



**Queensland University of Technology**  
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

Wu, Yongqiang, Tang, Maolin, Tian, Yu-Chu, & Li, Wei (2012) Energy-efficient virtual machine placement in data centers by genetic algorithm. In *Lecture Notes on Computer Science*, Springer, Renaissance Doha City Center Hotel, Doha, pp. 315-323.

This file was downloaded from: <http://eprints.qut.edu.au/53767/>

© Copyright 2012 Springer

**Notice:** *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

# Energy-efficient Virtual Machine Placement in Data Centers by Genetic Algorithm

Grant Wu<sup>1</sup>, Maolin Tang<sup>1</sup>, Yu-Chu Tian<sup>1</sup>, and Wei Li<sup>2</sup>

<sup>1</sup> School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD 4001, Australia,  
`{m.tang, y64.wu, y.tian}@qut.edu.au`

<sup>2</sup> School of Information and Communication Technology, Central Queensland University, Rockhampton, QLD 4702, Australia  
`w.li@qut.edu.au`

**Abstract.** Server consolidation using virtualization technology has become an important technology to improve the energy efficiency of data centers. Virtual machine placement is the key in the server consolidation. In the past few years, many approaches to the virtual machine placement have been proposed. However, existing virtual machine placement approaches to the virtual machine placement problem consider the energy consumption by physical machines in a data center only, but do not consider the energy consumption in communication network in the data center. However, the energy consumption in the communication network in a data center is not trivial, and therefore should be considered in the virtual machine placement in order to make the data center more energy-efficient. In this paper, we propose a genetic algorithm for a new virtual machine placement problem that considers the energy consumption in both the servers and the communication network in the data center. Experimental results show that the genetic algorithm performs well when tackling test problems of different kinds, and scales up well when the problem size increases.

## 1 Introduction

The ever increasing cloud computing has been resulting in ever increasing energy consumption and therefore overwhelming electricity bills for data centers. According to Amazon's estimations, the energy-related costs at its data centers account for 42% of the total operating cost. In addition, the ever increasing energy consumption may lead to dramatically increase in carbon dioxide emissions. So, it is desirable to make every possible effect to reduce the energy consumption in cloud computing.

Server consolidation using virtualization technology has become an important technology to improve the energy efficiency of data centers [1,2,3,4]. Virtual machine (VM) placement is the key in the server consolidation. In the past few years, many approaches to various VM placement problems have been proposed. However, existing VM placement approaches do not consider the energy consumption in communication network in the data center. However, the energy

consumption in the communication network in a data center is not trivial, and therefore should be considered in VM placement in order to make the data center more energy-efficient.

In this paper, we propose a genetic algorithm (GA) [5] for a new VM placement problem that considers the energy consumption in both the physical servers (PMs) and the communication network in the data center. Experimental results show that the genetic algorithm performs well with various test problems, and scales well when the problem size increases.

The remaining paper is organized as follows: Section 2 formulates the new VM placement problem; Section 3 presents the GA; Section 4 evaluates the performance and scalability of the GA; and finally Section 5 concludes this work.

## 2 Problem Formulation

Let's define

- $V$  a set of virtual machines
  - $P$  a set of physical machines
  - $v_i$  a virtual machine in  $V$
  - $v_i^{cpu}$  the CPU requirement of  $v_i$
  - $v_i^{mem}$  the memory requirement of  $v_i$
  - $p_j$  a physical machine in  $P$
  - $p_j^{cpu}$  the CPU capacity of  $p_j$
  - $p_j^{mem}$  the memory capacity of  $p_j$
  - $p_j^{w_{cpu}}$  the total CPU workload on  $p_j$
  - $p_j^{w_{mem}}$  the total memory workload on  $p_j$
  - $V_{p_j}$  the set of virtual machines assigned to physical machine  $p_j$
- The utilization rate of the CPU in physical server  $p_j$  is

