorithm considers priority and completion time. The paper [4] proposed a VM-assign load balance algorithm to find out the least loaded VM using the index table and the load is assigned to that VM if it is not the last used VM. The paper [5] proposed a cloud light weight algorithm which not only balances the load but also ensures QoS to the users. The paper [6] proposed an algorithm, which considers only the response time of the incoming request and reduces communication cost and extra computation at the server. The paper [7] proposed a heuristic model for load balancing based on the utilized power of the virtual machine. It considers the heterogeneous nature of the VMs and the cloud computing system.

Belgaum, M. R., et al[1] The load balancer implements the virtualization technique [10] where the resources of the physical server at the data centers [11] are allocated to a wide range of virtual machines each of which performs the function of a physical server Whenever the checkpointing occurs, the load of the physical server is analyzed, and without preempting the resources from the available resources, they are allocated to the virtual machine to accomplish the task. At any instance of time, the resources required or allocated must not be more than available.

Kaur, R., et al[2] Although a lot of fundamental research is being carried out in a field of cloud computing, but still it is in infancy stage. In today's era of computing, it is trending Internet technology, which suffers from various issues and challenges. This research addresses load balancing as one of the major challenges in cloud computing.

Chawla, A., et al[3] The proposed algorithm works best when no fault occurs in a VM. In future, an algorithm will be developed which automatically creates the migration of VM. The research work has been tested on a simulator. The results might differ in case of real cloud environment.

Dam, S., et al[4] In this paper, soft computing based algorithm on ant colony optimization has been proposed to initiate the load balancing under cloud computing architecture. Detail analysis of the results, indicates that the proposed strategy for load balancing not only outperforms a few existing techniques but also guarantees the QoS requirement of customer job. Though fault tolerance issues does not consider and all jobs are predicted with same priority here, which may not be the actual scenario. Researchers can proceed to include the fault tolerance and different function variation to calculate the pheromone value can be used for further research work.

Kumar, N., et al[5] In cloud computing environment, the major issue is how to distribute workload across all processing nodes so that all nodes gets equal job at a given instant of time. So this paper proposed fuzzy load balancing algorithm for solving load balancing problem in fuzzy cloud computing environment. The simulation result shows the comparison between fuzzy Hungarian and fuzzy row penalty method in terms of response time and space complexity in order to provide latest approach.

III. PROPOSED METHODOLOGY

Cloud computing is an emerging region that present a lot of possible benefits to different organizations and frequent user. It is the extended appearance of distributed computing. . It is based on on-demand-service model in which in sequence, software, infrastructure and previous services are offer as per the client prerequisite at a number of instance of time. Load balancing is utilized to distribute workload amongst multiple cloud systems or nodes to acquire enhanced resource utilization. It is the important means to accomplish proficient resource contribution and exploitation. Load balancing has happen to a challenge issue currently in cloud computing system. To get together the users vast number of demands, there is a require of distributed resolution since almost it is not forever probable or cost proficient to handle one or additional inoperative services. To tackle problems by Swarm based algorithm use nature inspired Optimization technique. To develop an efficient load balancing algorithm with hybridisation of partial swam optimization technique and Genetic algorithm. It is not to function on the resolution pool except to create a convinced coding denotation. So primary we require doing the coding for the difficulty to be tackled. The selection of the coding process to an enormous extent depends on the property of the difficulty and the intend of genetic hybridisation of partial swam optimization technique and Genetic algorithm. The classic genetic algorithm consequence the chromosome establishment of genes by binary codes. Evaluator from the data representation in this paper, it can be establish that it is a one-to-many mapping association among physical machines and VMs. consequently, this paper prefer tree structure to mark the chromosome of genes. That is towards say, each mapping solution is manifest as one tree, the scheduling and supervision node of the system on the primary level are the root nodes even as every of the N nodes on the instant level get up for physical machines and the M nodes on the third plane get up for the VMs on a convinced physical

machines. For the initialization of population, this paper mainly uses the method of spanning tree based on partial swam optimization technique and Genetic algorithm. We have the subsequent definition for the tree: This tree is a spanning tree build by the elements in the physical machine set and VM set. The root node of this tree is the predefined supervision source node. All of the physical machine nodes and VM nodes are built-in in this tree. Every of the leaf nodes are VM nodes. The pattern of the spanning tree is that it has to get together the specified load balancing conditions or it should create comparatively fine descendants throughout inheritance. This means the tree itself should as well be a moderately well individual. Consequently we can acquire the mapping relationship among physical machines and VMs during the subsequent procedures. primary, we compute the selection probability p (p is the proportion of a single VM consignment to the load sum of each the VMs) of every VM according to the VM load in the VM set; subsequent to that based on the probability p all of the logic disks are owed to the smallest number of loaded node in the physical machine set to construct the leaf node of the initial spanning tree. In that method the opening of those VM with added warm creature certain is increase and those VM through low heat can as fine is selected.

