

## 基于层次拓扑树的虚拟机节能分配算法<sup>\*</sup>

蔡立军<sup>1</sup>,何庭钦<sup>1†</sup>,孟涛<sup>1</sup>,陈磊<sup>2</sup>

(1. 湖南大学 信息科学与工程学院,湖南 长沙 410082; 2. 湖南大学 电气与信息工程学院,湖南 长沙 410082)

**摘要:**为了均衡分布式数据中心物理主机多维资源的利用率,减少物理主机使用数量,节约能耗,提出了一种基于层次拓扑树的虚拟机节能分配算法 HTES(hierarchical topology energy saving),此算法可以有效提升虚拟机分配效率.利用 Laplacian 矩阵,对大规模网络拓扑分割,建立了层次拓扑树模型.基于层次拓扑模型,根据虚拟机请求中 IP 地址与数据中心的距离,将虚拟机请求分组,从层次拓扑树模型中查询合适的物理主机区域,按虚拟机请求与物理主机的资源匹配度进行虚拟机的分配.将 HTES 与其他 3 种算法进行模拟仿真实验,从虚拟机分配时间、资源均衡率、能耗和物理主机使用情况等方面验证了 HTES 算法能够有效加快物理主机搜索速度,增加底层占用物理主机的集中度,降低底层物理主机的使用数量,达到节约能耗的目的.

**关键词:**数据中心;虚拟机分配;层次拓扑树;能源利用率

**中图分类号:**TP391

**文献标志码:**A

## Energy Saving Allocation Algorithm of a Virtual Machine Based on Hierarchical Topology Tree

CAI Lijun<sup>1</sup>, HE Tingqin<sup>1†</sup>, MENG Tao<sup>1</sup>, CHEN Lei<sup>2</sup>

(1. College of Information Science and Engineering, Hunan University, Changsha 410082, China;

2. College of Electrical and Information Engineering, Hunan University, Changsha 410082, China)

**Abstract:** In order to balance the utilization of multi-dimensional physical host resources and reduce usage number, a virtual machine allocation energy saving algorithm was proposed based on hierarchical topology tree (HTES) in the environments of distributed data center, which enhances allocation efficiency of the virtual machines. Laplacian matrix was then used to split large-scale network topology and to build hierarchical topology tree model. Furthermore, according to the distance between request IP address and the data center, HTES divided the virtual machines into groups, and searched the appropriate physical host region from hierarchical topology tree for allocation, which is based on the match degree between virtual machine requests and physical hosts. Simulation experiments were performed on HTES algorithms and other three algorithms, considering the virtual machine allocation time, resource balancing rate, energy con-

<sup>\*</sup> 收稿日期:2016-02-02

**基金项目:**国家自然科学基金资助项目(61174140, 61472127, 61272395), National Natural Science Foundation of China(61174140, 61472127, 61272395); 中国博士后科学基金资助项目(2013M540628, 2014T70767), China Postdoctoral Science Foundation(2013M540628, 2014T70767); 湖南省自然科学基金资助项目(14JJ3107), Natural Science Foundation of Hunan Province of China (14JJ3107); 湖南省教育厅科研优秀青年资助项目(15B087), Science Foundation for the Excellent Youth Scholars of Hunan Province of China(15B087)

**作者简介:**蔡立军(1964-),男,湖南常德人,湖南大学教授,博士

<sup>†</sup> 通讯联系人, E-mail: hetingqin@hnu.edu.cn

sumption and physical host usage, and other aspects. The results shows that the HTES is able to balance multi-dimensional resources physical hosts, reduce physical host usage, and save energy consumption.

**Key words:** data centers; virtual machine allocation; hierarchical topology tree; energy utilization

随着信息科技的不断发展,数据中心作为一种基础设施,已经被各行各业普遍使用.然而当前数据中心的发展也面临新的问题,数据中心的规模不断扩张,地理位置愈趋分散.多个分散的数据中心通过高速网络互联,共同组成了大型的分布式数据中心.在分布式数据中心内,用户通过按需付费的模式,向数据中心提交需求.数据中心根据用户地理位置,从较近的基础设施库分配资源并构建虚拟机为用户服务.然而,大规模分布式数据中心环境的虚拟机的分配问题面临新的挑战,主要表现为主机地理位置更为分散、底层资源规模更为庞大、多维异构资源、较高的能源消耗等.因此,合理的虚拟机(资源)分配策略是数据中心收益的保障,研究虚拟机分配算法具有重要意义.

目前已有许多学者对数据中心虚拟机的分配进行研究,取得了较多的优秀成果.一些研究成果集中在优化分布式数据中心的资源分配上<sup>[1-6]</sup>.文献[1]从服务供应商的角度,研究了分布式数据中心的收益最大化问题,提出了一种结合虚拟机分配的动态调价算法.文献[2-3]同样着眼于分布式数据中心的成本优化问题,从数据传输和资源分配两个角度设计了相应的数据管理系统和资源调度算法,最小化数据、成本低的同时,优化了数据的传输时间、提升了底层物理资源的利用率.文献[4]提出了一种基于温度感知的资源管理系统,通过动态调整服务器的功率,实现虚拟机分配和服务负载间的优化.文献[5]在分布式数据中心内,对虚拟机的分配请求建立 G/G/1/PS 队列,通过优化队列处理实现服务器负载和虚拟机分配的均衡,节约了数据中心能耗.文献[6]为提升高性能数据中心资源使用率,设计了 CAE 集成平台架构,实现了一种基于 Web 方式的高性能计算中心资源的解决方案.有些研究工作使用数据中心网络拓扑来优化虚拟机的分配,提升底层物理资源的利用率<sup>[7-10]</sup>.在 portland 网络拓扑上,文献[7]提出了 2 种启发式算法,通过分配虚拟机到最大链路能力和邻近的物理主机上,降低了网络开销,增加了底层资源利用率.文献[8]根据网络拓扑建立了 MNT 指标,优化资源分配.文献[9]在网络拓扑的基础上,通过虚拟机和链路的合并,增加

了拓扑中空闲网络设备数量,节约了能源.同样,在实际数据中心拓扑上,文献[10]通过对带宽过载的虚拟机进行合并,优化了网络传输消耗.此外,还有较多学者研究数据中心的能耗问题<sup>[11-14]</sup>,通过各种模型和方法减少数据中心的能源消耗.文献[11]同时考虑了虚拟机的分配和网络流的传输,通过建立线性规划模型,并行处理虚拟机的分配,节约能源消耗.文献[12]中将虚拟机的分配问题看做多商品流的成本最小化问题,通过 Benders 分解算法进行求解,减少了底层物理主机的使用数量,节约了能源消耗.文献[15]分析了云数据中心下资源分配和能源消耗问题,设计了一种节能框架,在减少成本的同时节约能耗.文献[16]研究了分布式数据中心内的能源节约问题,建立了最大化整数规划模型,通过虚拟机的合并,减少了物理主机的使用,从而实现能耗的节约.

以上的研究工作在处理大规模非树型随机网络拓扑<sup>[17]</sup>的虚拟机分配问题上,无法有效减少数据中心物理主机的使用数量,仍面临能耗较高的缺陷.网络拓扑的大规模性和随机性导致虚拟机分配时扫描的物理主机范围更为庞大,使用传统的算法效率较低,一方面表现在底层物理主机的搜索时间过长,降低了虚拟机的分配效率;另一方面底层物理主机分配后集中度较低,过高的分散性不利于物理主机的管理和维护.如 DCEERS 算法<sup>[18]</sup>通过 Benders 分解进行虚拟机分配,利用最小数量的物理主机承载虚拟机请求,虽然减少了物理主机的数量,在一定程度上降低了能耗,但并未考虑资源的均衡率,可能引起局部负载及单位时间功耗过大;ANT 算法<sup>[19]</sup>利用蚁群策略求解多目标虚拟机的分配问题,但却需要大量的迭代寻找最优,分配时间上较差.

