软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(7):1876−1887 [doi: 10.13328/j.cnki.jos.004918]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 基于负载高峰特征的虚拟机放置算法*

徐思尧[1], 林伟伟[1], 王子骏[2]

[1](华南理工大学 计算机科学与工程学院,广东 广州 510006)

[2](School of Computing, Clemson University, Box 340974, Clemson, SC 29634-0974, USA)

通讯作者: 林伟伟, E-mail: linww@scut.edu.cn, http://www.scholat.com/linweiwei

**摘 要**: 提出了一种基于虚拟机负载高峰特征的虚拟机放置策略,通过更好地复用物理主机资源来实现资源共享,从而提高资源利用率.在云环境下,当多个虚拟机的负载高峰出现在相同的时间段内时,非高峰时段的资源利用率就会明显偏低;相反,多个虚拟机只要负载高峰能错开在不同的时间,闲置的资源就能更充分地被利用.由于应用的负载通常具有一定的周期性,因此,可以利用虚拟机负载的历史数据作为分析的依据.基于虚拟机的负载高峰特征对虚拟机负载进行建模,建立虚拟机负载之间的相似度矩阵来实现虚拟机联合放置.使用CloudSim模拟实现了所提出的算法,并与基于相关系数的放置算法、随机放置算法进行了比较.实验结果表明:所提算法在平均 CPU 利用率上有 8.9%~12.4%的提高,主机使用量有 8.2%~11.0%的节省.

**关键词**: 高峰特征;相似度;资源复用;虚拟机放置;云计算

**中图法分类号**: TP316

中文引用格式: 徐思尧,林伟伟,王子骏.基于负载高峰特征的虚拟机放置算法.软件学报,2016,27(7):1876−1887. http://www.jos.org.cn/1000-9825/4918.htm

英文引用格式: Xu SY, Lin WW, Wang ZJ. Virtual machine placement algorithm based on peak workload characteristics. Ruan Jian Xue Bao/Journal of Software, 2016,27(7):1876−1887 (in Chinese). http://www.jos.org.cn/1000-9825/4918.htm

## Virtual Machine Placement Algorithm Based on Peak Workload Characteristics

XU Si-Yao[1], LIN Wei-Wei[1], James Z. WANG[2]

[1](School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China)

[2](School of Computing, Clemson University, Box 340974, Clemson, SC 29634-0974, USA)

**Abstract**: This paper proposes a novel method to allocate virtual machines by statistical resource multiplexing based on the characteristics of virtual machine's peak workloads. In a cloud environment, if the peak workloads of multiple virtual machines overlap, the resource utilization of the cloud system can be significantly low when all these virtual machines enter the non-peak workload phases. If the overlap of workload peaks among virtual machines can be avoided, resource utilization will not fluctuate so much (heavily loaded during peak period and largely idle during non-peak period). Since the workload of an application usually follows a cyclic pattern, historical data can be analyzed to predict future workload. This paper models the workload characteristics of virtual machines through monitoring their peak workloads. A similarity matrix of VM's workloads is used to allocate virtual machines so that their workload peaks will not overlap. The performance study using CloudSim demonstrates that the proposed virtual machine allocation algorithm improves the CPU utilization by 8.9% to 12.4% under different workloads compared to the random allocation algorithm. The number of hosts needed by the algorithm is also reduced by 8.2% to 11.0% under the same workload requirements.

* 基金项目: 国家自然科学基金(61402183); 广东省科技计划(2016A010101007, 2016B090918021, 2014B010117001, 2014A010103022, 2014A010103008, 2013B010202001); 广州市科技计划项目(201607010048, 201604010040); 中央高校科研业务费(2015ZZ0038)

Foundation item: National Natural Science Foundation of China (61402183); Guangdong Provincial Science and Technology Projects (2016A010101007, 2016B090918021, 2014B010117001, 2014A010103022, 2014A010103008, 2013B010202001); Guangzhou Civic Science and Technology Project (201607010048); Fundamental Research Funds for the Central Universities, SCUT (2015ZZ0038)

收稿时间: 2014-12-01; 修改时间: 2015-04-09; 采用时间: 2015-09-16

**Key words**: peak characteristics, similarity, resource multiplying, placement of virtual machine, cloud computing

云计算[1−3]是最近几年备受业界关注的新热点,而虚拟化技术是实现云计算环境的重要手段.通过虚拟化技术,云数据中心借虚拟机共享物理机资源池的方式,以物理机单元承载虚拟机,以虚拟机单元承载应用服务.云数据中心已经成为一个高性能计算机的集合,具有成千上万的物理服务器和网络设备.由于资源量大、异构性强,云数据中心对按需服务、资源动态弹性管理、服务质量等多方面的要求更加突出.但是,云数据中心现在面临着低效率、高成本、高能耗等问题.资料表明:我国的云数据中心的资源利用率普遍不高,平均只有 10%左右,而且服务器在较大一部分时间内处于空闲状态,即使服务器处于空闲状态,也会带来满负荷时 60%的功耗[4].目前,大部分的云数据中心的资源分配和管理大多是静态的,容易造成过度配置或过低配置两种极端状况.文献[5]显示,IBM 数据中心的服务器的整体利用率仅在 11%~50%之间.在需求的驱动下,数据中心的虚拟机数量与日俱增,虚拟机集群的资源调度迎来了新的考验.在拥有大规模虚拟机集群的云数据中心,虚拟机的数目和负载会随应用与用户的需求而经常变化,静态的资源调度分配方式往往会导致虚拟机出现资源浪费或资源不足的情况,而通过人工的动态资源调整的方式则会有明显的滞后性.所以,如何通过合理的虚拟资源的调度管理改进 IT 资源的供给与管理模式、提高资源利用率、降低服务成本,且同时保证服务质量,是当前亟需改善和解决的重要问题.合理的资源分配方案不仅能满足不同用户的各种需求,还能极大程度地提高资源利用率,避免资源的浪费和闲置.云计算系统的性能很大程度上是由虚拟机资源调度策略的优劣决定的.因此,云环境下的资源调度和虚拟机放置策略问题是当前云计算的研究热点,得到国内外的专家、学者们的广泛关注.

