Efficient Resource Allocation in Cloud Data Centers Through Genetic Algorithm

Ehsan Arianyan
Education & Research Institute for ICT
Tehran, Iran
ehsan_arianyan@itrc.ac.ir

Davood maleki
Education & Research Institute for ICT
Tehran, Iran
dmaleki@itrc.ac.ir

Alireza Yari
Education & Research Institute for ICT
Tehran, Iran
a_yari@itrc.ac.ir

Iman Arianyan
Education & Research Institute for ICT
Tehran, Iran
iman_aryanian@itrc.ac.ir

*Abstract***—** **Efficient resource scheduling and managing available resources is a great concern in modern datacenters. Cloud computing which is based on virtualization automates the process of distributing available resources among virtual machines. This paper proposes an enhanced version of resource allocation algorithm based on genetic algorithm. In other words, the decision accuracy in the process of virtual machine allocation to proper physical machines is enhanced. Usually, CPU and RAM of physical machines are considered in decision algorithms. However, this paper shows that to have a balanced cloud, it is necessary to consider more parameters in resource allocation algorithm. Moreover, the importance of considering weights for the parameters in decision algorithm is presented. It is shown that some parameters are more important than the others and should have higher impact in fitness function of genetic algorithm.**

Keywords- Genetic algorithm; Virtual machine; Physical machine; Cloud computing; Resource allocation.

I. INTRODUCTION

Cloud computing is emerging as a new and modern computing method in today's datacenters. Various heterogeneous resources, platforms, or software are provisioned on clouds through internet. The users can request the resources of the cloud anytime during their operation. The user's applications run on virtual machines (VM) specified for them. Based on the requirements of the application running on the VM, the cloud manager should provide appropriate resources for VM. Various kinds of applications running simultaneously on cloud have different resource requirements. So, allocating enough and appropriate resources to VMs is a challenging issue for resource manager. Virtualization technology which is the basis of cloud computing facilitates the process of sharing physical machines' (PMs) resources between VMs [1]. Hence, the resource manager aggregates all the available resources and enhances the resource utilization.

This paper takes advantage of genetic algorithm (GA) to solve the problem of resource allocation and proposes a new model to enhance the result of decision making process. A genetic algorithm is a search heuristic that imitates the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithm is the first evolutionary computation algorithm [2] which can solve many optimization problems and belongs to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

Often resource management tools consider only CPU, RAM, and disk parameters. Despite the other methods, this paper considers all the parameters which are important in resource allocation. Furthermore, this paper takes the importance of parameters into consideration by inserting weights of parameters in fitness function of genetic algorithm. Also, two resource allocation strategies are evaluated in two scenarios. The first strategy aims to consolidate all the resources in a way that VMs reside on the PMs that best fit for the application's resource needs. By so doing, the probability that some PMs become vacant is higher and we can turn them off. Thus, the power costs, cooling costs, and maintenance costs decrease. But, the second strategy puts VMs on PMs that have more available resources. Here, the applications have higher service level delivery and the probability of facing resource shortage problem is lower.

A. Paper Organization

In the sections that follow, first related works of other researchers is reviewed, and then GA scheduling algorithm is explained. In this section we discuss about chromosome representation, fitness function and constraints that are used in our algorithm. In section 4 we present all the input parameters that can be used to take an accurate decision for resource allocation. Section 5 describes the utilization rate equations defined to measure the efficiency of resource allocation algorithm. Then, simulation results that include effects of considering both the number and importance of input parameters is presented in section 6. Moreover, simulation results for "best fit" and "most available resource" scenarios are depicted in this section. The utilization rate and evolution time of genetic algorithm is also demonstrated in this section. Finally, conclusion is made in the last section.

978-1-4673-2073-3/12/$31.00 ©2012 IEEE

II. RELATED WORKS

There are many cloud resource management tools, three of which- that is Eucalyptus [3], OpenNebula [4], and Nimbus [5] are the most important. Each of these open source platforms has a specific resource provisioning mechanism.

