



面向成本优化的云计算资源分配方法研究

殷 波,张云勇,房秉毅,冯伟斌

(中国联合网络通信有限公司研究院 北京 100032)

摘 要:针对云计算运营环境中,云服务提供商租用云资源提供商的资源,向用户提供服务的场景,以优化云服务提供商的成本支出,增加其运营收益为目标,设计了一种面向成本优化的虚拟机资源分配方法。该方法全面考虑了现货、预约和按需3种资源分配模式,引入风险控制因子规避现货市场的资源风险,制定出成本优化的资源分配方案。所提方法在满足用户满意度基础上,有效实现了虚拟机资源成本的优化,增加了云服务提供商的收益。

关键词:成本优化;云计算;资源分配;数据中心

doi: 10.3969/j.issn.1000-0801.2014.11.004

Cloud Resource Allocation Method Based on Elastic Resource Adjustment

Yin Bo, Zhang Yunyong, Fang Bingyi, Feng Weibin

(China Unicom Research Institute, Beijing 100032, China)

Abstract: For cloud computing operating environment, cloud service providers leased cloud resources provider of resources, to provide the service to the user of the scene, taking the optimization of a cloud service provider costs, increase the operation profit as the goal, the design of the virtual machine resource allocation method for cost optimization was proposed. The method comprehensively considered the spot, appointment and on-demand resource allocation model, control factor was introduced to avoid the resource risk, the resource allocation scheme for the optimization of the cost was formulated. The proposed method meets the users' satisfaction, effectively realizes the optimization of virtual machine resource cost, increases the cloud service provider revenue.

Key words: cost minimization, cloud computing, resource allocation, data center

1 引言

云计算具有资源按需提供的特征,为用户以弹性伸缩的方式分配计算资源。该方式灵活高效地实现了资源的按需分配,提高了资源利用率,降低了运营成本。目前,以虚拟化技术为基础的云计算资源分配方法,主要以虚拟机为粒度,为用户提供服务,处理用户的应用请求。用户与云服务提供商签订严格的SLA,云服务提供商(SP)在保证用户SLA的前提下,根据用户应用的负载状况,自动弹性调整

资源分配量。云服务提供商将服务部署到云计算环境中,既节省了前期基础设施投入的成本,又进一步降低了后期软硬件资源的管理运维成本,为云服务提供商带来了直接经济收益。云服务提供商将服务部署到云环境中,云数据中心拥有者需要将云服务提供商的用户请求分配到各个虚拟机资源中进行处理。云数据中心的拥有者在本文中被称为云资源提供商,云资源提供商拥有数据中心,负责将云服务提供商的用户请求进行任务分配。

云计算将服务提供和资源提供两个过程解耦合,云资

源提供商以特定的方式提供资源,并保证资源的可用性以及可靠性,针对云服务提供商的用户请求负责制定合理的虚拟机分配方案进行资源提供,为用户请求分配合理的虚拟机资源。云服务提供商需要在保证用户满意度的基础上,为用户请求分配制定虚拟机资源使用方案,尽可能降低虚拟机资源的使用成本,基于云服务提供商的角度进行成本优化和资源分配是本文需要解决的主要问题。