IV. PROPOSED ALGORITHM

PSO is a swarm based heuristic optimization technique. It is used for identifying the optimal path of solution space. While putting up the load on specific virtual machine for processing of the resources, it moves along all the virtual machines and identifies the optimal machine to put the load. It is one of the mechanisms to identify the optimal V.M, which is load less, available and task map. So the relative energy and time utilization to process the node can be reduced.

Basic Steps for PSO:

1. Initialize population of particles with random position and velocities.

2. Calculate the fitness function value for each and every particle.

3. Compare current particle's fitness value with each particle's fitness value and find Pbest value.

There is variety of algorithms designed for balancing the load between various works. After completing the literature survey we are able to conclude that most of the load balancing algorithms proposed so far is complex, and not able to implement. PSO algorithm task will assign to the virtual machine in best fit manner [11]. I.e. task will check all the VM and assigns the task to proper VM which will have least memory wastage. User sends their task request to the cloud server that decides the VM to store the task. Cloud server will select the VM based on PSO algorithm. Our aim is to balance the load when there is an overload in VM. First step is to upload the file and cloud server will accept the request and it will transfer that request to VM. User control will initiate the process and give control to the VM Scheduler. Main function of VM Scheduler is performs the load balancing using PSO algorithm. Based on threshold value only we are finding the overloaded VM. After finding the overloaded VM next step is to migrate task from overloaded virtual machine to under loaded virtual machine.

Improved PSO Algorithm:

1. InitializepBest, gBest and p,S with 0s

2. Call initialize () [for particle generation]

3. Repeat while (false)

4. IfpBest<gBest

5. gBest = pBest

6. Do for i□0 to S

particle.get (i)

iftestProblem (i) =Target

Set condition true for while

7. pBest= minimum() [minimum in the particle]

8. Particle.get(gBest)

9.If Target_testProblem(pBest)<Target_testProblem(p)

p = pBest

10. AgetVelocity(gBest) [retrieve

Velocity of gBest particle]

b) Updateparticle (gBest) [update

Particle with effect to gBest]

c) S++ [make an increment in S's

Current value]

11. Else set condition true for while

12. End of function

V. WORKING OF PROPOSED ALGORITHM

The heuristic searching algorithm intends to discover the optimized consequence based on the survival of the fittest being theory. The objective is to preserve the balance load over the virtual machine (VM) and as well to reduce the create span time. Subsequent are the necessary steps of Genetic Algorithm.

Algorithm 1 input: MaxMut: Maximum Mutation Rate

PS: Population Scale

WN: quantity of Worker Node

G: Number of Jobs

W1, W2: entirety jobs weight and the standard jobs spanning weight W1+W2=1

α1, α2: Probability of fitness 1 likelihood of fitness 2 Flag = 0

Output: Elite1, WN: Best Solution

Mutation counter, α 0,

fitness1, S Φ

Elite1, WN,temp, Z1, Z2, PS, WN, Fitness1, S Φ

Z dijkstra's initialization

While mutation counter <= MaxMut

Do • α= random (0,1)

If α< α2 then

fitness = fitness 2 •

else

fitness= fitness 1

end if •

fori=1 to S do

fitnessi fitness(i)

End for

Elite1,WN individual with best fitness

W1, W2

Calculate pc

Ifrandom (0, 1) < pc then

temp1, Ncrossover (Z1, Z2)

End if

Ifrandom (0, 1) < pm then

Z1, WN mutation (temp)

End if

If (prev fit ==novel fit)

Elite1, WN Best solution

Flag ++

End if

Prev fit ==novel fit

If(flag==10)

