J Supercomput
DOI 10.1007/s11227-016-1630-1

CEVP: Cross Entropy based Virtual Machine Placement for Energy Optimization in Clouds

Xiaodao Chen[1] · Yunliang Chen[1] · Albert Y. Zomaya[2] · Rajiv Ranjan[3] · Shiyan Hu[4]

Abstract Big data trends have recently brought unrivalled opportunities to the cloud systems. Numerous virtual machines (VMs) have been widely deployed to enable the on-demand provisioning and pay-as-you-go services for customers. Due to the large complexity of the current cloud systems, promising VM placement algorithm are highly desirable. This paper focuses on the energy efficiency and thermal stability issues of the cloud systems. A Cross Entropy based VM Placement (CEVP) algorithm is proposed to simultaneously minimize the energy cost, total thermal cost and the number of hot spots in the data center. Simulation results indicate that the proposed CEVP algorithm can (1) achieve energy savings of 26.2 % on average, (2) efficiently reduce the temperature cost by up to 6.8 % and (3) significantly decrease the total number of the hot spots by 60.1 % on average in the cloud systems, by comparing to the Ant Colony System-based algorithm.

Keywords Cloud computing · VM placement · Optimization

✉ Yunliang Chen
cyl_king@hotmail.com

1 School of Computer Science, China University of Geosciences, Wuhan 430074, People's Republic of China

2 School of Information Technologies, The University of Sydney, Sydney, Australia

3 Newcastle University, Newcastle, UK

4 Department of Electrical and Computer Engineering, Houghton, MI 49931, USA

Published online: 06 February 2016

1 Introduction

The increasing complexity of data processing has perplexed various computing infrastructures where cloud computing has recently emerged as a promising candidate to tackle this big data challenge [1–3]. In cloud infrastructure, numerous virtual machines (VMs) have been widely deployed to enable the on-demand provisioning and pay-as-you-go services for customers [4–9]. The placement of VM has evolved as a critical problem for cloud providers. The main technical difficulty is the unbalanced usage of resources and the unnecessary activation of physical servers resulting from excessive VM requests from many different users. The problem becomes more challenging since modern cloud data centers typically adopt geographically distributed architecture due to the large complexity of the current cloud systems which consist of many interconnected data centers [10–13]. In this context, low-quality VM placement could lead to the large energy consumption of data centers [14–17].

This is why many existing VM placement algorithms have been designed to optimize various objectives including minimizing the power consumption, balancing the load among physical servers, maximizing resource utilization, and minimizing peak temperature. A traffic aware VM placement method is proposed to minimize the communication cost in [18], and an Ant Colony-based VM placement approach is proposed to minimize energy consumption and reduce computing resource waste in [19]. In [20], a genetic algorithm with a fuzzy logic controller cost function is designed for the VM placement problem. In [21], a genetic algorithm and a sequential quadratic programming are proposed, which considers the thermal-aware VM placement to minimize the peak temperature in the data center.

The energy efficiency and thermal stability issues have become critical in the cloud data center. The need to reduce power consumption and enhance the thermal profile is pressing. Dynamic Voltage Frequency Scaling (DVFS) is a popular power management technique to reduce the processor power and improve thermal stability. Through dynamically scaling down the voltage and frequency of processors during idle or non-intensive computation period, the energy consumption of the processor can be diminished and the thermal profile of the processor can be improved. The DVFS technique has been successfully applied in cloud computing communities [22–25]. In fact, the thermal profile is a particularly important issue in the data center due to the reason that the cooling cost of a data center is very significant [15]. For the thermal profile, there are two critical thermal parameters which are often considered in the literature, namely, the total thermal cost and the number of the hot spot. The total thermal cost is the summation of temperature of all blocks. The number of the hot spot is the number of blocks which has thermal values greater than a threshold. Both of them are important, since large total thermal cost means a cooling cost [26] while the excessive hot spot increases the possibility of transient faults [27]. Therefore, this work proposes Cross Entropy based VM Placement (CEVP) algorithm to minimize the energy cost and improve on-chip thermal profile simultaneously considering DVFS. Our contributions are as follows.

– We propose the Cross Entropy based VM Placement (CEVP) algorithm to simultaneously minimize the energy cost, total thermal cost and the number of hot spots in

the data center. With the power of cross entropy method, our VM placement algorithm generates the solution with both energy cost and thermal profile optimized simultaneously.
- Our CEVP algorithm incorporates Dynamic Voltage Frequency Scaling (DVFS) for minimizing the energy cost as well as the thermal cost.
- Our algorithm reduces the energy cost by average of 26.7 %, from the baseline VM placement algorithm based on ant colony system optimization.
- The total thermal cost is reduced by 3.8 % on average and the number of hot spots is decreased significantly by 58.0 %.

The remainder of the paper is organized as follows. Section 2 presents the preliminaries of the VM, the power model and the thermal model. Section 3 presents the mathematical problem formulation. Section 4 describes the proposed Cross Entropy based VM Placement (CEVP) algorithm. Our experimental results are presented in Sect. 5. Section 6 summarizes the paper.

2 Preliminary

In this section, the virtual cloud environment is introduced. After that, the energy model and thermal model are presented.

2.1 Virtual cloud environment

The virtual cloud environment is the cloud computing infrastructure provided by web service providers. Users of cloud services access the computing resources via conventional computer networking interfaces, such as Ethernet. The computing tasks of the users are carried out in the data center, each hosting a computer cluster with multiple physical machines [9]. Each physical machine hosts a number of virtual machines (VMs), under direct access and control of the end user. Each VM runs its own Operating System (OS) with a number of applications of the end users [28]. The hypervisor is responsible for allocating computing resources (CPU, memory, etc.) of the physical machine to each VM hosted by it. This architecture is created for the ease of on-demand creation, administration privileges and performance efficiency [29]. An example of the virtual cloud environment has been depicted in Fig. 1. VMs are deployed on physical computing nodes, and each of them consumes a certain amount of hardware resources. A typical DVFS-characterized cloud computing environment has multiple voltage frequencies of each computing nodes. A VM is able to be assigned to specify computing node with which the voltage frequencies can also be specified [30].

