

一种收益敏感的虚拟资源按需提供方法*

吴恒^{1,2,3}, 张文博¹, 张建华^{1,2,3}, 魏峻^{1,2}, 黄涛^{1,2}

¹(中国科学院 软件研究所 软件工程技术研发中心, 北京 100190)

²(计算机科学重点实验室(中国科学院 软件研究所), 北京 100190)

³(中国科学院大学, 北京 100049)

通讯作者: 吴恒, E-mail: wuheng09@otcaix.iscas.ac.cn

摘要: 基于虚拟机的资源重配置操作具有耗时长并伴随性能衰减的特点. 已有的 Cost-Oblivious 和 Cost-Aware 方法在面对 Internet 突变负载时存在因经常执行资源重配置操作而导致频繁 QoS 违约的不足. 提出了一种收益敏感 (benefit-aware) 的资源按需提供方法. 该方法首先考虑了 Web 应用在配置稳定期内的 QoS 获益, 并以此构建收益函数; 然后, 以收益率最大化原则作为资源重配置策略选择的依据. 实验结果表明, 该方法与 Cost-Oblivious 方法相比最多能节约 25% 的资源成本, 而与 Cost-Aware 方法相比则能有效减少 Web 应用在突变负载下的 QoS 违约次数.

关键词: 云计算; 虚拟化; 资源重配置; 收益敏感

中图法分类号: TP311 **文献标识码:** A

中文引用格式: 吴恒, 张文博, 张建华, 魏峻, 黄涛. 一种收益敏感的虚拟资源按需提供方法. 软件学报, 2013, 24(8): 1963–1980. <http://www.jos.org.cn/1000-9825/4388.htm>

英文引用格式: Wu H, Zhang WB, Zhang JH, Wei J, Huang T. Benefit-Aware on-demand provisioning approach for virtual resources. Ruan Jian Xue Bao/Journal of Software, 2013, 24(8): 1963–1980 (in Chinese). <http://www.jos.org.cn/1000-9825/4388.htm>

Benefit-Aware On-Demand Provisioning Approach for Virtual Resources

WU Heng^{1,2,3}, ZHANG Wen-Bo¹, ZHANG Jian-Hua^{1,2,3}, WEI Jun^{1,2}, HUANG Tao^{1,2}

¹(Technology Center of Software Engineering, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

²(State Key Laboratory of Computer Science (Institute of Software, The Chinese Academy of Sciences), Beijing 100190, China)

³(University of Chinese Academy of Sciences, Beijing 100049, China)

Corresponding author: WU Heng, E-mail: wuheng09@otcaix.iscas.ac.cn

Abstract: It is a challenging problem to provide QoS (quality of service) in the virtualization-based cloud computing environment. Existing efforts have addressed this challenge based on either Cost-Oblivious approaches, or Cost-Aware approaches. However, both approaches may suffer frequent QoS violations under typical flash crowd workload. For instance, both approaches ignore the benefit gained after configuration changes. In this paper, a benefit-aware approach according to the profit rate maximization principle is introduced to address this problem. Here, the benefit means the satisfaction percept of the duration that application continuously guarantees the QoS in the new configuration. Experimental results based on TPC-W benchmark show that this benefit-aware approach can save the costs of VM resources as much as 25% and can effectively reduce the QoS violations compared through a cost-aware approach.

Key words: cloud computing; virtualization; resource reconfiguration; benefit-aware

云计算是一种具有资源整合和资源按需提供能力的新兴计算模式^[1-3], 其核心问题之一是实现计算资源的

* 基金项目: 国家自然科学基金(61173004); 国家重点基础研究发展计划(973)(2009CB320704); 国家高技术研究发展计划(863)(2012AA011204); 国家科技支撑计划(2011BAH11B04)

收稿时间: 2012-05-14; 修改时间: 2012-08-20; 定稿时间: 2013-02-04

共享使用和动态配置^[2],从而达到提高资源利用率和降低资源使用成本的目的^[3].目前,Amazon EC2^[4],IBM SmartCloud^[5],OpSource Cloud^[6]等主流云计算平台纷纷采用基于虚拟机(virtual machine,简称 VM)的分时复用技术作为其资源管理的基础,并以 VM 作为最小资源单位按需提供服务.在这种计算模式下,虚拟机成为 Web 应用宿主运行支撑环境.如何应对 Web 计算环境开放性、动态性和多变性需求,保障 Web 应用性能,特别是保障关键 Web 应用,如电子商务、网上支付等性能,则成为学术界研究的热点和难点^[7-12].

首先,基于虚拟机的资源重配置操作具有耗时长和性能衰减的特点,如何提供 Web 应用的 QoS 保障具有挑战性:文献[7-9]分别基于分析模型、仿真模型和统计学方法构建 Web 应用负载与资源需求之间的映射关系,并采用“监测-调整-监测”实时资源供给策略保障 Web 应用的 QoS.然而,这些方法的有效性是建立在资源重配置操作本身对 Web 应用性能影响可以忽略的前提下^[10],并不适用于虚拟机场景.表 1 给出了 2VCPU 和 1G 内存的虚拟机配置下,TPC-W^[13]应用在 Shopping 模式和 100 并发下的执行 VM 在线迁移的实验结果,可见,Web 应用 JEE 服务器组件和 DB 服务器组件执行在线迁移操作时性能衰减最多可达 300%,且完成在线迁移操作的耗时也从 30s 到 108s 不等.因此,Jung^[11]明确阐述了现有云计算平台只能提供 Web 应用可用性保障,而无法为 Web 应用在资源重配置操作过程中提供 QoS 保障的观点^[11].本文将完成资源重配置操作耗时及其性能衰减简称为转换成本(transition cost).

Table 1 Performance impacts of VM live migration on Web components

表 1 VM 在线迁移对 Web 应用组件性能影响

Component	Response time (ms)	Response time during action (ms)	Time-Consuming (s)
WebServer	324	652	30
JEEServer	492	2 442	62
DBServer	611	2 740	108

其次,Gartner 报告明确阐述了高质、低廉的资源按需服务是未来云计算平台成败的关键要素之一^[12].然而如表 2 所示:一方面,虚拟机类型多样性的特点(例如,Amazon EC2 云计算平台提供了 Small,High-CPU 等多种类型的虚拟机),使得满足 Web 应用 QoS(quality of service)需求的虚拟机组合具有多样性;另一方面,虚拟机计算能力和价格具有非线性关系,这使得选择出经济的虚拟机组合进行资源供给面临挑战.本文将虚拟机资源的租用价格称为资源成本(resource cost).

Table 2 VM price model

表 2 虚拟机价格模型

Type	Amazon EC2 cloud platform		OpSource cloud platform	
	Configuration	Cost/hr (\$)	Configuration	Cost/hr (\$)
Small	1ECU, 1.7GB RAM, 320GB disk	0.05	1core, 2GB RAM, 320GB disk	0.138
Large	4ECU, 7.5GB RAM, 850GB disk	0.20	4core, 8GB RAM, 850GB disk	0.615
XLarge	8ECU, 15GB RAM, 1690GB disk	0.40	8core, 16GB RAM, 1690GB disk	1.227
High-CPU	5ECU, 1.7GB RAM, 350GB disk	0.04	5core, 2GB RAM, 350GB disk	0.355

已有研究工作提出了一种基于竞价-拍卖的资源动态分配方法^[14].该方法将用户竞价高低作为 VM 资源分配的依据,从而达到云计算平台收益最大化的目标.基于竞价-拍卖的方法未能考虑到转换成本和资源成本两个要素,因此本文称其为 Cost-Oblivious 方法.文献[15]提出了一种基于多路复用原理的资源动态分配方法.该方法针对不同 Web 应用峰值负载在时间上具有错开性的特点,将多个 Web 应用及其 VM 部署在同一物理机上,从而达到减少资源成本的目标.文献[11]提出了一种基于效用最大化的资源动态分配方法,该方法针对 VM 重配置操作具有转换成本的特点构建惩罚函数和奖励函数,并以奖励-惩罚最大化作为资源动态分配的判定标准.文献[16]提出了一种基于线性规划的资源动态分配方法.该方法的核心是:当 Web 应用资源需求变化时,优先选择当前负载下转换成本最小的 VM 重配置操作,并在此基础上选择满足 QoS 需求且资源成本最小/最大的 VM 资源进行供给/回收.本文将考虑了转换成本或资源成本的方法称为 Cost-Aware 方法.

然而,上述方法在负载突变(flash crowd)模式下均存在因频繁执行资源重配置操作而导致 Web 应用 QoS 违约的不足,见本文第 4 节.针对已有方法的不足,本文提出了一种收益敏感(benefit-aware)的资源按需提供方法.

该方法首先将 Web 应用在新配置稳定周期内的 QoS 保障及其保障程度描述为收益,然后基于归约法原则将资源成本和转换成本约束下的资源重配置问题转化为经济学收益问题.决策者投资问题是一个典型的非线性规划问题,因此,本文设计了基于收益/成本最大化的资源容量规划算法.该算法是以用户收益期望为目标控制虚拟化资源的供给,从而达到保障 QoS 约束和最小化资源成本的目标.基于 TPC-W 测试基准的实验结果显示,本方法与 Cost-Oblivious 方法相比最多能节约 25%的资源成本,而与 Cost-Aware 方法相比则能有效减少 Web 应用在负载突变模式下的 QoS 违约次数.

本文第 1 节是研究对象描述与分析.第 2 节介绍本文工作的体系结构及组成要素描述.第 3 节详细阐述收益敏感的按需提供方法原理及其关键技术.第 4 节通过对比实验验证本方法的有效性.第 5 节是相关工作介绍.第 6 节总结本文并提出未来工作.

