

## 基于最佳匹配拍卖的企业级网络资源分配策略

丛鑫, 訾玲玲, 沈学利

(辽宁工程技术大学电子与信息工程学院, 辽宁 葫芦岛 125105)

**摘 要:** 针对企业级网络中计算机拥有者的自私属性导致网络中可用资源节点数量不足和资源分配效率低的问题, 提出一种以拍卖机制为核心的企业级网络最优匹配资源分配 (OMRA) 策略, 在抑制节点自私性的同时提升网络可用资源数, 进而提升整个拍卖市场的运行效率。首先, 归一化不同类型资源的成本, 确定拍卖初始的资源请求价格; 其次, 设计了最优化匹配的拍卖算法, 最大化拍卖市场的收益; 再次, 运行服务请求预取算法, 使资源提供者能以当前的成交价格获取更多的任务请求, 保证资源提供者的收益; 最后, 利用请求价格和竞拍价格更新算法, 保证买卖双方在下一次拍卖过程中获得较高的优先级, 以获取更多的收益。实验结果表明, 与基于拍卖的云资源分配算法 (CRAA/FA) 相比, 所提 OMRA 策略能提升 10% 的资源分配效率和 11.4% 的市场收益率。

**关键词:** 最佳匹配拍卖; 服务请求预取; 企业级网络; 资源分配

**中图分类号:** TP393

**文献标识码:** A

**doi:** 10.11959/j.issn.1000-436x.2019186

## Resource allocation strategy based on optimal matching auction in the enterprise network

CONG Xin, ZI Lingling, SHEN Xueli

School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China

**Abstract:** To address the issue that the owners of computer are selfish in the enterprise networks, which caused the low available number of resource nodes and low efficiency of resource allocation, an optimized matching resource allocation strategy OMRA was proposed and its core was the auction mechanism. Selfishness was restrained and the number of available resources was increased by OMRA, so as the operating efficiency of the whole auction market was improved. First, the initial prices were determined by normalizing the costs of different type of resources on the beginning of auction. Secondly, an optimal matching auction algorithm was designed to maximize the interests of the auction markets. Then, service perfecting algorithm was performed such that the sellers could get more services at the current transaction value, thus ensuring the benefits of resource providers. At last, a request price updating algorithm was adopted to assurance that both sellers and buyers could get priorities in the next auction processing. Compared with the cloud resource allocating algorithm via fitness-enabled auction (CRAA/FA), the experiment results indicate that the efficiency of resource allocation improves by 10% and the benefits of market increase by 11.4%.

**Key words:** optimal matching auction, service prefetching, the enterprise network, resource allocation

收稿日期: 2018-02-05; 修回日期: 2019-07-09

通信作者: 訾玲玲, lingling19812004@126.com

基金项目: 国家自然科学基金资助项目 (No.61602227, No.61702241); 辽宁省教育厅科学研究一般项目 (No.LJYL019); 辽宁省科技厅博士启动基金计划资助项目 (No.201601365)

**Foundation Items:** The National Natural Science Foundation of China (No.61602227, No.61702241), The Foundation of Liaoning Educational Committee (No.LJYL019), The Doctoral Starting up Foundation of Science Project of Liaoning Province (No.201601365)

## 1 引言

大型企业需要数据中心来分析和处理每天产生的大量数据。现阶段，构建数据中心可以采用租用云计算平台和企业自建2种方式，但租用云计算平台有泄露数据隐私的风险，因此，企业往往更愿意建立自己的数据中心。通常大型企业会在不同地理位置建立子公司，但为每个子公司建立数据中心是巨大的资源浪费。另外，已有调查<sup>[1]</sup>表明，处于开启状态的计算机绝大部分时间都处于空闲状态。鉴于此，利用企业现有的计算机、服务器和网络构建数据中心是简单易行且投资较低的方案。

