一种面向云服务提供商的资源分配机制

殷 波* 王 颖 邱雪松 孟洛明
(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

摘 要：针对用户、服务提供商(SP)和基础设施提供商三者之间的市场行为所带来的收益问题，该文面向服务提供商提出了一种两阶段多虚拟机资源分配机制。首先，建立能力规划模型并引入粒子群算法，决策出服务提供商收益最大化的资源购买方案。然后，提出一种优化用户满意度的服务提供策略，优化了服务提供商的长期收益。最后仿真实验结果表明，该方法有效提高了服务提供商的收益，并取得了较好的用户满意度。

关键词：云计算；资源提供；用户满意度；收益优化
中图分类号： TP393 **文献标识码**： A **文章编号**：1009-5896(2014)01-0015-07
DOI: 10.3724/SP.J.1146.2013.00427

A Resource Provisioning Mechanism for Service Providers in Cloud

Yin Bo Wang Ying Qiu Xue-song Meng Luo-ming
(*State Key Laboratory of Networking and Switching Technology*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)

Abstract: Considering Service Providers (SP) revenue optimization issue, a two-stage multi-virtual machine resource provisioning mechanism is proposed in this paper. Firstly, a capacity planning model is proposed and the particle swarm algorithm is used to make resource purchase set, which can gain the maximize profit for SP. Then, a utility function to measure customer's satisfaction is proposed to optimize SP's profit in the long way. Simulation experimental results show that the proposed method improves effectively SP's profit, and gains better user's satisfaction.

Key words: Cloud computing; Resource provisioning; User's satisfaction; Profit optimization

1 引言

随着云计算的发展，越来越多的服务提供商(Service Providers, SP)为了节省前期基础设施投入，逐步将服务部署到基础设施提供商(IP)所提供的云环境中。面向云服务提供商的资源分配问题[1]，主要包括两个方面：(1)服务提供商-基础设施提供商(SP-IP)之间的虚拟机提供：SP 估计自己的需求量，向 IP 购买虚拟机，以花费的成本最小为目标，制定虚拟机实例的购买方案；(2)服务提供商-用户(SP-User)之间的服务提供：SP 根据用户请求的特点，为请求提供虚拟机，以满足用户需求，以收益最大化为目标，制定虚拟机实例的提供方案。

目前针对 SP 的资源分配问题，文献[2,3]通过建立合理的定价机制，对 SP 的应用服务定价，实现 SP 收益优化。文献[2]通过建立成本分析模型，来降低资源分配的总成本，但该文献的成本分析模型并未充分考虑用户偏好的因素，因此对用户满意度的度量不准确。文献[3]与文献[2]类似，使用进程共享机制设计了一种定价模型，通过应用之间的相关性分析，设计了两种收益驱动的调度算法，但忽略了用户满意度对调度策略的影响。文献[4]通过预测负载的动态特性来解决 SP-User 以及 SP-IP 的资源分配问题，提出一种新的收益模型来描述 SP 在特定服务级别协议(SLA)下的收入，并采用了一种收益驱动的启发式算法来实现 SP 的收益最大化。但文献[4]中的 SLA 同样不能充分反映用户偏好，不能有效描述用户满意度。为保证高服务质量，文献[5]设计了基于响应时间的收益模型，通过增加队列响应时间来增加或者降低收益。文献[6,7]提出了基于 SP 效用函数的服务提供策略，关注用户效用和 SP 收益的折中来增加 SP 收益。文献[8]提出了一种基于 SLA 的服务提供方法，采用启发式算法来获取近似最优解。还有研究通过将多服务系统的服务请求看作 M/M/m 队列模型[9]，或设计两阶段的多虚拟机

2013-04-01 收到，2013-08-15 改回
国家 863 计划项目(2013AA013502)，国家自然科学基金创新研究群体科学基金(61121061)和国家科技支撑计划(2012BAH06B02)资助课题
*通信作者：殷波 yinbo@bupt.edu.cn

规划模型[10]，来解决 SP 收益优化的资源分配问题。但该方法并未进行 SP-IP 的虚拟机提供，缺乏对资源提供集合的预判断。上述文献均基于单一虚拟机类型的资源分配，未考虑 IP 资源类型的多样性。