虽然近几年国内外学者在云计算资源管理和虚拟机资源调度优化方面进行了不少研究,然而,本项目组在前期的云计算调度相关研究[3,6−9]的基础上发现,在云数据中心的虚拟机资源调度方面仍然需要解决的问题有:

1) 虚拟机资源分配与调度未能实现面向应用的动态自适应.一般 Internet 应用的访问负载是随时间变化而变化的,如何根据应用负载模式,结合虚拟机资源的特征进行虚拟机资源的多目标优化,暂时没有很好的解决方法;
2) 为了实现应用自适应负载的需求,虚拟机放置的问题将复杂化.现阶段学术界和工业界重点关注虚拟机与物理主机之间的容量映射(NP 难问题),但没有考虑虚拟机承担的应用对资源的消耗类型、用户对应用的访问的时间与需求模式以及虚拟机应用负载特征匹配关系等因素;
3) 虚拟机迁移策略:当下学术界关于虚拟机迁移策略的研究主要考虑独占性资源,比如 CPU、内存、带宽等的利用率以及能耗,却未综合考虑共享性资源 I/O 的情形以及也未考虑虚拟机本身具有的本地动态伸缩能力.所以,综合考虑各方面因素,需要确定虚拟机迁移的时机、频率等因素,以确保迁移后在较长时间内能够保持稳定.

以上 3 个问题出现的根本原因是虚拟机的资源需求特征与物理主机资源配置的不匹配,所以,考虑虚拟机资源需求特征、虚拟机的资源配置与负载模式匹配关系等因素的虚拟资源调度将一直是需要研究与解决的一个基础性问题.

针对云数据中心的虚拟机资源联合调度问题,学者们在文献[10−13]中讨论了基于资源复用的虚拟机联合调度方法,提出的资源复用方法可以利用多个虚拟机的资源需求互补特征更好地实现物理机资源的共享,提高资源利用率和资源分配的服务质量.为了对虚拟机能资源互补地联合调度,需要描述虚拟机资源负载的互补性.从数学层面上看,我们可以直接想到的有欧氏距离、相关系数、余弦值等,这些数学量在一定程度上也能描述虚拟机负载的互补性.但我们也会感觉这些量描述得不够准确,因为虚拟机的资源互补,最主要是为了资源需求的错峰利用,只要负载高峰不重叠,就能提高资源的利用率.就是说,虚拟机的高峰负载是虚拟机负载的很重要的特征.而欧氏距离、相关系数、余弦值等这些基本数学量在描述负载相关性的过程中,无差别地对待高峰负载与非高峰负载,因而描述不够充分、细致.如何从数学的角度上更准确地描述负载资源的互补性,从而更好地指导虚拟机联合调度,是本文关注的重点.

为了克服传统参数对虚拟机负载特征的不充分、不准确,更好地实现虚拟机资源的复用,本文提出了一种

基于负载高峰特征的虚拟机联合放置算法,通过把负载高峰出现在不同时间的虚拟机加以联合,实现资源互补的效果,极大地提高了资源的利用率.它考虑到不同的虚拟机不同的时间具有不同的负载模式,也就是说,不同类型的虚拟机对某种资源的需求是不同的,这种不同体现在数量上或者时间分布上.一个虚拟机,在其初始化时会被配置固定的资源,比如 CPU、内存、带宽等,但通常在实际运行过程中,制约应用吞吐量的是某一种资源,本文选取CPU作为研究对象,研究虚拟机之间的CPU负载关系.当多个虚拟机的负载高峰出现在相同的时间段内时,非高峰时段的资源利用率就会明显偏低;相反,多个虚拟机只要负载高峰能错开在不同的时间,闲置的资源就能更充分地被利用.因此,负载高峰就是这个虚拟机的负载的最大特征.高峰是一个相对的概念,负载量多大才被认为是负载高峰呢?本文提出了两种负载高峰的定义.

(1) 以固定的某个数作为负载高峰的分界线.比如资源利用率大于 0.5 的,我们认为就是负载高峰,否则就是非高峰;

(2) 负载的平均值乘以一个固定的比率作为负载高峰的分界线.比如某资源平均利用率为 0.2,固定比率我们设为 1.5,则利用率大于 0.2×1.5=0.3 的情况即视为负载高峰,否则视为非高峰.

为了比较不同虚拟机负载的相似程度,本文引入高峰相似度公式,这个公式是在余弦相似度公式的基础上,对于负载高峰,我们赋予更大的权重,以突出负载高峰的特征.最后,在虚拟机放置策略上,核心是贪心算法,与其他虚拟机高峰负载相似度低的虚拟机将会优先选择放置.并且通过在CloudSim上进行模拟实验,验证了算法的有效性.

本文第 1 节介绍相关研究现状,主要介绍云计算资源复用、虚拟机配置的云资源调度算法的国内外研究情况.第 2 节给出基于负载高峰特征的虚拟机放置算法的详细设计.第 3 节给出所提算法的模拟实现与实验结果.第 4 节给出本文的结论以及对进一步的研究加以展望.

## 1 相关研究现状

云资源调度是云计算实现大规模应用和提升性能的关键技术,物理硬件资源毕竟有限,实现虚拟资源的高效灵敏、可动态伸缩的调度将大大地提高云资源的利用率.当前,基于虚拟化技术的云数据中心应用越来越广泛,因此,工业界和学术界在虚拟机调度方面展开了广泛研究.

