Traffic-aware and Reliability-guaranteed Virtual Machine Placement Optimization in Cloud Datacenters

Xuan Liu, Bo Cheng, Yi Yue, Meng Wang, Biyi Li, Junliang Chen
Beijing University of Posts and Telecommunications
Beijing, China
{*liuxuan0527, chengbo, yueyi, mengwang, lbyi0402, chjl*}*@bupt.edu.cn*

***Abstract*—With the increasing scale of cloud datacenters and rapid development of virtualization technologies, many cloud-based services have been deployed to meet requirements. Virtual machines (VMs) are placed on physical servers, and often provide virtual environment for cloud services. Therefore, virtual machines placement (VMP) problem has gradually attracted many attentions. It is meaningful that how to effectively and efficiently place VMs on servers to guarantee the service reliability and reduce the bandwidth consumption. In this paper, we first formulate VMP with a reliability model and a bandwidth consumption model, and analyse its complexity. Then we propose a VMP optimization approach to solve the problem and prove its effectiveness and efficiency. The core algorithm of our approach is an approximation algorithm to get VM partitions under the constraint of a specified reliability parameter. Then placement problem is transformed into matching problem between VM partitions with physical servers. Finally, the evaluation results show the effectiveness of the proposed approach and performance advancement over the existing approaches.**

***Keywords*-Traffic-aware; Reliability; Virtual Machine Placement; Cloud Datacenter**

I. Introduction

With the rapid development of cloud computing, exponentially increasing numbers of applications and services are deployed on Virtual Machines (VMs), which are all placed on physical servers [1]. As statistical analysis shown on the study [2], with the growing trend of datacenters, the probability of host servers failures is nontrivial. Node failures may lead to the loss of VMs allocated on the server, and all deployed services will fall into invalidation consequently. Reliability, as a substantial property of Quality of Service (QoS), should be guaranteed. An efficient approach to overcome the reliability problem is to create some VM replicas as backup and place them in a distributed manner. Besides, loosely placing VMs in multiple domains in the datacenter can relatively enhance the reliability and reduce the loss caused by node failures.

However, existing evidence [3] suggests that the traffic between VMs in an ordinary datacenter takes up approximately 80% of total traffic. Further, as Cisco indicated in [4], global datacenter IP traffic will be from 4.7 ZB/year in 2015 to 15.3 ZB/year in 2020 with practically threefold growth [5]. In this case, the bandwidth consumption profoundly affects cloud datacenters. If VMs are placed in a compact manner instead of loose manner, the bandwidth consumption could be reduced. Hence, it is a meaningful problem to balance the high reliability requirement and the low bandwidth consumption of Virtual Machines Placement (VMP) in a cloud datacenter. Existing tools like VMware Capacity Planner [6], IBM WebSphere CloudBurst [7] emphasize the allocation and scheduling of CPU, memory, disk and power consumption savings, but reliability and bandwidth consumption are not into consideration.

In this paper, we address the traffic-aware and reliability-guaranteed virtual machines placement as an optimization problem, deonted as TRVMPO problem. VMP demands include the VMs topology, bandwidth demands, a required reliability parameter and the communication cost matrix of hosts in datacenters. We first establish a reliability model, and then formulate the whole physical bandwidth consumption of VMP as an optimization objective function. The optimal solution to the TRVMPO is a placement strategy for each VM in order to minimize the function value. And we analyse the complexity of TRVMPO problem and prove the problem as NP-hard. We propose an approximation algorithm based on the minimum k-cut of graph to solve TRVMPO, and theoretically prove the effectiveness and efficiency of the approach. The core aspect of TRVMPO algorithm involves two parts: 1) the algorithm partitions VMs in accordance with VMs integration threshold to guarantee the required reliability; 2) VM partitions are matched with target hosts. Finally, we conduct our experiments and compare the performance with other VMP algorithms. The evaluation results show the effectiveness of proposed TRVMPO algorithm and performance advancement over the other approaches.

To summarize, we makes these following contributions:

- We propose a reliability model and a bandwidth consumption model. The former one deduces the VM integration threshold, which guarantees the reliability of VMP. The latter one combines VM bandwidth demands and network communication costs together, and formulte into an optimization objective function.
- We theoretically prove that obtaining the optimal solution is an NP-hard problem. Based on those two models mentioned above, we transform the VMP problem as

2159-6190/19/$31.00 ©2019 IEEE
DOI 10.1109/CLOUD.2019.00027

a matching problem. Then we propose the TRVMPO algorithm to minimize the objective function value.
- We compare our approach with other VMP algorithms in terms of experimental performance via simulations.

The rest of this paper is organized as follows. Section II introduces a review of related work. In Section III, we describe the problem, formulate models, and analyse the complexity of the TRVMPO problem. Section IV develops the TRVMPO algorithm in detail. Experimental results are shown in Section V to evaluate the performance and efficiency of TRVMPO algorithm. Finally, Section VI concludes the paper, with the summary and future work.