Eucalyptus takes advantage of a scheduling algorithm that supports three types of policies: greedy or first fit, round robin and power save policy [3]. Greedy algorithm maps VMs to the first fit available node, and round robin algorithm allocates VMs to nodes based on round robin method [6].

OpenNebula balances the VMs taking advantage of advance reservation, preemption scheduling, and VM placement constraints [4]. The default scheduler module in OpenNebula is matchmaking scheduler (MMS). First of all, MMS leaves out the nodes that do not have enough resources to run the VMs [6]. Then, a rank is assigned to each resource available in resource pool based on the information gathered by the monitoring drivers. Finally, the VMs are allocated to the VMs based on their ranks. Another scheduler that can be integrated with OpenNebula is Haizia [6]. It uses leases as a fundamental resource provisioning abstraction .

Nimbus uses some customizable tools like PBS and SGE. PBS is queuing system and SGE uses job scheduling hierarchically [5].

Elana et al. implement a genetic algorithm in order to find the best scheduling solution. In addition to the VM capabilities, their algorithm also considers well defined policies in order to find the proper planning solution and is optimized for performance and scalability [6]. They consider a weighted sum of available CPU, RAM, and Disk in their decision algorithm.

Hai Zhong et al. use an improved genetic algorithm for the automated scheduling policy [7]. Their algorithm uses the shortest genes and introduces the idea of Dividend Policy in Economics to select an optimal or suboptimal allocation for the VMs requests. They consider a simple sum of CPU, RAM, and Disk in their fitness function.

Zhongni Zheng et al. propose an optimized scheduling algorithm to achieve the optimization or sub optimization for cloud scheduling problems [8]. They also consider a simple sum of CPU, RAM, and Disk in their fitness function.

III. THE SCHEDULING ALGORITHM

A. Chromosome Representatin

In this paper the chromosome size is equal to the number of VMs and its value is presented by an integer arrange ranging from 1 to the number of PMs. In other words, to map M VMs to N PMs, each chromosome encodes a scheduler which holds M genes to represent the placement of VMs on PMs. Hence, the value in the i'th gene states the VM's number that resides on j'th PM. For instance, Fig. 1 shows VM1 resides on PM1, VM2 reside on PM1, and VM_M resides on PM_x that x index is smaller or equal N.

PM_1	PM_1	…	PM_j	…	PM_x
VM_1	VM_2	…	VM_i	…	VM_M

Figure 1. Chromosom representation

B. Fitness Function

In this paper the following fitness function is maximized:

$$F = \sum_{i=1}^{M} \sum_{j=1}^{N} C_{ij} X_{ij} \tag{1}$$

Where

$$Xij = \begin{cases} 1 & \text{if } i^{th} \text{ Request is assigned to } j^{th} \text{ node} \\ 0 & \text{if } i^{th} \text{ Request is not assigned to } j^{th} \text{ node} \end{cases} \tag{2}$$

$$C_{ij} = \sum_{k=1}^{\#\text{deterministic par.}} W_k O_k + \sum_{l=1}^{\#\text{linguistic par.}} W_l R_l \tag{3}$$

$$O_k = \begin{cases} P_k & \text{if best fit scenario is used} \\ Q_k & \text{if most available resource scenario is used} \end{cases} \tag{4}$$

$$P_k = \begin{cases} \frac{VM_k}{PM_k} \times 1000 & \text{if } \frac{VM_k}{PM_k} \le 1 \\ -\infty & \text{if } \frac{VM_k}{PM_k} > 1 \end{cases} \tag{5}$$

$$Q_k = \begin{cases} \frac{PM_k}{VM_k} \times 1000 & \text{if } \frac{VM_k}{PM_k} \le 1 \\ -\infty & \text{if } \frac{VM_k}{PM_k} > 1 \end{cases} \tag{6}$$

$$R_l = \begin{cases} \frac{\sum_{s=1}^{M} TEMP_s}{TEMP_l} \times 1000 & \text{if parameter} = \text{Temperature} \\ \begin{cases} \frac{\#VM_l}{\sum_{s=1}^{M} \#VM_s} \times 1000 & \text{if parameter} = \#\text{VM in best fit scenario} \\ \frac{\sum_{s=1}^{M} \#VM_s}{\#VM_l} \times 1000 & \text{if parameter} = \#\text{VM in most available scenario} \end{cases} \\ \frac{QoS_l}{\sum_{t=1}^{N} QoS_t} \times 1000 & \text{if parameter} = \text{QoS} \end{cases} \tag{7}$$

