JOURNAL OF CRITICAL REVIEWS

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

ACHIEVING OPTIMAL RESOURCE ALLOCATION IN CLOUD ENVIRONMENTS: A ROBUST LOAD BALANCING STRATEGY USING ENHANCED PSO AND VM LOAD BALANCING TECHNIQUES

Somesh Dnyandeorao Ghuge[1]*, Parag Digambarrao Thakare[2]

[1]*Scholar, [2]Assistant Professor
[1]*,[2]Department of Computer Engineering, Jagadambha College of Engineering & Technology, Yavatmal
[1]*Email: someshghuge62@gmail.com, [2]Email: spt2116@gmail.com

***Corresponding Author:** Somesh Dnyandeorao Ghuge
*Email: someshghuge62@gmail.com

Abstract:
Cloud computing has revolutionized the way resources are managed and utilized in modern IT infrastructures. Efficient resource allocation and load balancing are crucial for ensuring optimal performance and resource utilization in cloud environments. This paper presents a robust load balancing strategy that integrates an enhanced Particle Swarm Optimization (PSO) algorithm with Virtual Machine (VM) load balancing techniques to achieve optimal resource allocation in cloud environments. The proposed approach aims to minimize response time, maximize throughput, and improve resource utilization by dynamically allocating resources based on workload demands. Experimental results demonstrate the effectiveness and efficiency of the proposed strategy in achieving optimal resource allocation and enhancing the overall performance of cloud systems.

Keywords: Cloud computing, Resource allocation, Load balancing, Particle Swarm Optimization (PSO), Virtual Machine (VM), Optimization, Performance, Scalability.

1. Introduction:

Cloud computing has emerged as a dominant paradigm for delivering on-demand computing resources over the internet. It offers scalability, flexibility, and cost-effectiveness by allowing users to access a pool of shared computing resources, including storage, processing power, and applications, as services. However, efficient resource allocation and load balancing are critical for ensuring high performance, reliability, and scalability in cloud environments. Resource allocation involves assigning available resources to tasks or services based on their requirements, while load balancing aims to evenly distribute the workload across resources to avoid bottlenecks and optimize resource utilization.

Traditional approaches to resource allocation and load balancing in cloud environments often rely on static or heuristic-based methods, which may not adapt well to dynamic workload changes and may result in suboptimal resource utilization. Therefore, there is a need for more intelligent and adaptive techniques that can dynamically allocate resources based on workload demands and optimize the overall performance of cloud systems.

In this paper, we propose a robust load balancing strategy that combines an enhanced Particle Swarm Optimization (PSO) algorithm with Virtual Machine (VM) load balancing techniques to achieve optimal resource allocation in cloud environments. The enhanced PSO algorithm is designed to efficiently search the solution space and adaptively adjust resource allocation based on workload changes, while VM load balancing techniques help distribute the workload across VMs to avoid resource overloading and underutilization.

2. Literature Review:

Various techniques have been proposed in the literature for resource allocation and load balancing in cloud environments. Traditional approaches include static partitioning, round-robin scheduling, and threshold-based methods. However, these approaches may not be suitable for dynamic and heterogeneous workloads, leading to suboptimal resource utilization and performance degradation.
Particle Swarm Optimization (PSO) is a population-based optimization technique inspired by the social behavior of birds flocking and fish schooling. It has been widely used in various optimization problems, including resource allocation and load balancing in cloud computing. PSO algorithms aim to iteratively update the position of particles in the search space to find the optimal solution based on a predefined fitness function.

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

Virtual Machine (VM) load balancing techniques involve distributing the workload across VMs in a cloud environment to ensure optimal resource utilization and performance. These techniques may include dynamic VM migration, task scheduling, and resource provisioning based on workload characteristics and system constraints.

3. Proposed Algorithm:

The proposed load balancing strategy consists of the following steps:
Algorithm: VM-Assign Load Balancer Input:
No of incoming jobs j], J2..... jn Available VM k], K2••• ..... kn
Output: All incoming jobs j], J2..... Jn are allocated smallest amount loaded virtual machine between the accessible KhK2•••••••• .kn
1: Originally all the VM's have 0 allocations.
2: VM allocate load balancer preserve the index / allocate table of VMs which has no. of requests at present allocated to every VM.
3: When requests appear at the data canter it pass to the load balancer.
4: Index table is parsed and smallest amount loaded VM is particular for execution.
Case I: if establish a. ensure whether the selected least loaded VM is used instantly in the last iteration.
If YES go to step
4 to discover subsequently least VM if NO Least loaded VM is chosen
5: VM-assign load balancer precedes the VM id to the data center.
6: Request is allocate to the VM. Data center alert the VM-assign load balancer about the allowance.
7: VM allocate load balancer modernize the requirements hold by each VM.The enhanced PSO algorithm dynamically adjusts resource allocation based on workload changes and optimizes the overall performance of cloud systems by minimizing response time, maximizing throughput, and improving resource utilization.