2.2 Energy consumption model

The energy consumption model in the cloud environment is based on the model of complementary metal-oxide semiconductor (CMOS) devices [31]. The total power consumption is the summation of the dynamic power and the static power. The dynamic power can be calculated as:

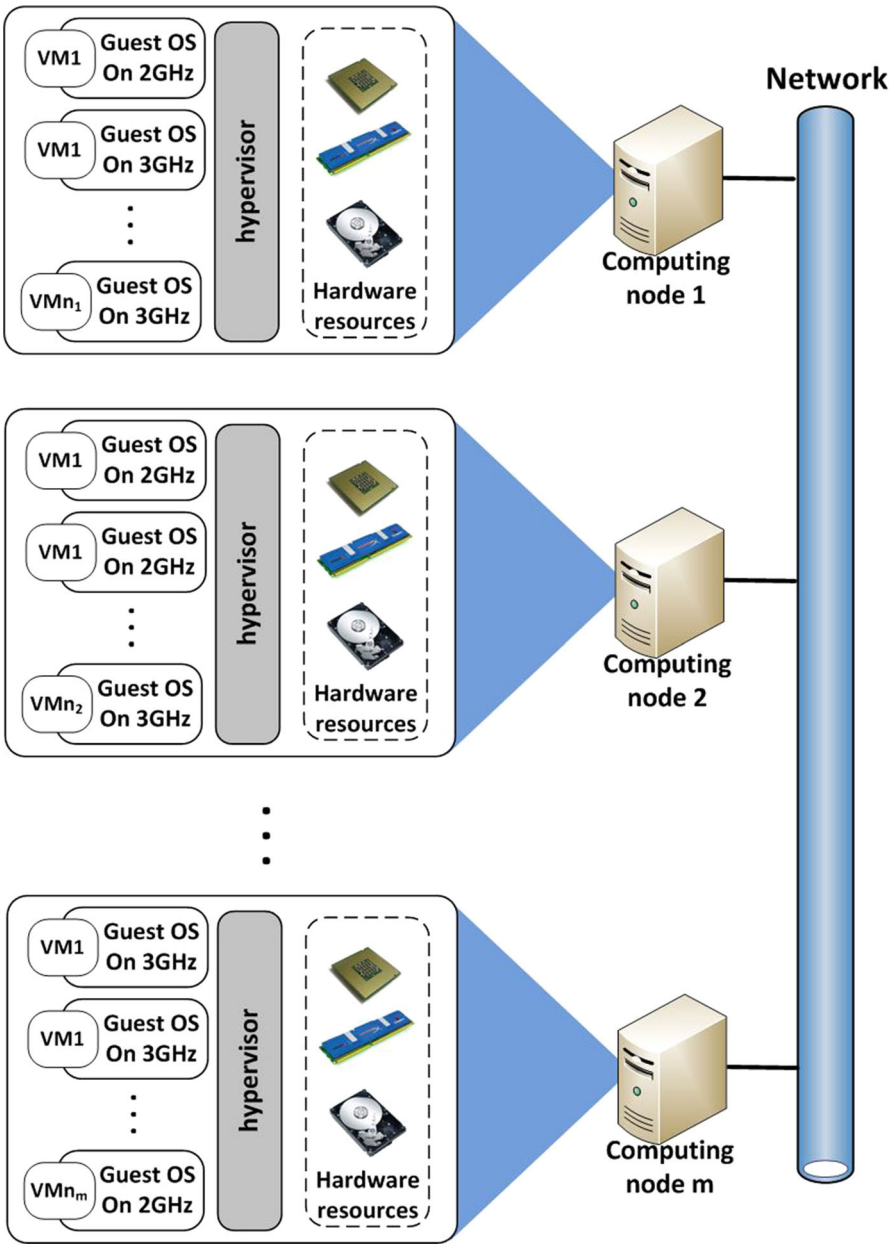


Fig. 1 A virtual cloud environment illustration

$$P_d = C \cdot {V_{dd}}^2 f \tag{1}$$

where C denotes the value of load capacitance, V_{dd} denotes the supply voltage, and f denotes the operating frequency. Because of the voltage scaling technique, the supply

voltage V_{dd} can be approximated proportional to the frequency f [32]. Therefore, the dynamic power consumption of a computing node can be estimated by a convex function $P_d = Cf^3$. The static power consumption are mainly caused by leakage current and the reverse bias junction current from the power supply [33]. Following the reference [33], the static power is calculated as:

$$P_s = V_{dd} I_{leakage} + |V_{bias}| I_{bias} \tag{2}$$

where $I_{leakage}$ denotes the value of the subthreshold leakage current, V_{bias} denotes the value of the body bias voltage, and I_{bias} denotes the value of the reverse bias junction current. At a certain technology node, $I_{leakage}$ is a function of supply power, meanwhile I_{bias} and V_{bias} are technology constants. As a result, for a certain technology node, the static power consumption P_s is also a function of the supply voltage V_{dd}. To consider the total energy consumption, both dynamic power and static power are included. As in the model proposed in [34], for a computing node j with a number of VMs, the node energy consumption can be estimated by

$$E_j = \sum_{i=1}^{N} \frac{c_i}{f_j} \cdot (P_d + P_s) \tag{3}$$

where N denotes the total number of VMs on the node j, c_i denotes execution cycles of task i, and f_j denotes clock frequency of the node j. In case of no VMs running on a computing node, the computing node can be turned off or halted to save power.

