NICE: Network-Aware VM Consolidation Scheme for Energy Conservation in Data Centers

Bo Cao[1], Xiaofeng Gao[1*], Guihai Chen[1], Yaohui Jin[2]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
*Corresponding author
[2] Network & Information Center, Shanghai Jiao Tong University
boscof@sjtu.edu.cn, {gao-xf, gchen}@cs.sjtu.edu.cn, jinyh@sjtu.edu.cn

***Abstract*—Energy conservation and network performance have become two of the most important issues in data center as the scale of cloud services continues growing. Recent researches usually consider these two issues separately. Energy conservation mainly deals with hosts, which reduces total energy consumption by consolidating virtual machine(VM)s to fewer hosts, and network performance mainly deals with network scalability and energy efficiency, which improves data center network(DCN) scalability by applying new network topologies or routing schemes and improves DCN energy efficiency by consolidating traffic. In this paper, we jointly consider these two issues and define *Combined VM Consolidation* (CVC) problem. We prove that CVC is NP-complete and is inapproximable by a factor of $3/2$ ε unless P = NP. Next, we propose NICE: *Network-aware VM ConsolIdation scheme for Energy Conservation in Data CEnter* to solve CVC. Instead of taking the unrealistic hypothesis that migration cost is negligible, a common assumption in most literatures, we precisely analyze VM migration cost according to real-trace experiments in a 6-server testbed via VMware. Massive simulations validate the efficiency of NICE. In all, to the best of our knowledge, we are the first work to combine VM consolidation with network optimization and migration cost.**

***Keywords*-cloud computing; data center; VM consolidation; energy saving; VM migration cost**

I. Introduction

As the increasing demands for a wide spectrum of cloud services, energy conservation becomes one of the most important issues in modern data center. There are many works [1]–[8] trying to save energy in data center, and the common way is through improving energy efficiency of hosts and other physical facilities. For instance, literatures [1]–[5] applied *live virtual machine(VM) migration* technology to conserve energy by first consolidating VMs to fewer hosts and then turning idle hosts off. This can be implemented via many management softwares such as Xen and VMware, and the service downtime due to migration is extremely short.

Apart from energy conservation, network performance is another important issue in modern data center, especially the scalability and energy efficiency of data center network(DCN). The common way to tackle the scalability issue is through applying new network topologies, among which DCell [9], BCube [10], VL2 [11], Portland [12] are some of the most excellent works. Meng *et. al.* [13] tried to improve DCN scalability from a different perspective. They mapped VM pairs with heavy mutual traffic to slot pairs in hosts with low cost connections in order to localize traffic as much as possible. In this way, the amount of traffic going through the core level switches is greatly reduced ,and this leads to a more scalable DCN. In [14], Li *et. al.* also tried to reduce the network cost during the VM placement procedure. However, Tso *et. al.* [15]define a novel distributed migration scheme to reduce VM communication cost. As for improving DCN energy efficiency, the common way is through dynamically adjusting network links and switches according to traffic demands. Abts *et. al.* [16] exploited the energy-proportional characteristic of modern plesiochronous links to reduce energy consumption of DCN by dynamically adjusting data transmission rates of existing links according to traffic rate demands. Chabarek *et. al.* [17], Heller *et. al.* [18] and Wang *et. al.* [19] designed novel traffic routing schemes to achieve DCN energy conservation by dynamically turning on-off switches and cables. Mann *et. al.* [20] and Fang *et. al.* [21] exploited VM migration technology to improve energy saving potential of DCN.

However, none of the above works jointly considered reducing host energy consumption and improving DCN scalability and energy efficiency. It is obvious that VM consolidation affects traffic distribution in DCN, and traffic distribution is directly related to the scalability and energy saving potential of DCN. Therefore, we propose to reduce host energy consumption and to improve DCN scalability and energy efficiency at the same time through VM consolidation. Instead of taking the unrealistic hypothesis that migration cost is negligible, in this paper we define cost function for VM migration through real-trace experiments in a 6-server testbed via VMware. We also illustrate the cost function for host energy consumption and network cost. Based on these cost functions, we formally define *Combined VM Consolidation* (CVC) which minimizes host energy consumption, network cost and migration cost. We prove that CVC is NP-complete by a reduction from classic *Bin Packing Problem.* We also analyze the hardness of approximation for CVC, say, CVC is inapproximable by a factor of $3/2 - \varepsilon$ unless P = NP, where ε is any given positive real number.

In order to solve CVC, we propose an effective scheme named NICE: *Network-aware VM ConsolIdation scheme for Energy Conservation in Data CEnter*. NICE consists of three subroutines. Firstly, we use *FirstFit-Decr* algorithm to pack all VMs into virtual hosts according to VMs' computing

978-1-4799-7615-7/14/$31.00 ©2014 IEEE

request, with an approximation ratio of $\frac{11}{9}$ to the optimal result. Secondly, we implement *Kernighan-Lin* algorithm to minimize total inter-host traffic. Finally, we design a greedy approach *Greedy-Map* which iteratively maps virtual host to physical host with minimum network cost and migration cost increase until all virtual hosts are mapped.

