Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

June 2014, 21(3): 62–70
www.sciencedirect.com/science/journal/10058885

The Journal of China Universities of Posts and Telecommunications

http://jcupt.xsw.bupt.cn

Virtual machine placement optimizing to improve network performance in cloud data centers

DONG Jian-kang (✉), WANG Hong-bo, LI Yang-yang, CHENG Shi-duan

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract

With the wide application of virtualization technology in cloud data centers, how to effectively place virtual machine (VM) is becoming a major issue for cloud providers. The existing virtual machine placement (VMP) solutions are mainly to optimize server resources. However, they pay little consideration on network resources optimization, and they do not concern the impact of the network topology and the current network traffic. A multi-resource constraints VMP scheme is proposed. Firstly, the authors attempt to reduce the total communication traffic in the data center network, which is abstracted as a quadratic assignment problem; and then aim at optimizing network maximum link utilization (MLU). On the condition of slight variation of the total traffic, minimizing MLU can balance network traffic distribution and reduce network congestion hotspots, a classic combinatorial optimization problem as well as NP-hard problem. Ant colony optimization and 2-opt local search are combined to solve the problem. Simulation shows that MLU is decreased by 20%, and the number of hot links is decreased by 37%.

Keywords cloud computing, data center network, virtual machine placement, traffic engineering, network performance

1 Introduction

Most of physical servers in cloud data center use virtualization technology [1–2]. Based on service level agreement (SLA) with cloud providers, the tenants order a group of virtual machines (VMs) which are placed in different hosts and allow communications from each other; each VM requires a certain amount of resources, such as central processing unit (CPU), memory and uplink/downlink bandwidth etc. to maintain the application performance isolation and security. Moreover, virtualization technology enables multiple virtual servers to run on the same physical machine (PM), which is helpful to improve resource utilization and then to reduce energy consumption. Therefore, virtualization can help cloud managers achieve orderly and on-demand resource deployment, which provides an effective solution to the flexible resource management.

For public cloud with virtualization, one of its major services is infrastructure as a service (IaaS), such as Amazon EC2 [3]. Tenants pay is to rent VM, and based on SLA, cloud providers take advantage of VM’s flexible placement on PM to optimize resources allocation so as to meet the tenants’ demands. Since different resource utilization is caused by different mappings between VMs and PMs, so for cloud providers, the main issue should be how to place multiple VMs demanded by tenants onto physical servers so as to achieve workload balance, optimize the resource utilization and avoid service performance degradation. From that on, the authors define the problem as VMP, which is right now becoming a hot issue in the current cloud computing research.

On the issue of VMP recently, researches mostly concentrate on the constraints of PM, such as CPU, memory and storage etc., but how to optimize network resources is with less concerned. For instance, recent studies [4–6] were proposed to reduce hosts number with virtualization to optimize energy consumption, Singh et al. [7] considered cloud data centers with server and storage virtualization facilities, and strived to increase load balancing at multiple layers. But they did not consider the

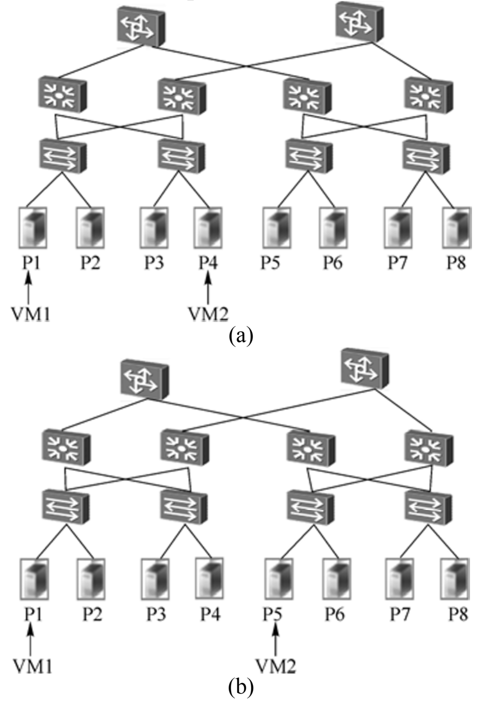
Received date: 21-10-2013
Corresponding author: DONG Jian-kang, E-mail: dongjk@bupt.edu.cn
DOI: 10.1016/S1005-8885(14)60302-2

impact of the network topology and the current network traffic. In addition, as the scarce resources in the data center, network resources are directly affect the application performance [8]. VMP not only can change the position of the VMs, which will correspondingly change the PMs, but also change the traffic sender and receiver between VMs, so that the layout of the network traffic can be changed. Different VMs on PMs may cause different mappings, which continuously cause different traffic distributions and network link utilization. Therefore, the authors mainly apply VMP to optimize the network traffic distribution and improve the network performance in data centers.