为此，本文提出一种用户满意度和 SP 收益折中的两阶段多虚拟机资源提供机制，该机制首先在虚拟机提供阶段，对按需市场和现货市场进行能力规划，制定出满足需求的虚拟机提供方案。随后在服务提供阶段，建立包含用户出价和请求响应时间的用户效用函数，来描述用户对 SP 服务满意度，并设计用户满意度优化的调度算法，实现 SP 收益优化基础上用户满意度的提高。

2 SP 的虚拟机提供

2.1 问题模型

目前 IP 提供虚拟资源的方式主要分为 3 种：按需市场、预约市场和现货市场。按需市场中的资源按小时收费，SP 根据应用需求随时增加或减少资源购买数量。预约市场是在使用虚拟资源之前，SP 与 IP 签订合同预定一定数量的资源，并支付定金。现货市场是 IP 回收 SP 预约的多余闲置资源重新进行拍卖，其它 SP 出价高者得到资源使用权，现货实例价格最低，但存在 IP 可以随时回收现货实例的风险。由于预约市场需要长期预约，本文在 SP-IP 的虚拟机提供阶段，对实时动态变化的工作负载仅针对按需和现货市场制定资源分配方案。

SP 将应用服务部署到云环境中，用 $r_i(\lambda_i, t_i, \mathrm{re}_i)$ 表示用户 i 向 SP 提交的工作请求，其中 λ_i 表示工作请求的大小，t_i 表示请求的持续运行时长，re_i 表示用户 i 需要向 SP 支付的费用。用 u_i 表示请求 r_i 所需的计算能力，一般情况下 $u_i = \lambda_i \cdot t_i$。用 Ω 表示全体实例类型的集合，用 p_i^o 表示按需市场单位时间内一个虚拟机实例 i 的价格；x_i^o 表示按需市场中在此次决策周期购买虚拟机实例 i 的数量，则按需市场中虚拟机实例的总成本表示为

$$\mathrm{ct}^o = \sum_{i\in\Omega} p_i^o x_i^o \tag{1}$$

由于现货市场的资源实例具有不稳定性，在使用现货市场的虚拟机实例时，具有资源使用权随时被 IP 收回的风险。因此，用 δ_i^s 表示虚拟机实例 i 的价格低于本文所出价格 p_i^s 的概率，即 SP 以价格 p_i^s 获取虚拟机实例 i 使用权的概率。p_i^s 表示在该决策周期内虚拟机实例 i 的价格，每个决策周期内 SP 对实例的出价不变。用 x_i^s 表示 SP 以价格 p_i^s 所购买的虚拟机实例 i 的数量，则现货市场中虚拟机实例的总成本表示为

$$\mathrm{et}^s = \sum_{i\in\Omega} p_i^s x_i^s \delta_i^s \tag{2}$$

2.2 SP 虚拟机提供模型及算法

通过上述的问题描述，建立虚拟机提供阶段 SP 收益优化的目标函数为

$$\left.\begin{aligned} &\min \quad \mathrm{et}(\boldsymbol{x}_i) = \gamma \sum_{i\in\Omega} p_i^o x_i^o + \eta \sum_{i\in\Omega} p_i^s x_i^s \delta_i^s \\ &\text{s.t.} \quad \sum_{i\in\Omega} c_i \cdot (x_i^o + x_i^s) \ge u_i, \quad x_i^o \in [0, N] \end{aligned}\right\} \tag{3}$$

其中 γ 和 η 表示按需市场和现货市场的购买比例，$\gamma + \eta = 1$。c_i 表示实例 i 的处理能力。在目标函数中需求解的未知量为 x_i^o 和 x_i^s，该问题的子问题在文献[11]中被证明为 NP 问题，因此需采用启发式算法对其进行求解。在启发式算法中，粒子群算法[12]与本文问题模型具有较为一致的映射关系，因此选取粒子群算法对其进行求解。总体的虚拟机提供方案表示为 Λ。

