

## 云计算虚拟机部署方案的研究

张笑燕<sup>1</sup>, 王敏讷<sup>1</sup>, 杜晓峰<sup>2</sup>

(1.北京邮电大学 软件学院, 北京 100876; 2.北京邮电大学 计算机学院, 北京 100876)

**摘要:** 提出了一种虚拟机部署方案, 该方案的目的是减少主机上的资源碎片。对不同规格的虚拟机在下一时间段内的增量进行了预测, 根据预测结果对资源池中主机上的可用中央处理器 (CPU) 和内存 (memory) 进行动态规划。该方案考虑了用户的行为习惯, 预先确定了资源的分配规则, 当用户申请虚拟机时, 直接将虚拟机部署到指定的主机上。最后在 CloudStack 平台上对该方案进行了实验验证, 实验结果表明该方案能够有效地减少资源碎片。

**关键词:** 虚拟机; 预测; 动态规划; 分配; 资源碎片

中图分类号: TP301

文献标识码: A

## Research on the method of virtual machine deployment in cloud computing

ZHANG Xiao-yan<sup>1</sup>, WANG Min-ne<sup>1</sup>, DU Xiao-feng<sup>2</sup>

(1. School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. School of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**Abstract:** A method of virtual machine deployment was proposed. The main purpose of the method was to reduce the resource fragments on the host. The method predicted the increment of the virtual machines of different specifications in the next period of time, and dynamically programmed the available CPU and memory of the host in the resources pool by the prediction. The method took into account the users' behaviors, pre-determined the resources allocation rules, when users applied a virtual machine, the virtual machine would be assigned to the specific host directly. In the end, the method was implemented on the CloudStack platform. The experiment demonstrates that the method can reduce the resource fragments effectively.

**Key words:** virtual machine; predict; dynamic program; assign; resource fragments

### 1 引言

云计算是近年来 IT 和互联网领域研究的热点, 它通过虚拟化技术为用户提供计算资源, 包括 CPU、内存、网络、存储等。云计算采用虚拟化技术后, 一项重要的功能需求就是将虚拟机部署在不同的物理机上, 实现对大规模基础资源有效、统一的管理<sup>[1]</sup>。不同的虚拟机部署方案导致了不同的资源分配形式, 造成不同的影响, 研究表明, 已有的很多方案大多是从节能、负载均衡、提高资源利用率等角度来考虑虚拟机的部署, 忽略了因物理主机

本身资源数量的限制而产生的资源碎片, 造成主机上有资源、资源总数足够, 但每台主机上资源数目不足, 虚拟机无法部署的现象。本文基于上述考虑, 提出一种新的虚拟机部署方案, 目的就是合理地规划、分配资源, 使用户请求的不同规格虚拟机都能够分配到主机上, 减少资源碎片, 实现资源的最大化利用。

### 2 相关工作

云计算环境下, 云虚拟机部署时需要从众多主机中寻找一个合适的主机来部署, 随着云计算技术

收稿日期: 2014-12-16; 修回日期: 2015-03-05

基金项目: 国家自然科学基金资助项目(61202081)

**Foundation Item:** The National Natural Science Foundation of China (61202081)

的日益发展,为了满足用户的不同需求,例如从负载均衡、能耗等角度来考虑虚拟机的部署。如今国内外学者提出了多种不同的部署方案,用户可以根据自己的需求来设定虚拟机部署方案。当用户向管理节点提交资源申请时,管理节点根据预先设定好的部署方案来选择主机。目前,云计算环境下常用的几种虚拟机部署方案有:轮询法<sup>[2]</sup>、加权轮询法<sup>[2]</sup>、贪心法<sup>[3]</sup>、最小链接法<sup>[2]</sup>等。轮询法指以轮转的方式依次将虚拟机部署到不同的主机上;该方法的优点是它不需要记录当前所有主机的状态,简洁性高,缺点是没有考虑主机的性能<sup>[2]</sup>。加权轮询法是对轮询法的改进,为了克服轮询算法的不足,用权值表示主机的性能,权值较大的主机将被部署更多的虚拟机;该方法的优点是考虑了主机的性能、利用率等因素,但是当服务请求变化较大时,容易导致负载不均衡<sup>[2]</sup>。贪心法指根据需要选取一种度量标准,将当前主机按这种度量标准排序,选择一个最优主机,例如,尽可能地将虚拟机部署到虚拟机数目最少的主机上、将同一网络内的虚拟机部署到同一台主机上等。最小链接法也是贪心法的一种,利用负载均衡器记录各个真实服务器的连接数,当用户请求到达时,负载均衡器将该连接请求分配到当前连接数最小的真实服务器上<sup>[2]</sup>;该方法考虑了负载均衡,但是没有考虑主机性能差距较大的情况。另外,文献[4]和文献[5]综合用户需求和负载分别提出2种虚拟机部署方案,文献[4]综合考虑用户的要求和物理资源以及服务器的动态负载能力,提出一种虚拟机部署算法 IVDA,实现了服务器动态负载的自适应调度机制;文献[5]通过计算各主机的目标函数值,选择最优的主机作为目标主机来部署虚拟机,实现了基于用户个性化需求的虚拟机自动部署,并在一定程度上减少了虚拟机频繁迁移的现象,但是均没有考虑资源利用率的问题。文献[6~8]从降低能耗的角度考虑虚拟机的部署,文献[6]提出的虚拟机部署方案在减少能耗的同时,通过最小化最大链路利用率来改善网络性能,实现了流量的均衡分布,减少了网络拥堵;文献[7]实现了 SLA(服务等级协议)和能耗之间的平衡;文献[8]提出的资源管理策略通过虚拟机的动态重新分配来减少能耗,但是没有考虑虚拟机迁移的代价。文献[9]提出了一种快速启发式虚拟机部署方案,减少了物理主机数目,优点是考虑了虚拟机迁移的代价。文献[10]提出使用冗余配置的虚拟机部署方案,优点在于考