The latest studies such as in Refs. [9–10] considered the network resource in VMP. In Ref. [9], the authors focused on the problem of network-aware with the goal of reducing the aggregate traffic in data center network (DCN), but it did not consider link capacity constraints, which may lead to the congested links in DCN. From point of view of traffic engineering, the MLU is one of the standards to measure the network performance. In Ref. [10], the authors mainly proposed VMP to be applied to solve the network congestion. Minimizing MLU is its main goal of the optimization, the authors of Ref. [10] assumed that more localized traffic means that greater percentage of data is exchanged within a rack or nearby racks, and traffic localization shares similar objective with the MLU goal. However, this proposal, on one hand, can reduce the congestion of the core layer in DCN, on the other hand, it will increase the risk of congestion between aggregation layer and access layer. Generally, the smaller the total traffic in the network, the more possibly MLU will be reduced, but the goals are not the same.

As can be seen from Fig. 1, a communication traffic is assumed between VM1 and VM2, VM1 is placed on P1 and VM2 is on the other PM. It is shown that the network performance is different, which causes the different total communication traffic and network link utilization. In Fig. 1(a), VM2 is placed on P4, the total communication traffic is less than that of P5 in Fig. 1(b). Obviously, MLU values in Fig. 1(a) and Fig. 1(b) are remarkably different.

In view of the above, a multi-resourced constraints scheme is presented. When meeting the constraints of PM resources (CPU, memory, etc.), the different mappings of VMs and PMs, the authors firstly try to minimize the total network traffic, and puts the VM pairs with large traffic on the same PM or same switch to reduce communication traffic in network and to improve the network scalability; and then try to minimize MLU so as to reduce the network congestion. Thus, the main aim in the article is both to optimize MLU and total network traffic in order to effectively improve the network performance. During modeling, minimizing total communication traffic was thought as a quadratic assignment problem (QAP) [11], which is not only a classic combinatorial optimization problem but also a NP-hard problem [12]. From point of view of traffic engineering, to minimize MLU is a NP-hard problem [13]. A proposal is given to solve the combinatorial optimization by ant colony optimization (ACO) joint with 2-opt local search, and compared with clustering algorithm, local search (LS) and simulated annealing (SA), the proposal obtains better optimization results.



(a)

(b)

Fig. 1 The impacts on network link by VMP

The main contribution of this article is summarized as following:

1) Address a VMP scheme based on multi-constraints in order to optimize total traffic and MLU.

2) Combine ACO with 2-opt to solve.

3) Run the algorithm on different topologies, such as Tree [8], VL2 [14], Fat-Tree [15] and analyze the fitness respectively.

The article is organized as following. Sect. 2 is the background and motivation. Sect. 3 is the design and model. Sect. 4 is the detailed algorithm. Sect. 5 provides simulations, which validates the running time and correctness of our algorithms. Sect. 6 is the related work, Sect. 7 is the conclusion and the point of future work.

2 Background and motivation

Cloud data center is a widely used platform for variety of important business, such as MapReduce, web search, social network etc. The mass deployment of these applications causes a bottleneck of the network resources, and the network resources become scarce resources [16]. Although the network in data center is with characteristics of high bandwidth and low latency, research shows that congestions were also emerged [14,16–17], especially when aggregation layer switches simultaneously connect with multiple core layer switches, there usually occurs the bandwidth convergence rate [18], the elephant flow collision in equal-cost multi-path (ECMP) [19], TCP-Incast etc. Refs. [14,16–17] show that frequent congestion leads to throughput decrease, packet loss and latency increase, which correspondingly cause the degradation of business application performance. Therefore, network congestion becomes a challenge in cloud computing.

Multi-rooted tree, fat-tree, VL2 and other topologies are used in the data center for their multi-paths fault-tolerance and traffic load balance. Since the conventional minimal spanning tree algorithms are outdated, ECMP [19] is applied randomly to assign different flows to different paths. However, ECMP only guarantees the traffic load balancing in static situation, and load imbalance will occur in DCN situation, because flow size and duration has great gap in DCN, the static hash will assign the two large flows to the same path, which will cause path congestions, while the other path is light-loaded.

In order to solve the uneven traffic load caused by static random distribution, the dynamic scheduling was proposed in Refs. [20–21] according to the flow by using OpenFlow [22] to achieve fine-grained traffic engineering. Al-Fares et al. [20] proposed an improved routing algorithm, which is applied to a dynamic flow scheduling system with multi-root tree topology, and readjusts the flow in the link to be uniformly and evenly is distributed so as to improve the utilization of the switch and avoid the link blocking. Benson et al. [21] focused on network routing variation, mice flow takes ECMP protocol and the elephant flow takes dynamic routing to change the existing switch or router strategy. These studies mainly take centralized scheduling strategies to optimize traffic and to avoid link congestion. However, such solutions demand the concentration of controller and make flow control become more complex.

In the virtualized data center, the network traffic between VMs readjusted by VM placement/migration can also be seen as the traffic engineering. The traditional methods in traffic engineering optimize routing protocols to bypass the congested links and maintain the link balance by minimizing MLU. In addition, the traditional traffic engineering aims at the traffic matrix which can be measured, and the sending and receiving ends are fixed; while VM placement can change the position of VM, that is, change its corresponding physical server, and then the sending or receiving ends. The traffic matrix can all be changed to achieve the optimization of the network traffic in DCN. Therefore, the main objective in the article is to apply VMP to improve the network performance and reduce the hot links in cloud data center.

3 Design and model

The article tries to achieve two objectives by applying VMP.