因此,本文提出了一种基于层次拓扑树的虚拟机节能分配算法.首先,将分布式数据中心的大规模网络随机拓扑进行拓扑分割,建立层次拓扑树.其次,在考虑底层物理主机多维资源均衡的前提下,扫描层次拓扑树,将虚拟机集中分配网络拓扑中的集中区域,降低底层物理主机的使用.通过关闭空闲物理主机达到节约能源的目的.最后,通过大量实验验证了算法的性能.基于层次拓扑树的虚拟机节能分

配算法优化了虚拟机的分配,提高了底层资源利用率,降低了能源消耗。

## 1 模型建立

### 1.1 预备知识

#### 1.1.1 分布式数据中心网络拓扑

网络拓扑是数据中心整体结构的一种表示和体现。数据中心所有物理主机、存储设备、网络设备通过网络链路彼此互连,共同组成网络拓扑。通常情况下,网络拓扑用图  $G = (V, E)$  表示。其中,  $V$  为节点集合,表示数据中心内所有的物理设备;  $E$  为边的集合,表示两两物理设备间的网络链路能力,即网络带宽。

在虚拟机的分配过程中,网络拓扑起着重要的作用。所有的虚拟机必须映射到网络拓扑中的物理主机上,占用物理主机的 CPU, MEM 资源,占用网络拓扑中多个物理主机间的链路带宽资源,甚至占用拓扑中存储节点的部分存储资源。一个虚拟机在分配过程中,需要扫描网络拓扑中的空闲物理主机,进行最终的资源分配。网络拓扑可以轻松反映虚拟机的分配情况和运行情况,能够方便监控物理主机的负载和运行,便于物理主机的资源调优和能源节约。当网络拓扑发生变化时,说明底层物理设备出现了故障,需要进行虚拟机的迁移和重新分配。

分布式数据中心由多个地理位置分散的小型数据中心组成。小型数据中心之间通过高速网络进行互连。每个小型数据中心彼此独立,可以拥有不同类型的网络拓扑和物理主机。通常,分布式数据中心的整体网络拓扑是随机的,底层物理主机资源是异构的。图 1 为分布式数据中心网络拓扑图。

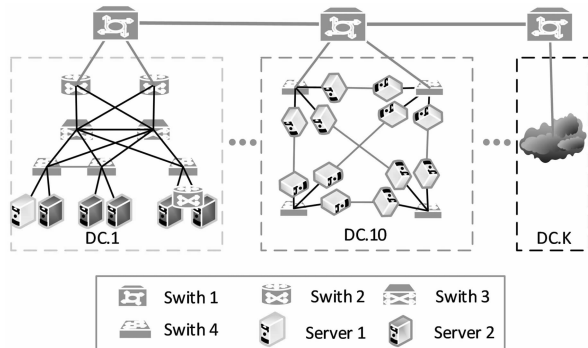


图 1 分布式数据中心网络拓扑图

Fig. 1 Network topology of distributed data center

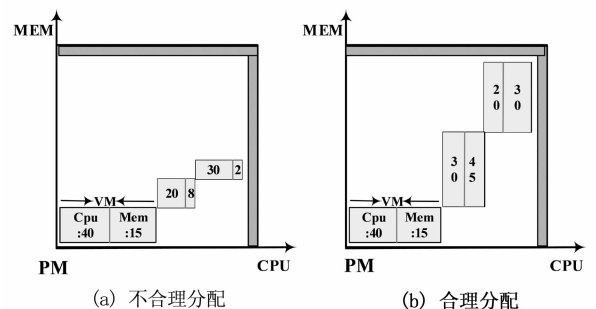
在图 1 中,分布式数据中心由  $K$  个小型数据中

心组成。每个数据中心拥有不同的网络拓扑和物理主机类型。分布式数据中心的网络拓扑呈现随机性和大规模性,物理主机拥有异构特性。网络拓扑的大规模性和随机性导致虚拟机分配时扫描的物理主机范围庞大,增加了虚拟机的搜索时间,降低了虚拟机的分配效率。此外,网络拓扑的大规模性必然存在大量物理主机和网络设备空闲的情况,带来庞大的能源开销,增加数据中心的成本;物理主机的异构性增加了虚拟机分配后物理主机多维资源间的不均衡分配,造成资源浪费;数据中心的分散性增加了额外的网络开销,浪费了网络带宽资源。

#### 1.1.2 单机多维资源的不均衡分配

在数据中心网络拓扑中,物理主机本身由多种资源组成(CPU, MEM 和存储),可看作多维资源向量。如果某维资源(CPU)过度分配,必然造成其他维资源(MEM 和存储)的浪费。只有均衡利用各维资源,才能更充分地发挥资源效率,提升底层资源的利用率,减少数据中心物理主机的使用数量,降低能源开销。在传统虚拟机分配过程中,虚拟机的随机分配往往导致单机多维资源的不均衡分配。

图 2(a)为单机多维物理资源的不合理分配情况。由图 2(a)可知,仅考虑了 CPU 和 MEM 两维资源,3 个虚拟机分配到了物理主机上,造成了物理主机 CPU 资源的利用率达到了 90% ( $40\% + 20\% + 30\%$ )。然而,物理主机的内存资源才使用 25% ( $15\% + 8\% + 2\%$ )。当新的虚拟机访问物理主机时,虽然剩余较多的内存资源,但是由于 CPU 资源的高利用率,导致物理主机无法承载新的虚拟机,从而造成了内存资源的大量浪费,出现单机多维资源的不均衡分配。图 2(b)中,描述了物理主机的均衡分配情况。同样是 3 个虚拟机,但是物理主机的 CPU 和 MEM 资源利用率相对均衡,都达到了 90%,不会出现单维资源的空闲浪费,能够更加充分地利用底层资源。



(a) 不合理分配

(b) 合理分配

图 2 单机多维资源不合理分配图

Fig. 2 Multi-dimensional resource allocation of stand-alone

单机多维资源的均衡分配能够提升底层物理资源的利用率,降低数据中心成本.此外,多维资源的均衡分配,可以在一定程度上减少底层物理主机的使用数量,达到降低数据中心能耗的目的.

## 1.2 问题描述

分布式数据中心内虚拟机的调度过程可描述为:位置各异的多个用户向数据中心提交一批虚拟机请求  $Vms = \{vm_1, vm_2, \dots\}$ . 每个虚拟机拥有 4 种属性,用  $vm_i = \{req\_cpu_i, req\_mem_i, req\_bw_i, ip_i, lifetime_i\}$  表示,其中  $req\_cpu_i$ ,  $req\_mem_i$ ,  $req\_bw_i$  分别表示 CPU, MEM 和带宽请求大小,  $ip_i$  表示用户的地理位置信息,  $lifetime_i$  表示当前虚拟机的生命周期. 分布式数据中心由网络拓扑图  $G = (V, E)$  表示. 其中,  $V$  为物理资源的集合,  $E$  为物理资源间网络链路带宽的集合. 在分布式数据中心内,用户提交的虚拟机请求将根据用户位置分配到较近的物理主机上,占用物理主机资源和网络带宽. 每个物理主机  $host_j = \{cab\_cpu_j, cab\_mem_j, cab\_bw_j, DC\}$  拥有 3 种资源 CPU, MEM 和带宽. DC (Data Center) 是物理主机所属子数据中心的标识,代表物理主机的位置信息.

在虚拟机的分配过程中,每个虚拟机  $vm_i$  只能分配到一台物理主机上  $host_j$ . 物理主机  $host_j$  的剩余资源能力必须满足虚拟机的请求. 一个用户可以提交多个虚拟机请求,同一用户的多个虚拟机应该分配在同一地理位置的物理主机上. 此外,分布式数据中心内,多个数据中心的物理主机通常为异构主机,拥有不同的 CPU, MEM 大小. 在分布式数据中心网络拓扑中,所有网络设备和存储设备不能独立承载虚拟机,不具备相应的计算和处理能力.

## 1.3 分配模型

根据分布式数据中心的虚拟机分配过程描述,本文的虚拟机分配目标是:均衡物理主机多维资源的分配,减少底层物理主机的使用数量,提升底层资源的利用率,节约数据中心的能源消耗.