2.3 Thermal model

The thermal management plays a crucial role in the overall performance of the data center. The VM scheduler targeting at reducing thermal dissipation cost significantly reduces the operational cost. Thermal Profile is considered in the proposed VM placement approach. Since the temperature of each computing node is dependent on the thermal profile of the entire data center, this work is focused on the chip level thermal states. The thermal dissipation is evaluated by two metrics.

- Total on-chip thermal cost, the summation of temperature of each cores. The total thermal cost is proportional to the cooling cost [26].
- Number of hot spot, the number of blocks whose thermal values exceed δ, the user-defined threshold. In our implementation, δ is set to be $340k$. Hot spots jeopardize the system reliability with transient faults [27].

The target of the proposed VM placement algorithm is to minimize the total thermal cost and the number of hot spots.

In the implementation, the widely used RC thermal model [15,35,36] is used to simulate the temperature profile of the data center. According to the model, the tem-

perature of each computing node is determined by the power consumption, physical dimension and location of the computing nodes. For completeness, the RC model is outlined as follows.

Let r_{ij} denote the thermal resistance between the computing node i and j. Thus, $r_{ij} = \frac{\Delta T_{ij}}{\Delta P_j}$, in which ΔT_{ij} is the temperature difference between node i and j. P_j is the power consumption of node j, and ϕ is the total number of computing nodes. As shown in [35], the temperature of each computing node can be calculated by Eq. 4.

$$\begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_\phi \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1\phi} \\ r_{21} & r_{22} & \cdots & r_{2\phi} \\ \vdots & \vdots & \ddots & \vdots \\ r_{\phi 1} & r_{\phi 2} & \cdots & r_{\phi\phi} \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_\phi \end{bmatrix}, \tag{4}$$

in which T_i and P_i are the temperature and power consumption of computing node i and j, respectively.

The proposed VM placement algorithm uses Monte Carlo method to evaluate each VM placement solution sample. Due to the significant number of samples, the straightforward implementation of the RC model incurs prohibitively high computing overhead. To resolve this, the temperature is estimated as the first-order estimation of the thermal profile [37]. Mathematically, $T = \frac{\delta}{k} Pd$, in which δ denotes the thickness of the processor of the computing node, k denotes the thermal conductivity of the material, A denotes the area of the processor and Pd denotes the power density.

With the first-order estimation, the temperature of the computing node is linearly proportional to its power density. In other words, the temperature of the computing node is determined by its own power consumption, as well as power consumption profile of the neighboring nodes. Figure 2 shows an example illustrating the correlation between the temperature and the power consumption profile of neighboring nodes. It is generated by a thermal modeling tool HotSpot [38]. Figure 2a illustrates that nodes consuming more power have higher temperature. Figure 2b demonstrates that even with the same energy cost, different VM placement solutions have different peak temperatures, due to different power consumption profiles.

3 Problem formulation

This study mainly focuses on the VM placement problem over computing nodes of the cloud center. The target of this work is to deploy VMs to specific nodes with specific core ID, meanwhile the voltages level of the computing node is also defined, such that the total energy of the computing nodes are minimized and the total heat diffusion is maximized. The mathematical formulation of the problem can be described as:

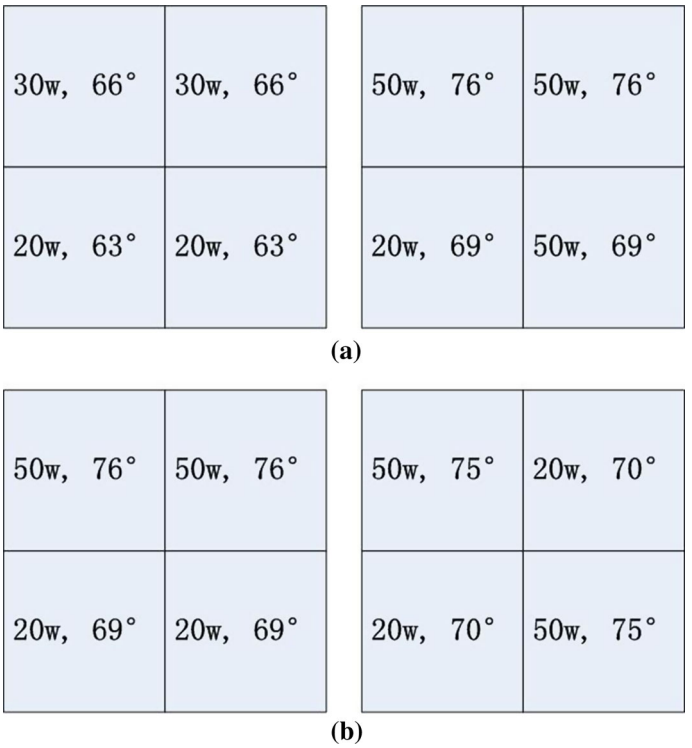


Fig. 2 A toy example illustrating the impact of power consumption profile on thermal profile. **a** Nodes consuming more power have higher temperature. **b** Higher level of heat diffusion introduces lower peak temperature

$$
\begin{aligned}
&\text{minimize } E_{total}\\
&\text{maximize } H_{total}\\
&\text{subject to } E_{total} = \sum E_{n_i}, \quad n_i \in N\\
&\qquad\qquad E_{n_i} = \sum E_{v_j}, \quad v_j \in V_{n_i}\\
&\qquad\qquad H_{total} = \sum H_{n_i,n_j}, \quad \forall v_i, v_j \in \Phi\\
&\qquad\qquad H_{n_i,n_j} = \frac{|Pd_{n_i} - Pd_{n_j}|}{dist_{ij}}\\
&\qquad\qquad f_{n_i} \geq \dot{f}_{n_i} \forall n_i \in N\\
&\qquad\qquad hw_{n_i} \geq \dot{hw}_{n_i} \forall n_i \in N
\end{aligned}
$$