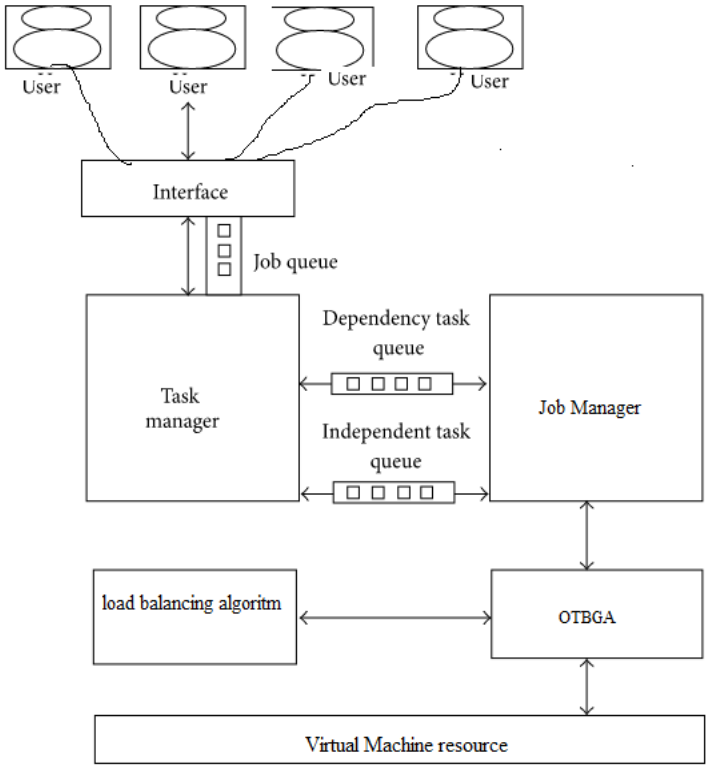

Fig-1 Working of Proposed Algorithm

4. Simulation and Result Analysis:

Cloud Analyst (Implementation Tool):

To implement our proposed technique in Java programming language Subsequent to the coding of algorithm, the code is compile by JDK version 6 and create class file is novel for implementation in Cloud Analyst tool. Operation system is Ubuntu 16.04, CPU is Intel® Core™ 2 Duo 3.0GHz and Memory is 2.0GB, and the disk capacity is 320GB Table 1 illustrate the evaluation of the discussed Load Balancing (LB) algorithms during dissimilar parameters similar to sprite, throughput, waiting Time. The comparison of these algorithms illustrates positive and negative consequence and we illustrate this as high and low term. As discussed pervious dissimilar algorithms show diverse results. Such that, Static algorithm believe fair to allocate the load. Except it is a reduced amount of multifaceted and not fault tolerant. Existing

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

algorithm is not fair and fault tolerant. In case of little tasks, it illustrates greatest result. In Existing algorithm, needs are prior identified. So it works enhanced and provide high throughput. All along with this, dynamic load balancing necessitate only current state of the system and has additional overhead and fault tolerance. Our proposed technique has high throughput and low response time. It has low overhead and performance since high priority tasks cannot effort exclusive of VM machine.

Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions. The choice of PSO parameters can have a large impact on optimization performance. Selecting PSO parameters that yield good performance has therefore been the subject of much research. The topology of the swarm defines the subset of particles with which each particle can exchange information. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solution, called particle, fly by the problem gap by following the present optimal particle. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The strength value is too stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbours of the particle. This location is called lbest. When an element takes all the populace as its topological neighbours, the top value is a worldwide finest and is called gbest.
The proposed approach uses particle swarm algorithm. It creates the initial data samples of response time and VM cost. It evaluates the objective function at each VM. It finds out the best lowest value of operation. It selects the new values based on current data. It then iteratively updates the response time VM cost.
Initialization:-

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.

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

```
UB2 requests sent=6187 , received=6187
UB4 finalizing. Messages sent:628, Received:628
UB4 requests sent=6006 , received=6006
UB3 finalizing. Messages sent:622, Received:622
UB3 requests sent=5946 , received=5946
Got response for 2300628 but it seems to be completed.
DC1-Broker finalizing, submitted cloudlets=1694 processing cloudlets=0 ,allRequestsProcessed=16202
UB1 finalizing. Messages sent:631, Received:631
UB1 requests sent=6058 , received=6058
Got response for 2300635 but it seems to be completed.
DC5-Broker finalizing, submitted cloudlets=1590 processing cloudlets=0 ,allRequestsProcessed=15237
Got response for 2300624 but it seems to be completed.
DC3-Broker finalizing, submitted cloudlets=1549 processing cloudlets=0 ,allRequestsProcessed=14841
Simulation completed.
************ Vm allocations in DC1
0->1696
1->1695
************ Vm allocations in DC2
0->1539
1->1538
************ Vm allocations in DC3
0->1551
1->1550
************ Vm allocations in DC5
0->1592
1->1591
*****Datacenter: DC1*****
User id         Debt
6               2051.2
**********************************
*****Datacenter: DC2*****
User id         Debt
8               2051.2
**********************************
*****Datacenter: DC3*****
User id         Debt
10              2051.2
**********************************
*****Datacenter: DC5*****
User id         Debt
12              2051.2
**********************************
Simulation finished at 3600100.0
upendra@upendra-Compaq-Presario-C700-Notebook-PC:~/exper$
```

Figure 4.1: Running Snapshot Proposed Systems

Configuration Simulation separated in three parts, Main Configuration where we can setup parameter similar to as simulation duration, user base, service broker policy, application configuration. Fig 4 illustrates the parameters values we have prefer for our experiment.

