2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops

Virtual Machine Placement for Improving Energy Efficiency and Network Performance in IaaS Cloud

Jiankang Dong, Hongbo Wang, Xing Jin, Yangyang Li, Peng Zhang, Shiduan Cheng
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, 100876, China
Email: {dongjk, hbwang, jinxing6044, yyli, zhp, chsd}@bupt.edu.cn

***Abstract*— Under the premise of ensuring application performance, how to place virtual machines (VMs) on physical machines (PMs) to improve resource utilization and reduce energy consumption is one of the major concerns for cloud providers in IaaS cloud. The existing VM placement schemes are mostly to reduce energy consumption by optimizing utilization of physical server or network elements, but the issue of aggressive consolidation of VM is ignored, which may lead to network performance degradation. To address the issue, this paper proposes a VM placement scheme based on a new two-stage heuristic algorithm to optimize network performance and to reduce energy consumption of physical servers and network elements, so as to finally achieve the tradeoff between energy efficiency and network performance. The simulations show that our solution achieves good results.**

Keywords- IaaS Cloud, Virtual Machine Placement, Network Performance, Energy Optimization

I. INTRODUCTION

Nowadays most of physical servers in cloud data center use virtualization technology [1]. Based on the service level agreement (SLA) with cloud providers, the tenants order a group of virtual machines(VM) which are placed in different hosts and enable the communication between each other; each VM requires a certain amount of resources, such as CPU, memory, storage, bandwidth to maintain application performance isolation and security. Moreover, virtualization runs multiple virtual servers on the same physical machine (PM), which is helpful to improve resource utilization and then to reduce power costs. Correspondingly, virtualization can also help cloud providers achieve orderly and on-demand resource deployment, which provides an effective solution to the flexible resource management and low energy consumption.

For public cloud with virtualization, one of its major services is infrastructure as a service (IaaS), such as Amazon EC2[2]. Tenants pay to rent VM, based on SLA; cloud providers take advantage of VM's flexible placement on PM to optimize resources allocation so as to meet tenants' demands. Since different resource utilization is caused by different mapping between VMs and PMs, for cloud providers, the main issue should be how to place multiple VMs demanded by tenants onto physical servers so as to minimize the number of active physical resources and reduce energy consumption, and correspondingly, operation and management costs will be reduced. Nowadays, VM placement is becoming a hot issue.

For VM placement, one major direction is to reduce energy consumption. Some studies [3-6] propose to reduce energy consumption by improving physical server utilization and reducing the number of hosts. However, the optimization of network resources is less concerned, and the studies do not consider the impact of network topology and current communication traffic. While as the scarce resources in cloud data centers, network resources actually have a direct impact on application performance [7]. Other studies [8-10] are either to improve network utilization or to optimize the route to reduce the number of network elements. However, excessive resource consolidation caused by optimizing the energy efficiency of PM or network devices may affect the application performance and increase SLA violation rate, especially when network traffic is aggregated, congestion problems will be easily caused by hot spots.

In view of this, cloud providers have to optimize the energy consumption of physical servers and network elements by consolidating VMs on less physical servers and optimizing network traffic to reduce the number of network elements so as to save energy. Moreover, network congestion by excessive consolidation and the tradeoff between energy efficiency and network performance also have to be considered,

The optimization of PM resources by VM placement is abstracted as a bin packing problem (BPP) [11] to minimize the number of active physical servers, while the optimization of network resources by using network topology and communication traffic is abstracted as quadratic assignment problem (QAP) [12] in order to minimize the total network traffic, the less traffic in the network, the fewer the number of the active network elements (switches, links, etc.). Meanwhile, it is necessary to optimize the maximum link utilization (MLU) in order to avoid network congestion and ensure the application performance, so we attempt to not only reduce the number of PMs and network elements for energy-saving but also ensure the application performance to avoid the congestion, which is a classic multi-objective optimization issue [13].

Therefore, our paper presents an optimal VM placement scheme on the basis of multiple resources constraints. When meeting the constraints of PM resources (CPU, memory, etc.) and network link capacity, cross-optimizing VMs placed on PMs for avoiding network congestion can maximize the resources utilization of PMs and network elements to reduce SLA violation, which allows the idle physical resources in a

978-0-7695-5023-7/13 $25.00 © 2013 IEEE
DOI 10.1109/ICDCSW.2013.48

sleeping state so as to finally reduce energy consumption in IaaS cloud.

