Embedding Virtual Machines in Cloud Computing based on Big Bang–Big Crunch Algorithm

Afshin Mahdavi
Afshinmahdavi96@yahoo.com
Department of Computer Engineering, Tabriz branch, Islamic Azad University, Tabriz, Iran
Ali Ghaffari*
Department of Computer Engineering, Tabriz branch, Islamic Azad University, Tabriz, Iran
A.Ghaffari@iaut.ac.ir

Received: 27/Dec/2019 Revised: 29/Apr/2020 Accepted: 09/May/2020

Abstract

Cloud computing is becoming an important and adoptable technology for many of the organization which requires a large amount of physical tools. In this technology, services are provided and presented according to users' requests. Due to the presence of a large number of data centers in cloud computing, power consumption has recently become an important issue. However, data centers hosting Cloud applications consume huge amounts of electrical energy and contributing to high operational costs to the environment. Therefore, we need Green Cloud computing solutions that can not only minimize operational costs but also reduce the environmental impact. Live migration of virtual machines and their scheduling and embedding lead to enhanced efficiency of dynamic resources. The guarantee of service quality and service reliability is an indispensable and irrevocable requirement with respect to service level agreement. Hence, providing a method for reducing costs of power consumption, data transmission, bandwidth and, also, for enhancing quality of service (QoS) in cloud computing is critical. In this paper, a Big Bang–Big Crunch (BB-BC) based algorithm for embedding virtual machines in cloud computing was proposed. We have validated our approach by conducting a performance evaluation study using the CloudSim toolkit. Simulation results indicate that the proposed method not only enhances service quality, thanks to the reduction of agreement violation, but also reduces power consumption.

Keywords: Cloud computing; Virtual machine; Big Bang–Big Crunch algorithm; Energy; Service level agreement**.**

1- Introduction

Cloud can be defined as a new computing technology and paradigm that provides scalable, on-demand, and virtualized resources for users [1]. Cloud computing is a computing model based on computer networks which presents a new pattern for providing, consuming and delivering computational services (such as infrastructure, software and other computational resources) via using network [2, 3]. Information technology resources are such as wireless sensor networks (WSNs) [4-6], mobile ad hoc networks (MANETs) [7-9] and Internet of things (IoT) accessed at users' request time and based on their needs; they are delivered in a flexible and scalable manner through the internet. That is, users only pay the costs of their own consumed electricity and water. In case cloud computing is applied, users will only pay the cost of services they have used [10-12]. A user and a service provider should sign an agreement so that a service can be provided and delivered to the user. In such an agreement, the level and content of the provided services, i.e. data quality, data management, costs, etc., are mentioned and announced by the providers of cloud computing.

Indeed, as mentioned above, cloud computing is a requested computational model which requires a large amount of physical tools. The services are provided as soon as they are requested by the users. Hence, excessive requests of cloud computing increases power consumption in data centers. These centers consume huge amounts of power which consequently leads to releasing large amounts of carbon. Resource optimization, power consumption reduction and virtual machine stabilization are considered as the major challenges in this research domain [13].

Systems can be used simply and straightforwardly through the application of cloud computing. Scalable resource management in cloud computing is made possible via virtual machines [14, 15]. As the number of requests for using virtual machines increases, cloud computing resources will be also increasingly used. As a result, scheduling resources can be considered as an effective method [16, 17]. As the number of requests increases, the number of virtual machines will be insufficient; thus, solutions are needed for enhancing scalability and

* Corresponding Author

reducing overheads in resource scheduling and enhancing service quality [18-20].

Since the application and utilization of cloud computing has increased significantly [21], approaches and solutions need to be proposed for enhancing service quality, service reliability, optimal resource use and saving power consumption. Virtualization carries out live migration of virtual machine in the network by guaranteeing the lowest drop. Durability of the fixed virtual machine can enhance the efficiency of the dynamic resources.

Service quality guarantee and reliability according to service level agreement is considered to be an irrevocable and indispensable requirement for cloud service providers [22].

Power consumption optimization in cloud data centers is composed of two steps: the first step is the effective allocation of virtual machines which is aimed at maximal utilization of available resources and saving energy [23-27]. The second step is the optimization of the allocated resources. Given the above-mentioned discussions, it is imperative that methods be presented for reducing costs of power consumption, data transmission, bandwidth and also for enhancing service quality in cloud computing.

In this paper, by capitalizing on Big Bang–Big Crunch algorithm [28] in the resource allocation stage, we made an effort to enhance QoS (quality of service) and reduce power consumption. In this paper, we consider the power consumption of different network elements that create the network topology in backbone networks as well as the power consumption in physical machines and data centers. Simulation results show that the proposed scheme is an energy efficient embedded scheme for cloud computing environment.

The rest of the paper is organized as follows: in Section 2, previous methods and works regarding power consumption in cloud computing are briefly reviewed. In Section 3, the proposed algorithm is presented. In Section 4, the required simulations for the proposed algorithm were carried out in Cloudsim environment and the obtained results were compared with those of other similar algorithms. In Section 5, the conclusions and findings are summed up and directions for further research on optimizing service quality in cloud computing are recommended.

2- Related Works

One of the critical issues which has attracted most users' attention is the optimal and maximal use of resources. Embedding virtual machine is one of the most fundamental approaches for virtualization in cloud computing [29]. Embedding or allocating resources can be done in different domains such as operating systems, management of data centers and the selection of the best virtual machine. It was aimed at meeting users' needs.

Embedding virtual machines is a mapping process from virtual machine to physical machine. In other words, it is the selection of the best and most ideal physical machine for virtual machine. The number of virtual machines in each class may be different depending on users' demands [30]. In the architecture of embedding a virtual machine, each data center has a series of physical servers. Each physical server is virtualized through several virtual machines [31].

The current problem in embedding virtual machines is concerned with how to embed virtual machines in achieving maximal efficiency. Here, the term efficiency refers to lesser use of load threshold, cost reduction, scalability and power consumption in cloud data center. However, it should be noted that some of the objectives are in contrast with each other. Hence, all of them may not be achieved in an embedding scheme. In case an appropriate method is not used in data centers, undesirable results such as increased power consumption, reduced service quality, reduced customer satisfaction and other problems might be imposed on service providers.