II. Related Work

Many approaches have been proposed to solve VMP problems with different kinds of optional constraints and optimization objectives, in different application scenarios.

In the study [8], authors tackle traffic-aware VMP problem in the cloudlet mesh. They represent a micro datacenter in a node deployed at the edge layer of the mobile network. A heuristic algorithm is designed to obtain the maximum number of available VMs into the cloudlet mesh and minimize the total inter-cloudlet traffic costs. However this work does not consider any QoS constraint. The work in the [9] proposes a two-tier approximate algorithm to address the scalability concern in datacenters. This work analyses the impact of the traffic patterns and the network architectures on performance of the given algorithm. The problem formulation assumes the number of VMs and placement positions are equal, so that graph-based Balanced k-min Cut method can conduct effectively. Besides, this work involves four types of datacenter network architectures, namely tree, VL2 [3], fat-tree [10], and BCube [11]. The evaluation demonstrates that the proposed VMP algorithm can get greater benefit for a multi-tier architecture BCube, but less benefit for VL2, which adapts load balancing techniques. This work compares proposed algorithm with a random placement in which any VM has equal probability to be placed to any host, and showed the value and limitation of utilizing VMP to improve network scalability. But no QoS constraint is taken into consideration neither.

In the study [12], the authors regards the reliability of VMP as an optimization objective. And this work proposes a redundant VMP optimization approach to heighten the reliability in the cloud environment. The approach decides an appropriate placement of primary VMs and backup VMs with k-fault-tolerance assurance. The proposed heuristic algorithm aims to find a maximum weight matching based on the bipartite graph. The author also raises the reliability by checkpointing mechanism in the cloud environment [13]. This work presents an optimal algorithm with the edge switch failure-aware feature, in order to enhance the reliability of each failure domain in the datacenter. However, these two works are only for fat-tree datacenters. In the study [14], authors address reliable VMP problem as placing at most H groups of k VMs on a minimum number of nodes (H and k are both positive integers here), with the availability parameter no less than a fixed value. This work denotes VMP availability as the probability of at least one set of normally-operating VMs during the requested lifetime. Subsequently it analyses two aspects of failures/availablities, namely Shared-Risk Node Grope failures [15] and single node failures. An exact Integer Nonlinear Program(INLP) and a heuristic algorithm are combined to solve this problem in both cases. This approach can achieve the best performance compared with other heuristics in its experiment, but it always expends larger running time than other heuristics. In the work [16], the author proposes the RAVDC algorithm based on Greedy Graph Growing Algorithm [17] to solve reliability-guaranteed Virtual Data Center (VDC) embedding problem. It can satisfy the specified reliability requirement and reduce bandwidth consumption of VDC up to 30% without consuming any extra energy. With the same QoS constraint and optimization objective, our approach selects C. Zuo's algorithm as one of experimental comparisons.

The work [18] optimizes multiple objectives of the VMP problem in datacenters. And this work proposes an improved approach EEKnEA to synthetically optimize four aspects, namely energy consumption, network load balance, resource utilization, and the robustness of resource allocation. EEKnEA is improved by exploiting an energy-efficient-oriented population initialization strategy based on the knee point-driven evolutionary algorithm (KnEA) [19].In the study [20], authors propose a convex optimization algorithm to solve Multi-level Join VM Placement and Migration (MJPM) problem in datacenters, to minimize the resource usage and the power consumption. This approach formulates a multi-objective optimization function of VMP, involving inter-server communication power consumption, total power consumption of PMs, migration cost of VMs, and maintenance cost of hardware. MJPM algorithm utilizes the nuclear norm and augments of objective function, to reach a globally optimal solution. With the similar optimization objective, X. F. Liu and Z. H. Zhan [21] present an ant colony system algorithm to minimize the number of active servers used for VMP. The presented approach can deal with a variety of VMP requests with different VM sizes in cloud datacenters. The work in [22] considers limited physical server resources and bandwidth as a group of constraints, and proposes an ant colony optimization algorithm to minimize the whole resource cost of the VMP.

The studies mentioned above are valuable basis of our work. Most of works focus on optimizing the resource utilization or emphatically improving reliability, but few focus on combination of reliability and traffic resources. So we take bandwidth consumption and reliability into comprehensive consideration and propose the solution in the following sections.

III. Virtual Machine Placement Optimization

In this section, we formally describe the TRVMPO problem and analyse the complexity. Table I clarifies significant notations that will be present throughout the whole paper.

