



云环境下基于租户的虚拟机分层放置策略^{*}

周焱霞¹, 王 硕¹, 陈 珺², 张 钊², 李 京¹

(1. 中国科学技术大学计算机学院 合肥 230027; 2. 新华通讯社技术局 北京 100803)

摘 要: 云环境下租户申请虚拟机来部署应用或者计算任务, 这些虚拟机之间的通信较频繁, 需要消耗较多的网络资源。为了减少网络资源的消耗和物理主机资源的浪费, 提出了基于租户的初始化放置算法, 并结合运行时放置算法调节虚拟机的位置。仿真结果表明, 新算法在达到放置目标的同时, 减少了需要重新放置的虚拟机数目, 降低了迁移代价。

关键词: 云计算; 通信代价; 虚拟机放置

doi: 10.3969/j.issn.1000-0801.2014.11.016

Tenant-Based Hierarchical Virtual Machine Placement Policy in Cloud

Zhou Yanxia¹, Wang Shuo¹, Cheng Jun², Zhang Zhao², Li Jing¹

(1. School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China;

2. Xinhua News Agency, Beijing 100803, China)

Abstract: In the cloud ecosystem, regulating virtual machine's position can help reduce the communication cost between virtual machines and the waste of physic machine's resource. Considering the target of placement in the initialization placement, an initialization placement algorithm base on tenant was proposed. Combined the algorithm with runtime placement algorithm, the placement goal was realized. Simulation results show that the proposed algorithm reduces the number of virtual machines that need to reposition and the cost of migration.

Key words: cloud computing, communication cost, virtual machine placement

1 引言

虚拟化技术的快速发展推动了云计算的发展, IaaS 提供商提供给用户虚拟机, 租户租用虚拟机完成自己的任务。因此, 如何放置这些虚拟机来降低物理主机的资源浪费, 同时减少虚拟机之间的通信代价, 降低网络资源消耗, 成为了一个非常值得研究的问题。

目前虚拟机放置算法大多是研究运行时放置^[1-3], 即为

了达到某个目标(如减少虚拟机之间的通信代价), 根据虚拟机的运行情况为虚拟机重新选择放置的物理主机。重置虚拟机需要对虚拟机进行迁移, 代价较大。虚拟机的初始位置对虚拟机的重新放置有很大影响。目前的虚拟机初始化放置算法大多是针对单个虚拟机的, 将物理主机的 CPU、内存、磁盘 I/O、网络 4 项资源的一项或多项作为指标为物理主机评级, 根据一定的策略(如最佳适应算法等)为虚拟机选择物理主机, 这种初始化放置算法并没有考虑

^{*} 新闻报道指挥调度供稿与采编发综合技术及系统研发基金资助项目(No.2012BAH17B00), 继续教育与科教融合工程——科教融合的资源中心与接入平台基金资助项目



虚拟机之间的联系。

本文提出了基于租户的虚拟机初始化放置算法,并将该算法与运行时放置算法结合,研究虚拟机的放置问题,在达到放置目标的同时,减少虚拟机重置时的迁移代价。当虚拟机初始化放置时,将同一租户的虚拟机当作一个整体处理;当虚拟机运行时,根据监控数据对虚拟机位置进行调整,最终达到最小化虚拟机之间的通信代价和最小化物理主机资源浪费的目标。

2 问题描述

虚拟机放置问题分为初始化放置和运行时放置^[4]。初始化放置是指在合适的物理主机上创建、部署虚拟机;运行时放置是指在初始化放置后,根据物理主机和虚拟机的运行数据从不同的角度出发,将虚拟机重新放置到另外一台物理主机上。本文将结合初始化放置和运行时放置,从全局的角度研究虚拟机放置问题,主要考虑以下3个约束条件。

(1) 资源限制

保证物理主机的资源充足,在运行过程中将物理主机资源(包括CPU、内存、网络等)的利用率控制在一定的阈值范围内,从而保证虚拟机的性能。

(2) 资源浪费和竞争

物理主机的资源可以用一个多维向量表示,不同的虚拟机放置方案使每个主机拥有的剩余资源不同。若各维的剩余资源平衡,则可以更好地使用这些资源;若剩余资源不平衡,将造成资源的浪费。如参考文献[5]将两个运行通信密集型任务的虚拟机放置在同一个物理主机上,两个任务的完成时间比两个虚拟机各自运行在不同物理主机上的完成时间多21.5%~22.3%。这是由于两个任务竞争网络资源,使网络资源成为瓶颈,造成任务完成时间的极度增加,而此时物理主机上的CPU、内存等资源还相对充足,但是该物理主机已不适合放置有通信需求的任务,这就造成