1 研究对象描述与分析

虚拟机作为封装计算资源的载体,具有安全隔离、高可靠和配置简单灵活等特点,已成为主流云计算平台按需提供的最小资源单位.表 2 给出了主流云计算平台的虚拟机价格模型,可知虚拟机计算能力(本文特指虚拟机配置中 CPU 个数和内存大小)与价格之间是一种非线性关系.例如,同是 Amazon EC2 云计算平台的虚拟机配置:对于 Small 和 Large 两种虚拟机配置,后者的计算能力供给是前者的 4 倍多,但承租价格却正好是前者的 4 倍;而对于 Small 和 High-CPU 两种虚拟机配置,虚拟机计算能力和承租价格的关系却正好相反.这种虚拟机价格的模型非线性特征,使得实现资源成本最小化的按需提供面临挑战.

同时,虚拟机作为 Web 应用宿主运行支撑环境,其运行时资源动态分配是满足 Web 应用资源需求动态变化、保障 Web 应用 QoS 的一种有效手段.图 1 给出了 3 类基于虚拟机的资源重配置操作:

- (1) VM 容量调整(local resizing):通过运行时修改 VCPU 的个数和内存的大小来达到改变资源供给总量的一种纵向资源调整技术,其本质是改变虚拟机实例的资源容量,又可细分为增加 VM 资源容量和减少 VM 资源容量两个操作.文献[17]详细描述了 VM 容量调整的步骤,并阐述了 VM 容量调整可被认为是“零耗时无性能开销”的观点.
- (2) VM 复制(VM replica):通过运行时增删 VM 实例数来达到改变资源供给总量的一种横向资源调整技术,可细分为新增 VM 实例和删除 VM 实例两个操作.文献[17]认为,VM 实例生命周期状态管理具有耗时性,这一特性使得传统实时资源供给策略在实际应用过程中生效具有滞后性,从而导致 Web 应用性能衰减.
- (3) VM 在线迁移(live VM migration):通过运行时变更虚拟机内存的物理位置而又不影响虚拟机服务的一种资源调整技术.它强调的是 VM 实例位置的变迁而不是资源总量的改变.文献[18]明确阐述了内存状态是影响在线迁移耗时和引起 Web 应用性能衰减的关键要素,本文中的表 1 也给出了在线迁移具有耗时长和性能衰减的实验结果.

因此,鉴于基于虚拟机的资源重配置操作具有耗时和性能衰减的特点,运行时的虚拟化资源动态提供应考虑转换成本要素.

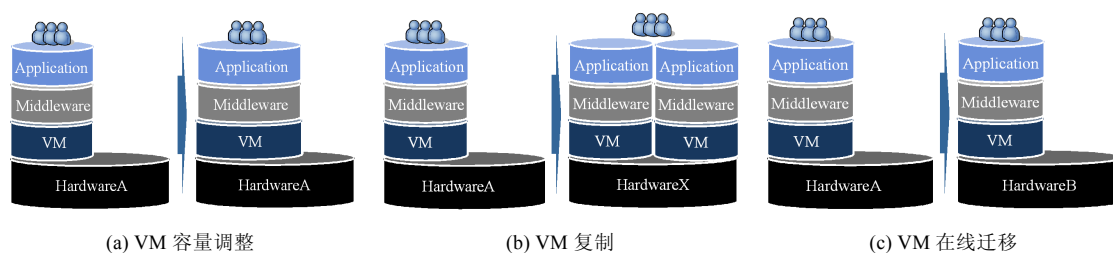


Fig.1 VM reconfiguration actions

图 1 基于虚拟机的资源重配置操作

综上所述,本文面临的是资源成本和转换成本约束下的虚拟机资源重配置策略选择问题(简称为策略选择问题),即,当 Web 应用负载变化导致 Web 应用资源需求变化时,应分配或回收多少虚拟机资源,以及通过何种资源重配置操作进行资源分配或回收操作,才能同时满足资源成本最小化和 QoS 保障的目标.一方面,由本节分析可知,虚拟机在资源重配置过程中具有成本属性^[19].它包括资源成本和转换成本两个要素.另一方面,合适的资源供给量是兼顾成本最小化和 QoS 保障的一个关键要素.文献[20]中指出,计算能力不足是最大社交网站 Friendster 衰落的重要原因之一.图 2 给出了 Friendster 计算能力供给和需求关系.由此可知,Web 应用的 QoS 保障隐含着潜在的收益.因此,基于虚拟机的资源重配置具有收益属性^[21].

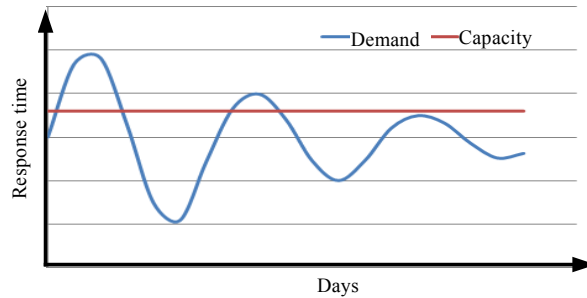


Fig.2 Capacity provisioning and user demand of FreindSter

图 2 FreindSter 容量供给和用户需求关系

策略选择问题本质上是虚拟机配置和资源重配置操作组合方案的选择问题.由于对于具体的虚拟机配置和资源重配置操作组合 i ,选择其成本 $Cost_i$ 和收益 $Benefit_i$ 是计算可知的,根据收益率最大化原则^[22],其最佳选择策略为

$$Selected(i) = \frac{Benefit_i - Cost_i}{Cost_i} \quad (1)$$

2 体系结构

本文基于 Xen 虚拟化技术构建了具有虚拟机资源按需提供能力的 Balloon 平台.该平台考虑了虚拟机资源成本和资源重配置操作的转换成本两个要素.

图 3 给出了 Balloon 平台的体系结构,主要包括 Web 应用性能模型、Web 应用性能衰减模型、虚拟机价格模型和资源调整引擎这 4 个模块.其中,

- Web 应用性能模型建立了 Web 应用负载和响应时间之间的映射关系.它是判定 Web 应用 QoS 约束是否满足的基础.
- Web 应用性能衰减模型建立了某负载下资源重配置操作和 Web 应用性能变化之间的映射关系.它是计算资源重配置操作转换成本的一个关键要素.
- 虚拟机价格模型描述了虚拟机配置及其承租价格之间的映射关系,见表 2.它是待供给的虚拟机资源成本计算的依据.
- 资源调整引擎是实现收益敏感的资源按需提供方法的关键模块,详见第 3 节.

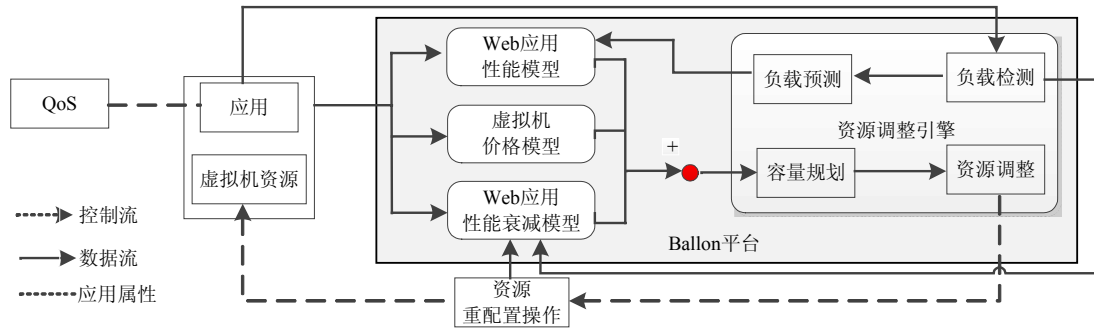


Fig.3 Architecture of Balloon platform
图3 Balloon 平台体系结构

2.1 基于排队论的Web应用性能建模

Web应用性能建模采用文献[22,23]中基于会话的方式描述,会话是多次事务的集合,例如,TPC-W^[13]包含14种基本事务类型.

图4给出了基于简单反馈回路(feedback circuits)的排队网性能模型^[24],本文称为 FQM(feedback based queuing model)模型,其中,Web应用事务请求的到达率为 λ_{income} ,事务请求经过 WebServer, JEE Server 和 DBServer 处理后有概率 P_{exit} 结束会话的生命周期.因此,每次进入 Web 应用的事务请求实际到达率为 λ'_{income} :

$$\lambda'_{income} = \lambda_{income} + (1 - P_{exit}) \times \lambda'_{income}$$

即

$$\lambda'_{income} = \frac{\lambda_{income}}{P_{exit}} \tag{2}$$

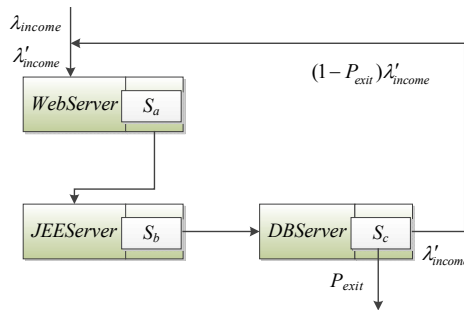


Fig.4 Performance modeling of TPC-W
图4 TPC-W 性能建模

根据李特尔定律可知,系统资源利用率 $\mu = \lambda S$,其中, S 为系统服务时间,则图4中 WebServer, JEE Server 和 DBServer 的资源利用率可分别表示为

$$\mu_{WebServer} = \lambda'_{income} \times S_a \tag{3}$$

$$\mu_{JEE Server} = \lambda'_{income} \times S_b \tag{4}$$

$$\mu_{DBServer} = (1 - P_{exit}) \lambda'_{income} \times S_c \tag{5}$$

其中, S_a, S_b, S_c 分别为 WebServer, JEE Server 和 DBServer 的服务时间.服务时间是由每个事务 t_i 的平均服务时间 (mean service time, 简称 MST) 组成的向量 $S = (S_1, S_2, \dots, S_M)$, 故 Web 应用在负载到达率 λ_{income} 与响应时间的关系为

$$RT_{FQM}(\lambda_{income}) = \frac{S_a}{1 - P_{exit} - \lambda_{income} S_a} + \frac{S_b}{1 - P_{exit} - \lambda_{income} S_b} + \frac{P_{exit} S_c}{1 - P_{exit} - \lambda_{income} P_{exit} S_c} \quad (6)$$

2.2 基于离线测试的Web应用性能衰减建模

本文采用离线测试的方法构建 Web 应用性能衰减表(performance degradation table,简称 PDT).该方法针对由 Web 应用组件 *comp*、虚拟机资源类型 *type* 和典型的工作负载 *w* 的组合分别执行资源重配置操作 *operate*,其中, $comp \in \{WebServer, JEE Server, DBServer\}$, $type \in \{Small, Large, XLarge\}$, 工作负载 *w* 为 TPC-W 的种典型负载模式 $w \in \{Browsing, Shopping, Ordering\}$.