粒子群优化(PSO)算法首先生成具有位置和速度参数的一群粒子，更新公式如式(4)，式(5):

$$\boldsymbol{v}_{i+1} = w\boldsymbol{v}_i + d_1\sigma_1(\boldsymbol{x}_{pb} - \boldsymbol{x}_i) + d_2\sigma_2(\boldsymbol{x}_{gb} - \boldsymbol{x}_i) \tag{4}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{v}_{i+1} \tag{5}$$

其中 $\boldsymbol{x}_{pb}$ 表示粒子本身达到过的最优位置，$\boldsymbol{x}_{gb}$ 表示整个粒子群到达过的最优位置。式(4)中 $\boldsymbol{x}_i$ 表示第 i 个粒子当前的位置，$\boldsymbol{v}_i$ 表示其当前速度。ω 表示粒子保持惯性的权重；d_1，d_2 表示粒子的加速度，分别表示粒子向本身最优位置和全局最优位置移动的趋势，σ_1，σ_2 是 (0,1) 之间均匀产生的随机数。

本文对粒子群优化算法进行了改进，重定义了算法中粒子的位置与速度，根据粒子在迭代过程中适应度值的变化，动态调整粒子速度更新的权重，以此提高粒子群算法的收敛速度。粒子重定义为

$$\boldsymbol{x}_i = \left(x_1^o, x_2^o, \cdots, x_m^o,\ x_1^s, x_2^s, \cdots, x_m^s\right) \tag{6}$$

其中 x_i^o 表示在按需市场购买的第 i 类虚拟机的数

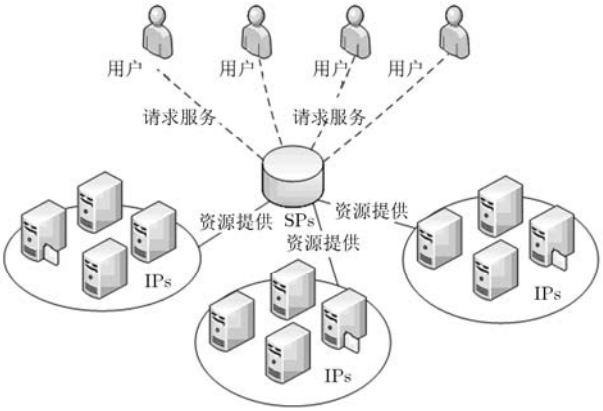

图 1 云环境角色关系图

量；x_i^s 表示在现货市场购买的第 i 类虚拟机的数量。相应地，将粒子的速度重定义为

$$\boldsymbol{v}_i = \left(v_1^o, v_2^o, \cdots, v_m^o, v_1^s, v_2^s, \cdots, v_m^s\right) \tag{7}$$

其中 v_i^o 表示在按需市场购买的第 i 类虚拟机数量的变化量；v_i^s 表示在现货市场购买的第 i 类虚拟机数量的变化量。通过式(4)，式(5)的更新迭代，改进算法得到最优化的虚拟机购买方案。

针对 PSO 算法容易陷入局部最优出现早熟收敛的现象，算法改进如下：动态调整在不同迭代阶段的惯性权重 ω 取值，使算法自适应地调整全局系数。在算法开始阶段搜索更多的解空间，而在算法后期能够更好地收敛到最优解范围，兼顾了搜索效率和搜索精度。

该方法将当前种群的迭代次数 k 和当前的惯性权重 $\omega(k) \in [0.0,1.0]$ 作为输入变量，进行惯性权重的动态调整，将下一代惯性权重的值 $\omega(k+1)$ 作为输出变量：

$$\omega(k+1) = \omega(k)\left(1 - k/(Km)\right) \tag{8}$$

其中 k 为算法的当前迭代次数；K 为算法总的迭代次数；惯性权重的初始值 $\omega(0)=1$；m 为整数，在算法开始前选定，且 $m \in [1,10]$。改进算法使粒子在进化前期以较大惯性权重值保证算法的全局搜索能力，而在收敛的后期阶段以较小惯性权重加快收敛，改善了算法的收敛性。由式(4)，算法通过自适应惯性权重动态调整后的形式为

$$\begin{aligned}\boldsymbol{v}_i^{k+1} &= \omega(k)\cdot\boldsymbol{v}_i^k + d_1\sigma_1\cdot\left(\boldsymbol{x}_{pb}(k) - \boldsymbol{x}_i^k\right)\\ &\quad + d_2\sigma_2\cdot\left(\boldsymbol{x}_{gb}(k) - \boldsymbol{x}_i^k\right)\end{aligned} \tag{9}$$