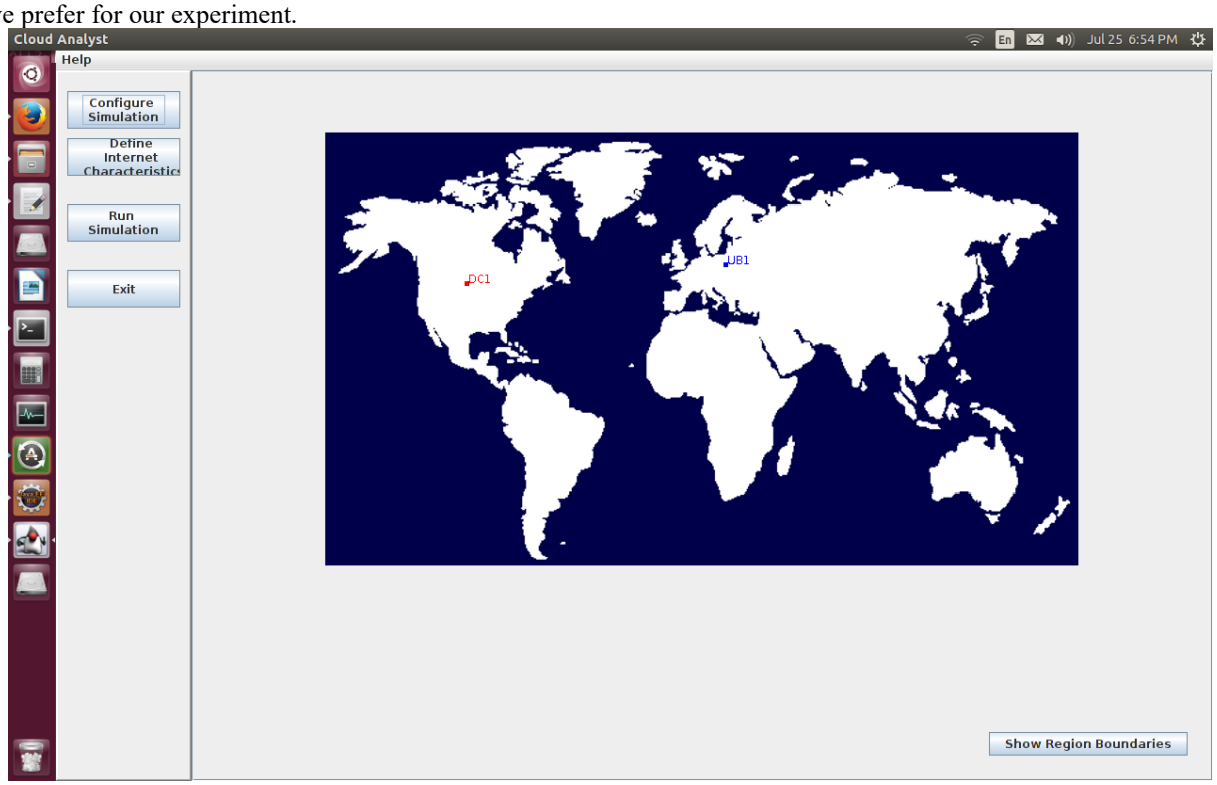

Figure 4.2: Graphical User Interface

We did an experiment in facility broker policy of cloud analyst, the experiment comprise sorting and subsequent to sorting map function will run to map the user bases through the data centre. Service broker policy is the policy by which an algorithm chooses to distribute load among the data centre. We use optimize response time facility broker policy in which data centre prefer according to their response time. We be relevant a sorting in the optimize response time service broker

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

policy and then discover out the results and evaluate with the result which is devoid of sorting .we are using our proposed algorithm for distribution of load.

