How to Cite:
Seresht, R. M. (2022). Presenting a multi-objective method to maintain load balancing to reduce energy consumption using tabu search and frog leaping algorithms in cloud computing VMs. *International Journal of Health Sciences*, *6*(S1), 12760–12769.
https://doi.org/10.53730/ijhs.v6nS1.8188

Presenting a multi-objective method to maintain load balancing to reduce energy consumption using tabu search and frog leaping algorithms in cloud computing VMs

Rezvan Mansoori Seresht
Master student of Computer Department, Central Branch, Islamic Azad University, Tehran Branch, Tehran, Iran

***Abstract*---**Cloud computing is considered one of the most important applied networks today. Many users around the world benefit from the services of this network. In this network, users first send their requests to cloud servers, and cloud servers then send the responses to these requests after processing them. In this process, the number of requests is much more than the number of servers; Therefore, it is necessary to perform a scheduling operation to ensure that the execution process is performed in the best possible way to reduce time and energy consumption. This research presents a multi-objective method to maintain load balancing to increase efficiency and reduce energy consumption by using tabu search and frog leaping algorithms in cloud computing VMs. This method consists of two parts: in the first part, the tasks are prioritized by the tabu search algorithm, and in the second part, the best machine is selected to be executed by the frog leaping algorithm. The simulation and comparison results indicate the acceptable performance of the proposed method.

***Keywords*---** Cloud computing, Scheduling, Load balancing, Frog leaping.

Introduction

Cloud computing and its services are widely used by users today. In this network, the more the users, the higher the profitability of network administrators (Belousova NA, et. al., 2020; El Ashiry EA, et. al., 2019). Therefore, network administrators are looking to increase trust and improve cloud network services to gain more users' trust. One way to increase quality is task scheduling to meet users' requests in the shortest possible time and without fault. Due to the very high number of user requests, scheduling should be done so that all users are

provided with access to the cloud computing space. Accordingly, a load balancing algorithm should be used to divide the tasks between data centers in the allotted time to provide services tailored to users' needs at the best possible time to achieve better and fairer results for all users to access the cloud space at the desired time.

Load balancing uses different algorithms to divide the tasks between virtual machines. Load balancing actually divides user requests between virtual machines according to response time [1]. Load balancing is a technique for distributing the load evenly across all network nodes. This technique helps nodes with low or high loads. If a node's load is greater than the threshold value, it is transferred to nodes with a lower load. Therefore, finding an optimal solution for load balancing is a major challenge in cloud computing. In most cases, the idle nature of some servers in the data center and the overloading of other servers to respond to user requests is impossible and unbearable. This means that in the data center, tasks are not properly allocated to the servers, leading to the imposition of idle server maintenance costs on the system. Therefore, appropriate techniques should be used to maintain load balancing to eliminate additional costs in the vast and complex cloud environment. The goals of load balancing [2] are:

- Better performance
- Achieving system steady-state
- Building a system with fault tolerance
- Making more corrections

Managing virtual machines to reduce costs and energy consumption in cloud computing

One of the most important issues in cloud computing systems is the arrangement of virtual machines in existing hosts. The layout problem can be examined from two aspects: 1) deployment of virtual machines in the hosts before processing the system (this can be considered a grouping problem called Bin Packing, and 2) optimization of the layout during system processing to achieve the lowest energy consumption in the data center.

This is about the location of virtual machines, that is, how to allocate virtual machines to hosts to minimize costs. It is called Bin Packing because hosts are like boxes in which goods or virtual machines must be placed so that the maximum capacity of the box, i.e. the host processor, is used and costs are minimized.

From a market perspective, it is important to understand the effects of load balancing in the cloud. Cloud computing has a fully automated service provider platform that allows users to purchase, remotely create, have dynamic scalability, and manage the system. Load balancing in cloud computing systems is currently considered a challenge.

A distributed solution is usually needed because it is not always possible to maintain one or more idle and inactive servers just to meet some of the requirements - or the task is not cost-effective. Obviously, it is impossible to

centrally assign tasks to specific servers due to the scale and complexity of these systems. Load balancing is required to ensure proper resource management of the service provider, which is recommended to the service provider. Load balancing ensures proper utilization of efficient resources for users based on demand and increased performance in the most appropriate time possible.

