研究与开发

一种超额认购虚拟数据中心的嵌入算法

鹿楚坤，闫芳芳，李东

（上海交通大学区域光纤通信网与新型光通信系统国家重点实验室，上海 200240）

摘　要：多租户数据中心环境下，保证云应用性能的一个重要因素是为租户应用提供可保证的通信带宽，这可以通过为每个租户提供一个独占的虚拟数据中心（VDC）来实现。研究了在物理数据中心网络中超额认购数据中心的嵌入问题。相对于一般虚拟数据中心，超额认购虚拟数据中虚拟机之间的流量模式更加复杂，因此首先利用线性规划方程阐述了流量模型及嵌入问题。对于虚拟机嵌入问题，提出了一种具有较低时间复杂度的启发式算法——分组扰动算法。最后，通过仿真实验将分组扰动算法和先前工作中提出的算法以及著名的 first-fit 进行了比较，实验表明所提算法在降低算法复杂度的同时提高了嵌入成功率。

关键词：数据中心；虚拟化；嵌入算法；扰动

中图分类号：TN919　　**文献标识码**：A

doi: 10.11959/j.issn.1000−0801.2017104

An algorithm for embedding oversubscribed virtual data center

LU Chukun, YAN Fangfang, LI Dong

State Key Laboratory of Advanced Optical Communication Systems and Networks,
Shanghai JiaoTong University, Shanghai 200240, China

Abstract: Predictable network performance is critical for cloud applications and can be achieved by providing tenants a dedicated virtual data center (VDC) with bandwidth guarantee. The embedding problem of oversubscribed data center in physical data center network was studied. Compared with the general virtual data center, the traffic pattern between the virtual machines in the over-subscription virtual data was more complicated. Therefore, the flow model and the embedding problem were described. A heuristic algorithm with lower time complexity was proposed for the embedding problem of virtual machine-packet perturbation algorithm. Finally, the simulation algorithm was used to compare the packet perturbation algorithm with the algorithm proposed in the previous work and the famous first-fit. The experiment result shows that the proposed algorithm improves the embedding success rate while reducing the complexity of the algorithm.

Key words: data center, virtualization, embedding algorithm, perturbation

1　引言

随着云计算的兴起，关于虚拟化技术在数据中心的应用研究也在迅速增加。虚拟化技术允许将多台虚拟机放置在同一台物理服务器中，并与其他虚拟机共同竞争物理数据中心的网络资源以完成彼此之间的通信[1]。但当属于不同租户的虚拟机竞争同一数据中心网络时，网络带宽的变化

收稿日期：2017−01−17；修回日期：2017−04−01

以及不可预期的数据分组丢失可能会导致无法为租户提供可保证的网络性能[2]。为不同的租户提供一个独占的虚拟数据中心（virtual data center，VDC），可以为同一租户不同虚拟机之间提供可保证的通信带宽，进而保证应用的性能。虚拟数据中心是指通过虚拟链路连接的虚拟机集合，虚拟机的数量以及虚拟机之间的通信链路带宽由租户指定。

在物理数据中心嵌入一个虚拟数据中心，包括虚拟机的放置以及带宽分配，即路由问题。Ballani 等人[3]提出了两种基于 Hose 模型的虚拟数据中心结构：虚拟集群（virtual cluster）和虚拟超额认购集群（virtual oversubscribed cluster，VOC），并在 Oktopus 系统实现了该结构。参考文献[4,5]给出了基于 Hose 模型的工作成果。这些研究组提出的嵌入算法均是基于嵌入 Hose 模型虚拟数据中心至单根树拓扑，两台服务器之间只能通过一条路径通信。但目前已经被提出的以及广泛使用的数据中心网络拓扑中，两台服务器之间一般存在多条通信路径。不同于在单根树拓扑中嵌入虚拟数据中心，多路径网络拓扑中的嵌入问题更加复杂，因为这涉及流量分解以及多路径路由的问题。事实上，虚拟数据中心的嵌入问题已经被证明为 NP 完全问题[5]。在之前的工作中[6]，提出了一种将虚拟集群嵌入 VL2 和 fat-tree（胖树）[7]等主流物理数据中心拓扑的启发式算法。

目前大部分运行在数据中心之上的云应用通常是以组件的形式组织起来的，同时各个组件内的流量要大于各组件之间的流量[1]。超额认购虚拟数据中心为两层树型结构，上层交换机与下层交换机的链路带宽小于虚拟机与下层交换机之间链路带宽之和，引入了带宽的超额认购。一个超额认购虚拟机数据中心更能有效应对这类应用的通信需求，并降低租户购买虚拟数据中心的费用。但是带宽的超额认购使得虚拟机之间的通信流量更加复杂，之前提出的算法无法有效处理这种复杂的流量特性，从而使得性能下降，同时时间复杂度较高无法适应数据中心流量快速变化的特性[8]。因此在本文中，将之前的工作扩展至超额认购虚拟数据中心嵌入问题，提出了一种多项式时间复杂度的启发式分组扰动算法，相关工作见表 1。分组扰动算法基于分治策略设计，可分为两个阶段。在第一阶段首先得到一个初步的放置方案，并在第二阶段对该放置方案进行调整。因此该算法可以视为一种将虚拟机放置与带宽分配分离的嵌入算法。同时该算法是拥塞感知的，因为当探测到拥塞发生时，扰动将会被触发以减少拥塞链路的负载。

分组扰动启发式放置算法基于以下原则设计。

- 基于分治策略，首先将物理网络分组并将每个虚拟集群的放置范围限制在对应的物理分组中；其次，在第一阶段只考虑组内虚拟机的流量约束，得到一个“初步可行”的放置方案；在第二阶段考虑全部的流量约束，并对第一阶段得到的放置方案进行调整。
- 若在第二阶段检测到发生拥塞，扰动算法首先在物理网络中寻找热点（拥塞最严重）链路，寻找对该链路流量贡献最大的服务器，并减少其中放置的虚拟机数目。

2　问题描述

根据路由方案的不同，典型的数据中心网络

表 1　与相关工作的比较

研究工作	虚拟数据中心请求	物理数据中心网络	主要贡献
Ballani 等人[3]	虚拟集群	树型拓扑	给出了虚拟集群和超额认购虚拟群的定义，贪婪算法
Kawamura 等人[4]	虚拟集群	树型拓扑	NP 难问题证明，动态规划算法
Zhu 等人[5]	虚拟集群	树型拓扑	NP 完全问题证明
本文工作	超额认购虚拟集群	树型拓扑以及一般拓扑	多项式时间复杂度的分组扰动启发式算法