虑了容错机制,将任意  $k$  个主机服务器失败所造成的影响最小化。文献[11]从实现资源的最佳利用角度提出了一个随机模型来优化数据中心整合,并提出了一种智能策略对物理服务器进行复用,优点是在有效提高资源利用率的同时提供一定的服务保障。文献[12]采用典型相关性分析法,提出了一种性能预测模型,在虚拟机部署同时有效地改善了性能问题。

通过对现有的虚拟机部署方案的研究发现,已有的很多方案大多是从减少服务器数量的角度来考虑节能问题或者考虑负载均衡问题以及资源利用率问题,但是都没有考虑虚拟机部署后对主机上物理资源(CPU、内存)的影响。本文从减少资源碎片的角度考虑,结合用户的行为习惯,提出一种新的云环境下虚拟机部署方案。

### 3 研究问题的描述

本文的研究内容是提出一种虚拟机部署方案,通过合理地部署虚拟机来减少主机上的资源碎片,当用户申请虚拟机时,避免在资源池内资源足够时因每台主机上资源不足而对已部署的虚拟机进行迁移。合理地部署虚拟机,减少资源碎片需要预先对主机上的可用资源进行规划,即需要预先设置分配规则,来确定主机的资源分配情况。

分配规则记录每台主机可部署不同规格的虚拟机数量和已部署的该规格虚拟机数量,预先设定好什么规格的虚拟机应当部署到哪台主机上,当用户申请虚拟机时,查找分配规则,将虚拟机部署到已设定好的主机上。

本文将 CPU 和内存作为主机资源碎片的研究对象,因此,定义如下参数。

主机集合  $HOST = \{host_1, \dots, host_k, \dots, host_K\}$ ,  $hcpu_k$  为第  $k$  台主机上 CPU 的剩余量,  $hmemory_k$  为第  $k$  台主机上内存的剩余量;虚拟机集合  $VM = \{vm_1, \dots, vm_i, \dots, vm_N\}$ , 其中  $vm_i(j, vcpu_i, vmemory_i)$ ,  $j$  为第  $i$  台虚拟机的规格,  $vcpu_i$  为第  $i$  台虚拟机的 CPU 数目,  $vmemory_i$  为第  $i$  台虚拟机的内存大小。

分配规则集合  $R-HOST = \{rhost_1, \dots, rhost_k, \dots, rhost_K\}$ , 其中  $rhost_k(rhost_k^1, \dots, rhost_k^j, \dots, rhost_k^j)$ ,  $rhost_k^j(num_k^j, rnum_k^j)$  为第  $k$  台主机上第  $j$  中规格的虚拟机的部署规则,  $num_k^j$  为第  $k$  台主机上最终可放置的第  $j$  种规格的虚拟机数目,  $rnum_k^j$  为第  $k$  台主机上已放置的第  $j$  种规格的虚拟机数目。

虚拟机部署结束后分别计算 CPU 碎片率  $\eta_{cpu}$  和内存碎片率  $\eta_m$ , 则问题转化为如何确定分配规则, 以降低  $\eta_{cpu}$  和  $\eta_m$ 。

#### 4 虚拟机部署方案的设计思路

在云环境下, 根据历史数据和用户的使用习惯, 用户在短期内申请的虚拟机数量不是随机的, 某一段时间内的虚拟机增量可以看作是最近时段历史值的延续。因此, 可以根据短期内虚拟机数目增量的历史值来预测下一时间段内虚拟机的增量。