By capitalizing on proper allocation of virtual machines or techniques in data resource migration among virtual machines, power consumption can be optimized in cloud computing. Here, some of the previous works on saving power consumption in cloud data centers and efficient techniques of embedding virtual machines are reviewed in short.

In [32], the authors investigated the issue of allocating virtual machines by focusing on maximal use of multi-dimensional resources and reducing power consumption. The problem of allocating virtual machine was solved by using honeybee algorithm via hierarchical clustering which was intended to reduce power consumption in servers.

In [33], the authors proposed a method for allocating dynamic resources of data centers based on functional demands. It was aimed at optimizing the number of active servers and supported green computation. An effective algorithm, namely variable item size bin packing (VISBP), was obtained which operates well in real environments by adjusting available resources in physical servers. Their proposed approach supports green computing by optimizing the number of servers used. Experimental results show that the VISBP has better performance in hot spots' migration and load balance when compared to the existing algorithm. On the other hand, the assumption in the paper that all physical machines are homogeneous with unit capacity may be the cardinal restriction of its application.

In [34], the authors proposed the idea of the pool of available physical resources which is presented like a backpack; it is solved by genetic algorithm so as to achieve optimal allocation. In this way, a better solution can be achieved by considering several multi-dimensional parameters in the demand for virtual machine. As a result, virtual machine migration is reduced and energy is saved as much as possible [35].

In [22], virtual machine was selected and embedded for reducing power consumption and enhancing service quality in virtualization and virtual machine stabilization. It was intended for achieving optimal resource efficiency. In this work, researchers used a computational evolutionary algorithm, i.e. compatible genetic algorithm, based on virtual machine stabilization method. Virtual machine was embedded by compatible genetic algorithm via various virtual machine selection policies such as minimum migration time, highest solidarity and correlation and random selection. It was implemented by different processors; it was found that compatible genetic method is desirably responsive to large areas. In cases where energy issue is of high significance, this method can guarantee high service level and service quality for large cloud computing systems with the least amount of agreement violation.

In [36], the issue of embedding virtual machine and data was regarded as an NP-Hard problem. The meta-heuristic algorithm of ant-colony optimization was used for solving this problem. In this method, a set of neighboring physical machines was selected for embedding data and virtual machines. Data are distributed in physical storage tools of the selected physical machines. For processing the capacity of each physical machine, a set of virtual machines is embedded in these physical machines so that the data stored in them can be processed. Simulation results indicate that this method selects physical machines close to each other and tasks are carried out in the allocated physical machines. In this way, the total time for doing tasks is reduced. However, in this scheme each virtual machine is mapped onto single physical machine, which shows that the amount of network resource allocated is not less than its demand and hence the proposed embedding algorithm may not give efficient result in terms of power consumption.

In [37], the authors presented a method for allocating virtual machine by using eagle strategy from Hybrid Krill Herd (KH). It was aimed at enhancing the use of services for internet users. Cloud computing services are presented through the communications of cloud data centers and through sharing the resources of virtual machines. Efficiency parameter is reduced in allocating virtual machine and in abnormal working load of cloud centers and in obstructions. Quality of experience (QoE) is reduced in data centers. The policy of optimizing virtual machine allocation eliminates the impact of remoteness from data center and obstruction impact. In this way, uninterrupted access with appropriate throughput is provided for optimizing the quality of experience. Furthermore, an additional optimization flag is added to the service level agreement. When the workload is explosive, the virtual machine is automatically expanded by using Hybrid Krill Herd (KH). If optimization flag is adjusted properly, optimization is done again. Change and reaction protocol and the agreement protocol were recommended for predicting optimal resources in virtual machines for avoiding abnormality and density. Initial parameters such as delay, packet delivery rate and throughput are continuously controlled and observed for activating re-optimization and accessing the quality of experience with minimal load. Experimental results indicate the enhanced capacity of hybrid KH algorithm on particle optimization algorithm, ant colony optimization algorithm and genetic algorithm.

In [38], using imperialistic competitive algorithms and genetic algorithm, researchers tried to allocate virtual machines to physical hosts. Here, the intersection operation of the genetic algorithm was mixed with IC. Simulation results indicate notable improvements with regard to power consumption of the proposed algorithm. This scheme infers that the users need to send the maximum resource requirement of the virtual machines and virtual links, which may lead to the resource over-subscription problem and cost ineffectiveness.

In [39], the authors developed a new and effective evolutionary method for allocating VMs which was capable of maximizing the energy efficiency of a cloud data center via reservation of VMs. A new fitness function was presented based on energy definition which can effectively reduce power consumption and use the resources of data centers based on reserve. According to this evolutionary algorithm, a VM allocation method was put forth which was able to carry out VM to PM mapping with the best energy efficiency. Finally, an efficient simulation engine was developed for maximizing heuristic solutions of optimal VM allocation which can reduce the required time for evaluating VM allocation solutions in each iteration. The experimental results of the simulation in CloudSim and cloud environment indicate that the method proposed in this study can achieve not only an optimal allocation solution for a set of protected VMs but also can obtain further VMs with fewer physical machines for achieving higher energy efficiency.

In [40], the authors presented a novel merge-and-split-based coalitional game-theoretic scheme for VM consolidation in heterogeneous clouds. At first, they partition PMs into different groups based on their workload levels, then they employ a coalitional-game-based VM consolidation algorithm (CGMS) in selecting members from such groups to form effective coalitions, performs VM migrations among the coalition members to maximize the payoff of every coalition, and finally keeps PMs running in a high energy-efficiency state.

In [41], the authors proposed a spatial task scheduling and resource optimization (STSRO) method to minimize the total cost of their provider by cost-effectively scheduling all arriving tasks of heterogeneous applications to meet tasks' delay-bound constraints. STSRO well exploits spatial diversity in distributed green cloud data centers (DGCDCs). In each time slot, the cost minimization problem for DGCDCs is formulated as a constrained optimization one and solved by the proposed simulated annealing-based bat algorithm (SBA). As a result, the acceptance ratio and resource utilization is not optimized.