分布式数据中心内,虚拟机的分配过程是连续的,分配过程不断循环,一次分配过程完成后另一次分配过程准备开始. 在多次虚拟机分配过程中,物理主机负载状态前后一致. 单个物理主机可以在不同轮次的分配过程中承载多个虚拟机请求. 为了均衡单个物理主机多维资源的均衡分配,必须在物理主机历史负载状态下考虑本轮分配,实现虚拟机分配后物理主机各维资源的剩余率均衡.

为了更好地描述物理主机多维资源的均衡情

况,文中定义了请求匹配度  $HMatch$  的概念,描述当前虚拟机分配到物理主机后各维资源的使用情况.

$$HMatch_{ij} = \left| \frac{(sur\_cpu_j - req\_cpu_i)}{cab\_cpu_j} - \frac{(sur\_mem_j - req\_mem_i)}{cab\_mem_j} \right| \times Is\_New_j \quad (1)$$

$$Is\_New_j = \begin{cases} 1, & (sur\_cpu < cab\_cpu) \text{ or } (sur\_mem < cab\_mem) \\ 2, & (sur\_cpu = cab\_cpu) \text{ and } (sur\_mem = cab\_mem) \end{cases} \quad (2)$$

公式(1)描述了虚拟机  $vm_i$  分配到物理主机  $host_j$  后,物理主机剩余资源的均衡程度. 其中,  $req\_cpu_i$ ,  $req\_mem_i$  表示虚拟机  $vm_i$  对 CPU 和 MEM 资源的请求大小.  $sur\_cpu_j$ ,  $sur\_mem_j$  表示物理主机  $host_j$  经过多轮虚拟机分配后剩余的 CPU 和 MEM 资源大小.  $cab\_cpu_j$ ,  $cab\_mem_j$  表示物理主机  $host_j$  的原始 CPU 和 MEM 资源大小. 本文仅考虑物理主机的 CPU 和 MEM 资源的分配均衡程度,暂未考虑存储和 I/O 等资源.

此外,从式(1)可知,请求匹配度  $0 \leq HMatch_{ij} < 2$ . 当虚拟机  $vm_i$  分配到物理主机  $host_j$  后, CPU 和 MEM 的剩余资源占总资源的比值相同,则匹配度为 0. 否则,匹配度值大于 0. 剩余资源间越均衡,则匹配度值越小. 通过匹配度的计算,可以快速获取虚拟机分配对物理主机资源使用的影响,越小的资源匹配度,则说明物理主机资源的利用率越高,越能减少底层物理主机的数量,实现节约能耗的目的.

在式(2)中,  $Is\_New_j$  描述了物理主机是否在历史分配过程中承载了虚拟机. 若当前物理主机是空闲物理主机,为承载任何虚拟机,则主机的剩余 CPU 和 MEM 能力与总资源能力相同. 否则,剩余能力小于总资源能力. 为了限制虚拟机随意分配到新的空闲虚拟机,本文用整数 1 和 2 来分别描述当前物理主机的状态.

在虚拟机分配过程中,虚拟机请求数量与底层物理主机数量不同,本文定义 0-1 变量  $x_{ij}$  表示虚拟机  $vm_i$  与物理主机  $host_j$  间的映射关系,见式(3).

$$x_{ij} = \begin{cases} 1, & vm_i \text{ is mapped to } host_j \\ 0, & vm_i \text{ is not mapped to } host_j \end{cases} \quad (3)$$

在式(3)中,虚拟机  $vm_i$  分配到物理主机  $host_j$  上,则  $x_{ij}$  为 1; 否则,  $x_{ij}$  为 0. 因此,分布式数据中心虚拟机分配的目标函数可以表示如下:

$$\begin{aligned} \min & \sum_{j \in Hosts} \sum_{i \in Vms} x_{ij} \times HMatch_{ij} \\ \text{st.} & \end{aligned} \quad (4)$$

$$\begin{aligned} & \text{for each } \text{vm}_i: \sum_{j \in \text{Hosts}} x_{ij} \leq 1 \\ & \text{for each } \text{host}_j: \begin{cases} \sum_{i \in \text{Vms}} x_{ij} \times \text{req\_cpu}_i \leq \text{cab\_cpu}_j \\ \sum_{i \in \text{Vms}} x_{ij} \times \text{req\_mem}_i \leq \text{cab\_mem}_j \\ \sum_{i \in \text{Vms}} x_{ij} \times \text{req\_bw}_i \leq \text{cab\_bw}_j \end{cases} \end{aligned}$$

$$\text{for each } \text{vm}_i, \text{host}_j: 0 \leq \text{HMatch}_{ij} < 2$$

在式(4)中,最小的物理主机匹配度表示所有虚拟机分配到物理主机后,物理主机剩余资源的均衡性最好,则其可以承载的虚拟机数量就越多,分配过程中使用的物理主机数量将减少,此外,当已使用的物理主机分配完之后,再将虚拟机分配到其他的空闲物理主机上,从而节约了数据中心能耗.表1为分布式数据中心虚拟机分配过程中的符号及相关术语.

表 1 符号术语表  
Tab. 1 Notation and terminology

符号	说明
$G = (V, E)$	$G$ 为分布式数据中心网络拓扑, $V$ 为物理资源的集合, $E$ 为物理资源间链路带宽的集合
$\text{vm}_i$	第 $i$ 个虚拟机, 虚拟集合为 $\text{Vms}$
$\text{req\_cpu}_i$	第 $i$ 个虚拟机的 CPU 资源请求大小
$\text{req\_mem}_i$	第 $i$ 个虚拟机的 MEM 资源请求大小
$\text{req\_bw}_i$	第 $i$ 个虚拟机的 BW 资源请求大小
$\text{host}_j$	网络拓扑 $G$ 中的第 $j$ 个物理主机, 物理主机集合 $\text{Hosts}$
$\text{cab\_cpu}_j$	第 $j$ 个物理主机的 CPU 资源大小
$\text{cab\_mem}_j$	第 $j$ 个物理主机的 MEM 资源请求大小
$\text{cab\_bw}_j$	第 $j$ 个物理主机的 BW 资源请求大小
$x_{ij}$	二元变量, 如果虚拟机 $\text{vm}_i$ 分配到物理主机 $\text{host}_j$ , 则 $x_{ij}$ 为 1, 否则, 为 0
$\text{Is\_New}_j$	二元变量, 如果物理主机 $\text{host}_j$ 已经分配了虚拟机, 则 $\text{Is\_New}_j$ 为 1, 否则, 为 2
$\text{HMatch}_{ij}$	虚拟机 $\text{vm}_i$ 分配到物理主机 $\text{host}_j$ 后, 物理主机的多维资源均衡度

## 2 基于层次拓扑树的节能分配算法

针对分布式数据中心的网络拓扑特点,结合分配模型,本节提出了一种基于层次拓扑树的虚拟机节能分配算法,简称 HTES. 首先,HTES 将分布式数据中心的大规模网络拓扑进行分配,建立了层次拓扑树,详细描述见 2.1 节;其次,在层次拓扑树的基础上,根据用户的地理位置,将虚拟机分配到网络拓扑中的某个区域,保证占用的物理主机相对集中,均衡物理主机多维资源分配,减少物理主机使用数量,节约能耗,详细的分配算法见 2.2 节.

### 2.1 拓扑分割建立层次拓扑树

分布式数据中心由多个地理位置分散的小型数据中心组成,整个网络拓扑具有明显的大规模性和随机性.在虚拟机的分配过程中,需要为每个虚拟机寻找合适的物理主机,从而必须搜索网络拓扑的所有物理主机.网络拓扑的大规模给虚拟机的分配过程带来了两个难点:①增加了底层物理主机的搜索时间,降低了虚拟机的分配效率.②扩散了底层物理主机的占用分布.例如,同一用户的多个虚拟机请求可能分配到不同地理位置的物理主机上,也可能分配到同一地理位置内的距离位置较远的多台物理主机上.这种分散性给底层物理主机的管理和维护带来了巨大困难,增加了节能策略的监控难度.

针对以上两个难点,本文对分布式数据中心的网络拓扑进行逐级分割,建立层次拓扑树.将整个拓扑转化为层次结构,保证每一层的节点数目逐层递减.当进行虚拟机分配时,根据虚拟机的分配数量,找到相应层级的拓扑子树寻找物理主机,不用再扫描整个网络拓扑,减少了物理主机的扫描时间,增加了虚拟机占用物理主机的集中度,便于虚拟机的快速分配和能耗节约.