相比于云平台，企业级网络具有以下特征：1) 网络中的计算机等设备供员工日常使用，员工从完全占用计算机的使用权和不耽误日常工作的角度出发，不愿意贡献自己的计算机作为网络中任务处理的节点；2) 网络中的计算机等设备随着员工日常上下班而开启和关闭，可用资源数随着时间发生变化；3) 网络中的计算机等设备的性能不同，其能提供资源的数目和连续工作时间长度也不同，即提供服务的质量是有差异的。针对上述特征，当网络中有可用资源时，如何分配这些资源成为有意义的研究工作。已有资源分配方案，如资源数预测<sup>[2]</sup>、服务级别协议（SLA, service level agreement）<sup>[3]</sup>、工作流和虚拟机分配<sup>[4-5]</sup>等，都是建立在云平台内节点资源充足的情况下，而没有考虑节点所提供资源的差异性，这就需要研究新的适用方案。虚拟支付方案是解决节点激励和资源分配的有效途径，其中拍卖机制已经被证明是极其高效的<sup>[6]</sup>，可以关联销售者和购买者以获取商业化的收益，最终达到纳什均衡<sup>[7-8]</sup>。例如，文献[9-11]改进了连续双边拍卖方法，建立了新的资源分配模型，从销售者的角度实现一定的性能和经济服务质量。

为了使连续双边拍卖适应企业级网络环境，需要考虑以下几个方面：1) 满足资源提供者（销售者）和资源租用者（购买者）的可接受程度需求；2) 构建销售者虚拟支付市场，当销售者贡献资源时，给予报酬（虚拟收益），并设置消费机制；3) 提升销售者和购买者匹配程度，在拍卖开始时被标记为高计算能力的节点可能随着任务的运行变成低计算能力节点，此时应避免将有高计算能力需求任务分配到该节点上。

因此，本文研究的是如何以虚拟市场的方式分配企业级网络的计算资源和带宽资源以满足任务需求，达到较高的资源分配效率及市场收益率。在虚拟市场中，当销售者提供的资源和购买者需求有差异、资源价格浮动性较大时，利用基于市场规律的拍卖机制解决资源分配问题是有效的。企业级网络资源分配应达到以下3个目标：目标1，不同部门销售者之间的资源负载动态平衡；目标2，销售者和购买者的接受程度高；目标3，整个拍卖市场获取最佳收益。与云平台的资源节点可以全天24h运行服务不同，企业网的节点不能持续长时间的工作（工作时间仅为8~10小时/天）。因此，资源销售者感知更多的服务请求到来时，会在本轮拍卖过程中提高请求价格，以获取更多的收益；当发现剩余工作时间不充足（为1~2h）时，会降低请求价格，以获取更多的服务请求。然而，高价格会导致其他销售者更容易赢得拍卖，剩余工作时间少的资源销售者即使降低价格也可能因为剩余服务请求不多而达不到预期收益目标。换言之，服务请求以轻负载原则分配给销售者，满足目标1。

销售者给出的请求价格较低，则在拍卖过程中会获得较高的优先级，从而获得较多的服务请求。购买者如果有较多的预算，也会获得较高的优先级，从而获得所需的资源。那么，如何设计拍卖过程中的匹配方案才能满足上述3个目标？举例分析如下： $R_1$ 和 $R_2$ 都满足服务请求C所需的资源，但 $R_1$ 更适合C的运行，则当 $R_1$ 和 $R_2$ 在拍卖过程中给出相同的请求价格时， $R_1$ 和C的最终交易价格要低于 $R_2$ 和C的，这样 $R_1$ 才能赢得本轮拍卖并获取C，从而满足目标2；从整个市场的角度来看， $R_1$ 完成C的运行时间要短于 $R_2$ ，而较短的运行时间能够使一定时间内，市场能容纳的服务数更多，整体收益更高，从而满足目标3。

基于以上分析，本文提出最优匹配资源分配（OMRA, optimized matching resource allocation）策略，相比于已有的连续双边拍卖方案，OMRA策略额外关注了销售者和购买者的可接受程度，来衡量销售者和购买者的最终成交价格 and 预期价格的差异程度，可以激发销售者提供更多的可用资源，购买者在当前拍卖进程中放置更多的服务请求。