基于 PSO 的虚拟机提供算法(PSOVP)的步骤为：

步骤 1　初始化：将粒子群规模数设定为 N，将算法执行的最大迭代次数设定为 K。位置参数 $\boldsymbol{x}_i$ 和初始速度 $\boldsymbol{v}_i$ 均随机生成。根据式(3)计算出粒子的适应度 $\mathrm{ct}(\boldsymbol{x}_i)$，将适应度排序得到全局最优初始位置 $\boldsymbol{x}_{gb}$，并设定粒子个体最优初始位置 $\boldsymbol{x}_{pb}$；

步骤 2　迭代控制：如果 $k == K$，则转到步骤 4；

步骤 3　迭代过程：(1) 对每个可行的粒子，计算 $\mathrm{ct}(\boldsymbol{x}_i)$，如果 $\mathrm{ct}(\boldsymbol{x}_i) \le \mathrm{ct}(\boldsymbol{x}_{pb})$，则 $\boldsymbol{x}_{pb} = \boldsymbol{x}_i$；如果 $\mathrm{ct}(\boldsymbol{x}_{pb}) \le \mathrm{ct}(\boldsymbol{x}_{gb})$，则 $\boldsymbol{x}_{gb} = \boldsymbol{x}_{pb}$；(2)进行速度和位置更新；在位置更新中虚拟机实例的选择按照局部适应优先的原则进行；(3)转到步骤 2；

步骤 4　结果输出：输出虚拟机提供方案 Λ 及其适应度值 $\mathrm{ct}(\boldsymbol{x}_i)$。

3　SP 的服务提供

3.1 用户效用函数

现有的服务提供策略将用户效用与请求响应时间建立关系，作为用户满意度的评价指标。此外，用户购买应用服务所支付的成本也是影响用户满意度的一个重要方面。本文基于经济学中的效用理论，建立请求响应时间和服务成本的用户效用函数为

$$s(p_i, t_i) = s_i^0 - p_i^{-\alpha} - t_i^{-\beta} \tag{10}$$

其中 α 和 β 表示用户对金钱和时间的偏好。s_i^0 表示用户的初始满意度，与应用请求的能力大小相等。如图 2 所示，该图中的无差异曲线表示了时间和金钱的边际替代关系，其中纵坐标表示用户向 SP 支付的金钱，横坐标表示用户的请求响应时间，s_1, s_2 表示用户效用函数。图 2 中同一条曲线上的点所表示的用户效用值相同。不同曲线的用户效用值不同，点 A 和点 C 的用户效用值相同。在保持价格相同的前提下，可以通过降低请求响应时间来实现用户效用的优化，即从点 C 移动到点 B。在保持响应时间相同的前提下，可以通过降低价格来实现效用优化，即从点 A 移动到点 B。

3.2 用户满意度优化算法

用 $R = \{r_1, r_2, \cdots, r_m\}$ 表示应用请求，用 $I = \{i_1, i_2, \cdots, i_n\}$ 表示第 1 阶段求得的虚拟机提供集合 Λ 中的实例。对每个请求 $r_i \in R$，定义如下变量：ct_i 表示 SP 为请求 r_i 分配虚拟机实例支出的成本，p_{ij} 表示为请求 r_i 分配的每个虚拟机实例 j 的价格；re_i 表示 SP 完成请求 r_i 所能获得的收入，$\mathrm{re}_i = \sum_i p_i$，$p_i$ 表示处理请求 i 云服务提供商所获得的收入；用 rpt_{ij} 表示将请求 r_i 分配到虚拟机实例 j 上的剩余处理时间。引入性能指标 PI_i 来度量不同类型虚拟机实例性能的差异。$PI_j = c_j / c_o$，其中 c_j 表示虚拟机类型 j 的能力大小，c_o 表示单位实例的处理能力，$\mathrm{rpt}_{ij} = t_i / PI_j$。