1) By minimizing the total traffic using VMP in DCN, it can readjust the traffic layout between VMs, and let the VMs with large traffic be placed on the same PM or on the same switch.

2) The issue of minimizing MLU is also considered, which allows the network traffic to be allocated evenly and avoid congestion hotspots.

Table 1 Key notations and their meaning

Symbol	Description
n_{p}	Number of PMs, indexed by $m=1,2,...,n_{\mathrm{p}}$
n_{v}	Number of VMs, indexed by $i=1,2,...,n_{\mathrm{v}}$
$\pi(i)$	Mapping function , PM on which VM *i* is placed
e_i	The traffic between VM *i* and external communications of data center
l_{st}	Network link utilization
x_{st}^{ij}	The traffic which is assigned to the link (s,t) from traffic demand of VM *i* and VM *j*
$\boldsymbol{A}$	Communication traffic matrix, (a_{ij}) is the traffic between VM *i* and VM *j*
$\boldsymbol{B}$	Communication cost matrix, The communication cost is equivalent to the number switch that the traffic between PMs traverse.

To estimate the traffic matrix based on users demands and network topology, then to decide which PM can host VM. The traffic statistics depends on the hypervisors of VM or switch. The basic inputs are network topology, link capacity, traffic routing and traffic demand, it doesn't need to place the traffic demands to the appropriate link so as to get a more balanced traffic distribution.

3.1 Optimization of total traffic

Given two matrices: traffic matrix $\boldsymbol{A} = (\boldsymbol{a}_{ij})_{n_\mathrm{v} \times n_\mathrm{v}}$, communication cost matrix $\boldsymbol{B} = (\boldsymbol{b}_{hp})_{n_\mathrm{p} \times n_\mathrm{p}}$, a_{ij} represents the traffic between VM *i* and VM *j*, b_{hp} represents the communication cost of PM *h* and PM *p*, n_v is the number of VMs, and n_p is the number of the PM. Here, the communication cost refers to the number of switches which traffic through between PMs. The more number of switches, the greater the cost of communication.

The goal is to find a mapping function $\pi(i)$ to meet VM *i* placed on PM, a VM can only be placed on a single PM, but a PM can place multiple VMs.

The objective function can be formally expressed as:

$$\min \ T_{\mathrm{cos}t} = \sum_{i,j=1}^{n_\mathrm{v}} a_{ij} b_{\pi(i)\pi(j)} + \sum_{i=1}^{n_\mathrm{v}} e_i g_{\pi(i)} \tag{1}$$

where e_i is the traffic between VM *i* and external communications of data center, $g_{\pi(i)}$ is the communication cost from the PM which places VM *i* to the external switch. The second part of the function is seen as a fixed number, but it has to be considered if affecting MLU, and it will be affected due to the different location of VM.

3.2 Optimization of MLU

x_{st}^{ij} is the traffic which is assigned to the link (s,t) from traffic demand of VM *i* and VM *j*. c_{st} represents the capacity of network link (s,t).

Network link utilization l_{st} is expressed as:

$$l_{st} = \frac{\sum_{i,j} x_{st}^{ij}}{c_{st}} \tag{2}$$

To let l_{st} be as small as possible, the problem can be expressed as:

$$\left.\begin{aligned} &\mathrm{Min} \ \ L_{\mathrm{cos}t} = \max\{l_{st}\} \\ &\mathrm{s.t.} \\ &\qquad \sum\nolimits_j x_{st}^{ij} - x_{ts}^{ij} = 0; \quad s \neq \pi(i), \pi(j) \\ &\qquad \sum\nolimits_{i,j} x_{st}^{ij} \leqslant l_{st} c_{st} \\ &\qquad x_{st}^{ij} \geqslant 0 \end{aligned}\right\} \tag{3}$$

3.3 Optimization of total traffic and MLU

The goal is to minimize MLU and total traffic:

$$\mathrm{Min} \quad f = T_{\mathrm{cos}t} + \gamma L_{\mathrm{cos}t} \tag{4}$$

where, γ is the weight coefficient. It is a multi-objective optimization problem and is also a classic combinatorial optimization problem, so the authors use a heuristic algorithm and design an optimization approach by combining ACO with 2-opt local search.

Table 2 Key notation of ACO algorithm

Symbol	Description
p_{ih}^k	Probability ant *k* selects which VM *i* is placed on PM *h*
η_{ih}	Heuristic information to optimize physical server with which that VM *i* is placed onto PM *h*
τ_{ih}	Pheromone by which VM *i* is placed onto PM *h*
ρ	Volatile factor

4 Algorithm

4.1 ACO algorithm for VMP

ACO [23] found an optimal algorithm from the positive feedback and the distributed cooperation. It operates through paralleled cooperation and is consistent with the characteristics of the cloud computing; it is a kind of heuristic search algorithm based on ant colony optimization; it makes full use of the biological ant colony characteristics to search the collective optimal feature for the shortest path from the ant nest to the food by simply individual information. This process shares some similarities with the solution to the optimization, and it achieves better results in dealing with combination optimization [24].