本文主要工作如下。

1) 设计了节点的资源归一化方法，以基准资源

节点数值为标准，分别计算高能力节点和高 I/O 能力节点资源数，从而计算出不同节点的运行成本，为拍卖过程中的竞拍价格提供依据。

2) 设计了销售者/购买者最佳匹配拍卖算法，以最大化整个拍卖市场的收益。设计了服务请求预留算法，使销售者以当前价格获取更多的服务器请求，进一步增加收益。

3) 设计了请求价格更新算法，引导销售者在拍卖过程中动态调整请求价格，以在下一轮拍卖中得到较高的优先级。

4) 搭建了由服务器和计算机组成的企业级网络模拟环境，系统评估了服务请求接受率、收益率、支付率、资源分配率和市场共享率等参数。

## 2 企业级网络资源拍卖模型和参数定义

拍卖市场有 3 个实体，分别是销售者、购买者和货物。虚拟市场的交易行为主要分为 3 个阶段，商品定价阶段、商品交易阶段和商品支付阶段。相应地，在企业级网络中，计算和带宽资源的拥有者是销售者，任务的发布者是购买者，资源是货物。本文重点关注虚拟市场的货物拍卖机制，为了避免网络的服务器成为网络瓶颈，引入另外 2 种实体：服务代理和资源代理。在拍卖过程中，资源没有起拍的参考价格，以供需关系为原则确定成交价格，销售者提交当前的请求价格给资源代理，购买者提交当前的竞拍价格给服务代理，如果请求价格和竞拍价格相匹配，则立即执行交易。

### 2.1 资源拍卖模型

图 1 展示了 ORMA 使用的拍卖模型，分为 4 个过程：服务请求分配过程、拍卖过程、资源分配过程和虚拟网络映射过程。其中，本文对前 3 个过

程进行分析讨论，而虚拟网络映射过程是后续的研究工作。在服务请求分配过程中，服务代理收集购买者的服务请求，依据请求的数量和价格归一化不同类型资源的价格，并依据此价格建立竞拍价格序列。在资源分配过程中，资源代理收集销售者提供的可用资源，依据提供的资源数量和请求价格归一化不同类型资源的价格，并依据此价格建立请求价格序列。在拍卖过程中，运行最佳匹配算法匹配销售者和购买者，匹配成功则计算出最终成交价格，并以此价格交易，同时销售者执行服务请求预留算法，以容纳更多的服务请求。经过以上过程，资源被合理地分配给适合的服务请求，最终最大化整个市场的收益。

为了降低整个拍卖过程的计算量，将销售者分组形成多个资源域，每个资源域设置一个资源代理。分组原则可以采用将连接在同一路由器下的计算机节点分成一组，这些节点拥有相似的计算能力和传输数据能力。因此，同一资源域的计算机节点的请求价格相同，分享由购买者支付的费用。此外，拍卖过程存在拍卖人角色，该角色可以用独立的服务器充当，其作用是运行最佳匹配算法以确定哪些销售者和购买者相匹配。这可能会导致图 1 的架构是中心化的，但该服务器仅与资源代理和服务代理进行消息通信，通信内容为请求价格和竞拍价格、剩余资源数和需求资源数，这些消息内容和数量都较少，且最佳匹配算法的运行时间很短，因此该服务器不会成为瓶颈节点。同时，利用分布式计算来处理资源分配问题已有成熟的研究方案<sup>[12]</sup>，可作为服务器非中心化的参考。