在虚拟机联合调度的问题上,一般假设云数据中心的物理机数为 $m$,虚拟机数为 $n$,则 $n$ 个虚拟机部署到 $m$ 个物理机的解空间是 $m\times n$,所以通常把虚拟机放置问题看成是装箱问题,这是一个 NP 难的组合优化问题.现阶段,学者们很多采用一些启发式算法进行求解[14−18].文献[14]使用遗传算法,以物理机数目、服务等级协议(service level agreement)、虚拟机迁移次数为优化目标进行求解,但没有考虑物理机的负载均衡.文献[15]对CPU、内存资源约束的放置问题进行建模,考虑虚拟机的配置时间和迁移时间两个因素,实现云资源的分配.文献[16]使用分组遗传算法,以部分虚拟机不相容作为约束条件来解决服务器的整合问题.文献[17]提出了一种基于蚁群算法的服务器负载均衡方法,由于每一只蚂蚁在建立各自的结果集之后仅对信息素局部更新,导致算法收敛速度过慢.文献[18]提出了一种基于遗传算法的负载均衡虚拟机放置算法,该方法结合虚拟机放置时的状态和历史数据,以系统的负载平衡作为优化目标.而在非启发式算法方面,文献[10]提出了一种新型的虚拟机资源估计方法,它定义了有效负载的概念,是指将虚拟机的负载分为固定的需求和动态负载.然后把服务器联合看作随机装箱问题,提出了基于有效负载虚拟机放置算法.文献[11]定义了一种新的衡量两个虚拟机负载峰值相关性的函数,据此进行虚拟机合并来实现节能.算法的有效性通过:(1) 基于网页搜索的多个集群的扩展应用程序负载;(2) 在真实数据中心的利用率数据得到验证.实验结果表明,该方法实现了13.7%的节能和15.6%的QoS服务质量改善.文献[12]提出了一种基于工作负载预测的虚拟机整合算法,预测模型使用自回归预测模型和指数平滑预测模型,并用CloudSim进行仿真模拟,结果显示,算法能有效减少物理服务器的使用数量、虚拟机迁移数和服务等级协议违例.此外,文献[13]提出了一个联合虚拟机资源估计模型,模型将虚拟机的历史负载数据分为趋势部分和随机部分,用时间序列对负载进行预测,然后由负载相关系数矩阵选择负载互补性高的虚拟机两两配对,达到联合配置的效果,但虚拟机联合仅局限于两两配对.

在商业虚拟化产品方面,Amazon 是目前被认为推广基础设施云最为成功的企业之一,代表产品为弹性云计算(elastic compute cloud,简称 EC2).Amazon 将其提供给用户使用的资源分为 8 类,并分别说明其资源配置及费用.用户依据自己的业务需求提出租用申请,在线提交.数据中心调度算法依据用户特征和使用的资源种类、数量和时间等信息找到合适的数据中心资源,将资源信息反馈给用户,交付使用.Amazon 的云资源调度基本上结合了成本优先(采用不同地域成本不同的收费调度分配策略)、满足不同用户需求(预先配置好典型应用虚拟机,如普通、高内存要求、高强度计算等 8 种资源配置)、负载均衡(采用轮询算法)等策略[19,20].

通过相关研究现状分析,可以总结云计算资源调度目前主要存在以下几个问题.

(1) 云计算环境下,数据中心需要对用户应用资源的需求按需分配.由于云数据中心的物理资源具有规模大、异构性强的特点,导致物理机之间负载不均衡,大量的物理资源碎片的产生,资源的利用率低.随着用户请求的持续增长,云数据中心的物理机也不断增加,系统规模日益扩大,物理机中存在的资源碎片将导致云数据中心资源极大的浪费;

(2) 大部分目前提出的虚拟机联合调度算法具有明显的局限性,没有综合考虑云资源的多样性,适用范围小,具有较大的优化改进空间,并且对同一物理机内的负载均衡考虑得较少;

(3) 现有的虚拟机联合调度算法大部分仅考虑单独的虚拟机资源需求,没有联合考虑多个虚拟机的资源互补性;虽然文献[13]利用资源复用进行虚拟机联合调度,从一定程度上改进了虚拟机联合调度效果,但虚拟机联合仅局限于两两配对,没有考虑多个虚拟机之间的资源需求互补;

(4) 目前研究的虚拟机联合调度算法对虚拟机的负载数据特征研究得较少,传统的相关系数等指标在描述资源的差异性上可能还存在不够准确的问题.从发现数据特征的角度切入,通过研究其分布、峰值等规律发现资源的互补性,进而更好地指导虚拟机联合调度.

为此,本文从负载数据特征的角度切入,通过分析数据特征来实现虚拟机联合调度.我们希望将不同虚拟机的负载高峰错开在不同的时间,研究通过发现虚拟机负载高峰的特征,突出负载高峰在虚拟机联合调度中的作用来实现基于负载高峰特征的虚拟机放置算法,从而让多个虚拟机更好地共享物理机资源和提高资源利用率.

## 2 基于负载高峰特征的虚拟机放置算法设计

### 2.1 带权高峰相似度

我们要研究虚拟机负载的高峰特征,那么首先要界定高峰与非高峰.本文设计了两种定义高峰的方法.

**定义 1(固定高峰分界线)**. 在某个时间区间内,给定 $P(0<P<1)$,若 $t_i$ 时刻资源利用率值 $X_i>P$,则点$(t_i,X_i)$称为高峰点,否则称为非高峰点.

特别地,有可能某虚拟机的负载一段时间内一直比较小,均没有超过 $P$ 值,那么这个虚拟机就不存在高峰点;相反,如果某个虚拟机负载一直比较大,$P$ 值又设得比较小,那么这个虚拟机的所有点都是高峰点.

**定义 2(固定高峰比率)**. 在某个时间区间内,资源平均利用率为 $\bar{x}$,给定一比值$\rho(\rho>1)$,若 $t_i$ 时刻资源利用率值 $X_i> \rho\bar{x}$, 则点$(t_i,X_i)$称为高峰点,否则称为非高峰点.

如果$\rho$比较大,那极端情况是 $\rho\bar{x}$ 甚至大于 1,那么也会导致没有高峰点.但由于$\rho>1$,所以非高峰点必然存在.

这里的 $P$ 和$\rho$的大小都是自己定义的,后面的实验会给出不同的 $P$ 和$\rho$值对虚拟机组合的影响.直线 $y=P$, $y=\rho\bar{x}$ 都称为高峰分界线.

区分了高峰点与非高峰点后,我们需要比较两个虚拟机的负载是否相似,特别是它们的高峰是否出现在靠近的时间.高峰出现的时间越重叠,相似度越高.于是,本文引入余弦相似度计算方法,并且为了突出高峰点的特征,我们对高峰点赋予了更大的权重,得到带权高峰相似度(简称高峰相似度)的计算公式:

$$\gamma=\frac{\sum_{i=1}^{n}u_iv_iX_iY_i}{\sqrt{\sum_{i=1}^{n}u_i^2X_i^2\sum_{i=1}^{n}v_i^2Y_i^2}} \tag{1}$$

其中,$X_i,Y_i$ 分别表示 $t=i$ 时虚拟机 $A,B$ 的负载,即 CPU 的利用率;$u_i$ 表示 $X_i$ 的权重;$v_i$ 类似.权重的确定,要根据这

个点是否为高峰点来分别讨论.首先给定一个高峰点的总权重 $Q$,则非高峰点的总权重为$(1-Q)$.对于非高峰点,则平均分配总权重$(1-Q)$,即,若有 $n$ 个非高峰点,则这 $n$ 个非高峰点的权重都是 $\frac{1-Q}{n}$;对于高峰点,则根据超出高峰分界线的多少按比例分配总权重$Q$,比如高峰分界线为 $P$,高峰点 $X_i$ 的权值为 $Q\frac{X_i-P}{\sum_{X_i\in H}(X_i-P)}$,$H$ 为高峰点集.

