

云环境下基于神经网络和群搜索优化的资源分配机制*

孙佳佳^{1,2}, 王兴伟^{1,2}, 高程希^{1,2}, 黄敏^{1,2}

¹(东北大学 信息科学与工程学院, 辽宁 沈阳 110819)

²(中国科学院 网络化控制系统重点实验室, 辽宁 沈阳 110016)

通讯作者: 王兴伟, E-mail: wangxw@mail.neu.edu.cn

摘要: 在云环境下, 各种闲置资源可以通过池化形成资源池, 进而利用虚拟化技术将资源池中的不同资源组合以服务的形式提供给用户使用, 因此需要合理而有效的机制来分配资源. 针对云环境下资源的特点, 将经济学和智能方法相结合, 提出了一种基于双向组合拍卖的智能资源分配机制. 在该机制中, 提出了基于体验质量(quality of experience, 简称 QoE)的威望系统, 引入威望衰减系数和用户信誉度, 降低拍卖中恶意行为造成的影响, 为资源交易提供 QoE 支持. 对拍卖中的竞价决策, 综合考虑多种因素, 提出了基于 BP 神经网络的竞标价格决策机制, 不仅可以合理确定竞标价, 而且使价格可以动态适应市场变化. 最后, 由于组合拍卖胜标确定问题是 NP 完全的, 因此引入群搜索优化算法, 以市场盈余和总体威望为优化目标, 得到资源分配方案. 仿真研究结果表明, 该机制是可行和有效的.

关键词: 云计算; 双向组合拍卖; 体验质量; 威望; BP 神经网络; 群搜索优化

中图法分类号: TP316

中文引用格式: 孙佳佳, 王兴伟, 高程希, 黄敏. 云环境下基于神经网络和群搜索优化的资源分配机制. 软件学报, 2014, 25(8): 1858-1873. <http://www.jos.org.cn/1000-9825/4555.htm>

英文引用格式: Sun JJ, Wang XW, Gao CX, Huang M. Resource allocation scheme based on neural network and group search optimization in cloud environment. Ruan Jian Xue Bao/Journal of Software, 2014, 25(8): 1858-1873 (in Chinese). <http://www.jos.org.cn/1000-9825/4555.htm>

Resource Allocation Scheme Based on Neural Network and Group Search Optimization in Cloud Environment

SUN Jia-Jia^{1,2}, WANG Xing-Wei^{1,2}, GAO Cheng-Xi^{1,2}, HUANG Min^{1,2}

¹(College of Information Science and Engineering, Northeastern University, Shenyang 110819, China)

²(Key Laboratory of Networked Control System, The Chinese Academy of Sciences, Shenyang 110016, China)

Corresponding author: WANG Xing-Wei, E-mail: wangxw@mail.neu.edu.cn

Abstract: In cloud environment, all kinds of idle resources can be pooled to establish a resource pool, and different kinds of resources can be combined as a service to the users through virtualization. Therefore, an effective scheme is necessary for managing and allocating the resources. In this paper, economic and intelligent methods are employed to form an intelligent resource allocation scheme based on double combinatorial auction with respect to the characteristics of resources in cloud environment. In the proposed scheme, a reputation system on the basis of quality of experience (QoE) is devised, and the reputation attenuation coefficient and the user credit degree are introduced to decrease the negative effects of malicious behaviors on resource auctions, providing QoE support to resource dealing. In order to determine bidding price rationally, a bidding price decision mechanism based on back propagation (BP) neural network is presented to comprehensively consider various influence factors to make price adapt to the fluctuating market. Finally, due to the fact that the problem of winner determination in combinatorial auction is NP-complete, a group search optimization algorithm is adopted to find

* 基金项目: 国家杰出青年科学基金(61225012, 71325002); 教育部高等学校博士学科点专项科研基金(20120042130003); 中央高校基本科研业务费专项资金(N110204003, N120104001)

收稿时间: 2012-12-26; 修改时间: 2013-6-26, 2013-09-09; 定稿时间: 2013-12-09

the specific resource allocation solution with market surplus and total reputation optimized. Simulation studies are conducted to demonstrate the feasibility and effectiveness of the proposed scheme.

Key words: cloud computing; double combinatorial auction; quality of experience; reputation; back propagation neural network; group search optimization

作为分布式处理、并行处理和网格计算的发展和商业实现,云计算已经成为当今的研究热点^[1].在云环境下,能够从动态虚拟化的资源池中向用户或者各种应用系统按需提供计算能力、存储能力或者虚拟机服务等.由于资源池中的资源具有异构性、动态性和分散性等特点,使云环境下的资源优化分配成为一项富有挑战性的工作^[2].

研究表明,引入经济学理论可以有效地解决资源分配问题.在传统的网格环境下,已经提出了一些适用于资源分配的经济学模型^[3],例如商品市场模型、标价模型、议价模型、招标合同模型、合作交换模型和拍卖模型等.由于拍卖模型的高效与灵活,使其成为有效的资源分配方法的基础.很多资源分配机制都使用了拍卖模型,并且取得了较好的效果.文献[4]研究了3种基于拍卖的资源分配机制:第一价格封标拍卖、逆向拍卖和双向拍卖.结果表明:第一价格封标拍卖有利于资源方,逆向拍卖更利于用户方,双向拍卖对双方更公平.文献[5]在传统的连续双向拍卖基础上提出了稳定连续双向拍卖机制,不仅提高了市场效率,而且提高了调度效率.文献[6]提出了一种扩展的第二价格封标拍卖模型,引入了分析者角色,设计了基于隐马尔科夫模型的分析者预测算法,保证了较高的胜标概率.文献[7]将资源分配环境建模为双向市场,通过改进连续双向拍卖,提出了一种基于盈利的连续双向拍卖模型,用于实时资源分配,不仅缩短了资源分配的时间,而且提高了资源提供者的盈利.文献[8]提出了一种基于反复拍卖的分配协议,用于共享计算资源,同时给出了以所有用户效益最大化为目标的竞标算法,从而达到了资源分配博弈的均衡状态,提高了用户的长期利润.文献[9]提出了一种基于双向组合拍卖的网格资源分配模型,通过网格用户和业务提供者的双向组合拍卖来实现资源的分配,给出了网格资源定价算法,以解决资源分配基础之上的计费问题,可以高效地完成资源分配及定价.文献[10]提出了一种基于双向拍卖机制的网格资源分配方法,给出了网格资源分配框架,采用统一拍卖方式,可以灵活调节交易双方的付费.文献[11]建立了一种多赢家网格资源拍卖模型,给出了一种多赢家拍卖算法,实现了系统收益最大化,提高了资源利用率.

网格环境下基于拍卖模型的资源分配机制,对于解决云环境下的资源分配问题具有参考价值.很多学者在以往研究的基础上,将拍卖模型应用于云环境下的资源分配问题.文献[12]融合了批量匹配思想和逆向拍卖协议,提出了逆向批量匹配拍卖机制,设计了两次惩罚策略防止交易欺诈,并且利用智能算法得到优化的资源分配方案.文献[13]通过组合多个受限英式拍卖,建立了一种改进的周期性拍卖模型,采用了能力带描述特殊用户的存储能力,不仅可以最大化卖方效益,而且减少了胜标确定时间.文献[14]在云环境下计算资源分配时采用了动态第二价格封标拍卖机制,利用了该拍卖机制固有的激励相容特性,保证了合理的利润和有效的资源分配.文献[15]提出了一种基于市场的连续双向拍卖模型,分析了不同调度策略的影响,说明调度策略的选择取决于调度者对风险的态度及其认知和融合能力.文献[16]提出了方便用户灵活交换闲置资源的云市场框架,并为市场设计了一种基于贝叶斯博弈的双向拍卖定价模型,研究了基于该模型的相关定价策略.文献[17]基于连续双向拍卖协议提出了一种启发式云资源分配算法,基于M/M/1排队系统建立了资源分配模型,运用纳什均衡得到优化分配方案.文献[18]提出了基于知识的连续双向拍卖模型,基于历史交易信息,利用学习算法动态地确定资源价格,可以有效地决策出稳定的交易价.文献[19]在传统双向拍卖的基础上提出了多路双向拍卖机制,用于云服务交易,可以在经济效益方面实现最优分配.文献[20]综合了现有的定价策略,针对连续双向拍卖存在的问题,提出了连续逆向拍卖机制,并且基于标签和约束集合设计了一种投标描述语言.文献[21]采用了一种能够自适应调节资源价格的按比例投标拍卖模型,研究了资源总量受限环境下的最优竞标决策问题.文献[22]将云环境下的虚拟机分配问题建模为组合拍卖问题,并且证明了对该问题基于组合拍卖的分配机制比固定定价机制更有效率,云提供者可以获取更多盈利.文献[23]提出了一种基于组合拍卖的市场机制.该机制包含了用于预订服务的期货市场和用于即时分配的现货市场.两种市场均通过混合整数规划,以全局经济效益最大化为目标得到最优分配.文献[24]提出了一种基于双向组合拍卖的动态资源定价模型.用户根据响应时间出价,资源提供者根据系