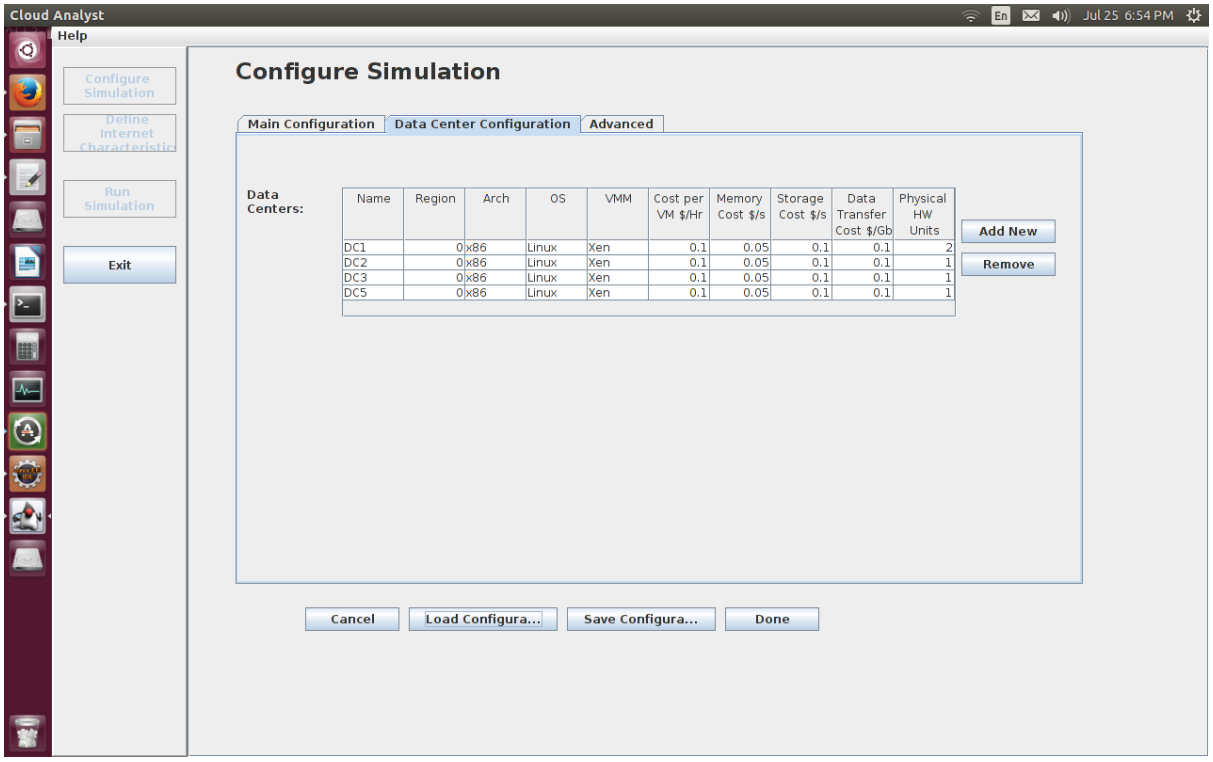

Figure 4.3: VM Allocation Using Existing and Proposed System

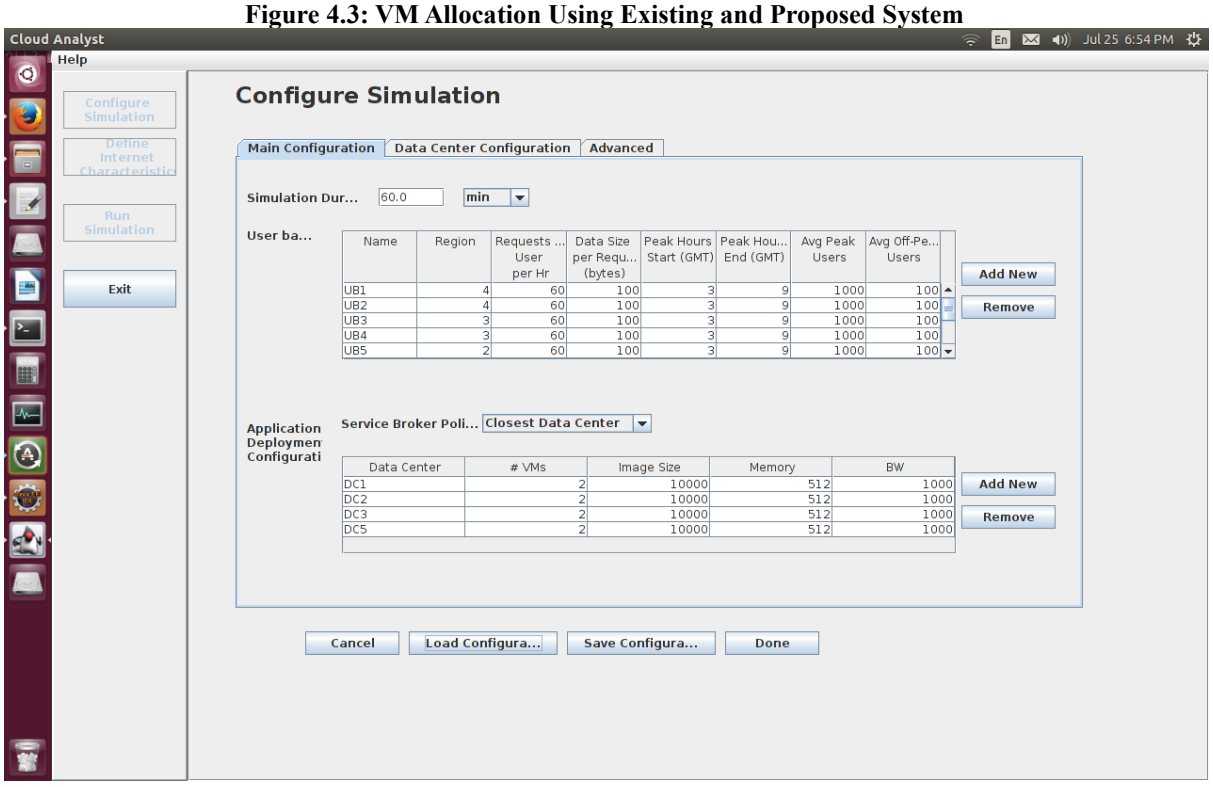

Figure 4.4: Configure the Overall Simulation

The brokering policies we obtain in to the deliberation to evaluate existing and proposed throttled algorithm is Optimize Response Time. Simulation period is set to 60 minutes. A variety of scenarios are formed by conveying dissimilar component values for User base and Deployment configuration. In primary situation there are three data center and Ten User Base is formed as shown in Fig . And in instant two data centre and five user base is formed for analysis. Data centre configuration has to be completed previous two main configurations as data centre created in this tab will be displayed for

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

collection in major configuration. Data centre region, architecture it's effective on, operating system, cost for accounting, and dealing out elements such as cores, memory, processor speed and VM-policy can be configured as revealed in data centre allocation with existing and proposed System.

