



A multi objective virtual machine placement method for reduce operational costs in cloud computing by genetic

Reza Sookhtsaraei¹, Mirmorsal Madani², Atena Kavian³

¹ Faculty member of Payam Noor University, Department of computer engineering and information technology, Tehran, Iran

² Faculty member of Kordkuy Islamic Azad University, Department of Computer Engineering, Kordkuy, Iran

³ Amirkabir University, Department of Computer Engineering, Tehran, Iran

E-mail: ¹reza.sookhtsaraei@gmail.com, ²mt_madani@yahoo.com, ³atena_kavian2008@yahoo.com

ABSTRACT

Increase of demand in using cloud computing caused increase of operational costs consumption energy and resources consumed. As a result moreover satisfying services with quality requested through service level agreement SLA, we must reduce consuming energy and cost resulting from resources used. According to too much researches conducted on reducing consumption energy, in this article we will focus on reducing operational costs resulting from wasting resources using technique of mapping virtual machines to service providers whose the most important one is using on demand provisioning model which will prevent from wasting resources and more exploitation of cloud computing and reduction in operational costs. One of the key aspects of the process is considering load feature of virtual machines. But most tasks done in this field do not care about it and on the other hand only optimization is noticed. Due to existence of faults and also high level of exploitation of genetic algorithms in finding fine results in searching space, in this article a method was presented based on multi-purpose genetic algorithm, where by considering task feature of virtual machines. Several objectives are considered for optimizing during mapping process. Comparing results gained from algorithm suggested with a random algorithm and one algorithm in the first multi-purpose choice, we can come to this conclusion that the algorithm suggested will establish better results on providence in resources and reducing operational costs.

Keywords: *Cloud computing, virtual machine, multi-purpose genetic algorithm, optimization.*

1 INTRODUCTION

In recent years cloud computing has been known as the most popular calculation bed for hosting and providing services based on internet [1], so that most organizations chose this technology as their calculation patterns in solving problems related to IT[2]. To maintain acceptance established in applying cloud computing, we have to take up some measures so that two sides (providers and clients) involved in this technology can reach the maximum level of satisfaction in using it. Based on this, to reach client satisfaction and for meeting service quality requested by them, big cloud computing providers such as Amazon, Google, Microsoft, IBM

and so on created new data centers in different geographical areas so that by establishing redundancy and trust capability they can remove errors made during presenting services and this way they can improve quality level and reliability in these services. On the other hand, as demand in using cloud computing is increased, gaining providers' satisfaction is not an easy task due to increase of operational costs (consuming energy, resources consumed). In [3] it is shown that with increase of demand energy consumption in data center of cloud computing had a 400-percent increase compared to the previous decade. As a result, to control operational costs, providers of cloud computing have to reduce consuming energy

and costs resulting from resources used, moreover meeting services with the quality requested through SLA. Since too much research investigated energy consumption reduction in cloud computing [4-10], we focus on reducing operational costs resulting from wasting resources.

One of the most important technologies along with cloud computing is virtualization. In this technology we can share resources of a machine between several applications by using performance isolated platforms called virtual machines. One of the most prominent advantages of virtualization is applying on demand provisioning model. The model guarantees that users must be provided the resources they need whose results will be preventing from waste of resources and more exploitation of cloud computing and also reduction in operational costs. Despite virtualization, one of the subjects raised is process of mapping virtual machines to service providers. The process is highly important since it has a great effect on exploitation of resources. If the process of mapping virtual machines to service providers is not conducted accurately and with full awareness with resource condition, then operational costs will increase due to waste of resources. As a result, the process was raised as one of the most important research fields on cloud computing [4,9-11,13-22] and to solve it various methods are used such as linear programming [13,14,5], constraint programming [16,17], bin packing [18,19,20], Ant colony algorithm [4] and genetic algorithm. One of the key aspects in mapping virtual machines to service providers which can cause increase of efficient in cloud computing is considering feature of working load of virtual machines during the process [12], but most of tasks done on this do not care about it. Also in most researches conducted on this, only optimization is cared, while in real problems we can consider various objectives. Due to defects mentioned earlier, a method was presented in this article which cares about several goals for optimization by considering feature of working load of virtual machines. Since multi-purpose optimization problem is among NP-hard problems, due to high level of genetic algorithms in finding desirable results in searching space we used this type of algorithm for finding the best response in this article. By comparing results gained from simulating suggested multi-purpose genetic algorithm with a random algorithm and one multi-purpose algorithm, we can come to this conclusion that suggested algorithm can establish better results on resource providence and in reducing operational costs. Next, the article is organized as follows: in section 2 several recent activities conducted on