了CPU和内存资源的浪费。参考文献[6]中用 R_i 表示第 i 维剩余资源占第 i 维总资源的比例,当物理主机有 n 维资源时,浪费的总资源为 $W = \sum_{k=1}^n |R_k - R_1|$ 。因此, W 越大表示物理主机的资源浪费越严重。

(3) 通信代价

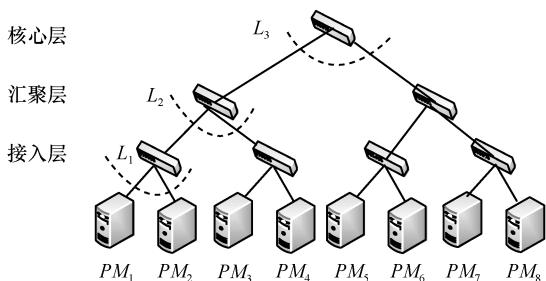
虚拟机之间的通信需要消耗物理主机的网络带宽,其通信代价取决于它们所在的物理主机之间的通信代价。

3 数据中心建模

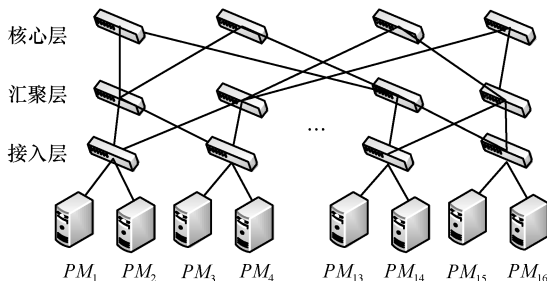
目前云计算中心网络拓扑大多是3层架构,包括核心层、汇聚层和接入层。常见的几种拓扑包括树型结构、胖树型结构^[7]等,如图1所示。不同的拓扑逻辑都可以转换成树型拓扑^[3]。

为了便于描述,将接入层、汇聚层和核心层的交换机分别记为 L_1 、 L_2 和 L_3 , L_1 下面管辖的物理主机记为 L_1 的子节点,与 L_2 相连的 L_1 记为 L_2 的子节点,与 L_3 相连的 L_2 记为 L_3 的子节点。实体 E 的容量 S 表示 E 上还可以启动的虚拟机数量,其中实体 E 可以是交换机 L_1 、 L_2 、 L_3 中的任意一个,实体 E 的容量是其子节点的容量之和。 L_i 用三元组 (ID, S, L_{i-1}) 表示,其中 ID 是 L_i 的标识, S 表示 L_i 所能容纳的虚拟机数量, L_{i-1} 是 L_i 的子节点。物理主机的资源由多个 $(key: value)$ 对表示,其中 key 可以是CPU、RAM等, $value$ 表示相应的值。

一台物理主机的资源可以由CPU、内存、磁盘、网络带宽等描述。在申请虚拟机时,虚拟机的CPU数目和内存大小是确定的,而磁盘I/O和网络带宽是未知的,它们是根据虚拟机的实际运行情况确定的。因此本文采用二维向量 $CAP = (CAP_{CPU}, CAP_{RAM})$ 表示物理主机的资源, $VCAP = (VCAP_{CPU}, VCAP_{RAM})$ 表示虚拟机的资源,由于虚拟机的CPU和RAM的使用率一般不会达到100%,因此为了提高物理



(a) 树型结构



(b) 胖树型结构

图1 云计算中心网络拓扑

主机的资源利用率,现在的虚拟化产品允许给客户机分配的 CPU 和内存数量都多于物理主机上实际有的资源,即支持 CPU、内存的过载分配^[8],过载比例分别记为 O_C 和 O_R 。若按 CPU 过载比率计算,该物理主机可以开启 $CAP_{CPU} \times \frac{(1+O_C)}{VCAP_{CPU}}$ 台虚拟机;若按 RAM 的过载比率计算,该物理主机可以开启 $CAP_{RAM} \times \frac{(1+O_R)}{VCAP_{RAM}}$ 台虚拟机。因此,最终该物理主机可以开启的虚拟数量为 $\min(CAP_{CPU} \times \frac{(1+O_C)}{VCAP_{CPU}}, CAP_{RAM} \times \frac{(1+O_R)}{VCAP_{RAM}})$,由此可以得到交换机 L_1, L_2, L_3 所能开启的虚拟机数量。

