A GA-Based Load Balancing Strategy for Virtual Machines in the Cloud Environment

He Luo †, Yanqiu Niu, Zhengzheng Liang, Xiang Fang
School of Management
Hefei University of Technology, Hefei, China
Tel: (+86) 18956078020, Email: luohe@hfut.edu.cn

Abstract. Virtualization technologies help people to use cloud without concern the detailed process in the cloud environment. Virtual Machines (VMs) can be replaced among different Physical Machines (PMs) based on the virtualization technology. In this paper, we propose a load balancing strategy for virtual machines so as to reduce the high cost of the PMs. The parameters of CPU, memory and disk are taken into account during the process of VMs placement, and an improved genetic algorithm is proposed considering PMs load balancing in the process of crossing and mutation.

Keywords: Virtual machine replacement, load balancing, cloud computing, genetic algorithm

1. INTRODUCTION

Cloud computing is a brand new service pattern based on the current internet. All elements are virtualized in the cloud environment so that users can enjoy different services easily without knowing the provider of the service and detailed process of the service. It is similar when we use water, electronic, or gas.

One of the key technologies is virtualization which provides a new way to manage resources in clouds. Virtual Machines (VMs) are generated and run on the Physical Machines (PMs). From the users' view, they are like using traditional computers which is actually provided by VMs. So, generally, there are no differences between using traditional computers and using VMs for a user. On the other side, different VMs can be run on the same PMs or different PMs depend on the strategy of resource management in Clouds. That is to say, VM managers take responsibility of VM so that users can get its service without concern about the detailed operational process of VMs. It becomes an important problem for VM managers to keep the effect of the running VMs facing the increasing number of cloud users and various demands of them. Therefore, users' demands have a directly impact on how to manage VMs and how to balance VMs among different PMs.

Facing the VMs Placement Problem (VMPP), the performance of CPU on the PMs is considered in the recently researches, and the goal of optimization is to maximize the utility of all the VMs given the constrains. However, during the process of cloud service, users may require various performances of VMs. For example, one user needs low performance of CPU such as 50Hz while needs high performance of hard disk such as 300GB for the storage task. The other user may also need low performance of CPU, but just need low performance of hard disk such as 10GB for the document task. The VMs allocated to these two users should not be the same, and then the VMs placed on the PMs will also in a different way. Due to the diversity of the user needs, more attributes of VMs should be taken into account during the VM placement. In this paper, we propose a genetic algorithm based load balancing strategy for virtual machine placement with multi-attributes in the cloud environment.

The main contributions of this paper including the following:(1) Multi-attributes of VMs. The parameters of CPU, memory and disk are taken into account during the process of VMs placement. It turns VMPP to a Three-Dimensional Packing Problem(TDPP) which is NP-hard problem. (2) Genetic Algorithm Based Load Balancing Strategy. The goal of the optimization is to maximize the physical machine utilization and minimize the physical machine load variance giving the constraints, and an improved genetic algorithm is proposed considering PMs load balancing in the process of crossing and mutation.

2. RELATED WORKS

In order to meet users' demand, VMs are generated and placed on the PMs. Owing to cloud suppliers' demands of ROI and users' demands of SLA, problems of VM placement have the following characteristics:

(1) Focusing on utilization of resource. Utilization of

resource is an important factor of cloud suppliers which is directly related to the operational costs of cloud suppliers. A problem of VM placement about multi-objective ant colony algorithm proposed by (Gao Y et al, 2013) is used to obtain solutions that can meet the demands, and reduce the waste of resources and energy. On the basis of the uncertain demand and prices in the future, optimization program of VM placement based on random integral programing is used to improve utilization of resource, reduce user costs proposed by (Chaisiri S et al, 2009). The optimal algorithm of overhead migration, placement of trade-off between energy and migrated times is an overall optimization strategy proposed by (Verma A et al, 2008).

(2) Demands of system's robustness. If load of VMs is completely determined by the user, it will cause the phenomenon that load of PMs is fluctuant. So, in order to make the operation of VMs stable, there are greater demands of robustness. For example, (N. Bobroff et al, 2007) propose a configured method of dynamic VMs to reduce violation of SLA, the result shows that this method can reduce violation of SLA by about 20%. A configured policy of dynamic VMs proposed by (Mi H et al, 2010) uses quadratic exponential smoothing to predict future workloads, uses genetic algorithm to refactor effectively, and experiments show that this strategy can improve utilization of resource, reduce consumption of data center. (E. Feller et al, 2011) propose a colony algorithm based on meeting current load, which is aim to make the number of PMs minimal. (Fang W et al, 2013) propose a VMPlanner way that can optimize position of VMs and traffic routing simultaneously, considering the network impact on the data center, and analyzing topology's characteristic and traffic patterns of the data center.