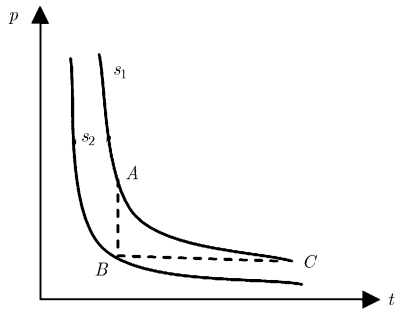

图 2 用户效用的无差异曲线

在服务提供阶段设定一个处理每个请求的最低收益 $\mathrm{pf}_{\min}$，之后在此基础上进行用户满意度的优化。根据上述分析，为处理请求 r_i 所需资源的成本计算公式为

$$\mathrm{ct}_i = \sum_j p_{ij} \cdot \mathrm{rpt}_{ij} \tag{11}$$

SP 针对每个到达的请求 r_i 在保证最低收益 $\mathrm{pf}_{\min}$ 的基础上，求解用户满意度优化的资源分配方案。则处理请求 r_i 需对用户收取的费用为

$$p_i = \mathrm{pf}_{\min} + \mathrm{ct}_i \tag{12}$$

式(10)可化简为

$$s_i = s_i^0 - \left(\mathrm{pf}_{\min} + \sum_j p_{ij} t_i c_i / c_o\right)^{-\alpha} - t_i^{-\beta} \tag{13}$$

式(13)中的变量为 c_i 和 p_{ij}，在收益约束的基础上进行的用户满意度优化，其目标函数为

$$\left.\begin{aligned} \max \quad & s_i = s_i^0 - \left(\mathrm{pf}_{\min} + \sum_j p_{ij} t_i c_i / c_o\right)^{-\alpha} - t_i^{-\beta} \\ \text{s.t.} \quad & \sum_i \mathrm{ct}_i \le \mathrm{ct}(\boldsymbol{x}_i) \end{aligned}\right\} \tag{14}$$

因此，SP 执行用户请求所获得总收益的计算公式为

$$\mathrm{pf} = \sum_i p_i - \mathrm{ct}(\boldsymbol{x}_i) \tag{15}$$

本文设计了一种服务提供算法(SPAP)对上述问题模型进行求解。算法基于贪婪思想，为不同的用户请求提供合理的虚拟机映射方案，一方面保证 SP 的收益，另一方面保障用户的满意度。服务提供算法(SPAP)的具体步骤如下：

输入：虚拟机集合 Λ，待处理请求集合 R

输出：虚拟机分配方案

步骤 1 从待处理请求集合中选择请求 $r_i \in R$，进行虚拟机分配；

步骤 2 计算并更新 Λ 中现有虚拟机类型对应于请求 r_i 的 ct_i 和 rpt_{ij}；

步骤 3 计算将不同类型虚拟机分配给请求 r_i 时，所得到的用户满意度；

步骤 4 在分配方案中选择使 SP 开销 ct_i 最小且用户满意度最高者，作为分配给请求 r_i 的结果；

步骤 5 若待处理请求集合不为空，则选择下一条请求 $r_j \in R$，执行步骤 2；否则，算法结束。

4 SP 资源分配框架

为部署上述虚拟机提供和服务提供方法，本文提出了一种 SP 资源分配框架，如图 3 所示。资源监测模块用来采集 IP 的实例信息，包括可用的虚拟机实例的类型和价格。用户信息采集模块用来获取用户请求、偏好参数等信息。规划分析模块由虚拟机提供模块和服务提供模块组成，监测模块采集到的数据传递给规划分析模块后，虚拟机提供模块针对用户的能力需求以及可用的虚拟机资源类型，制定虚拟机提供方案，对应于 PSOVP 算法。服务提供模块对每个用户请求进行用户满意度优化的请求调度，制定将请求分配到虚拟机实例的分配方案，对应于 SPAP 算法。决策执行模块根据规划分析模块制定的资源分配策略执行 IP 的虚拟机实例的购买或租用操作。

