Performance Analysis of Dynamic Load Balancing Algorithms with different Service Broker Policies using Cloud Analyst

[1]Shivani kumari, [2]Anita Ganpati
[1]M.Tech. Scholar, [2]Professor
Department of Computer Science
Himachal Pradesh University
Shimla, India

Abstract- Cloud computing is on-demand availability of computing resources, users can use these resources anywhere, anytime with the help of internet connectivity on pay per usage basis. Users request for the services on cloud is increasing day by day so, for better performance need to balance all the request and answered those request in minimum time. For that, load balancing is required in cloud computing. Load balancing is the method of assigning an equal amount of work among nodes in cloud to improve both resource utilization and job response time, also avoid a condition where some of the nodes are overloaded while others are doing small amount of work. It also ensures that every node in the distributed network does almost equal amount of work at particular amount of time. Main objective of the study is to analyze the performance of different dynamic load balancing algorithms i.e. Ant colony optimization, honey bee, particle swarm optimization and throttled with respect to different service broker policies. The performance metrics are overall data processing time, response time and total cost. For this, Cloud Analyst tool is used. From the findings of the study, it may be concluded that Ant colony optimization performs better with optimize response time service broker policy.

Index Terms: ***Cloud Computing, Load Balancing, Load Balancing Algorithms, Ant colony optimisation, Honey Bee, Throttled, and Particle Swarm Optimisation, Cloud Analyst.***

I. INTRODUCTION

Cloud computing is a model that provides on-demand availability of computing resources like servers, networks, application and software access these resources at anywhere, anytime through internet connectivity. Users pay according to the pay-as-you-go model [12].

The most widely used definition of the cloud computing model is introduced by National Institute of Standards and Technology (NIST) [10] as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

In distributed environment, client requests are generated randomly. For better performance on cloud, it is important to balance all the request so that no node should be overload or underload for that load balancing is required [3].

II. LOAD BALANCING IN CLOUD COMPUTING

Load balancing is a method in cloud computing to distribute to total amount of work load to all nodes in the distributing computing equally, and avoid a situation that some of the nodes are heavily weighted and some are lightly weighted. Load balancing ensures that every node in the network does approximately equal amount of work so that it minimizes the response time and maximizes the resource utilization. For proper load balancing, load balancer are used. Load balancer is a device that acts as a reverse proxy and distributes the network load across all the servers. Main things in load balancing are load estimation, load comparison, different system stability, system performance, interaction between nodes, nature of the work to be transferred, and many others things to be considered. The main aim of load balancing in cloud computing is to improve the performance of computing in cloud, backup plan in case of system failure, reduces associated cost and response time for working in cloud, also maximizes the availability of resources [6]. Major goals of load balancing are [9]:

- Improve performance
- Cost-effective
- Having backup plan in case of system failure.
- To maintain system stability
- To accommodate future modifications

III. NEED OF LOAD BALANCING

Load balancing in cloud computing is used to distribute the total amount of work equally among all the nodes in the network, so that no node can be overloaded (doing large amount of work) or underloaded (doing small amount of work). Load balancing helps to increase the performance of the system, improving throughput, response time of the system, minimizing the overall cost and maximizes the availability of the resources [2]. It also helps in achieving green cloud computing:

Reduced Energy consumption

The main idea of energy consumption is the reduction in idle power i.e. the power consumption of the idle that leads to power saving. Load balancing helps in avoiding overheating of nodes by balancing the workload across all nodes, which helps to reduce the amount of energy consumed.

Reduces Carbon emission

Through load balancing, energy consumption is reduced. Carbon emission is also reduced, that can helps in achieving the Green computing [7].

IV. LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING

Load balancing algorithms are classified into three types based on who initiated the process

a) Server initiated
b) Receiver initiated
c) Symmetric

Depending on the current state of the system, load balancing algorithms can be divided into 2 categories:

a) Static: Static algorithms does not depend on the current state of the system, but requires some knowledge like memory, processing and power performance. In static algorithms, decision about balancing the node is done at compile time [2].
b) Dynamic: Decision of load balancing is depend on the current state of the system. No knowledge is needed about the system. Dynamic algorithms considers different policies like selection policy, transfer policy, location policy, and information policy to balance the load.