In [42], the authors proposed an energy aware clustered load balancing system in which, heterogeneous resources are clustered into different groups by using a partitioning-based clustering algorithm. The clustering reduces number of resources needs to be searched and hence minimizes the time required for resource discovery. An energy aware best-fit virtual machine (VM) allocation is used for reducing the power consumption. The process allocations to VMs are done based on best-fit allocation strategy for optimal space utilization.

The above-mentioned brief review of the related works shows that several methods have been developed and proposed for embedding and allocating virtual machines. Nonetheless, controlling and reducing power consumption and enhancing service quality still remain as serious challenges in cloud computing. The method proposed in this paper for optimizing efficiency and reducing power consumption in cloud computing by embedding virtual machine via Big Bang–Big Crunch algorithm are discussed in the following section.

3- The Big Bang–Big Crunch Optimization Algorithm

Big Bang–Big Crunch optimization algorithm is a meta-heuristic population based evolutionary scheme presented by Erol and Eksin [28] . It includes the following stages:

Step 1: (Big-Bang phase): An initial generation of N candidates is generated randomly in the search space, similar the other evolutionary search algorithms.

Step 2: The cost function values of all the candidate solutions are computed.

Step 3: (Big Crunch phase): This phase comes as a convergence operator. Either the best fit individual or the center of mass is chosen as the Centre point. The Centre of mass is calculated as:

$$x_c = \frac{\sum_{i=1}^{N} \frac{x_i}{f^i}}{\sum_{i=1}^{N} \frac{1}{f^i}} \qquad (1)$$

Where in Eq. (1), x_c is the position of the center of mass, x_i is the position of the candidate, f_i is the cost function value of the i*th* candidate, and N is the population size.

Step 4: New candidates are calculated around the new point calculated in Step 3 by adding or subtracting a random number whose value decreases as the iterations elapse, which can be formalized as:

$$x^{new} = x_c + \frac{\gamma\rho(x_{max} - x_{min})}{k} \qquad (2)$$

Where in Eq. (2), γ is a random number, ρ is a parameter limiting search space, x_{min} and x_{max} are the upper and lower limits, and k is the iteration step.

Step 5: Repeat step 2-4 until stopping criteria has not been achieved.

4- The proposed Method

In this paper, the technique of optimally allocating virtual machines to the requests was proposed for enhancing efficiency and service quality and reducing power consumption in cloud computing. Accordingly, Big Bang–Big Crunch meta-heuristic method was proposed for allocating virtual machines to the requests. In the next stage, since the objective of the proposed method was to reduce the utilization of physical machines for reducing power consumption and enhancing system efficiency, system should be planned in such a way that the highest number of physical hosts should be put in the sleep mode. Hence, sleep/awake technique and threshold were used for determining hosts which should be turned off.

4-1- Fitness Function

The value and fitness of a solution should be measured by a fitness function so as to find out how appropriate the response of that solution is. Based on effective parameters about the quality of a solution, this function attributes a value to the solution. In the proposed algorithm, by applying this function on all the solutions, the fitness of each one is measured; the solution with the best value which may be maximal or minimal according to the policy of placing parameters is considered as the most suitable solution. The following parameters were used in the proposed method. Table 1 gives the parameters used in the formulas.

Table 1. Parameters used in the proposed method

Parameter	***Description***
V	The set of virtual machines
P	The set of physical hosts
V_i	One virtual machine in the V set
V_i^{CPU}	The amount of required processor for virtual machine i
V_i^{mem}	The amount of required main memory for virtual machine i

P_j	One physical host in the set P
P_j^{CPU}	The processing capability of physical machine P_i
P_j^{wcpu}	The total workload of the processor of the physical host Pj
V_{jp}	The set of virtual machines allocated to physical hosts

In the physical host, μj is measured according to Eq. (3):

$$\mu_j = P_j^{w_{cpu}} / P_j^{cpu} \quad (3)$$

Where, in Eq. (3) μ_j refers to efficiency rate of the processor of the physical host p_j. Consequently, power consumption will be computed as follow [35].

$$P(u) = k.P_{max} + (1-k).P_{max}.u \quad (4)$$

Where in Eq. (4), P_{max} is the maximum power consumed when the server is fully utilized; k is the fraction of power consumed by the idle server; and u is the CPU (central process unit) utilization. In this way, the fitness function for measuring the value of each response is computed via Eq. (5).

$$fitness = 1/\sum_{j=1}^{N} P(u_j) \quad (5)$$

In the first stage of the proposed method, after a virtual machine is requested by a user, virtual machines are randomly allocated to requests according to Big Bang–Big Crunch algorithm. Given the total allocation methods, *N* allocation methods are defined as the initial population. In the third stage, solutions change their directions. Then, in the fourth stage, the best virtual machine allocation is searched by orbiting around the initial response so that a more optimal allocation is obtained. After a certain number of the iteration of the algorithm, the best virtual machine allocation to the requests is determined.

Excessive use of resources, especially the processor, leads to the overheating that resource and, consequently, heat loss and power loss. Furthermore, resource reception requests which arrive at the resources with overload should stay in the service reception queues; hence, it results in the increased response time, violation of service level agreement and reduced service quality. On the other hand, resources with an amount of load less than normal which are on need to consume energy so that they can keep their physical machines in the awake mode. They use a negligible portion of their capacities and have poor efficiency; this is considered as a waste of resource and power. Hence, it is necessary that a technique be used for determining hosts with low load and hosts with excessive load. For doing so, efficiency threshold is determined for processors of the physical machines.

For finding machines which should migrate, we need to determine machines with overload or machines with less than usual load. The existence of hosts with overloads leads to reduced system efficiency and service quality and, also, increased temperature of the system. As a result, it is imperative that hosts with overload be determined and that working loads be distributed in a balanced way in the entire system. The method developed in [43] was used for determining threshold.

Figure 1 indicates an example for virtual network embedding in physical machines.