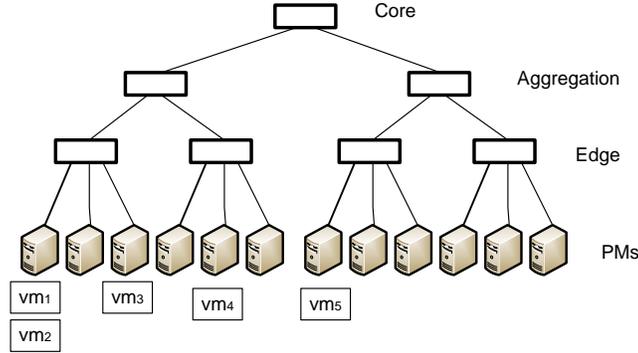
$$\mu_j = p_j^{w_{cpu}} / p_j^{cpu} \quad (1)$$

Thus, according to the server energy consumption model defined in [6], the energy consumption of physical server  $p_j$  when its CPU usage is  $\mu_j$  is

$$E(p_j) = k_j \cdot e_j^{max} + (1 - k_j) \cdot e_j^{max} \cdot \mu_j \quad (2)$$

where  $k_j$  is the fraction of energy consumed when  $p_j$  is idle;  $e_j^{max}$  is the energy consumption of physical server  $p_j$  when it is fully utilized; and  $\mu_j$  is the CPU utilization of  $p_j$ .

It is assumed that the communication network topology of the data center is a typical three-tier one as shown in Fig. 1 [7]. The VMs in the data center may communicate with each other through the communication devices, such as switches, which also consume a non-trivial amount of energy and it has been shown that this energy consumption is largely independent of the load through the communication devices [8]. Thus, we use the following method to approximate the energy consumption in the communication network in the data center.



**Fig. 1.** The communication network of a data center

We categorize the communication between a pair of VMs into four types: The first type is that the pair of VMs are on the same PM. The communication between  $vm_1$  and  $vm_2$  in Fig. 1 is an instance of the first type. The second type is that the pair of VMs are placed on two different PMs, but under the same edge. The communication between  $vm_1$  and  $vm_3$  in Fig. 1 is an example of the second type. The third type is that the pair of VMs are placed on two different PMs under different edges, but under the same aggregation. The communication between  $vm_3$  and  $vm_4$  in Fig. 1 is an example of the third type. The fourth is that the pair of VMs are placed on two different PMs under different edges and different aggregations. The communication between  $vm_4$  and  $vm_5$  in Fig. 1 is an example of the fourth type.

The first type of communication does not use any network communication device; the second type of communication uses one network communication device; the third communication involves in three network communication devices; and the fourth type of communication is done through five network communication devices. Therefore, the energy consumptions incurred by the four types of communication are different. In fact, the first type of communication does not incur any energy consumption in the communication network; the energy consumption of the second type communication is less than that of the third type, which is in turn less than that of the fourth type as the more network communication devices are used, the more energy is consumed in the communication network.

Let  $C_1$ ,  $C_2$ ,  $C_3$  and  $C_4$  be the sets of VM pairs between which there exists communication and the type communication belong to the first, second, third and fourth, respectively; and

$$C = C_1 \cup C_2 \cup C_3 \cup C_4 \quad (3)$$

For each communication  $c \in C$ , the energy consumption for transferring a unit of data is

$$e(c) = \begin{cases} 0, & \text{if } c \in C_1; \\ e_2, & \text{if } c \in C_2; \\ e_3, & \text{if } c \in C_3; \\ e_4, & \text{if } c \in C_4; \end{cases} \quad (4)$$