Remapping is one of the methods used in virtual data center environments. In this method, the number of resource requests of each virtual machine at regular intervals is calculated, and the amount of increase or decrease in its required resource in the future is estimated by a prediction algorithm [3]. Then, using a proposed MFR algorithm, the virtual machines are re-mapped to the physical machines so that the contract breach rate is kept below the maximum acceptable value. In the algorithm presented in [4], a set of network links are presented to divide the brokers into two separate categories. The VMFlow method presented in [5] has succeeded in reducing the cost of network power in data centers. This method initially assumed that only one virtual machine could be deployed on each server. Then, VMs are placed according to the connections between them, so maximum utilization is done from each network link and switch, and the rest of the switches are turned off. This method is then expanded to the point where multiple virtual machines can be deployed on a single server. The disadvantages of this method include placing multiple virtual machines on one server and ignoring the connections between them. In this case, several unrelated virtual machines may be deployed on one server. In this case, communication between brokers will increase, leading to increaed energy consumption, enhanced process transfer. In [6], the task tone mechanism is proposed for the cloud to control the use of load balancing and distributed rate limiting (DRL). Load balancing is used to evenly distribute tasks to different servers. Therefore, the associated costs can be reduced, and the DRL is used to ensure that resources are distributed in a way that the fair allocation of resources is maintained. It is a simple algorithm that is easy to implement due to its very low computational and communication overhead. The environment in which this algorithm is used is a unified framework for cloud control. Among the load balancing criteria, overhead and resource utilization are considered.

The Queue-Idle-Join algorithm proposes a load balancing algorithm for the dynamic scalability of web services. This algorithm provides load balancing on a large scale with distributed distributors. First, the load balancing of idle processors across the distributors is attempted to provide access to each idle processor to each distributor. Then, tasks are assigned to the processors to reduce the average queue length of each processor [7]. Researchers have also proposed a central load balancing policy for VMs, which evenly balances the load in cloud computing or a distributed virtual machine environment. This policy increases the system's overall performance while ignoring fault-tolerant devices. This method utilizes general state information for load balancing decisions and increases efficiency by more than 20%. The environment in which this algorithm is used is cloud computing, in which the criteria of efficiency and response time, operational capacity, and resource utilization are considered [8]. In [9], the existing hosts for each new virtual machine are examined to find an item that has the appropriate free resources to host the virtual machine. As soon as it is found, it matches the virtual machine with it. Existing hosts for the next virtual machine

will be re-examined in order from the previous location, and the first host that can host the virtual machine will be re-selected. This process continues until all VMs are allocated. The purpose of using the Round Robin method is to distribute VMs fairly among existing hosts.

In general, due to the high number of user requests and the use of cloud space, a solution must be found so that users are less in line to wait for their tasks to be answered. The proposed load balancing algorithms try to reduce runtime and meet users' needs by providing a set of tasks. This study presented a multi-objective method to maintain load balancing to reduce energy consumption by using tabu search and frog leaping algorithms in cloud computing VMs.

Simulation scenario

The proposed algorithm is implemented by MATLAB. The details and conditions of implementation are as follows:

- The number of tasks varies from 50 to 1000
- The number of resources required for each task varies from 30 MB to 60 MB
- Time of death of each task: 20 to 100 seconds
- Number of machines: 100 to 400
- Resource size of each machine:50m to 120m
- Tasks are defined as a two-dimensional array, each row of which is related to a task, and its columns to the task specifications, including creat time, required resources, and time of death.
- Machines are implemented in a two-dimensional array, each row of which is for one machine.

The method proposed in the thesis is simulated on the San Diego dataset with the following features:

- Virtual machine
- Physical machine

The validation dataset consists of a dataset for tasks and a dataset for machines. The task dataset has the following structure. This structure is for those tasks that have not yet been done.

Table 1: Task dataset before scheduling and execution

Task_id	**Size**	**Remain**	**Create time**	**Time of Death**	**Start time**	**End time**	**pm_id**

- task_id: task number
- size: the total number of resources required by the task
- remain: The number of resources required by the task until it is finished.
- Time of Death: The length of time a task can wait and then disappear.
- Create time: When the task is created.
- Start Time: When the task starts to run on the machine
- End_TIME: When the task is fully executed
- Pm_id: The number of the machines that executed the task

Table 2 shows the dataset of physical machines that have not yet performed any task.