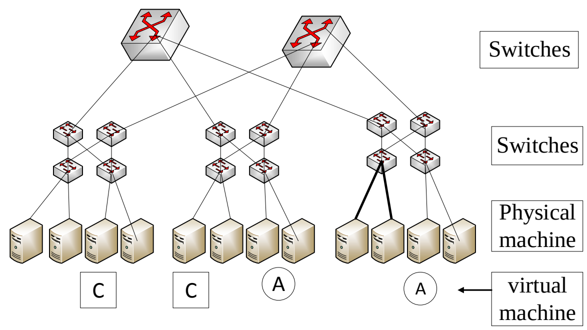

Fig. 1. An example of virtual network embedding: Physical network.

Degree of CPU use of all host machines was considered as a criterion for determining overly used machines and less used machines. If a host machine is specified as an overly used machine, some of the virtual machines of this host machine will migrate to another host machine. Also, in case the host machine is determined as a less used machine, some of the virtual machines of this host machine will migrate to another host machine. Then, according to sleep/awake technique, unused machines will go into sleep/awake modes. According to Beloglazov method, if CPU use is less than low threshold (LT) value, that host machine will be regarded as a less used machine. On the other hand, if CPU use is more than up threshold (UT) value, it will be considered as an overly used machine. In this way, LT and UT are used for specifying less used and overly used machines.

As shown in Fig. 2, by defining threshold boundaries, we will deal with three types of hosts. That is to say, hosts with values less than low threshold are regarded as hosts with low working loads. Hosts with working loads which are higher than up threshold value are known as hosts with overloads. Finally, hosts with working load between UT and LT are considered to be the ideal hosts.

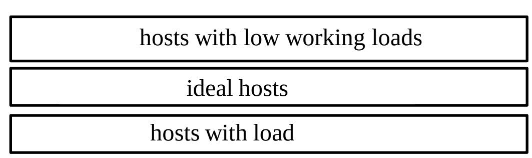

Fig. 2. Classification of hosts in terms of working load

In the proposed method, we tend to reduce the utilization of working machines in line with reducing power consumption and enhancing system efficiency. After low-load hosts are determined, we can migrate virtual machines of these hosts to other hosts according to sleep-awake algorithm and change their mode to the sleep mode. In this way, power consumption is reduced; even by turning off these hosts, their power consumption can be reduced to zero. Hence, it can be argued that turning off an idle host can prevent the loss of power consumption.

Algorithm 1 shows Pseudo code of possible migration.

Algorithm 1. Pseudo code of possible migrate

```
Input: VMList: G^V(N^V, E^V), Physical list: G^P(N^P, E^P)
Output: allocation of VMs, Embedded list
for i=1 to all host (PMs)
  if (is_underload (host_i)==true)
   VM_selction=Random Selection (between all VMs in Host_i);
      VM_selction used as input for allocation agent
    else
    foreach VM in VMList do
    If ((Possible_migrate (all Host_i )==true)
        all Host_i VMs migrate
          end if
            end for
         end if
    end for
  return migration_List
end.
```

As a result of identifying a host with overload, there will be two possible modes. The first mode is that there are ideal hosts and by sending one virtual machine of the host with overload, we can modify this host into an idea host. If one machine of the host is identified as an overly used machine, a number of the virtual machines of this host will be migrated to another host. Random selection (RS) policy was recommended for selecting the virtual machine which should be migrated from the host. According to this policy, the system randomly selects one of the virtual machines for migration; as a result of this migration, the host is modified into an ideal host and it is no longer considered as a host with overload. Consequently, service quality is enhanced because no queue will be established for receiving services; the system will produce less heat and power consumption will be optimized. The second mode is that there are no ideal machines; in this case, the mode of one sleeping machine will be changed into awake mode. Then, the virtual machine of the host with overload will migrate to that host.

The time complexity of the proposed scheme with *m* number of physical machines (physical servers) and *n* number of virtual machines (VMs) is $O(m \log m + n^2)$. According to algorithm 1, the running time of calculation of bandwidth availability of each physical machine is $O(m \log m)$. On the other hand, embedding n virtual machine to m physical machine using algorithm 1 and Big Bang–Big Crunch algorithm is $O(n^2)$. Hence, the time complexity of the proposed scheme can be written as: $O(m \log m + n^2)$

Figure 3 indicates the flowchart of the proposed scheme. The set of *v* nodes are grouped such that those that are not connected to each other in the topology graph are put into one group. Separate groups are created for all the other nodes that are connected to each other to avoid embedding nodes that are connected into the same substrate node.