We summarize our contributions as follows:

- We define migration cost quantitatively based on real-trace experiments, and we analyze host energy consumption and network cost thoroughly.
- We formally formulate *Combined VM Consolidation* (CVC) problem and prove its computational complexity and hardness of approximation. To the best of our knowledge, we are the first work jointly considering host energy consumption, network cost and migration cost in VM consolidation.
- We propose an effective scheme NICE to solve CVC, which is a three-step heuristic.
- We validate the efficiency of NICE via massive simulations under various scenarios.

The rest of the paper is organized as follows. In Section II, we review some related works. In Section III, we present important observations about date center host and DCN, and introduce our real-trace experiments on VM migration cost. Problem formulation and proof of CVC's complexity and hardness of approximation are presented in Section IV. We describe our scheme NICE in Section V. Evaluation results are shown in Section VI. We conclude our work in Section VII.

II. Related Work

In [2] Verma *et. al.* first introduced various formulation of the cost-aware Application placement problem. For each formulation, they described model assumptions, analyses of practicality of these assumptions and information required to solve corresponding problems. Their work showed comprehensive theoretical and experimental evidence to establish the efficacy of pMapper and they proposed *MP ratio* to combine migration cost and power consumption. But it didn't consider network performance, neither the communication cost. Similar to [2], Jung *et. al.* [3] built a system that optimizes power consumption, performance benefits, and transient costs incurred by adaptation. In [4], Zu *et. al.* proposed an efficient power-aware resource scheduling strategy that reduces data center's power consumption effectively based on VM live migration.

Chabarek *et. al.* [17] state that power consumption are becoming a significant technology challenges that threaten to slow bandwidth growth. They described the power issues in routers, and advocated power-awareness be a primary objective in the design and configuration of DCN. Heller *et. al.* [18] proposed a system for reducing energy consumption of DCN by dynamically turning on-off links and switches while imposing little or no harm to network performance. They implemented their design on a hardware testbed using OpenFlow and Nox. Their results show that there are great potential of energy saving for dynamic DCN. To make dynamic DCN more practical, Wang *et. al.* [19] used data mining technique to predict coming traffic demands and adjusted DCN according to the predicted workload. They also considered correlations between traffic in the process of traffic consolidation, which can further improve the performance of dynamic DCN.

In [20], Mann *et. al.* proposed a framework for VM placement and migration. They took both the network topology and network traffic demands into account so that the network energy consumption can be minimized while satisfying as many network demands as possible. Fang *et. al.* [21] did similar work except that they assumed that instead of only one VM per server, several VMs can be placed on one server. Shang *et. al.* [22] mainly focused on how to save energy in high-density data center network from the routing perspective. In [23], Huang *et. al.* proposed enery-aware VM placement in which they considered application dependencies to reduce network energy consumption.

III. Problem Analysis

In this section, we first present our definition of cost functions for host and network. Then we introduce our real-trace experiments on VM migration cost.

A. Power Consumption of Host

There are many studies working to quantitatively define power consumption of host in modern data center. Most studies show that CPU consumes more power than other activities, such as memory and I/O, and these activities are usually positively proportional to CPU utilization. In [24], Economou *et. al.* demonstrated such observation by experiments, which confirms that CPU takes the main part of power consumption. Similar result can be found in [25], in which Fan *et. al.* claim that using CPU utilization as the main factor to estimate the power consumption of a host is reasonably accurate. Based on their analyses, the power drawn by a host can be estimated using the following formula:

$$P_h(u) = \begin{cases} P_h^{\mathrm{I}} + (P_h^{\mathrm{F}} - P_h^{\mathrm{I}})(2u - u^{\alpha}), & u > 0 \\ 0, & u = 0 \end{cases} \tag{1}$$

$P_h(u)$ denotes the total power of a host with CPU utilization u. P_h^{I} denotes idle power of host, while P_h^{F} denotes the power when host works at full CPU utilization. α is a parameter used to minimize the squared errors varying with different host types, 1.4 in Fan's experiment. We use Eqn. (1) to estimate power consumption of host in data center. For example, if the idle power of one host is$100w$, the full power is $150w$ and the utilization is 0.5, its power usage is $125w$ according to Eqn. (1) while $\alpha = 1$.

B. Traffic Cost of DCN

Data center network is a group of cable-linked switches which provide host to host and host to Internet communication. As the scale of cloud data center continues growing, the core level of DCN often has too much traffic to handle. In order to mitigate this scalability issue, we can make more traffic transmit locally by consolidating traffic. Besides scalability, energy saving of DCN has also attracted significant attention recently.

Traditionally, switches are turned on all the time. However, recent power measurement studies present that dynamically turning on and off switches on demand leads to a more energy-efficient DCN. In [18], Heller *et. al.* observe that when the rate of all the ports in a switch goes from 0% to 100%, the power of the switch increases at most 8%. This shows that turning off a switch yields most energy saving. Similar result can be found in [19]: an idle switch with no active ports consumes more than half of the total energy consumed by a switch with all ports turned on. If we can turn off unused switches through traffic consolidation, we can save a large amount of energy in DCN. Therefore instead of defining network cost as its power consumption, we define traffic cost as follows to localize traffic as much as possible:

$$P_t(r_{ab}, h_a, h_b) = r_{ab} \cdot f(h_a, h_b) \tag{2}$$

Here r_{ab} is the traffic rate between hosts h_a and h_b. $f(h_a, h_b)$ denotes the communication cost between hosts h_a and h_b, which is positively proportional to communication distance between h_a and h_b (usually represented as hop-counts). In this way, minimizing traffic cost will localize traffic and lead to a more scalable and more energy-efficient DCN.