为了进行网络拓扑的分割,本文引入了规范割 Normalized-cut(Ncut)和 Laplacian 矩阵,证明了 Laplacian 矩阵的第二特征值是 2 分 Ncut 的最优解.

**定义 1** 规范割 Normalized-cut(Ncut)是衡量网络拓扑中两个节点子集间相似度的标准,公式如下:

$$\text{Ncut}(A, \bar{A}) = \frac{\text{cut}(A, \bar{A})}{\text{vol}(A, V)} + \frac{\text{cut}(A, \bar{A})}{\text{vol}(A, V)}$$

$$A \in V \text{ and } \bar{A} \in V \quad (5)$$

$$\text{cut}(A, \bar{A}) = \sum_{i \in A, j \in \bar{A}} w_{ij} \quad (6)$$

$$\text{vol}(A, V) = \sum_{i \in A, j \in V} w_{ij} \quad (7)$$

式中: $V$  为网络拓扑  $G$  中的节点集合; $w_{ij}$  为网络拓扑的邻接矩阵  $\mathbf{W}$  中的值,表示节点  $i$  与节点  $j$  间的权值.邻接矩阵  $\mathbf{W}$  为对称矩阵,其中  $w_{ij} = w_{ji} \geq 0$ .使用规范割 Ncut 进行拓扑分割度量,能够较好地避免小区域的产生.

**定义 2** Laplacian 矩阵是网络拓扑图的一种有效表示方式,一个 Laplacian 矩阵对应一个非负权重的无向有权图. Laplacian 矩阵的常用表示如下:

$$\mathbf{L}_{\text{sym}} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \quad (8)$$

式中: $\mathbf{I}$  为单位矩阵; $\mathbf{D}$  为网络拓扑中节点的度矩

阵;  $\mathbf{W}$  为邻接矩阵. 每个节点的度  $d_i = \sum_j w_{ij}$ , 表示与该节点相连的所有边的权值和.

Laplacian 矩阵具有如下特性:

- 1) 任意  $n$  维向量  $\mathbf{f} \in R$ , 有:  $\mathbf{f}^T \mathbf{L} \mathbf{f} = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \times (f_i - f_j)^2$ .
- 2)  $\mathbf{L}$  为半正定矩阵.
- 3)  $\mathbf{L}$  的最小特征值为 0, 对应的最小特征向量为值 1 的常向量  $\mathfrak{A}$ .
- 4)  $\mathbf{L}$  有  $n$  个非负实特征值  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ .
- 5) 如果网络拓扑  $G$  是  $C$  的连通部件, 那么  $\mathbf{L}$  有  $C$  个等于 0 的特征值.

根据 Ncut 和 Laplacian 矩阵的定义和性质, 可得 Laplacian 矩阵的第二特征值是 2 分 Ncut 的最优解, 证明过程如下:

**证明** 设划分向量  $\mathbf{f}$  如下:

$$\mathbf{f}(v_i) = \begin{cases} \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}}, & \text{if } v_i \in A \\ -\sqrt{\frac{\text{vol}(A,V)}{\text{vol}(\bar{A},V)}}, & \text{if } v_i \in \bar{A} \end{cases}$$

根据  $\mathbf{L}$  的性质 1, 得

$$\begin{aligned} \mathbf{f}^T \mathbf{L} \mathbf{f} &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 = \\ & \frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} (f_i - f_j)^2 + \frac{1}{2} \sum_{i \in A, j \in A} w_{ij} (f_i - f_j)^2 + \\ & \frac{1}{2} \sum_{i \in \bar{A}, j \in \bar{A}} w_{ij} (f_i - f_j)^2 + \frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} (f_i - f_j)^2 = \\ & \frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} (f_i - f_j)^2 + \frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} (f_i - f_j)^2 = \\ & \frac{1}{2} \text{cut}(A, \bar{A}) \left( \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}} + \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(\bar{A},V)}} \right)^2 + \\ & \frac{1}{2} \text{cut}(A, \bar{A}) \left( -\sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}} - \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(\bar{A},V)}} \right)^2 = \\ & \text{cut}(A, \bar{A}) \left( \frac{\text{vol}(V,V)}{\text{vol}(A,V)} + \frac{\text{vol}(V,V)}{\text{vol}(\bar{A},V)} \right) = \\ & \text{vol}(V,V) \text{Ncut}(A, \bar{A}) \end{aligned}$$

此外:

$$\begin{aligned} \mathbf{f}^T \mathbf{D} \mathbf{f} &= \sum_{i \in A} \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}} d_i \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}} + \\ & \sum_{i \in \bar{A}} \left( -\sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}} \right) d_i \left( -\sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}} \right) = \\ & \frac{\text{vol}(\bar{A},V)}{\text{vol}(A,V)} \sum_{i \in A} d_i + \frac{\text{vol}(A,V)}{\text{vol}(\bar{A},V)} \sum_{i \in \bar{A}} d_i = \end{aligned}$$

$$\frac{\text{vol}(\bar{A},V)}{\text{vol}(A,V)} \text{vol}(A,V) + \frac{\text{vol}(A,V)}{\text{vol}(\bar{A},V)} \text{vol}(\bar{A},V) = \text{vol}(V,V)$$

将上述两式结合, 可得:  $\text{Ncut}(A, \bar{A}) = \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$

根据  $\mathbf{L}$  的性质 3, 可得:

$$\begin{aligned} (\mathbf{D} \mathbf{f})^T \mathfrak{A} &= \sum_{i=1}^n d_i f_i = \sum_{i \in A} d_i \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}} - \\ & \sum_{i \in \bar{A}} d_i \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(\bar{A},V)}} = \text{vol}(A,V) \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}} - \\ & \text{vol}(\bar{A},V) \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(\bar{A},V)}} = \\ & \sqrt{\text{vol}(A,V) \text{vol}(\bar{A},V)} - \\ & \sqrt{\text{vol}(\bar{A},V) \text{vol}(A,V)} = 0 \end{aligned}$$

因此, Ncut 的最小化:  $\min_f \frac{\mathbf{f}^T \mathbf{L} \mathbf{f}}{\mathbf{f}^T \mathbf{D} \mathbf{f}}$

st.

$$\begin{aligned} f_i &\in \left( \sqrt{\frac{\text{vol}(A,V)}{\text{vol}(A,V)}}, -\sqrt{\frac{\text{vol}(A,V)}{\text{vol}(\bar{A},V)}} \right) \\ (\mathbf{D} \mathbf{f})^T \mathfrak{A} &= 0 \end{aligned}$$

由于最小化的 Ncut 是一个广义 Rayleigh 商, 根据性质可得  $\lambda_1 = 0, \mathbf{D}^{1/2} \mathfrak{A}$  为其最小特征值, 可得  $\mathbf{f}^T \mathbf{L} \mathbf{f} / \mathbf{f}^T \mathbf{D} \mathbf{f}$  的第二小特征值是 Ncut 的最优解.

根据 Laplacian 矩阵与 Ncut 间的特性, 本文将分布式数据中心的网络拓扑转化为 Laplacian 矩阵, 然后对其求第二特征值, 从而将网络拓扑分割成耦合度最小的两个子集. 再依次对两个子集进行拓扑分割, 最终形成层次拓扑树. 图 3 为网络拓扑分割的过程.