mapping virtual machines to service providers were presented and their faults are mentioned. In section 3, process of mapping and formulas used is presented officially. In section 4, suggested multi-purpose genetic algorithm is shown. In section 5, simulation results are presented and in the end the article is finished by providing conclusion and expressing future works.

2 WORK RELATED

In this section several researches conducted on replacing virtual machines to physical ones are presented. We investigate works done in this section from three views:

a) Works in which overloading of physical machines are not considered:

In [4], an Ant colony algorithm is presented which deals with mapping problem in permutation form from virtual machines to physical ones. The aim of the algorithm is to present a solution so that it can reduce level of consumed energy and resources wasted simultaneously. In this algorithm, virtual machines are mapped based on level of desirability and probability of movement to physical machines which are selected randomly. In [23], an algorithm of dynamic resource allocation was presented which was based on threshold where it allocates virtual machines based on their workload changes to physical machines. To do this it applies a method based on threshold for optimizing the process. The algorithm reestablishes virtual machines in dynamic form based on needed workload changes in cloud applications and based on this it creates providence in using resources and can increase efficiency.

b) Works in which providence in wasted resources is not considered:

In [24] a developed algorithm First Fit Decreasing (FFD) was suggested in which each physical machine was provided one grade as an advantage and based on this virtual machines are mapped to physical machines with higher level of score. To prevent from deficient immigration, two values of threshold are determined as R_{low} and R_{high} . If a physical machine consumes more than R_{high} , it is called Highly-loaded and if it consumes less than R_{low} , it is called Lowly-loaded. Then virtual machines which are only placed in these two types of machines will have the permission to migrate.

In [12] a modified bin packing algorithm is suggested in which virtual machines can be divided into two sets of CPU intensive and Data intensive. The algorithm maps virtual machines to physical machines asked on two rules so that the least number of physical machines can be determined.

The first rule is in this way that the maximum number of virtual machine which is focused on data mapped on a physical machine should not be more than k , since too much mapping this type of virtual machine to a physical one can increase competition for availability to disk. The second rule is that the maximum resource demanded by virtual machines should not be more than the maximum number of resources present in physical machines.

C) Activities in which physical machine being overhead and providence in wasted resources are not considered:

In [25] algorithms of Hybrid and Dynamic Round Robin (DRR) are presented whose aim is providence in consuming energy using mixture. DRR has been an improvement in algorithm RR in which two rules are used for combining virtual machines. The first rule is so that if a virtual machine ended and other virtual machines are being performed on that physical machine then the physical machine does not establish a new virtual machine. The second rule indicates that if one physical machine does not accept a new virtual machine for a long time, then all the virtual machines will move into another physical machine and the machines will turn off.

In [26] Green computing algorithm is presented in which condition of all the virtual machines are supervised and added physical machines will turn off, if the system efficiency does not become low.

In [27] an architecture called Green Cloud is shown which can reduce consuming energy of data centers. This architecture is capable of supervising cloud elements in on-line form and comprehensively, it can make virtual machines move in live form and also it can optimize mapping of virtual machines.

In [23] a Framework called VM planner was suggested for reducing energy costs resulting from the network elements present in data center (links and routing). Using dynamic migration of virtual machines and routing based on programmable flow-based routing present in modern data centers, this framework can reduce the consumption resulting from network elements moreover meeting traffic needs. To reduce consuming energy, VM planner considers topological features and traffic patterns and based on this it tries to turn off unnecessary network elements as much as possible.

