Energy-aware Virtual Machine Placement in Data Centers

Daochao Huang, Dong Yang, Hongke Zhang
School of Electronic and Engineering
Beijing Jiaotong University, Beijing, China
Email:08111043, dyang, hkzhang@bjtu.edu.cn

Lei Wu
IBM China Development Laboratory, Beijing, China
Email: wuleicdl@cn.ibm.com

***Abstract*—The energy efficiency of modern data centers has become a practical concern and has attracted significant attention in recent years. In contract to existing solutions that primarily focuses on only one specific aspect of management to reduce energy consumption, this paper explores the balance between server energy consumption and network energy consumption to present an energy-aware joint virtual machine (VM) placement. Given the definition of VM placement fairness, the basic algorithm of VM placement which fulfills server energy consumption constraints is conducted. Then, we further formulate the VM placement as an optimization problem which considers application dependencies to reduce network energy consumption. We design a joint algorithm that efficiently solves the VM placement problem for very large problem sizes. Using simulations, we conduct a comparative analysis on the impact of the data center architectures, server constraints and application dependencies on the potential performance gain of energy-aware VM placement. Compared to existing generic methods, we show a significant performance improvement such as efficiently reducing the number of physical machines used to save server energy consumption, decreasing the communication distance between VMs to obtain data center network energy consumption efficiency, improving scalability of data centers.**

I. Introduction

Modern Virtualized Data Center is increasingly a hosting platform for a wide spectrum of composite applications–such as search engines, social networks, video streaming, medical services, electronic commerce, grid computing and network-based cloud computing. As the trend towards more communication intensive applications in data centers and increasing size of data center will only continue, the energy consumption of network and server resources that underpin will grow. On the one hand, with more communication intensive applications deployed in data centers, the bandwidth usage between virtual machines (VMs) is rapidly growing. This has drawn extensive attention from academia with respect to network energy consumption. On the other hand, with cooling energy demand and energy costs increasing, the server power consumption management has to be considered. Therefore, the engineering challenges and cost of managing the power consumption of large data centers and associated cooling drive the need to reduce data center energy use.

Most current approaches (e.g. [1][2][3][4][5][6]) to reduce energy consumptions primarily focuses on only one specific aspect of management, such as minimizing network consumption, balancing thermal distribution, or maximizing resource usage. For example, [1] provides a Traffic-aware VM Placement Problem (TVMPP) model that is able to place VMs in data centers with large number of data exchanges and high network bandwidth usage, greatly reducing the bandwidth consumption within the data center, thus saving network energy consumption and improving data center scalability. Several other common VM placement algorithms include Constraint Programming [2], Bin Packing Stochastic [3], Integer Programming [4], Genetic Algorithm [5][6]. These methods can make the maximum effective use of the data center server resources, and reduce energy consumption by shutting down servers that are not working or hibernating servers running low workloads. While data center energy use has received much attention recently, there has been less attention paid to consider both server energy consumption and the energy consumption of the data center transmission and switching network. With many practical applications, minimizing the total energy consumption in a data center requires the formulation of a joint optimization problem.

In this paper, we explore the balance between server energy consumption and data center network energy consumption to present a joint VM placement. The algorithm takes into account both constraints of servers and the application dependencies among VMs of composite applications. Our target is to, meeting the conditions of the server-side constraints, minimize network data transmission, reduce energy consumption in data centers. To this end, we design a joint algorithm that efficiently solves the VM placement problem for very large problem sizes. Our simulation results show that the algorithm can efficiently adjust the allocation of resources and significantly reduce the transmission of data center traffic.

The rest of the paper is organized as follows. Section II describes a formalization of the proposed VM placement problem; While Section III derives the utility function meeting server-side constraints and the application-aware communication cost function expression; in the fourth section, the experimental results of the analysis under different data center network architectures are presented based on the proposed VM placement algorithm; Section V summarizes this paper.

II. PROBLEM FORMULATION

Formally, we consider a data center network with dependency graph $G(V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set

978-1-4673-0921-9/12/$31.00 ©2012 IEEE

of VMs and E is the set of edges defined as $E = (v_i, v_j) : v_i, v_j \in V$. Two VMs v_i and v_j are dependent with each other if any communication takes places between them. Let $P = \{p_1, p_2, ..., p_m\}$ denote set of physical machines. Then, the set of VMs hosted by physical machine $p \in P$ can be denoted by $V(p)$. Let $W(v_i, v_j)$ denote traffic demand for each edge which is directly proportional to the traffic transferred between v_i and v_j. At time t, the real traffic is given by $W_{ij}(t)$. Further, let $Load_v$ be the vector of load requirements of VM v, such as the vector of CPU, memory and storage. Let $Capacity_p$ denote the available server-side capacity of physical machine p. The aggregate resource demands of VMs located at physical machine p at time t are defined as $Load_p(t)$. The resource demands of VM v hosted by physical machine p at time t are denoted as $Load_{pv}(t)$. Next, let $Distance(p_k, p_l)$ be the number of switches on the communication path from physical machine p_k to p_l. Given that the indicator function of VM placement is defined as:

$$I_{ik} = \begin{cases} 1 & If\ VM\ v_i\ is\ assigned\ to\ PM\ p_k \\ 0 & otherwise \end{cases} \quad (1)$$

From the above, $\sum_{k}^{|P|} I_{ik} = 1$ is satisfied, in other words, one VM v_i must be located at a physical machine p_k. Similarly, let $I_{ik}^{jl} = I_{ik} \times I_{jl}$, $I_{ik}^{jl} = 1$ denote the situation where VM v_i is assigned to p_k and v_j is assigned to p_l. Therefore, $I_{ik} + I_{jl} \leq 1 + I_{ik}^{jl}$.