(3) Meeting different demands of preferences. Users' preferences of properties are different. For example, some users need resources of intensive computing who have the higher preference for CPU properties. Some users need storage-intensive resources who have the higher preference for hard disk properties. Some users need communication-intensive resources who have the higher preference for bandwidth properties. Therefore, complementary should be considered in the VM placement, in order to improve utilization of resource. For example, (Li X, et al, 2013) propose a divided model of multidimensional space, with considering the phenomenon that utilization of multidimensional resource is unbalanced, which can reduce consumption by balancing the utilization of multidimensional resource, reducing the number of run PMs.

3. PROBLEM DESCRIPTION

The problem of VM placement can be further described as that N independent virtual service resource which is described as $VM = \{VM_1, VM_2, \ldots, VM_N\}$ are placed in M PMs which are described as $Datacenter = \{host_1, host_2, \ldots, host_M\}$. In the T period which denotes cycle of placement, utilization of data center's resources needs to be improved while overall properties of system need to be ensured.

Definition 1: *Load of PMs' CPU. It is equal to the ratio that is all number of CPU in the VMs to all number of CPU in the same PM.*

$$HLC = \frac{\sum_{i=1}^{m} VC_i}{HC} \tag{1}$$

Let HC denote the number of CPU in the PM H_j and VC_i denote usage of CPU in the VMs.

Definition 2: *Load of PMs' memory. It is equal to the ratio that is all number of memory in the VMs to all number of memory in the same PM.*

$$HLM = \frac{\sum_{i=1}^{m} VM_i}{HM} \tag{2}$$

Let HM denote the number of memory in the PM H_j and VM_i denote usage of memory in the VMs.

Definition 3: *Load of PMs' hard disk. If H_j, which denotes PMs has m VMs, load of PMs' hard disk is equal to the ratio that is all number of hard disk in the VMs to all number of hard disk in the same PM.*

$$HLD = \frac{\sum_{i=1}^{m} VD_i}{HD} \tag{3}$$

Let HD denote the number of hard disk in the H_j and VD_i denote usage of hard disk in the VMs.

Definition 4: *Load of PMs H_j that is equal to usage of one PM. Load of PMs H_j can be calculated via:*

$$HL = \omega_1 HLC + \omega_2 HLD + \omega_3 HLM \tag{4}$$

where ω_1, ω_2, ω_3 denote different load weight of source, and different weights show preferences of different properties when users use VMs.

Definition 5: *Utilization of PMs that is equal to utilization of data center' resource. If there are m PMs and N VMs, utilization of PMs is calculated using the following equations. And the value is equal to the load average of PMs.*

$$HAU = \omega_1 \frac{\sum_{i=1}^{N} VC_i}{\sum_{j=1}^{M} HC_j \times HT_j} + \omega_2 \frac{\sum_{i=1}^{N} VM_i}{\sum_{j=1}^{M} HM_j \times HT_j} + \omega_3 \frac{\sum_{i=1}^{N} VD_i}{\sum_{j=1}^{M} HD_j \times HT_j} \tag{5}$$
$$= \frac{1}{\sum_{j=1}^{M} HT_j} \sum_{j=1}^{\sum_{j=1}^{M} HT_j} HL_j$$

Let HT_j denote whether H_j is open and ω_1, ω_2, ω_3 denote different load weights of source. Different weights show different users' focus of demands. Meantime, one target

of utilization is to make the number of PMs minimal.