虚拟机之间的通信代价主要由它们所在的物理主机之间的通信代价决定。本文将物理主机之间最短路径所经过的交换机数量作为物理主机之间的通信代价,将在同一台物理主机上的虚拟机之间的通信代价定义为 1,与同一集群下虚拟机之间的通信代价相同,这样避免为了降低虚拟机之间的通信代价,而优先将同一个租户的虚拟机放置在同一个物理主机上的情况,减少了对同一物理主机资源的竞争。图 1(a)所示的树型网络拓扑的通信代价矩阵可以表示为:

$$C = \begin{bmatrix} 1 & 1 & 3 & 3 & 5 & 5 & 5 & 5 \\ 1 & 1 & 3 & 3 & 5 & 5 & 5 & 5 \\ 3 & 3 & 1 & 1 & 5 & 5 & 5 & 5 \\ 3 & 3 & 1 & 1 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 1 & 1 & 3 & 3 \\ 5 & 5 & 5 & 5 & 1 & 1 & 3 & 3 \\ 5 & 5 & 5 & 5 & 3 & 3 & 1 & 1 \\ 5 & 5 & 5 & 5 & 3 & 3 & 1 & 1 \end{bmatrix} \quad (1)$$

虚拟机之间的通信权重 T 表示单位时间内虚拟机之间的通信量和通信代价的乘积。假设两个虚拟机在同一个交换机 L_1 下,它们之间的通信代价最小,则这两个虚拟机之间的通信权重 T 也达到了最小值,记为 $T_{\min}$ 。

4 虚拟机的放置过程

为了解决虚拟机放置问题,本文设计了一个分层的虚拟机放置过程,为租户的请求生成一个合适的放置方案,如图 2 所示。首先对数据中心建模,并根据用户的请求和虚拟机的初始放置策略部署虚拟机;然后监控部署好的虚拟机运行情况,得到虚拟机之间的关系图,分析关系图并找到需要重新放置的虚拟机;最后根据运行时放置算法重新放置这些虚拟机。在虚拟机的放置过程中,最重要的就是初始化放置和运行时放置。

4.1 虚拟机初始化放置

常用的初始化放置算法为随机算法,即随机选择一个满足虚拟机资源需求的物理主机放置虚拟机,而 OpenStack^[9]中采用资源(CPU、内存等)评级的方法,选择评分最高的物理主机放置虚拟机,这些算法都是将每个虚拟机作为独立个体处理。云环境下大部分的租户一次申请多个虚拟机,且同一租户虚拟机之间的通信比不同租户虚拟机之间的通信更频繁,虚拟机的运行模式更相似。考虑到同一租户虚拟机之间的这些特性,本文将一个租户的虚拟机作为一个整体考虑,更有利于降低虚拟机之间的通信代价,减少物理主机的资源浪费。

本文提出了基于租户的虚拟机放置算法,放置过程分为两个步骤,具体介绍如下。

(1) 为虚拟机选择目标集群

为虚拟机选择目标集群是指确定将虚拟机放置在哪些接入交换机下,目标是使虚拟机之间可能存在的通信权值最小。为了计算方便,假设同一租户的所有虚拟机相互之间都存在通信,并假设它们之间的通信量为 1 个单位,则同一租户的虚拟机之间的通信权值 $TW = \sum_{i \neq j} cost_{ij}$,其中 $cost_{ij}$ 表示虚拟机 i 和 j 之间的通信代价。