Based on the above, we propose a novel two-stage greedy algorithm to solve. Firstly, we combine minimum cut hierarchical clustering with best fit (BF) algorithm. With hierarchical clustering, minimum cut algorithm enables the VMs with large traffic to be placed on the same PM or the same access switch, so finally to reduce the network traffic; and then according to the clustering results, we apply BF to minimize the number of PMs. Secondly, we apply local search algorithm to minimize the maximum link utilization (MLU) and to keep a balanced distribution of network traffic to reduce link congestion. Certainly, our algorithm will be able to adapt to the heterogeneous feature of PM and VM. The simulation results show that our novel scheme achieves better results compared with best fit decreasing (BFD) and Random algorithms.

Our paper is organized as follows: Section II presents the related work. VM placement is described and modeled in Section III. Section IV puts forward VM placement algorithm. The simulation is shown in Section V. Section VI concludes the paper.

II. RELATED WORK

There are two focuses on VM placement problem for reducing energy consumption. One is to consider how to place VM in accordance with the physical server resources. Verma *et al.* [3] dynamically re-adjust server's location and consider the cost of application migration and energy, with a simple algorithm; it shows that dynamic migration technology realizes low energy cost. Bobroff *et al.* [4] adopt prediction techniques while minimizing the number of active PMs, and present mechanism for dynamic migration of VMs based on a workload forecast. Cardosa *et al.* [5] reset max, min, share and other VM parameters to meet users' demands and to provide a new PM resource allocation method; it consolidates multiple VM onto PM to improve resource utilization and reduce power consumption. Wang *et al.* [6] consider the consolidation of VM bandwidth with PM bandwidth as a stochastic packing NP-hard problem, it shows certain size of VM is loaded onto a PM with a probability distribution, and the goal of optimization is to minimize PM number. However, [3-6] only consider PM optimization, and ignore other resource.

The PM optimizing schemes above either consider CPU constraints [3] or PM bandwidth constraints [6], and they neglect network topology and VM communication traffic. The other type [8-10] is to consider how to place VM to optimize network resources. Meng *et al.* [10] are to improve the network scalability in data center network with a traffic-aware VM placement scheme. By optimizing VM's location in the PM host, the traffic between VMs is related to the network physical distance, and VMs with large traffic can be placed on PMs nearby to reduce the total network traffic. Mann *et al.* [8] are to reduce energy consumption by VM migration technology and network routing optimization. Such solutions only assume to meet physical servers needs. Fang *et al.* [9] are to optimize both VM placement and traffic flow routing so as to turn off as many unneeded network elements as possible for power saving, and [8-10] only optimize network resources and neglect physical server resource optimization.

Currently, some studies consider application performance when placing VMs, and the similar studies are in [14-16]. Jiang *et al.* [14] attempt to improve physical node utilization with VM placement, and to optimize the network link utilization by changing flow routing, but they do not optimize total traffic in data center network, so their objective is different from our scheme. Gupta *et al.* [15] address application-aware allocation of n VM instances to physical hosts from a single pool, which meets SLA requirements, and at the same time, improves the utilization of hardware resources. However, they do not consider network MLU in data center. Beloglazov *et al.* [16] propose a novel adaptive heuristics for dynamic consolidation of VMs based on an analysis of historical data from the resource usage by VMs. Although this proposed algorithms significantly reduces the energy consumption while ensuring a high level of adherence to the SLA, it does not optimize network resource, so it is also different from our objective.

III. Design and Model

A. Energy model

1) Energy consumption of physical servers

The optimization of physical server is abstracted as a multi-dimensional resource constraints packing problem [11], and our objective is to minimize the number of physical servers.

$$
\begin{array}{lll}
\min & E_{ser} = \sum_{m=1}^{M} Y_m \cdot E_{ser}^{m} & 1.1 \\
\mathrm{s.t.} & \sum_{i=1}^{N_m} X_{i,m} \cdot \vec{S}_i \preccurlyeq Y_m \cdot \vec{H}_m & 1.2 \\
 & \sum_{m=1}^{M} X_{i,m} = 1 & 1.3
\end{array}
\qquad (1)
$$

Let N_m denote the number of VMs on PM *m*; E_{ser}^{m} denote the energy consumption of PM *m*. Equation 1.2 means the total capacity of multiple VMs on the same PM is less than the capacity of the PM. Equation 1.3 means that any VM can only be placed on a single PM.