在实验过程中,分别执行增加 VM 资源容量、减少 VM 资源容量、新增 VM 实例、删除 VM 实例和 VM 在线迁移这 5 种 *operate* 操作,记录响应时间 $RT(comp, type, w)$ 和执行资源重配置操作 *op* 过程中的响应时间 $RT'(comp, type, w, op)$, 计算并记录响应时间衰减值 $\Delta RT(w) = RT'(comp, type, w, op) - RT(comp, type, w)$.

表 3 为 TPC-W 基础测试 JEE Server 进行在线迁移操作性能衰减表.

Table 3 Performance impacts table for JEE server during VM migration

表 3 VM 在线迁移过程中 JEE Server 组件的性能衰减表

Configuration	1core 2.4GHz			2core 2.4GHz			4core 2.4GHz		
Workload (request/s)	100	300	500	100	300	500	100	300	500
ΔRT (ms)	253	498	2 100	198	471	1 850	148	448	1 400

在实际运行过程中,以组合条件 Web 应用组件 *comp*、虚拟机资源类型 *type*、当前工作负载 *w* 和资源重配置操作 *operate* 作为输入,从性能衰减表(PDT)中查找与当前负载 *w* 最相近的负载 *w'* 记录,并根据此记录值计算当前 Web 应用执行资源重配置操作 *op* 的响应时间:

$$RT_{operate} = RT(comp, type, w, operate) + \Delta RT(w') \quad (7)$$

3 一种收益敏感的按需提供方法

如图 3 所示,资源调整引擎由负载监测、负载预测、容量规划和资源调整这 4 个组件构成.负载监测组件选用具有跨平台、高性能和可扩展性的 Hyperic HQ^[25].负载预测组件基于统计学方法,选用开源软件 common-math^[26].容量规划组件是资源调整引擎的核心.它是资源重配置的决策者.资源调整组件是资源重配置的具体执行者.

由前文的分析可知,资源重配置(虚拟机配置和资源重配置操作)最佳选择即当收益/成本最大化约束满足时.由于成本与资源重配置的转换周期和稳定周期有关,而收益与资源重配置的稳定周期有关,因此,本节首先给出了资源重配置整体周期、转换周期和稳定周期关系的计算方法,如图 5 所示;然后详细描述了收益函数和成本代价函数的构建方法;最后给出了收益/成本最大化的非线性求解算法.

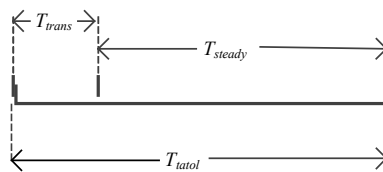


Fig.5 Relationship among T_{total} , T_{trans} and T_{steady}

图 5 T_{total} , T_{trans} 和 T_{steady} 之间的关系

3.1 资源重配置整体周期、转换周期和稳定周期计算

3.1.1 资源重配置整体周期的预测

资源重配置整体周期是指 Web 应用 *s* 在某时刻 *t* 监测的负载为 w_t^s 、Web 应用在负载区间 $[w_t^s - b, w_t^s + b]$ 持续的时间.由于 Web 应用负载变化^[27]具有相关性、突变性、随机性、自相似性,使得 Web 应用在负载区间

$[w_t^s - b, w_t^s + b]$ 持续的时间变化可被看作是平稳的随机序列.而 ARMA 模型^[28]是一种描述平稳随机序列的最常用的模型,因此,本文采用 ARMA 模型来预测资源重配置整体周期 T_{total} .

用 E_j^p 表示 Web 应用第 j 次资源重配置整体周期的数学期望值,用 E_j^m 表示 Web 应用第 j 次资源重配置整体周期的实测值,则根据 k 次实测的 Web 应用资源重配置整体周期,基于 ARMA 模型计算 Web 应用第 $j+1$ 次整体周期的数学期望公式为

$$E_{j+1}^p = ARMA(1, k-1) = \gamma \times E_j^m + \frac{(1-\gamma)}{k-1} \sum_{i=1}^{i=k-1} E_{j-i}^m \quad (8)$$

其中, γ 取值越大,说明 Web 应用当前资源重配置整体周期的实测值对 Web 应用下一个资源重配置整体周期的估算结果影响越大. γ 参数的确定采用具有快速收敛特点的自适应滤波(adaptive filter)算法^[29],其中,前 k 次资源重配置整体周期的实测值和数学期望值之间的误差为

$$\varepsilon_j = \gamma \times |E_j^p - E_j^m| + \frac{1-\lambda}{k-1} \times \sum_{i=1}^{i=k-1} \varepsilon_{j-i} \quad (9)$$

$$\gamma = 1 - \frac{\varepsilon_j}{\max_{i=1,2,\dots,k} \varepsilon_i} \quad (10)$$

本文规定 γ 的初始值为 0.5,表示在计算 Web 应用第 $j+1$ 次整体周期的数学期望时,第 $j+1$ 次资源重配置整体周期的实测值和历史 $k-1$ 资源重配置整体周期实测值的权重是一样的.

3.1.2 转换周期的计算

根据文献[17,18]对 VM 重配置操作的耗时分析以及本文第 1 节对 VM 重配置操作的描述,可得转换周期计算公式:

$$T_{trans} = f \times \frac{mem}{bw} + \frac{disk}{bw} + b \quad (11)$$

其中, $disk$ 表示 VM 模板的大小; bw 表示当前可用带宽; b 是一个常量,例如表示 VM 实例启动时间; mem 表示虚拟机内存大小; bw 表示可用网络带宽; f 表示一个衰减因子,用于描述在线迁移过程中源 VM 内存页更新率对在线迁移时间的影响.

3.1.3 稳定周期的计算

根据图 5 资源重配置整体周期、转换周期和稳定周期的关系,可计算得出 Web 应用的稳定周期:

$$T_{steady} = T_{total} - T_{trans} \quad (12)$$

3.2 收益函数

收益函数用于描述用户对 Web 应用在新配置下稳定周期及在该过程中 QoS 保障的满意度,以有效防止应用在突变负载(flash crowd)下频繁违反 QoS 约束的不足,本文将收益函数描述为 $Benefit = sf \times T_{steady}$.其中, T_{steady} 表示稳定周期, sf 表示满意度因子.

文献[27]认为,用户的满意度与负载成正比,即负载越大,用户对当前 Web 应用性能保障的满意度越高;本文则以用户关于资源重配置整体周期的数学期望 $E(T)$ 为目标,构建分段式满意度函数:

$$sf = \begin{cases} \left(\frac{T_{steady}}{E(T)}\right)^m \times b, & T_{steady} < E(T), m > 0 \\ b, & T_{steady} \geq E(T) \end{cases} \quad (13)$$

其中,

- b 表示满意度因子常数,本系统默认实现值 $b=100$;
- m 由用户指定,用于描述满意度因子变化趋势,本文设置 $m=1$,表示预测的 Web 应用稳定周期与期望稳定周期比值与满意度成线性关系,即稳定周期预测值越接近稳定周期期望值,用户满意度越高;
- $E(T)$ 由用户指定,例如,本文希望至少在 95% 的时间内要保障 Web 应用的 QoS,而经过离线测试发现,资

源重配置操作最长耗时约为 20s,因此设定 $E(T)=380s$.

3.3 成本函数

根据本文第 1 节的分析,资源重配置成本 C_{total} 可划分为转换周期所需考虑的转换成本 C_{trans} 和稳定的周期所需考虑的资源成本 C_{steady} 两个方面,即 $C_{total}=C_{trans}+C_{steady}$.

3.3.1 转换周期成本

转换周期是指完成资源重配置操作的时间.该过程 Web 应用 QoS 保障具有不确定性,而资源重配置操作的转换成本是产生不确定性的主要原因.因此, $C_{trans}=pf \times T_{trans}$,其中, T_{steady} 表示转换周期, pf 表示惩罚因子.文献[30]认为,在资源重配置过程中,如果 Web 应用 QoS 无法保障,则其惩罚因子取值与负载成正比,即,此时负载越大,其惩罚因子的取值就越大.本文借鉴文献[30]的思想,构建分段式惩罚函数如下:

$$pf = \begin{cases} c, & RT(trans) > QoS, k > 0, W_{curr} < 100 \\ \left(\frac{W_{curr}}{100}\right)^k \times c, & RT(trans) > QoS, k > 0, W_{curr} \geq 100 \\ 0, & RT(trans) \leq QoS \end{cases} \quad (14)$$

其中,

- W_{curr} 表示当前负载.
- $RT(trans)$ 表示在当前负载 W_{curr} 下执行资源重配置操作的响应时间,可通过本文第 2.2 节查表所得.
- k 由用户指定,用于描述惩罚因子变化趋势,本文设置 $k=1$,表示惩罚因子取值与负载大小是线性关系,即当服务质量违约时,当前负载越大,惩罚因子就越大.
- c 表示惩罚因子常数,由于资源重配置调整的一种极端情况为 $T_{total}=T_{trans}$,即 Web 应用在持续执行资源重配置操作,而本文约定此时 Web 应用收益等于 Web 应用转换周期成本.由于本文实验的最大负载为 2 200 请求/秒,故根据公式(14),取值为 $c=5$.