III. ENERGY-AWARE VM PLACEMENT

In this section, we present an energy-aware VM placement model that enables data centers to systematically place VMs on condition that both server energy efficiency and network energy savings are achieved. We consider both server capacity constraints and application multi-tier inherent dependencies in this model.

We provide modeling details in the following subsections.

A. Server-side constraints and VM placement

Let $y = (y_p, p \in P)$ denote a VM placement approach satisfying proportional fairness [7][8] assuming that (a) resources allocation policy is feasible, in other words, resources (such as CPU, physical memory, storage space, etc.) allocated to each VM is less than the total capacity of the hosting physical machine. (b) for any other alternative resource allocation approach $\tilde{y} = (\tilde{y}_p, p \in P)$,the following condition is satisfied: $\sum_{p:p\in P} w_p \frac{\tilde{y}_p - y_p}{y_p} \leq 0$, where w_p is the weight of server performance or its willingness to offer resources. As we can see, proportional fairness depends on the difference between two placement approaches, which means that both fairness and efficiency are integrated in one model. Let $U_p(\cdot)$ denote the utility function of placement approach (denoted by $y_p(t)$) for physical machine p at time t, for all $p \in P$, the overall utilities can be denoted as

$$\sum\nolimits_{p\in P} U_p(y_p(t)) = \sum\nolimits_{p\in P} w_p \log y_p(t) \quad (2)$$

Then, the server-side objective function is given by maximizing the aggregate utilities of all physical machines in the data center.

In general, multiple VMs can reside on the same host and each VM occupies part of resources. Let $Load_{pv}(t)$ denote the load of VM v which resides on PM $p \in P$, and $Load_p(t)$ denote the aggravate loads of PM $p \in P$, the following constraints must be satisfied: $Load_p(t) = \sum_{v:v\in V(p)} Load_{pv}(t)$. Here $Load(t)$ is defined as the dimensional vector of load requirements of VM, when CPU, memory and storage are considered, $d = 3$, $Load(t)$ is given by $Load(t) = \left(Load^{CPU}(t), Load^{memory}(t), Load^{storage}(t)\right)$. Further, we define as the available server-side capacity on PM $p \in P$ regarding its CPU, memory, storage etc. To ensure that the total load on any physical machine is less than or equal to its capacity, the following must hold:

$$\sum_{v:v\in V(p)} Load_{pv}(t) \leq Capacity_p \quad (3)$$

We study the problem of placing VMs on a set of physical hosts in this paper. Typically, tightly packing VMs onto a small number of servers and turning off other servers is an effective way to maximize machine utilization and reduce server energy consumption. However, concentrating workloads on a subset of the system resources can cause heat imbalances and create hot spots, which may impact cooling costs and degrade server life and performance. An effective strategy should consider tradeoffs energy efficiency and fairness. To reflect these two considerations, in our analysis, proportional fairness model is utilized to obtain the server-side VM placement $(\mathbf{P}_1)$:

$$\begin{aligned} (\mathbf{P}_1): \max \quad & \sum_{p:p\in P} U_p(Load_p(t)) \\ \text{Subject to} \quad & \sum_{v:v\in V(p)} Load_{pv}(t) = Load_p(t), \forall p \in P \\ & \sum_{v:v\in V(p)} Load_{pv}(t) \leq Capacity_p, \forall p \in P \\ \text{Over} \quad & Load_{pv}(t) \geq 0, p \in P, v \in V \end{aligned} \quad (4)$$

In order to facilitate the subsequent derivation of the formula, let $y_p(t) = Load_p(t)$. To maximizing the overall aggregate utilities of data center and obtain the optimal solution of $(\mathbf{P}_1)$, a Lagrange function is defined as:

$$\begin{aligned} & L(x, y; \lambda, \eta) = \\ & \sum_{p:p\in P} \left\{ U_p(y_p(t)) + \lambda_p \left(\sum_{v:v\in V(p)} x_{pv}(t) - y_p(t) \right) \right\} \\ & + \sum_{p:p\in P} \eta_p \left(Capacity_p - \sum_{v:v\in V(p)} x_{pv}(t) - \varepsilon_p^2 \right) \end{aligned} \quad (5)$$

Where both $\lambda = (\lambda_p, p \in P)$ and $\eta = (\eta_p, p \in P)$ are Lagrange multiplier vectors$\varepsilon^2 = \left(\varepsilon_p^2, p \in P\right)$ is the relaxation factor vector. Let λ_{pv} denote the load requirements of VM v. Let η_p be the available capacity of physical machine p. Let the total resources occupied by all the VMs residing on physical machine p be denoted by $\sum_{v:v\in V(p)} x_{pv}(t)$. $\varepsilon_p^2 \geq 0$

will indicates there are remaining resources exist on physical machine p.

Theorem 1: The basic datacenter VM model $(\mathbf{P}_1)$ is a convex programming problem, the optimal resources distribution solution for each VM v, i.e., $x_{pv}(t)$, exists, but not unique, while the total load requirements of physical machines, i.e., $y_p(t)$, have unique solution.

The proof of this theorem can be found in Appendix A.