Mutation count ++

Particle swarm creates particles at random uniformly within bounds. If there is an unbounded component, particles warm creates particles with a random uniform distribution.Particle swarm shifts the creation to have the bound as an endpoint. In proposed approach particle swarm calculates initial response time and cost of VM. It evaluates the objective function at all VM. It records the current response time res (i) of each cloudlet at VM i. In subsequent iterations, res (i) will be the location of the best objective function that has been found. The approach updates the swarm as follows. For response i and cost j, which is at VM (i): Choose a random subset S of N values other than i. Find fbest(S), the best objective response function and cost(S) with the best objective cost function.
Parameters

- User
- Cloudlet
- Data centres
- Virtual Machine Manager or VM Manager (VMM)
- VMM creates the VM on the basis of resources
- Virtual Machine (VM)
- Resource Provisional (RsP)
- Resource Provider or Resource Owner (RP) after finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).

res[] = res[] + c1 * rand() * (pbest[] - cost[]) + c2 * rand() * (gbest[] - cost[]) (a)
cost[] = cost[] + res[] (b)

res[] is the VM response time, cost[] is the current VM running cost. pbest[] and gbest[] are defined as stated before and rand () is a random number between (0,1). c1, c2 are learning factors. c1 = c2 = 2.

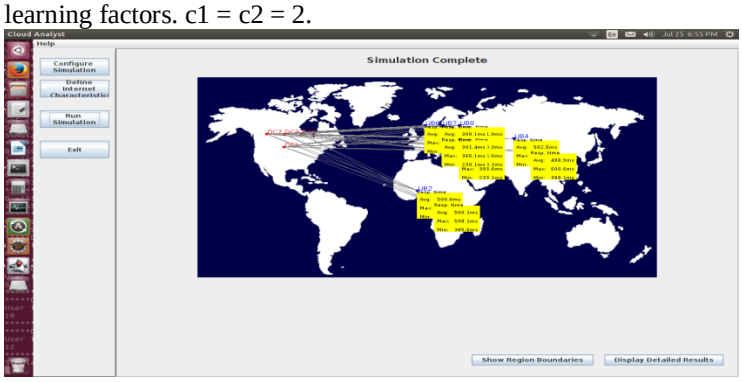
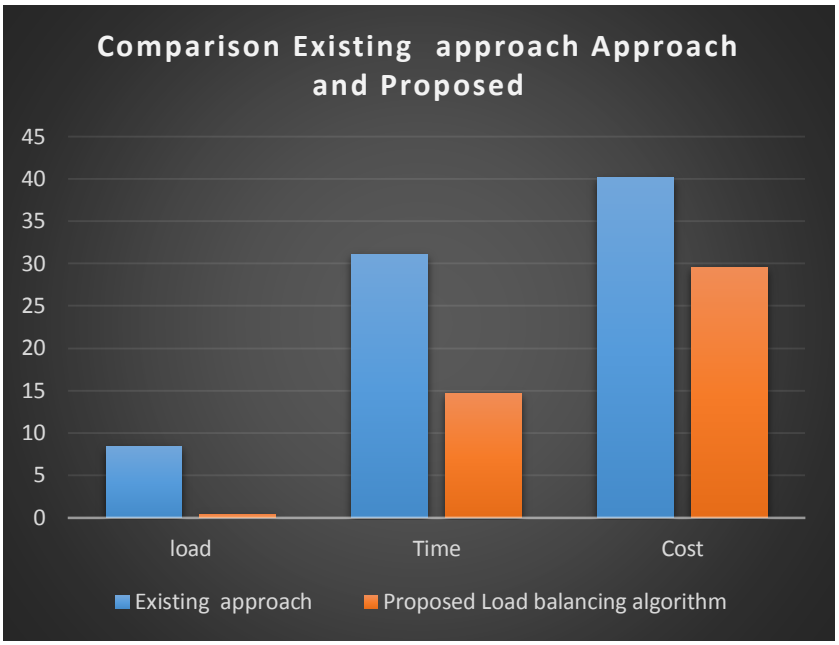


Fig.1: Completed the Simulation Display the Results on GUI



Existing algorithm necessitate low communication and its operational is fair. Afford a detailed comparison of dissimilar algorithms over dissimilar parameters like fairness, concert, speed, complexity. Our proposed algorithm is additional proficient according to subsequent facts, swarm based heuristic optimization technique believe fair to distribute the load; it has high throughput, good response time and less complex than previous algorithms. The foremost advantage of swarm based heuristic optimization technique is time limitation and utilize equal period to entire every task. To illustrate during experiment identify that swarm based heuristic optimization technique is enhanced than Existing algorithm in stipulations of total execution time and cost. as well, swarm based heuristic optimization technique has superior load balancing performance