统当前负载要价,优于传统的固定定价算法,而且由于采用了组合拍卖协议,可以满足云环境下组合多种资源提供服务的需求.文献[25]基于双向拍卖模型,考虑了参与者的服务质量要求和资源的供求关系等因素,提出了一种适应性云计算资源分配机制,保证了市场参与者的收益,提高了用户满意度和云计算系统的资源利用率.文献[26]提出了一种云计算环境下基于双向拍卖的资源分配模型.基于遗传算法求解云资源优化分配方案,不仅可以满足更多买方的需求,而且提高了云资源利用率.

综上所述,多数现有研究只支持单种资源交易,不支持多种资源组合,对云环境下用户多样化服务需求的适应性不强;基本上都忽略了资源交易中保障用户体验质量(quality of experience,简称 QoE)的重要性,未充分考虑市场机制中普遍存在的欺诈问题和恶意行为对资源分配可能产生的负面影响,没有设置相应的机制予以解决.此外,多数现有方法在决策竞标价格时忽略了历史竞标数据中的有效信息,导致所决策出的价格不能适应市场变化.实际上,不合理的竞标价以及未考虑恶意行为的无 QoE 保障的分配机制会损害供需双方的利益,并且影响匹配效率,从而导致云环境无法向用户提供可信赖的云资源交易平台.

针对以上不足,本文以双向组合拍卖协议为基础,提出了一种基于神经网络和群搜索优化(group search optimization,简称 GSO)^[27]的资源分配机制.通过双向组合拍卖协议,可以支持任意组合多种资源的拍卖;通过基于 QoE 的威望系统抵制拍卖中的恶意行为,同时保障资源交易的 QoE;采用基于 BP(back propagation)神经网络^[28]的竞标价格决策机制合理地确定竞标价,以市场盈余和总体威望为优化目标,采用 GSO 解决胜标确定问题,支持云环境下资源的高效分配.

1 整体框架

本文提出的云环境下的资源分配机制主要包括 5 种角色:云服务提供者(cloud service provider,简称 CSP)、提供者代理(provider agent,简称 PA)、云服务消费者(cloud service consumer,简称 CSC)、消费者代理(consumer agent,简称 CA)和拍卖中介(auction intermediary,简称 AI).其中,CSP 是拥有资源的服务提供者,CSC 是需要租用资源的服务消费者,可以是个人或单位.

在传统的公有云和私有云等云环境下,主要依托大型数据中心为用户提供计算能力和存储能力等,通常采用集中式管理,一般以标价等统一定价形式出售资源;虽然也可以采用拍卖等方式进行价格决策,但是价格通常在一段时间内比较稳定.相对于上述的经典云环境,本文提出的资源分配机制更适合于分布式云^[29]和社交云^[30]等云环境.在分布式云中,服务提供者通常不依赖于大型数据中心提供服务,而是利用地理上分散的小/微型数据中心为用户提供服务,这样不仅可以避免大型数据中心带来的高能耗、散热和温度与湿度控制等问题,而且有利于提高全网资源利用率.社交云主要强调资源共享,用户可以通过在线社交网络发现并与朋友分享自己的闲置资源.由于分布式云和社交云等环境下的资源通常来源于地理上分散的小/微型数据中心,因此一般采用分布式管理;同时,在这样的环境中,云服务提供者的资源拥有量有限,云服务消费者的数量多变,资源价格受供求关系变化的影响大,需要更为灵活的定价策略,通常需要通过合理的价格决策和优化的资源分配方案,将有限的资源分配给适合的消费者.本文基于神经网络进行价格决策,采用 GSO 给出资源分配方案,可以较好地适应上述需求.

本文提出的资源分配机制的功能主要在 PA、CA 和 AI 中实现.PA 和 CA 负责为其所服务的对象提供决策支持,例如初始化标的、决策竞标价格、提交标的、评分等;AI 负责收集标的、运行胜标确定算法、通知拍卖结果、发布市场运行数据和管理威望系统等.系统框架如图 1 所示.

该分配机制的主要工作流程如下:

- (1) CSC 需要服务时向 CA 提交相关信息,CA 根据任务要求初始化 CSC 的竞标信息;CSP 存在空闲资源时向 PA 提交相关信息,PA 初始化 CSP 的竞标信息.
- (2) CA 和 PA 利用竞标价格决策机制产生竞标价,并提交给 AI.
- (3) AI 每隔一段时间或者当接收的标的数量超过某一阈值时启动胜标确定算法,执行匹配过程.如果未达到最大轮数,则发送消息给失败的参与者,允许其重新修改竞标价,再次执行匹配过程;否则,匹配

结束.

- (4) AI 把拍卖分配结果发布给每个参与者,胜标的 CSP 和 CSC 之间进行交易.
- (5) 交易后,双方根据交易方表现进行互评,AI 在威望系统中更新参与者的威望信息.

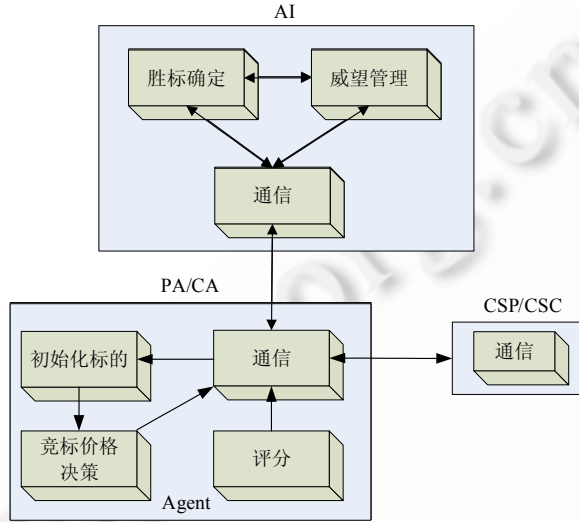


Fig.1 System framework

图 1 系统框架

2 基于 QoE 的威望系统

设计基于 QoE 的威望系统的目的,在于减少拍卖过程中的恶意行为,为资源交易提供 QoE 保障,提高双方的满意度.在交易结束之后,用户根据其自身对本次交易的综合感受对交易方进行评分,该评分可看作是 QoE 的量化值.本文对威望系统的设计主要是以 QoE 和交易额为基础刻画用户威望.为了使威望信息更加真实、可靠,考虑威望的时效性和用户信誉度这两个属性,共同确定用户的威望值.

首先,威望基于 QoE 和交易额;其次,威望具有时效性,随时间而衰减.综合这两个方面的信息,CSP_j 在参加第 k 次拍卖后的威望值 $reputation_{j,k}$ 可初步由公式(1)计算得到:

$$reputation_{j,k} = de(\Delta t_{k-1}^k) \times \frac{transaction_{j,k-1}}{transaction_{j,k}} \times reputation_{j,k-1} + (1 - de(\Delta t_{k-1}^k)) \times \sum_i \frac{R_{i,j,k} \times price_{i,j,k} \times QoE_{i,j,k}}{transaction_{j,k} - transaction_{j,k-1}} \quad (1)$$

其中,

- $transaction_{j,k}$ 表示 CSP_j 参加第 k 次拍卖后的累计交易额;
- $R_{i,j,k}$ 和 $price_{i,j,k}$ 分别表示 CSP_j 与 CSC_i 在第 k 次拍卖中的交易量和成交价,因此, $R_{i,j,k} \times price_{i,j,k}$ 为两者在第 k 次拍卖中的交易额;
- $QoE_{i,j,k} \in [0,1]$ 表示 CSC_i 对 CSP_j 在第 k 次拍卖中的表现的评分,1 表示完全满意,0 表示完全不满意;
- $de(\Delta t_{k-1}^k)$ 是威望随时间的衰减系数,如公式(2)所示:

$$de(\Delta t_{k-1}^k) = \begin{cases} 1, & \Delta t_{k-1}^k < t_{min} \\ \frac{t_{max} - \Delta t_{k-1}^k}{t_{max} - t_{min}}, & t_{min} \leq \Delta t_{k-1}^k \leq t_{max} \\ 0, & t_{max} < \Delta t_{k-1}^k \text{ 或 } k=1 \end{cases} \quad (2)$$

其中, Δt_{k-1}^k 为 CSP_j 参加的第 k 次拍卖和第 k-1 次拍卖的时间间隔, t_{min} 和 t_{max} 为时间间隔的下限和上限.