C. VM Migration Cost

Substantially, VM migration is to shift resources and to transmit memory pages, of which the latter incurs more cost. In order to perform VM migration, hosts need to do extra work, which increases their energy consumption. The VM memory pages are transmitted through DCN, which causes extra traffic cost.

In [26], Clark *et. al.* propose live VM migration, in which "live" means that candidate VM continues working during most of its migrating period, and the service downtime is very short. In [27], Voorssluys *et. al.* present case study of a modern Internet application to quantify the effect of live VM migration. Their results exhibit that even though live VM migration brings additional traffic, it still performs better than offline migration. The service downtime is an important metric in VM migration, and in live VM migration it is extremely short. This greatly reduces the risk of Service Level Agreement (SLA) violation. However, none of these works analyzed migration cost quantitatively, making it extremely difficult for us to precisely capture the cost of VM migration.

Correspondingly, we define traffic cost of VM live migration as Eqn. (2), and we quantify extra energy consumption of hosts incurred by live VM migration process through real-trace experiments. We set up a testbed consisting of 6 *Dell PowerEdge R710* hosts and perform VM live migration via VMware between two hosts. After connecting each host to a *powerbay series North Meter*, we record the power state of source and destination hosts during VM live migration. The memory size of migrating VM is 2040MB. We conclude from 50 times VM migration data that on average VM live migration lasts 14s, and the average power increase of pairwise hosts is about 12w. Figure 1 shows the power state of the two hosts which conduct VM live migration.

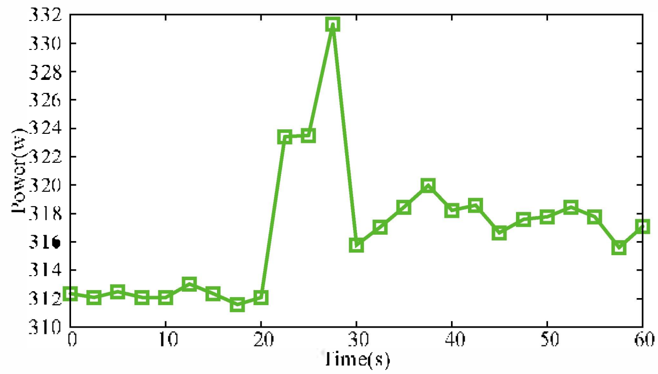


Fig. 1. The power consumption of pairwise hosts during migration period

We can see from Fig. 1 that total power of the two hosts stays stable before and after VM live migration, and changes rapidly when VM migration begins. We can also see that before and after VM migration the total power increases. This is mainly due to the small differences of distinct hosts.

We formally define VM migration cost as the sum of host power consumption and traffic cost:

$$C(s, h_c, h_d) = \begin{cases} P_h^{\mathrm{VM}} \cdot T_0 + s \cdot f(h_c, h_d), & h_c \neq h_d \\ 0, & h_c = h_d \end{cases} \tag{3}$$

Here P_h^{VM} is the power increase of hosts, T_0 is the time duration of VM migration. s is the memory size of candidate VM, and $f(h_c, h_d)$ is the communication cost between current host h_c and destination host h_d.

At this moment we are able to formulate the VM consolidation problem mathematically, which is described in details in the next section.

IV. Problem Formulation

In this section, we first formally define *Combined VM Consolidation* (CVC) problem as a constrained optimization problem. Then we prove the computational complexity and the hardness of approximation of CVC.

A. Problem Formulation

Our goal is to study the problem of consolidating VMs to reduce total host energy consumption and network cost as well as migration cost. Briefly speaking, we consider the common scenario where n VMs are running on m hosts, and we assume that traffic rates between arbitrary hosts, memory page sizes of VMs $(s_1, \ldots, s_n)$ and their computing requests $(cr_1, \ldots, cr_n)$ are all known. We conduct VM consolidation every T time, and our goal is to find a consolidation scheme which brings maximum cost saving for the whole system. Notice that we have to take off the energy cost spent for VM migration from the overall cost saving after migration, and thus the *Combined VM Consolidation* (CVC) problem can be formally defined as follows:

$$\max \quad \sum_{i=1}^{m}\Big(P_h(u_i)-P_h(u_i^{'})\Big)T-\sum_{j=1}^{n}C(s_j,h(j),h^{'}(j))+$$
$$\sum_{i=1}^{n}\sum_{j=1}^{n}\Big(P_t(r_{ij},h(i),h(j))-P_t(r_{ij},h^{'}(i),h^{'}(j))\Big)T \qquad (4)$$

subject to:

$$\sum_{h^{'}(j)=i} cr_j \leq 1, \quad \forall i, \qquad (5)$$
$$h^{'}(j)\neq \emptyset, \quad \forall j. \qquad (6)$$

In the above formulation, $h(j)$ denotes the host that vm_j is currently placed on, $h^{'}(j)$ denotes the host afterward and

$$u_i=\sum_{h(j)=i} cr_j, \quad u_i^{'}=\sum_{h^{'}(j)=i} cr_j.$$

Hence, $P_h(u_i)$ is the power of host h_i before consolidation, and $P_h(u_i^{'})$ is the power of host h_i after consolidation. Accordingly, $\sum_{i=1}^{m}(P_h(u_i)-P_h(u_i^{'}))T$ is the total energy conservation of all m hosts. $\sum_{j=1}^{n}C(s_j,h(j),h^{'}(j))$ is the total migration cost of n VMs. $\sum_{i=1}^{n}\sum_{j=1}^{n}(P_t(r_{ij},h^{'}(i),h^{'}(j))-P_t(r_{ij},h(j),h(j)))T$ is the total traffic cost conservation.