### 2.2 参数定义

为了说明 OMRA 策略的运行流程，定义如下

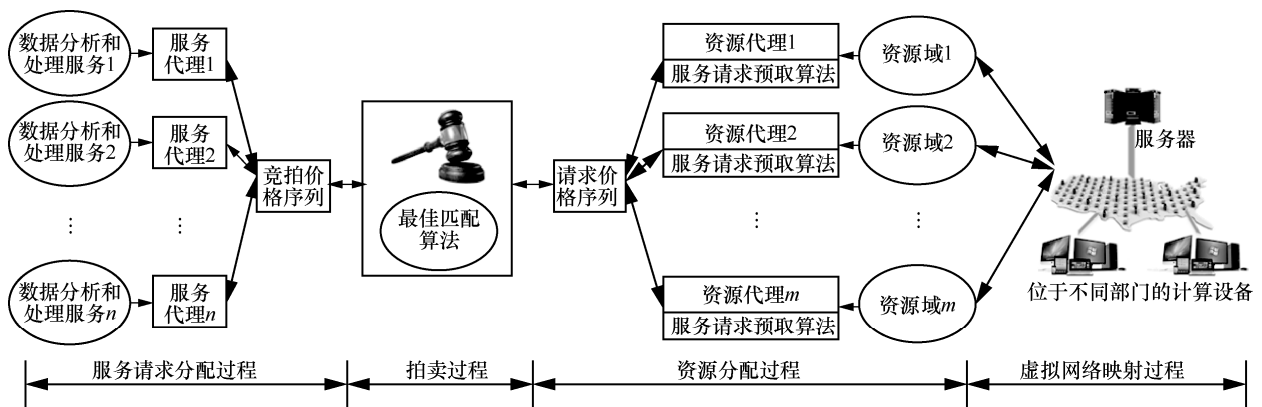


图 1 ORMA 拍卖模型

参数。

**定义 1** 资源节点。企业级网络中拥有计算和数据传输能力的可租用实体可与其他实体协作并提供服务请求所需软件的运行环境。

**定义 2** 基准运行时间。基准服务请求在基准资源节点从开始运行到结束所需要的时间，单位为 s。基准资源节点是指由企业级网络管理者设定的资源节点或者虚拟资源节点，且没有加速模块，如未安装高 I/O 数据传输能力的网卡等。基准服务请求是指仅申请基准资源节点即可满足运行条件的任务。

例如，服务请求 C 被分配到基准资源节点上运行，完成 C 需要时间为  $nt$ ，其中， $t$  为基准运行时间，单位为 s。C 运行在资源节点 B 上时，需要时间为  $mt$ 。在此基础上，如果已知某任务运行在基准资源节点上需要时间为  $T$ ，则此任务运行在 B 上需要时间为  $\frac{mT}{n}$ 。因此，可以计算出每次执行期内服务请求所需的基准资源节点数量，同时也能判断出资源节点的运行速率。

**定义 3** 平均服务请求率。一轮拍卖过程中，能够容纳的服务请求数与最大的可用资源数量的比值。平均服务请求率表示达到服务请求预期的 QoS 所需的基准资源数量，也能衡量不同资源域的资源数量。服务请求预期的 QoS 可以用费用预算来度量，费用预算是在服务请求在拍卖过程中可以给出的最大竞拍价格。

**定义 4** 资源归一化值。以基准资源节点的资源数值为标准，用于量化非基准资源节点的资源数值。假设服务请求在基准资源节点上运行需要资源为  $R_i$ ，在节点 B 上运行需要资源为  $R_j$ ，则基准资源节点与 B 之间的资源归一化计算为

$$\delta_{ji} = \frac{R_j}{R_i} \quad (1)$$

由于基准资源节点不含任何加速模块，通常  $\delta_{ji} \leq 1$ 。

根据式(1)可知，对于购买者，从基准资源节点购买服务请求需要资源为  $z$ ，从 B 购买服务需要资源为  $z\delta_{ji}$ 。假设基准资源节点和 B 给出相同的请求价格，应将购买者分配给 B，因其需要支付的费用少于分配给基准资源节点的费用。然而，对于销售者， $\delta_{ji}$  值越低，意味着运行成本越高，即相比于基准资源节点，销售者需要提供更多的 CPU 内核数，

更多的路由器端口才能使  $\delta_{ji}$  更低。因此，需要设计成本计算方法以保证销售者盈利。

### 2.3 高计算能力和高 I/O 能力节点资源数

随着 CPU 利用率的增加，CPU 的资源占用数也随之增加<sup>[13]</sup>。基于此理论，高计算能力节点的资源数计算方法为

$$R_{\text{CPU}} = R_i + (R_{\text{res}} - R_i) \text{CPU}(u) \quad (2)$$

其中， $R_{\text{res}}$  是资源节点当前提供的资源数，该值在每次拍卖过程中是变化的， $R_{\text{res}} \in [R_{\text{min}}, R_{\text{max}}]$ ， $R_{\text{min}}$  和  $R_{\text{max}}$  分别是该节点能提供的最小和最大资源数； $\text{CPU}(u)$  是当前 CPU 利用率，该值用于量化高计算能力节点所拥有的资源数值。