Definition 6: load variance of PMs. It shows discrete degree of PMs' load and average load. It can be calculated via:

$$\sigma = \sqrt{\frac{1}{\sum_{j=1}^{M} HT_j} \sum_{j=1}^{\sum_{j=1}^{M} HT_j} (HL_j - AHL)^2} \quad (6)$$

where AHL denotes average load which can be calculated via:

$$AHL = \frac{1}{\sum_{j=1}^{M} HT_j} \sum_{j=1}^{\sum_{j=1}^{M} HT_j} HL_j \quad (7)$$

Let vector S denote disposition scheme of VMs. The problem of PMs placement can be transformed into a multi-objective optimization problem. The target is to make utilization of PMs maximum which can be described as MAX HAU(S) and make load variance of PMs minimal which can be described as MIN $\sigma(S)$. Properties of VMs and PMs need to be satisfied with the following constraints.

$$x_{ij} \in \{0,1\}, i = 1,2,3,\cdots,N; j = 1,2,3,\cdots,M \quad (8)$$

$$\sum_{i=1}^{N} VC_i x_{ij} \le HC_j, j = 1,2,3\cdots M \quad (9)$$

$$\sum_{i=1}^{N} VM_i x_{ij} \le HM_j, j = 1,2,3\cdots M \quad (10)$$

$$\sum_{i=1}^{N} VD_i x_{ij} \le HD_j, j = 1,2,3\cdots M \quad (11)$$

$$\sum_{j=1}^{M} x_{ij} = 1, i = 1,2,3\cdots N \quad (12)$$

Let vector N denote the number of VMs which needs to be placed and M denote the number of PMs in the data center. Let vector VC_i denote the size of CPU, vector VM_i denote the size of memory, and vector VD_i denote the size of hard disk in the VM i. Let vector HC_j denote the size of CPU, vector HM_j denote the size of memory, vector HD_j denote the size of hard disk in the VM j. x_{ij} is binary vector in the equation (8), and if VM i is placed in PM j, then x_{ij} is equal to 1, else x_{ij} is equal to 0. Equations from (9) to (11) show that the sum of a type of resource, which is needed in the process of VM placement, has to be less than or equal to the total value of the PM. Equation (12) shows the unique constraint that any VMs has to be placed in one PM.

4. THE IMPROVED GENETIC ALGORITHM

According to the characteristics of the above problems, it can be equal to a bin-packing problem which is three dimensions and variable size. However, it is a NP-hard problem that N VMs are placed in M PMs. Meantime, the problem is a multi-objective problem of optimization which needs to be considered operation rate of PMs, utilization of PMs and load balancing of PMs. In this regard, the paper proposes an improved genetic algorithm to solve the above problems.

4.1 Codec

It is a basic problem of genetic algorithm that encodes the above problems. Chromosomes of solving problems have to contain two parts: Distinguish different PMs for open code of PMs; Identify how to make VMs place in suitable PMs.

Meanwhile, owing to the requirement of problem solving that makes the number of open VMs minimal, the gene length of chromosome encoding is uncertain.

The paper uses map <K,V> to show the structure of the solution, that's because the relationship of PMs and VMs is one-to-many. Let vector K denote ID of VMs, vector V denote ID of PMs. Any K can be mapped a unique V, one V can be mapped many V. And an example of <K,V> shows a chromosome.

4.2 Generate initial solution

The problem needs to be considered the utilization of VMs and the property of load balancing. The steps that generate initial solution of VM placement for the number of initial populations are as follows:

Step 1: calculate load of the current status according to equation (4).

Step 2: calculate ratio p between every resource of VM and total resource VMs.

Step 3: place VMs in open PMs with probability p and make load of PMs minimal. If the PM is unsatisfied, then it needs to restart a PM and update load of PM.

4.3 Fitness function

Fitness function is the evaluated standard of solutions in genetic algorithm, and the greater fitness is, the greater solution is. The evaluated standard of traditional bin-packing problem is to make the number of boxes minimal while ensures high utilization. However, the problem of VM placement in IaaS differs from traditional bin-packing problem. It not only needs to consider the number of open PMs and the utilization of PMs, but also needs to consider the impact of properties on load of VMs. The smaller the load variance of PMs is, the greater the stability and robustness of PMs are. So evaluated fitness function is as follows with considering above factors:

$$Fitness = \lambda_1 \times HAU + \lambda_2 \frac{1}{\sigma} \quad (13)$$

where let λ_1, λ_2 denote correspond weighting factor and $\lambda_1>0$, $\lambda_2>0$.

4.4 Genetic operators

(1) select operations

This algorithm uses the classic roulette choosing method that is used to determine selected probability of the chromosome according to the ratio between every chromosome and the population. Specific steps are as follows:

Step 1: calculate the fitness of individual population according to equation (13);

Step 2: get fitness according to step 1, calculate the ratio of every individual fitness.;

Step 3: choose individual according to roulette method.

