Using Genetic Algorithm in Profile-Based Assignment of Applications to Virtual Machines for Greener Data Centers

Meera Vasudevan[1], Yu-Chu Tian[1](✉), Maolin Tang[1], Erhan Kozan[2], and Jing Gao[3]

[1] School of Electrical Engineering and Computer Science, Queensland University of Technology, GPO Box 2434, Brisbane, QLD 4001, Australia
y.tian@qut.edu.au

[2] School of Mathematical Sciences, Queensland University of Technology, GPO Box 2434, Brisbane, QLD 4001, Australia

[3] College of Computer and Information Engineering, Inner Mongolia Agricultural University, 306 Zhaowuda Road, Hohhot 010018, Inner Mongolia, China
gaojing@imau.edu.cn

Abstract. The increase in data center dependent services has made energy optimization of data centers one of the most exigent challenges in today's Information Age. The necessity of green and energy-efficient measures is very high for reducing carbon footprint and exorbitant energy costs. However, inefficient application management of data centers results in high energy consumption and low resource utilization efficiency. Unfortunately, in most cases, deploying an energy-efficient application management solution inevitably degrades the resource utilization efficiency of the data centers. To address this problem, a Penalty-based Genetic Algorithm (GA) is presented in this paper to solve a defined profile-based application assignment problem whilst maintaining a trade-off between the power consumption performance and resource utilization performance. Case studies show that the penalty-based GA is highly scalable and provides 16 % to 32 % better solutions than a greedy algorithm.

Keywords: Data center · Energy efficiency · Application assignment · Resource scheduling · Genetic algorithm · Profiling

1 Introduction

Data centers are facing an escalation of services related to high-powered technologies such as artificial intelligence, IPv6, Remote Direct Memory Access (RDMA), virtualizations and cloud solutions. This in turn predictably increases the energy consumption and operation costs to power and maintain these systems at an alarming pace. A report from the Natural Resources Defense Council (NRDC) indicates that data centers consumed 91 billion kWh of electrical energy in 2013. This statistics is projected to increase by 53 % [1] by year 2020.

The necessity for green and energy-efficient measures has become very real and emerging for reducing carbon footprint and the exorbitant energy costs.

S. Arik et al. (Eds.): ICONIP 2015, Part II, LNCS 9490, pp. 182–189, 2015.
DOI: 10.1007/978-3-319-26535-3_21

Energy and cost distribution studies, e.g., Le *et al.* [2], have demonstrated that deploying green initiatives at data centers reduces the carbon footprint by 35 % at only a 3 % cost increase. However, energy-aware measures with simultaneous maximum performance efficiency and minimum energy consumption [3] are difficult to achieve. In most cases, deploying an energy-efficient solution inevitably degrades the resource utilization efficiency of the data centers.

To tackle this challenging issue, this paper presents a penalty-based genetic algorithm to solve the profile-based application assignment problem. The concepts of profiles and profile-based assignment of applications to Virtual Machines (VMs) have been recently established in our previous work [4]. A greedy algorithm has been proposed in our previous work [4] to solve the profile-based assignment problem. The work of this paper significantly improves our previous work by developing a penalty-based genetic algorithm (GA) for deriving a better solution for reducing energy consumption and increasing resource utilization.

The paper is organized as follows. Section 2 reviews related work and motivates the research. Section 3 describes and formulates the profile-based application assignment problem. The penalty-based genetic algorithm is presented in Sect. 4. Case studies are conducted in Sect. 5 to demonstrate the algorithm. Finally, Sect. 6 concludes the paper.

2 Related Work

Evolutionary algorithms such as genetic algorithms (GA) have been applied for job scheduling in data centers and cloud computing. A GA based task and VM scheduler is presented in a paper [5] for cloud systems with a bi-objective of makespan and average CPU utilization. The paper indicates that a good scheduling algorithm should satisfy both application and resource centric objectives. It proposes a penalty-based GA that satisfies energy, CPU and memory utilization objectives. Also, the problem size considered is significantly large.

An energy-efficient resource allocation method is presented in [6], which uses an open source GA framework called jMetal. The allocation objectives also include optimizing task completion times whilst satisfying computational and networking task requirements. The method ensures scalability and performance efficiency for a large number of tasks. Our approach in the present paper utilizes profiles built for both applications and VMs, allowing the penalty-based GA for very large problem sizes without compromising performance efficiency.