最后,为了保障威望系统的可靠性,避免出现威望诋毁或者威望炒作行为,本文引入用户信誉度来减小恶意

评分行为对威望系统造成的影响.如果 CSC_i 对 CSP_j 在参加的第 k 次拍卖中的表现的评分(即 $QoE_{i,j,k}$)与 CSP_j 在参加的第 k 次拍卖前的威望值 $reputation_{j,k-1}$ 差距很大,则认为 CSC_i 的评分不正常,存在威望炒作或威望诋毁嫌疑,属于可疑评分.令该差距 $deviation = |QoE_{i,j,k} - reputation_{j,k-1}|$,并根据 $deviation$ 值的大小把恶意评分行为分为 n 个等级,每个等级 l 对应一个可疑度 $sl \in (0,1]$,则 CSC_i 对 CSP_j 在第 k 次拍卖时的信誉度如公式(3)所示:

$$CR_{i,j,k} = \frac{N_{i,j,k} - \sum_{l=1}^n sl_l^*}{N_{i,j,k}} \quad (3)$$

其中, $N_{i,j,k}$ 为截至第 k 次拍卖时 CSC_i 对 CSP_j 的总评分次数, $N_{i,j,k}^*$ 为其中的可疑评分次数, sl_l^* 为相应的可疑度.

综上所述,在用户 QoE 和交易量的基础上,结合时效性和用户信誉度得到的威望计算方法如公式(4)所示:

$$reputation_{j,k} = de(\Delta t_{k-1}^k) \times \frac{transaction_{j,k-1}}{transaction_{j,k}} \times reputation_{j,k-1} + (1 - de(\Delta t_{k-1}^k)) \times \sum_i \frac{R_{i,j,k} \times price_{i,j,k} \times QoE_{i,j,k} \times CR_{i,j,k}}{transaction_{j,k} - transaction_{j,k-1}} \quad (4)$$

从公式(4)可以看出,用户威望值介于 0,1 之间.本文对所有用户进入拍卖市场前的威望初始值均设为 0.5,即,对刚“入市”的用户,系统因无法断定其品质而先默认其品质“一般”.

3 拍卖协议

3.1 提供者投标描述

CSP 拥有计算资源、存储资源和网络资源等,进行资源池化虚拟为服务以后,可以提供计算服务、存储服务和虚拟机服务等.其中,计算服务根据应用类型又可分为计算密集型服务和 IO 密集型服务.通过 CSP 提供的这些服务,可以满足 CSC 的不同任务需求.

CSP 投标时的参数集合为: {提供者标识符 ID; CPU(\$/(MIPS×hour)), memory(\$/(GB×hour)), storage(\$/(100GB×hour)) 和 bandwidth(\$/(Mbps×hour)) 的单位资源单位时间价格和随时间变化的供应量; 对软件和平台的支持集合; 对交易方威望的最低要求}. 用七元组 $\{CSP_ID, \{price_c, q(c, t)\}, \{price_m, q(m, t)\}, \{price_s, q(s, t)\}, \{price_b, q(b, t)\}, support_set, r_reputation\}$ 表示 CSP 的卖方标的.

3.2 消费者投标描述

对应 CSP 提供的服务, CSC 可以根据其任务要求按需进行投标. 一个 CSC 可以投递多个买方标的, 这些标的可以是“异或”关系, 只要求有 1 个中标即可; 也可以是“或”关系, 可以有 1 个或多个中标.

CSC 投标时的总参数集合为: {消费者标识符 ID; 竞标价格; 开始时间; 结束时间; CPU(MIPS); memory(GB); storage(GB); bandwidth(Mbps); 任务大小(MI); 数据量(GB); 用户偏好; 分割阈值; 对软件和平台的需求; 对交易方威望的最低要求}. 其中, 分割阈值表示可以把该任务划分给不同 CSP 的最大数量. 若分割阈值为 1, 则表示该任务不可分割.

CSC 根据其任务类型, 申请 CSP 提供的相应服务. 因此, CSC 某次投标的买方标的实际是上述总参数集合的一个子集, 例如:

- (1) 虚拟机服务: $\{CSC_ID, bid_price, start_time, end_time, CPU, memory, storage, bandwidth, software \& platform, partition_threshold, r_reputation\}$.
- (2) 存储服务: $\{CSC_ID, bid_price, start_time, end_time, storage, partition_threshold, r_reputiton\}$.
- (3) 计算密集型服务: $\{CSC_ID, bid_price, delivery_time, deadline, task_size, preference, partition_threshold, r_reputation\}$.
- (4) IO 密集型服务: $\{CSC_ID, bid_price, delivery_time, deadline, task_size, data_size, memory, storage, preference, partition_threshold, r_reputation\}$.

用户偏好(preference)分为 3 种类型: 越早开始越好、越晚开始越好和按时最好, 如图 2 所示. 它影响 CSP 对 CSC 任务执行的调度安排.

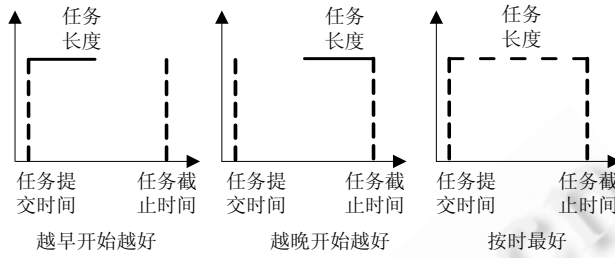


Fig.2 User preference

图 2 用户偏好

3.3 提供者总报价计算

由于在 CSP 的卖方标中指定的价格是对单位资源单位时间的报价,因此需计算 CSP 对 CSC 所需服务的总报价:

(1) 对于虚拟机服务,CSP_j对 CSC_i所需服务的总报价计算如公式(5)所示:

$$ask_price_{ij} = \sum_{R_j \in Rset} (RQ_{ij} \times price_R_j \times length_i) \tag{5}$$

其中,RQ_{ij}是 CSC_i所需服务对资源 R_j的需求量,price_{R_j}是 R_j的单位资源单位时间报价,Rset={CPU,memory,storage,bandwidth}.

(2) 对于存储服务,总报价计算如公式(6)所示:

$$ask_price_{ij}=storage_i \times price_s_j \times length_i \tag{6}$$

(3) 对于计算密集型服务,CSC_i根据任务大小确定任务报价,不需要考虑存储等费用,总报价计算如公式(7)所示:

$$ask_price_{ij}=task_size_i \times price_c_j \times length_i \tag{7}$$

(4) 对于 IO 密集型服务,CSC_i需要考虑使用 CPU、存储以及数据传送的带宽费用,总报价计算如公式(8)所示:

$$ask_price_{ij} = task_size_i \times price_c_j \times length_c_i + memory_i \times price_m_j \times length_m_i + storage_i \times price_s_j \times length_s_i + data_size_i \times price_b_j \times length_b_i \tag{8}$$

其中,对于虚拟机服务和存储服务,length_i=end_time_i-start_time_i;对于计算密集型服务,length_i是 CSC_i任务在 CSP_j上执行时需用的时间;对于 IO 密集型服务,length_{c_i},length_{m_i}和 length_{s_i}分别代表 CSP_j处理 CSC_i任务需用的 CPU 时间、memory 时间和 storage 时间,length_{b_i}是把 CSC_i的 data_size_i数据通过网络传送到 CSP_j需用的时间.