where E_{total} denotes the total energy consumption, E_{n_i} denotes the energy consumption of the computing node n_i and N denotes the set of all computing nodes; E_{v_j} denotes the energy consumption of the VM v_j running on the computing node n_i; H_{total} denotes the total heat diffusion, H_{v_i,v_j} denotes the heat diffusion in between the VM i and the VM j and Φ denotes the set of every touching computing nodes pair; Pd_{n_i} and Pd_{n_j} denote the power density of the computing node n_i and n_j, respectively; f_{n_i} denotes the voltage frequency of the computing node n_i, $\dot{f}_{n_i}$ is the minimum required voltage frequency of the computing node n_i and N is the set of computing nodes; hw_{n_i} denotes the hardware resources of the computing node n_i and

$\overline{hw}_{n_i}$ denotes the maximum hardware resource. In this formulation, the total energy consumption is calculated through summing up the energy consumption of all running computing nodes. The VM placement and the voltage frequency selection, which have to be with respect to hardware resource constraints and VM voltage frequency constraints, decide the energy consumption of computing nodes. Meanwhile, these also impact the power density, as well as the heat diffusion.

The above formulation is a multi-object optimization problem with a large solution space. In this case, the traditional linear programming method cannot be applied efficiently due to the large cost of computation. The following paragraph presents a Cross Entropy based VM Placement (CEVP) algorithm which is able to search a good quality solution in a large solution space for the proposed problem.

4 Cross Entropy based VM Placement (CEVP) algorithm

A Cross Entropy based VM Placement (CEVP) algorithm is proposed in this section. It first introduces the theoretical foundation for the cross entropy technique in Sect. 4.1. After that, the proposed CEVP algorithm is presented with details in Sect. 4.2.

4.1 The theoretical foundation for cross entropy

The Cross-Entropy method is the stochastic optimization framework based on importance sampling, which is first proposed in [39]. For completeness, the algorithm is briefly introduced here. Refer to [39,40] for further details.

Cross Entropy can be used to optimize a given objective function $S(X)$ over all $x \in X$. In general, there are two major steps in each iteration of the cross-entropy method.

- Generate a set of random data samples according to the given distribution
- Update the parameters of the distribution, given the set of elite samples

Suppose that our target is to optimize function $S(X)$ over the set $x \in X$. Denote the probability density function (PDF) by $f_x(x; v)$, which is parameterized by a finite-dimensional real vector v. The other version of the optimization problem is to estimate the probability that $S(x) > \gamma$. When $S(x) > \gamma$ becomes a rare event, γ is the max value in $S(x)$. Mathematically [39],

$$\ell = P(S(X) \geq \gamma) = E_u I_{\{S(X)\geq\gamma\}} = \int I_{\{S(X)\geq\gamma\}} f_x(x; v) \tag{5}$$

Since we are interested in the case where ℓ is a rare-event probability which means it should be very small, we introduced the other PDF denoted by $g_x(x)$, such that $g_x(x) = 0 \Rightarrow I_(\{S(X) \geq \gamma\}) f(x; v) = 0$ for all x. Using the PDF $g_x(x)$ we can represent ℓ as follows [39].

$$\ell = \int \frac{I_{\{S(X)\geq\gamma\}} f_x(x; v)}{g_x(x_i)} g_x(x_i) \mathrm{d}x \tag{6}$$

Therefore, since $X_1, \ldots, X_N$ are independent random vectors with PDF $g_x(x_i)$, the following is obtained [39],

$$\ell^* = \frac{1}{N}\Sigma_{i=1}^{N} I_{\{S(X)\geq\gamma\}}\frac{f_x(x_i;u)}{g_x(x_i)} \tag{7}$$

where ℓ^* is an unbiased estimator of ℓ, which is called importance sampling estimator. We need to find such a $g_x(x_i)$ to minimize the variance ℓ^*, which is the density of X conditional on the event $S(X) > \gamma$; that is [39],

$$g^* = \frac{I_{\{S(X)\geq\gamma\}} f_x(x;v)}{\ell} \tag{8}$$

Which means that we need to make sure that the Kullback–Leibler divergence between g and g^* is the minimal of the following equation [39].

$$\begin{aligned}\mathbb{D}(g^*, g) &= \int g^*(x) ln \frac{g^*(x)}{g(x)} \mathrm{d}x \\ &= \int g^*(x) lng^*(x)\mathrm{d}x - \int g^*(x) lng(x)\mathrm{d}x\end{aligned} \tag{9}$$

The problem is to choose a v such that $-\int g^*(x) lng(x)\mathrm{d}x$ is minimized, which is actually minimizing $-\int g^*(x) lnf(x;v)\mathrm{d}x$, equivalent to the following equation [39].

$$\begin{aligned}&max_v \int g^*(x) lnf(x;v)\mathrm{d}x \\ &= max_v \frac{I_{\{S(X)\geq\gamma\}} f_x(x;v)}{\ell} lnf(x;v)\mathrm{d}x\end{aligned} \tag{10}$$

According to [39], optimal v^* can be estimated as follows [39].

$$\frac{1}{N}\Sigma_{i=1}^{N} I_{\{S(X)\geq\gamma\}}\frac{f_x(x_i;u)}{f_x(x_i;w)} lnf(x;v) \tag{11}$$