本文首先分析了现有云计算环境的资源分配现状,抽象出面向成本优化的云服务提供商资源分配问题的数学模型,然后采用改进粒子群算法对问题进行了求解。

2 相关工作

目前基于云服务提供商进行保证 SLA 的资源分配策略的研究有很多。在基于定价机制进行资源分配的研究中,根据不同的资源购买方式,制定出成本分析模型是资源分配研究的核心。例如,参考文献[1,2]提出了一种在云计算环境下针对按需、现货和预约 3 种市场,设计面向 SP 收益优化的资源分配方法,然而该文献并未深入研究和探讨现货市场的风险问题。此外,为了提高 SP 的收益,参考文献[3]首先建立成本分析模型,并制定 SP 总成本最小的资源提供方案,以满足用户需求,然而,在该文献中,成本分析模型忽略了市场模型的多样性以及现货市场的风险对 SP 长期收益的影响,导致该文献方法不能准确度量成本和收益。参考文献[4]设计了一种定价机制,通过分析应用之间的相关性,提出了两种 SP 收益驱动的资源提供算法。然而,该文献同样没有考虑现货市场对资源分配方案的影响,同样导致成本和收益度量不准确。此外,参考文献[6]阐述了基于市场的资源提供策略的风险和优势。参考文献[5]对云计算环境的盈利模式进行了分析,包括分析服务收费和商业成本,并且确定了应用特征与云资源提供商资源价格及收益之间的关系,提出了一种基于应用特征的定价模型,综合考虑应用的工作负载环境、多服务系统的配置、服务级别协议、用户满意度、服务质量以及 SLA 违约惩罚、能耗成本等,制定出云服务提供商的收益模型,并且将服务系统看作 $M/M/m$ 队列模型,对优化问题进行建模和分析。然而,该文献并未针对目前业界主流的市场模型进行方案设计,并且同样忽视了现货市场的风险对模型的影响,同样具有成本度量不准确的问题。

本文针对现有研究的不足,基于目前的市场模型,从按需市场、预约市场和现货市场 3 种情况出发,设计了 SP

的资源分配模型,同时针对现货市场的不稳定性,引入风险控制因子,保证现货市场的可用性,实现了对市场模式的全面度量,并进一步对资源进行有效利用,提高资源利用率,降低资源分配成本,实现了成本优化。

3 面向成本优化的虚拟机资源分配问题描述

SP 将应用服务部署到云环境中,向用户提供服务,并对使用应用的用户进行收费。用户向 SP 提交的工作请求用 $r_i(\lambda_i, t_i)$ 进行表示, λ_i 表示任务资源需求大小, t_i 表示任务的运行时长。SP 在进行资源分配和任务调度的过程中,首先,需要对用户的任务请求进行度量,用 u_i 表示请求 r_i 所需的计算能力,一般情况下 $u_i = \lambda_i t_i$ 。

以亚马逊为例,目前的云计算运营商通常将其所拥有的虚拟机资源分为 3 类:预约类型、按需类型和现货类型。云资源提供商通常将云数据中心内的虚拟机进行上述分类,针对不同用户的任务请求,将任务分配到 3 种不同的虚拟机实例上去。SP 面向 3 种资源类型,制定面向成本优化的虚拟机购买方案,实现 SP 的收益优化。

针对云计算运营市场环境中的 3 种角色,整个 SP 资源分配过程如图 1 所示。云资源提供商拥有云数据中心,并且通过虚拟机分配决策模块对用户请求进行任务调度和资源分配。SP 根据用户请求量设计资源购买方案,租用云资源提供商的资源为用户提供服务,获取收益。制定 SP 资源购买方案的模块是虚拟机分配决策模块。本文的问题及其解决方法主要是在虚拟机分配决策模块进行策略制定。虚拟机分配决策模块通过信息采集模块对用户请求的信息进行采集,并通过资源监测模块采集云数据中心的各项资源信息。信息采集模块获取资源信息后,将数据发送至资源分配模块,本文所提方法被放置在资源分配模块,

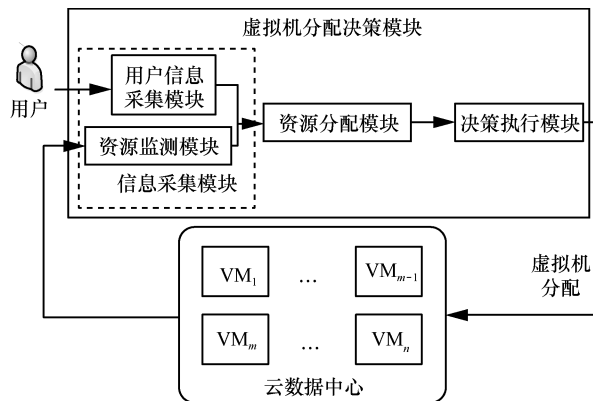


图 1 云计算资源分配框架



资源分配模块通过对数据进行处理,并使用资源分配方法,制定出满足 SP 收益最优的资源分配策略,并将该策略发送至决策执行模块。决策执行模块对云数据中心发送指令,进行虚拟机资源分配。在此过程中,SP 资源分配策略的制定原则是成本优化、收益最优。