3.4 胜标确定规则

双向组合拍卖的胜标确定问题可以分为优化目标和约束函数两部分.优化目标即最大化市场盈余,包括市场总盈余和单位时间市场盈余,以及参与者的总体威望信息.约束函数包括软件和平台约束、时间约束、价格约束、威望约束、分割约束、资源约束和服务约束等.由于一个 CSC 可以投递多个买方标的,为简化胜标确定问题,根据标的间的关系进行如下处理:若标的间是“或”关系,则将这些标的看作是不同的 CSC 的投标进行处理;若是“异或”关系,则限定这些标的中只有 1 个可以胜出.

上述问题可归结为优化资源分配矩阵,矩阵代表胜标确定问题的解,其中,

- 矩阵的行代表 CSC.
- 矩阵的列代表 CSP.
- 元素 R_{ij}∈[0,1]代表 CSC_i任务所需服务分配到 CSP_j上执行的比例,若为 0,则表示 CSC_i任务所需服务不分配到 CSP_j上执行;若为 1,则表示 CSC_i任务所需服务全部由 CSP_j执行.

优化目标及约束条件的形式化描述如下:

$$\max total_surplus = \sum_{i=1}^m bid_price_i \left(\sum_{j=1}^n R_{ij} \right) - \sum_{j=1}^n \sum_{i=1}^m R_{ij} \times ask_price_{ij} \quad (9)$$

$$\max unit_surplus = \sum_{i=1}^m \frac{bid_price_i}{length_i} \left(\sum_{j=1}^n R_{ij} \right) - \sum_{j=1}^n \sum_{i=1}^m R_{ij} \times \frac{ask_price_{ij}}{length_i} \quad (10)$$

$$\max total_reputation = \sum_{i=1}^m \sum_{j=1}^n R_{ij} \times reputation_i + \sum_{j=1}^n \sum_{i=1}^m R_{ij} \times reputation_j \quad (11)$$

s. t.

$$software \& platform_i \subseteq support_set_j, \forall i, \forall j \quad (12)$$

$$R_{ij}(s_time_i - start_time_i) = 0, \forall i, \forall j \quad (13)$$

$$R_{ij}(e_time_i - end_time_i) = 0, \forall i, \forall j \quad (14)$$

$$R_{ij}(finish_time_i - deadline_i) \leq 0, \forall i, \forall j \quad (15)$$

$$R_{ij}(bid_price_i - ask_price_{ij}) \geq 0, \forall i, \forall j \quad (16)$$

$$R_{ij}(reputation_j - r_reputation_i) \geq 0, \forall i, \forall j \quad (17)$$

$$R_{ij}(reputation_i - r_reputation_j) \geq 0, \forall i, \forall j \quad (18)$$

$$\sum_{j=1}^n \lceil R_{ij} \rceil \leq partition_i, \forall i \quad (19)$$

$$R_{ij} \in [\epsilon, 1] \cup \{0\}, \forall i, \forall j \quad (20)$$

$$\sum_{i=1}^m R_{ij} \times R_i(t) \leq q_j(R, t), \forall i, \forall j \quad (21)$$

$$\sum_{j=1}^n R_{ij} \in \{0, 1\}, \forall i \quad (22)$$

公式(9)表示最大化市场总盈余 $total_surplus$. 公式(10)表示最大化单位时间市场盈余 $unit_surplus$, 这可以兼顾任务的优先级, 使 $bid_price/length$ 值较大的买方标的胜标机会更大. 公式(11)表示最大化整体威望 $total_reputation$. 公式(12)是软件 and 平台约束, 表示对虚拟机服务 CSC_i 要求的软件 and 平台 $software \& platform_i$ 必须是 CSP_j 支持集合 $support_set_j$ 的子集. 公式(13)~公式(15)是时间约束, 其中, 公式(13)和公式(14)对应虚拟机服务和存储服务, 表示 CSC_i 任务的开始时间 s_time_i 和结束时间 e_time_i 须与 CSC_i 标的中要求的开始时间 $start_time_i$ 和结束时间 end_time_i 相同; 公式(15)对应计算密集型和 I/O 密集型服务, 表示 CSC_i 的任务需在最后期限 $deadline_i$ 前完成, 其中 $finish_time_i$ 表示 CSC_i 任务的完成时间. 公式(16)是价格约束, 表示 CSC_i 的投标价格要不低于 CSP_j 的报价. 公式(17)和公式(18)是威望约束, 分别表示 CSP 的威望 $reputation_j$ 和 CSC 的威望 $reputation_i$ 必须满足对方对威望的最低要求. 公式(19)和公式(20)是分割约束, 其中, 公式(19)表示 CSC_i 的任务最多由 $partition_i$ 个 CSP 分割执行; 公式(20)表示 CSC_i 的任务分割粒度不能过细, R_{ij} 不能小于 ϵ . 公式(21)是资源约束, 表示所有 CSC_i 的资源需求要小于 CSP_j 所能提供的资源量, 其中, $R_i(t)$ 表示在 t 时刻 CSC_i 所需资源 R 的量, $q_j(R, t)$ 是 CSP_j 在 t 时刻能够提供资源 R 的量. 公式(22)是服务约束, 表示 CSC_i 的任务要么完全分配, 要么全不分配.

4 竞标价格决策机制

引入竞标价格决策机制的目的在于让 PA 和 CA 合理决策, 增加双方收益. 影响最终竞价的因素是复杂而多变的, 难以找到全面考虑所有因素的竞价计算公式. 而理论已经证明, 任意函数都可由一个三层神经网络以任意精度逼近^[31]. 因此, 本文引入 BP 神经网络求解该问题. 首先为样本中各参数设定参照系, 例如, 买卖双方的出价和报价参照亚马逊云平台^[32]等的资源标价; 根据各参数的参照系, 设定参数值的上下界; 令样本参数在对应的上下界间随机取值, 形成初始样本. 在随机产生初始样本时, 设置如下过滤规则: 买方出价不能高于预算(R1), 卖方报价不能低于成本(R2), 最迟开始时间不能早于拍卖系统当前时间(R3), 当前时段不能早于拍卖系统当前时间(R4), 当前负载不能超过系统满载容量(R5)等. 通过这些过滤规则, 自动剔除不合理的样本, 合理的样本作为训练样本加入到训练样本集中. 若合理的样本数少于训练所需最少样本数, 则重复以上过程, 直至训练样本集中的样本数不少于训练所需最少样本数. 使用训练样本集对 BP 神经网络进行训练, 保存训练好的 BP 神经网络. 使用训

练好的 BP 神经网络求得买卖双方的竞标价.竞标结束后,将竞标价及其对应的输入参数组合作为历史竞标数据加入到训练样本集中,作为继续训练的样本.

4.1 市场供求关系预测

市场供求关系是影响双方竞标价格的因素之一,可以根据前 $k-1$ 次交易中的供求关系预测第 k 次交易时的供求关系情况,以便于 PA 和 CA 进行竞价决策.

本文采用一次指数平滑法预测市场供求关系.第 k 次的平滑预测值 SA_k 如公式(23)所示:

$$SA_k = \psi sd_{k-1} + (1-\psi)SA_{k-1} \tag{23}$$

递推可得:

$$SA_k = \psi \sum_{z=0}^{k-1} (1-\psi)^z sd_{k-z-1} + (1-\psi)^k SA_0 \tag{24}$$

其中, $\psi(0 \leq \psi \leq 1)$ 是平滑系数; sd_k 是第 k 次的实际供求关系; sd_0 和 SA_0 均设为 1,即,假设初始时市场供求关系是平衡的.从公式(24)可以看出,第 k 次的市场供求关系预测值是前 $k-1$ 次实际供求关系的指数加权.

4.2 CA 竞标价格决策

CA 利用 BP 神经网络进行竞价决策时,用于训练神经网络的样本可以形式化表示为

$$Sample_{CA} = \{(supply_demand, latest_start_time, budget, current_period, reputation, risk_preference), bid_price\}.$$

该样本是由一组输入和一个输出组成.输出参数即 CA 通过 BP 决策的出价,而输入参数作为影响 CA 决策出价时考虑的主要因素,包括:

- (1) *supply_demand*(供求关系):若供大于求,则适当降低出价;反之,则提高出价.
- (2) *latest_start_time*(最迟开始时间):此时间与当前时间间隔短,代表需求急,出价应高.
- (3) *budget*(预算):出价的上限.
- (4) *current_period*(当前时段):高峰时段的出价要高.
- (5) *reputation*(自身威望):自身威望值高,中标机会大,出价可低.
- (6) *risk_preference*(风险偏好):冒险者比避险者出价低.