Constraint (5) ensures that the total computing requests from all VMs placed on one host will not exceed the capacity of this host. Constraint (6) guarantees that at least one host is used to accommodate vm_i.

Since the current host power $\sum_{i=1}^{m}(P_h(u_i))T$ and traffic cost $\sum_{i,j=1}^{n}(P_t(r_{ij},h(i),h(j)))T$ are known, we can transform this maximum optimization problem into its equivalent minimization form as follows:

$$\min \sum_{i=1}^{m}P_h(u_i^{'})T+\sum_{j=1}^{n}C(s_j,h(j),h^{'}(j))+$$
$$\sum_{i,j=1}^{n}P_t(r_{ij},h^{'}(i),h^{'}(j))T \qquad (7)$$

Now CVC becomes a minimization problem, jointly considering host power consumption, network cost and migration cost. Total energy consumption of hosts can be calculated through Eqn. (1), network cost through Eqn. (2) and migration cost through Eqn. (3).

We also can use CVC to formulate VM placement problem where the main goal is to place n VMs on m empty hosts with minimum cost. We can achieve this by removing migration cost from (7). In this way, CVC becomes a *Combined VM Placement problem* formulation, which places VMs in cost minimizing manner.

Table I summarizes main notations used in this paper.

TABLE I
MAIN NOTATIONS

Symbol	Description
$V=\{vm_i\}$	Set of VMs
$H=\{h_i\}$	Set of hosts
$R=\{r_{ij}\}$	traffic rate matrix
$F=\{f(i,j)\}$	communication cost matrix
$CR=\{cr_i\}$	Set of VM computing requests
$S=\{s_i\}$	Set of VM memory sizes
$VH=\{vh_i\}$	Set of virtual hosts
$PH=\{ph_i\}$	Set of physical hosts
T	Time interval of conducting NICE
$P_h(u)$	power of host while working at utilization u
$P_t(r_{ab},h_a,h_b)$	cost of traffic with rate r_{ab} going from h_a to h_b
$C(s,h_c,h_d)$	cost of migrating VM from h_c to h_d
$h(i)$	host on which vm_i is placed

B. Complexity Analysis

Actually, as shown in [13], when the communication cost and traffic rates are arbitrary real values, subproblem of minimizing network cost in CVC falls into the category of *Quadratic Assignment Problem (QAP)* which is one of the most difficult problems in NP-hard class. However, CVC imposes special constraints on communication cost and traffic rates. Now, we analyze the computational complexity of CVC. As we all know, in the NP-complete prove process, proving that a special instance of one problem is NP-complete will indicate the original problem is NP-complete. Therefore, in this section, we prove that the special CVC instance in which both traffic cost and migration cost are ignored is NP-complete.

Theorem 1. *CVC is **NP**-complete.*

Proof: In [28], Vazirani formally define the decision version of *Bin Packing Problem* in this way: given n items with sizes $a_1,\dots,a_n\in(0,1]$ and a constant B, we need to find a packing of items into unit-sized bins so that the number of used bins is at most B. We can construct an instance of CVC from an instance of bin packing problem as follows:

- Given n VMs $vm_1,\dots,vm_n$ with computing requests $a_1,\dots,a_n\in[0,1]$, we have m hosts $h_1,\dots,h_m$ with computing capacity volumes $v_1,\dots,v_m$ and power consumption $p_1,\dots,p_m$. In this case, assume
$$v_1=\ldots=v_m=1, \text{ and } p_1=\ldots=p_m.$$
There is a special host h_0 who can accommodate all VMs, *i.e.,* $c_0>\sum_i a_i$ and when it is active, it consumes more power than the total power consumption of all other hosts, *i.e.,* $p_0>m\cdot p_m$. Our goal is to find the next-stage VM placement with minimum total power consumption.
- If there are VMs on host h_i, the power of h_i is p_i. Otherwise, its power is 0. Therefore the total power consumption of hosts is $num\cdot p_m$, since the power consumption of each host is the same. Here num is the number of active hosts in the new consolidation.
- The cost to migrate vm_i from one host to another is 0.

The VMs in CVC can be regarded as items while the hosts as bins. Now we can see that bin packing instance has a solution with B bins *iff* CVC instance has a solution with total power $B \cdot p_m$. Therefore, CVC is NP-complete. ■

Next, we prove CVC's harness of approximation.