Let  $l(c)$  be the amount of data that need to be transferred on the communication  $c$ . Then, the network energy consumption for transferring  $l(c)$  units of data is

$$E(c) = e(c) * l(c) \quad (5)$$

the virtual machine placement problem is to assign each virtual machine in  $V$  onto a physical machine in  $P$ , such that

$$\sum_{p_j \in P} E(p_j) + \sum_{c \in C} E(c) \quad (6)$$

is minimized  
subject to

$$\bigcup_{p_j \in P} V_{p_j} = V \quad (7)$$

$$V_{p_i} \cap_{p_i \neq p_j} V_{p_j} = \emptyset \quad (8)$$

$$p_j^{w_{cpu}} = \sum_{v_i \in V_{p_j}} v_i^{cpu} \leq p_j^{cpu} \quad (9)$$

$$p_j^{w_{mem}} = \sum_{v_i \in V_{p_j}} v_i^{mem} \leq p_j^{mem} \quad (10)$$

Constraints (7) and (8) make sure that each virtual machine will be assigned to one and only one physical machine; constraints (9) and (10) guarantee that the total CPU workload and the total memory on physical machine  $p_j$  will not exceed the CPU capacity and the memory capacity, respectively.

### 3 Genetic Algorithm

This section entails the GA for the VM placement problem. It discusses in detail the encoding scheme, genetic operators and fitness function of the GA as well as the description of the GA.

#### 3.1 Encoding scheme

A chromosome in this GA consists of  $|V|$  genes, each of which stands for a virtual machine. The value of a gene is a positive integer between 1 and  $|P|$ , representing the physical machine where the virtual machine is allocated. Fig. 2 shows a example VM placement and its corresponding chromosome.

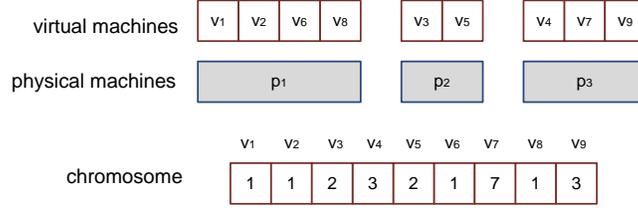


Fig. 2. An example of VM placement and its corresponding chromosome

### 3.2 Crossover

Since the length of chromosome is potentially long, linkage is a potential problem that should be considered. Because of this consideration, the GA adopts a biased uniform crossover operator, which is described in Algorithm 1.

---

#### Algorithm 1: Biased Uniform Crossover

---

**Input** : two parent chromosomes,  $C^i = x_1^i x_2^i \dots x_n^i$  and  $C^j = x_1^j x_2^j \dots x_n^j$   
**Output**: one child chromosome,  $C^k = x_1^k x_2^k \dots x_n^k$

- 1  $f^i \leftarrow fitness(C^i)$ ;
- 2  $f^j \leftarrow fitness(C^j)$ ;
- 3 **for**  $q = 1$  to  $n$  **do**
- 4     randomly generate a real value between 0 and 1,  $r$ ;
- 5     **if**  $r < f^i / (f^i + f^j)$  **then**
- 6          $x_q^k \leftarrow x_q^i$ ;
- 7     **end**
- 8     **else**
- 9          $x_q^k \leftarrow x_q^j$ ;
- 10    **end**
- 11 **end**
- 12 output  $C^k$ .

---

### 3.3 Mutation

The mutation operator simply randomly picks up a gene in the chromosome and inverts the value of the chosen gene. Algorithm 2 shows how the mutation operator works.

### 3.4 Fitness function

The fitness of an individual  $x$  in the population of the GA is defined in Eq. 11 below:

---

**Algorithm 2: Mutation**

---

**Input** : a chromosome,  $C = x_1x_2 \cdots x_n$   
**Output**: a mutated chromosome,  $C' = x'_1x'_2 \cdots x'_n$

- 1  $C' \leftarrow C$ ;
- 2 randomly generate a virtual machine  $i$ , where  $1 \leq i \leq |V|$ ;
- 3 randomly generate a physical machine  $p$ , where  $1 \leq p \leq |P|$ ;
- 4 replace  $x'_i \leftarrow p$ ;
- 5 output  $C'$ .