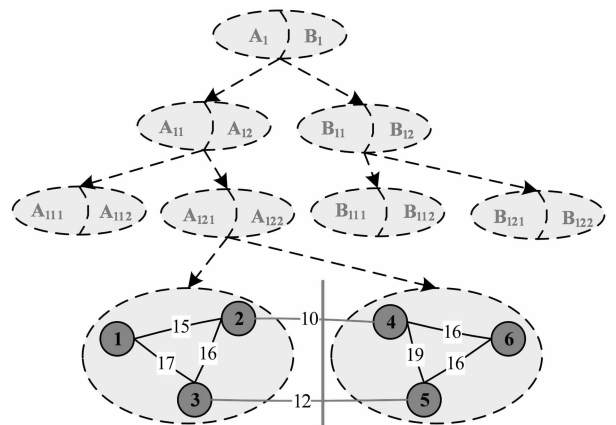


图 3 网络拓扑分割过程图

Fig. 3 Network topology segmentation process

从图中可以看出, 网络拓扑的分割过程为 2 分

迭代分割,通过多次迭代将网络拓扑分割成 2 个子拓扑.在整个分割过程中,必须保持多个数据中心间地位位置的分散特性.因此,分布式数据中心的层次拓扑树的第一层拓扑树就是多个地理位置分散数据中心拓扑,不进行过细的划分.此外,拓扑分割的目标是加快物理主机的搜索速度,减少分配时间,2 分迭代分割必然出现最下层次子拓扑中节点数量较少的情况.为此,本文设定了拓扑分割门限  $SplT$  和拓扑合并门限  $MerT$ .根据拓扑分割门限  $SplT$  判断是否继续进行网络拓扑的分割.根据拓扑合并门限进行节点数小于  $MerT$  的子拓扑的合并.因此,拓扑分割门限  $SplT$  必须大于拓扑合并门限  $MerT$ .

因此,分布式数据中心网络拓扑分割的具体过程如下:①根据分布式数据中心内小型数据中心的网络拓扑,建立第 1 层子拓扑集,保持分布式数据中心网络拓扑间地理位置的分散特性;②逐个遍历第 1 层子拓扑集,形成  $L$  矩阵,根据公式(8)进行第二特征值求解.根据求解结果计算相应的特征向量,按照特征向量中得到的节点列表,将网络拓扑分割成两个子拓扑;③逐层遍历所有子拓扑,生成下一层子拓扑集.当某个子拓扑中的节点数量少于最低分割门限  $SplT$ ,不再进行下一层拓扑分割;④当所有拓扑分割完成后,遍历最后一层的子拓扑集,将节点数少于  $MerT$  的子拓扑与其兄弟节点进行合并;⑤遍历所有层的子拓扑,统计每层子拓扑中节点数量的范围,结合层次号和地理位置(IP),建立索引方便查询.算法 1 详细描述了网络拓扑分割建立拓扑树的详细过程.

#### 算法 1 网络拓扑分割过程

输入:网络拓扑  $G$ ,拓扑分割门限值  $SplT$ ,拓扑合并门限值  $MerT$ .

输出:层次拓扑树  $TT$ .

##### Step1 初始化.

1)根据网络拓扑  $G$ ,生成相应的邻接矩阵  $W$  和度矩阵  $D$ .

2)初始化待分割的网络拓扑集  $TS$  和子拓扑集  $STS$  为空.

3)初始化层次拓扑树  $TT$  为空,层次序号  $IN$  为 0.

4)根据网络拓扑  $G$  和多个数据中心的地理位置分散关系,生成第 1 层子拓扑集,填充到  $STS$ ,计算总节点数量  $VN$ .更新层次拓扑树的顶层索引,包括层次序号、节点总数、子拓扑集合链接等,如下所示.更新完成后进入 Step2,开始拓扑分割. insert

$\langle IN, VN, TS \rangle$  to  $TT(IN)$ .

**Step2** 更新待分割拓扑集,将  $STS$  覆盖  $TS$ ,清空  $STS$ .更新当前层次拓扑树的层号  $IN = IN + 1$ .进入 Step3 开始拓扑分割.

**Step3** 判断  $TS$  是否为空.  $TS$  为空,则进入 Step5 合并节点数目过小的子拓扑.否则,进入 Step4,分割拓扑.

**Step4** 遍历待分割拓扑集  $TS$ ,生成子拓扑集  $STS$ .

1)计算子拓扑中节点数量  $VN$ .比较  $VN$  与分割门限  $SplT$  的大小.若大于,进入 2)进行拓扑分割.若小于,则不进行拓扑分割.进行下一个子拓扑的计算.如果  $TS$  遍历完成,进入 Step2 更新待分割拓扑集.

2)根据子拓扑  $G$ 、邻接矩阵  $W$  和度矩阵  $D$ ,按照公式(8)生成  $L$  矩阵  $L_{sym} = I - D^{-1/2}WD^{-1/2}$ ,其中  $I$  是单位矩阵.进入 3)计算第二特征值.

3)根据  $L$  矩阵特性,计算  $(D - W)x = \lambda Dx$  的第二小特征值.根据特征值求得想要的特征向量  $f$ .按照特性向量中的节点,将子拓扑分割成两个子集,加入  $STS$  中.进入 4)更新层次拓扑树.

4)计算中  $STS$  总节点数量  $VN$ ,更新层次拓扑树,如下所示.更新完成后重新进行 1),分割下一个子拓扑.

insert  $\langle IN, VN, TS \rangle$  to  $TT(IN)$ .

**Step5** 遍历  $IN$  层子拓扑,判断每个子拓扑中节点数量.若节点数量小于  $MerT$ ,删除当前子拓扑集和同一父节点的兄弟拓扑集.

## 2.2 区域性节能分配算法

根据分布式数据中心的网络拓扑特性,在构建层次拓扑树的基础上,本节提出了一种区域性虚拟机节能分配算法 HTES. HTES 的核心节能策略是减少底层物理主机的使用,通过关闭空闲物理主机实现节能.较传统的相同策略节能算法<sup>[12]</sup>而言,HTES 关注于分布式数据中心环境;根据用户 IP 分组虚拟机请求,就近提供物理主机资源;分割网络拓扑建立层次拓扑树,快速寻找合适的物理主机;区域性分配虚拟机到底层集中的物理主机上,缓解了传统零散分配带来的难管理、难节能、难纠错等缺点;考虑物理主机多维资源的均衡分配,提高底层物理主机的资源利用率.

本文采用区域性分配,将虚拟机分配到底层集中的物理主机上,缓解了物理主机分散带来的节能和管理困难.区域性虚拟机节能算法的主要流程分

为:虚拟机请求分组,物理主机搜索和区域性虚拟机均衡分配 3 个步骤。

虚拟机请求分组:①请求分组,根据虚拟机请求中 IP 地址属性,按照层次拓扑中顶层节点的地理位置(IP),将虚拟机请求进行分组.分组完成后,所有虚拟机请求都在距离自己最近的数据中心队列中.

②并发分配,并发对多个虚拟机请求分组进行相应的物理资源分配。

区域性物理主机搜索:①锁定搜索层次.根据分组完成后的虚拟机请求个数  $V\_size$ ,查询拓扑层次树中空闲节点数量(物理主机)充足的子拓扑,保证子拓扑的空闲数量大于虚拟机请求数量  $\alpha$  倍,更加均衡地分配虚拟机,均衡物理主机的多维资源.

②比较资源能力.比较虚拟机请求大小与空闲物理主机的剩余能力,选择剩余能力大于请求大小的物理主机.若物理主机搜索成功,数量等于  $\alpha \times V\_size$ ,则进行虚拟机分配.否则,遍历当前子拓扑的父亲拓扑,再次搜索合适数量的主机,直到找到  $\alpha \times V\_size$  个物理主机为止.