Theorem 2. *For any $\varepsilon > 0$, there is no approximation algorithm having a guarantee of approximation ratio less than $3/2 - \varepsilon$ for CVC, unless $P = NP$.*

Proof: If there were such an algorithm, then we can solve the deciding version of NP-hard problem *Partition* in polynomial time, which is to decide if there is a way to partition n non-negative numbers $a_1, \ldots, a_n$ into two sets, each set equals to $\frac{1}{2}\sum_i a_i$. We can construct an instance of CVC in this way: there are n VMs with sizes of $a_1, \ldots, a_n$ and the capacities of hosts $h_0, \ldots, h_m$ all equal to $\frac{1}{2}\sum_i a_i$. Obviously, the answer to the partition question is *yes* iff the n VMs can be packed into 2 hosts, and then answer is *no* iff more than 3 hosts are needed. Suppose that there is an approximation algorithm having a guarantee of approximation ratio less than $3/2 - \varepsilon$, then this approximation algorithm can solve the deciding version of partition question in polynomial time. This is because if the approximation result of CVC is 2, the answer to the partition question is *yes*. If the approximation result of CVC is 3 or more, the answer to partition question is *no* since the optimal result cannot be 2, otherwise the approximation ratio $3/2 - \varepsilon$ will not be satisfied ($3/2 > 3/2 - \varepsilon$). This is impossible unless $P = NP$! Therefore, for any $\varepsilon > 0$, there is no approximation algorithm having an approximation ratio less than $3/2 - \varepsilon$ for CVC unless $P = NP$. ■

V. NICE: An Efficient Scheme for CVC

Previous analyses show that CVC is NP-complete, which cannot be solved within polynomial time. In this section, we design an efficient heuristic scheme to solve CVC. As shown in Eqn. (7), our goal is to minimize host energy consumption, network cost and migration cost through VM consolidation. On one hand, we know from our VM migration cost experiment that VM migration lasts relatively short, and the total energy consumption and traffic cost it incurs are relatively small. On the other hand, DCN consumes much less energy than hosts in modern data center [16]. Based on these observations, we propose a scheme named NICE (***N**etwork-aware VM Conso**I**idation scheme for Energy **C**onservation in Data C**E**nter*) to solve CVC efficiently.

NICE consists of three subroutines: *FirstFit-Decr* algorithm for packing VMs into virtual hosts; *Kernighan-Lin* algorithm for adjusting these virtual hosts to minimize inter-host traffic; *Greedy-Map* algorithm for mapping virtual hosts to physical hosts. Algorithm 1 describes NICE.

A. FirstFit-Decr Subroutine

VMs $vm_1, \ldots, vm_n$ with computing requests $cr_1, \ldots, cr_n$ are to be placed on m hosts with computing capacity 1. Our goal is to pack all VMs into minimum number of virtual hosts. We use

Algorithm 1: NICE

Input: Set of VMs $V = \{vm_1, \ldots, vm_n\}$; set of VM computing requests $CR = \{cr_1, \ldots, cr_n\}$; Set of hosts $H = \{h_1, \ldots, h_m\}$; Traffic rate matrix $R = \{r_{ij}\}_{m \times m}$; Communication cost matrix $F = \{f(i,j)\}_{m \times m}$
Output: An VM consolidation assignment NICE for each vm_i

$k, VH \leftarrow$ FirstFit-Decr(V, CR) ; /* Group VMs */
$VH \leftarrow$ Kernighan-Lin(VH, R, k) ; /* Adjust k virtual hosts */
NICE $\leftarrow$ Greedy-Map(VH, PH) ; /* Map virtual hosts */

FirstFit-Decr algorithm to solve this sub-problem. We first sort all VMs in non-increasing order according to their computing request $cr_1, \ldots, cr_n$ and then pack n VMs into virtual hosts one by one. VH denotes the set of virtual hosts $vh_1, \ldots, vh_k$ while $u(vh_i)$ indicates the utilization of vh_i. The pseudo codes are described in Algorithm 2:

Algorithm 2: FirstFit-Decr

Input: V; CR
Output: k, VH

$k \leftarrow 1$; /* Initial number of virtual hosts k is 1 */
$u(vh_i) \leftarrow 0$; /* Initial utilization for $u(vh_i)$ is 0 */
Sort all VMs in non-increasing order according to cr_i;
for $i = 1$ **to** n **do**
  $flag \leftarrow$ **true**;
  for $j = 1$ **to** k **do**
    if $cr_i + u(vh_j) \leq 1$ **then**
      $u(vh_j) \leftarrow cr_i + vh_j$;
      Place vm_i on vh_j;
      $flag \leftarrow$ **false**;
    end
  end
  if $flag$ is **true then**
    $k \leftarrow k + 1$;
    $u(vh_j) \leftarrow cr_i$;
    Place vm_i on vh_k;
  end
end

Next, we prove the effectiveness of FirstFit-Decr algorithm.