3.3.2 稳定周期成本

稳定周期是指虚拟机资源为 Web 应用服务并满足 QoS 约束的周期,因此 $C_{steady}=f \times |\Delta price| \times T_{steady}$.其中, T_{steady} 表示稳定周期; $|\Delta price|$ 表示执行资源重配置操作后,资源成本的变化; f 表示单位资源成本支出的成本代价因子,由用户指定,由于根据公式(11.a)可知,Web 应用执行容量调整操作的转换成本为 0,即虚拟化资源重配置操作存在一种情况满足约束 $T_{total}=T_{steady}$.而本文约定此时 Web 应用收益等于 Web 应用稳定周期成本,又由于 $|\Delta price|$ 最大取值约为 0.5,如虚拟机从表 4 中 Small 配置转换成 Large 配置.因此,本文设置 $f=200$.

3.4 规划算法

规划过程就是选择出满足 QoS 保障和资源成本最小化目标的虚拟机配置和资源重配置操作的最优组合.根据归约法原则,其最佳组合即当公式(1)约束被满足时.本文约定资源重配置操作包括增加 VM 资源容量、减少 VM 资源容量、新增 VM 实例、删除 VM 实例、VM 在线迁移、增加 VM 资源容量和 VM 在线迁移组合(增加 VM 资源容量的前提是当前物理机需满足具有足够资源的约束,当约束不满足时,可采用先将 VM 迁移到空闲的物理机再进行 VM 容量调整的策略)共 6 种,用符号 $OpSet$ 表示资源重配置操作集合.同时约定虚拟机配置与表 1 中 Amazon EC2 云计算平台的一致,包括 Small, Large, XLarge 和 High-CPU 这 4 种,用符号 $VMConfSet$ 表示虚拟机配置集合.

同时,本文对如图 6 所示的算法做如下约定:

- 稳定周期期望 $E(T)$ 为常数,它是在 3.1.1 节离线测试的获得 $(E_1^p, E_2^p, \dots, E_n^p)$ 序列的基础上,选择最小方差作为取值的,即 $E(T) = \min \left(E_i^p - \frac{1}{n} \sum_{j=1}^n E_j^p \right)^2 (1 \leq i \leq n)$;
- \minPrice 函数表示最小资源成本, \maxPrice 函数表示最大资源成本;
- \minCost 函数表示最小转换成本, \maxCost 函数表示最大转换成本.

Algorithm 1. Planning algorithm based on BRM (benefit rate maximization)
Input: $OpSet$; $VMConfSet$; QoS ; $VMConfPrice$; w ; $E(T)$.
Output: Op ; $Op \in OpSet$; VM ; $VM \in VMConfSet$.

1. Predict the workload w' and calculate out the response time $rt=RT^{FQM}(w')$
2. **IF** $w > w+b$
3. //the predict total period is great than $E(T)$
4. **IF** $T_{total}^P > E(T)$
5. $VM = \minPrice(VMConfSet)$;
6. $Op = \minCost(VM, OpSet)$;
7. **ENDIF**
8. **ELSE** //the predict total period is less than $E(T)$
9. $Max=0$;
10. **FOR** $\sum T_{total,i} > E(T)$
11. **IF** $(benefit-cost)/cost > Max$
12. $VM = selected(VM)$;
13. $Op = selected(Op)$; $Max = (benefit-cost)/cost$
14. **ENDIF**
15. **ENDFOR**
16. **ENDIF**
17. **ENDIF**
18. **ELSE IF** $w < w-b$
19. //the predict total period is great than $E(T)$
20. **IF** $T_{total}^P > E(T)$
21. $VM = \maxPrice(VMConfSet)$;
22. $Op = \maxCost(VM, OpSet)$;
23. **ENDIF**
24. **ELSE** //the predict total period is less than $E(T)$
25. $Op = \minCost(OpSet)$;
26. $VM = \maxPrice(Op, VMConfSet)$;
27. **ENDIF**
28. **ENDIF**
29. **ELSE**
30. Wait the total period timeout and go to Line 2;
31. **ENDELSE**

Fig.6 BRM-Based planning algorithm

图 6 基于 BRM 的规划算法

当预测的 Web 应用稳定周期 T_{total} 大于用户关于资源重配置整体周期的数学期望 $E(T)$ 时,收益函数 Benefit 取得最大值,最佳资源重配置选择即当成本代价函数 C_{total} 最小化时.本文采用贪心算法求解,选择满足 Web 应用资源需求的成本最小(最大)的虚拟机资源进行供给(回收),供给(回收)过程选择转换成本最小的资源重配置操作(见第 4 行~第 7 行、第 21 行~第 24 行).

当预测的 Web 应用稳定周期 T_{total} 小于用户关于资源重配置整体周期的数学期望 $E(T)$ 时,资源分配过程如第 8 行~第 17 行所示,假设当前预测的 Web 应用稳定周期为 $T_{total,1}$,并基于此预测下一个 Web 应用稳定周期为 $T_{total,2}$,直到约束 $\sum T_{total,i} > E(T)$ 满足为止.根据第 3.1.1 节的描述,此时 Web 应用在周期 $T_{total,i}$ 的负载上限为 $W = W_{curr} + b \times i$;资源回收过程采用贪心算法(见第 25 行~第 28 行),首先选择转换成本最小的资源重配置操作,在此基础上,选择满足 Web 应用资源需求的成本最大的虚拟机资源进行回收.

4 实验设置与结果分析

实验包括 5 个部分:(1) 实验环境介绍;(2) 基于 FQM 模型的 Web 应用性能剖析准确性实验;(3) 基于 ARAM 模型的资源重配置整体周期预测准确性实验;(4) Benefit-Aware 方法和 Cost-Oblivious 方法在满足资源成本最小化方面的对比实验;(5) Benefit-Aware 方法和 Cost-Aware 方法在 QoS 保障方面的对比实验.

4.1 实验环境

如图 7 所示,实验环境由 6 台刀片机组成,每台刀片的配置都是 24-core 2.4GHz Intel Xeon CPU 和 24G 内存:

其中 1 台刀片作为存储服务器,用于存储如表 4 给出的虚拟机配置模板;1 台刀片作为负载发生器,模拟用户进行压力测试;4 台刀片安装虚拟化软件 Xen Server3.3,并构建虚拟机资源池。

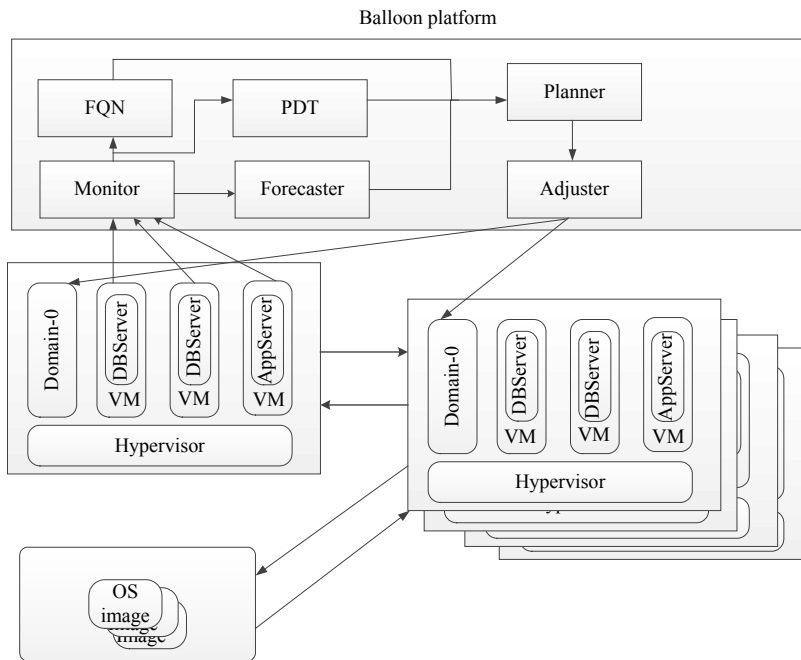


Fig.7 Testbed architecture

图 7 实验环境架构

Table 4 VM price model

表 4 虚拟机价格模型

Balloon cloud platform			
Type	Configuration	Cost/min (\$)	\$/CPU (\$)
Small	1core, 1GB RAM	0.11	0.11
Medium	2core, 2GB RAM	0.17	0.085
Large	4core, 4GB RAM	0.53	0.133

Web 应用采用 TPC-W 基准测试^[13],前端 Web 容器组件选用 HttpServer 2.0,中间 JEE 服务器组件选用 tomcat 6.0.23,后端数据库组件选用 IBM DB2 9.2,数据库采用默认设置,即 10 000 件商品和 1 440 000 个用户.工作负载采用典型的 Time-of-Day 和 Flash Crowd 两种模式:Time-of-Day 负载模式来源于 1998 年法国世界杯网站^[31];Flash Crowd 负载模式来自于惠普公司一台客户服务器^[32].