VM placement problems, which are NP-complete, have been solved successfully using GA. In [7], the authors minimize the energy consumption of servers and the communication network within the data centers using GA based VM placement. The work is extended in [8] to significantly improve the energy and performance efficiency with an enhanced hybrid genetic algorithm. Our work in the present paper proposes an energy-efficient penalty-based GA for allocating applications to VMs using a profiling method. The scope of this paper is on the application placement management of data centers.

3 Problem Formulation

The original problem of assigning applications to virtual machines is transformed into a constrained combinatorial optimization as below:

A binary decision variable $x_{ij}, i \in I, j \in J$ represents the assignment of an application a_i, $i \in I$, onto a VM V_j, $j \in J$:

$$x_{ij} = \begin{cases} 1 & \text{if } a_i \text{ is allocated to } V_j;\ i \in I, j \in J, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The CPU utilization of a VM V_j is denoted by $\mu[V_j]$ and is derived from a ratio of the CPU busy time to the time interval ($\lambda = 15\,\text{min}$). The total number of instructions to execute application a_i is given by IC_i.

$$\mu[V_j] = \frac{1}{\lambda} \cdot \sum_{i=1}^{N} \left[\frac{x_{ij} \cdot IC_i}{\mu_j^{CPU}} \right] \tag{2}$$

The total power consumption associated with a data center:

$$P = l[P_{idle} + (E_{usage} - 1)P_{peak} + (P_{peak} - P_{idle})U_{avg}] \tag{3}$$

where P_{peak} and P_{idle} represents the power consumed at the maximum and idle server utilization respectively. The Power Usage Efficiency (PUE) is represented by E_{usage}. U_{avg} is the average CPU utilization of all VMs across the data center for the time interval under consideration. $l \in L$ represents the number of active servers in the data center. The Energy Cost C_{ij} of executing application $a_i, i \in I$, on VM $V_j, j \in J$, is calculated as the product of the ratio of peak and idle power of the host physical machine and the execution time of application a_i on VM V_j:

$$C_{ij} = \frac{P_{peak}}{P_{idle}} \cdot \frac{IC_i}{\mu_j^{CPU}} \tag{4}$$

The constrained combinatorial optimization model for the assignment of a set of applications to VMs is given as:

$$F(obj) = \min \textstyle\sum_{j=1}^{M} \sum_{i=1}^{N} C_{ij} \cdot x_{ij} \tag{5}$$

$$\text{s.t.} \quad IC_j / \lambda \leq \mu_j^{CPU},\ \forall j \in J; \tag{6}$$

$$\textstyle\sum_{i=1}^{N} x_{ij} \eta_i^{mem} \leq \eta_j^{mem},\ \forall j \in J; \tag{7}$$

$$\textstyle\sum_{j=1}^{M} x_{ij} = 1,\ \forall i \in I; \tag{8}$$

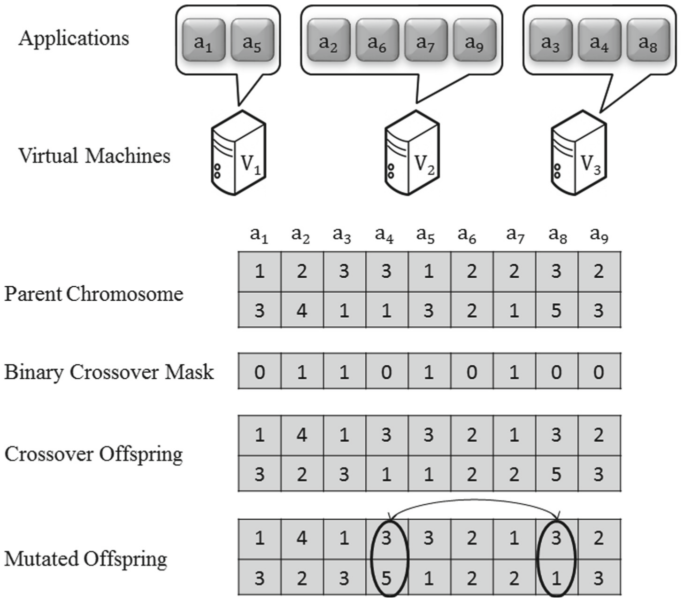
$$x_{ij} = 0 \text{ or } 1,\ \forall i \in I, j \in J. \tag{9}$$

The constraints in Eqs. (6) and (7) ensure that the allocated resources are within the total capacity of the VM. Constraint (8) restricts an application from running on more than one VM. The binary constraint of the allocation decision variable x_{ij} is given by (9).