Dynamic load balancing algorithms are

Ant colony optimisation algorithm: It is a nature inspired algorithm based on the behaviour of real ants introduced by Dorigo M. in 1996. It is a meta-heuristic algorithm for optimisation of problems. In general, Ants are blind but has the ability to find food source from their colonies and choose shortest path from their colony to food source. On the way ants moving, they lay down a chemical substance on ground called pheromone, while other ants follow path according to pheromone trail. More pheromone on the path increases the probability of path being followed by ants [8]. When a cloud user sends a request, ant starts its movement in forward direction visiting the node one by one according the probability and checking whether a node is overload or underload. If the ant finds an overloaded node, then it starts backward movement to the previous underloaded node to share data and balance the node [2].

Honey bee foraging algorithm: It is a nature based simulated algorithm, primarily based on the behaviour of honey bees for finding food sources [2]. To check service load, servers are grouped into virtual servers. Each virtual servers having its own server queue. Server processing a request from queue and calculates the profit on basis of CPU utilization, it represents the quality of bees shows in their waggle dance. Through duration of waggle dance, honey bees represented its profit. Each server takes the role of forager or scout. Server serving a request, calculates profit and compares with the colony profit. If the profit was high, the server stays at the current server and if the profit was low the server returns to the forager or scout behaviour. Thus, honey bee balances the load in cloud environment [6].

Throttled load balancing algorithm: It is a dynamic algorithm. In this algorithm, the request of cloud user is firstly send to load balancer. The load balancer contains the index table of all virtual machines. Main responsibility of load balancer is to check the index table of all machines with their states i.e. busy or available states. From starting, VM is at available state. The data center controller receives a new request from cloud user, the main responsibility of load balancer is to check the index table for finding the virtual machine that is appropriate for user request. If relevant virtual machine is found, load balancer returns the id of that machine, otherwise return -1 value [1].

Particle swarm optimisation algorithm: It is a nature inspired algorithm simulated on the behaviour of bird flocks. Particle swarm optimisation **(**PSO) is a meta-heuristics global search based optimisation technique mainly focuses on the minimization of total cost of execution of application on cloud environment. PSO balances the load by distribute task to the available resources [6].

Parameters for performance evaluation in load balancing algorithm:

Throughput: It is used to calculate the total number of jobs can be executed in a given interval of time. It needs to be very high to enhance the performance of the system.
Overhead: this parameter determines the amount of overhead involved while implementing a load balancing algorithm.
Fault-tolerance: It is the property that enables system to do processing even after the failure of any device in the system. The load balancing should be fault tolerance.
Response-Time: It is the amount of time taken to respond by a particular load balancing algorithm.
Performance: It is the amount of useful work done by system. Performance is measured in terms of efficiency, accuracy, speed etc. and reduce the overall cost of the system.
Migration Time: It is the time taken to migrate the jobs from one node to another node. Migration time of nodes will be less for enhancing the performance of system.

V. CLOUD ANALYST

In the study, dynamic load balancing algorithms are implemented using the following tools: windows 8.1 operating system, Eclipse 2018-12, Cloud Analyst and JDK 1.8.
Cloud Analyst: It is a graphical user interface (GUI) based toolkit, developed on the top of Cloud-Sim. It allows the researchers to evaluate the performance of load balancing algorithms under various parameters without having detailed knowledge of programming. Cloud analyst provides result in the form of chart and tables, by setting different parameters like storage, bandwidth, main memory etc. Elements of Cloud Analyst are:
Region: Whole world is divided into main 6 regions. Each region represents the continents of the world i.e. North-America, South-America, Europe, Asia, Africa, and Oceania.
User Base: A User base is a group of tens of thousands of users in a single unit. Main responsibility of user bases is to generate traffic for the simulation.
Data centre controller: Its main responsibility is to route the user requests that are received from user bases, and control the various activities of the data centres by creating and destructing the virtual machine.
Service Broker: It handles the traffic between user bases and datacentres. It uses three types of routing policies that are:
.