为了达到减小资源碎片的目的, 需要根据预测值对资源池内的资源进行动态规划, 在满足预留阈值的情况下尽可能地利用每台主机上的物理资源。当用户申请虚拟机时, 即可直接将虚拟机部署到指定的主机上。

然而, 实际情况中, 受各种因素的影响, 用户在该时间段内申请虚拟机数目的增量可能与预测值不等, 当用户申请虚拟机时, 若该规格虚拟机数量已达上限, 原有的分配规则便不能继续使用, 需要重新预测, 重新对资源池进行规划。因此, 本文将部署方案分为3个模块: 预测模块、规划模块与调整模块, 其中预测模块为规划模块提供数据, 调整模块确定重新进行资源规划的触发条件。

##### 4.1 预测模块

本文将虚拟机的规格定义为虚拟机所需要的 CPU 数量和内存大小。在现有的云服务环境中, 虚拟机规格的种类一般由云服务提供商设定, 然后供用户选择。如在某运营商提供的云环境中, 用户可订购的虚拟机规格是有限的; CloudStack 云平台中, 用户可选择的虚拟机规格种类也是由开发人员预先设定好。因此, 本文设定用户可订购的虚拟机规格的种类是有限的。每个时间段内的虚拟机数量的增量变化有一定规律, 为了避免用户操作失误, 如大量订购某规格的虚拟机而后又大量删除, 导致虚拟机数目突然增加接着减少的情况, 本文将一个时间段  $t$  内不同规格虚拟机数目的增量作为预测目标。

随着云计算技术的快速发展, 用户对虚拟机的需求会越来越大, 从长期来看不同规格虚拟机增量的时间序列曲线呈增长趋势, 从短期来看会有略微波动, 它是一种总体呈增长趋势的线性时间序列。对此种序列, 本文采用二次指数平滑法<sup>[13]</sup>来预测序列的下一个值。

二次指数平滑法是对一次指数平滑再做一次指数平滑, 求出一次指数平滑值和二次指数平滑值的差值, 然后将差值加到一次指数平滑值上, 在加上趋势变动值便得出近似于实际值的预测值。

预测算法如下。

1) 计算每种规格的虚拟机数目在  $I$  个  $t$  时间段内的增量, 从数据库中得到虚拟机规格集合为  $J=\{1, \dots, j, \dots, J\}$ , 共有  $J$  种不同规格的虚拟机。则所有规格虚拟机增量的集合可表示为  $MU=\{M_1, \dots, M_j, \dots, M_J\}$ , 其中  $M_j(\Delta m_1^j, \dots, \Delta m_i^j, \dots, \Delta m_I^j)$  表示第  $j$  种规格的虚拟机在  $I$  个  $t$  时间段内的增量序列,  $\Delta m_i^j$  表示第  $j$  种规格的虚拟机在第  $i$  个  $t$  时间段内的增量, 由式(1)得到,  $Y_i^j$  为第  $i$  个  $t$  时间段内用户申请的规格为  $j$  的虚拟机数目。

$$\Delta m_i^j = \begin{cases} Y_i^j - Y_{i-1}^j, & \Delta m_i^j > 0 \\ 0, & \Delta m_i^j \leq 0 \end{cases} \quad (1)$$

2) 对  $M_j$  通过式(2)和式(3)分别得到一次平滑值  $S_i^{j(1)}$  和二次平滑值  $S_i^{j(2)}$ ;  $a^j$  和  $b^j$  为参数, 由式(4)、式(5)得到;  $\alpha$  为平滑常数, 取值范围为  $[0, 1]$ , 通过反复试算来确定  $\alpha$  的最优值。

$$S_i^{j(1)} = \alpha \Delta m_i^j + (1 - \alpha) S_{i-1}^{j(1)} \quad (2)$$

$$S_i^{j(2)} = \alpha S_i^{j(1)} + (1 - \alpha) S_{i-1}^{j(2)} \quad (3)$$

$$a_i^j = 2S_i^{j(1)} - S_{i-1}^{j(2)} \quad (4)$$

$$b_i^j = \frac{\alpha}{1 - \alpha} (S_i^{j(1)} - S_{i-1}^{j(2)}) \quad (5)$$

3) 对第  $I+1$  期做预测, 在  $a_i^j$  的基础上加上一个变动趋势值  $b_i^j$ , 得到规格为  $j$  的虚拟机在第  $I+1$  个  $t$  时间段内的预测增量  $\Delta m_{I+1}^j$ 。

$$\Delta m_{I+1}^j = a_i^j + b_i^j \quad (6)$$

4) 重复步骤 2), 得到预测结果集合  $MN=\{\Delta m_{I+1}^1, \dots, \Delta m_{I+1}^j, \dots, \Delta m_{I+1}^J\}$ 。