结构可以分为两类：一类是以交换机为中心的数据中心网络；另一类是以服务器为中心的数据中心网络[7]。典型的以交换机为中心的数据中心网络拓扑包括 fat-tree 和 VL2[7]。在以服务器为中心的数据中心网络拓扑中，服务器同时承担路由的功能。典型的以服务器为中心的数据中心网络包括 Bcube 和 Dcell[7]。一个以交换机为中心或者服务器为中心的物理数据中心网络，均可以用 $G(V,V_S,E,A(V_S),C(E))$表示该数据中心网络，其中 V 表示数据中心网络中的节点集合，$V_S \subseteq V$ 表示服务器的集合，E 表示网络链路的集合，$A(V_S)=\{a_j|j \in V_S\}$、$C(E)=\{c_e|e \in E\}$分别表示服务器和链路的剩余容量。使用$\{1,\cdots, |V_S|\}$标记$|V_S|$个服务器，相应地可以将交换机标记为$\{|V_S|+1,\cdots, |V|\}$。服务器 $j \in V_S$ 存在 a_j 个剩余空间可以用于放置新的虚拟机，c_e 表示物理链路 $e \in E$ 的剩余带宽。本文假定物理链路的剩余带宽在嵌入算法返回解决方案之前不发生变化。根据虚拟超额认购抽象模型，一个超额认购虚拟数据中心请求可以表示为向量 $r:(N,S,B,O)$[3]，其中 N 表示本次请求的虚拟机总数，S 表示每组有 S 台虚拟机，B 表示每组内的虚拟机通过带宽为 B 的链路连接至一个虚拟组交换机（group switch），O 为超额认购比（oversubscription factor），表示虚拟组交换机通过带宽为 $B'=\dfrac{S\times B}{O}$ 的链路连接至一个虚拟根交换机（root switch）。基于 Hose 模型的虚拟超额认购集群抽象模型[3]如图 1 所示。

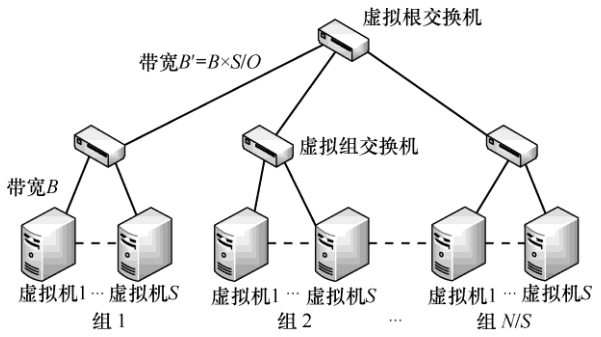


图 1　基于 Hose 模型的虚拟超额认购集群抽象模型

在物理数据中心中嵌入虚拟数据中心是指在物理数据中心中寻找合适的服务器放置这些，同时在放置虚拟机的服务器之间的路径上预留带宽，以保证虚拟机之间可以得到它们请求的通信带宽。因此一个虚拟数据中心嵌入问题的可行解决方案应该包括虚拟机的放置方案和相应的带宽分配方案，分别对应于虚拟机的放置问题和流量路由问题。虚拟机的放置问题是指不考虑带宽限制条件时将所有 N 个虚拟机放置于$|V_S|$服务器。令 $\Pi^r=\{\pi(i,j)\,|\,i=1,\text{L}\ ,N;\ j=1,\text{L}\ ,|V_S|\}$ 表示一次虚拟数据中心请求的虚拟机放置方案，其中放置方案变量 $\pi(i,j)=1$ 表示虚拟机放置在服务器中。对于一次虚拟数据中心请求，一个可行的放置方案应该满足如下约束。

每台虚拟机只能放置于一台服务器中：

$$\sum_{j=1}^{|V_S|}\pi(i,j)=1 \qquad (1)$$

放置于服务器的虚拟机总数不能超过该服务器的剩余容量：

$$\sum_{i=1}^{N}\pi(i,j)\leqslant a_j \qquad (2)$$

类似基于 Hose 模型的 VPN 路由问题[10]，对于一个给定的虚拟机放置方案，流量路由问题是指在满足服务器入口和出口带宽限制条件下寻找满足所有流量矩阵的可行路由方案。令 $Q(\Pi^r)=\{j\,|\sum_{i=1}^{N}\pi(i,j)>0,\ j=1,\text{L}\ ,|V_S|\}$表示至少放置一台虚拟机的所有服务器的集合，同时定义 $G_m(g)=\{g|\sum_{i=(m-1)\times s+1}^{N}\neq 0$，$g=1,\cdots,|V_S|\}$表示放置有第 m 组虚拟机的服务器集合。令 $b(s)$表示进入或者离开第 m 组服务器 s 的总流量。根据 Oktopus 系统[3]，$b(s)$由该服务器装载的虚拟机数量决定，同时根据虚拟超额认购集群抽象模型，$b(s)$可分为两部分：一部分为服务器 s 与放置有同组虚拟机服务器之间的流量 $b_{\text{intra}}(s)$；另一部分为服务器 s 与

放置有不同组虚拟机服务器之间的流量 $b_{\text{inter}}(s)$。

对于第一部分流量，$b_{\text{intra}}(s)$受放置于服务器的虚拟机请求的总带宽以及同一组内其他虚拟机请求的总带宽约束：

$$b_{\text{intra}}(s)=\min\left(\sum_{i=1}^{N}\pi(i,s)\times B,(S-\sum_{i=1}^{N}\pi(i,s))\times B\right),\quad s\in Q(\Pi^r) \tag{3}$$

对于第二部分流量，$b_{\text{inter}}(s)$受放置于服务器 s 的虚拟机请求的总带宽以及不同组虚拟机之间的通信带宽 B' 的约束：

$$b_{\text{inter}}(s)=\min\left(\sum_{i=1}^{N}\pi(i,s)\times B,B'\right) \tag{4}$$

同时出入服务器 s 的总流量满足：

$$b(s)=\min\left(\sum_{i=1}^{N}\pi(i,s)\times B,B'+(S-\sum_{i=1}^{N}\pi(i,s))\times B\right) \tag{5}$$