4.2 CEVP algorithmic flow

The philosophy behind cross entropy optimization method has been briefly illustrated in Fig. 3, which generates samples according to the PDF iteratively. Samples are evaluated by mapping solution vectors to the solution space. Note that, in Fig. 3, θ denotes the solution space and $f(\theta)$ denotes the objective function. At the end of each iteration, elite samples are selected according to values of $f(\theta)$. Based on these elite samples, the PDF can be improved by updating the characterizing parameter v. In the new iteration, the new PDF will be utilized to generate new samples. This procedure is executed iteratively, and in each iteration, the lower bound of the object function τ is adaptively improved. In this work, the optimal VM placement solution can be approached iteratively. Viewing from the high level, the VM placement can be approached iteratively in a cross entropy optimization scheme. It first initializes the PDF of each computing node placement, and generates VM placement samples according to the PDF of each computing node. Note that the sample here denotes

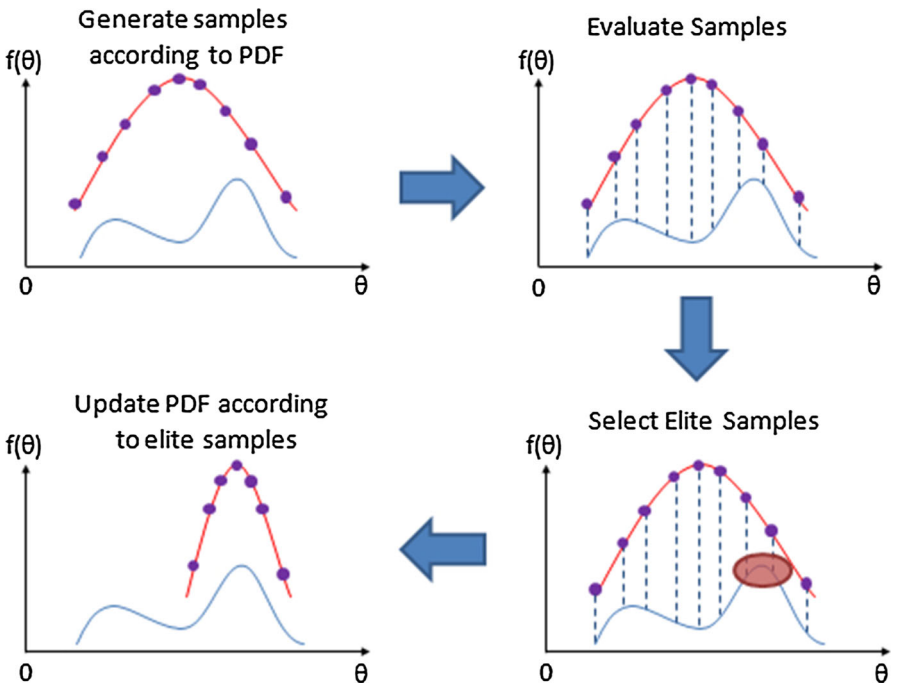


Fig. 3 Cross entropy optimization scheme

the VM status of its running frequency and identification. After that, the samples are evaluated in terms of the energy consumption and the profile thermal. A certain number of elite samples are selected for to update the PDF for next iteration.

For the practical VM placement problem, the number of VM placement solutions can be significant. As a result, it is inefficient to enumerate all possibilities. By the cross-entropy optimization framework, this work proposes a Cross Entropy based VM Placement (CEVP) algorithm to determine the VM placement with both lower energy cost and low temperature cost. The CEVP algorithm is presented in Algorithm 1. The initializing procedures perform when the algorithm starts. PDF characterizing parameters $\overrightarrow{\mu^{(1)}}$ and $\overrightarrow{\sigma^{(1)}}$ for possible VM placement are initialized. Note that, in this algorithm, $\mu_i^{(1)}$ implies the possibility of generating solution samples with certain patterns; that is, assign every VM to a specific core with a specific frequency level. Therefore, for each VM, the PDF is in a 3-D space, where the z-axis stands for the probability of the assignment, x-axis and y-axis stands for the core ID and the frequency ID, respectively. According to the initial parameter values, L samples can be generated. These samples denote the candidate VM placement solution which might have the global minimum energy cost and heat diffusion cost. Note that, in practice, a cost function which sums up the energy cost and the heat diffusion cost with weighted factors is employed to evaluated samples. After generating samples, the top $\hat{M}$ elite samples can be selected according to the cost function. Using elite samples, the PDF characterizing parameters $\overrightarrow{\mu^{(1)}}$ and $\overrightarrow{\sigma^{(1)}}$ can be updated for the next iteration.