5) 根据预测结果, 计算预测到的虚拟机总数  $N$ 。

$$N = \sum_{j \in [1, J]} \Delta m_{I+1}^j \quad (7)$$

##### 4.2 规划模块

规划模块根据预测模块的预测结果, 即下一个  $t$  时间段内不同规格虚拟机的增量, 分别对资源池中每台主机上的剩余资源 (CPU、内存) 进行动态规划。为了简化模型, 本文将规划算法分为3部分: 对主机进行排序、对虚拟机进行排序、分配虚拟机。

本文提出的部署方案的目的是为了减小资源碎片，避免不必要的迁移，而产生碎片和迁移的情况一般都发生在主机已被占用的资源即将达到最大阈值的时候，为此，先对剩余可用资源较多的主机进行动态规划；同时，为了保证虚拟机分配的灵活性，先将资源分配给规格较大的虚拟机。因此，规划模块的算法设计应当遵循 2 个规则：剩余可用资源较多的主机先规划，规格较大的虚拟机先被分配资源。

规划算法详细步骤如下。

1) 计算资源池中可用的资源数量，设主机数量为  $K$ ，统计每台主机上剩余可用的 CPU 数量、内存大小，以及每台主机上已有的不同规格的虚拟机数量，得到主机集合  $HOST = \{host_1, \dots, host_k, \dots, host_K\}$ ，其中  $host_k(hcpu_k, hmemory_k, rnum_k^1, \dots, rnum_k^j, \dots, rnum_k^J)$ 。按剩余可用的 CPU 数量从多到少对主机进行排序，并对主机设置优先级，CPU 数量越多的优先级越高，得到的主机序列满足

$$HOST = \{host_k(hcpu_k, hmemory_k, num_k^1, \dots, num_k^j, \dots, num_k^J) | hcpu_k \geq hcpu_{k+1}, k \in [1, K], j \in [1, J]\} \quad (8)$$

则资源池中总资源数  $CCPU$  和  $CMEMORY$  分别为

$$\begin{cases} CCPU = \sum_{k \in [1, K]} hcpu_k \\ CMEMORY = \sum_{k \in [1, K]} hmemory_k \end{cases} \quad (9)$$

2) 对预测模块得到的不同规格的虚拟机按 CPU 数目从少到多进行排序，得到虚拟机列表  $VM = \{vm_1, \dots, vm_i, \dots, vm_N\}$ 。虚拟机规格列表满足

$$VM = \{vm_i(j, vcpu_i, vmemory_i) | vcpu_i \leq vcpu_{i+1}, j \in [1, J], i \in [1, N]\} \quad (10)$$

计算预测到的虚拟机所需资源总量  $VCPU$  和  $VMEMORY$

$$\begin{cases} VCPU = \sum_{i \in [1, N]} vcpu_i \\ VMEMORY = \sum_{i \in [1, N]} vmemory_i \end{cases} \quad (11)$$

3) 分别比较  $CCPU$  和  $VCPU$ 、 $CMEMORY$  和  $VMEMORY$  的大小，判断资源池内资源数量是否满足，若不满足，提醒管理员添加资源配置。使  $CCPU$  和  $VCPU$ 、 $CMEMORY$  和  $VMEMORY$  满足

$$\begin{cases} VCPU \leq CCPU \\ VMEMORY \leq CMEMORY \end{cases} \quad (12)$$

4) 按主机的排列顺序分别对每台主机的可用资源进行动态规划。背包算法是动态规划算法的一种，本文在 0-1 背包算法<sup>[14]</sup>的基础上设计每台主机的资源规划算法。将每台主机看成一个背包，每台虚拟机看成一个物品，为物品设定 2 个价值，价值的大小分别等于每台虚拟机所需的 CPU 数量和内存大小，这样当背包容量达到最大时，背包内物品的 2 类价值也最大，2 类价值分别设为  $fcpu$  和  $fm$ 。对每台主机用一次规划算法，每台虚拟机从小到大遍历当前主机的 CPU 总数和内存总数，每次比较是否应该放入该虚拟机，并记录遍历过程中该主机不同容量时对应的最大价值。当最后一台虚拟机遍历完成，得到的是最大价值。根据之前的记录找出分配到该主机的虚拟机，将这些虚拟机从虚拟机列表中删除，得到新的虚拟机列表重新遍历下一台主机。对第  $k$  台主机分别设置 CPU 和内存的状态转移方程

$$fcpu[i]_k = \max \{fcpu[i-1][cv][mv], fcpu[i-1][cv-vcpu_i][mv-vmemory_i]+vcpu_i\} \quad (13)$$

$$fm[i]_k = \max \{fm[i-1][cv][mv], fm[i-1][cv-vcpu_i][mv-vmemory_i]+vmemory_i\} \quad (14)$$