Based on the VM placement model $(\mathbf{P}_1)$, the derivation yields: $\partial L(x,y;\lambda,\eta)/\partial y_p(t)=0$, then

$$y_p(t)=w_p/\lambda_p \tag{6}$$

such that:

$$\begin{aligned}&\hat{L}(x;\lambda,\eta)=\\&\sum_{p:p\in P}\left\{w_p\log\left(\tfrac{w_p}{\lambda_p}\right)-w_p+\sum_{v:v\in V(p)}x_{pv}(t)\lambda_p\right\}\\&+\sum_{p:p\in P}\eta_p\left(Capacity_p-\sum_{v:v\in V(p)}x_{pv}(t)-\varepsilon_p^2\right)\end{aligned} \tag{7}$$

Let $\partial\hat{L}(x;\lambda,\eta)\Big/\partial\lambda_p=0$, then $\lambda_p=w_p\Big/\sum_{v:v\in V(p)}x_{pv}(t)$. Simplifying (7) with respect to λ_p, we get:

$$\begin{aligned}&\hat{L}(x;\eta)=\sum_{p:p\in P}\left\{w_p\log\left(\sum_{v:v\in V(p)}x_{pv}(t)\right)\right\}\\&+\sum_{p:p\in P}\eta_p\left(Capacity_p-\sum_{v:v\in V(p)}x_{pv}(t)-\varepsilon_p^2\right)\end{aligned} \tag{8}$$

Let $\partial\hat{L}(x;\eta)/\partial x_{pv}(t)=0$, it is driven that $y_p(t)=\sum_{v:v\in V(p)}x_{pv}(t)=w_p/\eta_p$. Compared to (6), we get $\lambda_p=\eta_p$. It is concluded that, when $\lambda_p=\eta_p$, i.e., the load requirements of VMs are just equal to the availability capacity of physical machines, the optimal unique solution of $(\mathbf{P}_1)$ is obtained.

B. Application-aware VM Placement

Let $Cost(\cdot)$ denote the communication costs function of placement approach (denoted by $y_{p_k}(t)$) for physical machine p_k at time t, for all $p_k\in P$, the total communication costs of physical machines are defined as $\sum_{k=1}^{m}Cost(y_{p_k}(t))$. In practice, $Cost(\cdot)$ can be defined as $Cost(y_{p_k}(t))=w_{p_k}y_{p_k}(t)$, where w_{p_k} is the price of communication. Let the set of application dependencies for VM v_i be denoted by $D(v_i)$, the aggregate communication traffic for physical machine p_k is achieved, i.e., $y_{p_k}(t)=\sum_{\forall v_i\in V(p_k)}\sum_{\forall v_j\in D(v_i)}Distance(p_k,p_l)\times W_{ij}(t)\times I_{ik}^{jl}$. From network traffic perspective, to ensure that the total bandwidth requirement on any physical server is less than or equal to its capacity, the following must hold: $\sum_{i}^{|V|}\sum_{j,j\neq i}^{|V|}W_{ij}(t)\times I_{ik}\le Bandwidth_k$, where the $Bandwidth_k$ is the bandwidth capacity of PM p_k. Then, the application-aware objective function can be given by minimizing the aggregate communication costs among VMs deployed in the data center. Eventually, application-aware VM placement $(\mathbf{P}_2)$ is defined as:

$$\begin{aligned}&(\mathbf{P}_2):\min\quad\sum_{k=1}^{m}Cost(y_{p_k}(t))\\&\text{Subject to } y_{p_k}(t)=\textstyle\sum_{\forall v_i\in V(p_k)}\sum_{\forall v_j\in D(v_i)}Distance(p_k,p_l)\\&\qquad\times W_{ij}(t)\times I_{ik}^{jl}\\&\qquad\sum_{i}^{|V|}\sum_{j,j\neq i}^{|V|}W_{ij}(t)\times I_{ik}\le Bandwidth_k\\&\text{Over}\qquad W_{ij}(t)\ge 0,i,j=1,...,n\end{aligned} \tag{9}$$

To facilitate the subsequent description, let $x_k(t)=\sum_{i}^{|V|}\sum_{j,j\neq i}^{|V|}W_{ij}(t)\times I_{ik}$.

Theorem 2: The convex programming problem $(\mathbf{P}_2)$ has unique optimal solution R^*, i.e.if $R=\{\mathbf{x}|x_k(t)\le Bandwidth_k;x_k(t)\ge 0,k=1,2,...,m\}$, then $R^*\neq\emptyset$.

The proof of this theorem can be found in Appendix B. Assume that

$$\begin{aligned}\varphi(\mathbf{x},\boldsymbol{\mu})&=\sum_{k=1}^{m}Cost(y_{p_k}(t))\\&\quad-\sum_{k=1}^{m}\mu_k\left(Bandwidth_k-\sum_{i}^{|V|}\sum_{j,j\neq i}^{|V|}W_{ij}(t)\times I_{ik}\right)\\&=\sum_{k=1}^{m}Cost(y_{p_k}(t))-\sum_{k=1}^{m}\mu_k(Bandwidth_k-x_k(t))\end{aligned} \tag{10}$$