In [10] an architecture called Energy and Carbon-Efficient (ECE) was presented in which an interface called ECE Cloud Broker can map virtual machines to physical machines so that consuming energy of data center can be optimized and as a result Carbon Footprint is reduced. Broker does the mapping process based on data such as consumed energy and

rate of carbon production by energy resources in a data center. The activity conducted in this article is different with previous ones. In mapping virtual machines to service providers we suggest a multi-purpose genetic algorithm using it we can reduce number of overloaded service providers and also the level of wasted resources.

3 PROBLEM EXPRESSING

In a cloud, applications are performed in storage of service providers which were virtualized totally. The task of mapping virtual machines to storage of service providers can be considered a type of Bin packing problem in which virtual machines are in fact items and service providers are Bins. Feature of each item and bin can be described with a leash set in form of CPU, Memory and Bandwidth all of which indicate resources requested and leash of each bin indicates level of resources present. Next we will provide formulas used in mapping process and the reason of using it, also symbols used are shown in table 1.

Table 1. parameters used

Variable	Definition
S	Set all servers
$j \in S$	All of virtual machines that are assigned to the j -server
TR_j^p	The total processing capacity of j -server
TR_j^m	The total main memory capacity of j -server
UR_j^p	Used j -server processing capacity
UR_j^m	Used j -server main memory capacity
RR_j^p	The remaining processing capacity of j -server
RR_j^m	The remaining main memory capacity of j -server
TH^p	The threshold for all processors
TH^m	The threshold for the main memory of all the servers
D_{kl}^B	Required bandwidth between k and l virtual machine
UD_{ij}^B	Bandwidth allocated between i and j virtual machine
B_{ij}	Available bandwidth between i and j servers
D_k^p	Processing capacity required by the k -virtual machine
D_k^m	A memory request by k -virtual machine
TH^B	Bandwidth threshold

a) Objective functions

Assume that we placed n virtual machines in three names of Cpu_VMset, Memory_VMset and I/O_VMset which indicate set of virtual machines focused on processor, focused on memory and input and output, respectively. These sets are dependent on each other, in other words a virtual machine may include features of all three sets. n of virtual machine must be mapped on m service. For simplicity we assume that resources requested of a virtual machine are not more than resources which a service provider provides. The aim of mapping virtual machines to service providers in this article is simultaneous optimization of three following formulas (1, 2, 3)

$$Min \sum_{j=1}^{|S|} |RR_j^p - RR_j^m| \tag{1}$$

Using formula (1), we can gain level of wasted resource. The more minimum the formula (1), the less the value. e.g. if in a service provider level of using the main memory is more than processor, then mapping of the other virtual machine to this service provider will face same problems due to memory shortage and if the problem still exists the remaining processing resource in this service provider may not be usable and it will be considered as wasted resource.

Using formula (2) we can identify service providers overloaded and calculate level of overloading. For calculating value of this formula, m service provider is divided into five groups. The way of calculating level of overload for each group is different to other group. This grouping is conducted based on phrases of (4) to (13). Feature of each group is up next.

$$Min \sum_{j=1}^{|S|} O_j \cdot O_j = \begin{cases} \frac{\sum_{k=1}^{|M(j)|} D_k^p - TR_j^p}{TR_j^p} + \frac{\sum_{k=1}^{|M(i)||M(j)|} \sum_{l=1}^{|M(j)|} D_{kl}^B - B_{ij}}{B_{ij}} + \frac{\sum_{k=1}^{|M(j)|} D_k^m - TR_j^m}{TR_j^m} & i, j \in S \text{ (1) and (2)} \\ & \text{and (3) and (4)} \\ \frac{\sum_{k=1}^{|M(j)|} D_k^p - TR_j^p}{TR_j^p} & j \in S \text{ (1) and (5)} \\ \frac{\sum_{k=1}^{|M(j)|} D_k^m - TR_j^m}{TR_j^m} & j \in S \text{ (2) and (7)} \\ \frac{\sum_{k=1}^{|M(i)||M(j)|} \sum_{l=1}^{|M(j)|} D_{kl}^B - B_{ij}}{B_{ij}} & i, j \in S \text{ (3) and (6)} \\ \alpha \frac{\sum_{k=1}^{|M(j)|} D_k^p - TR_j^p}{TR_j^p} + \beta \frac{\sum_{k=1}^{|M(i)||M(j)|} \sum_{l=1}^{|M(j)|} D_{kl}^B - B_{ij}}{B_{ij}} + \gamma \frac{\sum_{k=1}^{|M(j)|} D_k^m - TR_j^m}{TR_j^m} & i, j \in S, \alpha \text{ and } \beta \text{ and } \gamma \in [0,0.5] \\ & \text{((1) and (8)) or ((3) and (9)) or ((2)and(10))} \\ 0 & \text{else} \end{cases}$$

$$Min \sum_{j=1}^{|S|} X_j \quad X_j = \begin{cases} 1 & |M(j)| > 0 \\ 0 & \text{else} \end{cases} \tag{3}$$