特殊情况下:

- 当 $P$ 充分大时,$X$ 中所有的点都是非高峰点,则此时非高峰点的权重为 $\frac{1}{n}$.可以理解为,此时的 $Q$=0;
- 当 $P$ 充分小时,$X$ 中所有的点都是高峰点,则此时高峰点的权重为 $\frac{X_i-P}{\sum_{X_i\in H(X)}(X_i-P)}$.可以理解为,此时的 $Q$=1.

记 $y=E$ 为高峰分界线,定义判断是否为非高峰点的辅助函数,则 $\sum_{i=1}^{n}w(X_i,P)$ 表示非高峰点的数目.

$$w(x,E)=\begin{cases}1, & x\leqslant E\\ 0, & x>E\end{cases} \tag{2}$$

则权重公式可以写为

$$u(x)=\begin{cases}Q\dfrac{x-P}{\sum_{X_i\in H}(X_i-P)}\\ (1-Q)\dfrac{1}{\sum_{i=1}^{n}w(X_i,P)}\end{cases} \tag{3}$$

由公式(3)可以计算出每一个点的权重,再代入公式(1),即可计算出两个虚拟机的负载的高峰相似度.

### 2.2 虚拟机联合放置算法

该算法的核心是贪心,每次选择的虚拟机都是与已选择的虚拟机的高峰相似度的和最小的那一个,当一个主机不能再放置虚拟机时,然后考虑下一个主机.虚拟机的状态可以分为未放置、被选中、不相容、已放置,分别对应的虚拟机集为未放置集、临时选择集、不相容集、已放置集.

我们假设所有主机拥有的资源一样,包括 CPU 主频、内存、带宽等资源,且除了 CPU,其他资源是充足的.算法的过程如图 1 所示.

```
1   Input: HostList, vmList, Workload Data   Output:Allocation of VMs
2   calculate the peak similarity between VMs to get the similarity matrix V;
3   put all virtual machines in the unplaced set[1];
4   while there exist VMs unplaced;
5       foreach vm in unplaced set
6           calculate the V_i of each vm in the unplaced set, find the minimum one V_k;
7           remove the VM k from the unplaced set, add it to the temporary selection set[2];
8           if the VMs in the temporary selection set does not meet the SLA[3] then
9               remove the VM k from the temporary selection set, add it to the incompatible set[4];
10      all the VMs in unplaced set has conducted SLA check, then provision the VMs in the temporary selection
          set together in a host physically;
11      put the VMs in the temporary selection set in the placed set[5], empty the temporary selection set;
12      put the VMs in the incompatible set in the unplaced set, empty the incompatible set;
13  return the allocation of VMs
```

Fig.1 Virtual machine placement algorithm based on peak workload characteristics

图 1 基于负载高峰特征的虚拟机放置算法

说明:

- 步骤 2、步骤 3 是算法的初始化阶段;
- 步骤 4~步骤 12 是遍历虚拟机的搜索阶段,循环执行,其中,步骤 5~步骤 12 可以分成两个小阶段:步骤

5~步骤 9 是循环搜索高峰相似度最小的虚拟机并检查 SLA;步骤 10~步骤 12 是标志处理阶段,为下一轮循环重置虚拟机状态标志.

任意虚拟机必存在于已放置集、未放置集、不相容集、临时选择集中的一个.这 4 个集合互不相交,在编程中可以用一个列表存储,用不同的数表示不同的状态,当虚拟机的状态发生改变,修改其值.

$V_i$ 表示虚拟机 $i$ 与临时选择集的虚拟机的高峰相似度系数的和,即 $V_i=\sum_{j\in S}V_{ij}$,$S$ 为临时选择集.

一轮,表示为某一个物理主机挑选虚拟机的过程中,从一个虚拟机没有放置到没有更多虚拟机还能放置的周期.

1. 已放置集(placed set),已经放置在主机上的虚拟机的集合;
2. 未放置集(unplaced set),一轮中,未经过检查该虚拟机是否满足 SLA 的虚拟机集;
3. SLA,这里的 Service-Level Agreement 是指检查 CPU 利用率是否超过 100%;
4. 不相容集(incompatible set),记录一轮中,与临时选择集的其他虚拟机不满足 SLA 检查的虚拟机集;
5. 临时选择集(temporary selection set),表示一轮要检查虚拟机是否满足 SLA 的虚拟机集.其中,若某虚拟机加入后经过 SLA 检查,则确定为即将放置的虚拟机;否则,从临时选择集移到不相容集.

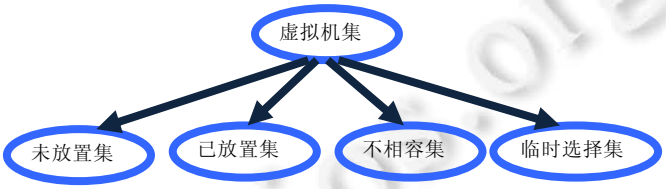


Fig.2 Classification of VM sets

图 2 虚拟机集的分类

$$V=\begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{pmatrix}$$

Fig.3 Similarity matrix of VMs

图 3 虚拟机相似度矩阵

下面考虑算法的复杂度.假设负载向量是 $m$ 维的,有 $n$ 个虚拟机.计算高峰相似度按步骤计算,先计算各点的权重,然后代入高峰相似度公式(1),算法复杂度为 $O(m)$.放置算法中,计算高峰相似度矩阵复杂度为 $O(m^2)$,算法中有两层循环,外层循环 $n$ 次,内层循环 $n,n-1,\ldots,1$ 次,平均循环 $n/2$ 次,每次循环都要检查 SLA,检查临时选择集的虚拟机是否满足 SLA 复杂度为 $O(m)$,故放置复杂度为 $O(m\times n^2)$.