Where "#linguistic par" and "#deterministic par" in equation 3, depicts the number of linguistic and deterministic parameters considered in our fitness function. These parameters are selected from table 1. It should be noted that, if all the parameters have the same weights, then W_k in equation 3 is equal to 1. Also, P_k is used in "best fit" scenario and Q_k is used in "most available resource" scenario. Moreover, if the requested resource is lower than available resource on PM, the value of P_k is set to a large value proportional to VM_k/PM_k in equation 5, and Q_k is set to a large value proportional to PM_k/VM_k in equation 6. Besides, PM_k and VM_k are the resource value available on PM and the resource value demanded by the VM, respectively. Resources are listed in table 1. It should be mentioned that equation 5 is used in "best fit" scenario and equation 6 is used in "most available resource" scenario.

C. *Constraints*

The constraints defined in equation 8 are set so that every VM inhibits on only one PM.

$$\sum_{j=1}^{N} X_{ij} = 1 \quad i = 1,2,..., M \tag{8}$$

$$\sum_{i=1}^{M} \sum_{j=1}^{N} X_{ij} \le M \tag{9}$$

IV. INPUT PARAMETERS

Six parameters that are used in decision making process are listed in table1. Every parameter is composed of some components depicted in third column of table 1. In most cases, CPU power which specifies the computational power a system can offer, is the most important parameter in decision making process [9]. The RAM and Disk parameters define the capacity and speed of RAM and disk respectively. Network parameter symbolizes network bandwidth and determines the amount of data that can pass through a network interface over time [9]. Temperature which shows the temperature of PMs is an important factor to decide where to put VMs. Sometimes, high sudden increase in temperature typify some defects in the system and this parameter can be used to establish a predicting failure mechanism [9]. QoS which stands for quality of service is a representative of requested service level agreement in decision process. In other words, the more money the user pays, the more QoS is set for his VM [9]. #VM which is the number of VMs residing on PMs, is an important factor to allocate VMs to PMs. More precisely, during evolution of genetic algorithm in best fit scenario, VMs are allocated to PMs in accordance with the number of VMs previously allocated to PMs. Thus, the more VMs abide on a PM, the more value of fitness function is for that PM. By so doing, VMs are first dedicated to PMs that more VMs are allocated to them and the probability of having vacant PMs is higher. Hence, such vacant PMs can be turned off and the maintenance costs are reduced. On the opposite side, in most available resource scenario VMs are allocated to PMs which have less previously allocated VMs.

TABLE I. INPUT PARAMETERS

No.	*Parameter*	*Parameter type*	*Components*	*Description*
1	CPU	Deterministic	CPU-cycle	CPU clock speed(GHZ)
2		Deterministic	CPU-core	Number of CPU cores
3		Deterministic	CPU-bus	CPU bus width and speed (MHZ)
4	RAM	Deterministic	RAM-cap	Node RAM capacity (GB)
5		Deterministic	RAM-Access time	RAM access speed (ms)
6		Deterministic	Cache-Hit Ratio	Cache-Hit Ratio
7		Deterministic	Cache-Access time	Cache Access time (ms)
8		Deterministic	Cache-CAP	Cache Capacity (MB)
9	Disk	Deterministic	Disk-cap	Capacity of Disk (GB)
10		Deterministic	HDD-Access time	Hard Access time (ms)
11	Network	Deterministic	NET-BW	Network bandwidth (Gbps)
12	Temperature	Linguistic	TEMP	temperature (K) for PMs
13	QoS	Linguistic	QoS	Quality of service for VMs
14	#VM	Linguistic	Number of VMs	Number of VMs on each PM