令 t_{sd} 表示从服务器 s 到服务器 d 的流量并且定义 t_{ss}=0 对于一个给定的虚拟机放置方案 Π^r，一个合法的流量矩阵[t_{sd}]应该满足如下约束：

$$\sum_{d\in G_m(g)} t_{sd}\leqslant b_{\text{intra}}(s) \tag{6}$$

$$\sum_{s\in G_m(g)} t_{sd}\leqslant b_{\text{intra}}(d) \tag{7}$$

$$\sum_{d\notin G_m(g)} t_{sd}\leqslant b_{\text{inter}}(s) \tag{8}$$

$$\sum_{s\notin G_m(g)} t_{sd}\leqslant b_{\text{inter}}(d) \tag{9}$$

$$\sum_{d\in Q(\Pi^r)} t_{sd}\leqslant b(s) \tag{10}$$

$$\sum_{s\in Q(\Pi^r)} t_{sd}\leqslant b(d) \tag{11}$$

放置第 m 组虚拟机的服务器 S 与其他放置虚拟机的服务器之间的通信流量由式（6）~式（11）约束决定。与 VPN 中 Hose 模型的对称性定义一样，这里进入某一服务器的流量和离开该服务器的流量相等。令 $T(\Pi^r)$表示在虚拟机放置方案 Π^r 下，由式（6）~式（11）约束可以得到的本次虚拟数据中心中虚拟机之间所有可能的流量矩阵。对于多路径路由问题，两个服务器之间的流量可以在它们之间的多条路径任意分配。每条路径可以包含多个交换机或者服务器。定义路由变量 f_{sd}^e 表示服务器 s 与服务器 d 之间的流量在链路 e 上分配的比例，其取值范围为 0 到 1。显然链路 e 的流量负载不能超过该条链路的剩余带宽 c_e，可表示为如下约束：

$$\sum_{s,d\in Q(\Pi^r)} t_{sd} f_{sd}^e\leqslant c_e, e\in E \tag{12}$$

通过计算每条链路的最大负载并判断其是否超过该链路的剩余带宽，可以验证某一路由方案 $F(\Pi^r)=\{f_{sd}^e \mid s,d\in Q(\Pi^r), e\in E\}$的可行性。定义 $u_e(\Pi^r)$为在放置方案 Π^r 下，链路 e 上的最大负载。给定虚拟机放置方案 Π^r 和流量路由方案 $F(\Pi^r)$，最大链路负载 $u_e(\Pi^r)$可由式（6）~式（11）约束的 LP 方程得到：

$$\max_{t_{sd}} \sum_{s,d\in Q(\Pi^r)} t_{sd} f_{sd}^e \tag{13}$$

为保证在虚拟机数据中虚拟机之间提供可保证的通信带宽，对于一个已经嵌入物理数据中心的虚拟数据中心，虚拟机之间通信涉及的每条物理链路都需要预留通过式（6）~式（11）、式（13）计算得到的最大带宽。根据 Karmarkar 算法[11]，计算一条链路的最大负载时间复杂度为 $O(\min(|V_s|,N)^{3.5}L^2)$，其中$|V_s|$为网络中服务器的数量，$N$ 为虚拟数据中心中虚拟机的数目，L 为输入的比特数。

3　启发式嵌入算法

综上，讨论了在给定虚拟机放置方案和路由方案的情况下，计算最大链路利用率的问题。如前所述，对于到来的一个虚拟数据中心嵌入请求，嵌入问题可分为流量路由问题和虚拟机放置问题两个子问题。对于流量路由问题，之前的工作已经得到了一种性能优越的多路径路由算法，即 K-最宽路径负载均衡路由算法[6]。应用该算法求解

路由问题，本节将介绍求解虚拟机放置问题的启发式算法。

3.1 分组算法

基于虚拟超额认购集群模型的虚拟数据中心，利用数据中心流量局部性的原理，构造带有超额认购的二层网络。组内的虚拟机通信流量大于组间虚拟机的通信流量。现代数据中心网络拓扑多为带超额认购的多层拓扑，局部的网络带宽资源更加丰富。在将虚拟机数据中心嵌入进物理网络时，利用超额认购虚拟数据中心流量局部性的特点以及物理网络局部带宽丰富的特点，将一组内的虚拟机放置在网络的同一区域。因此分组扰动算法首先会将物理数据中心分组，将一组虚拟机的嵌入限制在物理数据中心网络的一个子网范围之内。

因为虚拟机在同组之内的通信需求大于与其他组虚拟机的通信需求，因此分组算法得到的子网应该使得子网之内的物理机通信费用小于与其他子网中物理机的通信费用。目前广泛应用的数据中心网络拓扑可以归结为如图 2 所示的多根树拓扑。以服务器为顶点，服务器之间的通信费用（本文定义为服务器之间最短路径的跳数）作为连接顶点的边权重，顶点的权重为服务器的剩余容量，构造出辅助完全加权图 G^{aux}。

例如，图 2（a）服务器 s_1 剩余容量为 2，映射到完全图的顶点权重为 2。s_1 与 s_4 之间通信费用为 3 跳，映射到完全图（b）为连接顶点的边权重为 3。物理网络分组转化为辅助完全图的分割问题可描述为：求解将物理网络分成 k 组（即 k 组子网）的划分方案，满足以下约束。

- 物理网络分组之后，每个组内物理机之间通信费用之和应该最小，即子图边权重之和应该最小。
- 保证每个子网存在足够的容量放置虚拟机集群中一个组的 S 台虚拟机，即子图顶点的权重之和大于一组虚拟机数量 S，如下：

$$\min \sum_{l=1}^{k} w(V_l), w(V_l) = \sum_{i,j \in V_l}^{k} w_{ij} \tag{14}$$

$$\text{s.t.}\ \ V_1 \cup V_2 \cup \cdots \cup V_k = V \tag{15}$$

$$V_i \cap V_j = \phi, i \neq j \tag{16}$$

$$S \leqslant c(V_l) \leqslant \left\lceil \frac{c(V)}{k} \right\rceil, c(V_l) = \sum_{i \in V_l} c(i) \tag{17}$$

其中，$V_1, V_2, \cdots, V_j, 1<j<\frac{N}{s}$ 为最终得到的子图，代表物理网络子网，$w_{i,j}$ 为边权重，代表服务器之间的通信费用，$w(V_l)=\sum_{i,j \in V_l}^{k} w_{ij}$ 表示子图边权重之和，代表子网络内服务器之间的通信费用。$c(V_l)=\sum_{i \in V_l} c(i)$ 表示子图的顶点权重之和，代表对应子网络中剩余容量的总量，其中 $c(i)$ 为顶点权重，表示服务器的剩余容量。该图的分割问题可以归结为求解规模限制的聚类（size-constrained clustering）问题[12]。参考文献[12]总结了目前该问题的研究进展以及求解该问题一系列算法，例如 KaHIP 算法、LPA 算法。但这些算法的时间复杂