These five groups include:

- The first groups include service providers whose level of used resources was more than valid value and also some virtual machines focused on processor, focused on memory and I/O are mapped on them. This group is determined based on formulas (4) to (7).
- The second groups include service providers whose level of processor usage was more than threshold and also they have some virtual machines focused on processor. These groups can be determined using statements (4) and (8).
- The third groups includes service providers whose level of memory usage was more than level determined and also they have some virtual machines focused on memory. This group can be specified using statements (5) and (10).
- The fourth group includes service providers whose connection links are more than threshold level and also some virtual machines with demand of communication (focused on I/O) are mapped on them. This group is specified based on statements (6) and (9).
- The fifth group (using statements (4),(11),(6),(12),(5) and (13)) includes service providers in which one or several resources in them may be more than authorized level. They don't have virtual machines to be able to influence on those resources.

$$TR_j^P - UR_j^P < TH^P \quad (4)$$

$$TR_j^m - UR_j^m < TH^m \quad (5)$$

$$\forall i, j \in S, B_{ij} - \sum_{k=1}^{|M(i)|} \sum_{l=1}^{|M(j)|} D_{kl}^B < TH^B \quad (6)$$

$$\exists k, l, z \in M(j), k \in I/O_VMset$$

and $l \in Cpu_VMset$ and $z \in Memory_VMset$

$$k = l = z \text{ or } k \neq l \neq z \quad (7)$$

$$\exists k \in M(j), k \in Cpu_VMset \quad j \in S \quad (8)$$

$$\exists k \in M(j), k \in I/O_VMset \quad j \in S \quad (9)$$

$$\exists k \in M(j), k \in Memory_VMset \quad j \in S \quad (10)$$

$$\forall k \in M(j), k \notin Cpu_VMset \quad j \in S \quad (11)$$

$$\forall k \in M(j), k \notin I/O_VMset \quad j \in S \quad (12)$$

$$\forall k \in M(j), k \notin Memory_VMset \quad j \in S \quad (13)$$

The point that is worth to mention is that some service providers may have feature of several groups together which in this situation value of all of these groups in formula (2) should be calculated for the service provider.

Next part multi-purpose genetic algorithm is presented.

4 DESCRIBING SUGGESTED MULTI-PURPOSE GENETIC ALGORITHM

Since mapping of virtual machines to service providers is a type of bin packing, we can apply group genetic algorithm (GGA) which was suggested by Falkenauer [28].

As the name of GGA suggests, it is a development of normal genetic algorithm which was compared for solving grouping problems. There are also other types of genetic algorithm for solving grouping problems but what made GGA prominent is a special coding which was suggested by Falkenauer. As a result, in this article we used GGA. Suggested multi-purpose genetic algorithm code is whons up next.

Input:

Set of VMs with their associated resource demands (Cpu_D , $Memory_D$, $Bandwidth_D$) and set of hosts with their existing resources (Cpu_E , $Memory_E$, $Bandwidth_E$)

Output:

Set P contains the mapping information of virtual machines to servers

/ Initialization Phase**

1. Set Values of parameters,

$\alpha, \beta, \gamma, TH^P, TH^m, TH^B, NV$ (number of virtual machines),

NI (number of iteration), NC (number of chromosomes in each population)

2. Grouping virtual machines in three categories.

($Cpu_VMset, Memory_VMset, I/O_VMset$)