2) Energy consumption of network elements

For the optimization of the network resources, our objective is to minimize traffic in IaaS cloud, and we abstract this problem as QAP [12]. The large traffic between VMs is converged onto the same PM, or on the same switch. If the total communication traffic in hierarchy topology is smaller, then the number of network elements will be reduced, and the other idle network elements will be in a sleeping state, so the power consumption will be reduced. Our proposal not only can save the energy of network elements, but also improve the link utilization of the core tier.

$$
\min \quad E_{net} = \sum_{i=1}^{N_{swi}} E_{swi}^{i} + \sum_{i=1}^{N_{link}} E_{link}^{i} \qquad (2)
$$

Let E_{swi}^{i} represent the energy consumption of switch *I*; N_{swi} means the number of active switch. E_{link}^{i} is the power consumption of link *i*, N_{link} is the number of active links,

TABLE I. KEY NOTATION AND ITS MEANING

Symbol	Description
M	Number of PMs , indexed by $m = 1,..,M$
N	Number of VMs , indexed by $i = 1..N$
$\vec{H}_m$	d dimensional resource vector of PM m ,its value $\{\mathrm{H}_{\mathrm{m,1}}, \mathrm{H}_{\mathrm{m,2}}, \dots, \mathrm{H}_{\mathrm{m,d}}\}$, d is the number of resource types
$\vec{S}_i$	D-dimensional resource vector of VM i its value $\{\mathrm{S}_{\mathrm{i,1}},\mathrm{S}_{\mathrm{i,2}},\dots,\mathrm{S}_{\mathrm{i,d}}\}$
Y_m	Binary variable,1 indicates PM m is in the activation status ; 0 indicates that PM m is sleep
$X_{i,m}$	Binary variable , 1 indicates VM i is placed on the PM m , whereas 0
E_{ser}	Energy consumption of physical servers
E_{net}	Energy consumption of network elements
E_{DC}	Energy consumption of data center
$cost_{net}$	Maximum link utilization

and the calculation of N_{swi} and N_{link} by using Elastictree method [17]. For each traffic flow, the greedy bin-packer evaluates possible paths and chooses the leftmost route with sufficient free capacity in a certain hierarchy layer of a structured topology, such as a fat tree. Within a layer, paths are chosen in a deterministic left-to-right order, other than a random order, which would evenly spread flows. When all traffic flows have been assigned, the algorithm returns the active network subset (set of switches and links traversed by some traffic flow) plus each flow path.

3) Energy consumption of data center

$$E_{DC} = E_{ser} + E_{net} \tag{3}$$

We model the energy consumption of PM and network elements in cloud data center. The energy consumption of PM includes CPU, memory, storage, and network interface; and the energy consumption of network elements includes switches, links etc..

B. Network Performance

From the view of traffic engineering, minimizing the MLU is the main goal of the optimization.

$f_{s,t}^{i,j}$ represents the traffic which is assigned to the link (s,t) from traffic demand of VM i and VM j.Network link utilization $l_{s,t}$ is expressed as:

$$l_{s,t} = \frac{\sum_k f_{s,t}^{i,j}}{c_{s,t}} \tag{4}$$

To let l be as small as possible, the problem can be expressed as:

$$\begin{aligned} \min \quad & cost_{net} = max\,\{l_{s,t}\} \\ .\ \mathrm{s.t.} \quad & \textstyle\sum_j f_{s,t}^{i,j} - f_{t,s}^{i,j} = 0 \quad X_{i,s} = 1,\ X_{j,t} = 1 \\ & \textstyle\sum_j f_{s,t}^{i,j} - f_{t,s}^{i,j} = f_k \quad X_{i,s} = 1 \\ & \textstyle\sum_k f_{s,t}^{i,j} \le c_{s,t} \\ & f_{s,t}^{i,j} \ge 0 \end{aligned} \tag{5}$$