Ant *k* begins to construct solutions without any VM placement, ant *k* is randomly placed in VM *i*, in each construction step, with constraints of PM resource $\vec{\boldsymbol{H}}_h$, $p_{ih}^k(t)$ is defined as the probability by which ant *k* is selected to put VM *i* on PM *h*.

$$i = \begin{cases} \arg \max\limits_{u \in N_h^k(t)} \left\{\tau_{uh}^{\alpha}(t) \eta_{uh}^{\beta}(t)\right\}; & r \leqslant r_0 \\ I; \quad r > r_0 \end{cases} \tag{5}$$

$r \sim U(0,1)$, $r_0 \in [0,1]$ is the specific parameter values specified by the algorithm.

In *I*, ant *k* decides the probability to VM *i* on PM *h*:

$$p_{ih}^k(t) = \begin{cases} \dfrac{\tau_{ih}^{\alpha}(t) \eta_{ih}^{\beta}(t)}{\sum\limits_{u \in N_h^k(t)} \tau_{uh}^{\alpha}(t) \eta_{uh}^{\beta}(t)}; & u \in N_h^k(t) \\ 0; \quad \text{other} \end{cases} \tag{6}$$

$\tau(t)$ represents the pheromone that VM i is placed on PM h, the larger the pheromone value, the greater the likelihood of placement. N_h^k represents the collections of VMs on which all the unallocated VMs can be placed.

The objective function is expressed as a combination of distance matrix and traffic matrix potential vector. $\boldsymbol{A}=(\boldsymbol{a}_{ij})_{n_\text{v}\times n_\text{v}}$ is traffic matrix, $\boldsymbol{B}=(\boldsymbol{b}_{hp})_{n_\text{p}\times n_\text{p}}$ is communication cost matrix, vector $\boldsymbol{A}_i$ and $\boldsymbol{B}_h$ can be defined as:

$$\left.\begin{aligned}
&\boldsymbol{A}_i=\sum_{j=1}^{n_v}a_{ij};\quad i=1,2,\ldots,n_\text{v}\\
&\boldsymbol{B}_h=\sum_{p=1}^{n_p}b_{hp};\quad m=1,2,\ldots,n_\text{p}\\
&\eta_{ih}=\frac{1}{A_iB_h}
\end{aligned}\right\}\tag{7}$$

η_{ih} is heuristic inclination, and it also represents the potential benefits when VM i is assigned to PM h. The smaller the value of $\boldsymbol{A}_i\cdot\boldsymbol{B}_h$ is, the greater the possibility of VM assigned to h.

$$\tau_{ih}(t+1)=(1-\rho)\tau_{ih}(t)+\Delta\tau_{ih}^{i\text{best}}(t)+\Delta\tau_{ih}^{g\text{best}}(t)\tag{8}$$

ρ is a volatile factor.

$$\Delta\tau_{ih}^{\text{best}}(t)=\begin{cases}\dfrac{Q}{f}\\ 0\end{cases}\tag{9}$$

The value of f is determined by the Eq. (4). The smaller this value is, the better is the corresponding placement, the greater τ_{ih}^{best}. Q is a fixed value determined by ant colony algorithm. $\Delta\tau_{ih}^{i\text{best}}$ is the best optimal solution of the current iteration. $\Delta\tau_{ih}^{g\text{best}}$ is the best global optimal solution. The best ant is selected from the global optimal solution or the current iteration to enhance the pheromone.

4.2 ACO joints with 2-opt local search

For large data calculation, ACO has a long way-finding, slow convergence and high time complexity. To reduce the time complexity by reducing the ant number or the number of iterations may fall into local optimum, that is, all individuals may find the same solution, so it may be failure to find the global optimum. Based on the above, we will combine 2-opt local search with ACO to improve search speed and accelerate the convergence speed.

To accept the modification when an exchange has a better solution compared with other exchanges in possible random order.

$$\begin{aligned}\Delta f(\pi,i,j)=&(a_{ii}-a_{jj})(b_{\pi(j)\pi(j)}-b_{\pi(i)\pi(i)})+\\
&(a_{ij}-a_{ji})(b_{\pi(j)\pi(i)}-b_{\pi(i)\pi(j)})+\\
&\sum_{l=1,l\neq i,j}^{n_v}[(a_{li}-a_{lj})(b_{\pi(l)-\pi(j)}-b_{\pi(l)\pi(i)})+\\
&(a_{il}-a_{jl})(b_{\pi(j)-\pi(l)}-b_{\pi(i)\pi(l)})\end{aligned}\tag{10}$$

If $\Delta f(\pi,i,j)<0$, then accept the modification.

4.3 Algorithm description

Algorithms 1 ACO algorithm for VMP

Input: network topology, link capacity, traffic routing, traffic demand
Output : VM to PM mapping function $h=\pi(i)$

```
Initialization parameters $\alpha,\beta,$nAnts,ncMax
While nc ≤ ncMax do
  For node i= 1 to $n_\text{v}$ do
    For ant k= 1 to nAnts do
      Calculate $N_h^k(t)$, $\eta_{ih}$ using formula (7)
      If $N_h^k(t)\neq\varnothing$ then
        If $r\leqslant r_0$ then
          Select I according to the random proportional rule
          using formula (5)
        else
          Calculate $p_{ih}^k(t)$ select i using formula (6)
        Endif
      Endif
    Endfor
  Endfor
  Refine iterative optimal solution with local search using
  formula (10)
  For each map $(i,h)\in\pi$ do
    Update the pheromone deposits on using formula (8) (9)
  End for
End while
Return $\pi^{\text{best}}$
```

5 Evaluation

5.1 Simulation setup

The authors use C++ to develop an improved ACO simulation program, select the clustering algorithm [25], LS algorithm and SA algorithm, then compare these three algorithms with ACO.

