INTERNATIONAL JOURNAL OF RESEARCH CULTURE SOCIETY
Monthly Peer-Reviewed, Refereed, Indexed Journal
Volume - 8, Issue - 1

ISSN(O): 2456-6683
[Impact Factor: 7.148]
January - 2024

IJRCS

DOIs:10.2017/IJRCS/202401006 --:-- Research Paper / Article / Review

Cloud Environment using Meta Heuristic Techniques

Dr. B. Chitradevi
Assistant Professor, Department of Computer Applications, Faculty of Science and Humanities,
SRM Institute of Science and Technology, Tiruchirapalli – 621 105
Email - chitradevi.b@ist.srmtrichy.edu.in

Abstract: *Cloud provides resources to the users based on their requirements by using several resource allocation schemes. Reliable resource allocation is one of the major issues of cloud computing. The objective of this paper is to provide an efficient and reliable resource allocation system for cloud environment. In this regard, this research work proposes Meta heuristic optimization algorithms for resource allocation in cloud. Hence way, resource utilization has achieved high performance and improvement over other traditional techniques. The simulation result shows that the proposed method has reduced the network overhead, average response time, and memory load. As a result, the proposed heuristic scheme is a promising and well-constructed scheme for cloud computing investors.*

Key Words: *Resource Allocation, Cloud Environment, Meta heuristic, Load Balance.*

1. INTRODUCTION :

The recent changes in the computer industry have led to the emergence of new computing paradigms like cloud computing. Due to its ability to host large computer systems, offer a variety of services for users to virtually access a host of resources through the internet, and allow users to only pay for the resources they use, cloud computing systems have become a popular computing paradigm. One of the most crucial components of contemporary data centres is virtualization technology, which is also incredibly beneficial for infrastructure-based services that are hosted on the cloud. As client computing demands have increased, so has power consumption in environmentally friendly data centres. The issue of energy consumption has increased in tandem with the expansion in the quantity and scale of cloud data centres, which pushes up energy costs for service providers and significantly contributes to the emission of dangerous greenhouse gases.

As a result, numerous energy-saving techniques have been suggested in available papers. Recently, researchers have become more interested in this topic in the hopes that the migration of virtual machines (VMs) will lessen the demand for numerous real computers. Moving virtual machines can also have a big influence on energy efficiency [1]. Distributed load balancing on physical computers is essential since upholding the service level agreement (SLA) is a significant difficulty when providing assistance for cloud customers.

In the cloud, a load balancing technique is used to evenly divide work across the available computer resources. It is essential in the cloud environment since it improves resource utilization, quickens answers, and cuts down on task completion times. Studies show that a cloud data centre’s performance may be adversely affected by an unequal resource distribution. Hence, load balancing is essential to ensuring the cloud's continuous success. Yet, in a cloud environment, the data centre must consistently host the cloud service [2]. Because of this, cloud data centres consume a lot of power, raising their operating costs and having an influence on the environment's carbon footprint.

2. Literature Review :

To enhance VM allocation, optimize resource consumption, balance multidimensional resources, and lower communication traffic, Ali et al. [3] proposed an enhanced chaotic binary GWO approach. Sasan et al. [4] suggested utilizing a chaotic hybrid optimization algorithm to discover an ideal assignment issue homogeneously and to simplify the VM with the often heterogeneous servers in cloud data centres in order to foresee and reduce energy usage in cloud computing.

Table 1. Analysis of Existing System

Author	Technique	Benefits	Limitation
Adhikari et al. [5]	LB-RC	Reduce execution costs by using the QoS option.	Task deployment policies' qualities are not taken into account.
Jena et al. [6]	QMPSO	Equalizes the workload by reassigning the load to the suitable VMs while taking into account each VM's fitness score.	It is limited to independent tasks
Golchi et al. [7]	hybrid firefly and IPSO	Task response time has increased.	Energy efficiency not maintained
Haidri et al. [8]	CPDALB	Better load balancing results in a heterogeneous environment.	Various other parameter also need to be focused other than Load balancing
Pourghaffari et al. [9]	EDF-VD	Spit task will improve load balancing outcomes.	Task scheduling can produce far better outcomes.
Kaur et al. [10]	TSFPA	Make span of tasks has been reduced as compared to other techniques	Task load balance must be taken into consideration.