According to Karush-Kuhn-Tucker KKT conditions [9][10] and (10), it is derived that:

$$\frac{\varphi(\bar{\mathbf{x}},\bar{\boldsymbol{\mu}})}{\partial x_k(t)}=Cost_x(y_{p_k}(t))+\bar{\mu}_k\ge 0,x_k(t)\ge 0 \tag{11}$$

$$\frac{\varphi(\bar{\mathbf{x}},\bar{\boldsymbol{\mu}})}{\partial \mu_k}=-(Bandwidth_k-x_k(t))\le 0,\bar{\mu}_k\ge 0 \tag{12}$$

$$\frac{\varphi(\bar{\mathbf{x}},\bar{\boldsymbol{\mu}})}{\partial x_k(t)}x_k(t)=(Cost_x(y_{p_k}(t))+\bar{\mu}_k)x_k(t)=0 \tag{13}$$

$$\bar{\mu}_k\frac{\varphi(\bar{\mathbf{x}},\bar{\boldsymbol{\mu}})}{\partial \mu_k}=-\bar{\mu}_k(Bandwidth_k-x_k(t))=0 \tag{14}$$

Where $\boldsymbol{\mu}=\{\mu_k,k=1,...,m\}$ denotes KKT multiplier vector.

C. Energy-aware Joint VM Placement

From the above and in view of $(\mathbf{P}_1)$ and $(\mathbf{P}_2)$, an energy-aware joint VM algorithm is given. On the one hand, to save server-side energy consumption, maximizing the utilization of physical machines (i.e. minimizing the total resource wastage) in data centers is one of the placement objective functions. On the one hand, to obtain data center network energy consumption efficiency, application-aware VM placement is provided to minimize traffic burden among tightly coupled applications. These two objectives may be conflict when considered together. As a result, an effective strategy considering tradeoffs between these objectives is needed.

Let $U_{p,\max}$ denote the maximum utility of PM $p \in P$, the definition of resource wastage can be given by $U_{p,\max} - U_p$. Therefore, the bigger the differences of utility are, the more resources are wasted. Then, the energy-aware joint VM placement is posed as a multi-objective optimization problem of simultaneously minimizing total resource wastage and communication costs:

$$\begin{aligned}(\mathbf{P}_3): \min & \sum_{k=1}^{m} \left(U_{p_k,\max} - U_{p_k}\left(Load_{p_k}(t)\right)\right) \\ & \text{and } \min \sum_{k=1}^{m} Cost\left(y_{p_k}(t)\right) \\ \text{Subject to } & \sum_{v:v\in V(p_k)} Load_{p_k v}(t) \le Capacity_{p_k} \\ & \sum_{i}^{|V|} \sum_{j,j\neq i}^{|V|} W_{ij}(t) \times I_{ik} \le Bandwidth_k\end{aligned} \tag{15}$$

The proposed joint VM placement attempts to achieve two (possibly) conflicting objectives. To solve this problem, Fuzzy logic [15] is employed to provide an efficient way of combining conflicting objectives and expert knowledge. For multiple objectives, fuzzy logic allows the mapping of values of different objectives into linguistic values characterizing levels of satisfaction. The linguistic values are mapped into the interval [0 1] by the membership function associated with each objective. Therefore, the solutions of the multi-objective function given by $(\mathbf{P}_3)$ are evaluated using the following proposed fuzzy-logic system.

Consider our joint VM placement problem described above. Two linguistic variables - resource wastage (r), traffic cost (c) - are introduced and one linguistic value and a corresponding fuzzy set are defined for each variable, namely, fuzzy sets small resource wastage and low traffic cost. Membership functions of these fuzzy sets are decreasing functions of the variable values, since the smaller the value, the higher is the degree of satisfaction. The search algorithm seeks to find solutions that are nearest to each individual goal. Hence, the evaluation of a solution can be expressed by the following fuzzy rule:

IF VM placement solution $y_p(t)$ has small resource wastage (sr) AND low traffic cost (lc) , THEN $y_p(t)$ is a good solution.

With respect to some VM placement solution $y_p(t)$ in the fuzzy set of good solutions, its membership value is given by $\psi\left(y_p(t)\right)$. Using the ordered weighted-averaging fuzzy operator proposed by Yager [16], the above fuzzy rule evaluates to the following, $\psi\left(y_p(t)\right) = \theta \min\left(\psi_r\left(y_p(t)\right), \psi_c\left(y_p(t)\right)\right) + (1-\theta)\, avg\left(\psi_r\left(y_p(t)\right), \psi_c\left(y_p(t)\right)\right)$(θ is set to 0.5 in the simulation experiments described in Section IV), in which $\psi_r\left(y_p(t)\right)$ and $\psi_c\left(y_p(t)\right)$ represent the membership degree of solution $y_p(t)$ in the fuzzy sets defined by sr ans lc, respectively. Given that the most desirable solution is the one with the highest membership in the fuzzy sets $\{sr, lc\}$, it is driven that, the solution with the highest $\psi\left(y_p(t)\right)$ is the one that best meets all the goals and is reported as the best VM placement solution.