Assume there are 256 VMs and 64 PMs, under different network topologies. Tree, VL2 and fat-tree are selected to conduct the simulation. For tree, the shortest path routing is applied, ECMP is applied for fat-tree, and valiant load balancing (VLB) for VL2. For VM traffic matrix, the experiments took the traffic patterns in Ref. [9].

ACO parameter values affect so much; these parameters greatly depend on better convergence. Relevant parameters of ACO are initialized values in Table 3. We select nAnts in each generation to search path, and ncMax refers to maximum number of iterations. These parameters are mainly with reference to Ref. [23].

Table 3 Parameter ACO algorithm

α	β	ρ	ncMax	nAnts
1	2	0.1	200	31

5.2 Results

5.2.1 Optimization total traffic without optimization MLU

Fig. 2 shows the results of total traffic without optimizing MLU. Clustering algorithm has better optimization effect for total traffic, LS is less effective, and ACO almost has no difference from SA and LS. The total traffic by tree topology is smaller than by fat-tree.

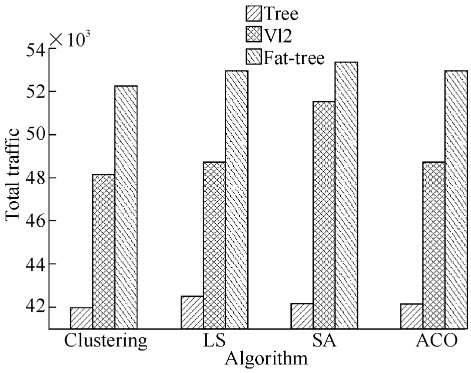


Fig. 2 Total traffic before optimizing MLU

Fig. 3 shows MLU of different algorithms on different topologies. Fig. 4 shows minimum link utilization of different algorithms on different topologies. As it is seen in Fig. 3 and Fig. 4, tree topology has better total optimization, and it also has the largest MLU. It can be concluded that the tree topology is more likely to generate the congestion among the same VM traffic. This also confirms our viewpoint: although Fat-Tree has large total traffic, it can make traffic more evenly distributed due to multi-path connections. For VL2, there is a great gap between minimum link utilization and MLU, so there is uneven traffic distribution.

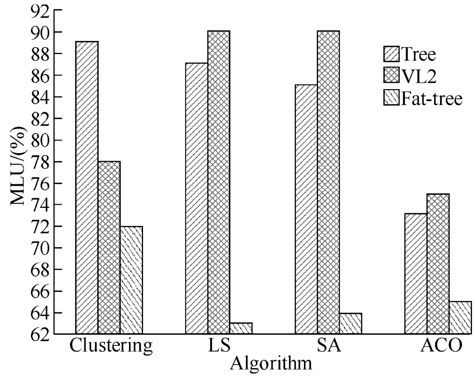


Fig. 3 MLU before optimizing MLU

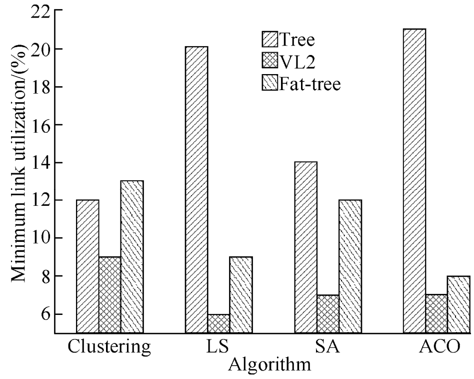


Fig. 4 Minimum link utilization before optimizing MLU

5.2.2 Optimization of total traffic and MLU

Eq. (4) and algorithms 1 are used not only to optimize the network total traffic, but also to optimize the network link utilization. ACO-TM refers to the situation in which ACO both optimizes the total traffic and MLU, while ACO-T refers to the situation in which ACO only optimizes the total traffic. By using the same data as in experiment 1, the traffic is applied among the same VMs to different network topologies. Fig. 5 shows the comparison of total traffic using LS, ACO-T and ACO-TM in different topologies. Compared with ACO-TM and LS, ACO-T's optimization effect is the best while that of ACO-TM is the poorest, however, the gap between them is slight. Fig. 6 shows the comparison of MLU using LS, ACO-T and ACO-TM in different topologies. Compared with LS and ACO-T, ACO-TM greatly reduces MLU, and Fat-Tree is decreased by 20%. As can be seen from Fig. 5 and Fig. 6, although ACO-TM slightly increases the total traffic, it most successfully reduces MLU and achieves better network performance. For different topologies, ACO-TM can effectively reduce MLU. As fat-tree and VL2 use multi-paths, their optimizing effects are more remarkable.