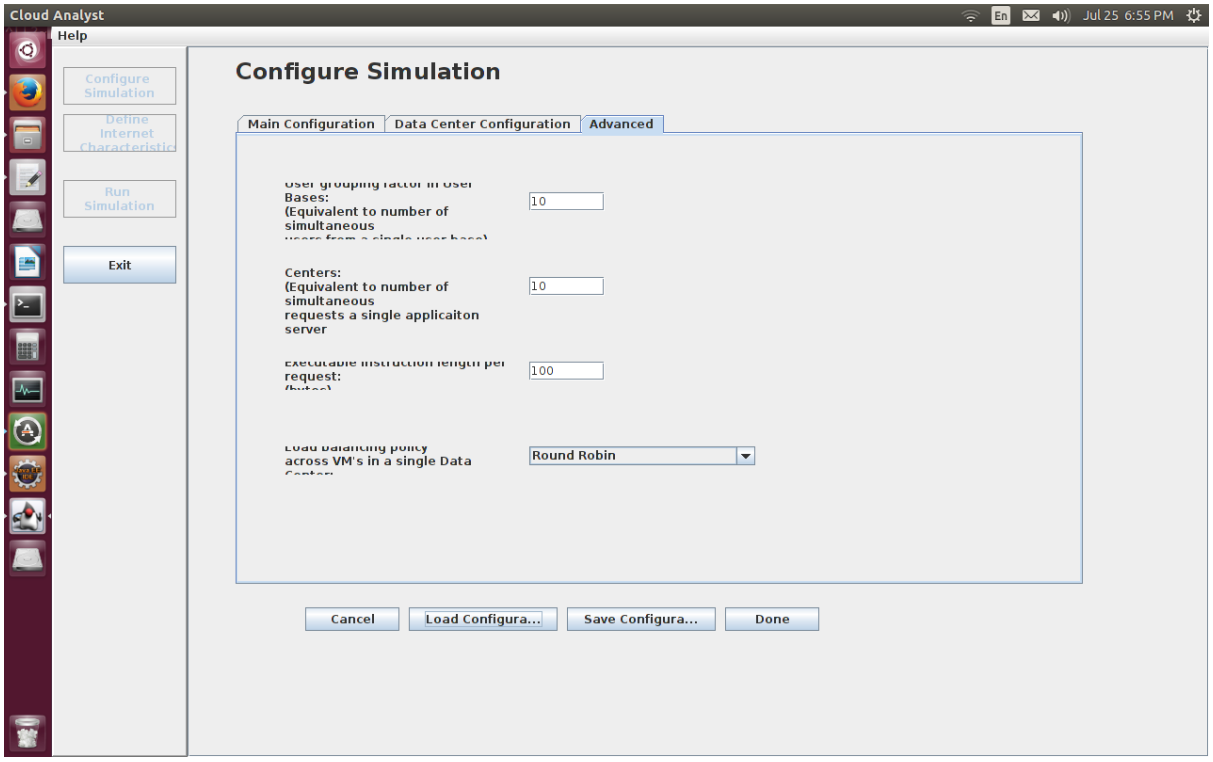

Figure 4.5: Select the Proposed Algorithm

For our experiment we can choose dissimilar load balancing policy from the drop-down. As we have created novel policy for enhanced Throttle Algorithm we can choose it from propped system simulation.