Theorem 3. *The tight bound for FirstFit-Decr algorithm is $k \leq \frac{11}{9} opt + \frac{6}{9}$, where k is the output of FirstFit-Decr and opt is the optimal number of virtual hosts.*

Proof: It is proven in [29] that $FFD \leq \frac{11}{9} opt_B + \frac{6}{9}$, where FFD is the output of First Fit Decreasing algorithm in bin-packing problem and opt_B is the optimal value. Thus, $k \leq \frac{11}{9} opt + \frac{6}{9}$. ■

B. Kernighan-Lin Subroutine

After FirstFit-Decr subroutine, n VMs are packed into k virtual hosts. Since the communication cost between VMs in the same virtual host is 0, we hope to minimize inter-host traffic by adjusting these k virtual hosts so that total traffic cost can be reduced. This problem can be rephrased as a *Graph Partitioning Problem*, which is to partition graph vertices into k disjoint groups with minimum edge cut cost. The maximum size of the groups is limited, varying according to different scenarios, and each edge in the graph has corresponding cost. Kernighan-Lin algorithm [30] minimizes the total cost of edge cut by iteratively choosing two partitions from original k partitions and exchanging some of their vertices to reduce total cut cost. In NICE, we treat VMs as vertices, virtual hosts as partitions and traffic rate between VMs as edge cost. After applying Kernighan-Lin algorithm to k virtual hosts generated by FirstFit-Decr subroutine, we get a new VH set whose inter-host traffic is minimized.

C. Greedy-Map Subroutine

In the last subroutine, we use a greedy algorithm to map k virtual hosts to k physical hosts. In each iteration, we find vh_{gm} and ph_{gm} from the remaining virtual hosts and physical hosts, whose mapping incurs minimum traffic cost and migration cost C_{tm} which is calculated according to Eqn. (2) and Eqn. (3). Continue doing this until all virtual hosts are mapped. The pseudo codes are described in Alg. 3.

Algorithm 3: Greedy-Map

Input: R, F, VH, H
Output: Mapping of virtual hosts to physical hosts

```
for i = 1 to k do
    for vh ∈ VH, h ∈ H do
        Find vh_gm and ph_gm with minimum C_tm;
        Map vh_gm to ph_gm;
        Delete vh_gm from VH and delete ph_gm from H;
    end
end
```

VI. Evaluation

In this section, we first present our experiment settings, then we show our evaluation results with corresponding analyses.

A. Experiment Settings

As shown in section III, in order to quantify traffic cost and migration cost in CVC, communication cost $f(s,d)$ is introduced, which varies according to different circumstances and network topologies. In our evaluation, we set $f(s,d)$ positively proportional to the length of transmission path between host s and d, and this length depends on the specific network topology of DCN. Although many new DCN topologies have been proposed, such as DCell [9], BCube [10], VL2 [11], Portland [12], network topology used in most modern data centers is more likely to be multi-rooted tree topology. In the simulation, we have conducted experiment in which we compare the performance of NICE under different DCN topologies.

As for traffic rate matrix, even though traffic rates between VMs vary from time to time, it is shown in [13] that a large fraction of inner VMs traffic rates are relatively constant. Besides, VM consolidation doesn't affect the total cost of outer traffic, *i.e.,* the traffic between host and outer Internet. Therefore when we conduct NICE, we can use current traffic rates to calculate traffic cost to find out the best consolidation scheme and adjust VMs according to this scheme. In the next round of NICE, we use renewed traffic matrix to recalculate traffic cost. In our simulation, we generate traffic rate matrix based on two models. The first global traffic model assumes that each VM sends traffic to all other VMs at equal rate, and the rates for different VMs vary. The second local traffic model partitions VMs into groups, and each VM only communicates with VMs in the same group. We mix up these two models to get the traffic rate matrix used in our simulation, which can reflect traffic pattern in reality according to [13]. Simulations are implemented on Visual Studio 2012 and carried out on a 64-bit Intel Xeon E5645 2.5GHZ 2-processors Server with 32GB memory. The various parameters are listed in Table II:

TABLE II
NICE Simulation Parameters

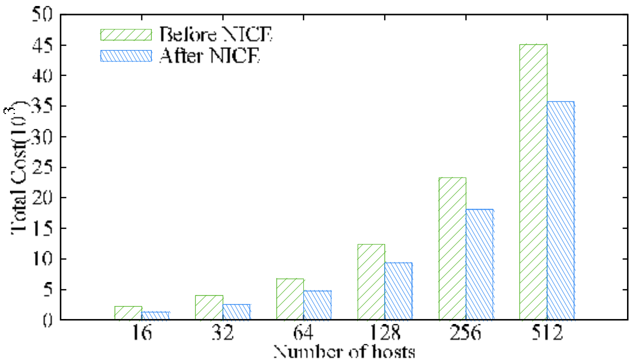
System Parameter	Settings
number of hosts	from 16 to 512
number of VMs	from 32 to 1024
Network Topology	Tree, VL2, FatTree, BCube
T / T_0 / P_h^{I} / P_h^{F} / P_h^{VM}	600s / 14s / 100w / 150w / 12w

B. Simulation Results

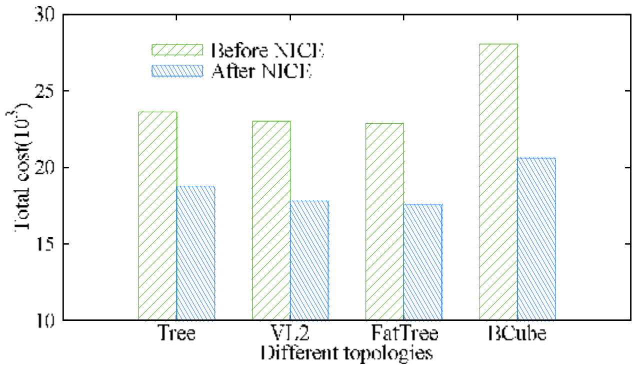
As shown in Section II, [2] [3] dealt with VM consolidation problem, but they didn't consider network performance improvement issue. [17] [18] [19] [20] [21] only tried to improve network energy efficiency while [13] [23] only dealt with VM placement problem, taking no consideration of VM consolidation. Since there is no other VM consolidation scheme jointly considering host energy consumption, network cost and migration cost at the same time, we show the efficiency of NICE through the comparison of total cost before and after NICE. The total cost is the sum of host energy consumption, network cost and migration cost. Before NICE, n VMs are randomly placed in m hosts.