(2) 将虚拟机映射到集群的物理主机上

目标是将虚拟机均匀分配在目标集群内,由于一个用户申请的虚拟机的运行任务有相关性,这将导致其对资源

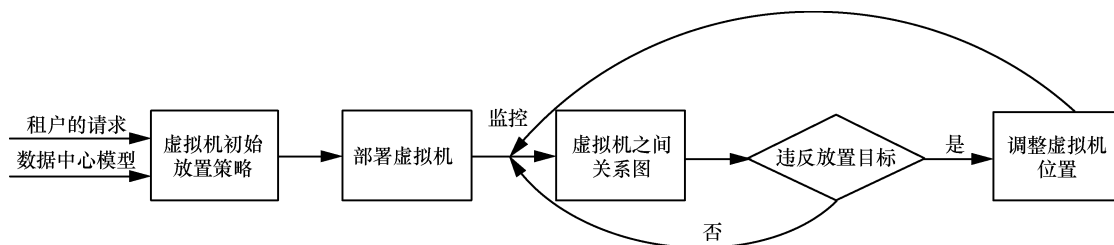


图2 虚拟机分层放置过程



的需求可能也有相关性,因此尽量使虚拟机均匀地分布在物理主机上。例如将都运行 CPU 密集型或都运行 I/O 密集型任务的虚拟机放置在同一个物理主机上,会导致剩余资源的不均衡或者资源竞争,最终造成资源的浪费。假设一个用户申请 N 个虚拟机, D 表示虚拟机分配的均衡性,则有 $D = \sum v^2$, 其中 v 表示这次请求分配在目标集群内各个主机上的虚拟机数目。

基于租户的初始化放置算法的目标为 $\min D$ 和 $\min TW$, 约束条件为 $\sum_{j=1}^M a_{ij} = 1$ 和 $\sum_{i=1}^N a_{ij} \leq S_j$, 其中 $i = [1, 2, \dots, N]$, $j = [1, 2, \dots, M]$, M 表示物理主机数量, N 表示用户请求的虚拟机数量, S_j 表示物理主机 j 所能容纳的虚拟机数目, a_{ij} 表示虚拟机 i 是否放置在物理主机 j 上, 若是, $a_{ij} = 1$; 否则, $a_{ij} = 0$ 。

通过以上分析可知,基于租户的初始化放置算法的目标是最小化同一租户虚拟机之间的通信权值,并且使虚拟机均匀地分布在目标集群上;约束条件是使一个虚拟机只放置在一个物理主机上,并且每个物理主机上放置的虚拟机数目都小于或者等于其容量。基于租户的初始化放置算法的具体过程如下。

输入: 用户请求的虚拟机数 N 、通信代价矩阵 C 、树型拓扑。

输出: 虚拟机到物理主机的映射矩阵。

begin

CTreeXtemp = Tree

if(temp.data < N)

return false

else

if(temp->ChildPtr == NULL)

if(temp->data >= N)

putVMonCluster(temp, N)

else

putVMonCluster(temp, temp->data)

VMInitPlacement(N-temp->data, C, temp->next)

else

VMInitPlacement(N, temp->ChildPtr, C, temp)

end

假设云计算中心的资源如图 3 的左侧树型结构所示, 圆圈中的数字表示该实体下的物理主机所能容纳的虚拟机总数。当某租户申请 7 个虚拟机时, 首先判断云中容量是否满足租户的请求, 由于实体 E_1 的容量为 38, 大于 7, 接着判断实体 E_1 的左孩子实体 E_2 的容量为 27, 也大于 7, 则继续将请求向下转发到实体 E_4 , 实体 E_4 的容量是 15, 大于 7, 则选定实体 E_4 为目标集群, 该租户请求的虚拟机将调度到该集群内。同时, 调整该请求经过路线上的各交换机的容量, 并将云中资源的状态调整为图 3 中右侧所示的树型结构。

4.2 虚拟机运行时放置

云平台是一个动态环境, 包括虚拟机频繁的启停、虚拟机动态的创建和删除、虚拟机负载的动态变化。由于虚拟机在初始放置时并不知道虚拟机之间具体的通信模式, 因此在初始放置时可能违反了放置目标, 此时就需要对虚拟机的位置进行动态调整。

虚拟机的运行时放置主要分为两个步骤, 具体介绍如下。

(1) 找出需要重新放置的虚拟机

需要重新放置的虚拟机包括两大类: 一是虚拟机之间的通信权重比率超过阈值; 二是物理主机的资源浪费量超过阈值。针对第一种情况, 首先需要获得虚拟机之间的关系图, 如图 4 所示, 遍历关系图并找出图中所有的连通分量, 同属于一个连通分量的虚拟机之间存在通信, 然后计算出连通分量中所有虚拟机所属的实体。若连通分量中存在不同的实体, 则将该连通分量内所有虚拟机标记为待迁