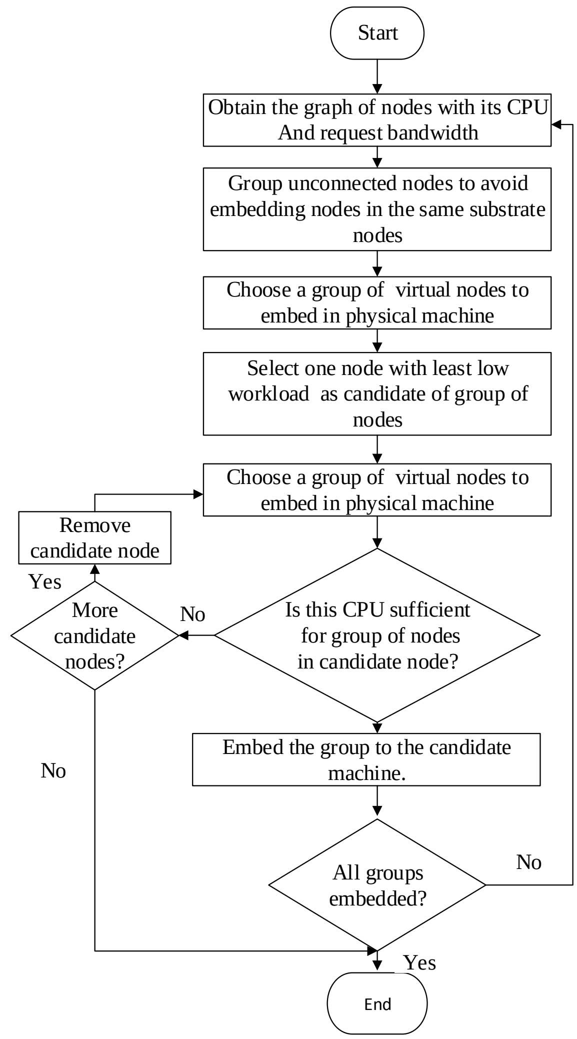

Figure 3. Flow chart of the proposed scheme

5- Performance Evaluation

For evaluating the proposed method, we simulated, investigated and compared it with genetic algorithm and PABFD (Power Aware Best Fit Decreasing) algorithm [43]. CloudSim was used for simulating the proposed method which is a well-known Toolkit in Java language for simulating cloud computing. In the respective scenario, a data center was simulated by 100 heterogeneous physical nodes. Each node was considered with the following specification: a CPU core with identical performance of 1000, 2000 or 3000 millions of commands at each second (MIPS), 8 gigabyte of RAM memory and one terabyte storage space. The degree of power consumption of a physical host ranges from 175 watt with 0% CPU use to 250 watt with 100% CPU use. Each virtual machine needs 250, 500, 750 or 1000 MIPS, 128 megabyte RAM and one gigabyte storage space. The user records requests for supplying 290 homogeneous virtual machines which simulates the capacity of the entire data center. Each virtual machine executes a web application program or any other application with a variable working load. For the sake of usefulness and efficiency, CPU was modeled according to a randomly distributed variable. The application program for 15000 MIPS is equal to a 10-minute execution on 250 CPU units with 100% usefulness. At the beginning, VMs were considered according to the requested specifications and 100% usage. Each experiment was executed for 10 times and the results were obtained according to average values.

One of the significant parameters of the proposed method is the number of iterations of the algorithm which is regarded as the condition for the termination of the algorithm. The number of iterations affects the results and it has been experimentally demonstrated and determined which is shown in Fig. 4. In these experiments, the proposed scenario was executed with differing number of iterations. The vertical axis of Fig. 3 indicates power consumption based on kilowatt per hour and the horizontal axis indicates the number of iterations of the algorithm. It was observed that the best power consumption was related to 100 iterations of the algorithm.

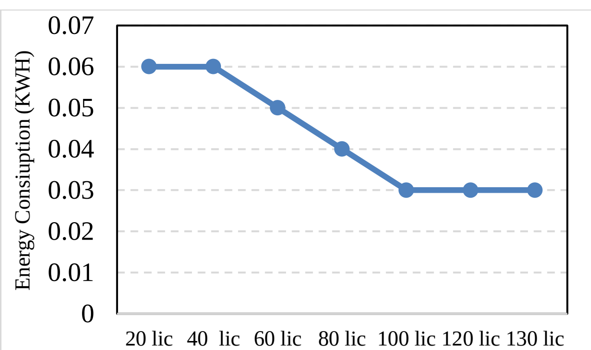

Fig. 4. The impact of the iterations in Big Bang–Big Crunch algorithm

In one experiment, the degree of power consumption was investigated by assuming IQR as host overload detection algorithm and RS (Random Selection) and MMT (Minimum Migration Time) algorithms were considered as VM selection algorithms. The proposed algorithm, genetic algorithm and PABFD algorithm were implemented in 10 random executions. As shown in Fig. 5, with regard to using RS algorithm, the proposed method had the average power consumption of 47 watts per hour which was the lowest power consumption in comparison with GA (Genetic Algorithm) and PABFD algorithms. Furthermore, with respect to using MMT algorithm, the proposed algorithm had the lowest power consumption of 45 watts per hour in comparison with GA and PABFD.

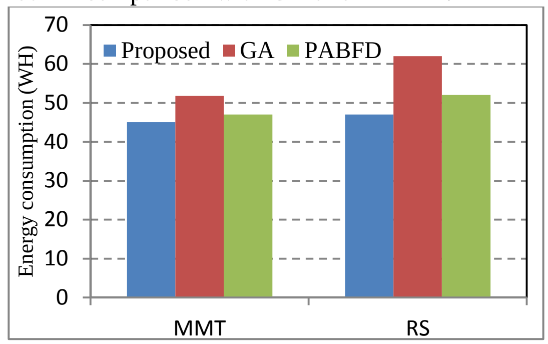

Fig. 5. Average power consumption by considering IQR as host overload detection algorithm in 10 executions

In another experiment, the degree of violated SLA (service level agreement) was investigated by assuming IQR as host overload detection and MMT and RS were used as VM selection algorithms. As shown in Fig. 6, by using MMT algorithm as VM selection algorithm, the degree of violated SLA for the proposed method was 1.321% which was better than those of GA and PABFD.

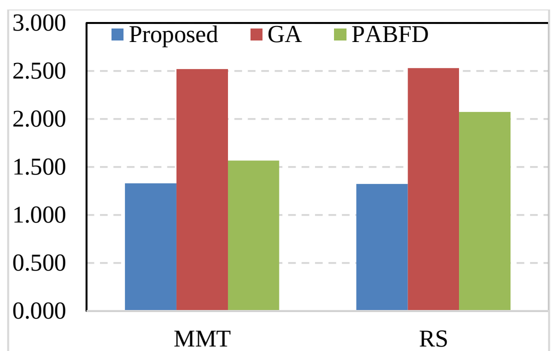

Fig. 6. Average rate of violated SLA by assuming IQR as host overload detection algorithm in 10 executions

In another experiment, power consumption was examined by assuming MAD as host overload detection and RS and MT were used as VM selection algorithms. The proposed method, GA and PABFD were implemented in 10 random executions. The results indicate that the proposed method had the lowest power consumption of 47 watts per hour in comparison with GA and PABFD. Moreover, regarding the use of MMT, the proposed method had the lowest power consumption (45 watts per hour) in comparison with the other two algorithms.