a. **Closest datacenter:** data centre having less network latency from the user is selected.
b. **Optimise response time policy:** This policy first select the data centre using closest data centre policy, after that closest datacentre is select according to estimated response time.
c. **Dynamically reconfigurable routing with load:** This policy is an extension of closest data centre policy. Here, the number of virtual machines are increases or decreases that are allocated in the datacentre.

VI. Simulation and Analysis of Result

To analyse the performance of dynamic load balancing algorithms i.e. Ant colony optimisation, Honey bee, particle swarm optimisation and throttled algorithm using CloudAnalyst tool. For the performance evaluation of various dynamic algorithms, create two datasets by setting various parameters such as data center settings. Data centre configuration is for setting the data centers. In the study, four data centres are considered having 100,75,50,25 numbers of virtual machines. User bases recognised as USB1, USB2, USB3, USB4, USB5, and USB6. Each users bases consist of thousands of users and located in different regions.

This study takes two different cases for the performance evaluation of these dynamic load algorithms.

CASE 1: VMs having equal number of processors:

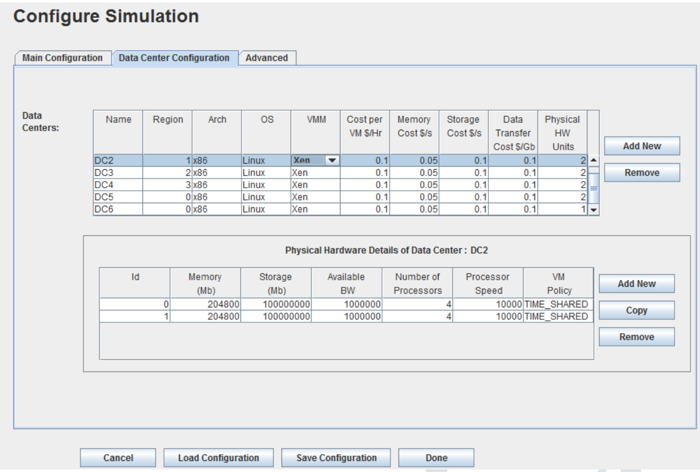

Figure 2: Data Center Configuration

Figure 2 depicts the configuration simulation screen which includes data centers configuration that includes the whole configuration of data centres.

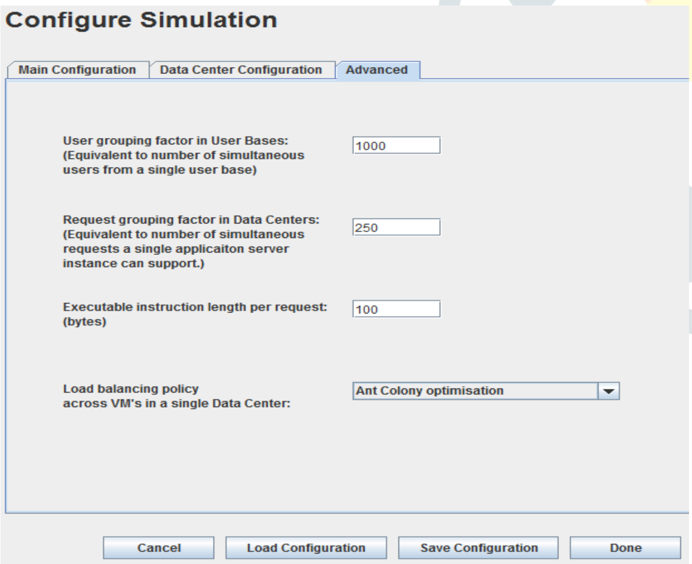

Figure 3: Advanced Configuration

Simulation results of dynamic algorithms for case 1 are shown below tables 1, 2, 3.