Table 2: Machine dataset before scheduling

All Vm count	Vm count	Working time	resource remain	size of resource	Pm_id

- Pm_id: Physical machine number
- Size of resource: The total resources in the machine
- resource remain: The number of free resources of the physical machine
- Working time: The total time the machine has been working
- VM count: The number of virtual machines in this machine
- all VM count: all virtual machines that the machine has run

The specifications of the system on which the simulation is performed are as follows:

- CPU: INTEL CORE I5
- 2GB RAM
- VGA 1GB
- OS win 7 32 bit
- 500GB HDD

Comparison methods

Three methods are used for comparison, as described below:

The first method

This heuristic algorithm begins with a set of tasks not mapped to resources. Then, a set of tasks with a minimum completion time is selected. Then, the task with the most completion time is selected to be mapped to resources from this set. The mapped task is then removed from the set. This process continues until the entire set becomes empty. In general, this method aims to minimize the execution time of those tasks taking up more execution time. This algorithm outperforms those algorithms where smaller tasks are executed while larger tasks are left on hold, and some resources are left idle. Therefore, it can be said that this algorithm creates more efficient mapping and more balanced load balancing at the resource level [10]. The above method is used for comparison because it seeks to minimize the execution time of tasks. This reduction in time leads to a reduction in energy consumption and our proposed method also seeks to reduce energy consumption.

The second method

In [11], a dynamic load balancing strategy based on the ant colony algorithm is presented. Here, scheduling speed increase and switching between processors decreases. Also, tasks are selected to run, and the properties of the tasks are known in the advanced mode. Compatibility threshold helps dynamic load balancing of processors. Our proposed method uses a combination of two algorithms, namely the colonial competitive algorithm and the genetic algorithm. For this reason, we have used the genetic algorithm to compare the above method.

The third method

In [12], a fault-tolerant scheduling method of real tasks based on the bee approach is presented. In this study, researchers have proposed a new scheduling algorithm using an optimization method based on a combined bee algorithm with the knowledge of the real-time scheduling tasks scope in multiprocessor environments for fault tolerance (FT). The comprehensive simulation results indicate the higher productivity and efficiency of the new primary-backup based fault-tolerant scheduling (PBFTS) scheme than other fault-tolerant schemes.

Comparison criteria

The following criteria have been used to compare the proposed method with other methods:

- Fault
- Energy consumption
- Load distribution

Findings

Energy consumption

This criterion determines how much energy the machines consume to perform the task. This is simulated under two different scenarios. In the first scenario, the number of machines is constant, and the number of tasks increases step by step. Figure 1 shows the amount of energy consumed in each step. In this scenario, it is assumed that each task unit consumes one unit of energy.