Our goal is to minimize MLU and total energy consumption. This is a multi-objective optimization problem.

$$\min \qquad g = \ E_{DC} + r \cdot cost_{net} \tag{6}$$

IV. VM PLACEMENT ALGORITHM

We propose a two-stage heuristic algorithm. Firstly, if there is no network congestion, the optimization of energy consumption is considered as a priority, and we combine minimum cut hierarchical clustering with BFD algorithm to solve the multi-objective optimization problem. With Hierarchical clustering, minimum cut algorithm enables the related VMs to cluster together to minimize the total network traffic; According to the clustering results, we apply BF to reduce the energy consumption caused by PMs. Secondly, In case of network congestion, we minimize the MLU to optimize network performance.

A. Hierarchical Clustering Algorithm based on Minimum Cut

Most of the data centers are three-tier architecture [18]. For network topology and network traffic between the VMs, the VMs with the large traffic should be placed on the same PM or with the same switch to ensure the application performance and reduce the number of the network equipments. We solve QAP with hierarchical clustering algorithm based on the traffic between VMs.

Let G=(V, E) be a connected undirected graph, where V is a collection of VMs, E is the traffic between VMs. Hierarchical clustering is achieved by using the minimum cuts in graph G. Given a node set Q⊆ V, $\delta(\mathrm{Q})$ denotes the set of all edges with one end in Q and the other end in V\Q. A cut consists of all edges that have one end in Q and the other end in V\Q, where Q is a node set such that $\mathrm{Q} \neq \emptyset$ and $\mathrm{Q} \neq \mathrm{V}$; that cut is denoted $(\mathrm{Q}, \mathrm{V}\backslash\mathrm{Q})$.

Let every edge $ij \in \mathrm{E}$ be assigned a nonnegative capacity $\mathrm{c}(ij)$.The capacity of a cut is defined as the sum of the edge in it, i.e., $\mathrm{c}(\mathrm{Q}, \mathrm{V}\backslash\mathrm{Q}) = \sum_{ij\in\delta(\mathrm{Q})} \mathrm{c}(ij)$ .The minimum cut problem is to find a cut in G with smallest capacity.

The minimum cut of G is expressed by binary tree T(V). For a binary tree T(V), left subtree TL is the node in Q, its weight is the sum of the edge in Q, $\mathrm{W}(\mathrm{TL}) = \sum_{ij\in\mathrm{Q}} \mathrm{c}(ij)$, right subtree TR is the node of V\Q, the weight is the sum of the edge in V\Q, $\mathrm{W}(\mathrm{TR}) = \ \sum_{ij\in\mathrm{V}\backslash\mathrm{Q}} \mathrm{c}(ij)$, if $\mathrm{W}(\mathrm{TL}) < \mathrm{W}(\mathrm{TR})$, swap the left sub- tree TL with right subtree TR, which means the VMs traffic of left subtree TL is larger than that of right subtree. Leaf nodes of a binary tree T (V) represent only one VM, the branches mean a collection of VMs after clustering. This algorithm is defined as MC-BT. The algorithm is described in Algorithm 1.

B. VM Placement Algorithm based on BF

T(V) is obtained from MC-BT. Preorder tree traversal results in a vector called VMlist, which consists of the successive leaf nodes of tree T. We place all VMs nodes using VMlist. As can be seen in the previous discussions, Algorithms 1 MC-BT algorithm

```
Input: Graph G=(V,E)
Output : Binary Tree T(V)
Initial cut S
Initial Binary Tree T
While G has more than one node do
      Pick two distinct node s and t
      Compute a minimum capacity cut $\delta(S')$ separating s and t
      If $c(S', V\backslash S') < C$
            $C \leftarrow c(S', V\backslash S')$ and $S \leftarrow S'$;
      Endif
      Left subtree $TL \leftarrow G_s(V)$,compute $W(TL)$
      Right subtree $TR \leftarrow G_t(V)$, compute $W(TR)$
      if $W(TL) < W(TR)$
            $TL \leftrightarrow TR; G_s \leftrightarrow G_t$
      Endif
      Replace G by $G_s$ and $G_t$
Endwhile
Output T(V)
```

VM neighbors have larger traffic between each other in VMlist. The larger distance between a pair of VM nodes is, the smaller traffic between them is.