3. Challenges with the load balancing in existing technique:

By offering various services and resources based on user expectations, cloud computing alters human life in various fields and environments. Yet, load balancing is one of the problems that cloud computing is now facing. The following problems with the load balancing strategy are [10]

(1) VM migration: This approach allows for the creation of many virtual machines (VMs) on a single physical machine as part of the virtualization process. These virtual machines (VM) are self-governing in nature and have various configurations. When a physical system becomes overcrowded, these VM are utilized to move the data via a VM migration strategy. This approach occasionally encounters algorithm connection issues [10].

(2) Single point failure: Certain dynamic load balancing methods rely on a centre node approach system, which means that all network or data movement decisions are made in relation to the centre node. A suitable algorithm that manages and distributes the work load equitably and is independent of the centre node in the event of a centre node crash will thus need to be created. If the centre node crashes down, the entire network of systems will also crash.

(3) Algorithm complexity: An algorithm for load balancing must be implemented in a straightforward manner and in accordance with demand. When we utilize complicated algorithms, the system's efficiency and accuracy suffer.

(4) Scalability of the load balancer: Different services are offered to users as needed during the load balancer section's operation. If a suitable load balancer is implemented, the user demand for power storage can be efficiently resolved.

(5) Policy Selection: When tasks or data are chosen to be transferred from one VM to another, the selection policy is employed. According to the amount of overhead necessary for migration from one location to another, this policy determines the data or task.

4. Methodology :

Some load balancing techniques are used in the optimized energy-efficient resource management model for cloud computing architectures to successfully reduce energy consumption while distributing workloads among the servers. The system's main goal is to efficiently perform job scheduling that reduces energy consumption in a cloud environment as well as load balancing by transferring load from jammed servers to under loaded ones. Even though many such systems have been created with this goal in mind, the amount of energy used can be drastically reduced by using a mix of algorithms. The suggested system models and replicates cloud infrastructures using the Cloud Analyst tool. This job adheres to a certain process to balance workload and resource allocation in order to reduce energy usage. The process flow of the newly created energy-efficient load balancing system is shown in below figure,