## 3 算法模拟与实验结果

为了验证提出的基于负载高峰特征的虚拟机放置算法的有效性,本文使用 CloudSim 模拟实现了我们所提出的算法,并给出了算法的实验结果.实验包括 4 个部分:(1) 不同高峰分界线对算法效果的影响分析;(2) 不同高峰权重对算法效果的影响分析;(3) 本文设计的算法与随机联合算法、基于相关系数的联合算法的比较分析;(4) 本文设计的算法与随机联合算法、基于相关系数的联合算法的执行速度比较.

### 3.1 算法模拟实现

本文主要通过扩展 *VmAllocationPolicy* 来实现虚拟机放置策略.对于基于负载高峰特征的虚拟机放置算法,首先计算相似度矩阵,然后将所有虚拟机放到未放置集,枚举所有的虚拟机.对于每一个虚拟机,枚举所有未放置集的虚拟机,计算未放置集中所有 $V_i$,找出其中最小的一个 $V_k$,把虚拟机 $k$ 从未放置集中移除,加入临时选择集;若临时选择集中的虚拟机对选择主机不满足 SLA,则将虚拟机 $k$ 从未放置集中移除,加入不相容集;未放置集的虚拟机遍历完后,将临时选择集的虚拟机放置在一个主机中;将临时选择集的虚拟机移至已放置集,清空临时选择集;将不相容集的虚拟机移至未放置集,清空不相容集.如此,就确定了第 1 个物理机中放置的虚拟机,继续枚举所有虚拟机的步骤,得到其他虚拟机的联合结果.基于相关系数的联合算法、随机联合算法实现代码也类似,主要的不同之处在于 *findMinVmSimilarity*()函数,前者变成查找关于相关系数的 $V_i$,后者变成随机选择;对于基于相关系数的联合算法,不同的是相似度矩阵变成了相关系数矩阵,其他同上;对于随机联合算法,不需要计算

一个矩阵,通过简单的随机选择虚拟机进行联合.

下面给出扩展的 *VmAllocationPolicy* 和扩展的 Vm 的简化代码.

```
public class VmAllocationPolicySimilarity extends VmAllocationPolicySimple {
  public boolean allocateHostsForVmList(List<ExtendedVm>vmlist) {
    for (int i=0; i<numOfVms; i++) {
      if (!searchVmforHost(vmlist,vmSimilarity,vmUsedSign,numOfHostUsed,selVmList)) {
        //searchVmforHost()给一个主机寻找放置在其中的虚拟机
        List<ExtendedVm>theVmlist=getHostList().get(numOfHostUsed).getVmList();
        numOfHostUsed++;
        selVmList.clear();
        i--;
        //循环虚拟机,对每一个主机,查找放置在其中的虚拟机,不成功表示已经没有虚拟机能再放得下
          选择主机了
      }
    }

    private boolean searchVmforHost(List<ExtendedVm>vmlist,
      double vmSimilarity[][],int vmUsed[],int hostId,List<ExtendedVm>selVmList) {
    //vmlist 表示所有的虚拟机列表;
    //vmSimilarity 表示虚拟机的相似度矩阵;
    //vmUsed 表示虚拟机的状态,3 种状态:未放置、不相容、已放置.
    //hostId 表示主机 Id.
    //selVmList 表示一轮中,被选择的虚拟机.
      for (int i=0; i<numOfVms; i++) {
        intidx=findMinVmSimilarity(vmSimilarity,vmUsedMark);   //查找最小的 V_i
        if (idx!=-1) {
          selVmList.add(vmlist.get(idx));
          if (allocateAHostForVmList(selVmList)) {
            //检查 selVmList 中的虚拟机 CPU 利用率是否超出 100%
            //若成功,则放置虚拟机并标记,省略了这部分代码.
            return true;
          } else {
            selVmList.remove(selVmList.size()-1);
            //若不成功,则把最后一个放进来的虚拟机移出,标记为不相容,继续搜
            //索其他虚拟机,直至所有虚拟机都搜索一遍,同样省略了部分代码.
          }
        } else
          break;
      }
      return false;
    }
}
```

### 3.2 实验结果

CloudSim[21]是澳大利亚墨尔本大学的网格实验室和 Gridbus 项目宣布推出的云计算仿真软件,它是在离散事件模拟包 Sim-Java 上开发的函数库,继承了 GridSim 的编程模型,采用了成熟的虚拟化技术,将数据中心的资源虚拟化为资源池,支持云计算的资源管理和调度模拟,支持云计算的研究和开发.本文采用 CloudSim 中 PlanetLab 的 CPU 利用率数据[21],数据以 5 分钟为采样单位,24 小时则有 288 个点.为方便研究,实验中的 SLA 是各主机的 CPU 利用率不超过 100%.下面的实验分为 4 个部分:实验 1 研究不同高峰分界线对算法效果的影响;实验 2 研究不同高峰权重对算法的影响;实验 3 比较随机放置算法[22]、基于相关系数的放置算法与设计算法的效果[13],并比较这几种算法的执行时间.第 1 个、第 2 个实验是为第 3 个实验作铺垫,通过实验来分析实验参数对算法效果的影响.然后,我们可以选取比较理想的参数值来使算法进一步优化.

#### 3.2.1 不同高峰分界线对算法效果的影响

下面 3 个图(图 4~图 6)的数据分别是 PlanetLab 的 3 个月的数据,横坐标表示不同的高峰分界线,纵坐标表示经虚拟机联合配置后的主机平均 CPU 利用率,图例表示不同的高峰点总权重.使用算法为以固定高峰分界线分界的基于高峰特征的相似度虚拟机放置算法,简称“固定高峰法”.以固定高峰-平均值比例的基于高峰特征的相似度虚拟机放置算法我们简称为“固定比率法”.因为“固定高峰法”具有更好的配置效果,因此本小节的实验以“固定高峰法”为例.