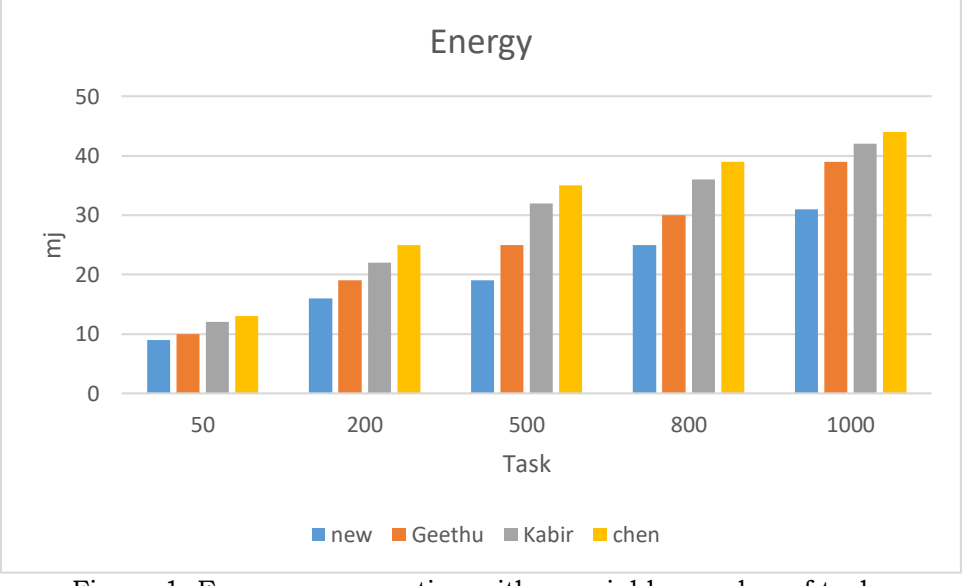

Figure 1: Energy consumption with a variable number of tasks

In this scenario, the proposed method performed better. The graph shows an increase in each step because the number of machines is constant, and the number of tasks has increased in each step, leading to higher energy consumption.

Load distribution

This criterion determines how well the tasks are distributed among the machines. The more uniform the load distribution, the higher the system efficiency. How the load is distributed between the machines is determined using the following equation:
load-balancing = (max time- min time) / (avg time)
To evaluate this criterion, the number of tasks is considered constant in the simulation, and the number of machines is increased in each step. The relevant results are presented in Figure 2.

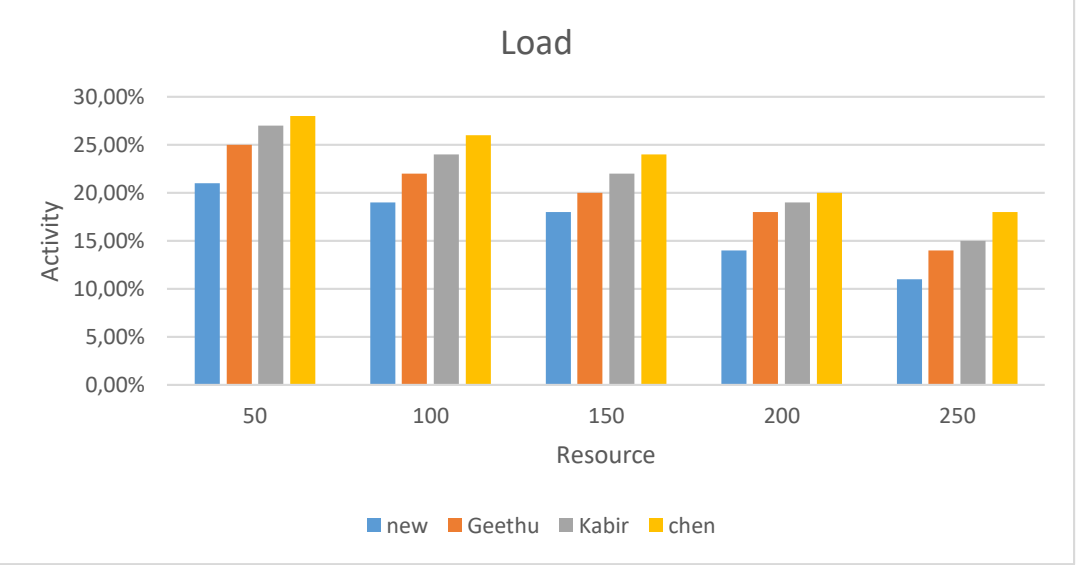

Figure 2: Load distribution

Load distribution is done more equitably in the proposed method, and the energy consumption is also reduced.

Fault

A fault occurs if a task is forced to migrate to another machine for full execution after it is mapped to another machine. This migration leads to increased energy consumption. The simulation and comparison results are shown in Figure 3. The number of resources is constant and the number of tasks is considered variable in the simulation scenario.