本文基于云服务提供商的角度,对用户请求进行资源分配,在完成用户请求的描述后,接下来对云数据中心的资源进行描述。数据中心所能提供的虚拟机资源类型用 Ω 表示,按需市场环境下使用虚拟机实例 i 的单位时间的价格用 p_i^o 表示; n_i^o 表示在此次决策周期内从按需类型中选取的用于处理用户请求的虚拟机实例的数量,则云资源提供商分配按需类型虚拟机实例的总成本可表示为:

$$cost^o = \sum_{i \in \Omega} p_i^o n_i^o \quad (1)$$

用 p_i^d 表示预约类型虚拟机实例 i 单位时间内的价格; n_i^d 表示在此次决策周期内从预约类型中选取的虚拟机实例 i 的数量,则云资源提供商分配预约类型虚拟机实例的总成本可表示为:

$$cost^d = \sum_{i \in \Omega} p_i^d n_i^d \quad (2)$$

现货类型是云资源提供商将按需类型和预约类型虚拟机分配完成之后,将闲散资源组织在一起,进行资源提供的一种资源分配方式。因此,现货类型的虚拟机资源具有不稳定性、可靠性和可用性较低等特点。但闲散的现货类型虚拟机资源的优势是其成本消耗极低,使用现货类型的虚拟机实例能有效降低成本支出,是云资源提供商资源回收利用的一种方法。在使用现货资源进行虚拟机分配时,需要规避资源风险。

本文设定了现货类型资源的风险规避因子。具体方法是,用 δ_i^s 表示云资源提供商以成本支出 p_i^s 将任务分配到虚拟机实例 i 的概率,本文将该因子称为风险规避因子。现货市场中虚拟机实例的价格用 p_i^s 表示。用 n_i^s 表示云资源提供商以成本 p_i^s 在此次决策周期内分配给任务请求的虚拟机实例 i 的数量。使用云资源提供商提供的现货类型虚拟机实例的总成本表示为:

$$cost^s = \sum_{i \in \Omega} p_i^s n_i^s \delta_i^s \quad (3)$$

通过上述的问题描述,建立 SP 面向成本优化的虚拟机分配的目标函数如下:

$$\begin{aligned} \min \quad cost(x_i) &= \varphi \sum_{i \in \Omega} p_i^d n_i^d + \gamma \sum_{i \in \Omega} p_i^o n_i^o + \eta \sum_{i \in \Omega} p_i^s n_i^s \delta_i^s \quad (4) \\ \text{s.t.} \quad &\sum_{i \in \Omega} c_i (n_i^o + n_i^s) \geq u_i, \quad n_i^o \in [0, N] \end{aligned}$$

其中, γ 和 η 表示按需类型和现货类型虚拟机的分配比例, $\varphi + \gamma + \eta = 1$ 。 c_i 表示虚拟机实例的处理能力。设置该参数的目的是为了控制现货类型虚拟机资源的分配比例,进一步规避现货类型资源的不稳定风险。在目标函数中需求解的未知量为 n_i^o 和 s_i^o , 求解该问题为 NP 问题, 采用启发式算法对其进行求解。在目前现有的启发式算法中, 求解方法与本文所提问题的数学模型较为一致的是粒子群算法, 因此使用粒子群算法便于对问题进行映射和求解。在本文中, 粒子群算法的每一个粒子表示一种虚拟机实例的购买方案, 总体的虚拟机资源提供方案为 Λ 。