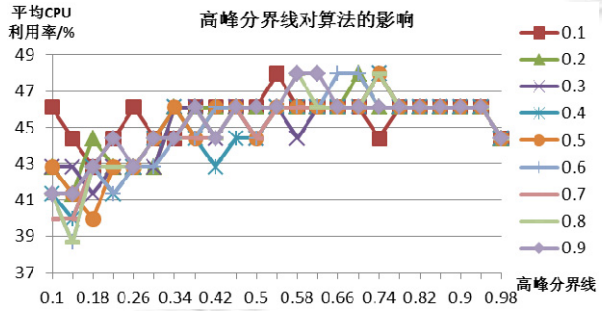


Fig.4 Situation 1 of the algorithm effect under dirrerent peak lines

图 4 高峰分界线对算法的影响情况 1

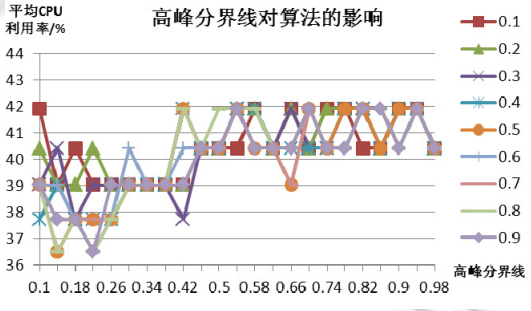


Fig.5 Situation 2 of the algorithm effect under dirrerent peak lines

图 5 高峰分界线对算法的影响情况 2

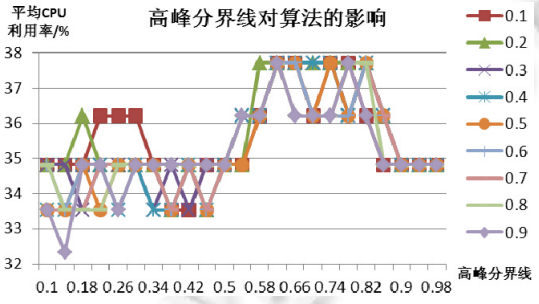


Fig.6 Situation 3 of the algorithm effect under dirrerent peak lines

图 6 高峰分界线对算法的影响情况 3

从实验结果可以分析出:图 4 中的最大值主要出现在[0.54,0.74]区间内,图 5 的最大值则主要出现在[0.5,0.94]区间内,图 6 则集中在[0.58,0.82]区间内.而在图 4 中,在[0,0.34]范围内整体利用率偏低,类似地,图 5 在[0,0.46]范围内也偏低,图 6 中同样在[0,0.5]区间相对偏低.因此在选择高峰分界线的数值时,选择[0.5,0.74]区间的数值,会使配置算法能够获得更好的效果,概率有所提高.

实验结果分析与解释如下.

首先,高峰分界线对算法的效果与负载数据有关系,极端情况下,假如所有的虚拟机都处于某一稳定负载,比如都处于 0.1 的 CPU 利用率上下,那么高峰分界线的选择,大约在(0,0.2]变化时会对配置效果产生影响,而在(0.2,1.0)基本上差异不大.由于虚拟机的负载有多种多样的情况,有的是平衡的低负载,有的是正态曲线般的负载,有的只有某段时间的高峰负载等.正是多种多样的负载情况集合在一起,才使得不同高峰分界线的选择会引起算法效果的波动变化.高峰分界线的确定无法用准确的公式给出,但比较有效的方法是:根据历史负载数据遍历不同的高峰分界线,得出高峰分界线在某区间内可以使配置算法效果更优.

3.2.2 不同高峰权重对算法的影响

我们在考察不同高峰权重对算法的影响时,需要固定高峰分界线.有了前一实验的基础,我们选择比较良好的高峰分界线区间内的数值作为参数.下面 3 个实验结果图(图 7~图 9)的高峰分界线的参数是区间[0.5,0.8]内的数值.横坐标是不同的高峰点总权重,纵坐标是平均 CPU 利用率.

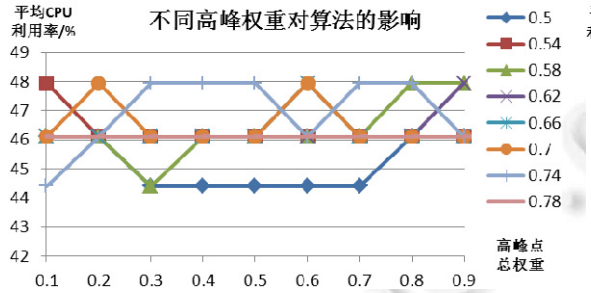


Fig.7 Situation 1 of the algorithm effect under dirrerent peak weights

图 7 不同高峰权重对算法的影响情况 1

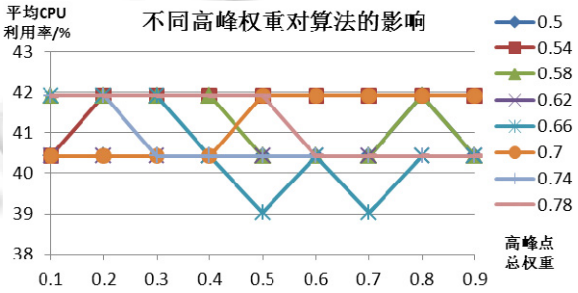


Fig.8 Situation 2 of the algorithm effect under dirrerent peak weights

图 8 不同高峰权重对算法的影响情况 2

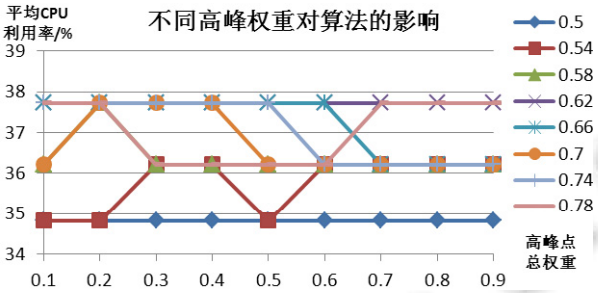


Fig.9 Situation 3 of the algorithm effect under dirrerent peak weights

图 9 不同高峰权重对算法的影响情况 3

由图 7~图 9 的实验结果可以看出:不论高峰点总权重是多少,都至少有一个高峰分界值使得平均 CPU 利用率达到最大值.由此,我们可以得出结论:高峰点总权重的值对本算法的虚拟机联合配置效果影响不大.

实验结果分析与解释:

实验数据显示,当取某一高峰分界线的值使算法达到最优时,大多数情况无论高峰总权重的值是多少,配置效果依然最优.说明算法最主要的是要找出最优的高峰分界线的值,此时,高峰总权重的值倒不是那么重要.

3.2.3 不同算法的比较

本文采用随机放置算法、基于相关系数的放置算法与“固定高峰法”、“固定比值法”比较.比较指标为主机的平均 CPU 利用率和使用主机数(见表 1).其中,Correlation 表示基于相关系数的放置算法,Random(max)表示多次随机配置的最好的结果,Random(ave)表示多次随机的平均结果,FixedBound 表示“固定高峰法”在不同高峰分

界线的最好结果,FixedRatio 表示“固定比值法”在不同的高峰-平均值比值的最好结果.