The membership functions for the fuzzy set $\psi\left(y_p(t)\right)$ are linear decreasing functions. To determine the lower and upper bounds of the membership functions, assuming that $Load(t) = \left(Load^{CPU}(t), Load^{memory}(t)\right)$ represent the CPU and memory requirements of all VMs. The ideal minimum number of servers needed to host all the VMs is given by $n_{\min} = \max\left(\frac{\sum_{p\in P}\sum_{v\in V(p)} Load_{pv}^{CPU}(t)}{\sum_{p\in P} capacity_p^{CPU}}, \frac{\sum_{p\in P}\sum_{v\in V(p)} Load_{pv}^{memory}(t)}{\sum_{p\in P} capacity_p^{memory}}\right)$. The maximum number of servers for hosting VMs is $n_{\max} = \min(m, n)$. Let R_{lower} and R_{upper} denote the lower and upper bounds of resource wastage respectively, it is derived that, $R_{lower} = \sum_{i=1}^{n_{\min}} w_{p_i} \log\left(capacity_{p_i}/Load_{p_i}(t)\right)$, $R_{upper} = \sum_{i=1}^{n_{\max}} w_{p_i} \log\left(capacity_{p_i}/Load_{p_i}(t)\right)$. Let $\overline{L}$ be the average communication distance between two servers communicating with each other ($\overline{L}$ is different under different data center network architecture and can be deduced from [1]). The lower and upper bounds of communication costs are $C_{lower} = \overline{L}\cdot \sum_{i=1}^{n_{\min}} |D(v_i) = 1|$ and $C_{upper} = \overline{L}\cdot \sum_{i=1}^{n_{\max}} |D(v_i) = 1|$, where C is the total communication costs.

D. Implementation

A prototype implementation approach is proposed in this section for the joint VM placement. Corresponding to server-side constraints and application-aware attributes, two types of resource mapping are involved in virtualized data center resource management – the mapping from application workloads to resource requirements and the mapping from virtual resources to physical resources. A two-level control architecture (see Fig.1) naturally supports these two mappings through local controllers at the virtual-container level and a global controller at the resource-pool level. A local controller implemented in every virtual machine is responsible for determining the amount of resources needed by an application to guarantee its performance. A global controller at the data center level determines VM placement and resource allocation. Besides, the monitoring system of the data center measures system information including resource usage of each server and communication traffic between coupled applications and collects them into a centralized profiling repository. The profiling and modeling components utilize the system measurements to create models of resource wastage and communication costs, which are used by the global controller to optimize its placement decisions.

Suppose there are n VMs with application dependency matrix D, and the size of each VM is represented as a d-dimensional vector in which each dimension corresponds to one type of the resources (e.g. CPU, memory and storage) that is requested by applications packaged to run on this VM. Resources on physical servers are allocated as "slots" along multiple dimensions according to the resource demands of VM requests. The first step of solving the proposed joint VM placement problem is intuitively equivalent to finding a map-

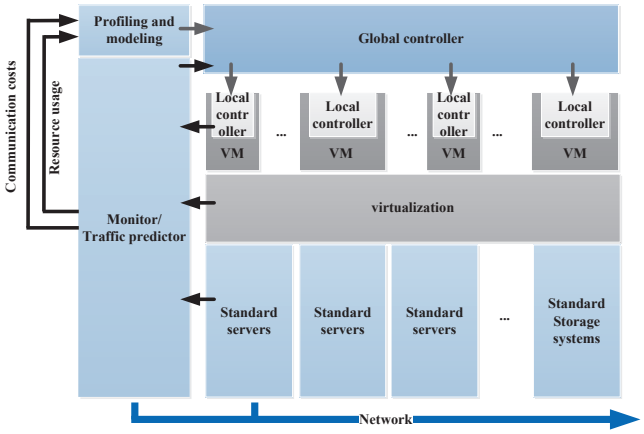

Fig. 1. Two-level control architecture and information flow for the proposed joint VM placement

Algorithm 1 **VMGrouping**

INPUTS: VMs $V=\{v_1,v_2,...,v_n\}$, Physical machines $P=\{p_1,p_2,...,p_m\}$, Communication distance $L=\{Distance(p_k,p_l)\}$, Traffic weights $W=\{W(v_i,v_j),i,j=1,...,n\}$, PM capacity $Capapcity=\{Capacity_{p_k},p_k\in P\}$, Application dependencies matrix $D=\{D(v_i),i=1,2,...,n\}$

OUTPUT: best VM placement solution

$n \leftarrow$ *size of* D {Find out VM count}
$m \leftarrow$ *size of* L {Find out PM count}
Initial the value of ξ {Set the tolerance of deviation from desired solution}
$\psi(y_p(t)) \leftarrow 0$ {Set the initial value of membership function }
VMKCut(D , k **)**{Partition n VMs into k groups $\{g_i\}$,where k is the number of physical machine clusters. }
SlotGrouping($\{g_i\}$,V , P,$Capacity$,k**)**{Partition slots into k groups $\{s_i\}$ }
Assign g_i to s_i , $\forall i=1,...,n$ {One-to-one mapping between VM group and slot group}
Update $\psi(y_p(t))$ {Update membership value of VM placement solution $y_p(t)$ }
for each VM group i=1 to k **do**
  if $|1-\psi(y_p(t))|>\xi$ {The best VM placement solution has not been found.}
    VMGrouping(L^{s_i},W^{g_i}, D^{g_i} **)**{ L^{s_i}:communication distance for s_i , W^{g_i}:traffic weights for g_i, D^{g_i}: Application dependencies matrix for g_i, recursively call VMGrouping }
  end if
  if $|1-\psi(y_p(t))|\le\xi$ **then**
    Best solution found for g_i; **continue**
  end if
end for