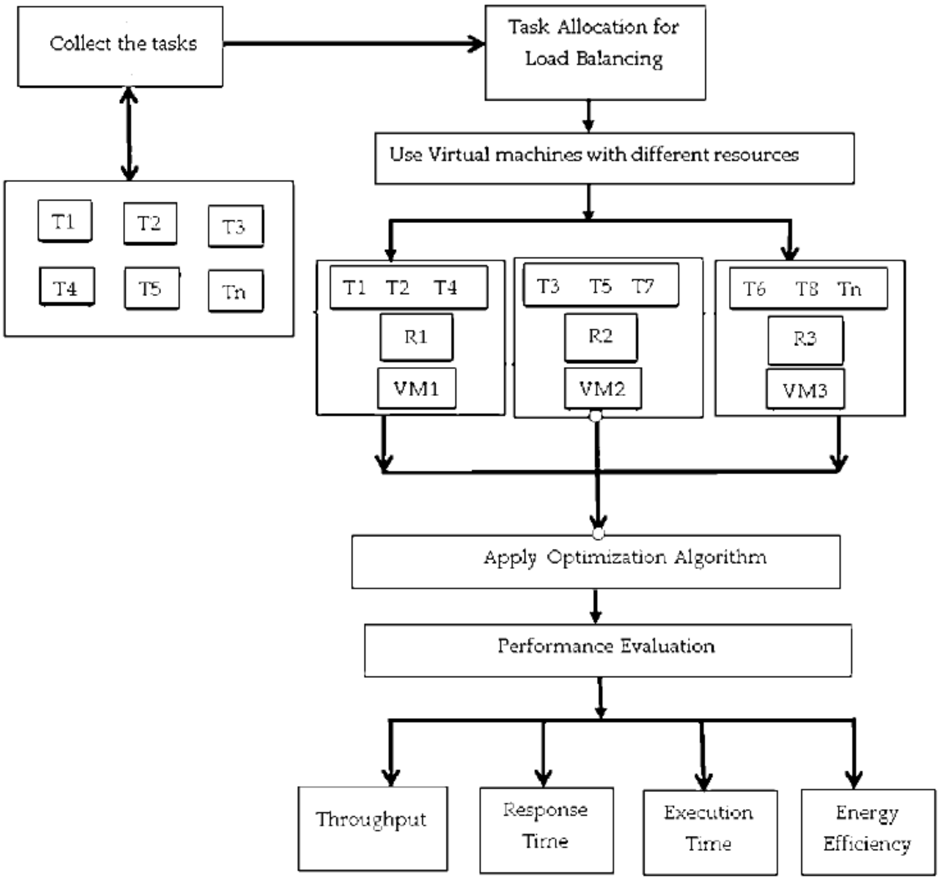

Figure 1. Flowchart of the suggested system

The workflow in above figure begins with the collection of many jobs and their inputs into a task, which is then distributed to various virtual machines based on the load placed on each machine by a scheduler. A proposed method then evaluates the performance of a number of machines in accordance with their functions, redistributing resources to improve system performance. The suggested optimal energy-efficient load balancing solution operates by following particular computational processes. This methodology's steps are as follows:

- To generate the desired number of tasks, enter the necessary number of parameters.
- To distribute the various resources across each virtual machine, host the specified number of servers.
- Launch all of your virtual machines there and allot storage for them using the data center.
- Use optimization techniques to determine the most effective resource scheduling, load balancing between servers, resource migration, and energy consumption calculations.
- Calculating the energy usage, execution time, and pre- and post-migration times is necessary for cloud load balancing.

4.1 Proposed system

Suppose that there are m tasks T={t_1,t_2,,...,t_m}, and n virtual machines VM={(vm0)_1,(vm)_2,...,vm)_n}. It is a NP-complete problem which has n^m ways to allocate these tasks to VMs. When a task is bound to a virtual machine (vm)_i the occupation time of (vm)_j depends on the length of task and the CU of the VM. The execution time for t_ij onvmj is given by t_ij

$$t_{ij}=L_i/C_j$$

where $i\in\{1,2,\ldots,m\}$,
$j\in\{1,2,\ldots,n\}$,
$L_{(i)}$ represents the length of task t_i,
and C_j denotes the CU (Compute Unit) of virtual machine $(vm)_j$.

Proposed Method: The population is evaluated and selected to create a new generation by the fitness function. To find the correctness of the schedule the fitness function is used:

fitness $(P)=\min(U_1,U_2,\ldots,U_n)$

where U_1,U_2,…..,U_n are the chromosomes that represent the time taken to finish all the cloudlets execution for their respective assignment.

Algorithm

- Step 1: Start
- Step 2: Calculate the process priority as per their time
- Step 3: Initialize the population as per the process priority
- Step 4: Evaluate the fitness function to determine the fitness of each individual.
- Step 5: Select the fittest chromosome.
- Step 6: Perform crossover mapping over chromosomes by applying KNN algorithm.
- Step 7: Perform mutation by changing the genes of individual parents.
- Step 8: Add the chromosome to a new population
- Step 9: Repeat steps 2 to 7 for all new arrival process
- Step 10: Exit

5. Result and Discussion :

This section focuses on the computational tests that are performed to determine how well the Tproposed method performs. The Cloudsim tool was used to simulate the proposed approach. The fundamental substrate of this toolkit is Java. All of these tests were performed on a machine equipped with an Intel(R) Core(TM) i5-457 processor, four 2.9 GHz Processors, eight GB of RAM, and a 64-bit Windows operating system. In this section, we examine the simulation outcomes that required the least amount of time and energy. Table 2 shows how the parameters are represented. This experiment employs two distinct datasets to assess the efficacy of the proposed approach. They have 100-500 tasks in each distribution, which is biased to the left and right. Unlike right skewed, which has more little sized activities and fewer large sized obligations, left skewed has less small sized jobs and more large sized tasks.