By BF algorithm, we place different sizes of VM nodes in VMlist into the corresponding PMs. We place VMs in VMlist sequence. For a new VM, we search from the first PM until finding the one which best matches this new arrival. Only when all active PMs cannot accommodate this VM, a new PM can be allocated.

The time complex of BF is $O(n^2)$, and space complexity is $O(n)$. This algorithm is defined as Best Fit with hierarchical clustering algorithm (BF-HC). The algorithm is described as algorithm 2.

C. Local Search Algorithm

We can calculate the current network by the traffic matrix between VMs. When currently the network congestion does not occur, we only use BF-HC algorithm to optimize the energy consumption in IaaS cloud; when the network hot link occurs, on the basis of BF-HC, local search algorithm is applied to optimize the link utilization and to avoid network congestion, which is defined as BF-HC-LS algorithm.

Algorithms 3 BF-HC-LS algorithm

```
Input: X ( current VM placement ), A ( traffic matrix ), TP ( network
topology ) ,Nmax( maximum iterations)
Output : matrix $X_{best}$
Initial X, A, TP
While s = 1 to Nmax do
      Select the VM with the largest congestion , random exchange
            with the VM under the neighboring switch
      Get new X'
      Compute HLN'
      If HLN '<$HLN_{best}$
            $X_{best} \leftarrow$ X 'and $HLN_{best} \leftarrow$ HLN';
      Else
            A certain probability to accept the X ' , the HLN'
      Endif
Endwhile
```

Algorithms 2 BF-HC algorithm

```
Input: physical resource vector PMlist, binary tree T(V).
Output : matrix X of the mapping between VM and PM
initialize VM vector group VMlist (preorder tree traversal T(V) leaf
   nodes sequentially into the VMlist)
Foreach $VM_i$ in VMlist do
   Foreach $PM_m$ in PMlist do
         If (isAllocable($VM_i$, $PM_m$) and $PM_m$.spare < $PM_{best}$.spare)
            Best← m
         Endif
   Endfor
   $X_{i,m} \leftarrow 1$, Allocation($VM_i$, $PM_{best}$)
Endfor
Output X
```

We choose the VM which generates the largest congested traffic, and then randomly exchange this VM with the VM under the neighboring switch to calculate the objective function: hotspot link number (HLN). If the objective function value decreases, then accept this exchange; if not, accept according to a certain probability; repeat the number of iterations by return. The algorithm is described as algorithm 3.

V. EVALUATION

A. Simulation Setup

We use C++ to develop our BF-HC and BF-HC-LS. The most common approximation algorithms to solve the BPP are next fit decreasing (NFD), first fit decreasing (FFD) and BFD [19] , but we select BFD for its better effects, together with Random algorithm, to compare with our BF-HC-LS .

Data center is a common hierarchical topology, such as multi-rooted tree [18], VL2 [20], fat-tree [21] etc, so we choose the most common fat-tree topology, and routing policy is Elastictree [17].

There are three basic inputs in our simulation: VM resource vector group, PM resources vector group and traffic matrix between the VMs. For VM resource vector group, Amazon EC2 [22] provides a flexible choice to meet different application needs, so we select the VM size and configuration provided by Amazon EC2. For VM traffic matrix, our experiments take the traffic patterns in [10, 23].

B. Simulation Result

1) No Hotspot Pattern

Calculate network link utilization with the traffic matrix between the VMs. If no hotspot occurs, and then take into account the optimization of energy consumption.

With the different scale of the VMs in the data center, a group of VMs 100, 200 and 300 are selected. They have different CPU size, memory capacity, storage capacity and network bandwidth. We apply Random, BFD and BF-HC to these three groups of VM to calculate the required number of PMs, switches and energy consumed.

Figure 1 shows the number of PMs required for various types of algorithms, BFD and BF-HC require less PMs than random algorithms, and BF-HC requires almost the same number of PMs as BFD.