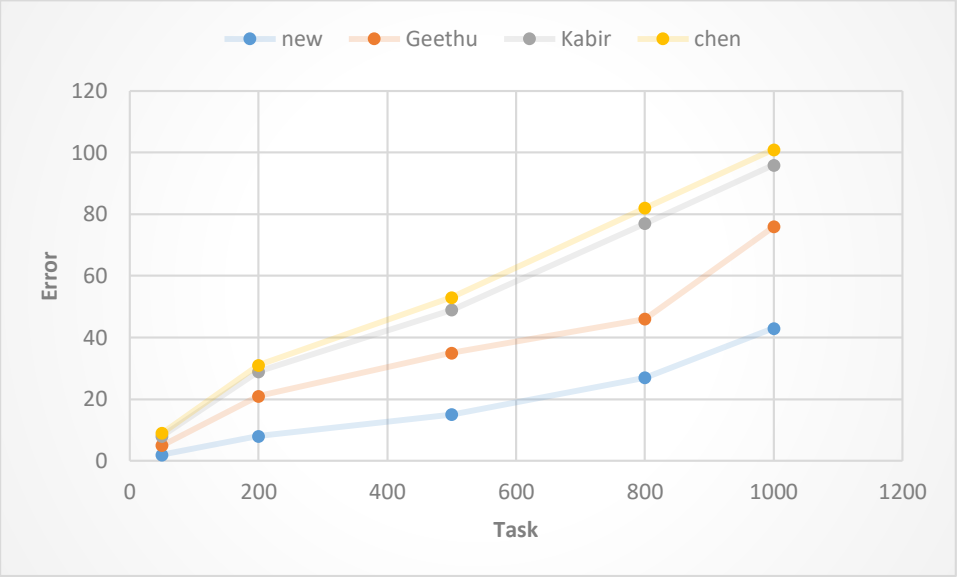

Figure 3: Comparing the number of faults

The simulation results indicate the better performance of the proposed algorithm.

Conclusion

One of the most important issues in cloud computing systems is the arrangement of virtual machines in existing hosts. The layout problem can be examined from two aspects: 1) deployment of virtual machines in the hosts before processing the system (this can be considered as a grouping problem called Bin Packing, and 2) optimization of the layout during system processing to achieve the lowest energy consumption in the data center.

This is about the location of virtual machines, that is, how to allocate virtual machines to hosts to minimize costs. It is called Bin Packing because hosts are like boxes in which goods or virtual machines must be placed so that the maximum capacity of the box, i.e. the host processor, is used and costs are minimized.

- Since one of the most important problems of data centers is high energy consumption and the unbalanced distribution of tasks, reducing energy consumption in these data centers requires more efficient use of available processing resources such as virtual machines and physical hosts. How virtual machines are deployed in hosts in data centers is important and debatable because using fewer hosts is equivalent to less energy consumption. One way to reduce energy consumption in data centers is to reduce the processing load of the hosts because the lower the processing load of the hosts, the lower their energy consumption. However, a host in idle mode will consume only 30% less power than when it uses all of its processing power; that is, it consumes about 70% of its peak power. Combining tabu search and frog leaping algorithms can be proposed to

assign tasks to virtual servers on physical hosts to improve load distribution on machines.

This study analyzed the proposed method, which was a combination of the ant colony and bee colony algorithms. As mentioned earlier, this algorithm was used for task scheduling in cloud computing. The simulation was performed based on several criteria. The simulation results showed the better performance of the proposed method. Two different scenarios were considered for validation: in the first scenario, the number of tasks is variable, and in the second scenario, the number of machines is variable. In each scenario, all important criteria were evaluated. The MATLAB software was used for the simulation. Figure 5 shows the improvement achieved by the proposed method in terms of each criterion.