参数解释如下。

$i$  取  $1 \sim N$ ， $N$  为预测到的虚拟机的总数；

$fcpu[i]$  和  $fm[i]$  分别表示从前  $i$  台虚拟机中取虚拟机部署到主机上， $fcpu$  和  $fm$  能达到的最大值；

$cv$  和  $mv$  分别表示遍历过程中主机的 CPU 容量和内存容量， $cv \in [0, hcpu_k]$ ， $mv \in [0, hmemory_k]$ ；

$fcpu[i-1][cv][mv]$  表示当主机的 CPU 容量和内存容量分别为  $cv$  和  $mv$  时，不放入第  $i$  台虚拟机，主机的  $fcpu$  价值；

$fcpu[i-1][cv-vcpu_i][mv-vmemory_i]+vcpu_i$  表示当主机的 CPU 容量和内存容量分别为  $cv$  和  $mv$  时，放入第  $i$  台虚拟机，主机的  $fcpu$  价值。

式(13)表示取不放入第  $i$  台虚拟机和放入第  $i$  台虚拟机后两者  $fcpu$  中的最大值；

$fm[i]$  同上；

实际应用中，选择当放入第  $i$  台虚拟机后主机的 2 类价值均大于不放入第  $i$  台虚拟机时的价值时，即同时满足式(13)和式(14)则放入第  $i$  台虚拟机。

5) 反向查找出每台主机上应放置的虚拟机。对第  $k$  台主机，若第  $i$  台虚拟机同时满足式(15)和式(16)，则记录第  $i$  台虚拟机的规格，第  $k$  台主机可放

置的相应规格虚拟机数量  $num_k^j$  加 1。

$$fcpu[i-1][hcpu_k - vcpu_i][hmemory_k - vmemory_i] + vcpu_i > fcpu[i-1][hcpu_k][hmemory_k] \quad (15)$$

$$fm[i-1][hcpu_k - vcpu_i][hmemory_k - vmemory_i] + vmemory_i > fm[i-1][hcpu_k][hmemory_k] \quad (16)$$

$i$  从  $N-1$  反向取值，对每台主机采用上述分配算法，记录每台主机可容纳的每种规格虚拟机的数目，得到集合  $R-HOST=\{rhost_1, \dots, rhost_k, \dots, rhost_K\}$ 。

### 4.3 策略调整模块

实际环境中某时间段内虚拟机数目的增量和预测到的数据不可能完全相等，当某种规格的虚拟机增量发生变化时，需要重新对资源池中的可用资源进行规划，以满足用户的需求。考虑以下 2 种情况。

第一种情况，当记录的资源池中某种规格的虚拟机数目达到上限时，按照之前的分配规则，分配给该规格虚拟机的资源即将被全部占用，当用户继续申请虚拟机时，会因资源不足，导致虚拟机无法迅速部署。针对这种情况，当用户申请规格为  $j$  的虚拟机时，查找每台主机的  $rnum_k^j$  和  $num_k^j$  值，当满足式(17)时，进入预测模块和规划模块，预测虚拟机增量，并对资源池的资源进行规划，得到新的分配规则，后台管理节点按照新的分配规则部署用户申请的虚拟机。这一过程中，已部署的虚拟机不需要迁移，将作为历史数据供预测模块使用。

$$\sum_{k \in [1, K]} rnum_k^j = \sum_{k \in [1, K]} num_k^j \quad (17)$$

第二种情况，当方案初次使用时，在没有规划数据的情况下，需要应用规划策略建立初始的分配方案。

## 5 应用场景

根据上述模型设计过程，当用户申请虚拟机时，首先判断分配规则是否为空，若为空则进入预测模块。若不为空则判断该规格虚拟机的数目是否已达到上限，如果已达到上限，则触发调整模块，更新虚拟机分配规则，然后按主机优先级从高到低的顺序，根据更新后的分配规则将虚拟机部署到相应的主机上；若未达上限，则直接根据分配规则按主机优先级从高到低部署虚拟机。完成分配后，记录该虚拟机信息，作为下一轮预测的历史数据。