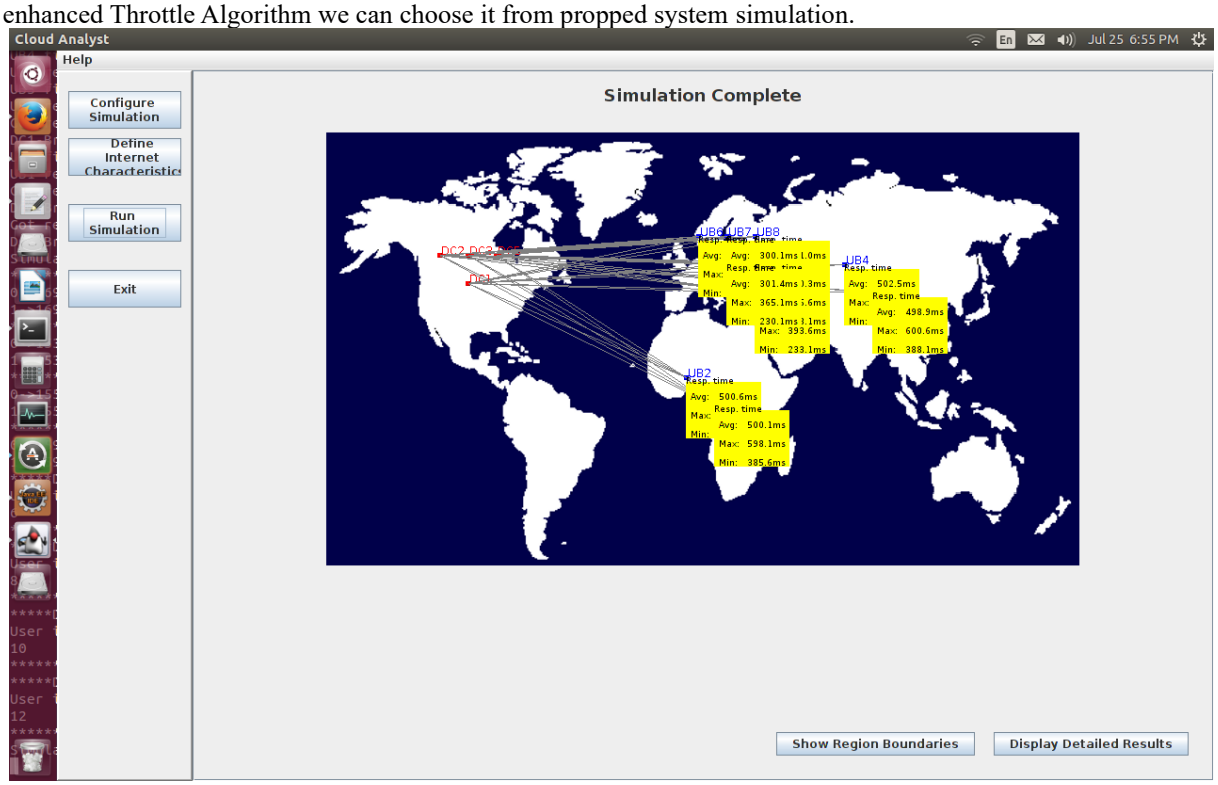

Figure 5.6: C4mpleted the Simulation Display the Results on GUI

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

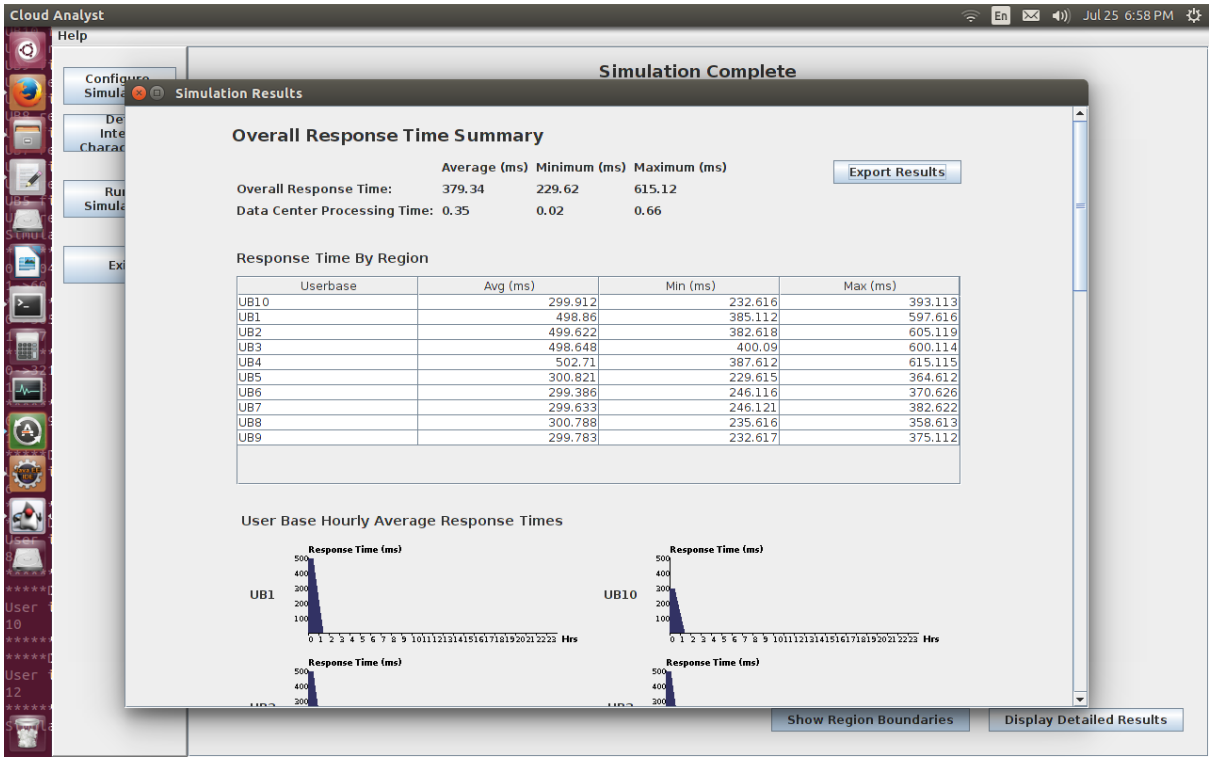

Figure 4.7: Overall Response Time Summary

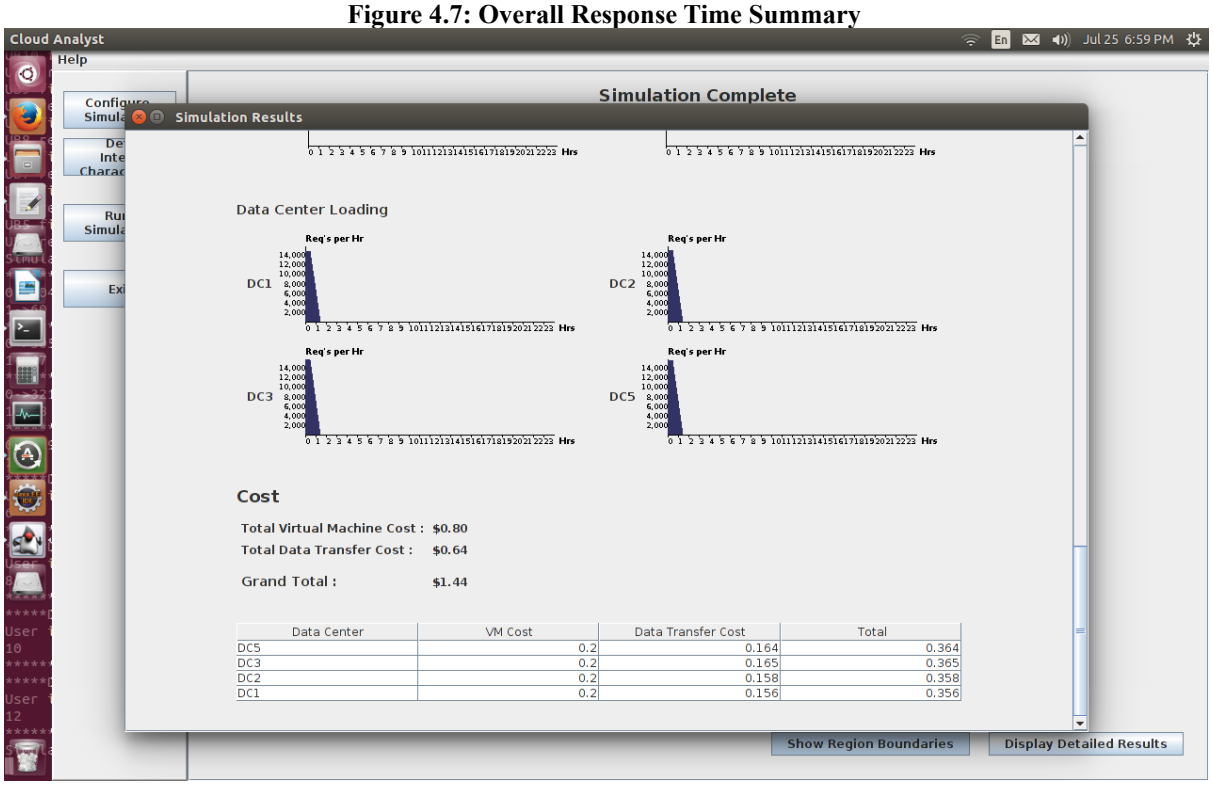

Figure 4.8 Show the Data Centre Loading and Cost

The proposed algorithm implemented for simulation. Java language is use for implementing VM load balancing algorithm. The consequence based on Particle Swarm Optimization Approach VM Load Balancing algorithm for overall response time of the cloud. In this min (ms) time, max (ms) time to different number of virtual machines are analysed. Illustrate the consequence based on proposed algorithm VM Load Balancing algorithm for data Centre giving out time of the cloud. In this min (ms) time, max (ms) time to dissimilar number of virtual machines are analyse.

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

Algorithm	load	Time	Cost
Existing approach	8.4121	31.1427	40.1428
Proposed Load balancing algorithm	0.3261	14.7390	29.5175

Table 4.1: The Performance Comparison Traditional Approach and Proposed

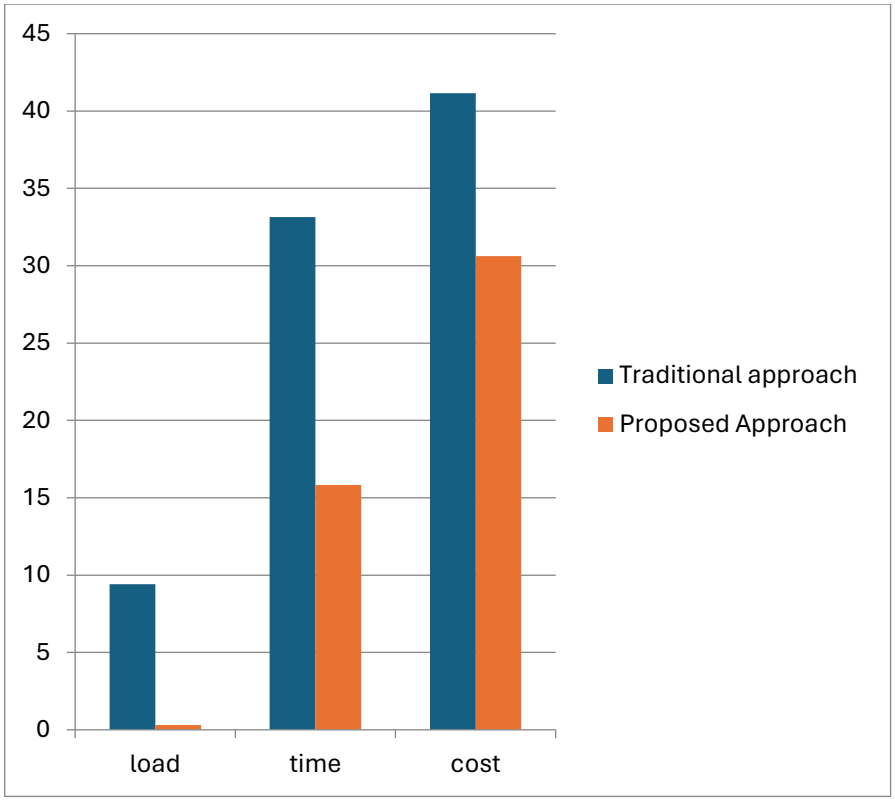

Figure 4.9: The Performance Comparison Traditional Approach and Proposed Approach

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, OTBGA believe fair to distribute the load; it has high throughput, good response time and less complex than previous algorithms. The foremost advantage of OTBGA is time limitation and utilize equal period to entire every task. To illustrate during experiment identify that OTBGA is enhanced than Existing algorithm in stipulations of total execution time and cost. as well, OTBGA has superior load balancing performance.

5.1 Conclusion

In observation of the existing load balancing in VM resources scheduling, load balancing in cloud computing environment this research near a scheduling approach on VM load balancing based on PSO, genetic algorithm. allow for the VM resources scheduling in cloud computing environment and with the benefit of genetic algorithm, this process according to historical data and present states computes in advance the pressure it will have on the whole system when the current VM service resources that require deploying are arranged to each physical node, and then it prefer the resolution which will have the least influence on the system subsequent to understanding. In this way, the process achieves the most excellent load balancing and reduces or avoids dynamic migration thus determine the problem of load imbalancing and high migration cost cause by traditional scheduling algorithms. considering that resource nodes in cloud computing environment are indecisive, the perception of multifaceted networks was introduced to converse the kind of resource nodes. completely merge such description to set up a load balancing technique of cloud computing based on multifaceted networks. during simulation results, the partial swam optimization technique based on Genetic algorithm accessible by this work can attain preferable performance.

5.2 Future Work

In the in the interim, this work has compare and analyzed the algorithm in situation below which the population sizes are dissimilar and will persist researching in future works. Resource load balancing is a NP-hard optimization; consequence outputs of this work will pave the method for optimization in erstwhile fields.