3. Descending sorting virtual machines based on the formula 14

4. Creating initial population based on mapping of presorted virtual machines to servers that are randomly selected.

5. For all Chromosomes in first population calculate rank of each Chromosomes based on sum of formulas

(1),(2),(3) and save them in oldrank array

/ Iterative Phase**

6. While (number of iteration \leq NI)

7. For all Chromosomes in pre population

8. Randomly select two Chromosomes.

9. Merge two chromosomes are selected based on the grouping crossover*.

10. Import two new chromosomes in a temporary population.

11. End for

12. Do the grouping mutation* operation for all chromosomes in temporary population.

13. For all Chromosomes in temporary population calculate rank of each Chromosomes based on sum of

formulas (1),(2),(3)

14. For new population:

Select $K_1\%$ of new population from top ranking chromosomes of temporary population.

Select $K_2\%$ of new population from top ranking chromosomes of old population.

Select $K_3\%$ of new population randomly generated chromosomes.

15. End while

/* Solution Phase */

16. Select topmost chromosome from final population as Solution

* Grouping crossover and grouping mutation has been proposed in [28].

Figure 1. Suggested multi-purpose genetic algorithm

$$\frac{D_k^P}{\sum_{j=1}^{|S|} TR_j^P} + \frac{D_k^m}{\sum_{j=1}^{|S|} TR_j^m} + \frac{D_k^B}{\sum_{j=1}^{|S|} \sum_{i=1}^{|S|} B_{ij}} \quad k \in VM_List \quad (14)$$

Figure suggested multi-purpose genetic algorithm is shown. From now on suggested algorithm is called CMPGGA. This algorithm is formed on three steps of initialization, iterative and solution. In the first step at first parameters needed are valued and then three groups of virtual machines are created. next oblivious of grouping of previous step and based on formula 14, all the virtual machines are ordered in falling form. After arranging, virtual machines are mapped to service providers who were selected randomly and those with enough resources needed by virtual machines so that the first population can be gained. In the last phase of initialization, rank of each chromosome is calculated and saved in the first population. In the iterative step in each repetition a new population is created. In each repetition for all chromosomes of the previous generation, act of group mixing is done. Selection of each chromosome for mixing is in random from. Two chromosomes created in each mixture activity, enter a temporary population. After the end of mixing and entrance of all the chromosomes created in temporary population, act of group mutation will be done for all the chromosomes of the population. Then rank of each chromosomes of temporary population is calculated and saved. The last phase of Iterative step, is

establishing new population. Selection of chromosomes of new population is based on figure3. The reason behind random selection of the new population is not placing in the local minimum and creation of generations so that we can cover more areas of searching domain.

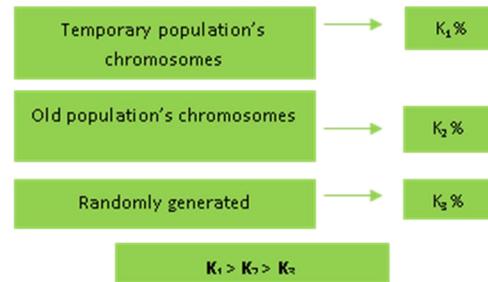


Fig. 2. Forming new population

In the final stage; Solution, among chromosomes of final population, the chromosomes with the highest level of significance is selected as the best VMPGGA solution.

5 ANALYSIS OF EFFICIENCY

To analyze efficiency of VMPGGA we created a simulator based on Java. To this, we consider a data center with 20 single-atom physical service. Features of these physical service providers are selected based on uniform distribution of values shown in table 2.

Table2. Features of physical service providers

Resource	Capacity
CPU (MIPS)	1000-3000
Memory (GB)	1-10
Network Bandwidth (Gbps)	1-5

Also 50to100virtual machines are considered for mapping on this data center. Features of virtual machines are also selected based on the same distribution of values presented in table 3.

Table 3. Features of virtual machines

Resource	Demand
CPU (MIPS)	250-1500
Memory (MB)	500-5000
Network Bandwidth (Mbps)	500-2500

Remaining needed parameters are selected based on table4

Table 4. Values of remaining parameters used

Parameters	Values
α, β, γ	0.5
TH^p	$0.005TR_i^p$
TH^m	$0.005TR_i^m$
TH^B	$0.005B_{ij}$
NC (number of chromosomes in each population)	20
NI (number of iteration)	50

To analyze efficiency of VMPGA, we compare it with two algorithms of Multi Objective First Fid Decreasing and Random. In latter algorithm first virtual machines are ordered based on value gained out of formula 14 in falling form and then in the list of service providers which are ordered in falling form and based on a formula similar to formula 14 , the first service provider with sufficient space for the virtual machine is selected and then virtual machine will be mapped in it. Next, results gained out of assessment will be presented.