4 Penalty-Based Genetic Algorithm

Our assignment problem is a combinatorial optimization problem, which is NP-hard. Thus, a steady-state genetic algorithm can be used to solve the problem. This section presents a penalty-based genetic algorithm for the profile-based application assignment problem. The objectives of the penalty-based GA include: minimizing energy consumption; and maximizing resource utilization. Every iteration of the algorithm creates a population consisting of a set of chromosomes representing a possible assignment solution. The initial population consists of chromosomes generated by random allocation of applications to VMs. The following is a description of the genetic operators in the genetic algorithm. Figure 1 represents the working of the genetic operators. The chromosomes are represented by value encoding and parent chromosomes are derived from the roulette wheel selection. Uniform crossover and mutation by selecting and exchanging two genes is applied to the parent solutions to produce the offspring solutions.The termination condition is that cycle is repeated for each generation until a maximum number of generations is reached or an individual is found which adequately solves the problem.

Fitness. The fitness function determines the quality of the solution when compared to an optimal solution. The fitness function effectively penalises an allocation solution that violates the CPU and memory constraints discussed in Eqs. (6) and (7). The lower the energy cost and penalty in terms of resource utilization



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
Parent Chromosome	1	2	3	3	1	2	2	3	2
	3	4	1	1	3	2	1	5	3
Binary Crossover Mask	0	1	1	0	1	0	1	0	0
Crossover Offspring	1	4	1	3	3	2	1	3	2
	3	2	3	1	1	2	2	5	3
Mutated Offspring	1	4	1	3	3	2	1	3	2
	3	2	3	5	1	2	2	1	3

Fig. 1. Value encoding, uniform crossover using binary mask and mutation by selection and exchange of two genes

efficiency, the higher the fitness function. Feasible solutions have a positive fitness value, whereas infeasible solutions incur a negative fitness. The fitness function is derived as:

$$F(X) = w_1 \cdot \bar{F_{obj}} - \frac{w_2}{M} \cdot \sum_{j=1}^{M} \left[\phi_j^{cpu} + \phi_j^{mem}\right] \tag{10}$$

The weights $[w_1, w_2]$ associated with the fitness function is currently set to $[2, 1]$. The multiplicative inverse of the objective function discussed in Eq. (5) is represented by $\bar{F_{obj}}$. In order to normalise and scale the objective function $\bar{F_{obj}}$ to a range of $[1, 10]$, we use:

$$\bar{F_{obj}} = \left[\frac{F_{worst} - F_{obj}}{F_{worst} - F^{\star}}\right] \cdot \left[\frac{F^{\star}}{F_{obj}}\right] \cdot r + 1 \tag{11}$$

Where, the range $r = 9$. The best (minimized) and worst objective function is represented by $F^{\star}$ and F_{worst}, respectively. The penalty for CPU and memory constraint violations are derived as follows:

$$\phi_j^{cpu} = \begin{cases} 0, & \text{if } U_{avg} = 1 \\ \lambda \cdot \mu_j^{cpu}/IC_j, & \text{if } 0 < U_{avg} < 1 \\ 2 & \text{if } U_{avg} = 0 \end{cases} \tag{12}$$

$$\phi_j^{mem} = \begin{cases} 2(1 - 1/\alpha), & \text{if } \alpha \geqslant 1, \\ 2, & \text{otherwise,} \end{cases} \qquad \alpha = \frac{\eta_j^{mem}}{\sum_{i=1}^{N} x_{ij} \cdot \eta_i^{mem}}. \tag{13}$$