V. UTILIZATION RATE

The utilization rate which is the percentage of consumed resources divided by the total available resources, is an important measure to evaluate the performance of a system. For instance, CPU utilization in a PM determines utilization performance of leveraging CPU resource available on that machine. We define utilization rate of individual physical resources for parameters depicted in table 1, utilization rate of PMs, and total average system utilization rate according to equations 10 to 13.

$$U^{j}_{par} = \sum_{k} VM_k / PM_j \qquad k \in \{ VM_k \mid VM_k \to PM_j \} \tag{10}$$

$$U_{par} = \sum_{j=1}^{N} U^{j}_{par} / N \tag{11}$$

$$U_{PMj} = \sum\nolimits_{par} U^{j}_{par} / \text{number of parameters} \tag{12}$$

$$U_{ave}PMs = \sum_{j=1}^{N} U_{PMj} / N \tag{13}$$

Where U^{j}_{par} is the utilization rate of an especial parameter on PM_j which equals the summation on all the division values obtained by dividing the desired resource of VM_k resided on PM_j to available resource on PM_j. Moreover, U_{par} is the average utilization rate of an especial parameter on all PMs. For example, U_C, U_R and U_D are utilization rate of CPU, RAM and Disk, respectively. Besides, U_{PMj} is utilization rate of PM_j which is average of utilization rate of all the parameters of PM_j depicted in table 1. Also, $U_{ave}PMs$ is the average of utilization rate of all PMs.

VI. SIMULATION RESULTS

This paper simulates resource allocation scenario using Matlab software. Also, in all our test scenarios coming in this section PMs and VMs have the characteristics shown in table 2. It should be noted that, the values depicted in table 2 are normalized and have normalized units. Two scenarios are evaluated in this section: best fit scenario and most available resource scenario. Also, four experiments are executed to show the effects of considering new parameters in decision making process, considering parameters' importance in final results, the difference between "best fit" and "most available resource" scenarios, and to show the evolution time and utilization rate of resource allocation algorithm.

TABLE II. RESOURCES VALUES FOR TWO VMS AND THREE PMS

	CPU	RAM	Disk	Net
VM1	2	3	4	6
VM2	1	3	2	3
PM1	2	3	4	3
PM2	3	4	5	7
PM3	1	6	7	2

A. Effects of considering different number of parameters in allocaiton result

Since the cloud nodes are heterogeneous, they have different capabilities. Moreover, the scheduling algorithm should consider all the influencing parameters in decision making process. For instance, consider a problem of resource allocation which is aimed to be solved with best fit scenario. If we only consider CPU, RAM, and Disk parameters, the resource allocation algorithm have a solution which is shown in table 3. When NET is taken into account, the scheduling algorithm has a different solution for resource allocation which is mapping VM1 on PM2 and VM2 on PM1. Thus, in order to take a right decision we should take all these parameters into account.

In another test the effects of considering the number of VMs residing on PMs in addition to the CPU parameter is evaluated. The allocation results shown in table 4 states that when the #VMs parameter is taken into consideration in best fit scenario, VMs are first mapped to the PMs that have already more VMs on them. As a result, the probability that PMs remain vacant is higher and the empty PMs can be turned off. In this test, 2 PMs are turned off and the utilization rate of PMs is decreased. Consequently, there would be more savings in operational and maintenance costs.

TABLE III. ALLOCAITON RESULTS WHEN TAKE NET PARAMETER INTO CONSIDERATION IN ADDITION TO CPU, RAM, AND DISK

	Allocation Results with CPU, RAM, and Disk	***Allocation Results with CPU, RAM, Disk, and NET***
Final allocation	*(1,3)*	*(2,1)*

TABLE IV. ALLOCAITON RESULTS WHEN TAKE #VMS PARAMETER INTO CONSIDERATION IN ADDITION TO CPU PARAMETER USING BEST FIT SCENARIO