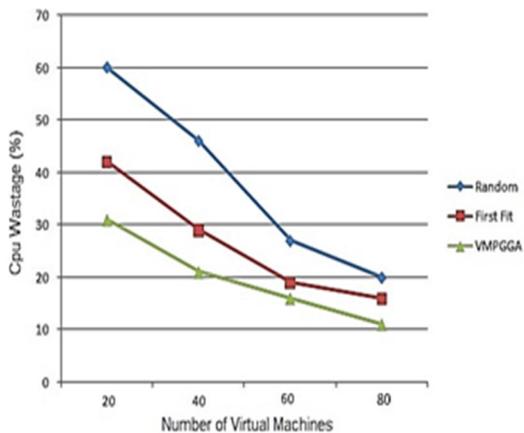


Fig. 3. level of wasted processing to various numbers of virtual machine

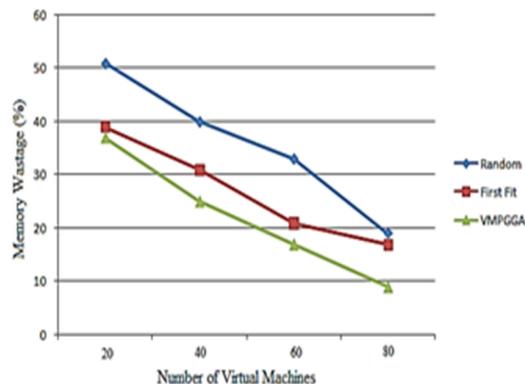


Fig. 4. level of wasted memory to various numbers of virtual machine

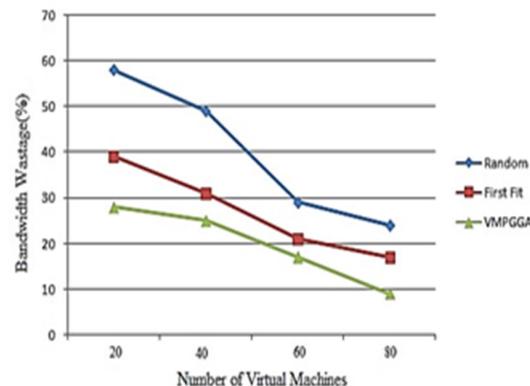


Fig. 5. Level of wasted bandwidth to various numbers of virtual machine

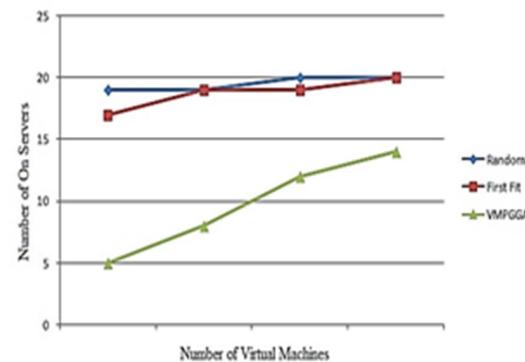


Fig. 6. Number of service providers used to various numbers of virtual machine

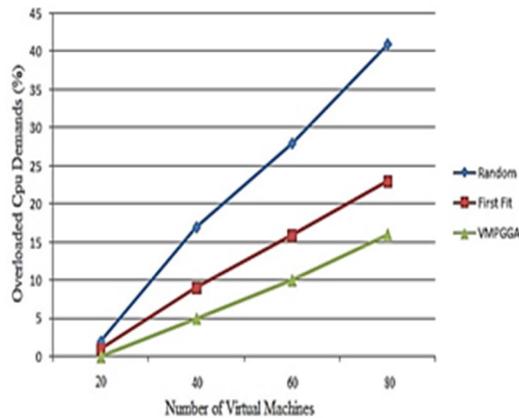


Fig. 7. Level of request of overload creator in using processor in service providers used to various numbers of virtual machine.