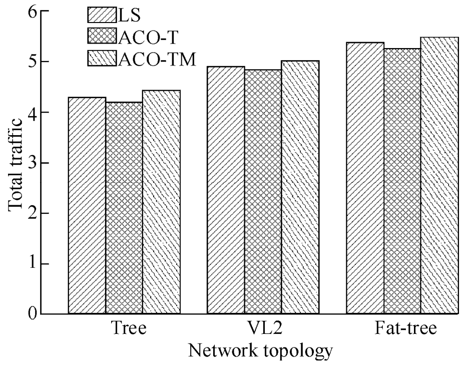


Fig. 5 Total traffic variation caused by ACO-T and ACO-TM

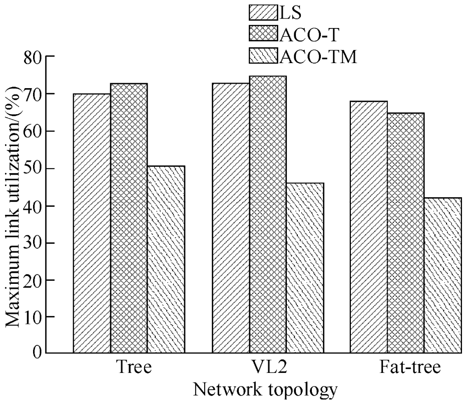


Fig. 6 MLU variation caused by ACO-T and ACO-TM

5.2.3 Hotspot patterns

If overloading some traffic when generating VM traffic matrix, the network link will produce hot spots which will cause the congestion. Fig.7 shows the variation of hotspots number using ACO-T and ACO-TM in different topologies. Compared with ACO-T, the number of hotspot links of ACO-TM is significantly decreased, and the hotspot link number of Fat-Tree decreases by 37%. Although ACO-TM cannot completely avoid the network congestion, they greatly improve the network performance.

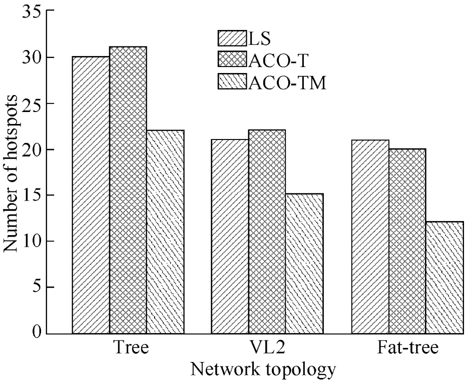


Fig. 7 Hotspot number variation caused by ACO-T and ACO-TM

5.3 Discussions

5.3.1 Performance

ACO's execution time is in the second-level. Due to the parallel features of ACO, ACO can be run on many machines and parallel computing is used to reduce the time and improve the performance. The second-level execution time is acceptable to solve the in the data center, particularly to solve NP-complete problem. By comparison, although the time performance of ACO is a little weak, ACO shows more accurate result. The experiments verify that it is feasible to apply ACO to solve the problems.

5.3.2 Convergence

Better convergence is also expected when accompanying with the improved accuracy, local search is so used to improve ACO convergence. Here, the authors mainly concern the number of iterations and the number of ants. In the experiment, when taking time performance into account, the maximum iteration number is 500. A group of VMs is selected, and different maximum iterations and ant number are set, when iteration number is 200 and the ant number is 31, there is little difference between experimental results.

5.3.3 Robustness

After initializing ACO parameter, iterations times and ant number, the authors conduct experiments and observe the results of each experiment in order to eliminate the noise data. At last, the robustness of ACO is analyzed. Multiple runs and less jitter results can verify the less possibility for local optimum, and ACO has quite good robustness.

6 Related work

The objectives of VMP are mainly for energy savings [4–5], fault tolerance [26] and QoS management [6], etc.. However, these studies did not consider the network resources, which may cause uneven network traffic and low utilization of network resources, so the proposal is to optimize the network traffic and improve the network performance on the condition of meeting the needs of physical server resources, and Refs. [9–10,27–29] are similar to our solution.

Jiang et al. [27] attempted to improve physical node utilization and to optimize the network link utilization by changing traffic routing. In fact, the traffic routing path will be changed, so this scheme did not consider the optimization of total traffic in DCN.

Meng et al. [9] used traffic-related to improve the network scalability. By optimizing VM position in physical host, VM traffic is associated with their network physical distances, VM with larger communication traffic can be placed between two physical hosts nearby so as to reduce the total traffic. Biran et al. [28] proposed to allocate a placement which not only satisfies the predicted communication demands but is also resilient to demand time-variations. However, these two schemes can't consider the optimization of MLU.

Wen et al. [10] presented an efficient online algorithm to reduce congestion with controllable VM migration traffic as well as low time complexity. When VM are initially placed, the traffic is localized, which is similar to the target of minimizing MLU. Thus, the scheme proposes to substitute minimum total traffic for minimizing link utilization, but we consider that these two goals are different.

Shrivastava et al. [29] optimized network performance, traffic or end-to-end delay by using VM migration. The scheme was recommended on overloaded VM migration to balance PM workload, its main objective is to eliminate the overloaded PM and minimize the congestion. Thus, the main focus of that scheme is on the network traffic of VM migration, which is inconsistent with our optimization goal.