Algorithm 1: The Cross Entropy based VM Placement (CEVP) algorithm

1: Initialize parameters of mean and variation with Gaussian distributions for all VMs sources: $\overrightarrow{\mu^{(1)}} = \{\mu_1^{(1)}, \mu_2^{(1)}, \mu_3^{(1)}, ..., \mu_N^{(1)}\}$ and $\overrightarrow{\sigma^{(1)}} = \{\sigma_1^{(1)}, \sigma_2^{(1)}, \sigma_3^{(1)}, ..., \sigma_N^{(1)}\}$, where $\mu_1^{(i)}$ is the mean value,
σ is the standard deviation and N denotes the total number of VMs. Initialize the threshold τ.
Set $k = 1$.
2: Repeat steps 3 to 9 till the pre-defined converge criteria of solutions being satisfied.
3: Generate L solution samples with respect to the PDF parameters $(\overrightarrow{\mu^{(t)}}, \overrightarrow{\sigma^{(t)}})$
4: Evaluate M solutions $S = (s_1, s_2, s_3, ..., s_M)$ by computing their thermal and heat diffusion cost function.
5: Select top $\hat{M}$ elite samples $\hat{S} = (\hat{s}_1, \hat{s}_2, \hat{s}_3, ..., \hat{s}_{\hat{M}})$ according to the threshold τ of cost function values.
6: Update PDF parameters $\overrightarrow{\mu^{(t)}}, \overrightarrow{\sigma^{(t)}}$: $\mu_i^{(t+1)} = \mu_i^t + \frac{\sum_{j \in I}(s_j - \mu_i^t)}{\hat{M}_i}$ and $\sigma^{(t+1)} = \sqrt{\frac{\sum_{i \in I}(s_i - \mu_i^{(t+1)})^2}{\hat{M}_i}}$
$\forall i \in [1, N]$, where I and $\hat{M}_i$ denote the set of and the number of solutions corresponding to $\mu_i^{(t+1)}$ respectively.
7: Update the threshold value τ.
8: t = t + 1;
9: Check the converge criteria with updated $\overrightarrow{\mu^{(t)}}, \overrightarrow{\sigma^{(t)}}$.

In the next iteration, a new set of solution samples are generated correspondingly. The above proceeds iteratively, until the pre-defined stop criteria are satisfied. The stop criterion for the proposed CEVP algorithm is constructed as follows. When the best VM placement solution samples in the current iteration, which are close to that in the previous iteration fair enough, the CEVP algorithm stops. Note that this CEVP algorithm is easily parallelized in the parallel programming environment. The reason is that, for each solution sample, the evaluation process of this solution sample are independent from other solution samples.

5 Experimental results

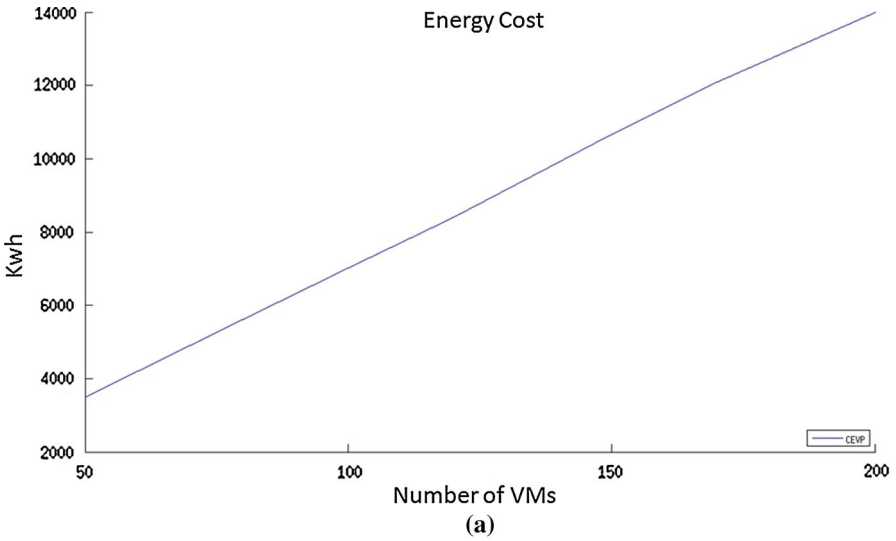
The experiments are carried out on a simulated system. The proposed Cross Entropy based VM Placement (CEVP) algorithm is implemented in *C++* and tested on a dell workstation with Intel *i*7 3.07 GHz CPU and 24G memory. Our experiments are performed over a set of computer nodes with five different voltage frequency levels ranging from 1.8 Ghz to 2.5 Ghz. Each of the computer node has four codes and the floor planning of them are generated by following [41]. There are 500 test cases evaluated in the experiment to avoid transient anomalies. In each test case, there are 50 to 200 VMs randomly generated with a minimum require voltage frequency-level value, meanwhile same number of host are available as candidate hosts. To evaluate the proposed algorithm, we compare the proposed CEVP algorithm that considers multi-objects with an Ant Colony System algorithm which is adapted from [19] in terms of energy consumption, thermal cost and hot spot numbers. The thermal values are obtained using the thermal modeling tool HotSpot [38] (Table 1).

We have following observations:

Table 1 Comparisons of energy consumption, thermal cost, number of hot spots and runtime between the proposed CEVP algorithm and the ant colony system algorithm

Task set size	Ant colony system				LP based algorithm				Improvement		
	Energy	T. cost	Hotpot #	CPU (s)	Energy	T. cost	Hotpot #	CPU (s)	Energy (%)	T. cost (%)	Hot spot # (%)
50–100	7397.9	28,122.4	18.3	0.2	5352.8	26, 334.2	11.8	6.3	27.6	6.8	55.5
101–200	14,488.2	54,208.6	37.1	0.4	10,612.8	52,221.3	23.7	8.9	26.7	3.7	56.3
201–300	23,953.2	88,954.6	62.1	1.1	17,625.7	86,736.4	38.8	13.3	26.4	2.5	59.9
301–400	33,410.4	123,701.0	87.07	2.1	24,654.9	121,272.1	54.4	17.6	26.2	2.0	60.1

– The proposed CEVP algorithm saves the energy consumption cost up to 27.6 % from the ant colony algorithm. The average energy saving of the proposed CEVP algorithm is 26.7 % over the ant colony algorithm. The effectiveness of the proposed CEVP on energy saving can be demonstrated.
– The proposed approach can improve the temperature cost up to 6.3 % by comparing to the Ant Colony System algorithm. It does not improve as much as the energy savings due to the heat sinks of the computing cores.