---

$$fitness(x) = \begin{cases} E^{min}/E(x), & \text{if } x \text{ is feasible;} \\ E^{min}/(E(x) + E^{max}), & \text{otherwise.} \end{cases} \quad (11)$$

where  $E^{min}$  is a lower boundary of the total energy consumption,  $E^{max}$  is an upper boundary of the total energy consumption, and  $E(x)$  is the total energy consumption when VM placement  $x$  is adopted.

The fitness function penalizes a solution that violates any of those constraints, and make sure that the fitness value of any infeasible solution is less than that of any feasible solution and that the less energy consumption and the greater the fitness value is.

### 3.5 The description of the GA

Algorithm 3 is a high-level description of the GA.

## 4 Evaluation

The GA has been implemented in Java. Since there are no benchmarks available for the new VM placement problem, we have to randomly generate test problems to test the GA. We use a set of experiments to evaluate the proposed GA with respect to performance and scalability. Table 1 shows the characteristics of those randomly generated test problems:

In all the experiments, the population size of the GA was 200, the probabilities for crossover and mutation were 0.5 and 0.1, respectively, and the termination condition was “no improvement in the best solution for 20 generations”.

In these randomly generated test problems, the VMs’ CPU and memory requirements were randomly generated and the values were both in [300, 3000], and the PMs’ CPU and memory capacities were both randomly picked up from {1000, 1500,  $\dots$ , 55000}. The parameters about the communication network were:  $e_2 = 1$ ;  $e_3 = 3$ ; and  $e_4 = 5$ . The amount of data need to be transferred between each pair of VMs in  $C$  was randomly generated and the value was a whole number between 1 and 9 (units). The parameters about the servers in the data center were:  $k_1 = k_2 = \dots = k_{|P|} = 0.7$ .

For each of the randomly generated test problems, we used the GA to solve it. Considering the stochastic nature of the GA, we repeated the experiments 10

---

**Algorithm 3:** The GA

---

```
1 generate a population of PopSize individuals, Pop;  
2 find the best individual in Pop;  
3 while the termination condition is not true do  
4   for each individual x in Pop do  
5     | calculate its fitness value  $f(x)$ ;  
6   end  
7   for each individual in Pop do  
8     | use the roulette selection to select another individual to pair up;  
9   end  
10  for each pair of parents do  
11    | probabilistically use the biased uniform crossover operator to produce  
12    | an offspring;  
13  end  
14  for each individual in P do  
15    | probabilistically apply the mutation operator the individual;  
16  end  
17  find the best individual in Pop;  
18  if the best individual in Pop is better than the current best individual then  
19    | replace the current best individual with the new best individual;  
20  end  
21 end  
21 decode the best individual and output it.
```

---

**Table 1.** Characteristics of test problems

Test problem	VM (#)	PM (#)
1	100	20
2	200	40
3	300	60
4	400	80
5	500	100

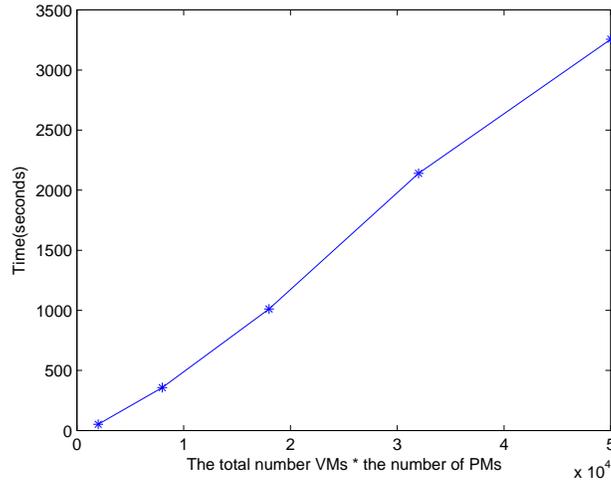
times, and recorded the solutions and computation times. Since it was difficult or impossible to know the optimal solutions to those test problems and therefore to know the quality of the solutions generated by the GA, we implemented an First Fit Decreasing (FFD) algorithm in Java, and used it to solve those test problems. The FFD algorithm is one the most popular heuristic algorithms for bin packing problems. Since VM placement problems can be easily transformed into a bin packing problem, the FFD algorithm is often used to tackle VM placement problems [9]. Since the FFD algorithm is a deterministic one, we only ran it once for each of the test problems. We evaluated the performance of the GA by comparing the quality of the solutions generated by the GA with the quality of the solutions produced by the FFD-based heuristic algorithm. Table 2 shows the experimental results.