When uses roulette method, the greater the individual fitness is, the bigger the probability of choice is. And it ensures a greater fit individual to be preserved, and ensures solution to have a better global convergence. In the meantime, there is also the possibility that small fitness of individual has been selected which is avoid to make solution trapped in local optima.

(2) Crossover

Crossover is a core operator of genetic algorithm whose properties is largely decided by crossover. Crossover has to consider two principles, the one is that new solution after cross is a feasible solution, another is that the search space can be increased after cross. Specific steps are as follows:

Step 1: select chromosomes by round robin to select two chromosomes T1, T2 randomly;

Step 2: select VMs and select VMs of PMs as the part of cross;

Step 3: delete VMs and use the principle of seeking common ground that is to preserve the command mapping' VMs of PMs in chromosome T1, T2 and delete different mapping of PMs;

Step 4: Insert VMs and use principle of greed to make deleted VMs reinsert PMs.

(3) Mutational operator

Mutational operator is an important operator of genetic algorithm, which maintains the diversity of the population and adjusts the loci of individual population. Specific steps are as follows:

Step 1: Set mutational probability Pm and select a chromosome according to mutational probability randomly.

Step 2: select two PMs in the two selected chromosomes.

Step 3: exchange VMs of the two PMs which are selected randomly.

Step 3: randomly selected two physical machine virtual machine were exchanged;

Step 4: after the end of step 3, if the chromosome is a feasible solution, then the mutational process is ended, else the mutational operation is restarted.

The basic flow of BLGA algorithm is as follows:

Step 1: make the population initial, set the parameters, including the maximum times of iterations Gmax, cross probability Pc and genetic probability Pm;

Step 2: calculate the individual fitness of the population according to equation (13);

Step 3: determine whether the condition of termination is met, if the condition is met, optimal solution is outputted, else enters into step4;

Step 4: utilize roulette method, select individual of population into the next generation;

Step 5: generate randomly a number r between [0,1], and determine whether r <Pc is met. If the condition is met, the crossover operation is performed to form a new individual, else goes to step6;

Step 6: generate randomly a digital q between [0,1], and determine whether q <Pm is met, if the condition is met, the crossover operation is performed to form a new individual, else goes to step2.

5. EXPERIMENTS

5.1 Experimental environment

During the experiment, the memory of host is 3G, the hard disk is 650G and the CPU is 3.20GHz. The experiment uses My Eelipse8.5 and jdk1.6.0_10 to run in the Windows XP and uses java to program.

Firstly, we set relevant parameters of the experiment. The experiment sets 12 PMs to consist of the data center, and there are 4 different types of PMs, and every type has 3 PMs. 4 types of PMs' configuration are shown in Table 1.

Table 1. Four different configurations of Physical Machines

PM	CPU(HZ)	Memory（GB）	Disk(GB)
1	1000	4	2000
2	1500	8	2500
3	2000	6	1500

4	3000	6	1000

Table 2. List of Virtual Machines

VMs	CPU（HZ）	Memory（M）	Disk（GB）
1	50	384	56
2	110	473	100
3	156	952	354
4	287	456	286
5	445	998	385
6	221	524	225
7	498	998	58
8	348	554	96
9	226	1024	256
10	68	788	468
11	99	678	425
12	145	884	412
13	278	768	245
14	338	984	354
15	421	397	68
16	64	481	428
17	245	672	79
18	335	754	59
19	477	851	114
20	244	421	228
21	114	578	338
22	158	675	227
23	258	542	338
24	438	924	447
25	423	925	168

In the experiment, 25 VMs are set. Different types of users on the VMs' demands are different because the size of VMs is completely determined by the user. Some users belong to compute-intensive users whose demands for CPU and memory are large. Some users belong to storage-intensive users whose demands for hard drives are large.

Therefore, in order to prevent accidental experimental results, the specific parameters of the experiment in the VM is generated within a certain range randomly.

In the experiment, set the parameters of VM, the CPU is 50HZ ~ 500HZ, the memory is 384M ~ 1024M, and the hard disk is 50G ~ 200G. So the specific demands of 25 VMs which are generated randomly are shown in Table 2.