**Table 1** Experiment parameters

**表 1** 实验参数

| 实验参数 | 取值(单位) |
|---|---|
| 物理节点 | 150(台) |
| 虚拟机数 | 500(个) |
| 高峰点总权重 | 0.4 |

图 10 给出比较不同算法的平均 CPU 利用率,图 11 给出比较不同算法所用的主机数.从图 10 可以看出:最优的是“固定高峰法”,其次是“固定比率法”,“相关系数法”与随机法的最好结果差不多,最差是随机法的平均结果.图 11 的结果也是类似:在需要主机数目上,“固定高峰法”最少,“固定比率法”其次,“相关系数法”与随机法最好结果差不多,随机法的平均结果显示需要最多.

从数值上看:“固定高峰法”与“相关系数法”、随机法的最好结果、随机法的平均结果相比,在利用率上平均有 8.93%,9.52%,12.37%的提高;在主机数上,平均有 8.18%,8.68%,11.01%的提高.而“固定比值法”与这些方法相比,利用率上也平均有 4.53%,5.10%,7.83%的提高和主机数上的 4.31%,4.82%,7.25 的提高.

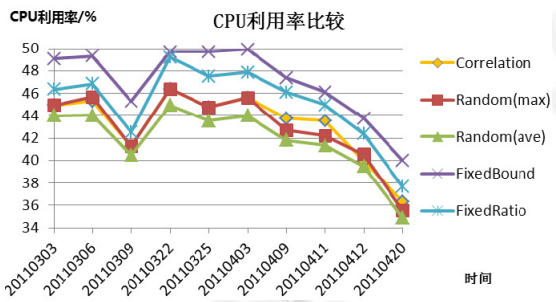


Fig.10 Comparision of CPU utilization

图 10 CPU 利用率比较

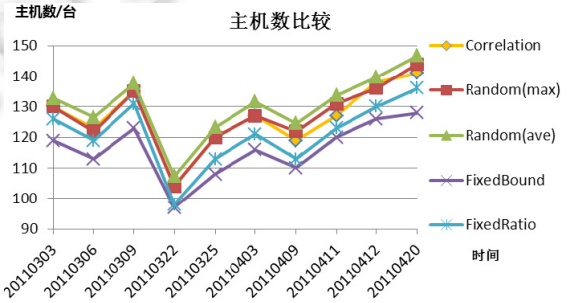


Fig.11 Comparsion of host number

图 11 主机数比较

3.2.4 不同算法的执行时间比较

同样是比较“固定高峰法”、“固定比率法”、“相关系数法”和随机法,4 种方法的算法复杂度都是 $O(mn^2)$,$m$ 是负载向量维数,$n$ 是虚拟机数.负载数据是以 5 分钟为采样单位,一天 24 小时,即 $m$ 这里等于 24×60/5=288.$n$ 选取两个数:100,200 作为样例.每种算法执行 20 次,图 12 所示为平均每次所需运行时间.

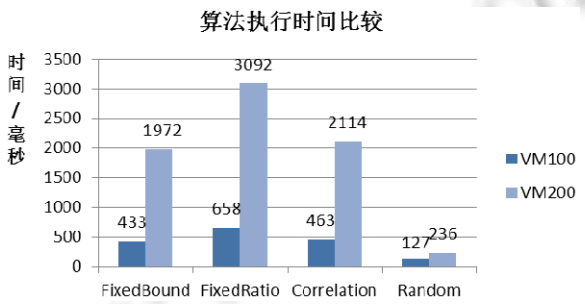


Fig.12 Comparsion of execution time

图 12 执行时间比较

可以看出:随机算法的执行速度最快,因为其计算量少;“固定高峰法”和“相关系数法”差不多;而“固定比率法”则最慢,而且速度差距明显.原因在于:“固定比率法”在计算高峰相似度的权值时需要先遍历负载数据来求

得平均值,带来了更多的计算.所幸的是,该算法主要用于对云计算数据中心进行周期性(非实时的)静态配置,较长的计算时间也是能忍耐的.

## 4 结论与展望

本文提出两种负载高峰的定义,并且使用带权高峰相似度公式来表示不同虚拟机之间的负载高峰的关系;利用虚拟机 CPU 高峰负载特征进行虚拟机联合调度,在平均利用率和主机量上,与随机算法和基于相关系数的选择算法相比,本文提出的基于负载高峰的虚拟机放置算法有明显的提升.更进一步地,“固定高峰法”的表现最好,花销时间上,与“相关系数法”基本一样或者说更少,比“固定比率法”明显要少,却达到了更好的联合效果.总之,实验结果验证了本文提出的带权高峰相似度能够更准确地描述虚拟机负载的互补性,特别是可以描述出资源负载高峰对负载的特征作用.

目前提出的基于负载高峰特征的虚拟机放置算法只考虑了 CPU 的单一资源,制约虚拟机运行的还有内存、带宽等多种资源.不同虚拟机往往对某种资源需求比较大,而对其他资源相对较少.如何利用不同虚拟机对不同类型资源使用上存在的需求差异对虚拟机进行整合,将是下一步研究的重点方向.

**致谢** 在此,我们向对相关项目的进行和论文撰写工作做出贡献和提出宝贵意见的学者,尤其是在华南理工大学计算机科学与工程学院计算机系统研究所(华系/csAsc)工作的老师和同学表示感谢.

**References**:

[1] Foster I, Zhao Y, Raicu I, Lu SY. Cloud computing and grid computing 360-degree compared. In: Proc. of the Grid Computing Environments Workshop (GCE 2008). 2008. 1−6. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4738445&tag=1

[2] Chen K, Zheng WM. Cloud computing: System instances and current research. Ruan Jian Xue Bao/Journal of Software, 2009,20(5): 1337−1348 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3493.htm [doi: 10.3724/SP.J.1001.2009.03493]

[3] Lin WW, Qi DY. Survey of resource scheduling in cloud computing. Computer Science, 2012,39(10):1−6 (in Chinese with English abstract).

[4] Deng W, Liu FM, Jin H, Li D. Leveraging renewable energy in cloud computing datacenters: State of the art and future research. Chinese Journal of Computers, 2013,36(3):82−98 (in Chinese with English abstract).