Table I: Notations

Symbol	Meaning
D	VMP request
V	VM set of the VMP request
m	the number of VMs
L	link set of the VMP request
B_{ij}	bandwidth demand between VM i and VM j
B	bandwidth demand matrix
σ	permutation function
S	server set, or host set
$v(S_i)$	the number of deployed VMs on a server S_i
R	reliability parameter
K	VM integration threshold
H_{ij}	hops along communication path between VM i and j
P_{ij}	communication cost between slot i and j
P	communication cost matrix
B^*	the whole bandwidth consumption of the VMP
X	solution of the VMP scheme
$G = (V, E)$	VMP request graph
$w(\{v_i, v_j\})$	weight of the edge connecting node v_i and v_2
k	the number of slot-clusters, or VM partitions
n	the number of slots
$\{c_1, c_2 \dots c_k\}$	slot-cluster set
$Avai(S_i)$	the max number of available slots on host S_i
η	approximation ratio

A. Problem Description

When VMP request arrives, we define it as $D = \{V, L\}$, where V represents the set of VMs within this request and L represents the set of links of each VM pair. Specifically, we denote m as the number of VMs. Each element in set L can be denoted as $< vm_i, vm_j, B_{ij} >$. That means a link connecting with a VM pair, with a corresponding bandwidth demand. Besides, L also describes the VM network topology of the VMP request. It is the basis that VMs have to be deployed on physical servers. Those servers belong to nodes in the datacenter network, as same as switches. A slot refers to one allocation of resource on a host, and the resource includes CPU, memory and disk storage [9]. Each slot can be occupied by a VM, and each host can be resided on by multiple slots. So VMP problem can be transformed as matching problem between VMs and slots with corresponding permutation function formulated as $\sigma : [1 \dots n] \rightarrow [1 \dots n]$.

B. Reliability Model

The failure domain in the datacenter includes physical servers, switches and power supply units. Based on the work in [23], the reliability of the VMP is defined as the available ratio of VMs for a worst-case of a server failure. We denote S as the server set, and $v(S_i)$ as the number of deployed VMs on the server S_i. Hence, the reliability parameter can be formulated as shown in the equation 1.

$$R = 1 - \frac{\max v(S_i)}{\sum v(S_i)}, S_i \in S \tag{1}$$

With the reliability parameter R, VM integration threshold K can be deduced. The threshold K indicates the max number of VMs that can be deployed on a host, and can be formulated as present in the equation 2.

$$K = \lfloor (1 - R) \cdot m \rfloor \tag{2}$$

Figure 1 illustrates the relation between VMP and the reliability, and calculates the corresponding bandwidth consumption in different VMP schemes. Figure 1-(a) shows a request of VMP with each bandwidth demand upon each link. Figure 1-(b) shows a simple datacenter network. Obviously, if two VMs are placed on the same host, the communication between them would not occupy any external bandwidth of the physical link. Otherwise, the VM pair may consume bandwidth along their traffic path related with the datacenter topology. Figure 1-(c), 1-(d), 1-(e), 1-(f) show four different schemes of the placement request in Figure 1-(a). In Figure 1-(c), the reliability of VMP is valued 75%, and 56 bandwidth units are consumed. If we allocate the VMs in pairwise way like Figure 1-(d) shows, the reliability value will decrease to 50%, and 34 bandwidth units are consumed. The scheme in Figure 1-(e) also adopts the pairwise way and has the same reliability with figure 1-(d), but it takes 24 bandwidth units under this VMP scheme. Moreover, the way shown in Figure 1-(f) just consumes 10 bandwidth units, but the reliability decreases to 25%.

It can be inferred that high reliability and low bandwidth consumption conflict with each other. Compact VMP scheme can reduce bandwidth consumption, but may deteriorate the reliability. Conversely, loose placement can improve the reliability, but may cause high bandwidth consumption.

More importantly, different placement schemes with the same reliability constraint may lead to different bandwidth consumption. Hence, under a specific reliability constraint, the optimization goal is to choose an apposite VMP scheme that makes bandwidth consumption as less as possible.

C. Bandwidth Consumption Model

When a VM pair is placed on the same host, the Hypervisor of the host will be in charge of their communication, with no extra bandwidth consumption of physical links. Therefore, we formulate the bandwidth consumption of the VMP request as shown in the equation 3, where H_{ij} indicates hops along the communication path between VM i and VM j. When VM i and VM j are placed on the same host, $H_{ij} = 0$. We use P_{ij} to represent the communication cost between slot i and slot j. So the matrix $P_{n \times n}$ is fixed in relation to the datacenter network topology. The bandwidth demand between VM i and VM j can be denoted as B_{ij}.