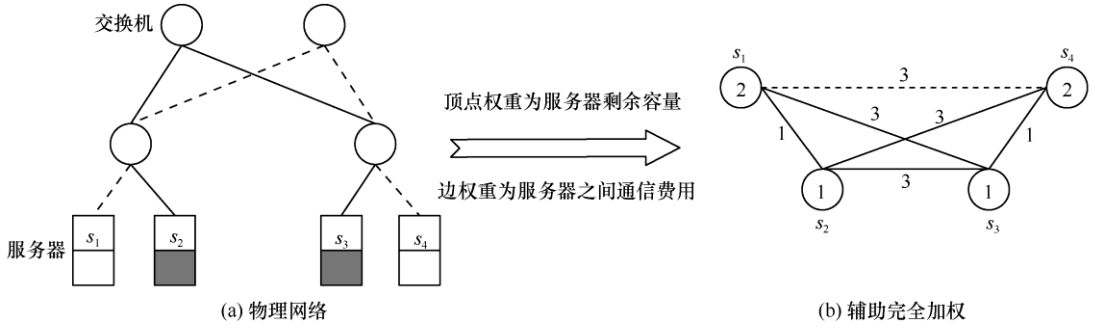


图 2 多根树拓扑

度为指数时间复杂度。参考文献[13]给出了无规模限制聚类问题的一个算法，时间复杂度为 $O(|V|k)$。该算法分为初始化阶段与（k–1）个后续阶段。在初始化阶段，将所有待分组顶点放进一个顶点集 V_1 中，任选一顶点标记为 V_1 的首顶点。在第 j 个后续阶段，将$(V_1,V_2,\cdots,V_j)$中距离所在集合首顶点最远的顶点 v_i，移入 V_{j+1} 并标记为首顶点。所有距离 v_i 小于所在集合首顶点的顶点将被移入 V_{j+1}。在该算法结束之后，规模限制条件 $S \leqslant c(V_l) \leqslant \left\lceil \frac{c(V)}{k} \right\rceil$ 并不一定对求得的所有顶点集满足。因此需要添加一个平衡阶段。首先定义 $c(V_l) > \left\lceil \frac{c(V)}{k} \right\rceil$，称该顶点集过载；若顶点 v 移入 V_p，不会导致顶点集过载，则称该顶点集 Vp 为顶点 v 可行的顶点集。在平衡阶段，设过载的顶点集 V_l 中距离首元素最远的顶点为 v，若 v 存在多个可行顶点集，将 v 移入其可行顶点集 V_l 中距离首元素 v 最近的顶点集。

3.2　虚拟机放置算法

即使是在简单树拓扑中的虚拟数据中心，嵌入问题便已经被证明为 NP 完全问题，因此提出了一种多项式时间复杂度的启发式分组扰动算法来解决这一难题。分组扰动算法分为两个阶段，第一阶段为虚拟机放置阶段，在该阶段根据若干流量矩阵判断是否符合组内流量的带宽需求，得到一个“初步可行”的虚拟机放置方案。第二阶段为带宽分配以及链路负载检查阶段，为第一阶段得到的虚拟机放置方案预留带宽，并检查是否存在拥塞链路。当在第二阶段由于链路拥塞导致路由分配失败时，虚拟机放置的扰动策略将会被触发。扰动算法通过修改现有的放置方案，减少对拥塞链路流量贡献最大的服务器上的虚拟机数目，来消除拥塞。

首先介绍在嵌入算法中使用的变量。定义 X^r 表示所有未被放置的虚拟机的集合，其初始值为 $\{VM_1,VM_2,VM_3,\cdots,VM_N\}$。定义优先级 $f_{\sum}[s]$用于量化服务器 s 对于网络拥塞的流量贡献，并对于所有的服务器 s=1,2,⋯,$|V_s|$，该值初始化为 0。扰动算法将移除具有最高优先级指标的服务器上的虚拟机，并以该服务器为中心，寻找服务器放置被移除的虚拟机。若有多个服务器具有相同的优先级，将从中随机选择一个。

图 3 列出了分组扰动算法第一阶段的详细过程。基于分治策略，在该阶段首先调用第 3.1 节介绍的分组算法，将物理网络分为 M 个子网络 $\{G_1^{\text{aux}},G_2^{\text{aux}},\text{L},G_M^{\text{aux}}\}$，并将每个虚拟机组的放置范围限制在对应的物理网络组中，调用图 4 描述的组内放置算法计算每个虚拟机集群在对应的物理网络组内的放置方案。对于放置失败的虚拟机组，将在剩余的数据中心网络中继续该算法直至所有的虚拟机组放置成功，或者在一次循环中没有新的虚拟机组被放置，则返回失败。

图 4 给出了组内放置算法的详细过程。在放置算法开始之前，利用负载均衡路由算法[6]，计算路由变量 f_{sd}^e。然后对子网络中的每个服务器尝试放置虚拟机，首先尝试放置虚拟机填满服务器，若检测到拥塞发生，则减少放置在该服务器中的虚拟机数目直到找到该服务器可以放置的虚拟机最大数量。在寻找每台服务器可以放置的虚拟机最大数量时，只考虑由 $b_{\text{intra}}(s)$表示的每个虚拟机组之内的通信需求。由于线性规划在边界处取得最大值，因此选择使用置换流量模式（permutation traffic pattern）[14]替代由式（6）、式（7）得到的所有可行的流量矩阵来计算链路是否发生拥塞，降低算法复杂度，适应数据中心快速变化的流量。在置换流量模式下，一个源节点 s 发送其所有流量至另一个单一节点 d。假设在一次放置过程中，已经有 4 台服务器<s_1, s_2, s_3, s_4>放置有虚拟机，左移该向量作为流量的目的节点<s_4, s_3, s_2, s_1>，即 s_1 发送全部流量至 s_4。这样可以得到式（18）所示的第一个流量矩阵，在该矩阵中元素 t_{ab} 表示服务