Algorithm 2 **VMKCut**

INPUTS: Graph weight matrix G, number of physical machine clusters k

OUTPUTS: VM groups $\{g_i\}$

$n \leftarrow$ *size of* G **for** each slot group $i=1$ to k **do**
Compute a Gomory-Hu tree for G and obtain $n-1$ cuts $\{b_j\}$ {These $n-1$ cuts contain the minimum weight cuts for all VM pairs}
Sort $\{b_j\}$ by increasing weight
for each VM group $i=1$ to k **do**
  $g_i \leftarrow \varnothing$
  Find the minimum j such that removing $\{b_1,...,b_j\}$ will partition G into two components: g_i and $G \leftarrow G \setminus g_i$, $n=n-|g_i|$
end for
return $\{g_i\}$

Algorithm 3 **SlotGrouping**

INPUTS: VMs $V=\{v_1,v_2,...,v_n\}$, Physical machines $P=\{p_1,p_2,...,p_m\}$, PM capacity $Capapcity=\{Capacity_{p_k},p_k\in P\}$, VM groups $\{g_i\}$

OUTPUTS: slot groups $\{s_i\}$

for each slot group $i=1$ to k **do**
  $s_i \leftarrow \varnothing$ {initial slot group}
  for each PM k=1to m **do**
    $V(p_k) \leftarrow \varnothing$; {Initial VMs assigned to PM p_k }
    for each VM v_j in group g_i
      if $Load(v_j) \le Capacity_{p_k}$ **then**
        $V(p_k) \leftarrow V(p_k) \cup \{v_j\}$; Update capacity of p_k
        $s_i \leftarrow s_i \cup \{slot_{v_j}\}$ {Update slot group}
        $g_i \leftarrow g_i \setminus \{v_j\}$ {Update VM group}
      else
        $P \leftarrow P \setminus \{p_k\}$ {Update PM set}; **break**
      end if
    end for
    if $|g_i|>1$ {Multiple VMs in g_i } **then continue**
    end if
  end for
end for
return $\{s_i\}$

ping of VMs to slots such that VM pairs with heavy mutual traffic are assigned to slot pairs with low-cost connections. By assigning these n VMs to slots, we obtain a graph G with the n VMs as nodes and D as edge weights and a Balanced Minimum K-cut Problem [17][18] is provided to find a k-cut with minimum weight whose removal partitions G into k disjoint subsets. Therefore, these n VMs are partitioned into k groups of different size. The pseudo-codes for the algorithm are described in Algorithm 1 and Algorithm 2. The second step of our joint VM placement is to map each VM group to appropriate physical machines in the closest clusters under server-side constraints. The pseudo-code of this procedure is captured in Algorithm 3.

IV. Simulations

Through the problem formulation, we can notice that the communication distance, weight matrices and θ are the three determining factors for optimizing the VM placement. Consequently, we focus on the following questions: 1) What is the performance of our mechanisms under varying traffic demand and network architectures? 2) How well do the mechanisms minimize the total energy consumption in a data center? To answer these questions, we evaluate the performance gains and energy efficiency of the proposed scheme on simulated data center environments of various sizes. Here, we consider four currently widely-used data center network architectures (Tree[11], VL2[12], Fat-Tree[13], Bcube[14]) with different communication expression between two servers (see [1]).

Given that the four architectures mentioned above have significant differences, we evaluated how the performance gains due to our joint VM placement are affected and compared the performance of the proposed solution against the one obtained with random placement algorithm. In the simulation, we generated a scenario that includes 16 servers (server number labeled as 1, 2... 16) for the data center network topologies, varying the traffic demands of VMs and the interdependencies amongst them that represents various application topologies, as well as varying the load characteristics of the PMs and VMs. The traf-

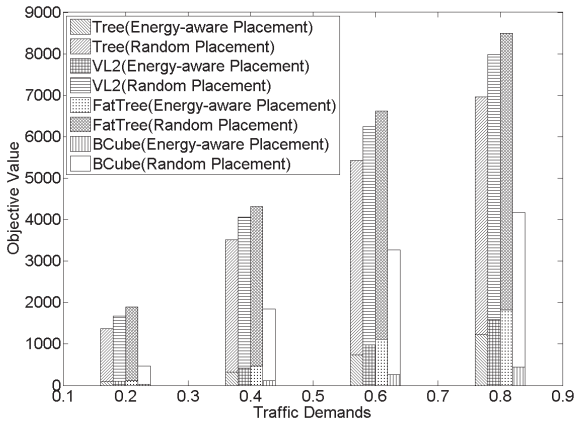

Fig. 2. Energy-aware placement vs. objective value achieved by random placement

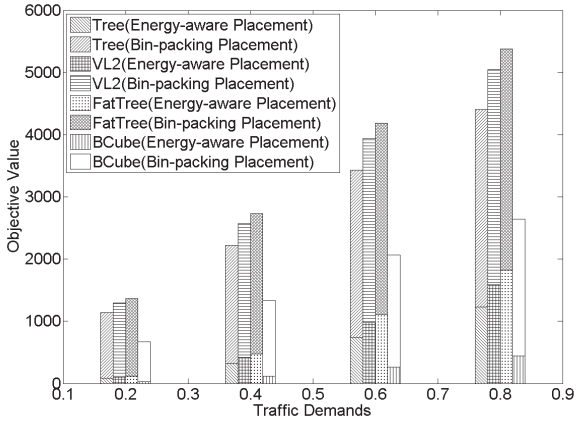

Fig. 3. Energy-aware placement vs. objective value achieved by bin-packing placement

fic demands of VMs meet the normal distribution $N(0.2, 0.1)$, $N(0.4, 0.1)$, N(0.6,0.1), N(0.8,0.1). Simulation results are the average of 1000 runs. Fig.2 gives the comparison results, where the abscissa of the figure stands for the average of traffic. For example, 0.2 represents that communication traffic between VMs meets the normal distribution with 0.2 as mean and 0.1 as variance. The top of the histogram indicates the potential enhancement from random placement model to our mechanism, since the resource wastage and communication costs should be as low as possible. In addition, the figure shows that the enhancement gap is larger under the BCube architecture compared to the other ones. We further compares the objective values of the energy-aware placement algorithm with that of the traditional method of bin-packing algorithm in Fig.3, and the results show that both communication cost and resource wastage of the energy-aware placement method are lower.