7 Conclusions

VMP problem is one of the challenging tasks in cloud data centers. In the VMP scheme, the authors consider multi-resource constraints of PM and attempt to improve network performance, optimize the total network traffic to improve the network scalability, and the network MLU to improve the network performance, avoiding the congestion. This is a much more complex problem both due to its quadratic nature (being the communication between a pair of VMs) and factors beyond the physical host such as the network topologies and the routing scheme. Lastly, ACO is combined with 2-opt local search algorithm as the solution, the simulation results show that our proposal achieves good results.

The article proposes a VMP method to improve the network performance and effectively alleviate the congestion in DCN. If VMP is combined with the dynamic flow adjustment method, better results may be achieved. This will be one of the future researches. In addition, VMP is an initial placement, how to achieve the live migration will be another task in future, because how to minimize VM migration cost without affecting application performance is an important issue.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61002011); the National High Technology Research and Development Program of China (863 Program) (2013AA013303); the Fundamental Research Funds for the Central Universities (2013RC1104); the Natural Science Foundation of Gansu Province, China (1308RJZA306); the Open Fund of the State Key Laboratory of Software Development Environment (SKLSDE-2009KF-2-08).

References

1. Barham P, Dragovic B, Fraser K, et al. Xen and the art of virtualization. ACM SIGOPS Operating Systems Review, 2003, 37(5): 164−177
2. Armbrust M, Fox A, Griffith R, et al. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28. Berkeley, CA, USA: EECS Department, University of California at Berkeley, 2009
3. Amazon EC2. http://aws.amazon.com/ec2/instance-typ- es/
4. Verma A, Ahuja P, Neogi A. pMapper: Power and migration cost aware application placement in virtualized systems. Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (Middleware'08), Dec 1−5, 2008, Leuven, Belgium. New York, NY, USA: Springer-Verlag, 2008: 243−264
5. Dong J, Jin X, Wang H, et al. Energy-saving virtual machine placement in cloud data centers. Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'13), May 13−16, 2013, Delft, Netherlands. Los Alamitos, CA, USA: IEEE Computer Society, 2013: 618−624
6. Bobroff N, Kochut A, Beaty K. Dynamic placement of virtual machines for managing SLA violations. Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM'07), May 21−25, 2007, Munich, Germany. Piscatawaw, NJ, USA: IEEE, 2007: 119−128
7. Singh A, Korupolu M, Mohapatra D. Server-storage virtualization: Integration and load balancing in data centers. Proceedings of the 2008 International Conference for High Performance Computing, Networking, Storage and Analysis (SC'08), Nov 15−21, 2008, Austin, TX, USA. Piscatawaw, NJ, USA: IEEE, 2008: 12p
8. Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture. ACM SIGCOMM Computer Communication Review, 2008, 38(4): 63−74
9. Meng X, Pappas V, Zhang L. Improving the scalability of data center networks with traffic-aware virtual machine placement. Proceedings of the 29th Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'10), Mar 14−19, 2010, San Diego, CA, USA. Piscataway, NJ, USA: IEEE, 2010: 9p