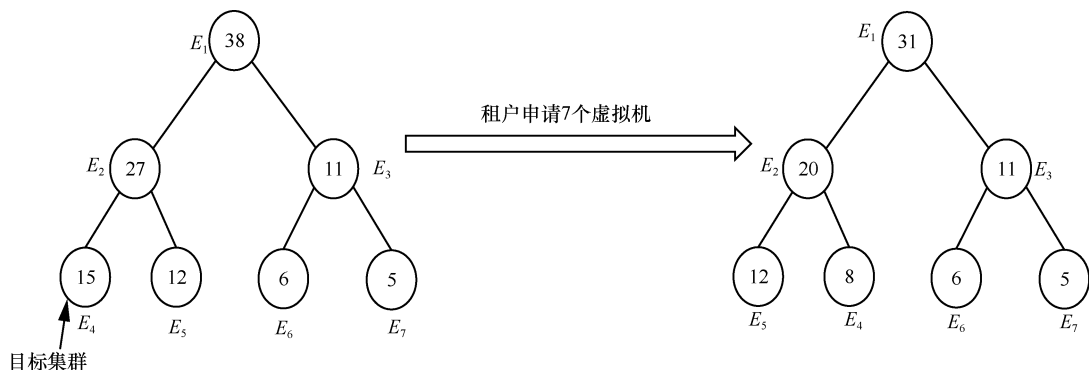


图 3 虚拟机的初始化放置及云中资源状态的变化

移虚拟机,综合考虑所有待迁移的连通分量中的虚拟机,根据物理主机的容量限制和目前该连通分量的虚拟机主要所在的目标集群,为这些虚拟机选择目标集群。针对第二种情况,需要计算每个物理主机的资源浪费量,如果资源浪费量超过阈值,则从该物理主机中找出消耗资源最多的虚拟机,并将该虚拟机加入待重新放置的虚拟机列表 VM_{list} 中。

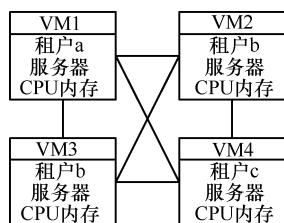


图4 虚拟机之间的关系

(2)将 VM_{list} 中的虚拟机放置到云平台中

虚拟机运行时放置的目标是最小化目标集群的资源浪费量和通信量,即 $\min T$ 和 $\min W$,其约束条件与基于租户的初始化放置算法的约束条件相同,运行时放置算法的具体过程如下。

输入:虚拟机关系图 G 、通信代价矩阵 C 。

输出:新的虚拟机和服务器的映射关系。

begin

$D_{list} = \text{CalculateDofPM}()$

for D_i in D_{list}

if($D_i > \text{MAXD}$)

找到消耗该维资源多的虚拟机

将该虚拟机加入 VM_{list} 中

end

end

遍历虚拟机关系图,找到连通分量

for 图中的连通分量

计算连通分量的虚拟机的所属集群

if(连通分量的虚拟机不属于一个集群)

标记这个连通分量,加入待迁移列表

end

end

确定每个连通分量放置的集群

将需要迁移的虚拟机加入 VM_{list}

for VMs in VM_{list}

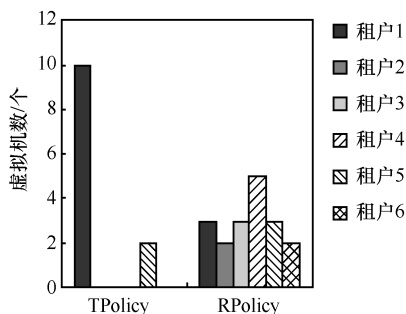
AddVMstoCloud(VMs)