Table 1, shows the average response time in ms of load balancing algorithms. Table 2, shows the comparison of dynamic load balancing algorithms according to data processing time in ms. Table 3, shows comparison according to total cost (in $).

Table 1: Comparison of Dynamic Load Balancing Algorithms (Average Response Time (ms))

Service Broker Policy	Ant Colony Optimisation	Honey Bee	Particle Swarm Optimisation	Throttled
Optimise Response Time	126.09	126.11	126.12	126.15
Closest data centre policy	125.76	125.76	125.77	125.80
Dynamically Reconfigure with load	128.89	128.99	128.97	129.04

Table 2: Comparison of Dynamic Load Balancing Algorithms (Average Data Processing Time (ms))

Service Broker Policy	Ant Colony Optimisation	Honey Bee	Particle Swarm Optimisation	Throttled
Optimise Response Time	8.38	8.41	8.41	8.41
Closest data center policy	8.37	8.40	8.40	8.40
Dynamically Reconfigure with load	11.57	11.64	11.63	11.64

Table 3: Comparison of Dynamic Load Balancing Algorithms (Total cost (in $)

Service Broker Policy	Ant Colony Optimisation	Honey Bee	Particle Swarm Optimisation	Throttled
Optimise Response Time	$46.71	$46.71	$46.71	$46.71
Closest data centre	$46.71	$46.71	$46.71	$46.71
Dynamically Reconfigure with load	$69.69	$46.71	$69.69	$69.69

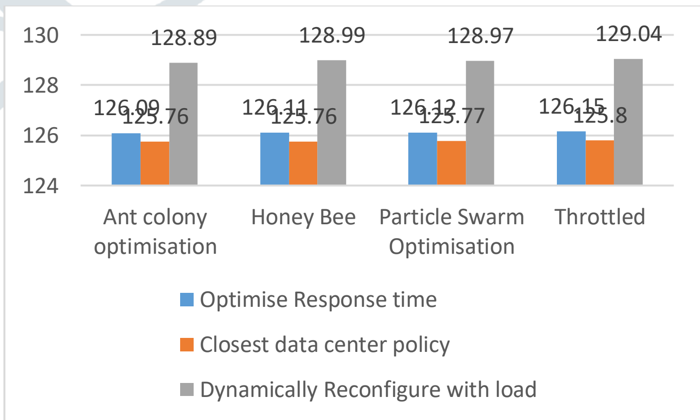

Figure 4: Comparison of Average Response time with three different service broker policies

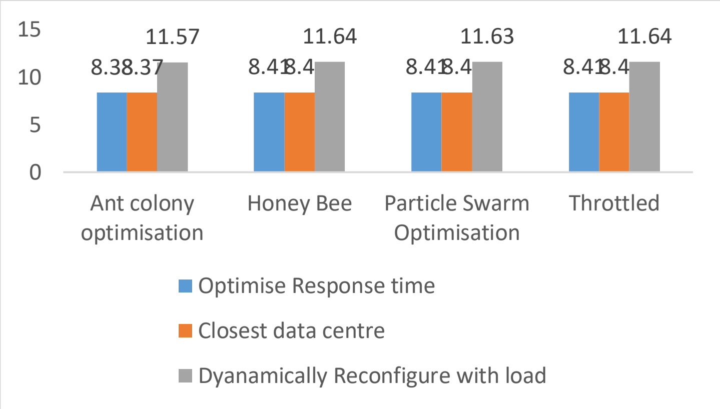

Figure 5: Comparison of Data Center Processing Time with different service policies.

Case 2: VMs having unequal number of processors:

In case 2, change the number of processors in data centre configuration settings number of processors are 2, and 4. And all the parameters settings are remain same as case 1.