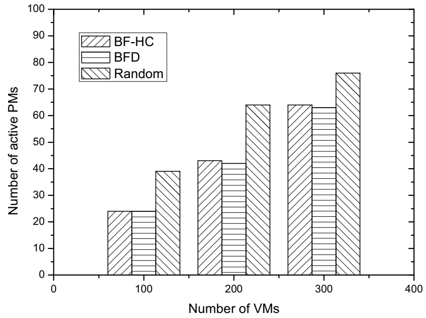

Figure 1. PM number comparison in no hotspot pattern

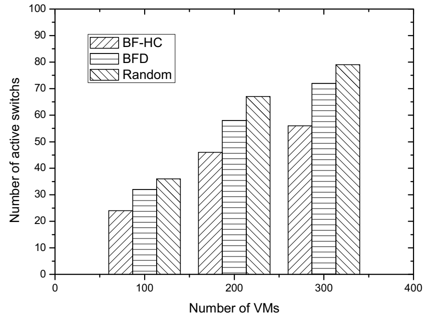

Figure 2. Switch number comparison in no hotspot pattern

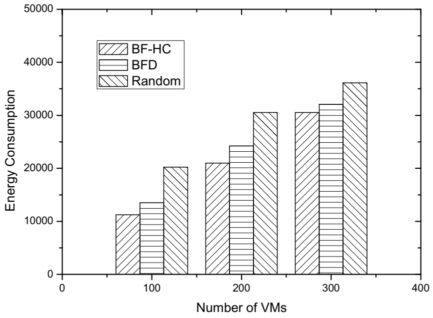

Figure 3. Energy consumption comparison in no hotspot pattern

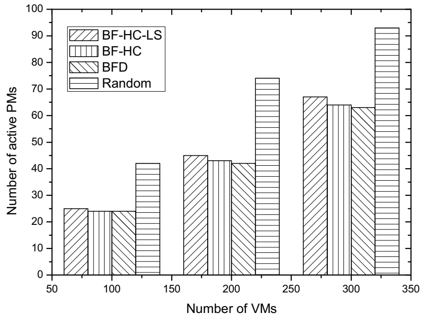

Figure 4. PM number comparison in hotspot pattern

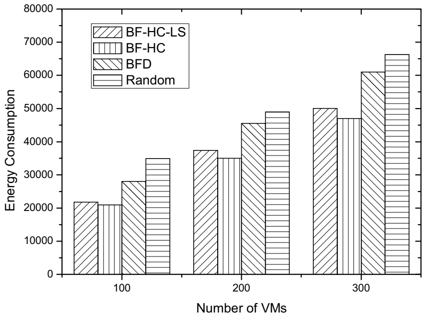

Figure 5. Energy consumption comparison in hotspot pattern

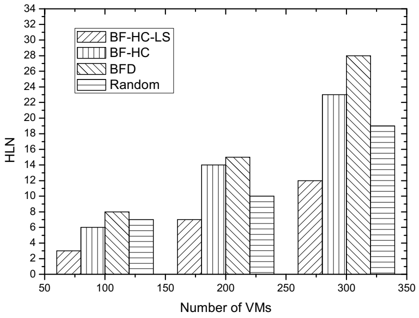

Figure 6. HLN comparison in hotspot pattern

Figure 2 shows the number of active switches by comparing BF-HC with BFD and Random algorithms in fat-tree topology. The number of active switches by BF-HC is averagely decreased by 28% than by BFD, and decreased by 45% than by Random algorithms.

Figure 3 shows the energy consumption of these three algorithms in the fat-tree topology. The energy consumption is calculated on the basis of equation (3), and it can be seen that the BF-HC energy consumption is the smallest.

2) Hotspot Pattern

If hotspot occurs, consider the optimization of performance as a priority, and then take into account the optimization of energy consumption.

Figure 4,5,6 respectively show the changes of PM number, energy consumption and hotspot link number by comparing BF-HC-LS with BF-HC, BFD and Random algorithms in the fat-tree topology. HLN by BF-HC, BFD and Random algorithm shows slight difference, and HLN by BF-HC-LS is the lowest.

Compared with BF-HC, the energy consumption by BF-HC-LS shows little change, but HLN by BF-HC-LS actually decreases a lot.