	Allocation Results with CPU	***Allocation Results with CPU and #VMs***
Final allocation	*(1,3)*	*(2,2)*
# of vacant PMs	*1*	***2***
U_{PMs}	*0.66*	***0.33***

B. Effect of considering parameters importance in allocation results

If we apply the effects of parameters weights in resource allocation process, the final mappings are different. For instance, consider the allocation problem for the system shown in table 2. The allocation results and also CPU, RAM, and Disk weights are shown in table 5. It is clear from this table that allocation results highly depend on the weights specified for input parameters. For instance, if we dedicate a greater weight for RAM parameter, the genetic algorithm maps VM2 to PM2 which have more RAM fitness. Hence, we should apply appropriate input parameters weights in genetic algorithm, so that final decisions become more precise.

Table 5 shows the utilization rates of CPU, RAM, and DISK when considering different weights for these parameters in best fit scenario in a system with characteristics shown in table 2. It is obvious from this table that the more the weights of each parameter, the more the utilization rate is for that parameter and vice versa. For instance, the fourth column of table 5 shows CPU utilization reduction by CPU weight reduction. Likewise, the fifth column of table 5 shows RAM utilization increase by RAM weight increase. However, since the configuration of this example is such that when increasing the value of CPU weight, the allocation algorithm presents the same response as the case of having the same weights for all of the parameters, the utilization rate remains the same in this case.

TABLE V. ALLOCATON RESULTS AND UTILIZATION RATES WHEN CONSIDERING DIFFERERNT WEIGHTS FOR CPU,RAM, AND DISK IN BEST FIT SCENARIO

W_{CPU},W_{RAM}, W_{DISK}	***1,1,1***	***10,1,1***	***-10,1,1***	***1,10,1***	***1,1,10***
Allocation Result	*(1,3)*	*(1,3)*	*(2,1)*	*(1,2)*	*(1,2)*
U_C	*0.66*	***0.66***	***0.38***	*0.44*	*0.44*
U_R	*0.50*	*0.50*	*0.58*	***0.58***	*0.58*
U_D	*0.42*	*0.42*	*0.43*	*0.46*	***0.46***

C. Best fit and most availabel resource scenarios

This paper evaluates two scenarios for resource allocation problem which are best fit scenario and most available scenario. The goal of best fit strategy is to map VMs on PMs that their available resources are best match to the resources that VMs request. In this scenario the probability of having vacant PMs is higher. Consequently, the unloaded PMs could be turned off, leading to fewer budgets for electricity, cooling system, and maintenance [10]. Public cloud infrastructure providers are looking for more benefits, and this scenario seems more appealing to them.

In "most available resource scenario", VMs are first map to the PMs that have more resources to service the VMs. So, the applications running on VMs have access to their required resources with a higher probability, and as a result have less computational time. This scheme is attractive for customers receiving service, who have a high tendency to execute their applications faster. Besides, this scheme is interesting in private cloud management to have applications with lower response times.

Table 6 shows that in "best fit scenario", because the requested resources of VM1 have more adaptability to the resources available on PM1, VM1 is allocated to PM1. Moreover, when we use "most available scenario", because there are more resources available on PM2, VM1 is allocated to PM2.

TABLE VI. ALLOCATON RESULTS USING "BEST FIT" AND "MOST AVAILABLE RESOURCE" SCENARIOS CONSIDERING CPU, RAM, AND DISK

	Final Point P Formula	Final Point Q Formula
1	(1,3)	(2,3)

D. Evolution time and utilization rate

Table 7 shows the allocation result of GA and three other probable allocations. As is clear from this table, the fitness function value of GA is more than the fitness function of other responses. Also, GA allocates VMs to PMs so that the average utilization rate of PMs is the most.

The PM configuration of the platform used to test the evolution time is the same as platform shown in table 2 but the parameters of VMs have all values equal to 0.25. Evolution time is shown in figure 2 by progressively incrementing the number of VMs. It is obvious from figure 2 that evolution time of genetic algorithm increases by adding the number of VMs. In this test, all the parameters shown in table 1 are taken into consideration in decision making process.