4 基于改进粒子群算法的虚拟机资源分配方法

改进算法的思想是在基本粒子群优化算法的基础上, 重定义了算法中粒子的位置与速度, 并通过自适应惯性权重调整操作, 这样便于算法在更大的搜索空间内寻求最优解; 此外, 能进一步对算法的收敛性进行优化, 使得算法在较小的时间内收敛求得解空间内的全局最优解。

(1) 粒子重定义

$$x_i = (x_1^o, x_2^o, \dots, x_m^o, x_1^d, x_2^d, \dots, x_m^d, x_1^s, x_2^s, \dots, x_m^s) \quad (5)$$

其中, x_i^o 表示按需类型中第 i 类虚拟机资源的分配数量; x_i^d 表示预约类型中第 i 类虚拟机资源的分配数量; x_i^s 表示现货类型中第 i 类虚拟机资源的分配数量。相应地, 粒子的速度重定义为:

$$v_i = (v_1^o, v_2^o, \dots, v_m^o, v_1^d, v_2^d, \dots, v_m^d, v_1^s, v_2^s, \dots, v_m^s) \quad (6)$$

其中, v_i^o 表示按需类型中的第 i 类虚拟机资源分配数量的变化量; v_i^d 表示预约类型中第 i 类虚拟机资源分配数量的变化量; v_i^s 表示现货类型中第 i 类虚拟机资源分配数量的变化量。

改进 PSO 算法的速度和位置更新与标准 PSO 算法类似, 通过对速度和位置进行不断更新获取解空间的最优解。本文接下来对惯性权重进一步改进, 以便于优化算法的收敛性。

(2) 惯性权重调整

PSO 算法的缺点是在求解过程中出现早熟收敛现象, 出现陷入局部最优的情况。本文在算法的不同阶段, 动态

调整速度的惯性权重 ω 取值: 首先在求解的早期阶段, 通过调整惯性权重来尽可能扩大算法的解空间求解范围, 然而在算法求解的后期阶段, 通过调整惯性权重促使算法加速收敛到最优解范围。采用该方法能同时保证搜索效率和搜索精度。

改进方法的输入变量为当前种群的迭代次数 k 和当前的惯性权重 $\omega(k) \in [0.0, 1.0]$, 输出变量为当前种群的下一迭代惯性权重的值 $\omega(k+1)$:

$$\omega(k+1) = \omega(k)(1 - k/(Km)) \quad (7)$$

其中, 当前迭代次数表示为 k ; 总的迭代次数表示为 K ; 将 $\omega(0)=1$ 设置为惯性权重的初始值; 另外, m 为整数, 且 $m \in [1, 10]$, 在算法开始前选定。通过如式(7)的惯性权重调整方法, 使种群中的粒子在算法的起始阶段保持较大的惯性权重从而增加其搜索解空间的范围, 提高寻优能力; 而后随着算法的迭代, 逐渐降低惯性权重值, 使种群中的各粒子以更大的概率向当前的全局最优解位置靠拢, 因此随着算法的逐渐迭代, 粒子群会逐渐趋向最优解, 从而在算法执行的后期使整个种群的求解收敛, 以便使算法的收敛性得到保证。算法通过自适应惯性权重动态调整后的形式为:

$$v_i^{k+1} = \omega(k)v_i^k + d_1\sigma_1(x_{pb}(k) - x_i^k) + d_2\sigma_2(x_{gb}(k) - x_i^k) \quad (8)$$