The energy-aware placement algorithm takes into account of inherent coupling of composite applications, so it can greatly reduce the total data center network traffic burden. TABLE I shows the comparison results among the energy-aware approach, the random approach, and the bin-packing algorithm.

TABLE I
REDUCTION RATE OF TRAFFIC VOLUME ACHIEVED BY OUR APPROACH

Topologies	**Number of VMs**				
	16	**32**	**64**	**128**	**256**
Tree(B)	0.4989	0.5077	0.4850	0.4898	0.4793
Tree(R)	0.6833	0.6880	0.6740	0.6769	0.6705
VL2(B)	0.4015	0.4096	0.3977	0.3926	0.3739
VL2(R)	0.6224	0.6254	0.6174	0.6139	0.6027
Fat-Tree(B)	0.3408	0.3536	0.3301	0.3203	0.3007
Fat-Tree(R)	0.5825	0.5904	0.5738	0.5692	0.5556
BCube(B)	0.7019	0.7099	0.6965	0.7001	0.6931
BCube(R)	0.8111	0.8163	0.8082	0.8107	0.8059

We also consider the impact of different network architectures, the number of VMs were taken as 16, 32, 64,128, 200, 256, respectively. All results are the average of the 1000 runs. We found that, in the Tree, VL2, Fat-Tree, BCube frameworks, the energy-aware algorithm saves about 68%, 62%, 58%, 80% traffic flow, respectively compared with the random algorithm, and saves about 50%, 40%, 33%, 70% traffic flow, respectively compared with the bin-packing method. We also found that in a fixed number of physical hosts, with the number of VMs increasing, reduction proportion of the traffic flow gradually decreases, being in line with the reality of the data center. The (B) and (R) in TABLE I are used to represent bin-packing and random placement, respectively.

Fig. 4 compares the performance of the proposed approach against four traditional methods: grouping genetic algorithm (GGA)[6], First-Fit Decreasing (FFD) heuristic, two stage heuristic algorithm[19], and random placement. The parameters are given as: (1)The servers used have a fixed capacity of 150, consisting of items of sizes uniformly distributed in (20,100); (2) In GGA, the size of population, generation number,crossover rate and mutation rate are 200, 1000, 0.8, 0.01 respectively; (3)The number of item-item conflicts and the bin-item conflicts in GGA and two stage heuristic algorithm are set as 5. We experimented with a diverse of problem instances and varying problem sizes. Compared to the traditional FFD and the genetic algorithm, the energy-aware placement method occupies the intermediate number of physical machines, that is, due to tradeoffs between multiple objectives(minimizing the total resource wastage and traffic burden simultaneously), the number of used servers in our approach is higher than FFD bin-packing placement and optimal resolution, but lower than random placement, grouping genetic algorithm and the two stage heuristic algorithm. Furthermore, with the increasing number of the VMs, the proposed approach is converged to optimal solution, i.e., the energy-aware joint placement is scalable and optimal.

V. CONCLUSION

This paper proposes a joint energy-aware and application-aware VM placement strategy based on the theory of multi-objective optimization, achieves the objective functions satisfying server-side constraints and network traffic flow awareness, analyzes the impacts on VM placements of the dependen-

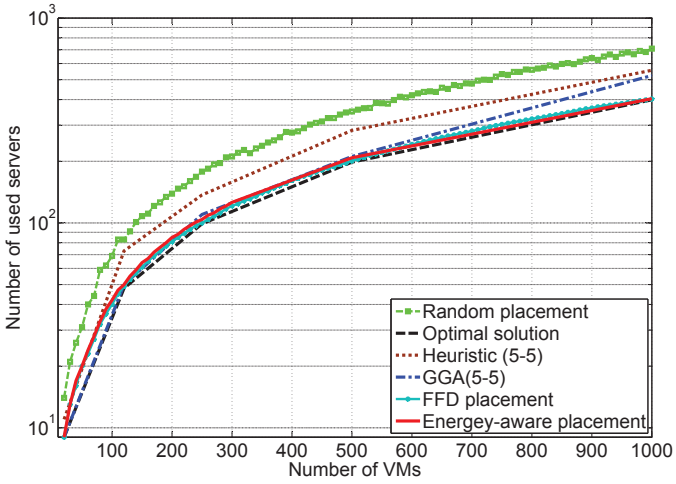

Fig. 4. Number of used physical machines for five placement algorithms

cies among data center infrastructure, server-side constraints, and applications.

Theoretical analysis of this paper is very meaningful in solving the issues of energy savings in data centers. Compared to existing VM placement algorithms that focus on only server-side constraints, power consumptions or transmission awareness, energy-aware VM placement algorithm can maximize the utilization of the physical machines while decreasing data center traffic by up to $50\% \sim 81\%$, given the conditions of the server-side constraints and different data center network architectures, significantly reduce server energy consumption and network energy consumption in data centers.