4.2 基于 FQM 模型 Web 应用性能剖析的准确性

基于 FQM 模型 Web 应用性能剖析的目的分为两个方面:(1) FQM 模型性能剖析的准确性与负载类型的关系;(2) FQM 模型性能剖析的准确性与多核的关系.因此,在负载类型方面,本文选用 Time-of-Day 和 Flash Crowd 两种负载模式;在多核测试方面,本文选用表 4 所示的 Small 和 Large 两种虚拟机配置.

图 8(a)、图 8(b)分别是 Small 和 Large 两种虚拟机配置在 Time-of-Day 负载模式下运行 24 分钟的测试结果,工作负载描述了由 50 请求/秒渐进式地增加到 240 请求/秒的过程.性能剖析结果显示,最大误差率分别是 6% 和 12%.图 8(c)、图 8(d)分别是 Small 和 Large 两种虚拟机配置在 Flash Crowd 负载模式下运行 24 分钟的测试结果,工作负载描述了从 200 请求/秒激增到 500 请求/秒,然后又稳定在 200 请求/秒的过程,最大误差率分别是 12%和 21%.

综上所述:FQM 模型能够很好地适用于各负载类型,尽管 Small 虚拟机配置在 Flash Crowd 负载模式下最大误差可达 12%,但整体平均误差却仅有 7%左右;FQM 模型不能很好地适用于多核的环境,这是因为本文在基于 FQM 的性能建模中未能考虑多核的影响.由于在本文实验环境中虚拟机配置中最多核数仅有 4 个,因此我们认为,选用 FQM 模型进行 Web 应用性能剖析是可接受的.

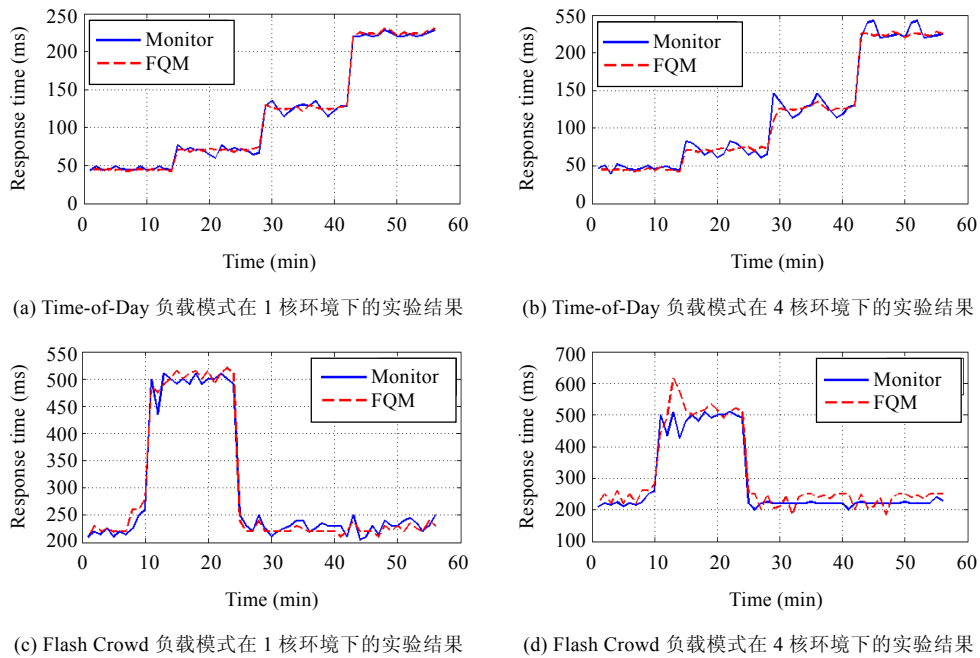


Fig.8 Prediction accuracy of FQM model under different workloads and different CPU cores

图 8 不同负载和 CPU 核数约束下 FQM 模型的预测准确性

4.3 基于ARAM模型资源重配置整体周期预测的准确性

本文以运行在 Small 虚拟机配置上的 JEE 服务器组件为例,以 30 分钟训练数据为初始输入,分别测试 ARAM 模型在 Time-of-Day 和 Flash Crowd 负载模式下资源重配置整体周期预测的准确性.

如图 9(a)所示,ARAM 模型在 Time-of-Day 负载模式下,资源重配置整体周期的最大预测误差为 10%;而在 Flash Crowd 负载模式下,ARAM 模型预测的最大误差超过 40%,分别出现在第 5 个周期和第 15 个周期,其原因是负载突变.但从第 6 个周期到第 14 个周期,ARAM 模型预测结果与实测值基本吻合,这说明 ARAM 模型具有快速收敛性,如图 9(b)所示.

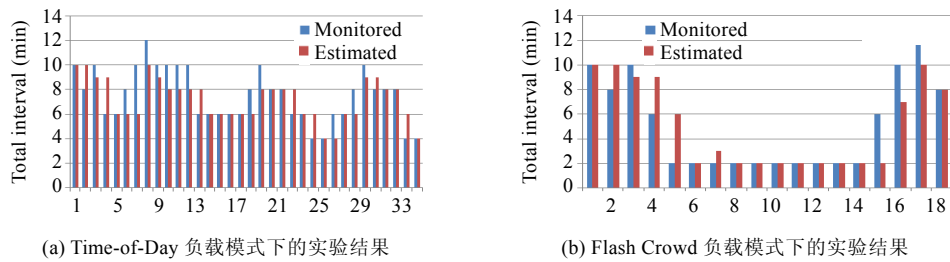


Fig.9 ARAM prediction accuracy for different workloads

图 9 不同负载模式下 ARAM 模型预测的准确性

4.4 Benefit-Aware方法和Cost-Oblivious方法对比实验

本文以 JEE 服务器组件为例,分别在 Time-of-Day 和 Flash Crowd 两种典型的负载模式下测试 Benefit-Aware(BA)方法和 Cost-Oblivious(CO)方法在满足资源成本最小化方面的对比实验.假设:

- (1) 虚拟机配置中 CPU 核数与其计算能力成正比,并已知 Small 配置的虚拟机最多承载 550 请求/秒,则 Large 配置的虚拟机最多可承载 2 200 请求/秒;
- (2) 每个资源重配置整体周期都足够长;
- (3) 约定图 10 中 L 表示 Large 配置的虚拟机,S 表示 Small 配置的虚拟机.

图 10(a)是 JEE 服务器组件在 Time-of-Day 负载模式下,负载从 550 请求/秒渐进式地增加到 2 200 请求/秒的实验结果.结果显示,与 Cost-Oblivious(CO)方法相比,Benefit-Aware(BA)方法能够有效节约资源成本 10%.其原因是:在 t_3 周期,虽然 Large 配置的计算能力正好是 Small 配置虚拟机的 4 倍,但 Cost-Oblivious(CO)方法无法感知到 1 个 Large 配置的虚拟机成本和 4 个 Small 配置的虚拟机成本之间的差异性.

图 10(b)是 JEE 服务器组件在 Flash Crowd 负载模式下,负载从 300 请求/秒突然增加到 2 200 请求/秒,然后稳定在 300 请求/秒的实验结果.结果显示,与 Cost-Oblivious(CO)方法相比,Benefit-Aware(BA)方法能够有效节约资源成本 25%.其原因是从 t_2 周期起,由于 Web 应用 QoS 约束未被违反,因此 Cost-Oblivious 方法不会进行资源回收操作.

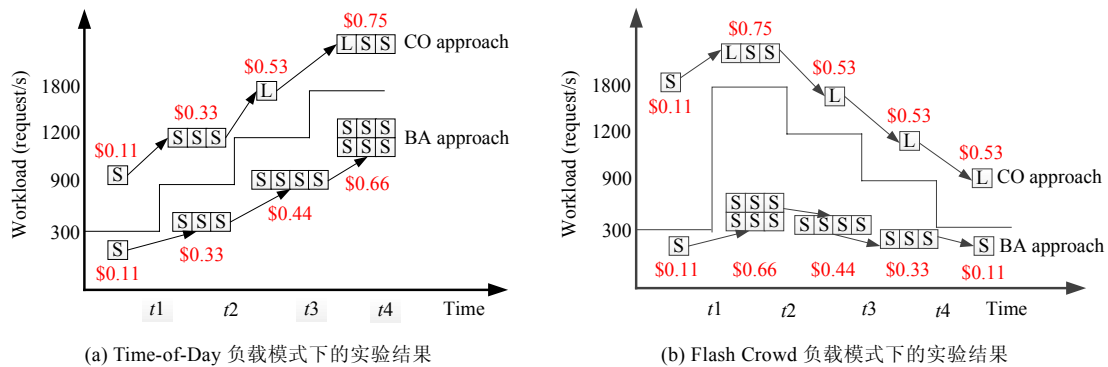


Fig.10 Comparison between the Benefit-Aware approach and the Cost-Oblivious approach

图 10 Benefit-Aware 方法和 Cost-Oblivious 方法对比

4.5 Benefit-Aware方法和Cost-Aware方法对比实验

4.5.1 Benefit-Aware 方法和 Infrastructure-Cost-Aware 方法对比实验

本文以 JEE 服务器组件为例,分别在 Time-of-Day 和 Flash Crowd 两种典型的负载模式下测试 Benefit-Aware(BA)方法和 Infrastructure-Cost-Aware(ICA)方法在满足兼顾资源成本最小化和 QoS 保障方面的对比实验.我们做以下假设:

- (1) 虚拟机配置中 CPU 核数与其计算能力成正比,并已知 Small 配置的虚拟机最多承载 550 请求/秒,则 Large 配置的虚拟机最多可承载 2 200 请求/秒.
- (2) 在 Time-of-Day 负载模式下,每个资源重配置整体周期都足够长;而在 Flash Crowd 负载模式下则正好相反.
- (3) 约定在图 11 中,L 表示 Large 配置的虚拟机,S 表示 Small 配置的虚拟机.