应用流程图如图 1 所示。

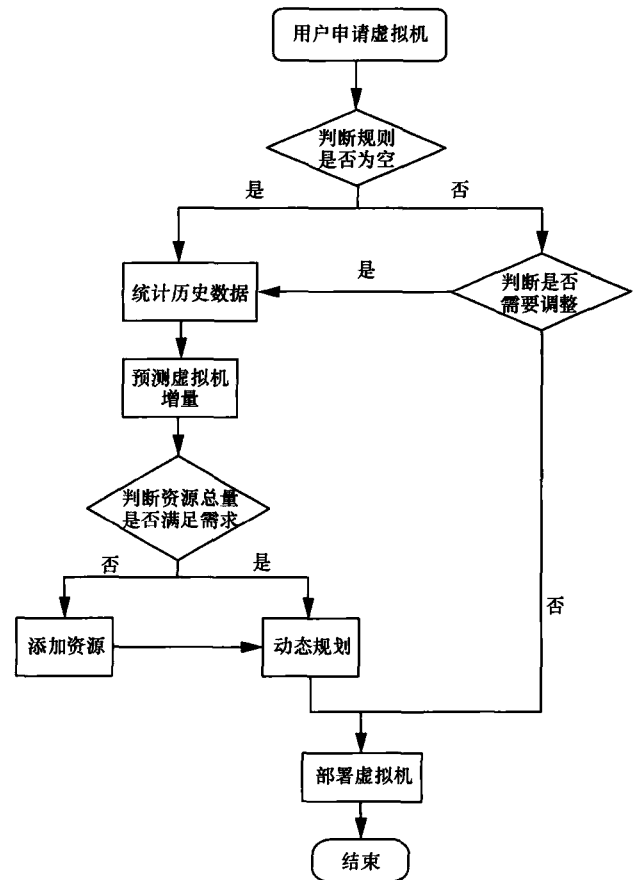


图 1 应用流程

## 6 测试

本文提出的部署方案在 CloudStack 平台上实现，在 CloudStack 云平台的虚拟机管理器中，资源发现通过 VirtualMachineManage 模块实现，本文提出的部署方案将在 VirtualMachineManageImpl 类中实现。除了本文提出的虚拟机部署方案外，本文还在 CloudStack 云平台上部署了另外 2 种虚拟机部署方案，与本文提出的部署方案做对比。

云计算环境中，用户对资源的需求是不同的，当用户需要大量虚拟机，而每台虚拟机的负荷不重时，用户希望将虚拟机启动在同一台主机上，关闭不必要的主机，减少资源浪费；而当虚拟机负荷较重时，用户可能希望将虚拟机均衡地分散到不同的主机上，尽可能地使每台主机都能有充足的资源<sup>[15]</sup>。因此，本文采用的 2 种对比方案如下。

方案 1 即为轮询法，虚拟机按主机顺序循环部署。

方案 2 即按主机顺序总是将虚拟机部署到虚拟机数目最多的主机上。

实验以当前时间作为实验开始，以虚拟机无法部署、需要后台管理员添加资源或需要进行虚拟机迁移才能部署时为实验结束。

实验结束后根据部署后的主机剩余资源量  $L_{CPU}$  和  $L_{MEMORY}$ ，计算 CPU 碎片率  $\eta_{cpu}$  和内存碎片率  $\eta_m$ 。

$$\begin{cases} \eta_{cpu} = \frac{L_{CPU}}{CCPU} \\ \eta_m = \frac{L_{MEMORY}}{CMEMORY} \end{cases} \quad (18)$$

虚拟机的部署过程是后台资源管理服务器寻找合适主机的过程，前台界面无法看到执行的结果。将从后台日志中查看返回的信息，以及从数据库中查看虚拟机的部署结果得出测试结果。虚拟机部署方案验证环境中，设定  $\alpha=0.9$ ，时间段  $t$  为 1 个月。

测试环境中有 5 台主机，去除预留资源，可用资源按顺序分别为：5 核 10G、6 核 10G、8 核 10G、7 核 10G、9 核 12G。

虚拟机的规格有 4 种：1 核 1G、1 核 2G、2 核 2G、2 核 4G。预测模块中，每个月每种规格虚拟机取出 9 组数据，根据日志和数据库查询得到第 10 个月实际增量和预测结果如表 1 所示。

表 1 预测结果

规格	前 10 个月的增量									10 月预测值	
1 核 1G	0	0	0	1	1	1	2	2	3	4	4
1 核 2G	0	1	0	0	1	1	2	3	3	4	3
2 核 2G	1	2	2	3	2	4	4	4	5	6	6
2 核 4G	1	2	3	3	3	4	4	5	6	7	7

从主机剩余资源的总数来看， $CCPU=35$  核， $CMEMORY=52G$ ，第 10 个月虚拟机增加的总数为  $V_{CPU}=34$  核， $V_{MEMORY}=52G$ ，满足式(12)增加的虚拟机应当能够全部部署。

实验结束后统计用户申请虚拟机的顺序为：2 核 2G、2 核 4G、2 核 4G、2 核 4G、2 核 2G、2 核 2G、1 核 1G、2 核 4G、2 核 4G、2 核 2G、1 核 1G、1 核 2G、1 核 2G、1 核 2G、2 核 2G、1 核 1G、2 核 2G、2 核 4G、2 核 4G、1 核 1G、1 核 2G，共 21 台。