Simulation: Cloud computing is required to see the application of real-time outlines, which calls for simulation. It can be used with a variety of tools, including Cloudsim, CloudAnalyst, iCanCloud, GroundSim, NetworkSim, SPECI, and Cloud Report. In this study, we make use of Cloudsim and CloudAnalyst.

CloudSim: A program called Cloudsim is used to model, simulate, and run tests on cloud computing infrastructures and services. The Cloud Computing and Distributed Systems Laboratory at the University of Melbourne developed the tool. It is being used in this research to implement various resource scheduling algorithms, including the First Come First Serve, Round Robin, Shortest Resource First, and Primarily Genetic algorithms. It may create several data centres with actual host machines and storage servers inside. These devices house several virtual machines running various cloudlets. The simulation of allocating and running a workload on a cloud infrastructure is possible with CloudSim.

Table 2. List of Parameters

Parameters	**Value**
Number of data center	5
Number of host	10
Host memory capacity	10 GB
Host bandwidth	2800 Mbps
Number of VMs	50
VM policy	Time_shared
VMM	Xen
Number of vCPU	[1-5]
Task MIPS	[200-15000]

6. CONCLUSION :

With the booming requests of cloud services today, there are issues of providing on-demand services and resources in a cloud environment and can’t be overlooked. Our work offers a persuasive of the time-saving method. The online heuristic resource scheduling algorithm called as a FA is conducted as experimental studies. We tried to balance the workload by arranging VM based on their processing power and arranging the cloudlets according to their Length The list of VM and cloudlets is then submitted to the broker for the allocation. Broker allocates through a FA Algorithm and allocation of resources is done. In this paper, we did a comparative analysis of the results of resource scheduling algorithms, Whale optimization Algorithm. The results of using the FA-KNN algorithm are load balancing, improved processor utilization, make span minimization, cost minimization, and maximized throughput. The FA gives better results as compared to the batch heuristic algorithms.

REFERENCES:

1. P. Kumar, A. Tharad, U. Mukhammadjonov and S. Rawat, (2021): Analysis on Resource Allocation for parallel processing and Scheduling in Cloud Computing: in IEEE, 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, pp. 1-6.
2. D. K. Jain, S. K. S. Tyagi, S. Neelakandan, M. Prakash and L. Natrayan, (2022): Metaheuristic Optimization-Based Resource Allocation Technique for Cybertwin-Driven 6G on IoE Environment: in IEEE Transactions on Industrial Informatics, vol. 18, no. 7, pp. 4884-4892.
3. Ali, M.; Masdari, M.; Gharehchopogh, F.S.; Jafarian A, (2021): Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing: Evol. Intell. 14, 1997–2025.
4. Sasan, G.; Masdari, M.; Jafarian, A. (2021): Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm: Clust. Comput. 24, 1293–1315.
5. Adhikari, M.; Amgoth, T. (2018): An Enhanced Dynamic Load Balancing Mechanism for Task Deployment in IaaS Cloud: in Proceedings of the 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 28–29; pp. 451–456.
6. Jena, U.; Das, P.; Kabat, M. (2020): Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment: J. King Saud Univ. Comput. Inf. Sci.
7. Golchi, M.M.; Saraeian, S.; Heydari, M. (2019): A hybrid of firefly and improved particle swarm optimization algorithms for load balancing in cloud environments: Performance evaluation. Comput. Netw. 162, 106860.
8. Haidri, R.A.; Katti, C.P.; Saxena, P.C. (2019): Capacity based deadline aware dynamic load balancing (CPDALB) model in cloud computing environment: Int. J. Comput. Appl. 1–15.
9. Pourghaffari, A.; Barari, M.; Kashi, S.S.(2019): An efficient method for allocating resources in a cloud computing environment with a load balancing approach: Concurr. Comput. Pract. Exp. 31, e5285.
10. Kaur, J.; Sidhu, B.K. A (2017): New Flower Pollination Based Task Scheduling Algorithm in Cloud Environment: in Proceedings of the 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India, 21–23; pp. 457–462.