10. Wen X, Chen K, Chen Y, et al. VirtualKnotter: Online virtual machine shuffling for congestion resolving in virtualized datacenter. Proceedings of the 32nd International Conference on Distributed Computing Systems (ICDCS'12), Jun 18–21, 2012, Macau, China. Los Alamitos, CA, USA: IEEE Computer Society, 2012: 12–21
11. Burkard R E, Cela F, Pardalos P M, et al. The quadratic assignment problem. Management Science, 1963, 9(4): 586–599
12. Woeginger G J. Exact algorithms for NP-hard problems: A survey. Combinatorial Optimization — Eureka, You Shrink: 5th International Workshop, Mar 5–9, 2001, Aussois, France. LNCS2570. Berlin, Germany: Springer, 2003: 185–207
13. Fortz B, Thorup M. Internet traffic engineering by optimizing OSPF weights. Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00): Vol 2, Mar 26–30, 2000, Tel Aviv, Israel. Piscataway, NJ, USA: IEEE, 2000: 519–528
14. Greenberg A, Hamilton J R, Jain N, et al. VL2: A scalable and flexible data center network. ACM SIGCOMM Computer Communication Review, 2009, 39(4): 51–62
15. Mysore R N, Pambori A, Farrington N, et al. PortLand: A scalable fault-tolerant layer 2 data center network fabric. ACM SIGCOMM Computer Communication Review, 2009, 39(4): 39–50
16. Benson T, Akella A, Maltz D A. Network traffic characteristics of data centers in the wild. Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC'10), Nov 1–3, 2010, Melbourne, Australia. New York, NY, USA: ACM, 2010: 267–280
17. Kandula S, Sengupta S, Greenberg A, et al. The nature of data center traffic: Measurements & analysis. Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement (IMC'09), Nov 4–6, 2009. Chicago, IL, USA. New York, NY, USA: ACM, 2009: 202–208.
18. CDC Infrastructur. 2.5 Design guide. 2007
19. Hopps C E. Analysis of an equal-cost multi-path algorithm. RFC 2992. 2000
20. Al-Fares M, Radhakrishnan S, Raghavan B, et al. Hedera: Dynamic flow scheduling for data center networks. Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10) Apr 28–30, 2010, San Jose, CA, USA. Berkeley, CA, USA: USENIX Association, 2010: 19
21. Benson T, Anand A, Akella A, et al. The case for fine-grained traffic engineering in data centers. Proceedings of the 2010 Internet Network Management Workshop/Workshop on Research on Enterprise Networking (INM/WREN'10), Apr 27, 2010, San Jose, CA, USA. Berkeley, CA, USA: USENIX Association, 2010: 2–2
22. McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69–74
23. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Computational Intelligence Magazine, 2006, 1(4): 28–39
24. Stützle T, Dorigo M. ACO algorithms for the quadratic assignment problem. Corne D, Dorigo M, Glover F. New Ideas in Optimization. New York, NY, USA: McGraw-Hill, 1999: 33–50
25. Hartigan J A. Clustering algorithms. New York, NY, USA: John Wiley & Sons, 1975
26. Nagarajan A B, Mueller F, Engelmann C, et al. Proactive fault tolerance for HPC with Xen virtualization. Proceedings of the 21st Annual International Conference on Supercomputing (ICS'07), Jun 17–21, 2007, Seattle, WA, USA. New York, NY, USA: ACM, 2007: 23–32
27. Jiang J W, Lan T, Ha S, et al. Joint VM placement and routing for data center traffic engineering. Proceedings of the 31st Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'12), Mar 25–30, 2012, Orlando, FL, USA. Piscataway, NJ, USA: IEEE, 2012: 2876–2880
28. Biran O, Corradi A, Fanelli M, et al. A stable network-aware VM placement for cloud systems. Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'12), May 13–16, 2012, Ottawa, Canada. Los Alamitos, CA, USA: IEEE Computer Society, 2012: 498–506
29. Shrivastava V, Zerfos P, Lee K, et al. Application-aware virtual machine migration in data centers. Proceedings of the 30th Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'11), Apr 10–15, 2011, Shanghai, China. Piscataway, NJ, USA: IEEE, 2011: 66–70

(Editor: ZHANG Ke-xin)

From p. 61

References

1. Damnjanovic A, Montojo J, Wei Y B, et al. A survey on 3GPP heterogeneous networks. IEEE Wireless Communications, 2011, 18(3): 10–21
2. Chandrasekhar V, Andrews J G, Gatherer A. Femtocell networks: A survey. IEEE Communications Magazine, 2008, 46(9): 59–67
3. Kim D, Choi S. Load balancing in open access femtocell based two-tier cellular networks. Proceedings of the IEEE Global Communications Conference (GLOBECOM'12), Dec 3–7, 2012, Anaheim, CA, USA. Piscataway, NJ, USA: IEEE, 2012: 5123–5129
4. Cho S, Choi W. Coverage and load balancing in heterogeneous cellular networks with minimum cell separation. IEEE Transactions on Mobile Computing, to appear
5. 3GPP TR 36.913 (V8.0.0). 3rd generation partnership project; technical specification group radio access network; requirements for further advancements for E-UTRA (LTE-advanced). 2008
6. Ye Q Y, Rong B, Chen Y D. User association for load balancing in heterogeneous cellular networks. IEEE Transactions on Wireless Communications, 2013, 12(6): 2706–2716
7. Wang J, Liu J G, Wang D Y, et al. Optimized fairness cell selection for 3GPP LTE--A macro-pico HetNets. Proceedings of the 74th Vehicular Technology Conference (VTC-Fall'11), Sep 5–8, 2011, San Francisco, CA, USA. Piscataway, NJ, USA: IEEE, 5p
8. Corroy S, Falconetti L, Mathar R. Dynamic cell association for downlink sum rate maximization in multi-cell heterogeneous networks. Proceedings of the IEEE International Conference on Communications (ICC'12), Jun 10–15, 2012, Ottawa, Canada. Piscataway, NJ, USA: IEEE, 2012: 2457–2461
9. Chen C, Baccelli F, Roullet L. Joint optimization of radio resources in small and macro cell networks. Proceedings of the 73rd Vehicular Technology Conference (VTC-Spring'11), May 15–18, 2011, Budapest, Hungary. Piscataway, NJ, USA: IEEE, 2011: 5p
10. Ghimire J, Rosenberg C. Resource allocation, transmission coordination and user association in heterogeneous networks: A flow-based unified approach. IEEE Transactions on Wireless Communications, 2013,12(3):1340–1351
11. Yu W, Lui R. Dual methods for nonconvex spectrum optimization of multicarrier systems. IEEE Transactions on Communications, 2006, 54(7): 1310–1322
12. Boyd S, Vandenberghe L. Convex optimization. New York, NY, USA: Cambridge University Press, 2004
13. Bertsekas D P. Nonlinear programming.2nd ed. Belmont, MA, USA: Athena Scientific, 1999
14. 3GPP TR 36.814 (V9.0.0). Further advancements for E-UTRA physical layer aspects. 2010

(Editor: ADA Lai Ti)