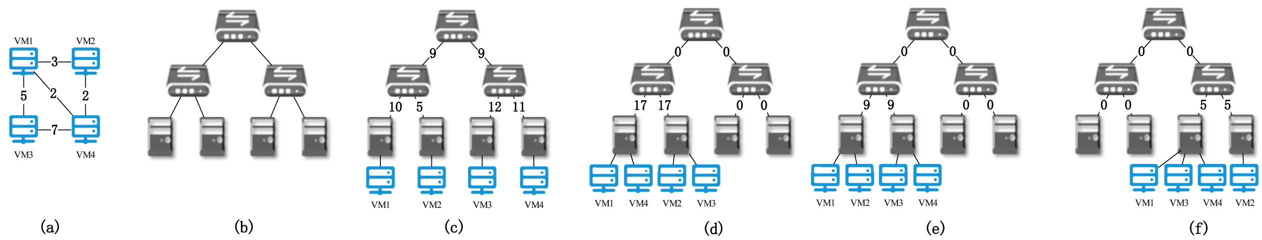


Figure 1: VMP request in the datacenter network with the reliability and the bandwidth consumption illustration. (a) VMP request topology with bandwidth demand. (b) datacenter network topology. (c)-(f) different VMP schemes.

Actually, it is a convenient approach that transform VMP problem into matching problem between n VMs and n slots with corresponding permutation function formulated as $\sigma : [1 \dots n] \rightarrow [1 \dots n]$.

$$B^* = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (H_{ij} B_{ij}) = \sum_{i,j=1}^{n} B_{ij} P_{\sigma(i)\sigma j} \tag{3}$$

Apparently, the number of slots can not be less than the number of VMs, which is a necessary condition for the VMP problem. We can add $n - m$ dummy VMs to make their numbers equal, so that permutation function can be effective.

D. Problem Formulation

The reliability constraint can be embodied in the cost matrix of slots. And the number of slots can be reckoned according to the number of VMs. TRVMPO problem can be described as follows: finding an applicable permutation to minimize the objective function as equation 4 shows. Furthermore, the objective function is equivalent to

$$\min tri(BX^T P^T X) \tag{4}$$

where $tri(M) = \sum_{i=1}^{n} M_{ii}$, that is the sum of main diagonal for matrix M. The matrix X denotes the solution of VMP scheme and present the placement of each VM. Specifically, X satisfies these following conditions:

1) $X_{ij} \in \{0, 1\} (\forall i, j)$
2) $\sum_{i=1}^{n} X_{ij} = 1 (\forall j)$
3) $\sum_{j=1}^{n} X_{ij} = 1 (\forall i)$
4) X belongs to $n \times n$ permutation matrix.

E. Problems Complexity Analysis

In this section, we analyse the complexity of TRVMPO problem. First, from the perspective of the optimization function, $B_{n\times n}$ and $P_{n\times n}$ are matrices of real values. So TRVMPO problem belongs to Quadratic Assignment Problem (QAM) with Koopmans-Beckmann form, which is a proved NP-hard problem [24].

Then we consider the solution space of permutation function, the size of solution space is $O(n^{r(n,K)})$, where $r(n, K) \approx n/K$. The complexity depends on n and K which are not constant. Due to the arbitrariness of the matrix B and the matrix P, the complexity of verifying optimal solution is also $O(n^{r(n,K)})$.

Finally, we analyse the computational complexity from the essence of TRVMPO problem. Multiple slots will be selected for the matching with VMs. We call a group of slots on the same host as a slot-cluster. In the complexity expression of the solution space, $r(n, K)$ is the number of slot-clusters. With the extreme situations taken into account, the number of slot-clusters ranges from 2 to n. We denote $k = r(n, K)$, and all VMs also should be separated into k partitions. Therefore, TRVMPO problem falls into Minimum k-cut Problem (MKP) [25]. MKP is formally described as follows: Given an undirected, weighted graph $G = (V, E)$ with n nodes, an assignment of weights to each edge is denoted as $w(\{v_i, v_j\})$. These k partitions $\{V_1, V_2 \dots V_k\}$ of G minimize the sum of weights of those edges connecting two nodes in different partitions. Equation 5 shows the formulation, where n and k are both positive integers.

$$\min \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \sum_{\substack{v_1 \in V_i \\ v_2 \in V_j}} w(\{v_i, v_j\}) \tag{5}$$

On the basis of MKP, Balanced Minimum k-cut Problem (BMKP) is introduced with a constraint condition that is the size of each partition is n/k [25]. In the TRVMPO problem, the size of each partition is no more than VM integration threshold K. The number of slot-cluster satisfies the condition $k = r(n, K) \geq n/K$, and the set of slot-cluster $\{c_1, c_2 \dots c_k\}$ can be given meanwhile. If the size of each slot-cluster is exactly equal to K, as $|c_i| = K, \forall i$, TRVMPO problem equals BKMP. In other words, when the TRVMPO problem is optimal, the corresponding BKMP is also optimal. That is because BKMP, as a special problem, belongs to the TRVMPO problem. It is proved that MKP and BKMP are both NP-hard [25], so the TRVMPO problem is also NP-hard.

IV. Algorithm

The analysis above shows that TRVMPO problem is NP-hard, and also belongs to QAM. In this section, we propose an approximation algorithm to solve the TRVMPO problem.