数据传输规模主要反映在对路由器的资源占用上<sup>[13]</sup>。一般来说，路由器包含 4 个部分：底盘、交换结构、线卡和端口。底盘和交换结构消耗的资源数在路由器开启的情况下是固定的，线卡和端口的资源占用数  $P_{\text{LC}}$  和  $P_{\text{port}}$  是随着进出路由器的数据量而动态变化的。高 I/O 通信能力节点消耗的资源数为

$$R_{\text{I/O}} = R_i + \Delta m P_{\text{LC}} + \Delta n P_{\text{port}} \quad (3)$$

其中， $\Delta m$  和  $\Delta n$  是高 I/O 通信能力节点相比于基准资源节点多出的线卡和端口数量； $P_{\text{LC}}$  和  $P_{\text{port}}$  的数值可以从文献[14]获取，该值用于量化高 I/O 通信能力节点所拥有的资源数值。

根据式(2)和式(3)可知，拍卖过程中，高能力节点有以下 2 种策略给出请求价格：1) 请求价格高于基准资源节点的请求价格，则在拍卖过程中获得较低的优先级，不能赢得拍卖；2) 请求价格低于或等于基准资源节点的请求价格，则能够赢得拍卖，但获得的收益要大于运行的成本，如式(4)所示。

$$R_{\text{CPU}} t_{\text{run}} \leq R_i \text{ 或 } R_{\text{I/O}} t_{\text{run}} \leq R_i \quad (4)$$

其中， $t_{\text{run}}$  是在高性能节点执行一次服务请求所需的时间。

图 2 反映在不同 CPU 利用率下，运行时间的变化趋势。横坐标表示资源节点当前提供的资源数与基准资源节点提供资源数的比值。当  $\frac{R_{\text{res}}}{R_i} < 1$  时，资源节点提供的资源数小于基准资源节点提供的资源数，此时服务请求在资源节点上的运行时间要大于在基准资源节点运行时间。当  $\frac{R_{\text{res}}}{R_i} > 1$ ，CPU 利用率从 20% 提升到 100% 时，服务请求运行时间会

降低，这意味着可以容纳更多的服务请求。在 CPU 利用率达到 80% 时，继续提升 CPU 利用率到 100% 不会显著降低服务请求时间，此时，新的服务请求到来时，会被该资源节点拒绝。

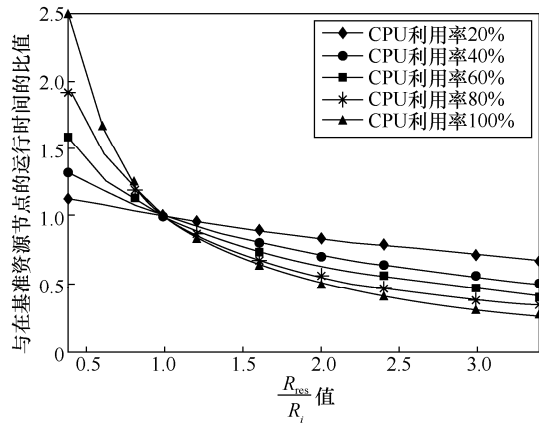


图 2 不同  $\frac{R_{obs}}{R_i}$  值下的服务请求运行时间比值

### 3 OMRA 策略拍卖机制

为了降低拍卖过程中的计算量，在构建 OMRA 策略模型时，需要先对请求价格和竞拍价格从高到低进行排序，选择 TOP-M 请求价格和 TOP-N 竞拍价格形成请求价格序列和竞拍价格序列。

在拍卖过程中，基于上述 2 个序列，定义  $Q_1$  为  $N'$  个购买者的集合， $Q_1$  中的每个购买者给出的竞拍价格都高于集合外的购买者的竞拍价格。定义  $Q_2$  为  $M'$  个销售者的集合， $Q_2$  中的每个销售者给出的请求价格都低于集合外的销售者的请求价格。定义  $Q_3$  为  $M'$  个销售者的集合， $Q_3$  中的每个销售者的  $\delta_{ji}$  值都低于集合外的销售者的  $\delta_{ji}$  值。构建企业级网络拍卖市场参与者为  $Q_1 \cup Q_2 \cup Q_3$ ，并遵循如下规则。