TABLE VII. ALLOCAITON RESULTS WITH CPU, RAM AND DISK WHEN USING GA IN BEST FIT AND MOST AVAILABLE RESOURCE SCENARIOS

	Allocation results	Fitness value	Average U_{PMs}
GA	(1,3)	**4785**	0.53
allocation 1	(1,2)	4483	0.49
allocation 2	(2,1)	4216	0.46
allocation 3	(2,3)	4002	0.44

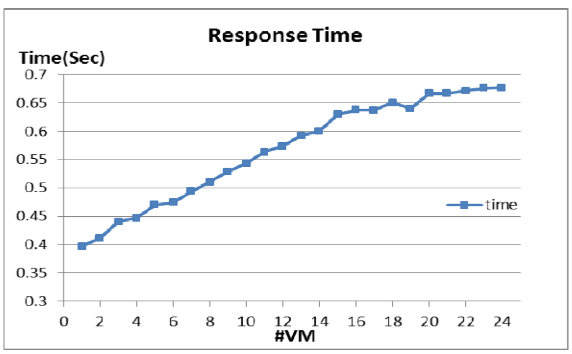

Figure 2. Response Time

VII. CONCLUSION

The issue of resource allocation in cloud environments was considered and genetic algorithm was adopted for resource scheduling. Results show that proposed model enhances the resource utilization and reduces operational and maintenance costs. Variant parameters that have influence on final decision were considered as input to allocation algorithm and their impact was evaluated in different test scenarios. It was shown that considering #VM parameter in decision algorithm leads to more savings in operational and maintenance costs. Moreover, the effect of considering the importance of parameters in decision algorithm was shown in simulation results. In other words, it was depicted that to have a precise decision, appropriate input parameters weights should be applied in genetic algorithm. Besides, "best fit" and "most available resource" scenarios were evaluated and their test results were shown. It was shown that "best fit" scenario tends to have more vacant PMs and to turn such PMs off. On the other hand, "most available resource" scenario tends to map VMs on PMs that have more available resources, so that the response time of applications decreases.

ACKNOWLEDGMENT

We thanks research institute for ICT that provide us platforms to test and run our system.

REFERENCES

[1] Ehsan Arianyan, Mohsen Tarighi, Mohammadreza Ahmadi, and Alireza Yari, “Virtual Machine Reallocation in Data Centers through Multi Parameter Decision Making Model”, International Conference on Information Communication and Management, 2011, pp. 211-215.

[2] Kai Zhu, Huaguang Song, Lijing Liu, Jinzhu Gao, Guojian Cheng, “Hybrid Genetic Algorithm for Cloud Computing Applications”, IEEE Asia-Pacific Services Computing Conference (APSCC), 2011, pp. 182 – 187.

[3] Eucalyptus,http://www.eucalyptus.com.

[4] OpenNebula, http://www.opennebula.org.

[5] Nimbus, http://nimbusproject.org.

[6] Elena Apostol, Iulia Baluta, Alexandru Gorgoi, Valentin Cristea. “Efficient Manager for Virtualized Resource Provisioning in Cloud Systems”, IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2011, pp. 511 – 517.

[7] Hai Zhong, Kun Tao, Xuejie Zhang, “An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems”,The fifth annual ChinaGrid conference, 2010, pp. 124-128.

[8] Zhongni Zheng, Rui Wang, Hi Zhong, Xuejie Zhang, “An Approach for Cloud Resource Schedulgin Based on Prallel Genetic Algorithm”, 3rd International Conference on Computer Research and Development (ICCRD), 2011, pp. 444 – 447.

[9] M. Tarighi, S.A Motamedi, and Ehsan Arianyan. “Performance Improvement of Virtualized Cluster Computing System Using TOPSIS Algorithm”, 40th International Conference on Computers and Industrial Engineering (CIE40), 2010, pp. 1-6.

[10] A. Beloglazov, J. Abawajy and R. Buyya. “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing”, journal elsevier, Future generation computer systems, vol. 28, no. 5, pp. 755-768.