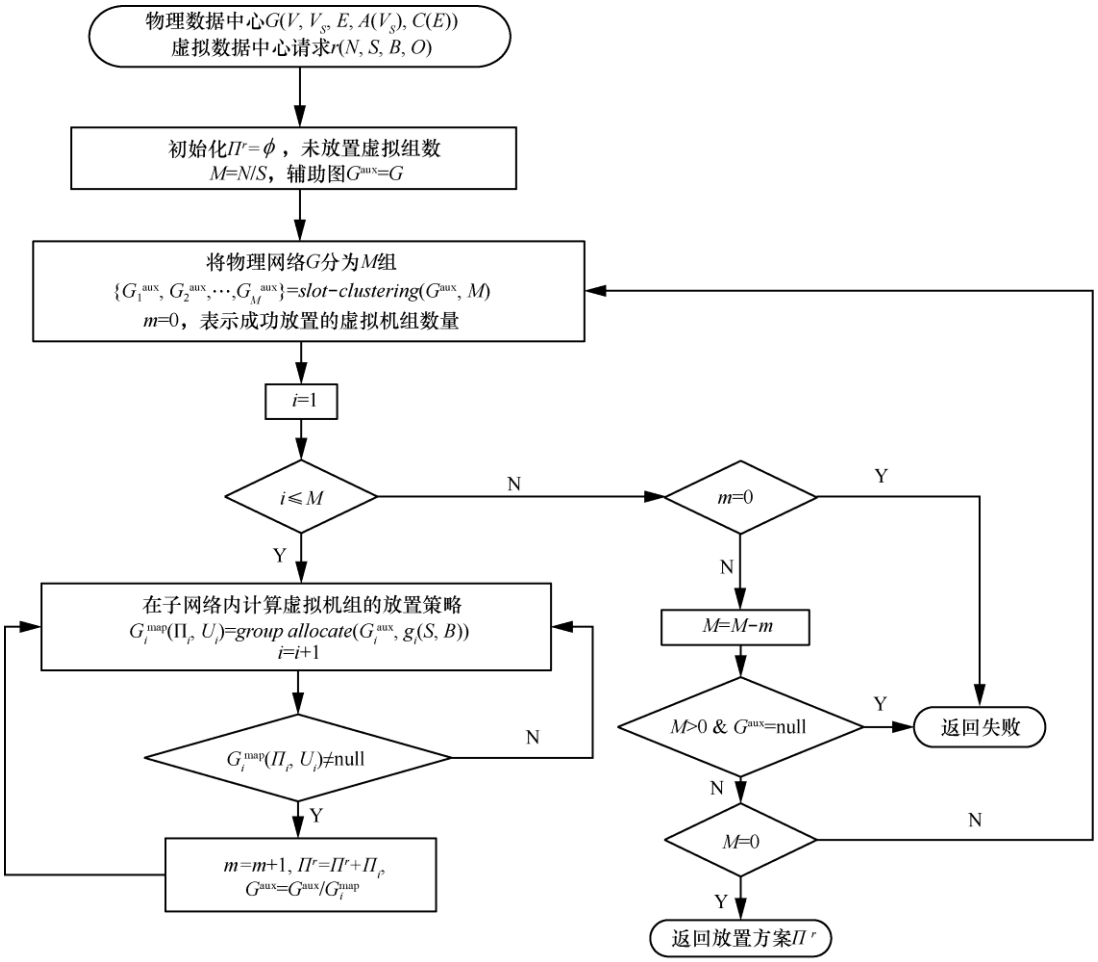


图 3　虚拟机放置算法（算法 1）

器 a 发送至服务器 b 的流量比例。依次右移寻找目的节点可以得到如下 3 个置换流量模式：

$$\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&1\\1&0&0&0\\0&1&0&1\\0&0&1&0\end{bmatrix}\qquad(18)$$

在前面的章节已经得到出/入服务器 1 与服务器 4 的总流量分别为 $b(1)$、$b(4)$。在第一个矩阵中，可以得到服务器 s_1 将流量发送到了服务器 s_4，因此 $t_{14}=\min(b(1),b(4))$。

通过计算 $\sum_{s,d\in Q(\Pi^r)} t_{sd} f_{sd}^e$ 得到链路的最大负载，分别计算 3 个流量矩阵下的负载，比较求出链路的最大负载。这样避免了之前工作[6]通过多次求解 LP 计算链路负载导致计算量大，无法适应数据中心流量快速变化的问题。在该组虚拟机放置完之后，检查链路的负载，即对子网络里的每条链路，通过解 LP 方程式（6）、式（7）、式（13）计算该链路的最大负载 $u_i(e)$。若 $u_i(e)<c(e)$，说明无拥塞发生，该组虚拟机放置成功并返回该组虚拟机放置方案以及需要预留的带宽 $G_i^{map}(\Pi_i,U_i)$。同时将该子网络 G_i^{aux} 中的服务器标记为该组虚拟机的可选服务器 S_i={服务器 $i|i\in G_i^{aux}$ }，为之后的放置方案调整做准备；若 $u_i(e)>c(e)$，说明链路发生拥塞，返回该组虚拟机放置失败。对于放置失败的虚拟机组，在更新链路带宽与服务器容量之后的数据中心网络 $G^{aux}\leftarrow G^{aux}/G_i^{map}$ 中继续该算法。