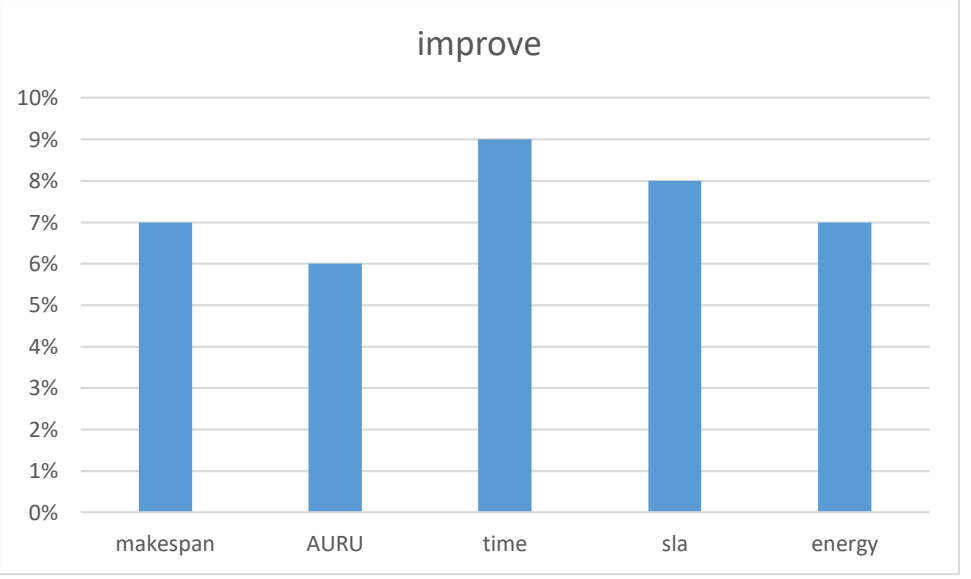

Figure 5: The amount of improvement achieved by the proposed method

Our future study focuses on the development of this type of load balancing and scheduling for the workflows with dependent tasks. This algorithm considers "priority" as the main parameter of service quality. Moreover, in the future, we will also attempt to improve this algorithm by considering other service quality factors.

References

1. G, lovasz, F, Nieedermeier, Hermann DE Meer, "Performance tradeoffs of energy-aware virtual machine Consolidation", 2012, DOI 10.1007/s10586, Page(s)481-496.
2. Beloglazov, Anton, Abawajy, Jemal, Buyy, Rajkumar," Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing", Elsevier Future Generation Computer Systems 28, 2012, Page(s)755-768.
3. Md. Shahjahan Kabir 1, Kh. Mohaimenul Kabir 2 and Dr. Rabiul Islam 3,PROCESS OF LOAD BALANCING IN CLOUD COMPUTING USING GENETIC

ALGORITHM, Electrical & Computer Engineering: An International Journal (ECIJ) Volume 4, Number 2, June 2018

4. Amin Jula, Elankovan Sundararajan, Zalinda Othman, Expert Systems With Applications, Expert Systems With Applications 41 (2014) 3809–3824
5. Blau B., Neumann D., Weinhardt C., Lamparter S., "Palnning and Pricing Service Mashups", In proceeding of 10th IEEE conference on E-Commerce Technology and 5th IEEE conference on Enterprise Computing, pp.19-26, (2008).
6. https://www.linkedin.com/pulse/20140810144258-814120-choosing-the-best-way-to-move-virtual-machines
7. Chapin S.J., Katramatos D., Karpovich J., Grimshaw A.S., "The Legion Resource Management System", In Proc. of the 5th Workshop on Job Scheduling Strategies for Parallel Processing, in conjunction with the International Parallel and Distributed Processing Symposium, pp. 162-178, (1999).
8. Ullman JD., "NP-complete scheduling problems", In Journal of Computer and System sciences, pp. 384-393,(1975).
9. Workflow Management Coalition, Workflow Management Coalition Terminology & Glossary, (1999).
10. Guangshun Yao, "Endocrine-based coevolutionary multi-swarm for multi-objective workflow scheduling in a cloud system", Springer-Verlag Berlin Heidelberg 2018
11. Jung,Gihun, Mong Sim, Kwang, Location-Aware Dynamic Resource Allocation Model for Cloud Computing,Environment,in proceedings of International Conference on Information and Computer Applications,Dubai,2018. page 789-795
12. Xuan chen," Research of Improved Shuffled Frog Leaping Algorithmin Cloud Computing Resources, International Journal of Grid and Distributed ComputingVol. 9, No. 3 (2018)
13. Belousova NA, Korchemkina YV, Matuszak AF, Fortygina SN, Shulgina TA, Kovtun RF, Permyakova NE. Digital environment components for the formation of students' information and analytical skills. Journal of Advanced Pharmacy Education & Research, 10(4), pp. 118-125. (2020)
14. El Ashiry EA, Alamoudi NM, Farsi NM, Al Tuwirqi AA, Attar MH, Alag HK, Basalim AA, Al Ashiry MK. The Use of Micro-Computed Tomography for Evaluation of Internal Adaptation of Dental Restorative Materials in Primary Molars: An In-Vitro Study. International Journal of Pharmaceutical Research & Allied Sciences, 8(1), pp. 129-137, (2019).