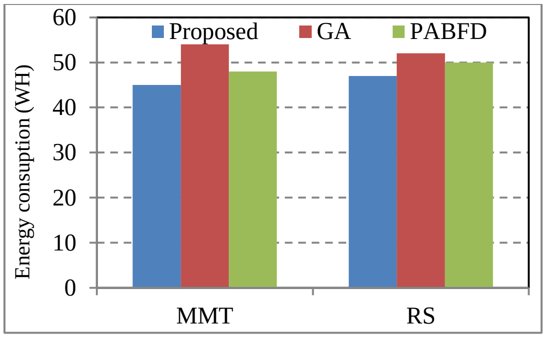

Fig. 7. Average power consumption by assuming MAD as host overload detection algorithm in 10 executions

As depicted in Fig. 7, it is observed that the proposed method had the lowest power consumption in comparison with GA and PABFD. Using RS and MAD, PABFD had better performance than GA. In this figure, the vertical axis denotes power consumption based on Watt per hour.

In another experiment, the degree of violated SLA was examined by considering MAD as host overload detection algorithm and MMT and RS were used as VM selection algorithm. The degree of violated SLA was randomly investigated in 10 executions. The average of ten executions is shown in Fig. 8. Given MMT as VM selection algorithm, the average degree of violated SLA for the proposed method was 1.038%. Also, given RS as VM selection algorithm, the degree of violated SLA was 0.435%. it can be maintained that the proposed method outperformed GA and PABFD with respect to the degree of violated SLA in using RS and MMT algorithms.

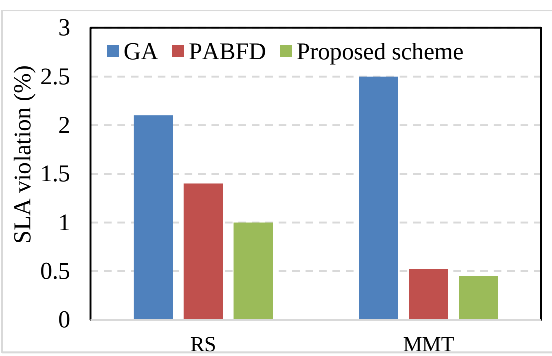

Fig. 8. Degree of violated SLA by using MAD as host overload detection algorithm in 10 executions

In on more experiment, power consumption was investigated by considering LR as host overload detection algorithm and RS and MMT as VM selection algorithms. GA and PABFD as well as the proposed method were implemented in 10 random executions. As depicted in Fig. 9, the proposed method had the lowest average power consumption of 56 watts per hour in comparison with GA and PABFD. Also, regarding the use of MMT algorithm, the proposed method consumed 44 watts per hour which was less than the power consumptions of GA and PABFD.

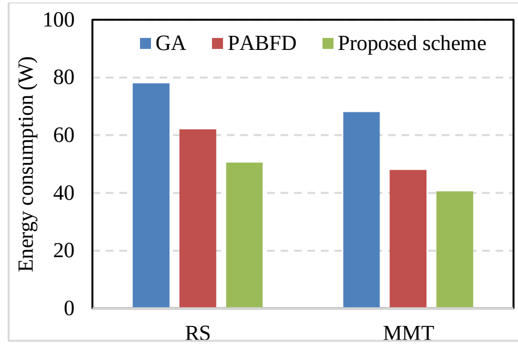

Fig. 9. Average power consumption by considering LR as host overload detection algorithm in 10 executions

In another experiment, LR, RS and MMT were considered as host overload detection algorithm and VM selection algorithms for investigating the degree of violated SLA. As depicted in Fig. 10, the degree of violated SLA for the proposed method regarding the use of MMT and RS algorithms were 1.087% and 0.672%, respectively. Hence, it can be highlighted that, given RS and MMT algorithms, the degree of violated SLA in the proposed method was better than those of GA and PABFD.

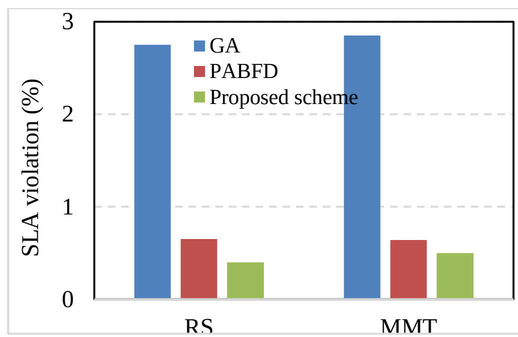

Fig. 10. Average degree of violated SLA by considering LR as host overload detection algorithm in 10 executions

Moreover, LRR, MMT and RS were, respectively, used as host overload detection algorithm and VM selection algorithm for checking power consumption. As Fig. 11 shows, the average power consumption of the proposed method was 46 Watts per hour which was less than those of GA and PABFD. Also, with respect to using MMT algorithm, power consumption of the proposed algorithm was shown to be 42 Watts per hour which was the least amount among the three algorithms.

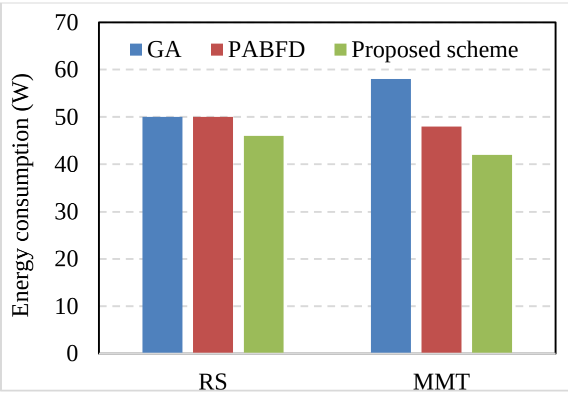

Fig. 11. Average power consumption with respect to LRR as host overload detection algorithm in 10 executions

Moreover, LRR, MMT and RS were respectively used as host detection and VM selection algorithms for examining the degree of violated SLA. As shown in Fig. 12, the degree of violated SLA for the proposed method regarding the use of MMT algorithm as VM selection algorithm was 1.076%. The degree of violated SLA for the proposed method, using RS algorithm as VM selection algorithm, was 0.418% which was better than those of GA and PABFD.