We first evaluate the performance of NICE against different problem scales. We set the number of hosts to $2^i (i = 4,5,6,7,8,9)$ and the number of VMs twice of the number of hosts. The data center topology we evaluate here is multi-rooted tree topology. Fig. 2(a) shows the total cost before and after conducting NICE. We can see that NICE can reduce large amount of total cost against different problem scales.

Next, we evaluate the performance of NICE against different DCN topologies, *e.g.,* Tree, VL2, FatTree and BCube. Communication cost matrices F are calculated for different DCN topologies. We set the number of hosts to 128 and the number of VMs to 256. Fig. 2(b) shows the total cost of the four topologies before and after NICE. We can see

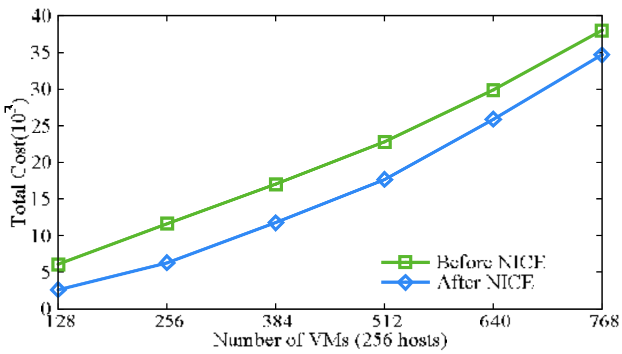


(a) Different problem scales

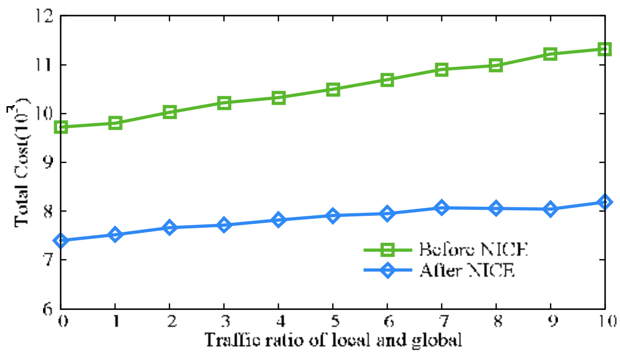


(b) Different DCN topologies

Fig. 2. Simulation results for NICE.



(a) Different density of VMs



(b) Different traffic ratios

Fig. 3. Simulation results for NICE.

that NICE works well for different DCN topologies. We can also see that before and after NICE, the total cost varies with different DCN topologies. These variations are mainly due to the differences of network topologies. We can see that total cost of BCube is the largest. This is because connecting the same number of hosts, BCube has more levels than the other three topologies, and this results in higher communication cost. In all, NICE can save about 20% total cost for different DCN topologies.

Fig. 3(a) shows the performance of NICE against different VM densities. We first set the number of hosts to 256 and then incrementally increase the number of VMs by half of the number of hosts. We can see that NICE still can save large amount of total cost as the density of VMs increases. But the percentage of total cost it saved slowly decreases. This is because as the number of VMs increases, utilization of data center increases, and this reduces the potential of saving energy through VM consolidation in data center. Therefore, when there are too many VMs running in the data center, we should not conduct NICE because this may do more harm than good.

In the end, we evaluate the performance of NICE against different mixture of global and local traffic model. We gradually set the percentage of local traffic rate in the total traffic rate from $0\%, \ldots, 100\%$ and evaluate the performance of NICE. Fig. 3(b) shows the simulation results and we can see that as the local traffic increases, the total cost increases. However, we can also see that as the local traffic increases, the total cost saving increases, too. This is because the more unbalanced the traffic is, the greater the potential of traffic cost saving is, and NICE can exploit this greater potential to save more traffic cost.

From the above results and analyses, we can conclude that NICE can solve CVC effectively under different scenarios.

VII. CONCLUSION

In this paper, we formulate *Combined VM Consolidation* (CVC) problem to reduce host energy consumption, traffic cost and migration cost in cloud data center. We analyze cost functions for host energy consumption and network cost thoroughly and formulate VM migration cost function by real-trace experiments in a testbed of 6 servers. We prove that CVC is a NP-complete problem, which is inappproximable by a factor of $3/2-\varepsilon$. Correspondingly, we propose an efficient heuristic scheme named NICE: *Network-aware VM ConsolIdation scheme for Energy Conservation in Data CEnter* to solve CVC. NICE consists of three subroutines: *FirstFit-Decr* algorithm for packing VMs into virtual hosts; *Kernighan-Lin* algorithm for adjusting virtual hosts; and *Greedy-Map* algorithm for mapping virtual hosts to physical hosts. Our simulation results validate the efficiency of NICE. In all, to the best of our knowledge, we are the first work jointly considering host energy conservation, network scalability and energy saving and migration cost of VM consolidation in cloud data center.