1) 设  $t$  代表拍卖过程的某个时刻， $t \in [t_0, t_1]$ ， $t_0$  和  $t_1$  分别是本轮拍卖开始时刻和结束时刻。拍卖初始，资源代理提交初始请求价格序列，服务代理提交竞拍价格序列。在  $t$  时刻，资源代理  $i$  更新请求价格  $r_i(t)$ ，服务代理  $j$  更新竞拍价格  $b_j(t)$ 。当可用资源数和服务请求数发生变化时，立即更新请求价格序列和竞拍价格序列。

2) 在拍卖过程中，拍卖人服务器持续关注 2 个序列，运行最佳匹配算法以保证整个拍卖市场获取最大收益。

3) 销售者利用服务请求预留算法确定由资源

代理发送来的服务请求能否满足需求的资源。如果满足，则通知资源代理与服务代理进行交易，否则，更新请求价格进入下一轮拍卖。

4) 拍卖过程的销售者和购买者均来自  $Q_1 \cup Q_2 \cup Q_3$ 。

5) 执行拍卖交易后，如果销售者还有可用资源，则通知资源代理更新请求价格，形成新的请求价格序列。

#### 3.1 最佳匹配拍卖算法

在企业级网络中，服务请求可以在不同资源域上运行，资源域也可以容纳不同的服务请求。因此，需要解决的关键问题是如何匹配服务请求和资源域（即销售者和购买者）来最大化整个拍卖市场的收益。为了解决上述问题，提出了最佳匹配拍卖算法，建立数学模型如下。

1) 依据请求价格序列和竞拍价格序列建立图  $G=(V, E)$ 。其中， $V = \{x_i \cup y_j | x_i \in X, y_j \in Y\}$ ， $X$  是请求价格序列， $Y$  是竞拍价格序列； $E = \{(x_i, y_j) | x_i \in X, y_j \in Y, b_{x_i}(t) \geq a_{y_j}(t)\}$ 。

2) 权重  $w(x_i, y_j)$ 。 $x_i$  和  $y_j$  相匹配，设置  $x_i$  当前的请求价格值为  $x_i$  的权重  $w(x_i)$ ， $y_j$  当前竞拍价格与  $\delta_{ji}$  乘积值为  $y_j$  的权重，则  $w(x_i, y_j) = w(x_i) - w(y_j)$ 。

上述的模型可利用已有的 KM(Kuhn-Munkres) 算法进行求解，但已知的 KM 算法及其改进算法的时间复杂度均为  $O(n^3)$ 。因此，本文设计了一种改进的 KM 算法，使算法复杂度降到  $O(n^2)$ 。

#### 算法 1 KM 改进算法

1) 输入  $G$ 。

2) 标定  $x_i, y_j$  的顶标值  $l(x_i)$  和  $l(y_j)$ ，使其满足条件  $l(x_i) + l(y_j) \geq w(x_i, y_j)$ ，其中， $x_i \in X, y_j \in Y$ 。给定顶标值为

$$l(x_i) = \max(w(x_i, y_j)), l(y_j) = 0$$

形成新的二分图  $G'$ 。

3) 计算初始匹配  $M$ 。在所有的  $y_j$  范围内，尽可能地找到  $x_i$  的可行匹配  $(x_i, y_j)$ 。 $(x_i, y_j)$  应满足以下条件： $x_i$  与  $y_j$  有  $\max(w(x_i, y_j))$ ，且  $x_i$  未与其他  $y_k$  建立匹配， $y_k \in Y - \{y_j\}$ ； $w_{ij} > w_{il} (j \neq l)$ ，且  $x_i$  与  $y_l$  已经建立了匹配。如果找到  $(x_i, y_j)$ ，则  $(x_i, y_j) \in M$ 。

4) 如果  $\forall x_i \in M$ ，KM 改进算法终止。如果

$\exists x_i \notin M$ ，则取节点  $u$ ，满足  $u \in G'$  且  $u \notin M$ 。初始化  $S = \{u\}$ ， $T = \emptyset$ 。