图 11(a)、图 11(b)分别是 JEE 服务器组件在 Time-of-Day 负载模式下(负载从 550 请求/秒渐进式地增加到 2 200 请求/秒)资源成本支出和 QoS 保障的实验结果.

结果显示,与 Infrastructure-Cost-Aware(ICA)方法相比,Benefit-Aware(BA)方法能够在保障资源成本最小化的基础上有效减少 QoS 约束违反的次数,其原因是 Infrastructure-Cost-Aware(ICA)方法采用的是实时资源供给

策略.

图 11(c)、图 11(d)分别是 JEE 服务器组件在 Flash Crowd 负载模式下(负载从 300 请求/秒突然增加到 2 200 请求/秒,然后稳定在 300 请求/秒)资源成本支出和 QoS 保障的实验结果.结果显示,与 Infrastructure-Cost-Aware(ICA)方法相比,虽然多支付了约 8%的资源成本,但能够有效减少 QoS 约束违反的次数.其原因是:在 t_2 周期和 t_4 周期,Benefit-Aware(BA)方法能够感知到该资源重配置整体周期具有易变性,故而采用转换成成本小的资源重配置操作(基于 VM 容量调整将 Small 配置虚拟机扩容成 Large 配置的虚拟机).

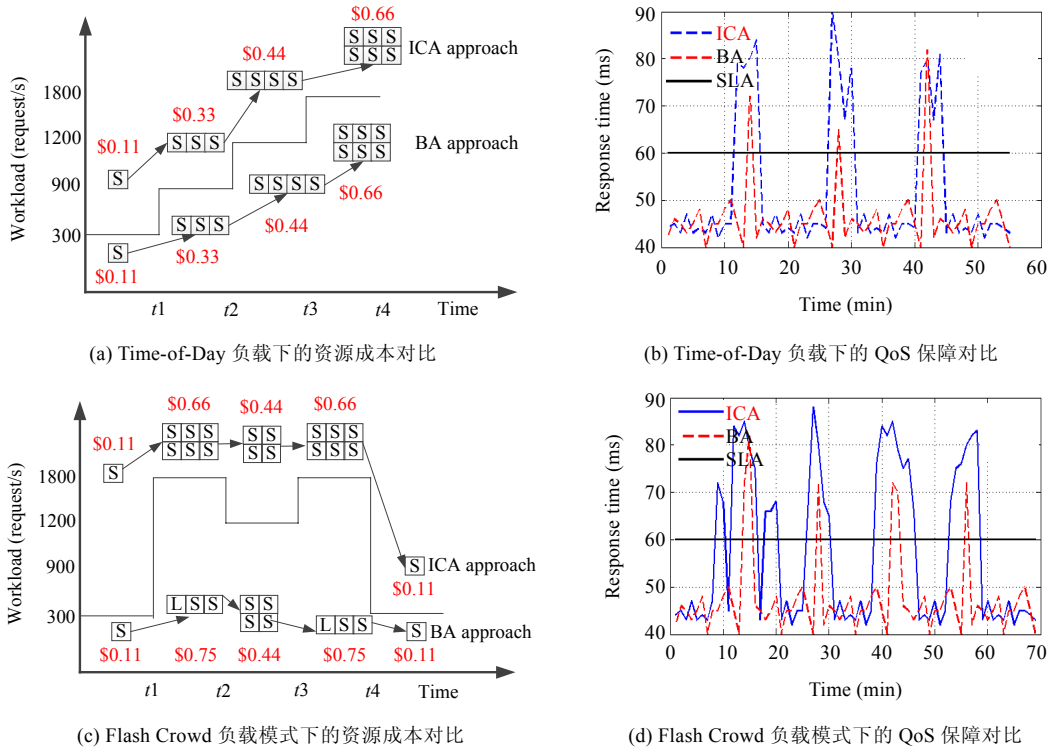


Fig.11 Comparison between the Benefit-Aware approach and the Infrastructure-Cost-Aware approach

图 11 Benefit-Aware 方法和 Infrastructure-Cost-Aware 方法对比

4.5.2 Benefit-Aware 方法和 Transition-Cost-Aware 方法对比实验

本文以 JEE 服务器组件为例,分别在 Time-of-Day 和 Flash Crowd 两种典型的负载模式下测试 Benefit-Aware(BA)方法和 Transition-Cost-Aware(TCA)方法在满足兼顾资源成本最小化和 QoS 保障方面的对比实验.我们做以下假设:

- (1) 虚拟机配置中 CPU 核数与其计算能力成正比,并已知 Small 配置的虚拟机最多承载 550 请求/秒,则 Large 配置的虚拟机最多可承载 2 200 请求/秒;
- (2) 每个资源重配置整体周期都足够长;
- (3) 约定图 12 中 L 表示 Large 配置的虚拟机,S 表示 Small 配置的虚拟机.

图 12(a)、图 12(b)分别是 JEE 服务器组件在 Time-of-Day 负载模式下(负载从 550 请求/秒渐进式的增加到 2 200 请求/秒)资源成本支出和 QoS 保障的实验结果.结果显示,与 Transition-Cost-Aware(TCA)方法相比,Benefit-Aware(BA)方法能够在保障 QoS 约束的基础上节约 10%的资源成本.其原因是,在 t_3 周期,虽然 Large 配置的计算能力正好是 Small 配置虚拟机的 4 倍,但 Transition-Cost-Aware(TCA)方法无法感知到 1 个 Large 配置的虚拟机成本和 4 个 Small 配置的虚拟机成本之间的差异性.

图 12(c)、图 12(d)分别是 JEE 服务器组件在 Flash Crowd 负载模式下(负载从 300 请求/秒突然增加到 2 200 请求/秒,然后稳定在 300 请求/秒)资源成本支出和 QoS 保障的实验结果.结果显示,与 Transition-Cost-Aware (TCA)方法相比,Benefit-Aware(BA)方法能够在保障 QoS 约束的基础上节约 25%的资源成本,其原因同上.

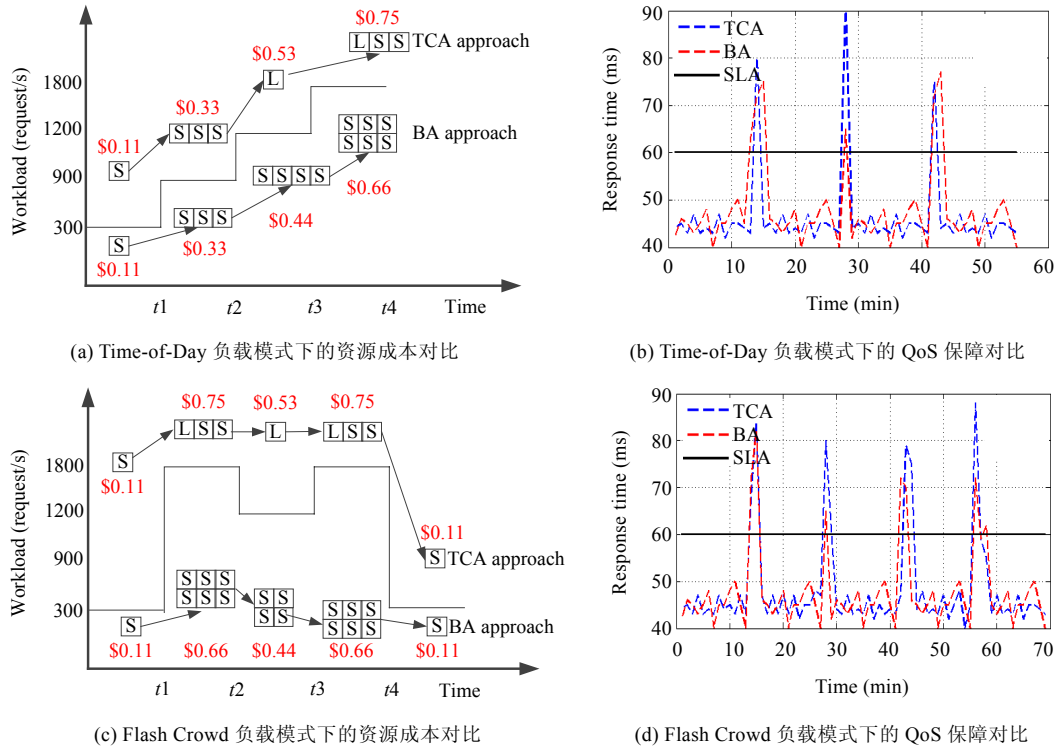


Fig.12 Comparison between the Benefit-Aware approach and the Transition-Cost-Aware approach

图 12 Benefit-Aware 方法和 Transition-Cost-Aware 方法对比

4.5.3 Benefit-Aware 方法和 Both-Cost-Aware 方法对比实验

Upendra 工作^[10]同时考虑了资源成本和转换成本两个要素,其本质是先选取转换成本最小化的资源重配置操作,在此基础上规划出资源成本最小的虚拟机资源,本文称为 Both-Cost-Aware 方法.实验以 JEE 服务器组件为例,分别在 Time-of-Day 和 Flash Crowd 两种典型的负载模式下测试 Benefit-Aware(BA)方法和 Both-Cost-Aware (BCA)方法在满足兼顾资源成本最小化和 QoS 保障方面的对比实验.我们做以下假设:

- (1) 虚拟机配置中 CPU 核数与其计算能力成正比,并已知 Small 配置的虚拟机最多承载 550 请求/秒,则 Large 配置的虚拟机最多可承载 2 200 请求/秒.
- (2) 在 Time-of-Day 负载模式下,每个资源重配置整体周期都足够长;而在 Flash Crowd 负载模式下则正好相反.
- (3) 约定图 13 中 L 表示 Large 配置的虚拟机,S 表示 Small 配置的虚拟机.