A. Host Selection and Slot-cluster Generation

According to the number of VMs and the integration threshold, candidate hosts can be selected. Each host can provide several candidate slots by required resources of VMs, namely CPU, memory, and disk storage. Each group of slots on each host naturally converge into a slot-cluster. The number of each element in the slot-cluster set $\{c_1, c_2 \dots c_k\}$ can satisfy the following quantitative relations:

$$\begin{cases} |c_i| = \min(K, Avai(S_i)) \\ \sum\limits_{i=1}^{k} |c_i| = n = (1+\alpha) \cdot m, 0 \le \alpha < 1 \end{cases} \tag{6}$$

$Avai(S_i)$ represents the max number of available slots on the host S_i. Simultaneously, communication cost matrix of slots $P_{n\times n}$ can be deduced.

B. Virtual Machine Placement Request Min k-cut and Matching with Slot-clusters

Firstly, a theorem of inequality need to be introduced.

Theorem 1. Assume two positive real number sequence as $0 \le x_1 \le x_2 \dots \le x_n$ and $0 \le y_1 \le y_2 \dots \le y_n$, for any permutation function $\sigma : [1 \dots n] \to [1 \dots n]$, the following inequalities 7 can hold [9].

$$\sum_{i=1}^{n} x_i y_{n+1-i} \le \sum_{i=1}^{n} x_i y_{\sigma(i)} \le \sum_{i=1}^{n} x_i y_i \tag{7}$$

The inequalities fit for optimizing $\sum\limits_{i,j=1}^{n} B_{ij} P_{\sigma(i)\sigma(j)}$ of the objective function in the equation 3.

Algorithm 1 shows the procedures of separating VMs into k partitions. This operation leads to the equivalent size between each partition and each slot-cluster. Before the matching operation, slot-clusters are sorted in descending order by $y(c_i) = \sum\limits_{t\in c_i, j\notin c_i} P_{tj}$, which is the whole communication cost of the slot-cluster c_i with other slot-clusters.

The following is the operation of virtual machine partition. For each edge in the VM request graph $G = (V, E)$, using *Gomory-Hu's* algorithm can obtain $n-1$ distinct min cuts and sort them by the sum of edge weights. Then according to these min cuts, each VM partition can be obtained with the equal size to the corresponding slot-cluster. That is to find the minimum j so that $g_1 \cup g_2 \cup \dots g_j$ can separate G into two parts as $G = G_0 \cup G_1$, where $|G_0| = \left|c_i^{'}\right|, G_0 \cap G_1 = \varnothing$. Partition will be iterative, until all nodes in G are handled.

Algorithm 1 VM min k-cut matching with slot-clusters

Input: VMP request graph $G = (V, E)$, slot-cluster cost matrix P, slot-cluster set $\{c_1, c_2 \dots c_k\}$
Output: VM partition set $\{V_1, V_2 \dots V_k\}$

$m, n \leftarrow$ the number of VMs, the number of slots in P
Add $n-m$ dummy VMs so that the number of nodes in G can be n, and update the request graph G
Generate extended matrix of bandwidth demand $B_{n\times n}$
for $i = 1$ to k **do**
  Calculate communication cost $y(c_i) = \sum\limits_{t\in c_i, j\notin c_i} P_{tj}$
end for
Sort slot-clusters in descending order by $\{y(c_i)\}$ and obtain set sequence $\{c_1^{'}, c_2^{'} \dots c_k^{'}\}$
Get $n-1$ distinct min cuts by *Gomory-Hu's* algorithm
Sort $n-1$ distinct min cuts in ascending order, by the sum of edge weights, and obtain the sorted min cut sequence $\{g_1, g_2 \dots g_{n-1}\}$
for $i = 1$ to k **do**
  $G_0 \leftarrow \varnothing$
  for $j = 1$ to $n-1$ **do**
    $G_0 \leftarrow G_0 \cup G_j$
    if $|G_0| = \left|c_i^{'}\right|$ **then**
      $V_i \leftarrow$ nodes set in G_0
      $G \leftarrow G - G_0$
    end if
  end for
end for
return $\{V_1, V_2 \dots V_k\}$

C. Mapping Virtual Machines with Hosts

Algorithm 2 shows how to place VMs on corresponding hosts. The number of candidate slots is no less than the number of VMs, so unmapped slots need to be reclaimed. This operation is also to withdraw those dummy VM requests.