5) 如果  $T \subset N_{G'}(S)$ ，转至步骤 6)；如果  $T = N_{G'}(S)$ ， $N_{G'}(S)$  是  $S$  的每个元素的邻居节点组成的集合，计算  $\alpha_i = \min_{x_i \in S, y_j \in T} (l(x_i) + l(y_j) - w(x_i, y_j))$ ，

$$\tilde{l} = \begin{cases} l(v) - \alpha_i & , v \in S \\ l(v) + \alpha_i & , v \in T, \text{ 将 } l \leftarrow \tilde{l}, G' \leftarrow G'_i \\ l(v) & , \text{ 其他} \end{cases}$$

6) 选择  $\forall y_i \in N_{G'}(S) - T$ 。如果  $y_i \in M$  且  $(y_i, z) \in M$ ，则  $S \leftarrow S \cup \{z\}$  与  $T \leftarrow T \cup \{y_j\}$ ，转至步骤 5)；获取  $M$  的增广路径  $P(u, y_j)$ ，计算  $M \leftarrow M \oplus E(P)$ ，转至步骤 4)。

算法 1 的时间复杂度分析如下。在搜索初始匹配时，采用的贪婪算法的时间复杂度为  $O(n^2)$ 。假设可以找到  $m$  条匹配路径，那么，剩余的  $n - m$  条匹配路径需要用 KM 算法寻找，则算法时间复杂度为  $O((n - m)n^3)$ 。如果  $m = n$ ，则算法时间复杂度为  $O(n^2)$ 。如果  $m = 0$ ，算法最坏的时间复杂度为  $O(n^2 + n^3)$ 。因此，改进的 KM 算法的时间复杂度在  $O(n^2)$  和  $O(n^3)$  之间，且贪婪思想在匹配过程中不会失效，故改进的 KM 算法的时间复杂度要优于 KM 算法，可以达到  $O(n^2)$ 。

### 3.2 服务请求预留算法

最佳匹配算法执行完毕时，部分服务请求已和资源域 RA 相匹配，但是否能被 RA 接受需要进一步计算。另外，相比于当前时间，越临近拍卖结束时间，RA 给出的请求价格越低，导致成交价格降低，收益锐减。故在当前的请求价格下，如何获取更多的服务请求是本节研究的问题。该问题建模如下。

1) 设置服务请求  $j$  的资源需求矩阵 **request<sub>j</sub>**，**request<sub>j</sub>** = { $r_1[j], r_2[j], \dots, r_k[j], dl[j]$ }。其中， $k$  是  $j$  需求的资源类型， $r_k[j]$  是需求的数量， $dl[j]$  是服务请求的预期截止时间。

2) 建立资源节点  $i$  的可用资源数矩阵 **available<sub>i</sub>** 和最大资源数矩阵 **maximum<sub>i</sub>**，**available<sub>i</sub>** = { $a_1[i], a_2[i], \dots, a_k[i], pt[i]$ }，**maximum<sub>i</sub>** = { $m_1[i], m_2[i], \dots, m_k[i], pt[i]$ }。其中， $a_k[i]$  是本轮拍卖过程  $i$  提供的  $k$  的可用数量； $m_k[i]$  是  $i$  提供的  $k$  的最大数量； $pt[i]$  是  $i$  的计划关闭时间，即企业级网络中  $i$  将在  $pt[i]$  时刻或