图 13(a)、图 13(b)分别是 JEE 服务器组件在 Time-of-Day 负载模式下(负载从 550 请求/秒渐进式地增加到 2 200 请求/秒)资源成本支出和 QoS 保障的实验结果.结果显示,与 Both-Cost-Aware(BCA)方法相比:(1) Benefit-Aware(BA)方法虽然多经历了一次服务质量违约,但本质上还是满足服务质量约束.因为本文约束在 95%的时间内保障 Web 应用服务质量,而 t_2 和 t_3 周期都足够长.(2) Benefit-Aware(BA)方法能够节约 25%的资源成本.其原因是:容量调整的转换成本比 VM 复制的转换成本下,Both-Cost-Aware(BCA)方法在 t_2 周期会优先选择转换成本最小的容量调整操作,从 1 个 Small 配置的虚拟机调整到 Large 配置的虚拟机,而不会选择通过 VM 复制到达同样满足资源需求的 3 个 Small 配置虚拟机.

图 13(c)、图 13(d)分别是 JEE 服务器组件在 Flash Crowd 负载模式下(负载从 300 请求/秒突然增加到 2 200 请求/秒,然后稳定在 300 请求/秒)资源成本支出和 QoS 保障的实验结果.结果显示,与 Both-Cost-Aware(BCA)方法相比:(1) Benefit-Aware(BA)方法需要多支付 10%的资源成本,其原因是 t_2, t_3 和 t_4 周期都足够短. Benefit-Aware(BA)方法在资源容量规划时会考虑转换周期与稳定周期之间的关系,如第 3.2 节收益函数的构建.因此, Benefit-Aware(BA)方法会采用“资源换稳定周期”的策略实现资源供给;而 Both-Cost-Aware(BCA)方法则在每个周期执行一次资源重配置操作,因此能够有效节约资源成本.(2) Benefit-Aware(BA)能够有效减少服务质量违约次数,其原因是: Benefit-Aware(BA)方法多支付的资源成本会使 t_2, t_3 和 t_4 周期满足 Web 应用资源需求,不需要执行资源重配置操作,避免了服务质量违约;而 Both-Cost-Aware(BCA)方法则会因执行资源重配置操作导致服务质量违约.

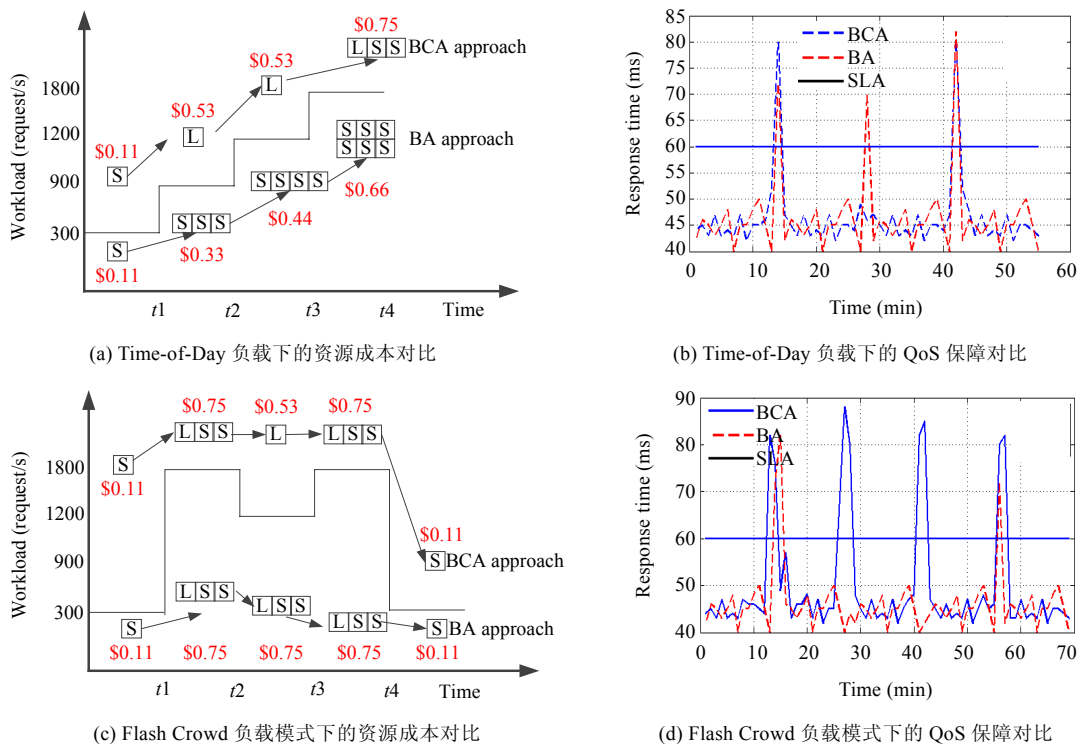


Fig. 13 Comparison between the Benefit-Aware approach and the Both-Cost-Aware approach
图 13 Benefit-Aware 方法和 Both-Cost-Aware 方法对比

5 相关工作

Cost-Oblivious 方法:这类方法的本质是以保障 Web 应用的 QoS 为最终目标,当 Web 应用的资源需求小于平台资源供给时,这类方法不会回收过剩资源.例如,文献[33–35]提出了基于“竞价-拍卖”的资源动态分配方法.该方法的本质是采用了高竞价者具有资源使用权的策略.上述方法的有效性建立在服务器端资源受限的前提下,解决的问题是有限资源的最优分配问题;而本文的假设前提是服务器端的资源充裕,解决的是 Web 应用资源的按需提供问题.

Infrastructure-Cost-Aware 方法:这类方法的本质是在保障 Web 应用 QoS 约束的同时,平台根据 Web 应用的资源需求采用按需供给的策略,从而达到资源成本最小化的目标.例如,文献[36–38]提出了基于两阶段效用最大化的资源分配方法,即建立每个应用的局部效用最大化函数和多个应用的全局效用最大化函数. Chandra^[9]提出

了基于时域排队模型的资源分配方法,建立负载和资源需求之间的映射关系,采用“监测-调整-监测”实时资源供给策略保障 Web 应用的 QoS.上述方法未能考虑转换成本,仅适用于逻辑资源(线程等)的按需提供,因为虚拟化资源供给具有耗时长并伴随性能衰减的新特点,忽略转换成本会导致频繁的服务质量违约,如第 4.5.1 节的实验所示.

Transition-Cost-Aware 方法:这类方法的本质是在资源重配置过程中考虑或规避转换成本的开销,通过降低转换次数来减少 QoS 违约.例如,Gueyoung^[11,16]提出了基于成本敏感的资源动态分配方法.该方法针对基于虚拟机的资源重配置操作具有转换成本的特点,以“奖励-惩罚”效用最大化作为资源动态分配的依据.文献[12,14]提出了基于多路复用的资源动态分配方法.该方法针对不同 Web 应用峰值负载在时间上具有错开性的特点,预先约定多个 Web 应用及其 VM 与物理机的映射关系,当 Web 应用的资源需求变化时,只允许执行本地资源调整,从而达到规避转换成本的目的.上述方法的本质是为了解决服务质量约束下 Web 应用的放置优化问题,当 Web 应用负载减少导致资源需求变化时,这类方法并不会执行 Web 应用重配置操作,导致资源成本浪费,如第 4.5.2 节的实验所示.

Both-Cost-Aware 方法:Upendra^[10]工作与本文工作类似,考虑了资源成本要素和转换成本要素.该方法的本质是优先选取转换成本最小的资源重配置操作,在此基础上规划出资源成本最小的虚拟机资源.但它并没有量化转换周期与稳定周期之间的关系,在负载突变模式仍存在频繁服务质量违约不足,如第 4.5.3 节的实验所示.

Benefit-Aware 方法:的本质是显示量化了转换周期与稳定周期之间的映射关系,以期望的稳定周期作为虚拟化资源规划的依据.其优点是能够实现较精确的虚拟化资源按需提供,满足服务质量约束下的资源成本最小化目标.其缺点是,该方法目前主要关注 CPU 资源的消耗,而 Web 应用通常会消耗多种类型的资源,比如网络和磁盘等,这些资源都可能成为瓶颈.因此,该方法目前无法适用于如文献[12,14,18]的 IO 密集型 Web 应用场景.另外,该方法假定稳定周期期望 $E(T)$ 为常数,实际场景中在线迁移操作耗时在文献[33]条件下有时很难通过离线测试获取的.因此,可考虑采用控制论^[36,39]方法进行改进.

6 总结和下一步的工作

本文的主要贡献是提出了一种面向在线 Web 应用的按需资源供给的方法,能够有效兼顾资源成本最小化和 QoS 保障目标.主要贡献包括:

- (1) 构建收益函数描述 Web 应用在稳定周期的 QoS 收益,有效防止应用在负载突变模式下频繁违反 QoS 约束.
- (2) 根据归约法实现了基于收益率最大化的资源容量规划算法,保障了资源成本最小化的目的.
- (3) 将本方法应用到 TPC-W 工业测试基准中.实验结果显示,本方法与 Cost-Oblivious 方法相比最多能节约 25% 的资源成本,而与 Cost-Aware 方法相比则能有效减少 Web 应用在负载突变模式下的 QoS 违约次数.

同时,本文还存在如下有待改进的问题:首先,在 Web 应用性能剖析方面,本文未能深入考虑多核和虚拟化开销的影响,实验结果显示,其预测误差随虚拟机核数增加而增加.因此,面向多核和虚拟化计算环境的 Web 应用性能建模是本文下一步工作的重点;其次,在规划算法方面,规划算法的本质是从所有虚拟机配置和资源重配置操作组合中选择出满足 QoS 保障和资源成本最小化需求的最优组合方案.然而,在资源重配置操作种类确定的前提下,该算法搜索空间与虚拟机配置种类数成指数关系.因此,规划算法优化也是本文下一步工作的重点.