[5] Ye KJ, Wu ZH, Jiang XH, He QM. Power management of virtualized cloud computing platform. Chinese Journal of Computers, 2012,35(6):1262−1285 (in Chinese with English abstract).

[6] Lin WW, Liang C, Wang JZ, Buyya R. Bandwidth-Aware divisible task scheduling for cloud computing. Software: Practice and Experience, 2014,44(2):163−174. [doi: 10.1002/spe.2163]

[7] Lin WW, Wang JZ, Liang C, Qi DY. A threshold-based dynamic resource allocation scheme for cloud computing. Procedia Engineering, 2011,23:695−703. [doi: 10.1016/j.proeng.2011.11.2568]

[8] Lin WW, Qi DY. Research on resource self-organizing model for cloud computing. In: Proc. of the 2010 Int’l Conf. on Internet Technology and Applications. 2010. 1−5. [doi: 10.1109/ITAPP.2010.5566394]

[9] Lin WW, Liu B, Zhu LC, Qi DY. CSP-Based resource allocation model and algorithms for energy-efficient cloud computing. Journal on Communications, 2013,34(12):33−41 (in Chinese with English abstract).

[10] Chen M, Zhang H, Su YY, Wang XR, Jiang GF, Yoshihira KJ. Effective vm sizing in virtualized data centers. In: Proc. of the IFIP/IEEE Int’l Symp. on Integrated Network Management (IM 2011). IEEE, 2011. 594−601. [doi: 10.1109/INM.2011.5990564]

[11] Kim J, Ruggiero M, Atienza D, Lederberger M. Correlation-Aware virtual machine allocation for energy-efficient datacenters. In: Proc. of the Conf. on Design, Automation and Test in Europe. EDA Consortium, 2013. 1345−1350. [doi: 10.7873/DATE.2013.277]

[12] Wei L, Huang T, Chen JY, Liu YJ. Workload prediction-based algorithm for consolidation of virtual machines. Journal of Electronics & Information Technology, 2013,35(6):1271−1276 (in Chinese with English abstract). [doi: 10.3724/SP.J.1146.2012.01131]

[13] Meng X, Isci C, Kephart J, Zhang L, Bouillet E, Pendarakis D. Efficient resource provisioning in compute clouds via VM multiplexing. In: Proc. of the 7th Int’l Conf. on Autonomic Computing. ACM Press, 2010. 11−20. [doi: 10.1145/1809049.1809052]

[14] Nakada H, Hirofuchi T. Toward virtual machine packing optimization based on genetic algorithm. In: Proc. of the 10th Int'l Work Conf. on Artificial Neural Networks. Part 2: Distributed Computing, Artificial Intelligence Bioinformatics Soft Computing and Ambient Assisted Living. LNCS 5518, Berlin, Heidelberg: Springer-Verlag, 2009. 651−654. [doi: 10.1007/978-3-642-02481-8_96]

[15] Hirofuchi T, Nakada H, Ogawa H, Itoh S, Sekiguchi S. Eliminating datacenter idle power with dynamic and intelligent vm relocation. In: Proc. of the Distributed Computing and Artificial Intelligence. Berlin, Heidelberg: Springer-Verlag, 2010. 645−648. [doi: 10.1007/978-3-642-14883-5_82]

[16] Agrawal S, Bose SK, Sundarrajan S. Grouping genetic algorithm for solving the server consolidation problem with conflicts. In: Proc. of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation. ACM Press, 2009. 1−8. [doi: 10.1145/1543834.1543836]

[17] Gao Y, Guan H, Qi Z, Hou Y, Liu L. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. Journal of Computer and System Sciences, 2013. [doi: 10.1016/j.jcss.2013.02.004]

[18] Hu J, Gu J, Sun G, Zhao T. A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In: Proc. of the 2010 3rd Int'l Symp. on Parallel Architectures, Algorithms and Programming (PAAP). IEEE, 2010. 89−96. [doi: 10.1109/PAAP.2010.65]

[19] Barroso LA, Hölzle U. The datacenter as a computer: An introduction to the design of warehouse-scale machines. Synthesis Lectures on Computer Architecture, 2009,4(1):1−108.

[20] Ostermann S, Iosup A, Yigitbasi N, Prodan R, Fahringer T, Epema D. A performance analysis of EC2 cloud computing services for scientific computing. In: Proc. of the Cloud Computing. Berlin, Heidelberg: Springer-Verlag, 2010. 115−131. [doi: 10.1007/978-3-642-12636-9_9]

[21] Calheiros RN, Ranjan R, Rose CAFD, Buyya R. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Wiley Online Library, 2010.

[22] Zamanifar K, Nasri N. Mohammad-Hossein Nadimi-Shahraki. Data-Aware virtual machine placement and rate allocation in cloud environment. In: Proc. of the 2012 2nd Int'l Conf. on Advanced Computing & Communication Technologies (ACCT). IEEE, 2012. 357−360. [doi: 10.1109/ACCT.2012.40]

**附中文参考文献**:

[2] 陈康,郑纬民.云计算:系统实例与研究现状.软件学报,2009,20(5):1337−1348. http://www.jos.org.cn/1000-9825/3493.htm [doi: 10.3724/SP.J.1001.2009.03493]

[3] 林伟伟,齐德昱.云计算资源调度研究综述.计算机科学,2012,39(10):1−6.

[4] 邓维,刘方明,金海,李丹.云计算数据中心的新能源应用:研究现状与趋势.计算机学报,2013,36(3):582−598.

[5] 叶可江,吴朝晖,姜晓红,何钦铭.虚拟化云计算平台的能耗管理.计算机学报,2012,35(6):1262−1285.

[9] 林伟伟,刘波,朱良昌.齐德昱.基于 CSP 的能耗高效云计算资源调度模型与算法.通信学报,2013,34(12):33−41.

[12] 魏亮,黄韬,陈建亚,刘韵洁.基于工作负载预测的虚拟机整合算法.电子与信息学报,2013,35(6):1271−1276.

**徐思尧**(1991－),男,广东肇庆人,硕士生,主要研究领域为云计算,大数据分析.

**王子骏**(1966－),男,博士,教授,博士生导师,主要研究领域为数据库,多媒体技术,分布式计算.

**林伟伟**(1980－),男,博士,副教授,CCF 会员,主要研究领域为分布式系统,云计算,大数据.