③请求迁移.如果当前顶层拓扑内所有空闲物理主机数量小于  $V\_size$ ,则分配相应数量的虚拟机请求,并将多余的虚拟机请求移动到邻近的数据中心队列中,等待分配.如果邻近数据中心内资源同样不足,则继续迁移到其他数据中心.如果整个网络拓扑资源不足,则返回最近数据中心等待其他虚拟机释放资源。

均衡分配:根据虚拟机请求队列和相应的物理主机队列,进行虚拟机的最终分配.按照公式(1)的定义,将虚拟机分配到请求匹配度最好的物理主机上,完成分配过程。

图 4 展示了基于层次拓扑树的虚拟机节能分配算法过程.从图中可以看出,虚拟机的分配过程主要分为虚拟机请求分组、根据层次拓扑树搜索物理主机和虚拟机分配 3 个步骤.分布式数据中心由 2 个地理位置分散的小型数据中心 A 和 B 组成.4 个用户在不同的位置向数据中心提交了 7 个虚拟机请求.根据用户 IP 和物理中心的位置,  $V1 - V4$  虚拟机请求分配到了 A,  $V5 - V7$  分配到了 B.根据虚拟机请求的数量,并发从 A 和 B 中寻找合适数量的子拓扑,进行虚拟机的分配,正如  $V1, V3 - V7$  所示.然而,当子拓扑的可用物理主机数量小于虚拟机请求数量时,从其父节点的拓扑中寻找其他物理主机进行分配,如  $V2$  所示.在整个虚拟机的分配过程中,根据虚拟机请求与物理主机的匹配度公式(1),进行物理主机的分配.如图中  $V1, V3, V4$  分配到了

原拓扑的物理主机,  $V2$  因其匹配度较差从而分配到了原拓扑的父节点.整个节能分配算法 HTES 的分配过程如算法 2 所示。

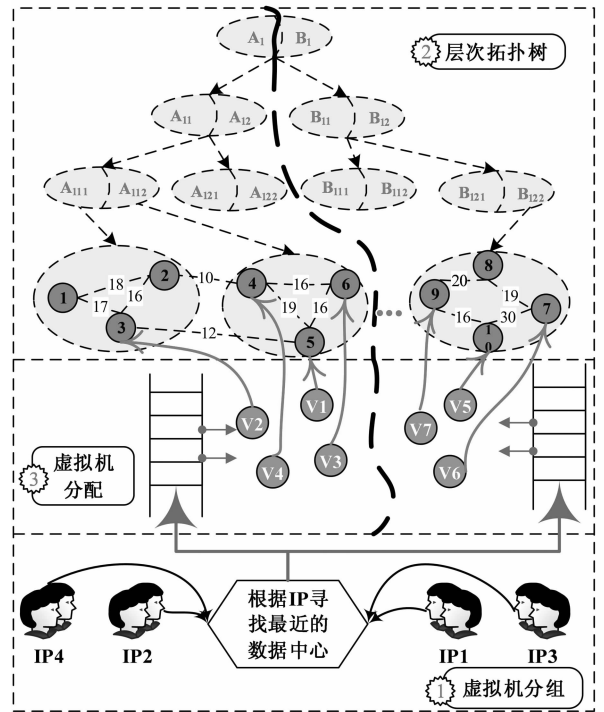


图 4 区域性虚拟机节能分配过程图  
Fig. 4 Regional virtual machine energy-saving allocation process

**算法 2** 基于层次拓扑树的虚拟机节能分配算法

输入:网络拓扑  $G$ ,虚拟机请求列表  $Vms$   
输出:虚拟机分配结果。

**Step1** 初始化.根据网络拓扑  $G$ ,利用算法 1 生成层次拓扑树  $TT$ ,监控数据中心所有物理主机的使用情况。

**Step2** 虚拟机请求分组.

- 1)根据虚拟机的 IP 和数据中心位置,将虚拟机分配到位置最近的数据中心排队队列中.
- 2)分配完成后,并发对多个数据中心进行虚拟机的分配.

**Step3** 区域性搜索物理主机.

- 1)根据分组完成后的虚拟机请求个数  $V\_size$ ,查询拓扑层次树中空闲节点数量(物理主机)充足的子拓扑,保证子拓扑的空闲数量大于虚拟机请求数量  $\alpha$  倍.
- 2)比较虚拟机请求大小与空闲物理主机的剩余能力,选择剩余能力大于请求大小的物理主机.



3)若物理主机搜索成功,数量等于  $\alpha \times V\_size$ , 进入 Step5 进行虚拟机分配. 否则,遍历当前子拓扑的父亲拓扑,再次搜索合适的数量主机,直到找到  $\alpha \times V\_size$  个物理主机为止.

4)当遍历到顶层拓扑,搜索完整个小数据中心的所有物理主机,物理主机数量不满足  $V\_size$  时,进入 Step4 进行虚拟机请求的迁移.

#### Step4 虚拟机请求的迁移.

1)当顶层拓扑内所有空闲物理主机数量小于  $V\_size$ ,则分配空闲主机数量的虚拟机请求,并将多余的虚拟机请求迁移到邻近的数据中心队列中,等待分配.

2)如果整个网络拓扑资源不足,则返回最近数据中心等待其他虚拟机释放资源.

**Step5 均衡分配虚拟机.**按照公式(1)的定义,根据请求匹配度的大小,将虚拟机分配到最好的物理主机上,均衡物理主机的多维资源,完成分配过程.

### 3 实验结果分析

为了验证 HTES 算法的性能,本文使用 Cloud-Sim3.5<sup>[17]</sup> 作为仿真平台,将 HTES 算法与贪婪算法 GRD<sup>[18]</sup>,蚁群算法 ANT<sup>[19]</sup> 和 DCEERS<sup>[12]</sup> 算法进行比较,从分配时间、物理主机数量、物理主机异构使用情况等方面进行了性能验证.

#### 3.1 实验环境

在 CloudSim 平台上,利用 brite 网络拓扑图生成工具,随机生成 10 个地理位置分散的数据中心网络拓扑,拓扑中包括 1 000 个物理主机和 200 个网络设备.表 2 为物理主机参数表.

表 2 物理主机参数表  
Tab. 2 The physical parameter of hosts

类型	CPU 核数	内存数/g	数量	功耗/(W · h <sup>-1</sup> )
G1	4	8	500	100
G2	8	8	300	180
G3	8	16	150	200
G4	16	24	50	350

此外,利用 Cloudsim 平台中的 DataCenter, Host, Vm, DataCenterBoker 等多个类,构建相应的 1 000 个物理主机属性信息和模拟对应的虚拟机请求信息.仿真了 4 种类型的物理主机模拟数据中心的异构特性:[〈4core-8g〉,〈8core-8g〉,〈8core-

16g〉,〈16core-24g〉]. 同样利用随机的方式,模拟不同用户在不同 IP 下的虚拟机请求,虚拟机请求数量从 10~500 随机波动.每个虚拟机的资源请求大小为 CPU[1~4],内存为[1~10]×512 M.

#### 3.2 参数设置

本节对实验中的参数值拓扑分割门限值 SplT, 拓扑合并门限值 MerT 和虚拟机分配算法中的  $\alpha$  参数进行讨论及测试.在网络拓扑分割的过程中,本文通过设定拓扑分割门限 SplT 和拓扑合并门限 MerT 来处理 2 分迭代分割出现的最下层子拓扑中节点数量较少的情况.本文通过固定合并门限,变动分割门限,或者固定分割门限,变动合并门限,通过不同的门限取值,观察不同设置下平均虚拟机分配时间.在网络拓扑中,对物理主机的搜寻速度越快,则相应的虚拟机分配时速度会更快.因此实验测试中,测试 100 个虚拟机请求的分配,设置拓扑分割门限值 SplT(10~20),合并门限值 MerT(0~5)时,从表 3 中可以看出,拓扑分割门限值 SplT(15~16),拓扑合并门限值 MerT(2~3)时,此时拓扑结构下物理主机的搜寻速度最快,虚拟机平均分配时间最少,显示出良好的分配性能.

表 3 100 个虚拟机下门限参数性能对比表  
Tab. 3 Threshold parameter performance comparison of 100 virtual machines

(SplT, MerT)	分配时间/s	(SplT, MerT)	分配时间/s
(10,3)	289	(18,3)	244
(11,3)	262	(19,3)	266
(12,3)	259	(20,3)	287
(13,3)	237	(15,0)	235
(14,3)	216	(15,1)	214
(15,3)	203	(15,2)	209
(16,3)	219	(15,4)	211
(17,3)	253	(15,5)	232

参数  $\alpha$  是从子拓扑空闲物理主机数与虚拟机请求数量的比值.如果从子拓扑空闲物理主机中等于虚拟机请求数量,则虚拟机的分配情况唯一,这种分配并非最优.因此从子拓扑空间选择空闲物理主机数量是要大于虚拟机的请求数量.本文在仿真实验中,设置固定虚拟机数为 100,变动  $\alpha$  取值为 1.0~2.0,从网络拓扑中,搜寻虚拟机请求数量  $\alpha$  倍的物理主机,表 4 给出了部分实验数据,可以看出,  $\alpha$  取值为 1.2~1.5 时,物理主机的平均利用率较高,浪费率较低.