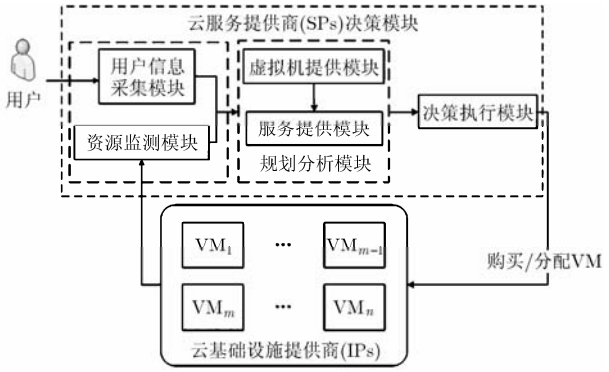

图 3 SP 自主资源分配框架

5 实验验证

5.1 仿真环境

根据上述模型，本文基于 Visual Studio2005 设计了实验的仿真模拟环境。本文所提算法作为 SP 决策模块的主体调度，将用户请求和实例数据作为输入，而将购买/分配方案作为输出。

本文参照亚马逊的实例类型和价格作为实验数据。采用国内某互联网公司一段时间内的任务量作为输入数据。表 1 为亚马逊 EC2 在 US East (N. Virginia)按需市场的实例类型和价格。实验采用的硬件配置是 Xeon E5620(2.4 G)4 核 8 线程的处理器，30 GB DDR 的内存，SAS 1 TB RAID5 硬盘，物理服务器使用 Windows Server 2008R2 SP1 操作系统。实验平台为虚拟机分配的相应参数与亚马逊 EC2 实例配置相同。为测定亚马逊 EC2 各类型实例的能力差异，将一段 1024-frame 1080p 的视频加密任务在 4 种实例上运行，运行时间如表 1 所示。表 2 中为本实验用到的参数值。表 3~表 6 中统计的值为近两年内现货市场采用相应价格区间的概率。实验从 SP 收益、单位收益、虚拟机实例利用率等方面对算法进行了仿真实验，并与文献[7]中的算法进行了对比。

表 1 亚马逊 EC2 按需市场的实例价格及基准程序测试的能力值

实例类型	Windows 价格(h)	平均时间(s)	方差(s)
小型实例(Small)	0.115	506.1	3.2
中型实例(Medium)	0.230	387.4	2.1
大型实例(Large)	0.460	113.5	1.7
超大型实例 (Extra Large)	0.920	40.6	1.3

表 2 实验中用到的参数值

参数	取值
粒子群迭代次数 K	100
用户偏好比 α/β	7.0, 4.0, 1.0, 0.5
现货市场购买比例 γ	0.8
按需市场购买比例 η	0.2

5.2 结果分析

图 4 所示是国内某搜索企业一年内平均每天随时间到达的请求量，本文对原始数据进行了初步的归一化处理，实验将该数据作为所提方法的输入数据。实验的评价指标如下：

SP 总收益　图 5 为不同算法获得的 SP 总收益对比图，从图 5 可得，随着任务量的增加，SP 获得的收益逐渐增高，SP 收益较之对比方法[7]提高了近 20%。主要原因在于，本文方法针对一段时间内的请求量进行资源规划，首先 PSOVP 算法以节省成本为目标使得 SP 在购买虚拟机过程中的支出更加合理。而在 SPAP 算法中，则进一步考虑了用户满意度，从长期来说，用户满意度的提升会对 SP 的总体收益带来正面影响。PSOVP 算法以成本最小化为目标进行多类型虚拟机资源分配是保证 SP 能够获得更多收益的主要原因。

表 3 现货市场小型实例的价格

价格区间	[0, 0.038)	[0.038, 0.040)	[0.040, 0.042)	[0.042, $+\infty$)
2011 年的概率(%)	0.03	49.60	41.26	9.11
2012 年的概率(%)	0.03	58.74	40.21	1.36

表 4 现货市场中型实例的价格

价格区间	[0, 0.225)	[0.225, 0.240)	[0.240, 0.400)	[0.400, $+\infty$)
2011 年的概率(%)	0.01	46.85	48.26	4.88
2012 年的概率(%)	0.02	57.64	40.21	2.37