算法流程如图 2 所示。

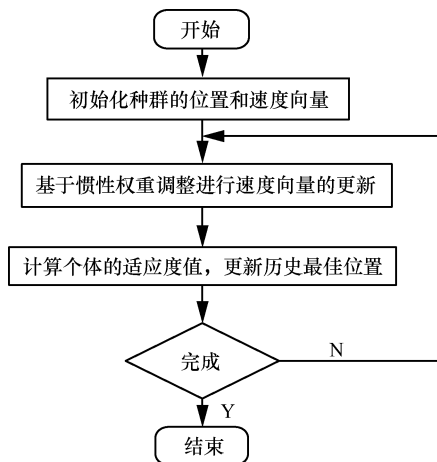


图 2 算法流程

本文所用 PSO 算法的具体步骤如下。

算法 1 基于 PSO 的虚拟机资源分配方法

步骤 1 初始化算法参数: 粒子种群规模数首先设定为 N , K 为算法执行的迭代次数。随机生成每个粒子的位置参数 x_i 和初始速度 v_i ; 对于上述初始化值的可行性进

行检查; 随后根据式(5)~式(8)计算当前各粒子位置所对应的适应度值 $cost(x_i)$, 进而可以得到对应全局最优解的初始位置 x_{gb} 和每个粒子的局部最优解位置 x_{pb} 。

步骤 2 迭代控制: 判断算法是否达到最大迭代次数, 是则转到步骤 4; 否则执行步骤 3;

步骤 3 迭代过程:

(1) 对于粒子群中的每个粒子, 计算其适应度 $cost(x_i)$, 如果优于当前局部最优解 $cost(x_i) \leq cost(x_{pb})$ 则更新当前局部最优解 $x_{pb}=x_i$; 如果优于当前全局最优解 $cost(x_{pb}) \leq cost(x_{gb})$, 则更新全局最优解 $x_{gb}=x_{pb}$;

(2) 根据粒子群算法的速度和位置更新式子, 对速度和位置进行更新;

(3) 执行步骤 2;

步骤 4 输出结果: 将虚拟机的购买方案 Λ 和当前购买方案下的适应度值 $cost(x_i)$ 进行算法输出。

5 仿真结果

本文设计了仿真实验, 以验证本文所提方法的效果, 评价指标为 SP 完成相同任务所需的云资源提供商资源总成本, 即处理相同情况下的用户请求, 减去所需的虚拟机资源总成本, 获得的 SP 总收益。实验采用 8 核 2 GHz AMD Opteron 2350 服务器和 4 核 2.4 GHz Intel Xeon X3220 系统, 所有服务器均安装 Xen 3.3 和 Linux 2.6.18 (64 bit 内核) 版本, 实验采用 Microsoft Visual Studio 2005 作为开发工具。实验中的用户任务到达的生成曲线服从正态分布。

本文中虚拟机实例采用亚马逊所提供实例的大小和标价, 见表 1~表 5。

表 1 Amazon EC2 按需类型实例价格和实例的基准测试能力

实例类型	价格/h	平均时间/s	方差/s
小型	0.115	506.1	3.2
中型	0.230	387.4	2.1
大型	0.460	113.5	1.7
超大型	0.920	40.6	1.3

表 2 小型虚拟机类型实例价格

价格区间	2013 年的概率	2014 年的概率
[0,0.041)	0.04%	0.04%
[0.041,0.050)	48.91%	57.74%
[0.050,0.062)	40.15%	41.25%
[0.062,+∞)	10.86%	0.97%



表 3 中型虚拟机类型的实例价格

价格区间	2013 年的概率	2014 年的概率
[0,0.226)	0.02%	0.03%
[0.226,0.241)	46.78%	56.64%
[0.241,0.41)	48.62%	40.19%
[0.41,+∞)	4.58%	3.11%

表 4 大型虚拟机类型的实例价格

价格区间	2013 年的概率	2014 年的概率
[0,0.36)	6.23%	5.98%
[0.36,0.47)	51.76%	47.11%
[0.47,0.68)	36.78%	46.97%
[0.68,+∞)	5.23%	0.64%

表 5 超大型虚拟机类型的实例价格

价格区间	2012 年的概率	2013 年的概率
[0,0.50)	0.75%	0.12%
[0.50,0.79)	39.85%	31.26%
[0.79,0.90)	42.16%	52.21%
[0.90,+∞)	17.24%	16.41%