表 4  $\alpha$  参数性能对比表  
Tab. 4 Parameter  $\alpha$  performance comparison

Value	虚拟机请求数	空闲物理主机数	平均利用率
1.0	100	100	0.71
1.2	100	120	0.73
1.4	100	140	0.72
1.6	100	160	0.68
1.8	100	180	0.63
2.0	100	200	0.59

通过对表 3 和表 4 的分析可知,当拓扑分割门限值 SplT (15~16)且拓扑合并门限值 MerT (2~3)时,算法拥有较快的物理主机搜寻速度;当  $\alpha$  取值为 1.2~1.5 时,物理主机的平均利用率较高,浪费率较低。

### 3.3 实验结果

本次实验从虚拟机分配时间、资源均衡度、资源浪费度、能耗和物理主机数量等 5 个方面对 HTES 算法进行验证。

图 5 为 4 种算法在不同虚拟机数量下的分配时间对比。从图 4 可以看出,随着虚拟机数量的不断增加(10~200),分配时间快速增大。在 4 种算法中,GRD 算法的分配时间最少,其次是 HTES 算法,ANT 算法时间最多。在 4 种算法中,GRD 算法随机分配虚拟机,将虚拟机分配到空闲资源最大的物理主机上,不考虑其他目标,其分配速度最快,HTES、DCEERS 和 ANT 以节能和提升底层资源利用率为虚拟机分配的目标,通过不同的分配策略实现节能,分配时间较 GRD 算法而言有所增加。HTES 算法在 3 种算法中分配时间最快。因为,HTES 算法将分布式数据中心网络拓扑进行分割,建立层次拓扑树,生成索引。根据用户 IP 将虚拟机请求分组,快速扫描层次拓扑,查询物理主机,将虚拟机请求集中的分配到底层物理主机,加快了分配效率,减少了物理主

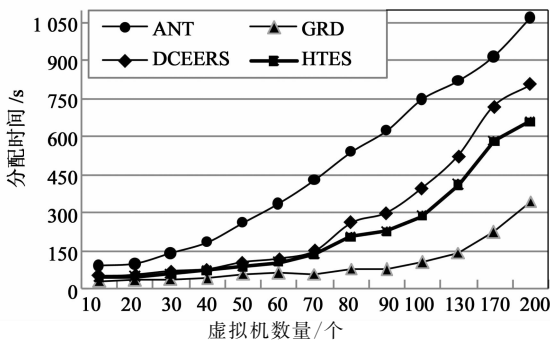


图 5 虚拟机分配时间对比图  
Fig. 5 Comparison of VM allocation time

机的使用数量,节约了能耗;DCEERS 算法将虚拟机分配问题看做最小商品流问题,通过 Benders 分解进行虚拟机分配,利用最小数量的物理主机承载虚拟机请求,同样减少了物理主机的数量,节约了能耗;ANT 算法利用蚁群策略求解多目标虚拟机的分配问题,同时优化底层资源的利用和能耗。然而 ANT 算法需要大量的迭代寻找最优,分配时间上较差。

物理主机内 CPU 和内存资源的均衡率对比如图 6 所示。从图 6 可以看出,随着虚拟机请求数量的不断增加(10~200),4 种算法的资源均衡程度不断变化。其中,HTES 算法均衡率最好,ANT 和 DCEERS 较差,GRD 最差。当虚拟机请求数为 10~50 时,GRD 和 ANT 的资源均衡逐级下降,HTES 和 DCEERS 两种算法围绕 0.65 和 0.6 两条水平线上下波动。当虚拟机请求数超过 50 时,ANT 和 GRD 的均衡率稍微回升,DCEERS 则开始下降,HTES 相对平稳且有所提升。在各种虚拟机请求数量下,HTES 都拥有较好的资源均衡率。因为,HTES 在进行虚拟机分配时,需要根据公式(1)计算虚拟机与物理主机间的匹配度,将虚拟机分配到合适的物理主机上,均衡物理主机多维资源的利用率;DCEERS 和 ANT 算法在尝试用最小的物理主机分配最多的虚拟机,随着虚拟机数量的增加,造成物理主机间资源的不均衡;GRD 算法随机分配虚拟机,当虚拟机数量较小时,占用底层高性能的物理主机,造成物理主机内的资源分配不均衡。随着虚拟机数量增加,高性能物理主机的资源利用率提升,各维资源的均衡度相应增加。

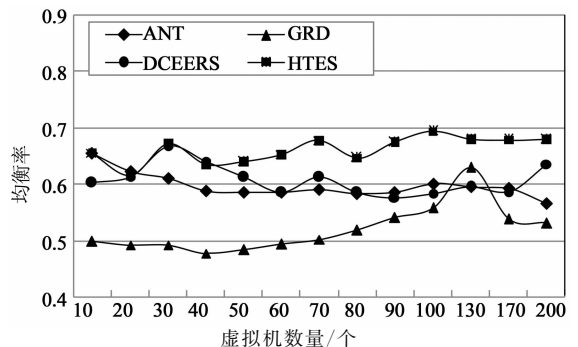


图 6 资源均衡率对比图  
Fig. 6 Comparison of resource balance rate

图 7 展示了 4 种算法的资源浪费情况。图中资源的浪费率是指 CPU 和内存浪费率的均值,从图 7 可以看出,在不同虚拟机请求数量下(10~200),4 种算法的资源浪费率不断变化,HTES 算法浪费率

最小,其次是 DCEERS 和 ANT,GRD 算法最大.在虚拟机数量较小时(10~30),HTES,DCEERS 和 ANT 3 种算法的资源浪费率相近,HTES 略差于 DCEERS 和 ANT 算法.随着虚拟机数量的不断增加,HTES 算法的浪费率明显优于其他 3 种算法.因为,GRD 算法的随机分配策略,并未考虑任何资源均衡目标,其分配时间快,资源浪费率高.DCEERS 和 ANT 通过蚁群和 Benders 分解,在优化虚拟机分配的同时,提高了资源利用率,节约了能耗;但是两者并未考虑物理主机多维资源的均衡分配问题.HTES 算法利用公式(1),在区域性分配虚拟机的同时,均衡地分配物理主机的多维资源(CPU 和 MEM),在提升底层资源利用率的同时,降低了各维资源的浪费率.

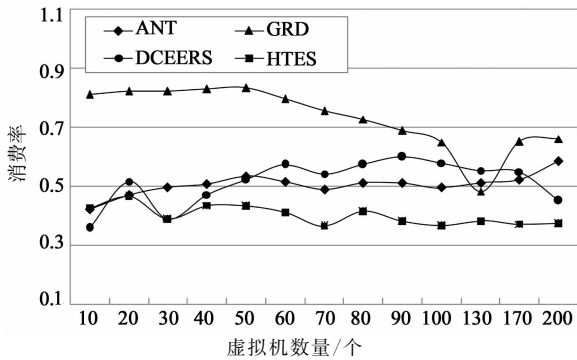


图 7 资源浪费率对比图

Fig. 7 Comparison of resource waste rate

图 8 为 4 种算法在不同数量虚拟机下占用物理主机的单位时间能耗情况,物理主机单位时间能耗是指虚拟机占用的所有物理主机功耗之和.从图 8 可以看出,随着虚拟机数据的不断增加(10~350),单位时间的功耗快速加大.4 种算法中 GRD 算法功耗增加最快,ANT 和 DCEERS 的功耗比较接近,HTES 算法的功耗最小.当虚拟机的数量为 10~150 时,其功耗增加速率相对较慢,4 条趋势线的坡度比较缓,HTES 算法功耗增加速度最慢.然而,随着虚拟机数量的不断增加,4 种算法的功耗急速增加,坡度加大,HTES 算法的功耗逐渐接近 ANT 和 DCEERS 算法.因为 HTES 算法、ANT 和 DCEERS 算法都是通过提供资源利用率,减少底层物理主机使用数量来节约能耗.HTES 算法额外考虑了物理主机多维资源均衡分配问题,进一步提升了物理主机的资源利用率.然而,随着虚拟机数量的不断增加,底层物理主机的空闲数量不断减少,单位时间产生的功效必然增加,最终变为一致(所有物理

主机都在使用).