Algorithm 2 Mapping partition

Input: VM partition set $\{V_1, V_2 \dots V_k\}$, slot-cluster set $\{c_1^{'}, c_2^{'} \dots c_k^{'}\}$
Output:

for $i = 1$ to k **do**
  for all not-dummy VM $v \in V_i$ **do**
    Map to corresponding slot $c_i^{'}$ and allocate required resources to each VM in the VM partition V_i
    Establish virtual links with other successfully mapped VMs
  end for
  if there exists unmapped slots in $c_i^{'}$ **then**
    Reclaim unmapped slots back to dependent host
  end if
end for

D. Effectiveness Analysis of Algorithm

The TRVMPO approach supports continual VMP requests. Upon the arrival of the $(i+1)$-th VMP request, previous i VMP requests have been handled, and those VMs of previous requests may be running their respective tasks. If we place whole VMs, including $(i+1)$-th request and all previous requests, we could get relatively better strategy that can save some bandwidth resources. However, it would result in VM migration and definitely impact on running VMs. It seems that overall reallocation benefits less, especially when the request frequency is relatively high.

Virtual Machine min k-cut and matching partitions with slot-clusters are core parts of the algorithm. These two parts are directly related to the effectiveness of the TRVMPO approach. For $k = r(n, K) \geq n/K$ and $2 \leq k \leq n$, the analysis in the study [26] indicates the approximation ratio for the *Gomory-Hu's* algorithm is $2 - 2/k$. So the approximation ratio η for the TRVMPO algorithm satisfies $1 \leq \eta \leq 2 - 2/n$. The TRVMPO algorithm reaches optimal when $\eta = 1$. In the non-worst case, η can be estimated as shown in the equation 8.

$$\eta \approx 2 - \frac{2K}{n} = 2 - \frac{2(1-R)\cdot m}{n} \tag{8}$$

Moreover, η can be represented as $\eta = 1 + \epsilon$, where ϵ is fixed and $0 \leq \epsilon < 1$. So the effectiveness of the TRVMPO approach is guaranteed.

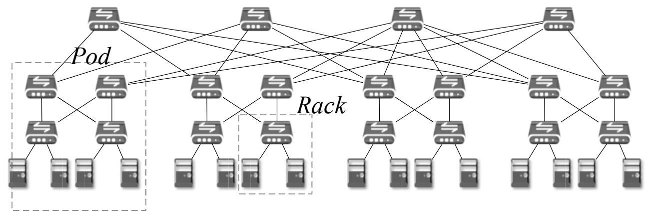
V. EXPERIMENTAL EVALUATIONS

A. Experiment Settings

In our experiment, we use the fat-tree datacenter network architecture which consists of three tiers of k-port switches and physical hosts, as shown in Figure 2, which also gives the communication cost matrix of all hosts.

We use 320 16-port switches and 1024 hosts to build a complete fat-tree datacenter network according to the above-mentioned description. Each VMP request includes 64 VMs. The bandwidth demand of each interconnected VM pair is randomly set in $[100, 500]$ Mbps. The reliability parameter ranges from 5% to 80%. We compare our TRVMPO algorithm with three existing VMP approximate algorithms:

- *BMKP.* [9] The BMKP is an approximate algorithm that can solve the VMP problem with large scale.
- *RA-2EM.* [16] The RA-2EM is a reliable VMP algorithm. Before VM mapping operation, RA-2EM will judge whether reliability could be guaranteed. If not, this operation will be postponed to the next.
- *RAVDC.* [16] Based on the Greedy Graph Growing Algorithm [17] algorithm, the RAVDC algorithm puts reliability-guarantee as a primary condition and bandwidth consumption as an optimization objective.



$$\begin{pmatrix} 0 & 1 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 1 & 0 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 3 & 3 & 0 & 1 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 3 & 3 & 1 & 0 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 0 & 1 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 1 & 0 & 3 & 3 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ \ldots \end{pmatrix}$$

Cost Matrix

Figure 2: Fat-tree data center network architecture.

B. Experiment Results and Evaluation

1) Practical Reliability: Figure 3 shows the practical reliability obtained by each algorithm. The result indicates that the BMKP can not adaptively satisfy the reliability requirement. The BMKP is based on balanced k-min cut algorithm and k is specified beforehand. The other algorithms can guarantee the reliability requirement.

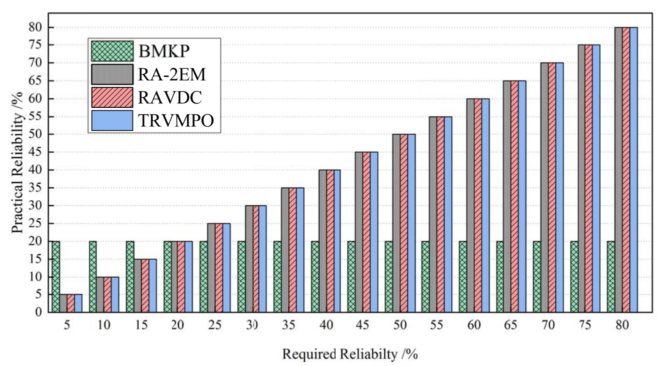


Figure 3: Comparison of practical reliability.