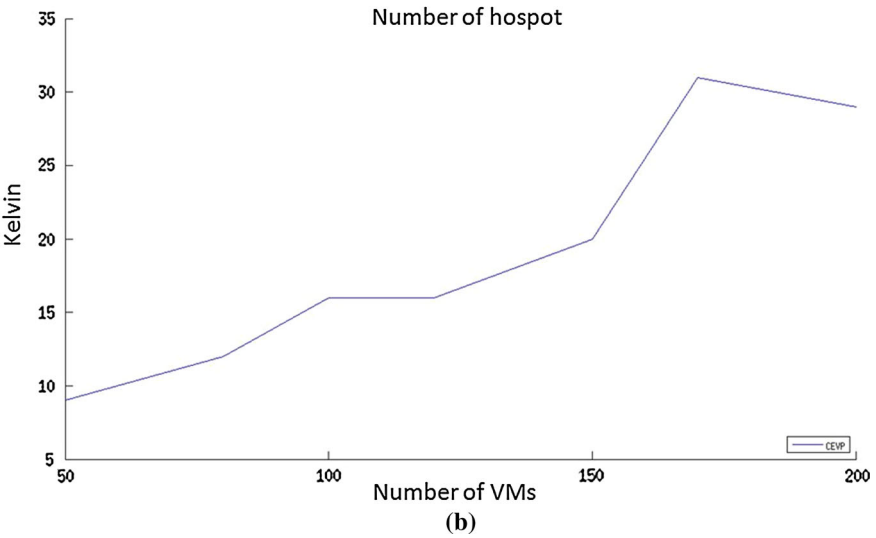


Fig. 4 Case study. **a** The curve of the energy consumption with increasing the workloads of VMs. **b** The curve of number of hot spots with increasing the workloads of VMs

– The total number of hot spots are largely improved by our heat diffusion term in the objective function by comparing to the Ant Colony System algorithm. The proposed CEVP algorithm reduces the number of hot spots 58 % on average. It is a big help to reduce the transient faults of the cloud system.
– Although the running time of our algorithm is greater than the one of the Ant Colony System algorithm, the proposed algorithm always runs within 55 s. The proposed algorithm almost linearly increases with the problem size which is quite efficient even with hundreds of VMs.

We also perform the case study that proportionally increases the workload on VMs with 50 available computing machines. The number of VMs ranges from 50 to 200. Different thermal cost and numbers of hot spots are obtained. The results are shown in Fig. 4. Following observations have been made.

– The energy consumption increases almost linearly. This is expected since the total number of tasks for the set of VMs is linearly proportional to the number of VMs in that set. In other words, the amount of tasks processed by each VM, across different cases, is almost constant. Consequent, the energy cost of each VM is approximately constant. As the number of VMs increase, the energy cost increases linearly.
– The number of hot spots does not increase at a constant rate with the number of VMs. The reason is the energy consumption plays a more significant role than the heat diffusion in data center management. When it is difficult to optimize against both energy consumption and heat diffusion, our algorithm favors optimizing the former over the latter. Therefore, our algorithm could generate solutions with the minimum energy cost, at the cost of more hot spots.

6 Conclusion

In this work, a Cross Entropy based VM Placement (CEVP) algorithm is proposed. It considered both the energy cost and thermal profile of the computing nodes in the cloud environment. The simulations have been conducted using based 400 test cases, which demonstrate the effectiveness of the proposed approach. By comparing with a Ant Colony System algorithm, the proposed CEVP algorithm significantly saves the energy cost by 26.7 % on average and reduces the thermal cost up to 6.8 %. Moreover, the proposed CEVP algorithm can largely reduce the number of hot spots by 58 % on average. This shows the effectiveness of our algorithm.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (Nos. 61501411 and 61440018) and the China Postdoctoral Science Foundation (2014M552112).

References

1. Wang L, Chen D, Hu Y, Ma Y, Wang J (2013) Towards enabling cyberinfrastructure as a service in clouds. Comput Electr Eng 39(1):3–14
2. Wang L, Kunze M, Tao J, von Laszewski G (2011) Towards building a cloud for scientific applications. Adv Eng Softw 42(9):714–722