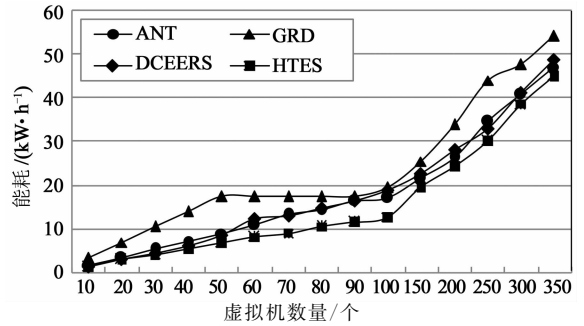


图 8 单位时间能耗对比图

Fig. 8 Comparison of energy consumption per unit time

图 9 为 4 种算法对底层异构物理主机的使用情况.从图 9 可以看出,4 种算法在相同虚拟机数量下占用的物理主机类型和数量各不相同.在 4 种算法中,GRD 算法偏向于占用高性能物理主机.在虚拟机数量较小(80~150)时,G4[16c,24g]类型的物理主机被全部占用.随着虚拟机数量的增加(150~300),G3[8c,16g]类型的物理主机被快速占用;相反,HTES 算法偏向于使用低性能的物理主机.在虚拟机数量较小(80~150)时,G1[4c,8g]和 G2[8c,8g]类型的物理主机被占用.随着虚拟机数量的增加(150~300),G3[8c,16g]和 G4[16c,24g]类型的物理主机少量被占用;ANT 和 DCEERS 算法则相对比较均衡,在各种虚拟机数量下,均衡占用 G1[4c,8g],G2[8c,8g],G3[8c,16g]和 G4[16c,24g]类型的物理主机.HTES 算法喜好占用低性能物理主机特性更方便进行分布式数据中心的节能.

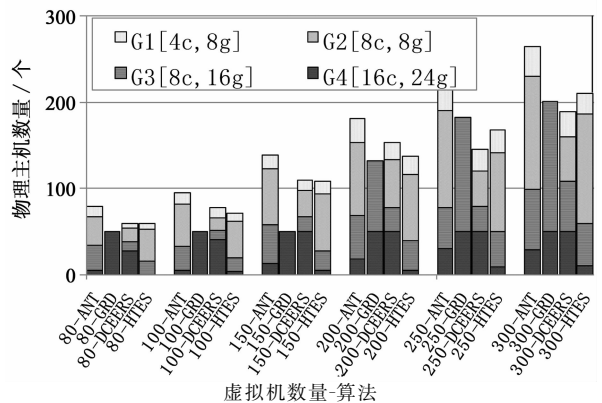


图 9 物理主机使用情况对比图

Fig. 9 Comparison of physical hosts usage

### 4 结 论

针对虚拟机分配中物理主机多维资源的浪费问

题,结合分布式数据中心网络拓扑的大规模性和地理位置的不同,本文提出了一种基于层次拓扑树的虚拟机节能分配算法.首先,将大规模网络拓扑按照地理位置的不同,划分成多个顶层子拓扑.其次,利用 Laplacian 矩阵和 Ncut 的特性,进行迭代二分切割,建立层次拓扑树.然后,根据虚拟机请求 IP 地址与数据中心位置的远近,将虚拟机请求分组,根据分组中虚拟机数量检索层次拓扑树,快速查询合适的区域性物理主机集合.最后,根据虚拟机请求与物理主机间的匹配度均衡分配物理主机的多维资源,提高了资源利用率,减少底层物理主机使用数量,通过关闭空闲物理主机实现了数据中心的节能.文中将 HTES 算法与 GRD 算法、DCEERS 算法和 ANT 算法进行比较,从虚拟机分配时间、资源均衡率、能耗和物理主机使用情况等方面验证了 HTES 算法的性能,表明 HTES 算法适合于分布式数据中心.

## 参考文献

- [1] ZHAO J, LI H, WU C, *et al.* Dynamic pricing and profit maximization for the cloud with geo-distributed data centers [C]//Conference on Computer Communications. Toronto: IEEE, 2014: 118–126.
- [2] TUDORAN R, COSTAN A, WANG R, *et al.* Bridging data in the clouds; an environment-aware system for geographically distributed data transfers[C]//Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on. Chicago:IEEE,2014: 92–101.
- [3] ALICHERRY M, LAKSHMAN T V. Network aware resource allocation in distributed clouds[C]//INFOCOM. Orlando: IEEE,2012: 963–971.
- [4] ISLAM M A, REN S, PISSINOU N, *et al.* Distributed temperature-aware resource management in virtualized data center [J]. Sustainable Computing: Informatics and Systems, 2015 (6): 3–16.
- [5] KO Y M, CHO Y. A distributed speed scaling and load balancing algorithm for energy efficient data centers[J]. Performance Evaluation, 2014, 79: 120–133.
- [6] 邓子云,章兢,白树仁,等.基于超级计算的 CAE 集成平台架构设计[J].湖南大学学报:自然科学版,2013,40(7): 80–85. DENG Ziyun, ZHANG Jing, BAI Shuren, *et al.* Architectural design of CAE integration platform based on super computation [J]. Journal of Hunan University: Natural Sciences, 2013, 40(7): 80–85. (In Chinese)
- [7] GEORGIU S, TSAKALOZOS K, DELIS A. Exploiting network-topology awareness for VM placement in IAAS clouds [C]//Cloud and Green Computing (CGC), 2013 Third International Conference on. Karlsruhe: IEEE, 2013: 151–158.
- [8] PENG Y, YUAN Y, HUANG X, *et al.* Research on maintainability of network topology for data centers [C]//2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN). Shanghai: IEEE, 2014: 317–321.
- [9] SHIRAYANAGI H, YAMADA H, KENJI K. Honeyguide: a VM migration-aware network topology for saving energy consumption in data center networks[J]. IEICE Transactions on Information and Systems, 2013, 96(9): 2055–2064.
- [10] JAIN N, MENACHE I, NAOR J S, *et al.* Topology-aware VM migration in bandwidth oversubscribed datacenter networks[M]. Berlin Heidelberg: Springer Berlin Heidelberg, 2012:586–597.
- [11] JIN H, CHEOCHERNAGNGARN T, LEVY D, *et al.* Joint host-network optimization for energy-efficient data center networking[C]// Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on. Boston: IEEE, 2013: 623–634.
- [12] SHUJA J, BILAL K, MADANI S A, *et al.* Data center energy efficient resource scheduling [J]. Cluster Computing, 2014, 17(4): 1265–1277.
- [13] HAMMADI A, MHAMDI L. A survey on architectures and energy efficiency in data center networks[J]. Computer Communications, 2014, 40: 1–21.
- [14] MOGHADDAM F A, LAGO P, GROSSO P. Energy-efficient networking solutions in cloud-based environments: a systematic literature review[J]. ACM Computing Surveys (CSUR), 2015, 47(4):1–32.
- [15] BELOGLAZOV A, ABAWAJY J, BUYYA R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing[J]. Future Generation Computer Systems, 2012, 28(5): 755–768.
- [16] GU L, ZENG D, GUO S, *et al.* Joint optimization of VM placement and request distribution for electricity cost cut in geo-distributed data centers [C]//Computing, Networking and Communications (ICNC), 2015 International Conference on. Garden Grove: IEEE, 2015: 717–721.
- [17] BUYYA R, RANJAN R, CALHEIROS R N. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: challenges and opportunities [C]// High Performance Computing & Simulation, HPCSim09, International Conference on. Leipzig: IEEE, 2009: 1–11.
- [18] MILLS K, FILLIBEN J, DABROWSKI C. Comparing VM-placement algorithms for on-demand clouds[C]// Cloud Computing Technology and Science (CloudCom), IEEE Third International Conference on. Athens: IEEE, 2011: 91–98.
- [19] GAO Y, GUAN H, QI Z, *et al.* A multi-objective ant colony system algorithm for virtual machine placement in cloud computing[J]. Journal of Computer and System Sciences, 2013, 79(8): 1230–1242.