APPENDIX A
PROOF OF THEOREM 1

Proof: The constraints of $(\mathbf{P}_1)$ are linear, provided that its constraint domain is convex set. In addition, $\frac{\partial^2 U_p}{\partial x_{pv}^2(t)} < 0$, $\frac{\partial^2 U_p}{\partial x_{pv}(t)\partial x_{pw}(t)} < 0$ are satisfied, thus, the Hesse matrix of the objective function is negative definition, i.e., the objection function is concave but not strictly concave with respect to $x(t) = (x_{pv}(t), p \in P, v \in V)$. According to [10], $(\mathbf{P}_1)$ is a convex programming problem, and the optimal solution exists but not unique. Since $\frac{\partial^2 U_p}{\partial y_p^2(t)} < 0$, the objection function is strictly concave with respect to $y = (y_p(t), p \in P)$, the unique optimal solution exists for total load requirements. ■

APPENDIX B
PROOF OF THEOREM 2

Proof: Based on the fact that $\mathbf{0} \in R$, $R \neq \emptyset$. Assuming $x_k(t) \geq 0, k = 1, 2, ..., m$, R is not empty bounded set. Given that the constraints of $(\mathbf{P}_2)$ are linear, it is ensured that R is a closed set. For the case that the objective function described in $(\mathbf{P}_2)$ is linear and continuous function, $(\mathbf{P}_2)$ is equivalent to seek the maximum values problem upon non-empty, bounded closed set. Thus, unique optimal solution exists, i.e., $R^* \neq \emptyset$. ■

ACKNOWLEDGMENT

The authors would like to thank Fei Song for helpful feedback on earlier versions of this paper.This work is supported in part by the Natural Science Foundation of China under Grant No. 60833002 and 60903150, in part by the National High-Tech Research and Development Program of China (863) (2011AA010701).

REFERENCES

[1] X. Meng, V. Pappas, and L. Zhang, *Improving the scalability of data center networks with traffic-aware virtual machine placement*, in In Proc. INFOCOM, 2010.

[2] H. N. Van, F. D. Tran, J.M. Menaud, *Autonomic virtual resource management for service hosting platforms*, Software Engineering Challenges of Cloud Computing, ICSE Workshop on, 2009.

[3] N. Bobroff, A. Kochut, and K. Beaty. *Dynamic placement of virtual machines for managing SLA violations*, Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on. pp. 119-128.

[4] S. Chaisiri, B.S. Lee, D.Niyato, *Optimal virtual machine placement across multiple cloud providers*, In 2009, IEEE Asia-Pacific Services Computing Conference, APSCC, 2009. pp. 103-110.

[5] H. Nakada, T. Hirofuchi, H. Ogawa, S. Itoh. *Toward virtual machine packing optimization based on genetic algorithm*, In IWANN '09: Proceedings of the 10th International Work-Conference on Artifcial Neural Networks, Springer-Verlag, 2009. pp. 651-654.

[6] S. Agrawal, S. K. Bose, S. Sundarrajan. *Grouping genetic algorithm for solving the server consolidation problem with conflicts*, In GEC '09: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, ACM, 2009, pp. 1-8.

[7] F. P. Kelly, A. Maulloo, David Tan. *Rate control for communication networks: shadow prices proportional fairness and stability*, Journal of Operations Research Society, 1998, 49(3): 237-252.

[8] M. Chiang, S. H. Low, A. R. Calderbank, J. C. Doyle. *Layering as optimization decomposition:A mathematical theory of network architectures*, Proceedings of the IEEE, 2007, 95 (1):255-312.

[9] H. W. Kuhn and A. W. Tucker, *Nonlinear Programming*, in Jerzy Neyman (ed.), Proceedings of the Second Berkeley Symposium on mathematical Statistics and Probability (Berkeley, Univ. of Calif. Press, 1951), 481-492.

[10] S. Boyd, L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press,2004.

[11] M. Al-Fares, A. Loukissas, and A. Vahdat, *A scalable, commodity data center network architecture*, in SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication. ACM,2008, pp. 63-74.

[12] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri,D. A. Maltz, P. Patel, and S. Sengupta, *VL2: A scalable and flexible data center network*, in SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication, 2009.

[13] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, *Dcell: a scalable and fault-tolerant network structure for data centers*, in SIGCOMM'08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication. ACM, 2008, pp. 75-86.

[14] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, *Bcube: A high performance, server-centric network architecture for modular data centers*, in SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication. ACM, 2009.

[15] L. A. Zadeh, et al. 1996 *Fuzzy Sets, Fuzzy Logic, Fuzzy Systems*, World Scientific Press.

[16] R. Yager, *On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making*, in IEEE Trans. on Systems, Man and Cybernetics,1998.

[17] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H.Freeman, 1979.

[18] H. Saran and V. V. Vazirani, *Finding k cuts within twice the optimal*,SIAM J. Comput., vol. 24, no. 1, pp. 101-108, 1995.

[19] R. Gupta, S. K. Bose, S. Sundarrajan, M. Chebiyam, A. Chakrabarti, *A two stage heuristic algorithm for solving server consolidation problem with Item-Item and Bin-Item Incompatibility Constraints*, In: Proceedings of IEEE Services Computing, Hawaii, USA, pp. 39-46, 2008.