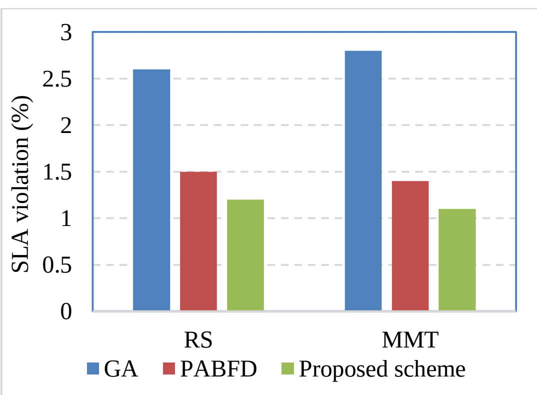

Fig. 12. Average degree of violated SLA with respect to using LRR as host overload detection in 10 executions

In sum, the results of simulations indicate that the proposed method along with LRR and MMT algorithms were desirable for detecting overload of physical host and selecting virtual machine. The average power consumption of the proposed method was 42 Watts per hour. The proposed method along with LRR and MMT was able to optimize power consumption for 11.90%, ESV for 5%, and EST for 49%.

6- Conclusion and Directions for Further Research

Cloud computing centers consume vast amounts of power which, consequently, lead to the release of significant amounts of carbon. Resource optimization, reduction of power consumption and stabilization of virtual machines are considered as critical challenges in this domain of study which has attracted many researchers' attention. Indeed, reduction of power consumption, reduction of SLA agreement violation and service level and optimal use of resources were the primary objectives of the present study. Also, heat production reduction, cost reduction and having more green pastures were regarded as the secondary objectives of the study. By capitalizing on Big Bang–Big Crunch algorithm in allocating virtual machines, the proposed method was able to optimize power consumption and service quality in cloud computing. Furthermore, the proposed method optimized agreement violation and power consumption. Simulation results indicate that the proposed method along with LRR algorithm was suitable for detecting overload of physical host and MMT algorithm was appropriate for selecting virtual machine. The average power consumption of the proposed method was 42 Watts per hour which is regarded as optimal power consumption. Also, the proposed method as well as LRR and MMT algorithms was able to optimize power consumption for 11.90%, ESV for 5% and EST for 49%.

As directions for further research, for covering the difference regarding the degree of violated SLA in service level, cloud service providers can provide more service level than that requested by the users. Hence, it is recommended that a two-objective fitness function be used for reducing power consumption and service level agreement violation.

References

[1] M. H. Ghahramani, M. Zhou, and C. T. Hon, "Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 6-18, 2017.

[2] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Generation Computer Systems*, vol. 79, pp. 849-861, 2018.

[3] M. Noshy, A. Ibrahim, and H. A. Ali, "Optimization of live virtual machine migration in cloud computing: A survey and future directions," *Journal of Network and Computer Applications*, vol. 110, pp. 1-10, 2018.

[4] A. Ghaffari, "Designing a wireless sensor network for ocean status notification system," *Indian Journal of Science and Technology*, vol. 7, no. 6, p. 809, 2014.

[5] A. Ghaffari and A. Rahmani, "Fault tolerant model for data dissemination in wireless sensor networks," in *2008 International Symposium on Information Technology*, 2008, vol. 4: IEEE, pp. 1-8.

[6] D. KeyKhosravi, A. Ghaffari, A. Hosseinalipour, and B. A. Khasragi, "New Clustering Protocol to Decrease Probability Failure Nodes and Increasing the Lifetime in WSNs," *Int. J. Adv. Comp. Techn.*, vol. 2, no. 2, pp. 117-121, 2010.

[7] A. Ghaffari, "Vulnerability and security of mobile ad hoc networks," in *Proceedings of the 6th WSEAS international conference on simulation, modelling and optimization*, 2006:

World Scientific and Engineering Academy and Society (WSEAS), pp. 124-129.

[8] A. Ghaffari, "Real-time routing algorithm for mobile ad hoc networks using reinforcement learning and heuristic algorithms," *Wireless Networks,* vol. 23, no. 3, pp. 703-714, 2017.

[9] R. Mohammadi and A. Ghaffari, "Optimizing reliability through network coding in wireless multimedia sensor networks," *Indian Journal of Science and Technology,* vol. 8, no. 9, p. 834, 2015.

[10] W. Shu, W. Wang, and Y. Wang, "A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing," *EURASIP Journal on Wireless Communications and Networking,* vol. 2014, no. 1, p. 64, 2014.

[11] M. Gahlawat and P. Sharma, "Survey of virtual machine placement in federated clouds," in *2014 IEEE International Advance Computing Conference (IACC)*, 2014: IEEE, pp. 735-738.

[12] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE transactions on parallel and distributed systems,* vol. 24, no. 6, pp. 1107-1117, 2012.

[13] M. Masdari, S. S. Nabavi, and V. Ahmadi, "An overview of virtual machine placement schemes in cloud computing," *Journal of Network and Computer Applications,* vol. 66, pp. 106-127, 2016.

[14] F. López-Pires, B. Barán, L. Benítez, S. Zalimben, and A. Amarilla, "Virtual machine placement for elastic infrastructures in overbooked cloud computing datacenters under uncertainty," *Future Generation Computer Systems,* vol. 79, pp. 830-848, 2018.

[15] S. Sotiriadis, N. Bessis, and R. Buyya, "Self managed virtual machine scheduling in Cloud systems," *Information Sciences,* vol. 433, pp. 381-400, 2018.

[16]A. Kamalinia and A. Ghaffari, "Hybrid task scheduling method for cloud computing by genetic and PSO algorithms," *J. Inf. Syst. Telecommun,* vol. 4, pp. 271-281, 2016.

[17]A. Kamalinia and A. Ghaffari, "Hybrid task scheduling method for cloud computing by genetic and DE algorithms," *Wireless Personal Communications,* vol. 97, no. 4, pp. 6301-6323, 2017.

[18] V. Priya and C. N. K. Babu, "Moving average fuzzy resource scheduling for virtualized cloud data services," *Computer Standards & Interfaces,* vol. 50, pp. 251-257, 2017.