图 3 为本文方法与传统单一模式定价模型的云服务提供商的收益对比。从图 3 中可以看出,本文方法在针对按需、预约和现货 3 种市场进行资源分配的基础上,通过引入风险控制因子规避现货市场的风险,有效利用了云市场的资源,提高了资源收益。目前在进行 SP 收益计算的方法中采用 FirstFit Profit 算法,基于先来先服务的原则,按

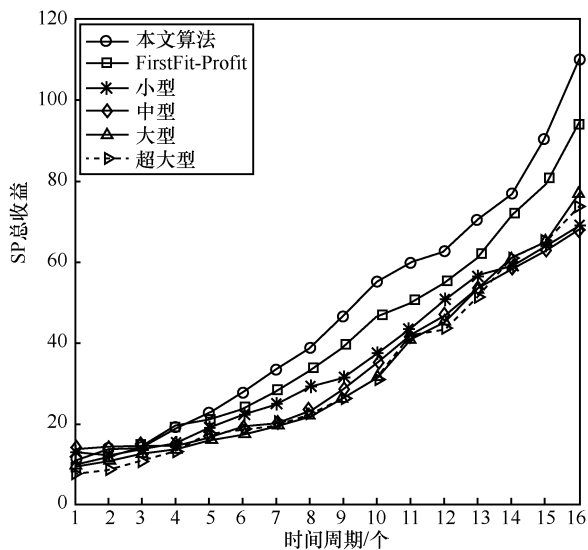


图 3 资源总收益对比

照先后顺序为用户请求分配当前最合适的云资源提供商资源,该方法与本文方法相比,会在一定程度上造成资源的浪费,无法利用现货资源较为便宜的优势,同时,不能合理分配 3 种市场的资源,该方法所产生的资源总成本较高。本文所提方法从两个方面实现了成本优化:首先,全面考虑了现货、预约、按需 3 种资源类型的购买模式,有效降低了资源成本;其次,对于具有风险的现货市场,引入风险控制因子,将 SLA 可靠性要求不高的用户请求分配到现货市场进行处理,有效实现了成本优化,提高了资源收益。

6 结束语

本文基于云服务提供商的立场,针对按需、预约和现货 3 种购买模式,提出了一种面向成本优化的虚拟机资源分配方法。本文在全面考虑 3 种市场的基础上,针对现货市场的资源风险,引入了风险控制因子,有效规避了现货市场的风险,抽象出成本优化的虚拟机资源分配模型。之后,采用改进的粒子群算法对问题进行了求解。本文所提方法在保证用户 SLA 的前提下,实现了云服务提供商的成本最小化,有效降低了资源购买成本,对增加云服务提供商的收益具有现实意义。

参考文献

- Sharma U, Shenoy P, Sahu S, *et al.* Kingfisher: cost-aware elasticity in the cloud. Proceedings of IEEE INFOCOM, Shanghai, China, April, 2011
- Sharma U, Shenoy P, Sahu S, *et al.* Kingfisher: a system for elastic cost-aware provisioning in the cloud. Dept of CS, UMSS, Tech Rep UM-CS-2010-005, 2010
- Thanakornworakij T, Nassar R and Leangsuksun C B, *et al.* An economic model for maximizing profit of a cloud service provider. Proceedings of Seventh International Conference on Availability, Reliability and Security (ARES), Prague, Aug, 2012
- Lee Y C, Wang C, Zomaya A Y, *et al.* Profit-driven service request scheduling in clouds. Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, Melbourne, May, 2010
- Cao J, Hwang K, Li K, *et al.* Optimal multiserver configuration for profit maximization in cloud computing. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1087~1096
- Irwin D E, Grit L E, Chase J S. Balancing risk and reward in a market-based task service. Proceedings of 13th IEEE International Symposium on High Performance Distributed Computing, Durham, June, 2004

(下转第 32 页)