竞标成功的价格信息直接被当作样本数据,用于训练神经网络.

4.3 PA 竞标价格决策

当 PA 利用 BP 神经网络进行竞标价格决策时,用于训练神经网络的样本可以形式化表示为 $Sample_{PA} = \{(supply_demand, current_load, cost, current_period, reputation, risk_preference), ask_price\}$,与 $Sample_{CA}$ 的结构类似.输出参数即 PA 通过 BP 决策的报价,而输入参数作为影响 PA 决策报价时考虑的主要因素,除了 *current_load* (当前负载)和 *cost*(成本)之外,其余 4 个参数的含义与 $Sample_{CA}$ 中的相同.

PA 与 CA 决策时的不同之处在于:由公式(22)可知,对于 CA 提交的买方标的,要么完全满足,要么全不满足;但是对于 PA 提交的卖方标的,允许出现部分满足的情况,即,只交易 PA 所拥有资源的一部分.竞标成功的价格信息直接存入样本数据库,用于训练神经网络.

5 胜标确定方法

双向组合拍卖的胜标确定问题已被证明是 NP 完全的^[33],需要采用启发式或者智能优化算法求解,寻找最优解或者次优解.基于此,本文选用 GSO 求解胜标确定问题.

5.1 解的优胜数

设 X_a 是基准解,则解 X_b 对 X_a 的优胜数定义如下:

$$A_{b,a} = \alpha \frac{TS(X_b) - TS(X_a)}{TS(X_a)} + \beta \frac{US(X_b) - US(X_a)}{US(X_a)} + \gamma \frac{TR(X_b) - TR(X_a)}{TR(X_a)} \tag{25}$$

其中, TS, US 和 TR 分别表示 *total_surplus, unit_surplus* 和 *total_reputation*; $\alpha + \beta + \gamma = 1$,表示 3 个指标各自的权重.

5.2 约束调整

资源分配方案只有满足约束条件才有意义,因此,求解时需在每次迭代后都根据约束条件对资源分配矩阵的值进行必要的调整:

- (1) 根据软件 and 平台约束、时间约束、价格约束和威望约束进行调整.如果矩阵某元素的值不为 0,且该元素对应的分配方案不满足约束,则把该元素上的值移至同行其他可满足相应约束的位置,如果没有这样的位置,则将该元素置 0.
- (2) 根据分割约束进行调整.如果矩阵的某一行存在不为 0 且小于 ε 的元素,或不为 0 的元素数量多于 $partition_i$,则令 R_i 和 $R_{i\#}$ 分别表示该行元素的最小值和次小值,令 $R_{i\#} = R_{i\#} + R_i$, $R_i = 0$, 重复该步骤,直至满足分割约束.
- (3) 根据资源约束进行调整.CSP 的资源需满足 CSC 任务的要求,对于违反资源约束的元素,把该元素上的值移至同行其他可满足资源约束的位置,如果没有这样的位置,则将该元素置 0;
- (4) 根据服务约束进行调整.如果矩阵某一行元素之和不为 1,则该行元素全部置 0.

5.3 群搜索优化

5.3.1 基本思想

GSO 是受自然界中群居动物搜索目标的过程启发而提出的一种群体智能算法.动物的搜索行为在寻找食物、伴侣或筑巢时经常用到,是群体动物共同参与的一种重要行为.搜索的成功依赖于 3 个方面:首先是搜索策略和资源在时间和空间上的分布;其次是个体定位资源的能力;最后是种群适应长期或短期环境变化的能力.本文采用的搜索策略把种群内个体分为领导者、跟随者和游走者,3 种角色分别具有不同的行为,相互配合共同完成搜索任务.

5.3.2 关键步骤

在 GSO 中,每个个体是群体中的一个成员,在 c 维搜索空间中,第 y 次迭代中的第 g 个成员的位置为 X_g^y , 仰角 $\phi_g^y = (\phi_{g_1}^y, \phi_{g_2}^y, \dots, \phi_{g_{c-1}}^y)$, 第 g 个成员的搜索方向为 $D_g^y(\phi_g^y) = (d_{g_1}^y, d_{g_2}^y, \dots, d_{g_c}^y)$. 仰角和搜索方向向量的转换可以根据笛卡尔坐标转换公式来计算,如公式(26)~公式(28)所示:

$$d_{g_1}^y = \prod_{q=1}^{c-1} \cos(\phi_{g_q}^y) \quad (26)$$

$$d_{g_h}^y = \sin(\phi_{g_{(h-1)}}^y) \prod_{q=h}^{c-1} \cos(\phi_{g_q}^y), h = (2, \dots, c-1) \quad (27)$$

$$d_{g_c}^y = \sin(\phi_{g_{(c-1)}}^y) \quad (28)$$

(1) 领导者行为

每一代中的领导者 X_p 是所有个体中对基准个体的优胜数最高的个体,其当前位置为 X_p^y . 领导者主要执行寻优操作,首先搜索正前方范围,然后在正前方的右侧任选一个方向搜索,最后是左侧方向搜索.领导者搜索时的最大偏角为 θ_{\max} ,最大的搜索距离为 l_{\max} . 公式(29)~公式(31)分别表示领导者向正前方搜索、向右侧任选一个方向搜索以及向左侧任选一个方向搜索.

$$X_z = X_p^y + r_1 l_{\max} D_p^y(\phi^y) \quad (29)$$

$$X_{right} = X_p^y + r_1 l_{\max} D_p^y(\phi^y + r_2 \theta_{\max} / 2) \quad (30)$$

$$X_{left} = X_p^y + r_1 l_{\max} D_p^y(\phi^y - r_2 \theta_{\max} / 2) \quad (31)$$

其中, r_1 服从 $N(0,1)$ 分布, r_2 服从 $U(0,1)$ 分布.

领导者在这 3 个方向中找到最好(适宜度最高)的位置,如果优于当前位置,则移动到该位置;否则,继续停留在当前的位置并产生一个新的仰角 $\phi^{y+1} = \phi^y + r_2 \eta_{\max}$, η_{\max} 为最大的转角.如果领导者在 ω 次迭代之后仍然找不到更好的位置,它就将仰角调到 0° .

(2) 跟随者行为

除了领导者之外,在种群中随机选择多个个体作为跟随者.跟随者向领导者方向靠拢搜索,追寻领导者搜索过的足迹,其位置更新方法如公式(32)所示:

$$X_g^{y+1} = X_g^y + r_3(X_p^y - X_g^y) \tag{32}$$

其中, r_3 服从 $U(0,1)$ 分布.

(3) 游走者行为

除领导者和跟随者之外,其余成员充当游走者角色,其搜索策略为随机游走.

首先产生一个随机距离 $l_g = \omega r_1 l_{max}$,然后向该位置移动,位置更新方法如公式(33)所示:

$$X_g^{y+1} = X_g^y + l_g D_g^y(\phi^{y+1}) \tag{33}$$

5.3.3 算法流程

- Step 1. 初始化群体成员数量、最大迭代次数、跟随者和游走者比例等参数.
- Step 2. 随机生成初始资源分配矩阵,并根据第 5.2 节进行必要的调整,随机选出其一作为基准个体.
- Step 3. 计算所有个体对基准个体的优胜数,根据优胜数对所有个体进行排序,选出领导者.
- Step 4. 如果当前迭代次数超过最大迭代次数,则转 Step 9.
- Step 5. 根据公式(29)~公式(31)执行领导者行为.
- Step 6. 根据跟随者和游走者的比例随机选出跟随者,根据公式(32)执行跟随者行为.
- Step 7. 根据公式(33)执行游走者行为.
- Step 8. 根据第 5.2 节对资源分配矩阵进行必要的调整,转 Step 3.
- Step 9. 输出领导者作为问题解,算法结束.