ACKNOWLEDGEMENT

This work has been supported in part by the National Natural Science Foundation of China (Grant number 61202024, 61133006, 61371084), China 973 project (2014CB340303, 2012CB316200), Shanghai Educational Development Foundation (Chenguang Grant No.12CG09), Shanghai Pujiang Program 13PJ1403900, and the Natural Science Foundation of Shanghai (Grant No.12ZR1445000). The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

REFERENCES

[1] L. Liu, H. Wang, X. Liu, X. Jin, W. He, Q. Wang, and Y. Chen. Greencloud: a new architecture for green data center. *International conference industry session on Autonomic computing and communications industry session*, pages 29–38, 2009.

[2] A. Verma, P. Ahuja, and A. Neogi. pmapper: power and migration cost aware application placement in virtualized systems. *Middleware*, pages 243–264, 2008.

[3] G. Jung, M. A Hiltunen, K. R Joshi, R. D Schlichting, and C. Pu. Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. *International Conference on Distributed Computing Systems (ICDCS)*, pages 62–73, 2010.

[4] Y. Zu, T. Huang, and Y. Zhu. An efficient power-aware resource scheduling strategy in virtualized datacenters. *International Conference on Parallel and Distributed Systems (ICPADS)*, pages 110–117, 2013.

[5] M. Lim, F. Rawson, T. Bletsch, and V. Freeh. Padd: Power aware domain distribution. *International Conference on Distributed Computing Systems (ICDCS), IEEE*, pages 239–247, 2009.

[6] P. Gao, A. R Curtis, B. Wong, and S. Keshav. It's not easy being green. *ACM SIGCOMM Computer Communication Review*, 42(4):211–222, 2012.

[7] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. Nguyen. Reducing electricity cost through virtual machine placement in high performance computing clouds. *International Conference for High Performance Computing, Networking, Storage and Analysis*, page 22, 2011.

[8] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. *ACM SIGCOMM Computer Communication Review*, 39(4):123–134, 2009.

[9] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. Dcell: a scalable and fault-tolerant network structure for data centers. *ACM SIGCOMM Computer Communication Review*, 38(4):75–86, 2008.

[10] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. Bcube: a high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Computer Communication Review*, 39(4):63–74, 2009.

[11] 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. *ACM SIGCOMM Computer Communication Review*, 39(4):51–62, 2009.

[12] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. *ACM SIGCOMM Computer Communication Review*, 39(4):39–50, 2009.

[13] X. Meng, V. Pappas, and L. Zhang. Improving the scalability of data center networks with traffic-aware virtual machine placement. *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–9, 2010.

[14] X. Li, J. Wu, S. Tang, and S. Lu. Let's stay together: Towards traffic aware virtual machine placement in data centers. *IEEE International Conference on Computer Communications (INFOCOM)*, 2014.

[15] F. Tso, K. Oikonomou, E. Kavvadia, and D. Pezaros. Scalable traffic-aware virtual machine management for cloud data centers. *International Conference on Distributed Computing Systems (ICDCS)*, pages 238–247, 2014.

[16] D. Abts, M. Marty, P. Wells, P. Klausler, and H. Liu. Energy proportional datacenter networks. *ACM SIGARCH Computer Architecture News*, 38:338–347, 2010.

[17] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. *IEEE International Conference on Computer Communications (INFOCOM)*, pages 457–465, 2008.

[18] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. Elastictree: saving energy in data center networks. *NSDI*, pages 17–17, 2010.

[19] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao. Carpo: Correlation-aware power optimization in data center networks. *IEEE International Conference on Computer Communications (INFOCOM)*, pages 1125–1133, 2012.

[20] V. Mann, A. Kumar, P. Dutta, and S. Kalyanaraman. Vmflow: leveraging vm mobility to reduce network power costs in data centers. *NETWORKING*, pages 198–211, 2011.

[21] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong. Vmplanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers. *Computer Networks*, 2012.

[22] Y. Shang, D. Li, and M. Xu. Energy-aware routing in data center network. *ACM SIGCOMM workshop on Green networking*, pages 1–8, 2010.

[23] D. Huang, D. Yang, H. Zhang, and L. Wu. Energy-aware virtual machine placement in data centers. *Global Communications Conference (GLOBECOM), IEEE*, pages 3243–3249, 2012.

[24] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan. Full-system power analysis and modeling for server environments. *Workshop on Modeling, Benchmarking, and Simulation*, pages 70–77, 2006.

[25] X. Fan, W. Weber, and Luiz A. Barroso. Power provisioning for a warehouse-sized computer. *ACM SIGARCH Computer Architecture News*, 35(2):13–23, 2007.

[26] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. *NSDI*, pages 273–286, 2005.

[27] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya. Cost of virtual machine live migration in clouds: A performance evaluation. *Cloud Computing*, pages 254–265, 2009.

[28] V. Vazirani. *Approximation Algorithms*. Springer, 2003.

[29] György Dosa. The tight bound of first fit decreasing bin-packing algorithm is ffd (i) ≤ 11/9opt (i)+ 6/9. *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, pages 1–11, 2007.

[30] BW. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 1970.