下面分别给出按方案 1、方案 2 和本文提出的虚拟机部署方案的部署结果。

方案 1 部署结果如表 2 所示。

表 2 方案 1 部署结果

主机	主机上部署的虚拟机	已部署的虚拟机数目	主机上的剩余资源
5 核 10G	2 核 2G	3 台	0 核 5G
	2 核 2G		
	1 核 1G		
6 核 10G	2 核 4G	4 台	1 核 2G
	1 核 1G		
	1 核 2G		
	1 核 1G		
9 核 10G	2 核 4G	3 台	4 核 0G
	2 核 4G		
	1 核 2G		
8 核 10G	2 核 4G	3 台	3 核 0G
	2 核 4G		
	1 核 2G		
7 核 12G	2 核 2G	3 台	1 核 6G
	2 核 2G		
	2 核 2G		

如表 2 所示，共部署了 16 台虚拟机，当第 17 台虚拟机到来时已无资源可分配，需要管理员添加资源或迁移已部署的虚拟机，实验结束。因此剩余 1 台 2 核 2G 的虚拟机、2 台 2 核 4G 的虚拟机、1 台 1 核 1G 的虚拟机、1 台 1 核 2G 的虚拟机无法部署，主机剩余资源总和满足这五台虚拟机，但是每台主机都不满足需求。碎片率为  $\eta_{cpu}=25.71\%$ ， $\eta_m=25.00\%$ 。

方案 2 部署结果如表 3 所示。

表 3 方案 2 部署结果

主机	主机上部署的虚拟机	已部署的虚拟机数目	主机上的剩余资源
5 核 10G	2 核 2G	3 台	0 核 3G
	2 核 4G		
	1 核 1G		
6 核 10G	2 核 4G	3 台	0 核 0G
	2 核 4G		
	2 核 2G		
9 核 10G	2 核 2G	3 台	3 核 0G
	2 核 4G		
	2 核 4G		
8 核 10G	2 核 2G	5 台	2 核 1G
	1 核 1G		
	1 核 2G		
	1 核 2G		
	1 核 2G		
7 核 12G	2 核 2G	4 台	0 核 3G
	1 核 1G		
	2 核 2G		
	2 核 4G		

如表 3 所示，共部署了 18 台虚拟机，当第 19

台虚拟机到来时已无资源可分配，需要管理员添加资源或迁移已部署的虚拟机，实验结束。因此剩余 1 台 2 核 4G 的虚拟机、1 台 1 核 1G 的虚拟机、1 台 1 核 2G 的虚拟机无法部署，主机剩余资源总和满足这 3 台虚拟机，但是每台主机都不满足需求。碎片率为  $\eta_{cpu}=14.29\%$ ， $\eta_m=13.46\%$ 。

按本文提出的部署方案的部署结果如表 4 所示。

表 4 本文提出的部署方案部署结果

主机	主机上部署的虚拟机	已部署的虚拟机数目	主机上的剩余资源
5 核 10G	2 核 4G 2 核 4G	2 台	1 核 2G
6 核 10G	2 核 4G 2 核 4G 1 核 2G	3 台	1 核 0G
7 核 10G	2 核 4G 1 核 2G 1 核 1G 1 核 1G 1 核 1G 1 核 1G	6 台	无剩余
8 核 10G	2 核 4G 2 核 2G 2 核 2G 2 核 2G	4 台	无剩余
9 核 12G	2 核 4G 2 核 2G 2 核 2G 2 核 2G 1 核 2G	5 台	无剩余

如表 4 所示，部署了 20 台虚拟机，主机剩余资源总和为 2 核 2G，当第 21 台虚拟机到来时触发调整机制，重新预测虚拟机数目，重新对资源池进行规划和分配，需要管理员增加资源，实验结束。碎片率为： $\eta_{cpu}=5.71\%$ ， $\eta_m=3.8\%$ 。

按 3 种方法部署虚拟机的资源碎片率如表 5 所示。

表 5 3 种方案碎片率比较

部署方法	CPU 碎片率	内存碎片率
方案 1	25.71%	25.00%
方案 2	14.29%	13.46%
本文部署法	5.17%	3.8%

从测试结果可以看出，从减少资源碎片的角度考虑，本文提出的虚拟机部署方案优于另外 2 种方法。

## 7 结束语

本文提出了一种考虑主机资源碎片的虚拟机部

署方案。该方案以预测虚拟机在下一时间段内的增量为基础，以减少资源碎片为目的，对资源池内的可用资源进行合理地规划，将虚拟机部署到合适的主机上。新提出的方案的优点在于减少了主机上的资源碎片，保证了资源的最大化利用，缺点是并没有考虑主机的性能问题。下一步将对该问题进行研究，在考虑主机性能方面做进一步改进，以提高方案的适用性。