VI. CONCLUSION

In this paper, cloud service ranking has been done in hybrid load balancing in the real-time scheduling. In this framework, the user can access the services with higher rank by communicating with the users which have accessed the services in the past. By applying this technique, ranking of services can be improved and there will be less overhead in the cloud. Simple ranking technique has been used in this framework. To improve the service ranking, other approaches can also be included that are based on rating the services. Other techniques can also be included to negate the checkpoint-based load balancing in real-time environment which reduce the prediction of services. Load balancing is a serious issue in cloud computing. The paper attempts to highlight the issue of load balancing. The paper analyses a proposed load balancing algorithm and also compares its performance with round-robin algorithm. Finally, this paper highlights the research gap and gives an approach to overcome it.

VII. REFERENCE

[1]. Belgaum, M. R., Soomro, S., Alansari, Z., & Alam, M. (2018). Cloud Service Ranking Using Checkpoint-Based Load Balancing in Real-Time Scheduling of Cloud Computing. Progress in Advanced Computing and Intelligent Engineering, 667–676. doi:10.1007/978-981-10-6872-0_64.

[2]. Kaur, R., & Ghumman, N. S. (2017). Task-Based Load Balancing Algorithm by Efficient Utilization of VMs in Cloud Computing. Big Data Analytics, 55–61. doi:10.1007/978-981-10-6620-7_7.

[3]. Chawla, A., & Ghumman, N. S. (2017). Package-Based Approach for Load Balancing in Cloud Computing. Big Data Analytics, 71–77. doi:10.1007/978-981-10-6620-7_9.

[4]. Dam, S., Mandal, G., Dasgupta, K., & Dutta, P. (2014). An Ant Colony Based Load Balancing Strategy in Cloud Computing. Smart Innovation, Systems and Technologies, 403–413. doi:10.1007/978-3-319-07350-7_45.

[5]. Kumar, N., & Shukla, D. (2017). Load Balancing Mechanism Using Fuzzy Row Penalty Method in Cloud Computing Environment. Lecture Notes in Networks and Systems, 365–373. doi:10.1007/978-981-10-3932-4_38.

[6]. Kaur, R., & Ghumman, N. S. (2017). A Load Balancing Algorithm Based on

[7]. Processing Capacities of VMs in Cloud Computing. Big Data Analytics, 63–69. doi:10.1007/978-981-10-6620-7_8

[8]. Zhu, Y., Zhao, D., Wang, W., & He, H. (2016). A Novel Load Balancing Algorithm Based on Improved Particle Swarm Optimization in Cloud Computing Environment. Lecture Notes in Computer Science, 634–645. doi:10.1007/978-3-319-31854-7_57.

[9]. Aslanzadeh, S., Chaczko, Z., & Chiu, C. (2015). Cloud Computing—Effect of Evolutionary Algorithm on Load Balancing. Computational Intelligence and Efficiency in Engineering Systems, 217–225. doi:10.1007/978-3-319-15720-7_16.

[10]. Khan, R. Z., & Ahmad, M. O. (2016). Load Balancing Challenges in Cloud Computing: A Survey. Proceedings of the International Conference on Signal, Networks, Computing, and Systems, 25–32. doi:10.1007/978-81-322-3589-7_3.

[11]. Hota, A., Mohapatra, S., & Mohanty, S. (2018). Survey of Different Load Balancing Approach-Based Algorithms in Cloud Computing: A Comprehensive Review. Computational Intelligence in Data Mining, 99–110. doi:10.1007/978-981-10-8055-5_10

[12]. Fang, Y., Wang, F., & Ge, J. (2010). A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing. Lecture Notes in Computer Science, 271–277. doi:10.1007/978-3-642-16515-3_34

[13]. Bhandari, A., & Kaur, K. (2018). An Enhanced Post-migration Algorithm for Dynamic Load Balancing in Cloud Computing Environment. Proceedings of International Ethical Hacking Conference 2018, 59–73. doi:10.1007/978-981-13-1544-2_6

[14]. Tang, L. (2018). Load Balancing Optimization in Cloud Computing Based on Task Scheduling. 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS). doi:10.1109/icvris.2018.00036