**Table 2.** Comparison of the performance of the GA and the performance of the FFD

Test Problem	FFD	GA			
	Energy (watts)	Energy (watts)	SD	Time (seconds)	SD
100	12746.46	10317.73	763.16	51.63	19.40
200	24862.72	22525.48	1322.34	357.58	75.28
300	42035.96	37555.60	1849.23	1011.44	286.42
400	56223.20	51796.05	1620.96	2139.52	507.66
500	70320.00	67912.29	1645.19	3256.46	518.43

It can be seen from the experimental results in Table 2 that the solutions produced by the GA are significantly better than those produced by the FFD. On average the solutions produced by the GA are 3.5%-23.5% better than those produced by the FFD.

In terms of computation time, the FFD took less than 1 millisecond to solve any of the five test problems. The computation time of the GA increased with the number of VMs and the number of PMs. It was observed that the computation time of the GA increased linearly with the product of the number of VMs and the number of PMs. Fig. 3 visualizes the observation. Given that this virtual machine placement problem is a static optimization problem, the computation time and the scalability of the GA are acceptable.



**Fig. 3.** The scalability of our GA

## 5 Conclusion

In this paper we have identified and formulated a new VM placement problem. The new VM placement problem considers not only the energy consumption in those physical servers in a data center, but also the energy consumption in the communication network of the data center. In addition, this paper has proposed a GA for the new VM placement problem. The GA has been implemented and evaluated by experiments. Experimental results have shown that the GA always generates a significantly better solution than the FFD-based algorithm for the VM placement problem.

In this work we used simple energy consumption models to calculate the energy consumptions in the physical servers and the communication network of a data center. However, our GA is independent from those energy consumption models. Thus, in the future we will use more accurate energy consumption models when they are available.

## References

1. Le, K., Zhang, J., Meng, J., Bianchini, R., Jaluria, Y., Nguyen, T.D.: Reducing electricity cost through virtual machine placement in high performance computing clouds. *Proceeding of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 1-12 (2011)
2. Jayasinghe, D., Pu, C., Eilam, T., Steinder, M., Whally, I., Snible, E.: Improving performance and availability of services hosted on IaaS clouds with structural constraint-aware virtual machine placement. *Proceeding of the IEEE International Conference on Services Computing*, 72-79 (2011)
3. Meng, X., Pappas, V., Zhang, L.: Improving the scalability of data center networks with traffic-aware virtual machine placement. *Proceeding of IEEE International Conference on Computer Communications*, 1-9 (2010)
4. Mishra, M., Sahoo, A.: On theory of VM placement: anomalies in existing methodologies and their mitigation using a novel vector based approach. *Proceeding of IEEE International Conference on Cloud Computing*, 275-282 (2011)
5. Goldberg, D.E.: *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley (1989)
6. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*, 28(5), 755-768 (2012)
7. Benson, T., Akella, A., Maltz, D.A.: Network traffic characteristics of data centers in the wild. *Proceedings of the 10th annual conference on Internet measurement*, 267-280 (2010)
8. Mahadevan, P., Sharma, P., Banerjee, S., Ranganathan, P.: Energy aware network operations. *Proceedings of the IEEE international conference on Computer Communications (INFOCOM)*, 25-30 (2009)
9. Xu, J., Fortes, J.A.B.: Multi-objective virtual machine placement in virtualized data center environments. *Proceeding of IEEE/ACM International Conference on Green Computing and Communications*, 179-188 (2010)