表 5 现货市场大型实例的价格

价格区间	[0,0.370)	[0.370,0.460)	[0.460,0.660)	[0.042, $+\infty$)
2011 年的概率(%)	6.17	51.71	39.68	2.44
2012 年的概率(%)	5.28	47.11	46.97	0.64

表 6 现货市场超大型实例的价格

价格区间	[0, 0.50)	[0.50, 0.79)	[0.79, 0.90)	[0.90, $+\infty$)
2011 年的概率(%)	0.75	39.85	42.16	17.24
2012 年的概率(%)	0.12	32.46	50.21	17.21

单位收益　用单位收益 m_0 来表示 SP 处理每个应用请求所能分别获得的收入，其中 $m_0 = \mathrm{pf}\big/\sum_i \lambda_i$。如图 6 所示，与文献[7]方法相比，首先由于在虚拟机购买过程中 PSOVP 算法制定了成本较小的资源购买方案，同时虚拟机分配过程中 SPAP 算法考虑了 SP 的最低成本收益以及用户偏好，因此本文方法能在有效提高 SP 总收益的基础上优化 22%的单位收益。

虚拟机实例利用率　图 7 所示为本文在 α/β=1 时使用的虚拟机实例的资源利用率。由于在虚拟机分配过程中将用户偏好引入用户满意度函数，较之前的方法，本文的方法在 SPAP 算法中为请求分配更能满足用户偏好的虚拟机实例。因此虚拟机分配方案一方面能够满足用户的应用需求，另一方面则

尽可能减少了资源浪费。

用户偏好对单位收益的影响 图 8 表明在不同用户偏好(α/β)的情况下，所获得的单位收益的对比图。当α/β从 7.0 降到 1.0 时单位收益在增加，当α/β从 1.0 下降到 0.2 时单位收益在减少。SPs 可以通过设定不同的用户满意度水平来划分收益区间，根据不同情况制定资源分配方案。

6 总结

本文针对 SP 资源分配过程中的收益优化问题，提出了一个两阶段的多虚拟机资源分配机制。首先针对按需市场和现货市场对应用需求进行能力规划，决策出满足服务提供商收益优化的资源提供方案。之后使用经济学中的效用理论，建立与请求响应时间和用户出价相关的用户效用模型，用来度量用户满意度。并提出了一种用户满意度与收益优化折中服务提供算法，该方法有效提高了用户满意度，有利于实现服务提供商的长期收益优化。最后，实验利用了亚马逊EC2的实例数据和国内某搜索企业的请求数据进行仿真。仿真结果表明，所提方法有效提高了 SP 收益，并实现了用户满意度的优化。