6 仿真实现与性能评价

6.1 仿真环境设置

本文基于 SimJava2.0 工具包,使用 JDK1.6 进行仿真实现.服务类型参照亚马逊云计算平台^[32]设定,各种资源的处理能力参照 TeraGrid 各合作单位资源列表^[34]设定,价格参照亚马逊云平台^[32]的资源标价设定;BP 竞价决策机制的参数参照文献[35]设定,见表 1;GSO 的参数参照文献[27]设定,见表 2.

Table 1 Parameter setting of BP bidding price decision mechanism
表 1 BP 竞价决策机制参数设置

参数	设定值	说明
<i>theta</i>	0.5	训练网络的学习因子
<i>hiddenNodeNum</i>	3	隐层节点数
<i>initialSampleNum</i>	30	随机产生初始样本数
<i>neededSampleNum</i>	15	训练所需最少样本数
<i>trainTimes</i>	10	训练次数

Table 2 Parameter setting of GSO
表 2 GSO 参数设置

参数	设定值	说明
<i>iterations</i>	20	迭代次数
<i>animalNum</i>	20	种群规模
<i>SCR_RATE</i>	0.3	游走者与跟随者比例
<i>iDim</i>	$m \times n$	资源分配矩阵大小
ω	$\lceil \text{Sqrt}(iDim) \rceil$	每个方向上试探的次数
θ_{max}	$PI(\omega \times \omega)$	搜索的最大偏角
l_{max}	$iDim \times iDim$	搜索的最大距离

本文进行性能对比选用的基准机制是文献[5]提出的稳定连续双向拍卖(stable continuous double auction,简

称 SCDA)机制,对比的数据都是在指定情形下每种机制运行 20 次取平均值得到的.后文图 3~图 8 中,纵坐标的值是把得到数据中最优的一组标记为 1,其余数据均为相对值.本文对仿真中的任务粒度、市场供求关系和市场规模分别进行了划分,以便于在各种情形下更全面地验证机制的性能,客观地进行对比分析.按照任务对资源的需求量,把任务分为小粒度、正常粒度和大粒度这 3 种类型.在微观经济学理论中,供求关系作为影响买卖双方行为和市场价格的关键因素,通常可以根据市场供给和需求之间是否失衡分为供不应求、供求均衡和供大于求,本文以此为基础,额外考虑资源极其充足的极端情况,将供求关系划分为 4 种类型,并参照文献[36]对具体数值进行设定,见表 3.在经济学理论中,市场规模表示购买者人数,对于以拍卖方式组织而成的虚拟云市场,由于作为购买者的 CSC 可以提交多个任务,因此本文以任务数而非 CSC 数作为划分市场规模的依据.在经济学中,一般将市场规模划分为大、中、小这 3 种类型,为了更精细地分析本文机制的性能,以此为基础,将市场规模细分为 6 种类型,并参照文献[37]对仿真时的任务数和 CSP 数进行设置,见表 4.公式(23)和公式(24)中的平滑系数 ρ 参照文献[38],设为 0.5.

Table 3 Supply-Demand relation division

表 3 供求关系划分

供求关系	仿真供求比取值
供小于求	[0.4,0.9)
供求平衡	[0.9,1.1]
供大于求	(1.1,2]
资源充足	(2,4)

Table 4 Market scale division

表 4 市场规模划分

市场规模	仿真任务数和 CSP 数设置
小规模	8 任务,4CSP
较小规模	16 任务,4CSP
中等规模	32 任务,4CSP
较大规模	64 任务,4CSP
大规模	128 任务,4CSP
超大规模	128 任务,8CSP

6.2 不同供求关系下的性能对比

固定任务粒度为正常粒度,市场规模为较大规模,图 3 和图 4 是本文机制(以下简称 DCA_GSO)和 SCDA 在不同供求关系下市场总盈余和单位时间市场盈余的对比情况.

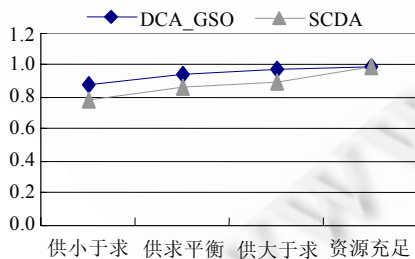


Fig.3 Comparison on total market surplus with different supply-demand relations

图 3 不同供求关系下市场总盈余对比

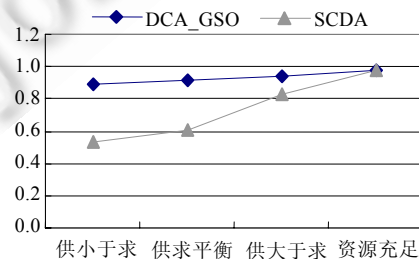


Fig.4 Comparison on unit time market surplus with different supply-demand relations

图 4 不同供求关系下单位时间市场盈余对比

从图 3 可以看出,DCA_GSO 的性能总体优于 SCDA.在有资源约束的条件下,DCA_GSO 性能要好于 SCDA,而且约束越大,差距越明显;仅在资源充足的条件下,两者性能基本持平,这是由于资源充足时 DCA_GSO 的性能

受限于 GSO 的随机性,难以保证不出现任务“分割”情况.图 4 和图 3 情况类似,只是两者之间差距更大一些,这是因为在 DCA_GSO 下服务可分割而并行执行,所需时间更短.

6.3 不同任务粒度下的性能对比

固定市场规模为较大规模,市场供求关系为供求平衡,图 5 和图 6 是两种机制在不同任务粒度下的市场总盈余和单位时间市场盈余的对比情况.

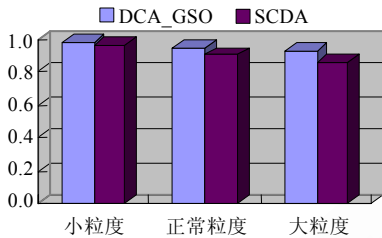


Fig.5 Comparison on total market surplus with different task granularities

图 5 不同任务粒度下市场总盈余对比

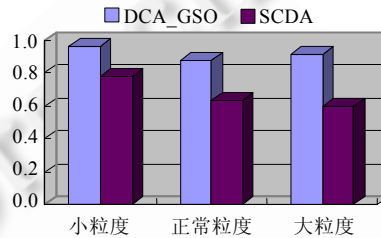


Fig.6 Comparison on unit time market surplus with different task granularities

图 6 不同任务粒度下单位时间市场盈余对比

从图 5 可以看出:在 3 种任务粒度下,DCA_GSO 的性能都优于 SCDA;而且随着任务粒度增大,性能差距也越大.主要原因是:由于任务粒度增大,分配时违反资源约束的机会就会增大,CSC 所需求的服务被分配到单一 CSP 时更容易违背约束.从图 6 可以看出,DCA_GSO 与 SCDA 单位时间市场总盈余差距更大.这同样是因为在 DCA_GSO 下,任务可以被多个 CSP 分割而同时执行,所需时间缩短,产生的单位时间市场盈余自然要比 SCDA 高很多.

6.4 不同市场规模下的性能对比

固定任务粒度为正常粒度,市场供求关系为供求平衡,图 7 和图 8 是两种机制在不同市场规模下的市场总盈余和单位时间市场盈余的对比情况.

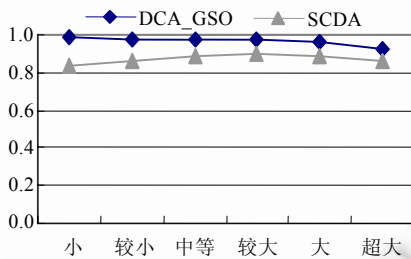


Fig.7 Comparison on total market surplus with different market scales

图 7 不同市场规模下市场总盈余对比

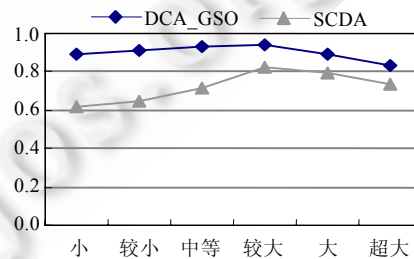


Fig.8 Comparison on unit time market surplus with different market scales

图 8 不同市场规模下单位时间市场盈余对比

从图 7 和图 8 可以看出:在 6 种市场规模下,DCA_GSO 性能都优于 SCDA.但是,随着市场规模的扩大,DCA_GSO 性能呈略微下降趋势.这是因为随着问题规模的增大,GSO 求解组合优化问题的效率变低.