3. Wang L, von Laszewski G, Younge AJ, He X, Kunze M, Tao J, Cheng F (2010) Cloud computing: a perspective study. New Gener Comput 28(2):137–146
4. Wang L, Chen D, Huang F (2011) Virtual workflow system for distributed collaborative scientific applications on grids. Comput Electr Eng 37(3):300–310
5. Wang L, Chen D, Zhao J, Tao J (2012) Resource management of distributed virtual machines. IJAHUC 10(2):96–111
6. Wang L, von Laszewski G, Chen D, Tao J, Kunze M (2010) Provide virtual machine information for grid computing. IEEE Trans Syst Man Cybern Part A 40(6):1362–1374
7. Wang L, von Laszewski G, Kunze M, Tao J, Dayal J (2010) Provide virtual distributed environments for grid computing on demand. Adv Eng Softw 41(2):213–219
8. Wang L, von Laszewski G, Tao J, Kunze M (2010) Virtual data system on distributed virtual machines in computational grids. IJAHUC 6(4):194–204
9. Zhang X, Yue Q, He Z (2014) Dynamic energy-efficient virtual machine placement optimization for virtualized clouds. In: Proceedings of the 2013 international conference on electrical and information technologies for rail transportation (EITRT2013)-volume II. Springer, Berlin, pp 439–448
10. Bilal K, Khan SU, Zhang L, Li H, Hayat K, Madani SA, Min-Allah N, Wang L, Chen D, Iqbal MI, Xu C-Z, Zomaya AY (2013) Quantitative comparisons of the state-of-the-art data center architectures. Concurr Comput Pract Exp 25(12):1771–1783
11. Teyeb H, Balma A, Ben Hadj-Alouane N, Tata S (2014) Optimal virtual machine placement in large-scale cloud systems. In: Proceedings of the 2014 IEEE 7th international conference on cloud computing (CLOUD), pp 424–431. IEEE
12. Zhang W, Wang L, Liu D, Song W, Ma Y, Liu P, Chen D (2013) Towards building a multi-datacenter infrastructure for massive remote sensing image processing. Concurr Comput Pract Exp 25(12):1798–1812
13. Zhang W, Wang L, Ma Y, Liu D (2014) Design and implementation of task scheduling strategies for massive remote sensing data processing across multiple data centers. Softw Pract Exp 44(7):873–886
14. Dong X, El-Gorashi T, Elmirghani JMH (2011) Green IP over WDM networks with data centers. J Lightwave Technol 29(12):1861–1880
15. Wang L, Khan SU, Dayal J (2012) Thermal aware workload placement with task-temperature profiles in a data center. J Supercomput 61(3):780–803
16. Wang L, Khan SU (2013) Review of performance metrics for green data centers: a taxonomy study. J Supercomput 63(3):639–656
17. Wang L, von Laszewski G, Huang F, Dayal J, Frulani T, Fox G (2011) Task scheduling with ann-based temperature prediction in a data center: a simulation-based study. Eng Comput 27(4):381–391
18. Meng X, Pappas V, Zhang L (2010) Improving the scalability of data center networks with traffic-aware virtual machine placement. In: Proceedings of the 2010 IEEE INFOCOM, pp 1–9
19. 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 Comput Syst Sci 79(8):1230–1242
20. Xu J, Fortes JAB (2010) Multi-objective virtual machine placement in virtualized data center environments. In: Proceedings of 2010 IEEE/ACM international conference on cyber, physical and social computing (CPSCom), pp 179–188. IEEE
21. Tang Q, Gupta SKS, Varsamopoulos G (2008) Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach. IEEE Trans Parallel Distrib Syst 19(11):1458–1472
22. Bodas D (2003) Data center power management and benefits to modular computing. In: Intel developer forum
23. Feng W-C, Ching A, Hsu C-H (2007) Green supercomputing in a desktop box. In: Proceedings of the 2007 IEEE international parallel and distributed processing symposium, 2007. IPDPS, pp 1–8. IEEE
24. von Laszewski G, Wang L, Younge AJ, He X (2009) Power-aware scheduling of virtual machines in dvfs-enabled clusters. In: Proceedings of the 2009 IEEE international conference on cluster computing, August 31–September 4, 2009, New Orleans, Louisiana, USA, pp 1–10
25. Wang L, von Laszewski G, Dayal J, Wang F (2010) Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS. In: 10th IEEE/ACM international conference on cluster, cloud and grid computing, (CCGrid) 2010, 17–20 May 2010, Melbourne, Victoria, Australia, pp 368–377
26. Sawyer R (2004) Calculating total power requirements for data centers. In: White Paper, American Power Conversion

27. Govindavajhala S, Appel AW (2003) Using memory errors to attack a virtual machine. In: Proceedings of 2003 symposium on security and privacy, pp 154–165. IEEE
28. Smith J, Nair R (2005) Virtual machines: versatile platforms for systems and processes. Elsevier, Amsterdam
29. Von Laszewski G, Wang L, Younge AJ, He X (2009) Power-aware scheduling of virtual machines in dvfs-enabled clusters. In: Proceedings of the 2009 IEEE international conference on cluster computing and workshops, pp 1–10. IEEE
30. vsphere 5.1 documentation. https://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vsphere.resmgmt.doc%2FGUID-F40F901D-C1A7-43E2-90AF-E6F98C960E4B.html
31. Kolodziej J, Khan SU, Xhafa F (2011) Genetic algorithms for energy-aware scheduling in computational grids. In: Proceedings of the 2011 international conference on P2P, parallel, grid, cloud and internet computing (3PGCIC), pp 17–24. IEEE
32. Weste NHE, Eshraghian K (1985) Principles of cmos vlsi design: a systems perspective. NASA STI/Recon Tech Rep A 85:47028
33. Martin SM, Flautner K, Mudge T, Blaauw D (2002) Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads. In: Proceedings of the 2002 IEEE/ACM international conference on computer-aided design, pp 721–725. ACM
34. Wei T, Chen X, Hu S (2011) Reliability-driven energy-efficient task scheduling for multiprocessor real-time systems. IEEE Trans Comput-Aided Des Integr Circuits Syst 30(10):1569–1573
35. Cong J, Zhang Y (2006) Thermal-aware physical design flow for 3-d ics. In: Proceedings of IEEE international VLSI multilevel interconnection conference. Citeseer, pp 73–80
36. Hung W-L, Xie Y, Vijaykrishnan N, Addo-Quaye C, Theocharides T, Irwin MJ (2005) Thermal-aware floorplanning using genetic algorithms. In: Proceedings of the 2005 international symposium on quality of electronic design, pp 634–639. IEEE
37. Han Y, Koren I (2007) Simulated annealing based temperature aware floorplanning. J Low Power Electron. 3(2):141–155
38. Huang W, Sankaranarayanan K, Skadron K, Ribando RJ, Stan MR (2008) Accurate, pre-RTL temperature-aware design using a parameterized, geometric thermal model. IEEE Trans Comput 57(9):1277–1288
39. Rubinstein RY, Kroese DP (2004) The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning. Springer, Berlin
40. De Boer P-T, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. Ann Oper Res 134(1):19–67
41. Sharon H, Thidapat C, Yun X, Robert D (2013) Enhancing multicore reliability through wear compensation in online assignment and scheduling. In: Proceedings of the 2013 design, automation and test in Europe conference and exhibition (DATE), pp 1373–1378