5 Case Studies

The profile-based application to VM placement framework targets a big class of data centers with consistent workloads and applications. Profiles are created for every application, physical server and VM from real data center workload logs consisting of CPU, memory and energy utilizations, collected over a period of seven days (the 12th to 19th of May, 2014) to build the profiles. The length of each application is determined by the Instruction Count (IC) and the computing capacity of each VM is in Million Instructions Per Second (MIPS). The application and VM parameter settings are shown below:

IC (instr)	IPS (inst/sec)	Memory (bytes)	P_{peak} (W)	P_{idle} (W)	E_{usage}
$[5, 10] \times 10^9$	$[1, 2] \times 10^9$	$[1000, 5000]$	350	200	2

In our case studies, a data center consisting of upto 2000 VMs is considered. Six different test problem sets are considered where the number of applications ranges from 500 to 5000 with corresponding number of VMs:

The implementation of our profile dependent penalty-based genetic algorithm is carried out with a pre-set population size of 200 individuals in each generation.

Problem	1	2	3	4	5	6
VMs	100	400	800	1200	1600	2000
Applications	500	1000	2000	3000	4000	5000

The termination condition is reached when there is no change in the average and maximum fitness values of strings for 10 generations. The number of maximum generations is set to be 200. The probabilities for crossover and mutation are configured to be 0.75 and 0.02, respectively.

The high scalability of the GA is established by solving the allocation problem for upto 2000 VMs and 5000 applications. Figure 2 displays the algorithm solution time with respect to the increasing problem size. As the test problem size $[M * N]$ increases, the solution time of the GA increases linearly.

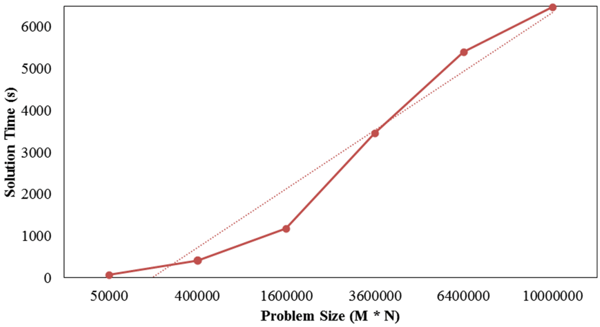


Fig. 2. Scalability of the penalty-based GA.

In order to evaluate the quality of solutions produced by our GA, we solve the test problems using a simple greedy algorithm. The genetic algorithm is stochastic in nature. The quality of allocation solutions in terms of energy consumption are assessed by using GA to solve 30 configurations of each of the test problems as shown in Fig. 3. The resulting mean of energy consumption and solution times is given in Table 1. According to the results, the GA produces 16 % to 32 % better solutions in terms of energy consumption than the greedy algorithm. Although the solution times are higher compared to the greedy approach for the increasing number of applications, the GA maintains an efficient trade-off with energy consumption.

A paired t-test is conducted for the two independent means provided by the GA and greedy algorithm for each of the six test problems. The null hypothesis is that there is no difference between the GA and greedy energy consumption means. The confidence interval is set at 95 % and a two-tailed hypothesis is assumed. The t-stat values are recorded in Table 1 and the p-values are all significantly less than 0.05. The results show that the difference between the means are significant and thus, the null hypothesis is rejected.

As shown in Fig. 4, the average sum of CPU and memory utilization efficiency of the penalty-based GA is 3 % to 22 % more efficient when compared to the greedy approach. Also the variance of the CPU utilization of the GA (0.56) is lower than that of the Greedy algorithm (0.90). This indicates the GA is more consistent in resource allocation.

Table 1. Energy and solution time performance (Energy unit: W; time unit: sec).

Genetic Algorithm				Greedy		T-Test			
Energy	SD	Time	SD	Energy	Time	t-stat	std. err	DF	crit 2-tail
12878.47	1227.90	69	7.53	15017.28	2	−9.54	224.18	29	2.045
21379.27	2404.85	412	38.98	28234.01	6	−15.61	439.06	29	2.045
27113.47	1086.97	1189	50.39	33482.86	9	−32.091	198.45	29	2.045
32001.33	1264.83	3459	341.61	38416.58	18	−27.778	230.92	29	2.045
47149.83	3107.03	5412	276.58	56115.70	27	−15.81	567.26	29	2.045
65904.90	2104.70	6484	250.01	78025.54	40	−31.54	384.26	29	2.045