由上所述，在第一阶段求解虚拟机放置方案

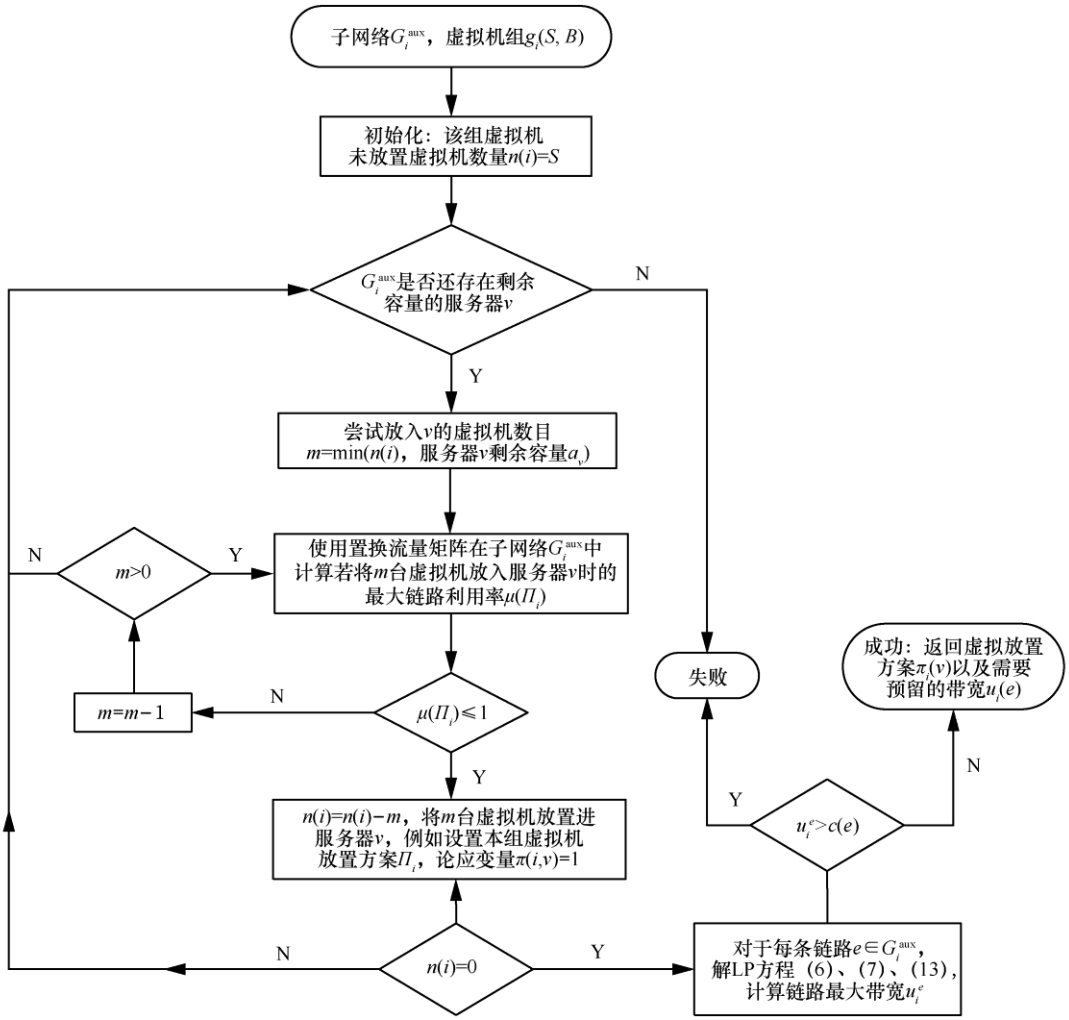


图 4　无超额认购情况下的放置算法（算法 2）

的过程中，只考虑了组内的通信需求，未考虑虚拟机组之间的通信需求，因此在第一阶段得到的虚拟机放置方案可以认为是“初步可行”的放置方案。在分组扰动算法的第二阶段，考虑不同虚拟机组之间的通信需求，对第一阶段得到的虚拟机放置方案做进一步的检查。若拥塞发生，则会触发扰动对现有的虚拟机放置方案做出调整。通过求解式（6）~式（13）给出的 LP 方程，得出物理网络在当前虚拟机放置方案下的最大链路利用率 $\mu(\Pi^r)$。若 $\mu(\Pi^r)>1$，表明检测到物理链路发生拥塞，则认为当前的虚拟机放置方案 $\mu(\Pi^r)$ 是一个不可行的放置方案。此时，应用扰动策略调整当前的放置方案。扰动策略通过调整已经放置好的虚拟机位置来减轻拥塞链路上的流量负载。

计算优先级 $f_{\sum}[s]$时，对于一个不可行的虚拟机放置方案 Π^r，在物理网络中，寻找具有最大链路利用率的拥塞链路：

$$\hat{e}(\Pi^r)=\arg\max_{e\in E}\frac{u_e(\Pi^r)}{c_e} \tag{19}$$

在之前，引入 $Q(\Pi^r)$表示当前虚拟机放置方案下，放置有至少一台虚拟机的服务器集合。对于任意一台服务器 $s\in Q(\Pi^r)$，优先级 $f_{\sum}[s]$为服务器 s 与其他服务器之间的路径上所有热点链路 $\hat{e}(\Pi^r)$ 的路由变量 $f_{sd}^{\hat{e}(\Pi^r)}$ 之和；对于其他服务器 $s\notin Q(\Pi^r)$，优先级 $f_{\sum}[s]$为 0。瓶颈服务器 $\hat{v}$ 为发送流量至拥塞链路最多的服务器。因此瓶颈服务

器 $\hat{v}$ 可以通过式（20）计算得到：

$$\hat{v} = \arg \max_{s \in Q(\Pi^r)} f_{\Sigma}[s] \quad (20)$$

带宽分配及扰动（算法 3）如图 5 所示，扰动将移除瓶颈服务器 $\hat{v}$ 上的虚拟机并放置在该虚拟机对应的可选服务器 S_i 中对拥塞链路贡献最小的服务器 $\hat{v}$ 上。

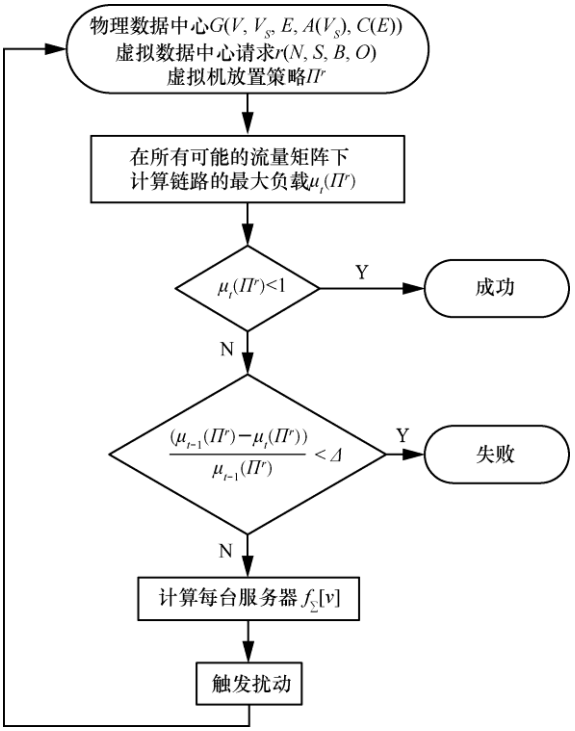


图 5　带宽分配及扰动（算法 3）

每次虚拟机移动的过程中，以虚拟机移动前和移动后网络的最大链路利用率的变化率 $\frac{(\mu_{t-1}(\Pi^r)-\mu_t(\Pi^r))}{\mu_{t-1}(\Pi^r)}$ 为标准，判断本次虚拟机的移动是否有效减轻了物理网络的拥塞。若小于某阈值 Δ，则认为本次调整未能有效降低网络负载，物理网络无法承载本次的虚拟数据中心嵌入请求，返回失败。同时使用 max_*perturbation* 控制循环的上限。

分组扰动算法在第一阶段得出虚拟机放置算法，然后在第二阶段进行路由分配以及放置方案的调整，可以认为是一种将虚拟机放置与路由分配分离的嵌入算法。分组扰动算法的时间复杂度分析如下。在第一阶段中的算法 1，对物理网络分组的时间复杂度为 $O(G_n|V|)$[13]，其中 $G_n=\frac{N}{S}$ 表示组数，$|V|$表示网络中的节点数。该算法的最大循环次数为 G_n。算法 1 中最耗费时间的是调用算法 2 的过程。调用算法 2 的时间复杂度为 $O(S|V|\min\left(\frac{|V_s|}{G_n},\frac{N}{G_n}\right)^{3.5} L^2)$。因此第一阶段时间复杂度为 $O(N|V||V_s|\frac{|E|}{G_n}\min\left(\frac{|V_s|}{G_n},\frac{N}{G_n}\right)^{3.5} L^2)$。算法 3 时间消耗主要为最多 max_*perturbation* 次计算最大链路利用率。使用负载均衡路由算法，在找到一条拥塞链路或者计算出最大链路利用率之前，需要求解最多$|E|$个由式（6）~式（13）表示的 LP 方程，因此算法 3 的时间复杂度为 $O(M_{ax}|E|\min(|V_s|,N)^{3.5}L^2)$，其中 M_{ax} 表示控制循环次数的变量 max_*perturbation*。