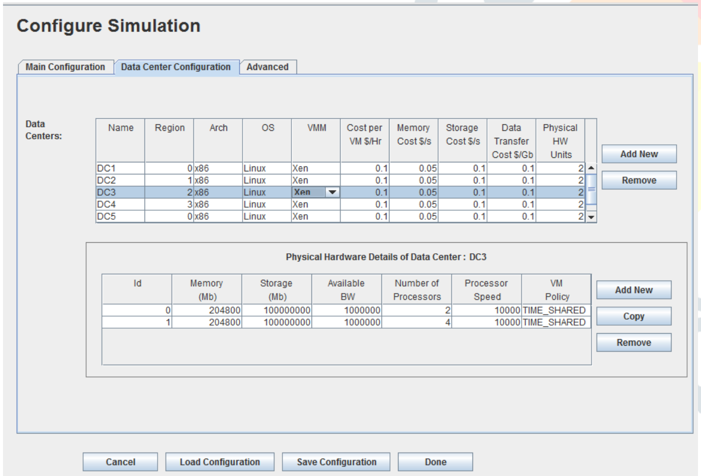

Figure 6: Virtual machines having unequal number of processors.

Table 4: Comparison of dynamic algorithms according to response time (ms) with different service broker policy

Service Broker Policy	Ant colony optimisation	Honey Bee	Particle Swarm optimisation	throttled
Optimise Response time	129.78	126.29	126.28	126.63
Closest data centre	129.62	125.97	125.95	125.97
Dynamically Reconfigure with load	136.17	139.69	139.64	139.60

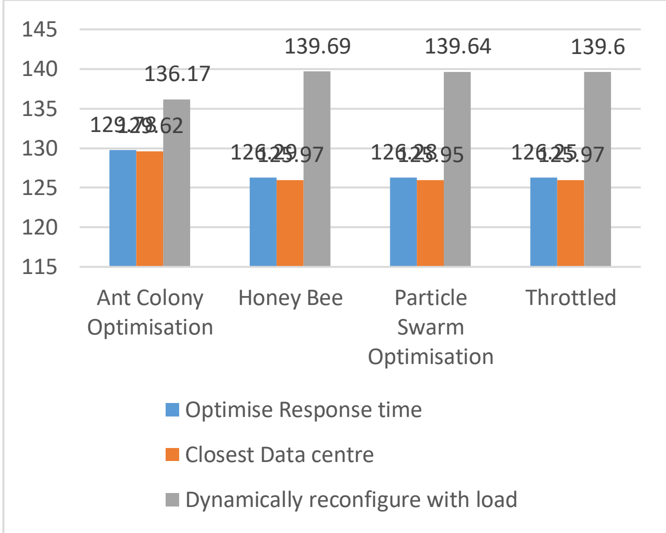

Figure 7: Response time with three different service broker policy.

Table 5: Data Centre processing time

Service broker policy	Ant colony optimisation	Honey Bee	Particle Swarm optimisation	Throttled
Optimise Response time	12.20	8.58	8.58	8.58
Closest Data Centre	12.25	8.57	8.57	8.57
Dynamically Reconfigure with load	18.80	22.37	22.27	22.27