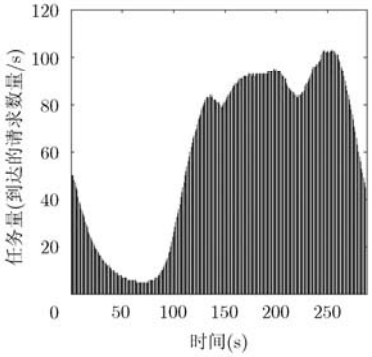

图 4 任务量随时间分布图

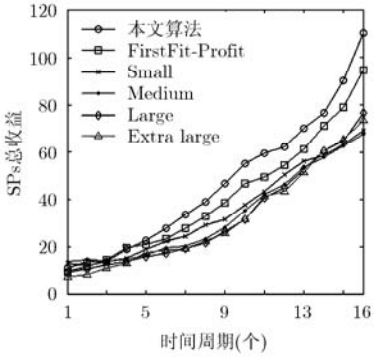

图 5 SP 总收益对比图

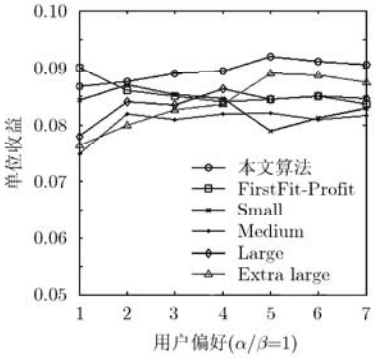

图 6 单位收益示意图

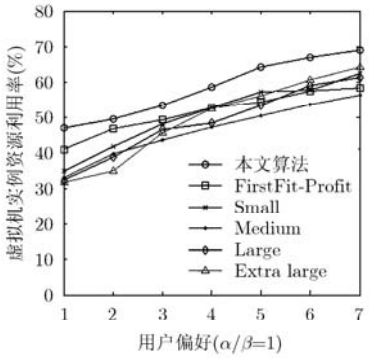

图 7 虚拟机实例利用率示意图

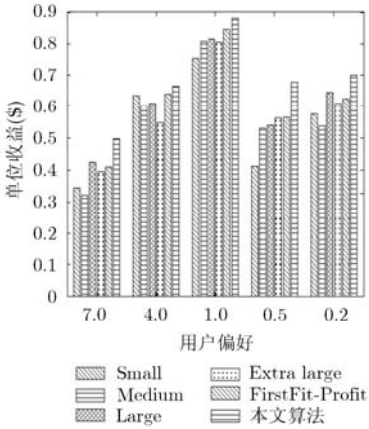

图 8 用户偏好和单位收益的关系图

参 考 文 献

[1] Chen Wei, Qiao Xiao-qiang, and Wei Jun. A profit-aware virtual machine deployment optimization framework for cloud platform providers[C]. 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), Honolulu, 2012: 17–24.

[2] Thanakornworakij T, Nassar R, Leangsuksun C B, *et al.*. An economic model for maximizing profit of a cloud service provider[C]. 2012 Seventh International Conference on Availability, Reliability and Security (ARES), Prague, 2012: 274–279.

[3] Lee Y C, Wang C, Zomaya A Y, *et al.*. Profit-driven service request scheduling in clouds[C]. 2010 10th IEEE/ACM International Conference on Cloud and Grid Computing (CCGrid), Melbourne, 2010: 15–24.

[4] Goudarzi H and Pedram M. Maximizing profit in cloud computing system via resource allocation[C]. 2011 31st International Conference on Distributed Computing Systems Workshops (ICDCSW), Minneapolis, 2011: 1–6.

[5] Li J, Wang Q Y, Deepal J, *et al.*. Profit-based experimental

analysis of IaaS cloud performance: impact of software resource allocation[C]. 2012 IEEE Ninth International Conference on Services Computing (SCC), Honolulu, 2012: 344–351.

[6] Mazzucco M, Vasar M, and Dumas M. Squeezing out the cloud via profit-maximizing resource allocation policies[C]. 2012 IEEE 20th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), Washington, DC, 2012: 19–28.

[7] Chen Jun-liang, Chen Wang, Zhou Bing-bing, *et al.*. Tradeoffs between profit and customer satisfaction for service provisioning in the cloud[C]. Proceedings of the 20th International Symposium on High Performance Distributed computing, ACM, San Jose, 2011: 229–238.

[8] Hyun J M, Yun C, Hacigu X, *et al.*. SLA-aware profit optimization in cloud services via resource scheduling[C]. 2010 6th World Congress on Services (SERVICES-1), Minneapolis, 2010: 152–153.

[9] Cao Jun-wei, Hwang Kai, Li Ke-qin, *et al.*. Optimal multiserver configuration for profit maximization in cloud computing[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2012, 24(6): 1087–1096.

[10] Chaisiri S, Bu-Sung L, and Niyato D. Profit maximization model for cloud provider based on Windows Azure platform[C]. 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phetchaburi, 2012: 1–4.

[11] Stage A, Setzer T, and Bichler M. Automated capacity management and selection of infrastructure-as-a-service providers[C]. IEEE Symposium Intergrated Network Management, New York, 2009: 20–23.

[12] Poli R, Kennedy J, and Blackwell T. Particle swarm optimization[J]. *Swarm Intelligence*, 2007, 1(1): 33–57.

殷　波：　女，1984 年生，博士生，研究方向为云计算.

王　颖：　女，1976 年生，副教授，硕士生导师，研究方向为云计算、通信软件与网络管理.

邱雪松：　男，1973 年生，教授，博士生导师，研究领域为网络与业务管理.

孟洛明：　男，1955 年生，教授，博士生导师，研究领域为通信网、网络管理、通信软件.