## 参考文献：

- [1] 钱琼芬, 李春林, 张小庆. 云数据中心资源管理研究综述[J]. 计算机应用研究, 2012, 29(7):2411-2415.  
QIAN Q F, LI C L, ZHANG X Q. Survey of virtual resource management in cloud data center[J]. Application Research of Computer, 2012, 29(7): 2411-2415.
- [2] 田文洪, 赵勇. 云计算资源调度管理[M]. 北京: 国防工业出版社, 2011.  
TIAN W H, ZHAO Y. Resource Scheduling Management of Cloud Computing[M]. Beijing: National Defense Industry Press, 2011.
- [3] Open stack compute administration manual[EB/OL]. <http://docs.openstack.org>.
- [4] HUANG F, LI D S. Efficient virtual machine deployment in large scale resource environment[A]. IEEE International Conference on Parallel and Distributed Systems(ICPADS'2010)[C]. Shanghai, China, 2010.752-757.
- [5] 彭红, 杨根兴, 蔡立志. 基于用户个性化需求的虚拟机部署机制[J]. 软件产业与工程, 2013, (1):37-41.  
PENG H, YANG G X, CAI L Z. Virtual machine deployment based on the needs of individual users[J]. Software Industry and Engineering, 2013, (1):37-41.
- [6] 董健康, 王洪波, 李阳阳. IaaS 环境下改进能源效率和网络性能的虚拟机放置方法[J]. 通信学报, 2014, 35(1):72-81.  
DONG J K, WANG H B, LI Y Y. Improving energy efficiency and network performance in IaaS cloud with virtual machine placement[J]. Journal on Communications, 2014, 35(1): 72-81.
- [7] KORD N, HAGHIGHI H. An energy-efficient approach for virtual machine placement in cloud based datacenters[A]. Proceeding of the 5th Conference on Information and Knowledge Technology[C]. Shiraz, 2013. 44-49.
- [8] BELOGLAZOV A, BUYYYA R. Energy efficient allocation of virtual machines in cloud datacenters[A]. Proceeding of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)[C]. Melbourne, 2010.577-578.
- [9] OHTA S. Virtual machine placement algorithms to minimize physical machine count[A]. Proceeding of the 15th Asia-Pacific Network Operations and Management Symposium[C]. Japan, 2013.1-3.
- [10] MACHIDA F, KAWATO M, MAENO Y. Redundant virtual machine placement for fault-tolerant consolidated server clusters[A]. Proceed-



ing of the 2010 IEEE Network Operations and Management Symposium[C]. Osaka, 2010.32-39.

[11] NANDI B B, BANERJEE A, GHOSH S C, *et al.* Stochastic VM multiplexing for datacenter consolidation[A]. Proceeding of the 9th International Conference on Services Computing[C]. Honolulu, 2012. 114-121.

[12] ANH V D, CHEN J L, WANG C. Profiling applications for virtual machine placement in clouds[A]. Proceeding of the 2011 IEEE International Conference on Cloud Computing(CLOUD'2011)[C]. Washington, USA, 2011.660-667.

[13] CHEN F Y, LI Q, LIU J H. Variable smoothing parameter of the double exponential smoothing forecasting model and its application[A]. IEEE International Conference on Advanced Mechatronic Systems(ICAMEchs)[C]. Tokyo, 2012. 386-388.

[14] 李军民, 傅云凤. 求解 0-1 背包问题的动态规划改进算法分析[J]. 西北大学学报, 2014, 44(5):729-732.

LI J M, FU Y F. Analysis of improved algorithm of dynamic programming solving 0-1 knapsack problem[J]. Journal of Northwest University, 2014, 44(5):729-732.

[15] 刘会改. 面向云计算架构的资源管理的研究与实现[D]. 北京邮电大学, 2011.

LIU H G. Research and Implementation of Cloud Computing-oriented Resource Management[D]. Beijing University of Posts and Telecommunications, 2011.

#### 作者简介:



张笑燕 (1973-), 女, 山东烟台人, 博士, 北京邮电大学教授, 主要研究方向为移动互联网软件与软件工程理论。



王敏 [通信作者] (1989-), 女, 安徽芜湖人, 北京邮电大学硕士生, 主要研究方向为云计算与移动互联网软件。E-mail:wangne402@163.com。



杜晓峰 (1973-), 男, 陕西韩城人, 北京邮电大学讲师, 主要研究方向为云计算与大数据分析。