2) Time Cost: Figure 4 shows the average placement time of the RA-2EM, RAVDC and TRVMPO algorithm, with different reliability requirements. With R ranging from 5% to 80%, the time cost of the RA-2EM increases rapidly, but RAVDC and TRVMPO are affected less. When R is low, the VM integration threshold K is big. The GGGP algorithm in the RAVDC needs more time to find VM groups. And the TRVMPO algorithm can get better efficiency when R is in $[45\%, 75\%]$. In this condition, the size of each partition, or slot-cluster, is more suitable for the partition operation and the matching operation.

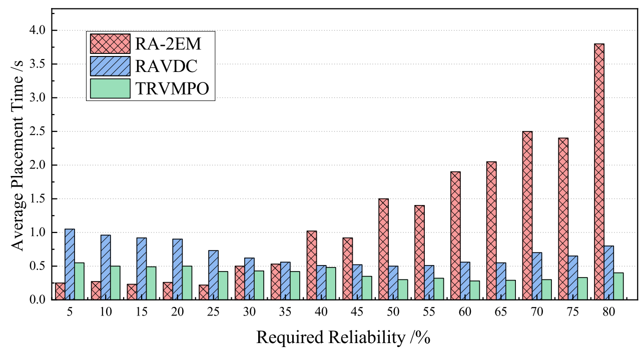


Figure 4: Comparison of average placement time.

3) Bandwidth Consumption: Figure 5 shows the bandwidth consumption of the RA-2EM, RAVDC, and TRVMPO, with different reliability requirements. No significant optimization can be observed in the RA-2EM algorithm. The TRVMPO algorithm and RAVDC algorithm can get the same objective value in some situations, because the placement strategies produced by these two algorithms are the same. Since there exists mapping failed cases in the RAVDC and no re-schedule operation will be utilized, the result will be easily nonoptimal. The TRVMPO algorithm transforms the placement problem into a matching problem. According to analysis in Section IV-D, the TRVMPO alogorithm can obtain VMP solution as optimal as possible with the approximation ratio $\eta = 1 + \epsilon$.

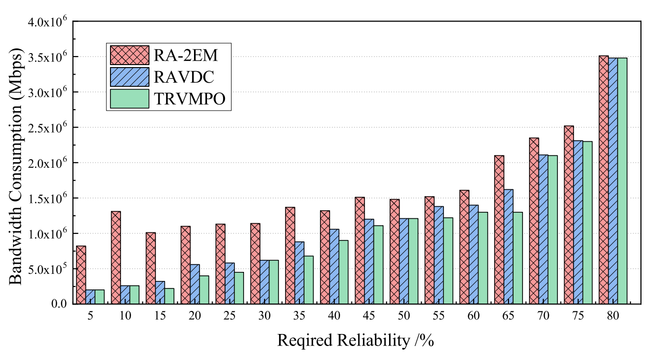


Figure 5: Comparison of bandwidth consumption.

4) Bandwidth Saving: Figure 6 presents the bandwidth saving of the RA-2EM, RAVDC, and TRVMPO algorithm, respectively, with the comparison with a random VMP approach [9], in which any VM can be allocated on any slot with equal probability. On account of the equation 4, the objective value of the random VMP approach can be computed as presented in the equation 9. So the saving percentage can be described as $(B_{rand} - B_t) * 100\%$, with $t \in \{\text{RA2EM, RAVDC, TRVMPO}\}$.

$$B_{rand} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} (BP^T)_{ij} \tag{9}$$

It can be seen from the data in Figure 6, the performance of the RA-2EM fluctuates randomly. Although the RA-2EM can guarantee the required reliability, it can not assuringly obtain much significant optimization, even worse than the random VMP approach. Strong evidence shows that the RAVDC and TRVMPO algorithm can guarantee the condition $B_{rand} \geq B$ and the average percentage of bandwidth saving is positive. The result also shows when R is low, these two algorithms can both conduct a relatively compact VMP scheme which can greedily save the bandwidth consumption. As analysis mentioned above, the TRVMPO algorithm can be more beneficial than the RAVDC sometimes. When R increases, the optimal scheme can save relatively less bandwidth due to the small VM integration threshold K.

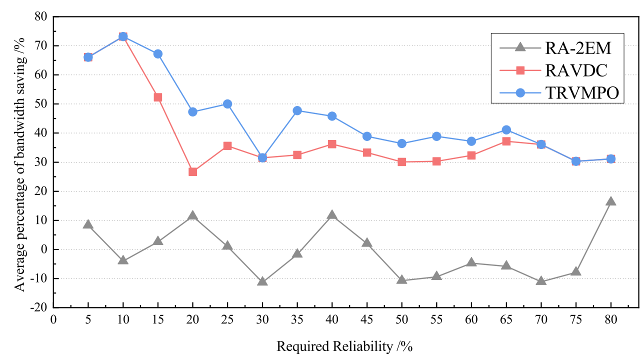


Figure 6: Bandwidth saving.