References:

- [1] Vaquero LM, Rodero-Merno L, Caceres J, Lindner M. A break in the clouds: Towards a cloud definition. ACM SIGCOMM Computer Communication, 2008,39(1):50-55. [doi: 10.1145/1496091.1496100]
- [2] Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee GH, Patterson D, Rabkin A, Stoica I, Zaharia M. Above the clouds: A Berkeley view of cloud computing. Technical Report, EECS-2009-28, Berkley: University of California Press, 2009. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>

- [3] Gartner Company. Gartner maps out the rapidly evolving market for cloud infrastructure as a service. 2011. <http://www.gartner.com/it/page.jsp?id=1622514>
- [4] Amazon Team. Amazon EC2. 2013. <http://aws.amazon.com/ec2/>
- [5] IBM Team. IBM SmartCloud. 2013. <http://www.ibm.com/cloud-computing/social/us/en/>
- [6] OpSource Team. OpSource cloud. 2013. <http://cloud-vpn.net/>
- [7] Bennani M, Manesce D. Resource allocation for autonomic data centers using analytic performance models. In: Proc. of the IEEE ICAC. 2005. 229–240. [doi: 10.1109/ICAC.2005.50]
- [8] Zhang Q, Cherkasova L, Smirni E. A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In: Proc. of the IEEE ICAC. 2007. 27–36. [doi: 10.1109/ICAC.2007.1]
- [9] Urgaonkar B, Shenoy P, Chandra A, Goyal P, Wood T. Agile dynamic provisioning of multi-tier internet applications. ACM Trans. on Autonomous and Adaptive Systems, 2008,3(1):1–39. [doi: 10.1145/1342171.1342172]
- [10] Sharma U, Shenoy P, Sahu S, Shaikh A. A cost-aware elasticity provisioning system for the cloud. In: Proc. of the Int'l Conf. on Distributed Computing Systems. 2011. 559–570. [doi: 10.1109/ICDCS.2011.59]
- [11] Jung G, Joshi K, Hiltunen M, Schlichting R, Pu C. A cost-sensitive adaptation engine for server consolidation of multitier applications. In: Proc. of the ACM/IFIP/USENIX Int'l Middleware Conf. 2009.
- [12] Gartner Company. Gartner maps out the rapidly evolving market for cloud infrastructure as a service. 2011. <http://www.gartner.com/it/page.jsp?id=1622514>
- [13] TPC Transaction Processing Performance Council. TPC-W benchmark. 2005. <http://www.tpc.org/tpcw/>
- [14] Clifton C, Leavens GT, Chambers C, Millstein T. MultiJava: Modular open classes and symmetric multiple dispatch for Java. ACM SIGPLAN Notices, 2000,35(10):130–145. [doi: 10.1145/354222.353181]
- [15] Jung G, Joshi K, Hiltunen M, Schlichting R, Pu C. Generating adaptation policies for multi-tier applications in consolidated server environments. In: Proc. of the IEEE ICAC. 2008. 23–32. [doi: 10.1109/ICAC.2008.21]
- [16] Jung GY, Hiltunen MA, Joshi KR, Schlichting RD, Pu C. Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures. In: Proc. of the Int'l Conf. on Distributed Computing Systems. 2010. 62–73. [doi: 10.1109/ICDCS.2010.88]
- [17] Xen Team. Linux foundation. 2013. <http://www.xenproject.org/>
- [18] Xu J, Zhao M, Fortes J, Carpenter R, Yousif M. On the use of fuzzy modeling in virtualized data center management. In: Proc. of the IEEE ICAC. 2007. 25–34. [doi: 10.1109/ICAC.2007.28]
- [19] Dornemann T, Juhnke E, Freisleben B. On-Demand resource provisioning for BPEL workflows using Amazon's elastic compute cloud. In: Proc. of the 9th IEEE/ACM Int'l Symp. on Cluster Computing and the Grid (CCGRID 2009). IEEE, 2009. 140–147.
- [20] Zhang Q, Cherkasova L, Smirni E. A regression-based analytic model for dynamic resource provisioning of multi-tier applications. In: Proc. of the IEEE ICAC. 2007. 27–36. [doi: 10.1109/ICAC.2007.1]
- [21] Hoff T. Friendster lost lead because of a failure to scale. 2007. <http://highscalability.com/blog/2007/11/13/friendster-lost-lead-because-of-a-failure-to-scale.html>
- [22] Krishnamurthy D, Rolia J, Majumdar S. A synthetic workload generation technique for stress testing session-based systems. IEEE Trans. on Software Engineering, 2006,32(11):868–882. [doi: 10.1109/TSE.2006.106]
- [23] Cherkasova L, Phaal P. Session-Based admission control: A mechanism for peak load management of commercial Web sites. IEEE Trans. on Software Engineering, 2002,51(6):669–685. [doi: 10.1109/TC.2002.1009151]
- [24] Ming Z, Yin JF, Yang W, Wang H, Xiao ZJ. A Web performance testing framework and its mixed performance modeling process. Journal of Computer Research and Development, 2010,47(7):1192–1200 (in Chinese with English abstract).
- [25] SpringSource Team. Hepric HQ. 2013. <http://www.hyperic.com/>
- [26] Apache Foundation. Apache commons. 2013. <http://commons.apache.org/proper/commons-math/>
- [27] Jung G, Joshi K, Hiltunen M, Schlichting R, Pu C. Generating adaptation policies for multi-tier applications in consolidated server environments. In: Proc. of the IEEE ICAC. 2008. 23–32. [doi: 10.1109/ICAC.2008.21]
- [28] Bagnall AJ, Janacek GJ. Clustering time series from ARMA models with clipped data. In: Proc. of the 10th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2004. 49–58. [doi: 10.1145/1014052.1014061]

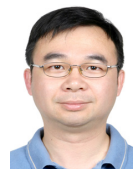
- [29] Zhu XY, Wang ZK, Singhal S. Utility-Driven workload management using nested control design. In: Proc. of the American Control Conf. 2006. 6033–6038.
- [30] Rolia J, Cherkasova L, McCarthy C. Configuring workload manager control parameters for resource pools. In: Proc. of the 10th IEEE/IFIP Network Operations and Management Symp. 2006. 127–137. [doi: 10.1109/NOMS.2006.1687545]
- [31] Arlitt M, Jin T. A workload characterization study of the 1998 world cup web site. IEEE Network, 2000,14(3):30–37.
- [32] Dilley JA. Web server workload characterization. Technical Report, HPL-96-160, Hewlett-Packard Laboratories, Technical Publications Department, 1996.
- [33] Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limapch C, Pratt I, Warfield A. Live migration of virtual machines. In: Proc. of the ACM/Usenix NSDI. 2005. 273–286.
- [34] Norris J, Coleman K, Fox A, Candea G. OnCall: Defeating spikes with a free-market application cluster. In: Proc. of the 1st Int'l Conf. on Autonomic Computing (ICAC 2004). 2004. 198–205.
- [35] Huang T, Chen NJ, Wei J, Zhang WB, Zhang Y. OnceAS/Q: A QoS-enabled Web application server. Ruan Jian Xue Bao/Journal of Software, 2004,15(12):1787–1799 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/15/1787.htm>
- [36] Liu X, Zhu X, Singhal S, Arlitt M. Adaptive entitlement control of resource containers on shared servers. In: Proc. of the IFIP/IEEE Int'l Symp. on Integrated Network Management. 2005. 163–176. [doi: 10.1109/INM.2005.1440783]
- [37] Urgaonkar B, Pacifici G, Shenoy P, Spreitzer M, Tantawi A. An analytical model for multi-tier internet services and its applications. In: Proc. of the ACM Int'l Conf on Measurement and Modeling of Computer Systems. 2005. 291–302. [doi: 10.1145/1064212.1064252]
- [38] Kim M, Noble B. Mobile network estimation. In: Proc. of the AAACM Conf. on Mobile Computing & Network. 2001. 298–309. [doi: 10.1145/381677.381705]
- [39] Wu H, Zhang W, Wei J, Huang T. A benefit-aware on-demand provisioning approach for multi-tier applications in cloud computing. Frontiers of Computer Science, <http://link.springer.com/article/10.1007/s11704-013-2201-8>

附中文参考文献:

- [24] 明仲,尹剑飞,杨巍,王慧,肖志娇.一种 Web 系统性能测试框架及其混合建模过程.计算机研究与发展,2010,47(7):1192–1200.
- [35] 黄涛,陈宁江,魏峻,张文博,张勇.OnceAS/Q:一个面向 QoS 的 Web 应用服务器.软件学报,2004,15(12):1787–1799. <http://www.jos.org.cn/1000-9825/15/1787.htm>



吴恒(1983—),男,湖南衡阳人,博士生,主要研究领域为网络分布式计算,软件工程.
E-mail: wuheng09@otcaix.iscas.ac.cn



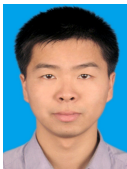
魏峻(1970—),男,博士,研究员,CCF 高级会员,主要研究领域为网络分布式计算,软件工程.
E-mail: wj@otcaix.iscas.ac.cn



张文博(1976—),男,博士,副研究员,CCF 会员,主要研究领域为网络分布式计算,软件工程.
E-mail: zhangwenbo@otcaix.iscas.ac.cn



黄涛(1965—),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为网络分布式计算,软件工程.
E-mail: tao@otcaix.iscas.ac.cn



张建华(1983—),男,博士生,CCF 会员,主要研究领域为网络分布式计算,软件工程.
E-mail: zhangjianhua07@otcaix.iscas.ac.cn