VI. CONCLUSION AND FUTURE WORK

VM placement problem in IaaS cloud is a hot topic nowadays. With the constraints of the resources of PM and network link capacity, the objective of our VM placement scheme is to optimize utilization of the PM and network elements so as to save energy, and to achieve the tradeoff between energy efficiency and network performance so as to reduce network congestion. Based on this objective, we propose a two-stage heuristic algorithm. Firstly, if no congestion occurs, the optimization of energy consumption is considered as a priority, and we combine minimum cut hierarchical clustering with BFD algorithm to optimize the PM and the network elements so as to save energy. Secondly, if network congestion occurs, the optimization of network performance is considered as a priority, and we apply local search algorithm to minimize the MLU and reduce congestion link. Simulation results show that compared with BFD and Random algorithms, our solution can optimize the distribution of network traffic and reduce network congestion with slight change in energy consumption.

The main objective in our paper is to achieve the tradeoff between energy efficiency and network performance. However, as the workload changes, and then the size of a VM and the mappings between VMs and PMs also need to be changed, so it is very necessary to consider the problem of VM migration. Therefore, our next research direction will concentrate on how to minimize the cost of the VMs dynamic migration without affecting application performance.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61002011); the 863 Program of China (No.s 2013AA013303, 2011AA01A102); the Open Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE-2009KF-2-08), Beijing University of Aeronautics and Astronautics; the 973 Program of China (No. 2009CB320505).

REFERENCES

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," ACM SIGOPS Operating Systems Review, vol. 37, 2003, pp. 164-177.

[2] "Amazon EC2," http://aws.amazon.com/ec2/.

[3] A. Verma, P. Ahuja and A. Neogi, "pMapper: power and migration cost aware application placement in virtualized systems," in Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, 2008, pp. 243-264.

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

[5] M. Cardosa, M. R. Korupolu and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on, 2009, pp. 327-334.

[6] M. Wang, X. Meng and L. Zhang, "Consolidating virtual machines with dynamic bandwidth demand in data centers," in INFOCOM, 2011 Proceedings IEEE, 2011, pp. 71-75.

[7] M. Al-Fares, A. Loukissas and A. Vahdat, "A scalable, commodity data center network architecture," in ACM SIGCOMM Computer Communication Review, 2008, pp. 63-74.

[8] V. Mann, A. Kumar, P. Dutta, and S. Kalyanaraman, "VMFlow: leveraging VM mobility to reduce network power costs in data centers," NETWORKING 2011, pp. 198-211.

[9] 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.

[10] X. Meng, V. Pappas and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1-9.

[11] E. G. Coffman Jr, M. R. Garey and D. S. Johnson, "Approximation algorithms for bin packing: A survey," Approximation Algorithms for NP-Hard Problems, 1996, pp. 46--93.

[12] G. Woeginger, "Exact algorithms for NP-hard problems: A survey," Combinatorial Optimization—Eureka, You Shrink!, Springer Berlin Heidelberg, 2003, pp. 185-207.

[13] K. Deb, "Multi-objective optimization," Multi-objective optimization using evolutionary algorithms, pp. 13-46, 2001.

[14] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, "Joint VM placement and routing for data center traffic engineering," in INFOCOM, 2012 Proceedings IEEE, 2012, pp. 2876-2880.

[15] A. Gupta, L. V. Kalé, D. Milojicic, P. Faraboschi, and S. M. Balle, "HPC-Aware VM Placement in Infrastructure Clouds," in IEEE Intl. Conf. on Cloud Engineering IC2E, 2013.

[16] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers," Concurrency and Computation: Practice and Experience, vol. 24, pp. 2012,1397-1420.

[17] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: saving energy in data center networks," in Proceedings of the 7th USENIX conference on Networked systems design and implementation, 2010, p. 17-17.

[18] M. Al-Fares, A. Loukissas and A. Vahdat, "A scalable, commodity data center network architecture," in ACM SIGCOMM Computer Communication Review, 2008, pp. 63-74.

[19] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, "Worst-case performance bounds for simple one-dimensional packing algorithms," SIAM Journal on Computing, vol. 3, 1974, p. 299.

[20] 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 ACM SIGCOMM Computer Communication Review, 2009, pp. 51-62.

[21] 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," in ACM SIGCOMM Computer Communication Review, 2009, pp. 39-50.

[22] "Amazon EC2 Instance Types ", http://aws.amazon.com/ec2/instance-types/.

[23] T.Benson, A. Akella and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in Proceedings of the 10th annual conference on Internet measurement, 2010, pp. 267-280.