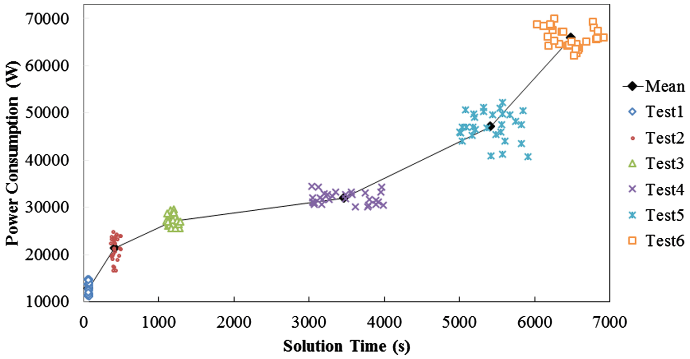


Fig. 3. GA energy consumption and solution time for 30 configurations of each test problem set.

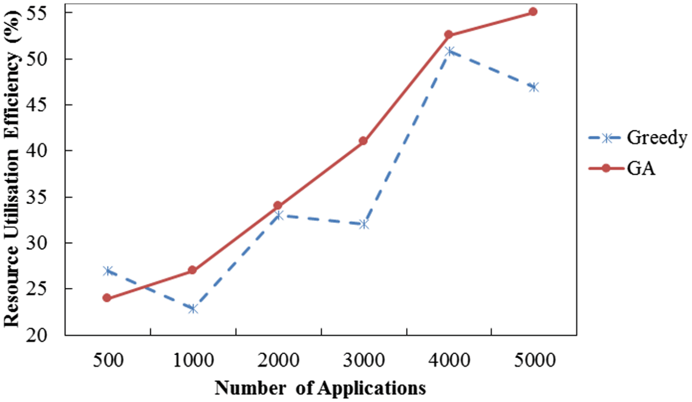


Fig. 4. Resource utilization efficiency.

6 Conclusion

A penalty-based genetic algorithm has been presented for profile-based assignment of applications to VMs. Improving our previous work significantly, it optimizes the energy consumption of data centers while maintaining utilization performance efficiency in terms of CPU and memory. The case studies have demonstrated that the algorithm is highly scalable and provides significantly better solutions than a greedy application placement approach.

Acknowledgments. Author J. Gao would like to acknowledge the support from the National Natural Science Foundation of China under the Grant Number 61462070, and the Inner Mongolia Government under the Science and Technology Plan Grant Number 20130364.

References

1. Whitney, J., Delforge, P.: Scaling Up Energy Efficiency Across the Data Center Industry: Evaluating Key Drivers and Barriers. Issue Paper, Natural Resources Defense Council (NRDC) (2014)
2. Le, K., Bianchini, R., Martonosi, M., Nguyen, T.: Cost- and energy-aware load distribution across data centers. In: Proceedings of HotPower, pp. 1–5 (2009)
3. Greenberg, A., Hamilton, J., Maltz, D.A., Patel, P.: The cost of a cloud: research problems in data center networks. ACM SIGCOMM Comput. Commun. Rev. **39**, 68–73 (2008)
4. Vasudevan, M., Tian, Y.-C., Tang, M., Kozan, E.: Profiling: an application assignment approach for green data centers. In: 40th IEEE Annual Conference of the Industrial Electronics Society, pp. 5400–5406. IEEE Press, Dallas (2014)
5. Sindhu, S., Mukherjee, S.: A genetic algorithm based scheduler for cloud environment. In: 4th International Conference on Computer and Communication Technology (ICCCT), pp. 23–27 (2013)
6. Portaluri, G., Giordano, S., Kliazovich, D., Dorronsoro, B.: A power efficient genetic algorithm for resource allocation in cloud computing data centers. In: 3rd International Conference on Cloud Networking (CloudNet), pp. 58–63. IEEE Press (2014)
7. Wu, G., Tang, M., Tian, Y.-C., Li, W.: Energy-efficient virtual machine placement in data centers by genetic algorithm. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part III. LNCS, vol. 7665, pp. 315–323. Springer, Heidelberg (2012)
8. Tang, M., Pan, S.: A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. Neural Process. Lett. **41**, 211–221 (2015). Springer, US