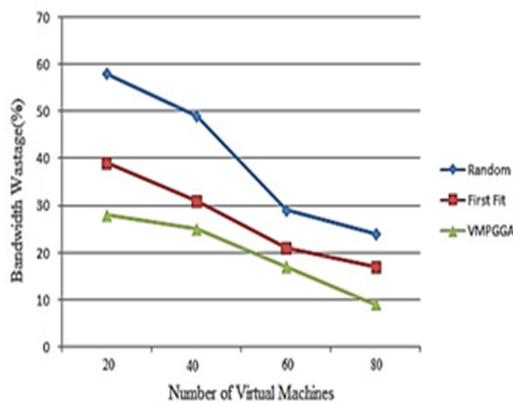


Fig. 8. Level of request of overload creator in using memory in service providers used to various numbers of virtual machine.

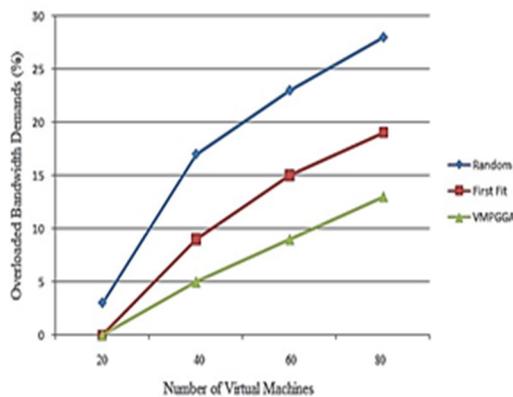


Fig. 9. Level of request of overload creator in using bandwidth in service providers used to various numbers of virtual machine.

6 CONCLUSION

Since most of tasks done in cloud computing do not care about workload of virtual machines and only care about optimization of an objective, in this article we focused our discussion on reducing operational costs resulting from wasting resource. By investigating tasks in which overloading of physical machines are not considered [4] [23], providence in resources wasted is not imposed [12] [4] also by activities in which overloading of physical machines and providence in resources wasted are not imposed [23] [25] [26] [27] we presented a method which deals with several goals for optimizing in mapping process by considering feature of workload of virtual machines. Then by comparing results gained from simulating suggested multi-purpose genetic algorithm with a random algorithm and an algorithm of the first multi-purpose choice based on factors such as level of processing loss, memory and bandwidth, number of service providers and also level of request of overload creator in using processor and service providers, we came to this conclusion that the algorithm suggested can establish better results on providence in resources and reducing operational costs by presenting an efficient method we hope to be able to achieve some results with less operational overload, less level of resources usage and at the same time the least energy consumption in cloud computing.

7 REFERENCES

- [1] A. Beloglazov, J. Abawajy, 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, Elsevier Science, 2012.8.
- [2] A. Dastjerdi, S. Garg, R. Buyya, "QoS-aware deployment of network of virtual appliances across multiple clouds", *Proceedings of the 3th IEEE International Conference on Cloud Computing Technology and Science*, 2011.11.
- [3] A. Khosravi, S. Garg, R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers", *Proceedings of the 19th International European Conference on Parallel and Distributed Computing*, 2013.10.
- [4] B. Speitkamp, M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers", *IEEE Trans. Services Computing*, vol. 3, no. 4, pp. 266-278, 2010.15.