4　性能仿真

在本次仿真中，构建一个构造数（constructing number）为 4 的 fat-tree 网络，该网络由 4 个 pod 组成，并通过 4 台核心交换机（core switch）连接，每个 pod 有 2 台聚合交换机（aggregation switch）以及 2 台边缘交换机（edge switch），每台边缘交换机连接至同一 pod 的两台聚合交换机。同时，构建一个 level-l 的 Bcube 网络拓扑。level-1 的 Bcube 由一台 4 端口交换机连接 4 个 level-0 的 Bcube 构成。每个 level-0 的 Bcube 由一台 4 端口交换机连接 4 台服务器构成。在仿真中考虑的两种数据中心网络，均具有 16 台服务器，一台服务器的最大容量为 4 台虚拟机，同时网络中每条链路的最大速率为 1 Gbit/s。在每次仿真中，都尝试将一个随机生成的虚拟机数据中心嵌入进物理数据中心。虚拟机数据中心的每台虚拟机的带宽需求由均匀分布随机生成，使用软件 CPLEX 求解 LP 方程。

首先，将本文提出的分组扰动算法与之前工作提出的扰动算法[6]以及 first-fit 算法进行了比较，仿真参数设置见表 2，仿真曲线如图 6 所示。first-fit 算法是求解装箱问题的经典启发式算法。该算法将虚拟机放置在搜索到的第一台具有剩余容量并且不会导致链路发生拥塞的服务器中。参考文献[6]提出的扰动算法维护一个未放置的虚拟机的列表，首先利用 first-fit 依次放置未放置列表中的虚拟机，若某次虚拟机的放置因为物理链路拥塞而失败，则在已经放置有虚拟机的服务器中选出对拥塞链路流量贡献最大的服务器，拿出该服务器上一台虚拟机，并将这台虚拟机重新放到未放置虚拟机列表中继续该算法。仿真中，上述 3 种算法均采用负载均衡路由算法，并且设置负载均衡路由算法中多路径参数 K=4，表明服务器之间的流量可以在 4 条路径上路由。分组扰动算法中量 max_*perturbation* 设置为 40。物理网络的剩余容量利用如下的分布产生。物理网络中所有的服务器以概率 p_{server} 具有最大的容量 4，链路的剩余带宽以概率 $p_{\text{bandwidth}}$ 为最大速率 1 Gbit/s。以 $1-p_{\text{server}}$ 的概率，服务器的剩余容量在 0~4 通过均匀分布产生，以的概率 $1-p_{\text{bandwidth}}$，链路的剩余带宽在链路最大速率的 0~100%通过均匀分布随机产生。分别对虚拟机数目变化以及虚拟机带宽变化

表 2　仿真参数

类别	参数	取值
基本参数	最大扰动次数 max_*perturbation*	40
	$p_{\text{bandwidth}}$	0.5
	p_{server}	0.5
虚拟机数量变化	虚拟机请求带宽/(Mbit·s^{-1})	400
	虚拟机数目/台	4~14
请求带宽变化	虚拟机数目/台	10
	虚拟机请求带宽/(Mbit·s^{-1})	200~400

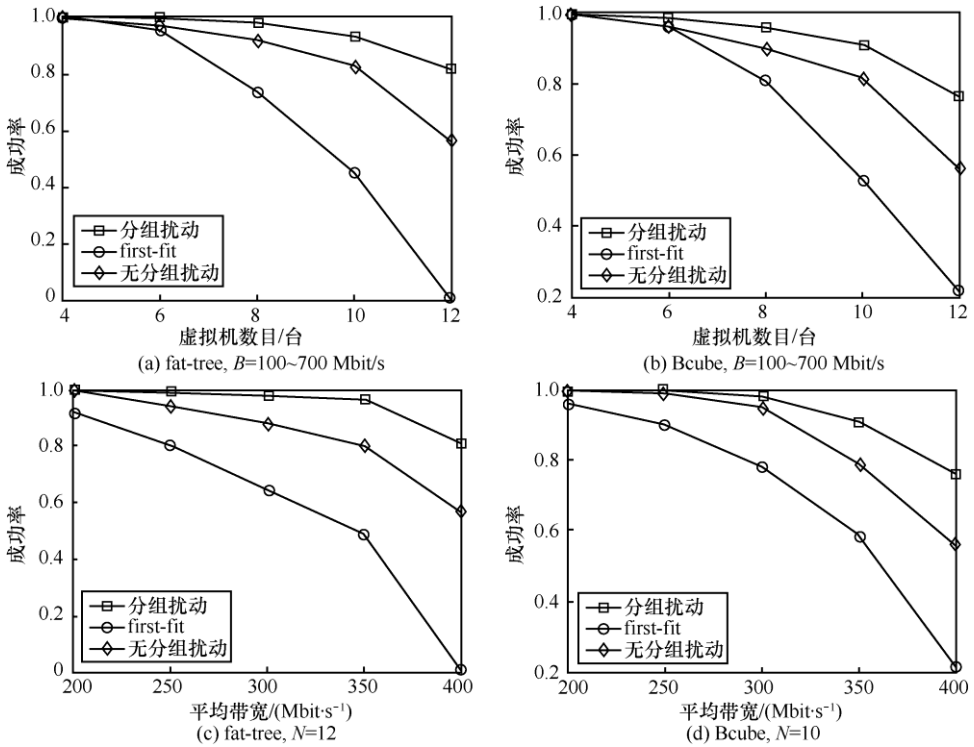


图 6　分组扰动算法与 first-fit 以及无分组扰动算法的比较

两种情形进行仿真。在第一种情形中，虚拟数据中心中的虚拟机请求带宽由 100 Mbit/s 和 700 Mbit/s 的均匀分布产生，即平均带宽为 400 Mbit/s，跟踪虚拟机数目从 4 台增加到 14 台时的性能表现。在第二种情形中，虚拟机的数目为 10 台，虚拟机请求的带宽由[$\bar{B}$ −100 Mbit/s, $\bar{B}$ +100 Mbit/s]的均匀分布产生。

由图 6 可知，first-fit 在带宽变化与虚拟机数目发生变化时的性能表现均较差。虽然不分组扰动算法在解决嵌入虚拟集群这种结构的虚拟数据中心表现出了良好的性能[6]，但是对于嵌入虚拟超额认购集群这种更加复杂的虚拟数据中心结构则表现出较差的性能。基于虚拟超额认购集群结构的虚拟数据中心流量构成更加复杂，该结构的特点使得流量更多地集中在同一组之内。而本文提出的分组扰动算法利用分层数据中心网络拓扑中局部网络资源丰富的特点，首先对物理网络分组，然后将每组虚拟机的放置范围限制在某一局部地区。