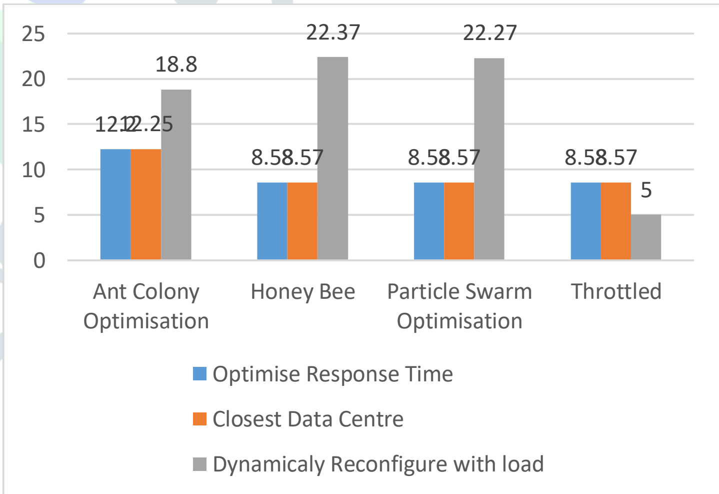

Figure 8: Data Centre Processing Time

Table 6: Cost comparison with respect to different number of virtual machine

Service broker policy	Ant Colony Optimisation	Honey Bee	Particle swarm optimisation	Throttled
Optimise Response Time	46.71	46.71	46.71	46.71
Closest Data Centre	46.71	46.71	46.71	46.71
Dynamically Reconfigure with load	69.72	70.33	69.85	69.85

After performing evaluation, it is analyzed that Ant Colony optimisation load balancing algorithm perform better with optimise response time service broker policy in case 1 i.e. when each data centre is equipped with having equal number of processors. In case 2, virtual machines having unequal number of processors Ant colony optimisation algorithm doesn't fit. Here, Particle swarm optimisation works better. Results of these study shows in Figure 4, 5, 7 and 8.

VII. CONCLUSION AND FUTURE WORK

This work analyze the behaviour of four dynamic load balancing algorithms in cloud environment. All algorithm taken into consideration with different performance metrics (response time, data centre processing time and total cost) according to different service broker policies like Optimise Response time, Closest Data centre and Dynamically Reconfigure with load are verified. According to experimental results, the Ant colony optimisation load balancing in combination with optimise response time perform better in heterogeneous cloud environment. In future, improve the performance of Ant colony optimisation.

REFERENCES

1) Behal, and A. Kumar, "cloud computing: performance analysis of load balancing algorithms in cloud heterogeneous environment," IEEE 2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence), pp. 200-205. Sept 2015.
2) T. Deepa, and Dr. D. Cheelu, "A comparative study of static and dynamic load balancing algorithms in cloud computing", International conference on energy, communication, data analytics, and soft computing, pp.14-9, Nov. 2015.
3) J. Rathore, B. Keswani and V. S Rathore, "Analysis of load balancing algorithms using cloud analyst", Emerging Trends in Expert Applications and Security. Springer, Singapore, 2019. pp. 291-298.
4) P. Geetha and Dr. C.R. Rene Robin, "A comparative study of load balancing algorithms in cloud environments", Data Analytics and Soft Computing (ICECDS). IEEE, pp. 806-810 2017.
5) Ms. Nitika, "Comparative analysis of load balancing algorithms in cloud computing", International journal of Advance research in computer engineering and Technology, no. 3 pp.120-124, May 2012.
6) S. Kumar, D. S. Rana, "various dynamic load balancing algorithms in cloud environment: A survey", International journal of computer applications, vol. 129- No. 6, pp.14-9, Nov. 2015.
7) S. Lamba, and D. Kumar, "A comparative study on load balancing algorithms with different service broker policies in cloud computing", International journal of computer science and information technologies, vol. 5(4), pp. 5671-5677,2014 .
8) M. Dorigo and Thomas Stutzle, "Ant colony optimisation: overview and recent advances, In Handbook of metaheuristics, springer, pp. 311-351.
9) R. Mishra, and A. Jaiswal, "Ant colony optimisation: A solution of load balancing in cloud", International journal of web & semantic technology", Vol. 3, No. 2, pp. 33, April 2012.
10) Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).
11) R. Buyya, R. Ranjan, and R. N. Calheiros, "Modelling And Simulation Of Scalable Cloud Computing Environments And TheCloudSim ToolKit: Challenges And Opportunities," Proc. Of The 7th High Performance Computing and Simulation Conference (HPCS 09), IEEE Computer Society, June 2009.
12) R. Kaur and N. Ghumman, "Hybird improved max min ant algorithm for load balancing in cloud, International conference on communication, computing & systems (Icccs-2014), 2014.