VI. Conclusion and Future Work

In this paper, we proposed a TRVMPO algorithm for optimizing bandwidth consumption with reliability guaranteed in VMP problem in cloud datacenters. We formally described a reliability model and a bandwidth consumption model, and deduced the optimization objective function. Leveraging both theoretical demonstration and experimental evaluations on different settings, our proposed approach was proved effective and efficient in solving traffic-aware and reliability-guaranteed VMP problems. Moreover, the TRVMPO algorithm can also be exploited in similar resource assignment problems with other optimization objective such as energy consumption of physical nodes.

For the future work, we will attempt to adopt some approaches to enhance reliability of VMs, such as VM backups and VM migration. More constrained conditions and optimization objectives will be collectively considered, for making VMP more efficient in cloud datacenters.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No.61772479), and the National Key Research and Development Program of China (2017YFB1400603).

References

[1] Y. Ma, S. Wang, P. C. K. Hung, C. Hsu, Q. Sun, and F. Yang, "A highly accurate prediction algorithm for unknown web service qos values," *IEEE Trans. Services Computing*, vol. 9, no. 4, pp. 511–523, 2016.

[2] E. Bauer and R. Adams, "Reliability and availability of cloud computing," *Wiley John + Sons*, 2012.

[3] 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 *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09, 2009, pp. 51–62.

[4] Cisco global cloud index: Forecast and methodology, 2015-2020. [Online]. Available: http://www.cisco.com

[5] J. Liu, S. Wang, A. Zhou, F. Yang, and R. Buyya, "Availability-aware virtual cluster allocation in bandwidth-constrained datacenters," *IEEE Transactions on Services Computing*, pp. 1–1, 2018.

[6] Vmware capacity planner. [Online]. Available: http://www.vmware.com/products/capacityplanner/

[7] Ibm websphere cloudburst. [Online]. Available: http://www-01.ibm.com/software/webservers/cloudburst/

[8] K. Li and J. Nabrzyski, "Traffic-aware virtual machine placement in cloudlet mesh with adaptive bandwidth," in *IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2017, Hong Kong, December 11-14, 2017*, 2017, pp. 49–56.

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

[10] 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 *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09, 2009, pp. 39–50.

[11] 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," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, ser. SIGCOMM '09, 2009, pp. 63–74.

[12] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R. N. Chang, M. R. Lyu, and R. Buyya, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Trans. Services Computing*, vol. 10, no. 6, pp. 902–913, 2017.

[13] A. Zhou, Q. Sun, and J. Li, "Enhancing reliability via checkpointing in cloud computing systems," *China Communications*, vol. 14, no. 7, pp. 1–10, July 2017.

[14] S. Yang, P. Wieder, R. Yahyapour, S. Trajanovski, and X. Fu, "Reliable virtual machine placement and routing in clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 10, pp. 2965–2978, 2017.

[15] P. Datta and A. K. Somani, "Graph transformation approaches for diverse routing in shared risk resource group (SRRG) failures," *Computer Networks*, vol. 52, no. 12, pp. 2381–2394, 2008.

[16] C. Zuo, "Study of multi-tenant data center network based on sdn," *Master Thesis Submitted to University of Electronic Science and Technology of China*, 2016.

[17] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[18] X. Ye, Y. Yin, and L. Lan, "Energy-efficient many-objective virtual machine placement optimization in a cloud computing environment," *IEEE Access*, vol. 5, pp. 16 006–16 020, 2017.

[19] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[20] T. H. Duong-Ba, T. Nguyen, B. Bose, and T. T. Tran, "A dynamic virtual machine placement and migration scheme for data centers," *IEEE Transactions on Services Computing*, pp. 1–1, 2018.

[21] X. F. Liu, Z. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Trans. Evolutionary Computation*, vol. 22, no. 1, pp. 113–128, 2018.

[22] C. Gao, H. Wang, L. Zhai, Y. Gao, and S. Yi, "An energy-aware ant colony algorithm for network-aware virtual machine placement in cloud computing," in *22nd IEEE International Conference on Parallel and Distributed Systems, ICPADS 2016, Wuhan, China, December 13-16, 2016*, 2016, pp. 669–676.

[23] Xuemin Wen, Yanni Han, Bing Yu, X. Chen, and Zhen Xu, "Improving revenue for reliability-aware vdc embedding in a software-defined data center," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, June 2016, pp. 1–2.

[24] E. M. Loiola, N. M. M. de Abreu, P. O. B. Netto, P. Hahn, and T. M. Querido, "A survey for the quadratic assignment problem," *European Journal of Operational Research*, vol. 176, no. 2, pp. 657–690, 2007.

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

[26] R. Gomory and T. Hu, "Multi-terminal network flows," *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, no. 4, pp. 551–570, 1961.