在分组扰动算法的第二阶段，使用 max_*perturbation* 限制扰动的最大次数。因此首先仿真比较了 fat-tree 拓扑数据中心下不同 max_*perturbation* 对虚拟数据中心嵌入成功率的影响。虚拟数据中心请求的虚拟机数目为 12 台，虚拟机请求带宽由 100 Mbit/s 和 700 Mbit/s 的均匀分布产生，平均带宽为 400 Mbit/s，最大调整次数变化范围为 0~35，仿真结果如图 7(a)所示。由图 7(a)可知，随着最大扰动次数的增大，成功率也随之增加，且在 20 之后基本不再增长。这是因为网络物理数据中心的网络带宽已经无法嵌入请求的虚拟数据中心。接下来，仿真了最大扰动次数对运行时间的影响，并与分组扰动算法做了比较，仿真结果如图 7(b)所示。由图 7(b)可知，在达到最大成功率之前，运行时间随着最大调整次数的增大而增大，但在 20 之后运行时间因为一次调整无法减少拥塞而导致 $\frac{(\mu_{t-1}(IT^r)-\mu_t(IT^r))}{\mu_{t-1}(IT^r)}<\Delta$，从而返回失败。最后可以看到，成功率最高时的分组扰动算法的运行时间依然比无分组扰动算法的运行时间降低了 30%左右。

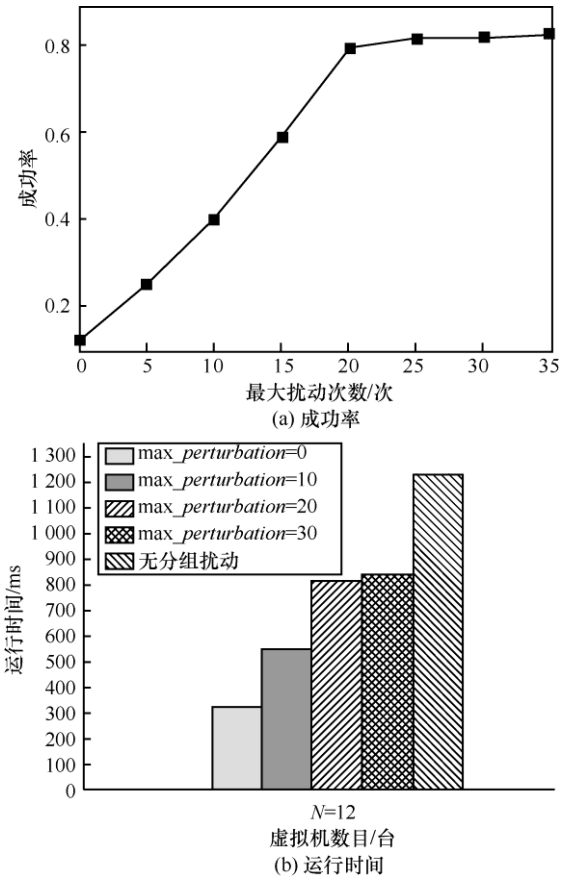


图 7 最大扰动次数与性能

5 结束语

本文研究了超额认购虚拟数据中心的嵌入问题。首先利用线性规划方程描述了基于 Hose 模型的超额认购虚拟数据中心的流量矩阵约束问题。由于数据中心嵌入问题是 NP 完全问题，因此提出了一种具有多项式时间复杂度的分组扰动启发式算法，并仿真了算法在两种典型的数据中心网络拓扑中的性能，仿真结果表明本文提出的分组扰动算法相较于 first-fit 算法表现出了显著优势。面对嵌入超额认购虚拟数据中心，分组扰动算法相较于笔者之前提出的无分组扰动算法表现优异，同时复杂度较低。分组扰动算法为嵌入虚拟

超额认购集群结构的虚拟数据中心提供了一种在性能和复杂度之间进行平衡取舍的算法。

参考文献：

[1] BARI M F, BOUTABA R, ESTEVES R, et al. Data center network virtualization: a survey[J]. IEEE Communications Surveys & Tutorials, 2013, 15(2): 909-928.

[2] MOGUL J C, POPA L. What we talk about when we talk about cloud network performance[J]. ACM SIGCOMM Computer Communication Review, 2012, 42(5): 44-48.

[3] BALLANI H, COSTA P, KARAGIANNIS T, et al. Towards predictable datacenter networks[C]//ACM SIGCOMM Conference, August 15-19, 2011, Toronto, Canada. New York: ACM Press, 2011: 242-253.

[4] KAWAMURA M, AKABANE S, ITO K, et al. Optimal bandwidth-aware VM allocation for Infrastructure-as-a-Service[J]. Computer Science, 2012, 5(4): 603-612.

[5] ZHU J, LI D, WU J, et al. Towards bandwidth guarantee in multi-tenancy cloud computing networks[C]//IEEE International Conference on Network Protocols, October 30-November 2, 2012, Austin, Texas, USA. New Jersey: IEEE Press, 2012: 1-10.

[6] YAN F, LEE T T, HU W. Congestion-aware embedding of heterogeneous bandwidth virtual data centers with hose model abstraction[J]. IEEE/ACM Transactions on Networking, 2016(99): 1-14.

[7] 魏祥麟, 陈鸣, 范建华, 等. 数据中心网络的体系结构[J]. 软件学报, 2013(2): 295-316.
WEI X L, CHEN M, FAN J H, et al. Architecture of the data center network[J]. Journal of Software, 2013(2): 295-316.

[8] MARRIS E. Rambunctious garden: saving nature in a post-wild world[M]. New York: Bloomsbury Publishing, 2013.

[9] ERLEBACH T, RUEGG M. Optimal bandwidth reservation in hose-model VPNs with multi-path routing[J]. IEEE INFOCOM, 2004, 4(4): 2275-2282.

[10] KUMAR A, RASTOGI R, SILBERSCHATZ A, et al. Algorithms for provisioning virtual private networks in the hose model[J]. IEEE/ACM Transactions on Networking, 2002, 10(4): 565-578.

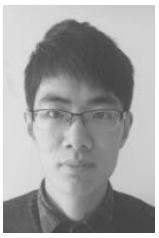
[11] KARMARKAR N. A new polynomial-time algorithm for linear programming[J]. Combinatorica, 1984, 4(4): 373-395.

[12] MEYERHENKE H, SANDERS P, SCHULZ C. Partitioning complex networks via size-constrained clustering[M]. Springer-Verlag: Springer International Publishing, 2014: 351-363.

[13] GONZALEZ T F. Clustering to minimize the maximum intercluster distance[J]. Theoretical Computer Science, 1985, 38(2-3): 293-306.

[14] DALLY W J, TOWLES B P. Principles and practices of interconnection networks[M]. Netherlands: Amsterdam Elsevier Publishing, 2004.

[作者简介]



鹿楚坤（1993−），男，上海交通大学硕士生，主要研究方向为云计算、数据中心网络虚化。

闫芳芳（1984−），女，上海交通大学讲师，主要研究方向为数据中心网络、分组交换结构及调度算法和光交换与多播。

李东（1948−），男，上海交通大学致远讲席教授，IEEE Fellow，主要研究方向为带宽交换理论、网络性能分析、无线通信网络。