5.2 Feasibility analysis

In this experiment, let maximum iterative testing Gmax be equal to 200, the population size be 20, crossover probability Pc be equal to 0.8, mutational probability Pm be equal to 0.15, λ_1=100 and λ_2=1. The CPU, memory and hard disk are equally important while ω_1, ω_2, ω_3 are equal to 1/3. The experimental results about utilization and load balancing of VMs which use many experiments to get by BLGA algorithm are shown in Figure 1.

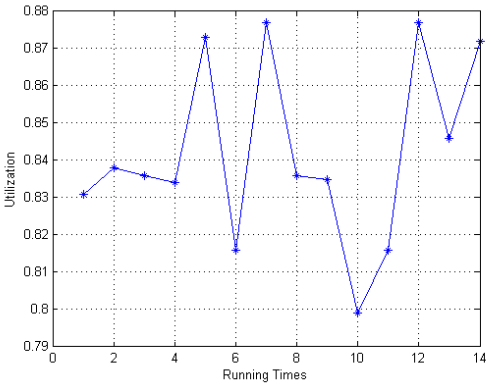

Figure 1 The utilization of PMs

According to the results which are run many times, we can get that the fluctuant range about utilization of PMs is [0.796-0.874]. Although utilization of PMs fluctuates, it remains at 0.8 or more, and the result is great. That's because the goal of BLGA algorithm is to improve utilization of resource about services. And it uses genetic algorithm crossover and mutational operators to select better chromosomes of fitness. Therefore, the PM can get better

utilization of resource. The experimental results show that BLGA for solving the problem of VMs placement is a viable strategy. The variance of load balancing also has the similar result which is shown in Figure 2.

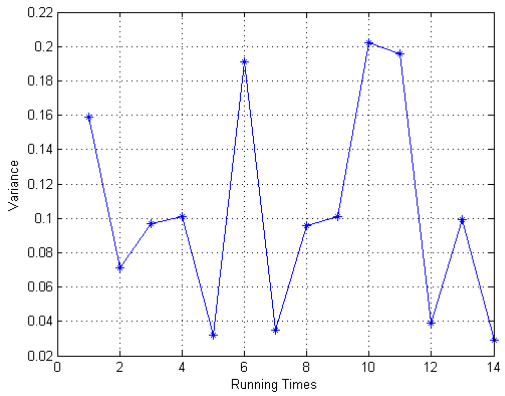

Figure 2 The variance of load balancing of PMs

6. CONCLUSION

In this paper, we propose a GA-based load balancing strategy for virtual machines in the cloud environment. The parameters of CPU, memory and disk are considered during the process of VMs placement, and the goal of the optimization is to maximize the physical machine utilization and minimize the physical machine load variance giving the constraints. An improved genetic algorithm is also suggested considering PMs load balancing in the process of crossing and mutation. The experiments show the effect of this strategy.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 71401048, 71131002, 71472058, and by Anhui Provincial Natural Science Foundation under Grant 1508085MG140.

REFERENCES

Gao Y, Guan H, Qi Z, Hou Y, Liu L. (2013) A multi-objective ant colony system algorithm for virtual machine placement in cloud computing[J]. Journal of Computer and System Sciences, 79(8): 1230-1242.

Chaisiri S, Lee B S, Niyato D. (2009) Optimal virtual machine placement across multiple cloud providers[C]. Proceedings of the IEEE Asia-Pacific Services Computing Conference, Singapore, 103-110.

Verma A, Ahuja P, Neogi A. (2008) pMapper: power and migration cost aware application placement in virtualized systems[J]. Middleware, 243-264.

Bobroff N, Kochut A, Beaty K. (2007) Dynamic placement of virtual machines for managing sla violations[C]. Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, Munich, Germany, 119-128.

Mi H, Wang H, Yin G, Zhou Y, Shi D, Yuan L. (2010) Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers[C]. Proceedings of the IEEE International Conference on Services Computing, Miami, Florida, 514-521.

Feller E, Rilling L, Morin C. (2011) Energy-aware ant colony based workload placement in clouds[C]. The Proceedings of the IEEE/ACM International Conference on Grid Computing, Lyon, France, 26-33.

Fang W, Liang X, Li S, Chiaraviglio L, Xiong N. (2013) VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers[J]. Computer Networks, 57(1): 179-196.

Li X, Qian Z, Lu S, Wu J. (2013) Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center[J]. Mathematical and Computer Modelling, 58(5): 1222-1235.