6.5 成功交易数对比

固定市场规模为超大规模,任务粒度为正常粒度,图 9 显示了两种机制在不同供求关系下成功交易数的对比情况.

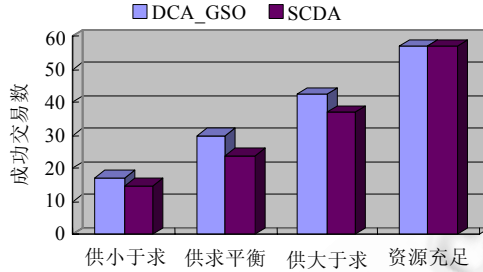


Fig.9 Comparison on successful transaction number with different supply-demand relations

图9 不同供求关系下成功交易数对比

从图9可以看出:在资源充足时,两者的交易成功数相同,达到最大值;在其他供求关系下,DCA_GSO的成功交易数更多,这是因为它能对任务进行分割,更加高效地利用资源,从而使更多的需求得到满足.

6.6 时间开销对比

固定任务粒度为正常粒度,市场供求关系为供求平衡,图10是两种机制在不同市场规模下时间开销的对比情况.其中,图10(a)将DCA_GSO中神经网络的训练时间计入时间开销,图10(b)是对神经网络采用离线训练的情况下的时间开销比较,即DCA_GSO的时间开销不包括神经网络的训练时间.

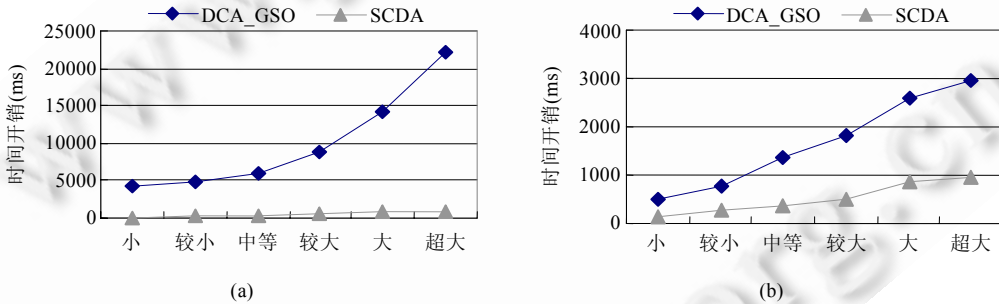


Fig.10 Comparison on runtime with different market scales

图10 不同市场规模下的时间开销对比

从图10(a)和10(b)可以看出,DCA_GSO的时间开销高于SCDA.这是因为随着市场规模的扩大,用于训练神经网络以及智能算法优化胜标确定的时间会明显增加.但是,云环境下的资源分配通常对即时性的要求不是很严格,而且当获取到足够的训练样本以后,系统可以采用离线方式训练神经网络,在线直接使用神经网络竞价,这样,DCA_GSO的时间开销会大为降低,如图10(b)所示.

综合考虑上述对本文机制的性能分析以及与基准机制之间的性能对比结果可以看出:在不同供求关系、任务粒度、市场规模下,本文机制的性能总体优于基准机制,但时间开销较大;如果采用离线方式训练神经网络,则本文机制的时间开销可显著降低.基准机制虽然性能表现明显不如本文机制,但时间开销较低.本文机制在性能上所表现出的优越性主要是由于采用了利用历史竞标数据的竞价决策机制和智能的胜标确定方法,这也是本文的机制与相关研究工作相比最大的不同点.

7 结论

云环境基于网络将分散的资源聚合起来,为用户提供服务.如何高效地分配资源和提供 QoE 保障等问题亟待解决.本文提出并仿真实现了一种云资源分配机制.基于双向组合拍卖机制,资源提供者和消费者在标的中分别标注相关信息,综合考虑各种影响竞标价格的因素,通过智能代理,利用基于 BP 神经网络的竞标价格决策机

制,合理确定竞标价.为了向用户提供 QoE 支持,本文提出了基于 QoE 的威望系统,引入了威望衰减系数和用户信誉度,降低恶意行为对资源分配的不利影响.拍卖中介协调整个拍卖流程,匹配双方标的,通过多轮拍卖,确定供需方之间的资源成交价.以此为基础,考虑市场总盈余,单位时间市场盈余和总体威望等目标进行胜标确定.由于双向组合拍卖的胜标确定问题是 NP 完全的,因此,本文基于 GSO 求解该问题,得到优化的资源分配方案.仿真研究结果表明,该机制是可行和有效的.对本文提出的机制进行原型实现,并在实际系统中加以试用,以使其实用化,是我们今后研究与开发工作的重点.

References:

- [1] Rehr JJ, Vila FD, Gardner JP, Svec L, Prange M. Scientific computing in the cloud. *Computing in science & Engineering*, 2010, 12(3):34–43. [doi: 10.1109/MCSE.2010.70]
- [2] Niyato D, Chaisiri S, Lee BS. Economic analysis of resource market in cloud computing environment. In: *Proc. of the 2009 IEEE Asia-Pacific Services Computing Conf. (APSCC 2009)*. Piscataway: IEEE Computer Society, 2009. 156–162. [doi: 10.1109/APSCC.2009.5394127]
- [3] Buyya R, Abramson D, Venugopal S. The grid economy. *Proc. of the IEEE*, 2005,93(3):698–714. [doi: 10.1109/JPROC.2004.842784]
- [4] Grosu D, Das A. Auctioning resources in grids: Model and protocols. *Concurrency and Computation: Practice & Experience*, 2006, 18(15):1909–1927. [doi: 10.1002/cpe.1037]
- [5] Tan Z, Gurd JR. Market-Based grid resource allocation using a stable continuous double auction. In: *Proc. of the 8th IEEE/ACM Int'l Conf. on Grid Computing (GRID 2007)*. Los Alamitos: IEEE Computer Society, 2007. 283–290. [doi: 10.1109/GRID.2007.4354144]
- [6] Xia QF, Sun WF, Xu ZC, Li MC. A novel grid resource scheduling model based on extended second price sealed auction. In: *Proc. of the 3rd Int'l Symp. on Parallel Architectures, Algorithms and Programming (PAAP 2010)*. Piscataway: IEEE Computer Society, 2010. 305–310. [doi: 10.1109/PAAP.2010.49]
- [7] Qureshi K, Nazir B, Shah MA. Markup based continuous double auction for resource allocation in market grid. *Engineering e-Transaction*, 2011,6(1):50–54.
- [8] Danak A, Mannor S. Efficient bidding in dynamic grid markets. *IEEE Trans. on Parallel and Distributed Systems*, 2011,22(9): 1483–1496. [doi: 10.1109/TPDS.2011.29]
- [9] Li L, Liu YA, Ma XL. Grid resource allocation based on the combinatorial double auction. *Acta Electronica Sinica*, 2009,37(1): 165–169 (in Chinese with English abstract).
- [10] Weng CL, Lu XD. A double auction method for resource allocation on computational grids. *Chinese Journal of Computers*, 2006, 29(6):1004–1009 (in Chinese with English abstract).
- [11] Li MC, Xu L, Sun WF, Lu K, Guo C. Grid resource allocation model based on incomplete information game. *Ruan Jian Xue Bao/ Journal of Software*, 2012,23(2):428–438 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3972.htm> [doi: 10.3724/SP.J.1001.2012.03972]
- [12] Wang XW, Sun JJ, Li HX, Wu C, Huang M. A reverse auction based allocation mechanism in the cloud computing environment. *Applied Mathematics & Information Sciences*, 2013,7(1L):75–84.
- [13] Wang XW, Wang XY, Huang M. A resource allocation method based on the limited English combinatorial auction under cloud computing environment. In: *Proc. of the 2012 9th Int'l Conf. on Fuzzy Systems and Knowledge Discovery (FSKD 2012)*. Washington: IEEE Computer Society, 2012. 905–909. [doi: 10.1109/FSKD.2012.6234231]
- [14] Lin WY, Lin GY, Wei HY. Dynamic auction mechanism for cloud resource allocation. In: *Proc. of the 10th IEEE/ACM Int'l Conf. on Cluster, Cloud, and Grid Computing (CCGrid 2010)*. Piscataway: IEEE Computer Society, 2010. 591–592. [doi: 10.1109/CCGR ID.2010.92]
- [15] Prodan R, Wiczorek M, Fard HM. Double auction-based scheduling of scientific applications in distributed grid and cloud environments. *Journal of Grid Computing*, 2011,9(4):531–548. [doi: 10.1007/s10723-011-9196-x]