end

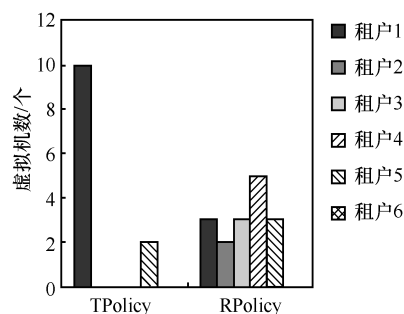
end

5 仿真实验分析

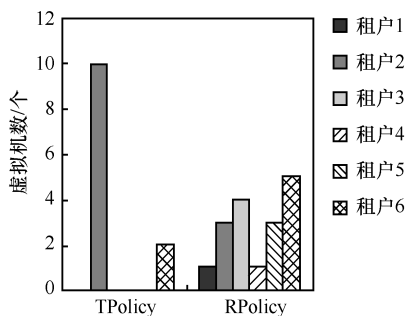
对图 1 (a) 所示的数据中心网络体系结构进行仿真,



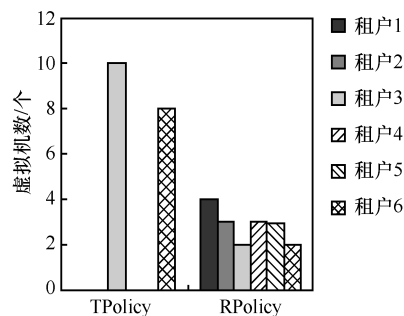
(a)集群0上各租户的虚拟机的分布



(b)集群1上各租户的虚拟机的分布



(c)集群2上各租户的虚拟机的分布



(d)集群3上各租户的虚拟机的分布

图5 租户的虚拟机在各集群内的分布



包括1个核心层交换机、2个汇聚层交换机、4个接入层交换机。每个接入层交换机下有2个物理主机,每个物理主机可以容纳9个虚拟机。将每个接入层交换机下的物理主机作为一个集群,分别记为“集群0”、“集群1”、“集群2”和“集群3”。假设有6个租户,每个租户申请10个虚拟机,则基于随机算法的初始化放置算法(RPolicy)和基于租户的初始化放置算法(TPolicy)的仿真结果如图5所示。

仿真中对不同的租户模拟不同的虚拟机通信模式,并设定虚拟机之间的通信流量。由图5可以看出,RPolicy的通信代价是TPolicy的1.6倍。

根据运行时虚拟机放置算法得出,TPolicy需要迁移的虚拟机数为4个,而RPolicy的待迁移虚拟机数为23个,迁移后通信代价分别降低了8.1%和43.5%,见表1。

表1 运行时算法调整结果

初始化 放置算法	需迁移的 虚拟机数目/个	迁移后通信 代价下降量
TPolicy	4	8.1%
RPolicy	23	43.5%

综合以上分析可知,在IaaS云环境下虚拟机之间的通信主要集中在同一租户的虚拟机之间,因此将同一租户的虚拟机放置在一个集群下,不仅可以降低虚拟机之间的通信代价,还可以减少运行时需要迁移的虚拟机数,从而降低迁移代价。

6 结束语

本文对云平台下虚拟机的放置进行了研究。首先对数据中心进行建模,然后将虚拟机的初始化放置和运行时放置结合,提出了基于租户的虚拟机放置算法,降低了虚拟机之间的通信代价,减小了虚拟机之间的资源竞争。下一步工作是通过进一步观察云环境下虚拟机和物理主机的运行数据,提出更加有效的虚拟机放置算法,丰富和完善虚拟机放置策略。

参考文献

- 1 Pan F, Jiang C f, Xu X H, *et al.* Placement strategy of virtual machines based on workload characteristics. *Journal of Chinese Computer Systems*, 2013, 34(3): 520~524
- 2 Meng X Q, Vasileios P, Zhang L, *et al.* Improving the scalability of data center networks with traffic-aware virtual machine

placement. *Proceedings of IEEE INFOCOM*, San Diego, CA, USA, 2010: 1154~1162

- 3 Biran O, Corradi A, Fanelli M, *et al.* A stable network-aware VM placement for cloud systems. *Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Ottawa, Canada, 2012: 498~506
- 4 Hu Y Y, Lin H, Li H B. Minimum-migration-cost VM placement in IaaS cloud. *Journal of Chinese Computer Systems*, 2014, 35(4): 878~882
- 5 Wu S, Wang F, Jin H, *et al.* Network I/O load based virtual machine placement algorithm in HPC cloud environment. *Development and Application of High-Performance Computing*, 2013(1): 50~57
- 6 Jing X, Fortes J A B. Multi-objective virtual machine placement in virtualized data center environments. *Proceedings of IEEE/ACM Intentional Conference on Green Computing and Communications & Intentional Conference on Cyber, Physical and Social Computing*, Hangzhou, China, 2010: 179~188
- 7 Leiserson C E. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Transactions on Computers*, 1985, 34(10): 892~901
- 8 KVM. <http://linux-kvm.org>, 2014
- 9 OpenStack. <http://openstack.org>, 2014

[作者简介]



周焱霞,女,中国科学技术大学硕士生,主要研究方向为云监控、虚拟机放置。

王硕,男,中国科学技术大学博士生,主要研究方向为分布式系统。

陈琨,女,新华通讯社技术局高级工程师,主要研究方向为计算机软件与理论。

张钊,男,新华通讯社技术局工程师,主要研究方向为计算机应用。

李京,男,博士,中国科学技术大学教授,主要研究方向为云计算、分布式系统、移动计算。

(收稿日期:2014-10-15)