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

Reference

[1] Mohamed Riduan Abid, Moulay Idriss El Ouadghiri,Michael Gerndt," Virtual Machines' Load-Balancing in Inter-Clouds " Virtual Machines' Load-Balancing in Inter-Clouds -2016.
[2] J. Li and J. Peng, Task Scheduling algorithm based on improved genetic algorithm in cloud computing environment, Journal of Computer Application, 31(01): 184-186, 2011.
[3] L. Zheng, The Research of resource scheduling key technology in cloud computing, Beijing: Beijing University of Posts and Telecommunication, 2014.
[4] T. Wang, Z. Liu, and Y. Chen, Load Balancing Task Scheduling Based on Genetic Algorithm in Cloud Computing, in Proceedings of 12th International Conference on Dependable, Autonomic and Secure Computing, 2014: 146-152.
[5] Z. Zhu and Z. Du, Improved GA-based task scheduling algorithm in cloud computing, Computing Engineering and Applications, 49(5):77-80, 2013.
[6] Zhao Y, Huang W. Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud[C]//INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on. IEEE, 2009:170-175.
[7] LiuH, LiuS, MengX, et al. Lbvs: A load balancingstrategy for virtual storage[C]//Service Sciences (ICSS), 2010InternationalConference on. IEEE,2010:257-262.
[8] Wang S C, Yan K Q, Liao W P, et al. Towards a load balancing in a three-level cloud computing network[C]//Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conferenceon.IEEE,2010,1:108-113.
[9] Xu G, Pang J, Fu X. A load balancing model based on cloud partitioning for the public cloud[J]. Tsinghua Science and Technology, 2013,18(1):34-39.
[10]DasguptaK,MandalB,DuttaP,etal.AGeneticAlgorithm(GA) based Load Balancing Strategy for Cloud Computing[J]. Procedia Technology,2013,10:340-347.
[11] Pan J S, Wang H, Zhao H, et al. Interaction Artificial Bee Colony Based Load Balance Method in Cloud Computing[M]//Genetic and Evolutionary Computing. Springer International Publishing, 2015: 4957.
[12] Mao Y, Chen X, Li X. Max–min task scheduling algorithm for load balance in cloud computing[C]//Proceedings of International Conference on Computer Science and Information Technology. SpringerIndia,2014:457-465.
[13] D. Chitra Devi and V. Rhymend Uthariaraj," Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreemptive Dependent Tasks" Volume 2016 (2016), Article ID 3896065, 14 pages http://dx.doi.org/10.1155/2016/3896065
[14] Mohamed Riduan Abid, Moulay Idriss El Ouadghiri,Michael Gerndt," Virtual Machines' Load-Balancing in Inter-Clouds " Virtual Machines' Load-Balancing in Inter-Clouds -2016.
[15] Shikha Garg, Rakesh Kumar Dwivedi† and Himanshu Chauhan‡," Efficient Utilization of Virtual Machines in Cloud Computing using Synchronized Throttled Load Balancing" 2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India, 4-5 September 2015.
[16] Wen, Wei-Tao, Chang-Dong Wang, De-Shen Wu, and Ying-Yan Xie, "An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment," In2015 Ninth International Conference on Frontier of Computer Science and Technology, pp. 364-369. IEEE, 2015.
[17] Chun-Cheng Lin, Member, IEEE, Hui-Hsin Chin, Student Member, IEEE, and Der-Jiunn Deng, Member, IEEE," Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System" IEEE SYSTEMS JOURNAL, VOL. 8, NO. 1, MARCH 2014.
[18] Zhang, Qi, Lu Cheng, and Raouf Boutaba, "Cloud computing: state-ofthe-art and research challenges," Journal of internet services and applications 1, no. 1, pp. 7-18, 2010.
[19] Cho, Keng-Mao, Pang-Wei Tsai, Chun-Wei Tsai, and Chu-Sing Yang, "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing,"Neural Computing and Applications 26, no. 6,pp. 1297-1309, 2015.
[20] Rimal, Bhaskar Prasad, Eunmi Choi, and Ian Lumb, "A taxonomy and survey of cloud computing systems,"INC, IMS and IDC ,pp. 4451,2009.
[21] Voorsluys, William, James Broberg, Srikumar Venugopal, and Rajkumar Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," In IEEE International Conference on Cloud Computing, , pp. 254-265. Springer Berlin Heidelberg, 2009.
[22]Hemant S. Mahalle, Parag R. Kaveri and Vinay Chavan, "Load Balancing On Cloud Data Centres" in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, issue 1, January 2013.
[23] Shridhar G. Domanal and G. Ram Mohana Reddy," Load Balancing in Cloud Computing Using Modified Throttled Algorithm" IEEE, International conference. CCEM 2013. In press.
[24] Brototi Mondal, Kousik Dasgupta and Paramartha Dutta, "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach" in Procedia Technology 4 (2012) 783 - 789, ELSEVIER C3IT-2012
[25] Wickremasinghe, Bhathiya CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments, MEDC Project Report, Distributed Computing Project, CSSE Dept., University of Melbourne, pp.433-659, 2009.
[26] Wickremasinghe, Bhathiya, Calheiros, Rodrigo N., Buyya, Rajkumar, CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications, Proceedings of 24th International

ISSN-2394-5125 VOL 11, ISSUE 03, 2024

Conference on Advanced Information Networking and Applications (AINA), IEEE Computer Society, pp. 446-452, 2010.