[19] M. Elhoseny, A. Abdelaziz, A. S. Salama, A. M. Riad, K. Muhammad, and A. K. Sangaiah, "A hybrid model of internet of things and cloud computing to manage big data in health services applications," *Future generation computer systems,* vol. 86, pp. 1383-1394, 2018.

[20] A. Satpathy, S. K. Addya, A. K. Turuk, B. Majhi, and G. Sahoo, "Crow search based virtual machine placement strategy in cloud data centers with live migration," *Computers & Electrical Engineering,* vol. 69, pp. 334-350, 2018.

[21]K. R. Remesh Babu and P. Samuel, "Service-level agreement–aware scheduling and load balancing of tasks in cloud," *Software: Practice and Experience,* vol. 49, no. 6, pp. 995-1012, 2019.

[22] P. R. Theja and S. K. Babu, "Evolutionary computing based on QoS oriented energy efficient VM consolidation scheme for large scale cloud data centers," *Cybernetics and Information Technologies,* vol. 16, no. 2, pp. 97-112, 2016.

[23] M. Abdel-Basset, L. Abdle-Fatah, and A. K. Sangaiah, "An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment," *Cluster Computing,* pp. 1-16, 2018.

[24] X. Fu, J. Chen, S. Deng, J. Wang, and L. Zhang, "Layered virtual machine migration algorithm for network resource balancing in cloud computing," *Frontiers of Computer Science,* vol. 12, no. 1, pp. 75-85, 2018.

[25] Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, "Green and sustainable cloud of things: Enabling collaborative edge computing," *IEEE Communications Magazine,* vol. 57, no. 1, pp. 72-78, 2018.

[26] H. Wang and H. Tianfield, "Energy-aware dynamic virtual machine consolidation for cloud datacenters," *IEEE Access,* vol. 6, pp. 15259-15273, 2018.

[27] M. S. Mekala and P. Viswanathan, "Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for IoT," *Computers & Electrical Engineering,* vol. 73, pp. 227-244, 2019.

[28] O. K. Erol and I. Eksin, "A new optimization method: big bang–big crunch," *Advances in Engineering Software,* vol. 37, no. 2, pp. 106-111, 2006.

[29] P. Zhang and M. Zhou, "Dynamic cloud task scheduling based on a two-stage strategy," *IEEE Transactions on Automation Science and Engineering,* vol. 15, no. 2, pp. 772-783, 2017.

[30] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *2009 IEEE Asia-Pacific Services Computing Conference (APSCC)*, 2009: IEEE, pp. 103-110.

[31] J. Gao and G. Tang, "Virtual Machine Placement Strategy Research," in *2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2013: IEEE, pp. 294-297.

[32] M. Hemalatha, "Cluster based BEE algorithm for virtual machine placement in cloud data center," *Journal of Theoretical & Applied Information Technology,* vol. 57, no. 3, 2013.

[33] W. Song, Z. Xiao, Q. Chen, and H. Luo, "Adaptive resource provisioning for the cloud using online bin packing," *IEEE Transactions on Computers,* vol. 63, no. 11, pp. 2647-2660, 2013.

[34] N. Janani, R. S. Jegan, and P. Prakash, "Optimization of virtual machine placement in cloud environment using genetic algorithm," *Research Journal of Applied Sciences, Engineering and Technology,* vol. 10, no. 3, pp. 274-287, 2015.

[35] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems,* vol. 28, no. 5, pp. 755-768, 2012.

[36] T. Shabeera, S. M. Kumar, S. M. Salam, and K. M. Krishnan, "Optimizing VM allocation and data placement for data-intensive applications in cloud using ACO metaheuristic algorithm," *Engineering Science and Technology, an International Journal,* vol. 20, no. 2, pp. 616-628, 2017.

[37] D. Kesavaraja and A. Shenbagavalli, "QoE enhancement in cloud virtual machine allocation using Eagle strategy of hybrid krill herd optimization," *Journal of Parallel and Distributed Computing,* vol. 118, pp. 267-279, 2018.
[38] F. Farhadian, M. M. R. Kashani, J. Rezazadeh, R. Farahbakhsh, and K. Sandrasegaran, "An efficient IoT cloud energy consumption based on genetic algorithm," *Digital Communications and Networks,* 2019.
[39] X. Zhang *et al.*, "Energy-aware virtual machine allocation for cloud with resource reservation," *Journal of Systems and Software,* vol. 147, pp. 147-161, 2019.
[40] X. Xiao, W. Zheng, Y. Xia, X. Sun, Q. Peng, and Y. Guo, "A workload-aware VM consolidation method based on coalitional game for energy-saving in cloud," *IEEE Access,* vol. 7, pp. 80421-80430, 2019.
[41] H. Yuan, J. Bi, and M. Zhou, "Spatial Task Scheduling for Cost Minimization in Distributed Green Cloud Data Centers," *IEEE Transactions on Automation Science and Engineering,* vol. 16, no. 2, pp. 729-740, 2018.
[42] K. R. Babu and P. Samuel, "Energy aware clustered load balancing in cloud computing environment," *International Journal of Networking and Virtual Organisations,* vol. 19, no. 2-4, pp. 305-320, 2018.
[43] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience,* vol. 24, no. 13, pp. 1397-1420, 2012.

Afshin Mahdavi received the B.S. degree in Computer Engineering from Azad University, Ardebil Branch, Iran in 2014, and M.Sc. degree in Computer engineering from Azad University, Tabriz Branch, Iran, in 2019. His research interests include Cloud computing and Computer networks.

Ali Ghaffari received his BSc, MSc. and Ph.D. degrees in computer engineering from the University of Tehran and IAU (Islamic Azad University), TEHRAN, IRAN in 1994, 2002 and 2011 respectively. As an associate professor of computer engineering at Islamic Azad University, Tabriz branch, IRAN, his research interests are mainly in the field of software defined network(SDN), Wireless Sensor Networks (WSNs), Mobile Ad Hoc Networks (MANETs),Vehicular Ad Hoc Networks(VANETs), networks security and Quality of Service (QoS). He has published more than 60 international conference and reviewed journal papers.