- [16] Shang SF, Jiang JL, Wu YW, Huang ZC, Yang GW, Zheng WM. DABGPM: A double auction Bayesian game-based pricing model in cloud market. In: Proc. of the IFIP Int'l Conf. on Network and Parallel Computing (NPC 2010). Heidelberg: Springer-Verlag, 2010. 155–164. [doi: 10.1007/978-3-642-15672-4_14]
- [17] Sun DW, Chang GR, Wang C, Xiong Y, Wang XW. Efficient Nash equilibrium based cloud resource allocation by using a continuous double auction. In: Proc. of the 2010 Int'l Conf. on Computer Design and Applications (ICCD 2010). Piscataway: IEEE Computer Society, 2010. 94–99. [doi: 10.1109/ICCD.2010.5541088]
- [18] Shang SF, Jiang JL, Wu YW, Yang GW, Zheng WM. A knowledge-based continuous double auction model for cloud market. In: Proc. of the 6th Int'l Conf. on Semantics, Knowledge and Grid (SKG 2010). Washington: IEEE Computer Society, 2010. 129–134. [doi: 10.1109/SKG.2010.22]
- [19] Vilajosana X, Lázaro D, Juan AA, Marqués JM. A multi-lane double auction for economic-based service management in the cloud. In: Proc. of the Intelligent Networking, Collaborative Systems and Applications. Heidelberg: Springer-Verlag, 2011. 117–148. [doi: 10.1007/978-3-642-16793-5_6]
- [20] Roovers J, Vanmechelen K, Broeckhove J. A reverse auction market for cloud resources. In: Proc. of the 8th Int'l Workshop on Economics of Grids, Clouds, Systems, and Services (GECON 2011). Heidelberg: Springer-Verlag, 2012. 32–45. [doi: 10.1007/978-3-642-28675-9_3]
- [21] Tsai CW, Tsai Z. Bid-Proportional auction for resource allocation in capacity-constrained clouds. In: Proc. of the 26th IEEE Int'l Conf. on Advanced Information Networking and Applications Workshops (WAINA 2012). Piscataway: IEEE Computer Society, 2012. 1178–1183. [doi: 10.1109/WAINA.2012.192]
- [22] Zaman S, Grosu D. Combinatorial auction-based allocation of virtual machine instances in clouds. *Journal of Parallel and Distributed Computing*, 2013,73(4):495–508. [doi: 10.1016/j.jpdc.2012.12.006]
- [23] Fujiwara I, Aida K, Ono I. Combinatorial auction-based marketplace mechanism for cloud service reservation. *IEICE Trans. on Information and Systems*, 2012,95(1):192–204. [doi: 10.1587/transinf.E95.D.192]
- [24] Hu ZG, Liu Y. Dynamic resource pricing based on combinatorial double auction in cloud environment. *Computer Engineering*, 2012,38(8):19–21 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-3428.2012.08.007]
- [25] Ding D, Luo SW, Ai LH. Adaptive double auction mechanism for cloud resource allocation. *Journal on Communications*, 2012, 33(Z1):132–140 (in Chinese with English abstract).
- [26] Wang XW, Wang XY, Huang M. A resource allocation model based on double auction under cloud computing environment. *Journal of Chinese Computer Systems*, 2013,34(2):309–316 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-1220.2013.02.024]
- [27] He S, Wu QH, Saunders JR. Group search optimizer: An optimization algorithm inspired by animal searching behavior. *IEEE Trans. on Evolutionary Computation*, 2009,13(5):973–990. [doi: 10.1109/TEVC.2009.2011992]
- [28] Robert HN. Theory of the backpropagation neural network. In: Proc. of the Int'l Joint Conf. on Neural Network (IJCNN'89). Piscataway: IEEE Computer Society, 1989. 593–605. [doi: 10.1109/IJCNN.1989.118638]
- [29] Endo PT, de Almeida Palhares AV, Pereira NN, Goncalves GE, Sadok D, Kelner J, Melander B, Mangs JE. Resource allocation for distributed cloud: Concepts and research challenges. *IEEE Network*, 2011,25(4):42–46. [doi: 10.1109/MNET.2011.5958007]
- [30] Chard K, Caton S, Rana O, Bubendorfer K. Social cloud: Cloud computing in social networks. In: Proc. of the 2010 IEEE 3rd Int'l Conf. on Cloud Computing (CLOUD 2010). Piscataway: IEEE Computer Society, 2010. 99–106. [doi: 10.1109/CLOUD.2010.28]
- [31] Hornik K, Stinchcombe M, White H. Multilayer feed forward networks are universal approximators. *Neural Networks*, 1989,2(5): 359–366. [doi: 10.1016/0893-6080(89)90020-8]
- [32] Amazon Web services. <http://aws.amazon.com/>
- [33] Rothkopf MH, Pekeč A, Harstad RM. Computationally manageable combinatorial auctions. *Management Science*, 1998,44(8): 1131–1147. [doi: 10.1287/mnsc.44.8.1131]
- [34] Teragrid compute and visualization resources. https://www.teragrid.org/web/user-support/compute_resources/
- [35] Subudhi B, Jena D. Nonlinear system identification using memetic differential evolution trained neural networks. *Neurocomputing*, 2011,74(10):1696–1709. [doi: 10.1016/j.neucom.2011.02.006]

- [36] Sim KM. Towards a unifying multilateral cloud negotiation strategy. In: Proc. of the Int'l Multi Conf. of Engineers and Computer Scientists 2013 (IMECS 2013). Hong Kong: Newswood Limited, 2013. 362–367.
- [37] Xu H, Li B. Anchor: A versatile and efficient framework for resource management in the cloud. IEEE Trans. on Parallel and Distributed Systems, 2012,24(6):1066–1076. [doi: 10.1109/TPDS.2012.308]
- [38] Mi HB, Wang HM, Yin G, Zhou YF, Shi DX, Yuan L. Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers. In: Proc. of the 2010 IEEE 7th Int'l Conf. on Services Computing (SCC 2010). Piscataway: IEEE Computer Society, 2010. 514–521. [doi: 10.1109/SCC.2010.69]

附中文参考文献:

- [9] 李立,刘元安,马晓雷.基于组合双向拍卖的网格资源分配.电子学报,2009,37(1):165–169.
- [10] 翁楚良,陆鑫达.一种基于双向拍卖机制的计算网格资源分配方法.计算机学报,2006,29(6):1004–1009.
- [11] 李明楚,许雷,孙伟峰,陆坤,郭成.基于非完全信息博弈的网格资源分配模型.软件学报,2012,23(2):428–438. <http://www.jos.org.cn/1000-9825/3972.htm> [doi: 10.3724/SP.J.1001.2012.03972]
- [24] 胡志刚,刘艳.云环境下基于组合双向拍卖的动态资源定价.计算机工程,2012,38(8):19–21. [doi: 10.3969/j.issn.1000-3428.2012.08.007]
- [25] 丁丁,罗四维,艾丽华.基于双向拍卖的适应性云计算资源分配机制.通信学报,2012,33(Z1):132–140.
- [26] 王兴伟,王学毅,黄敏.云计算环境下一种基于双向拍卖的资源分配模型.小型微型计算机系统,2013,34(2):309–316. [doi: 10.3969/j.issn.1000-1220.2013.02.024]



孙佳佳(1988—),女,山东淄博人,硕士,主要研究领域为云环境资源管理.
E-mail: sunplusplus@163.com



高程希(1990—),男,硕士,主要研究领域为云环境资源管理.
E-mail: gaocxresearch@gmail.com



王兴伟(1968—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为未来互联网,云计算.
E-mail: wangxw@mail.neu.edu.cn



黄敏(1968—),女,博士,教授,博士生导师,主要研究领域为物流与供应链管理.
E-mail: mhuang@mail.neu.edu.cn