- [5] B. Li, J. Li, J. Huai, T. Wo, Q. Li, L. Zhong, "Enacloud: an energy-saving application live placement approach for cloud computing environments", in Proceedings of the IEEE International Conference on Cloud Computing, 2009.19
- [6] C. Lin, P. Liu, J. Wu, "Energy-efficient virtual machine provision algorithms for cloud systems", in Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing, 2011.25
- [7] D. FILANI, J. HE, S. GAO, M. RAJAPPA, A. KUMAR, R. SHAH, R. NAAPPAN, "Dynamic data center power management:Trends, issues and solutions.", Intel Technology Journal, vol. 12, pp. 59-67, 2008.3
- [8] E. Feller, L. Rilling, C. Morin, "Energy-aware ant colony based workload placement in clouds", in Proceedings of the IEEE/ACM International Conference on Grid Computing (GRID), 2011.20
- [9] Emanuel Falkenauer: "Genetic Algorithms and Grouping Problems." John Wiley, 1998.28
- [10] F. Hermenier, X. Lorca, J. Menaud, G. Muller, J. Lawall, "Entropy: a consolidation manager for clusters", in Proceedings of the ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, 2009.17.
- [11] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in Proceedings of the 5th ACM USENIX Symposium on Networked Systems Design and Implementation, 2008.5.
- [12] H. Mi, H. Wang, G. Yin, Y. Zhou, D. Shi, L. Yuan, "Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers", in Proceedings of the IEEE International Conference on Services Computing, 2010.21.
- [13] H. Van, F. Tran, J. Menaud, "Performance and power management for cloud infrastructures", in Proceedings of the IEEE 3th International Conference on Cloud Computing, 2010.16.
- [14] J. Chase, D. Anderson, P. Thakar, A. Vahdat, R. Doyle, "Managing energy and server resources in hosting centers", in Proceedings of the 8th ACM symposium on Operating systems principles, USA: ACM, 2001.6.
- [15] J. Xu, J. Fortes, "Multi-objective virtual machine placement in virtualized data center environments", in Proceedings of the IEEE/ACM International Conference on Green Computing and Communications, 2010.22.
- [16] J. Yang, P. Liu, J. Wu, "Workload characteristics-aware virtual machine consolidation algorithms", Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science, 2012.12.
- [17] K. Kim, A. Beloglazov, R. Buyya, "Power-aware provisioning of virtual machines for real-time cloud services", Concurrency and Computation: Practice and Experience, vol. 23, no. 13, pp. 1491-1505, Wiley Press, 2011.9
- [18] L. Hu, H. Jin, X. Liao, X. Xiong, H. Liu, "Magnet: A novel scheduling policy for power reduction in cluster with virtual machines.", in Proceedings of the IEEE international conference on cluster computing , 2008.26
- [19] L. Liu, H. Wang, X. Liu, X. Jin, W. He, Q. Wang, Y. Chen, "Greencloud: a new architecture for green data center", in Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session, USA: ACM, 2009.27
- [20] M. Bichler, T. Setzer, B. Speitkamp, "Capacity planning for virtualized servers", in Proceedings of Workshop on Information Technologies and Systems (WITS), 2006.14
- [21] Q. Zhang, L. Cheng, R. Boutaba, "Cloud computing: state-of-the-art and research challenges", J. Internet Services Appl, vol. 1, no. 1, pp. 7-18, 2010.1
- [22] S. Chaisiri, B. Lee, D. Niyato, "Optimal virtual machine placement across multiple cloud providers", in: Proceedings of the IEEE Asia-Pacific Services Computing Conference, 2009.13
- [23] S. Srikantaiah, A. Kansal, F. Zhao, "Energy aware consolidation for cloud computing", in Proceedings of HotPower'08 Workshop on Power Aware Computing and Systems, 2008.18
- [24] S. Takeda, T. Takemura, "A rank-based vm consolidation method for power saving in datacenters", Information and Media Technologies, vol. 5, pp. 994-1002, 2010.7.
- [25] S. Takeda, T. Takemura, "A rank-based vm consolidation method for power saving in datacenters", Information and Media Technologies, vol. 5, no. 3, pp. 994-1002, 2010.24.
- [26] V. Tosic, H. Wada, A. Guabtini, K. Le-e, A. Liu, "Management towards reducing cloud usage costs", in Proceedings of the 7th IEEE Network Operations and Management Symposium (LANOMS), 2011.2
- [27] W. Lin, J. Wang, Ch. Liang, D. Qi, "A Threshold-based Dynamic Resource Allocation

- Scheme for Cloud Computing” , Procedia Engineering, vol. 23, pp. 695 – 703, Elsevier, 2011.23.
- [28] Y. Gao, H. Guan, Z. Qi, Y. Hou, L. Liu, “A multi-objective ant colony system algorithm for virtual machine placement in cloud computing”, Journal of Computer and System Sciences, vol. 79, no. 8, pp. 1230–1242, 2013.