之后关闭。

设计服务请求预留算法求解上述模型，如算法 2 所示。

#### 算法 2 服务请求预留算法

- 1) 输入 **request<sub>j</sub>**，**available<sub>i</sub>**，**maximum<sub>i</sub>**；
- 2) 输入请求价格 **asking<sub>[i]</sub>**，竞拍价格 **bidding<sub>[j]</sub>**；
- 3) 计算收益矩阵 **profit**，**profit** = { $p_1[i], p_2[i], \dots, p_j[i]$ }。  $p_j[i] = \frac{\text{asking}[i] - \text{bidding}[j]}{2}$ ；
- 4) 在 **profit** 中，选择拥有最大收益值的服务请求  $j$ 。if (每个  $j$  都被选出)，转至步骤 24)；
- 5) for (每个与  $i$  匹配的  $j$ ) { //资源回收
- 6)   if ( $j$  运行结束) {
- 7)     **available<sub>i</sub>** = **available<sub>i</sub>** + **request<sub>j</sub>**；
- 8)   }
- 9) }
- 10) if (**request<sub>j</sub>** < **available<sub>i</sub>**) {
- 11)   转至步骤 16)；
- 12) } else {
- 13)   暂时将  $p_j[i]$  从 **profit** 移除；
- 14)   转至步骤 4)；
- 15) }
- 16) **available<sub>i</sub>** = **available<sub>i</sub>** - **request<sub>j</sub>**；
- 17) 计算  $p_j[i]$  的运行截止时间，其等于开始时间与预期执行时间之和。
- 18) if (运行截止时间 <  $dl[j]$  && 运行截止时间 <  $dl[j]$ ) {
- 19)   **profit** = **profit** - { $p_j[i]$ }；
- 20)   将  $j$  放置到  $i$  的可接受服务请求序列中；
- 21) } else {
- 22)   拒绝  $j$ ；
- 23)   转至步骤 4)； }
- 24) 检查  $i$  的可接受服务请求序列；
- 25) if (有可执行序列) {
- 26)   正式可接受服务请求序列；
- 27) } else {
- 28)   将此次分配来的服务请求全部返回给相应服务代理；
- 29) }
- 30) if (**available<sub>i</sub>** 非空) {
- 31)    $i$  计算新的请求价格，进入下一轮拍卖；
- 32) }

### 3.3 请求价格更新算法

在 OMRA 策略中，只有当服务请求给出的竞拍价格高于其他服务请求时，才能获取预期的需求资源。相似地，资源域给出的请求价格低于其他资源域时，才能获取较多的服务请求。而且在拍卖开始时，资源域给出的请求价格高，临近拍卖结束时，给出的价格低。依据以上价格变化原则，设计请求价格更新算法，详细表述如下。

令  $st(t)$  代表拍卖开始时间， $\tau$  代表本轮拍卖持续的时间，则当前时间  $t$  满足

$$\tau \leq t \leq st(t) + \tau \quad (5)$$

剩余时间  $r(t)$  为

$$r(t) = st(t) + \tau - t \quad (6)$$

且满足  $r(t) \in [0, \tau]$ 。

在每轮拍卖过程中，资源域 RA 请求价格的最大值  $a_{\max}^i$  和最小值  $a_{\min}^i$  是随着时间变化的，设计 RA 的请求价格  $a(t)$  的变化式为

$$a(t) = a_{\min}^i + \kappa(x)(a_{\max}^i - a_{\min}^i) \quad (7)$$

其中， $\kappa(x)$  是相关于  $r(t)$  的函数。当  $r(t)$  减少时， $\kappa(x)$  值降低。当  $r(t) \rightarrow \tau$  时， $a(t) \rightarrow a_{\max}^i$ ，此时  $\kappa(\tau) = 1$ 。当  $r(t) \rightarrow 0$  时， $a(t) \rightarrow a_{\min}^i$ ，此时  $\kappa(0) = 0$ 。则定义  $\kappa(x)$  [15] 为

$$\kappa(x) = \kappa' + (1 - \kappa') \left( \frac{x}{x_{\max}} \right)^{\frac{1}{\beta}} \quad (8)$$

其中， $\beta \in R^+$  代表请求价格函数的曲率， $\kappa'$  是实常数，取值范围为  $\kappa' \in [0, 1]$ 。特别地，当  $\beta \rightarrow 0$  和  $\beta \rightarrow +\infty$  时， $\kappa(x)$  分别满足式(9)和式(10)。

$$\kappa(x) = \lim_{\beta \rightarrow 0} \left[ \kappa' + (1 - \kappa') \left( \frac{x}{x_{\max}} \right)^{\frac{1}{\beta}} \right] = \kappa' \quad (9)$$

$$\kappa(x) = \lim_{\beta \rightarrow +\infty} \left[ \kappa' + (1 - \kappa') \left( \frac{x}{x_{\max}} \right)^{\frac{1}{\beta}} \right] = 1 \quad (10)$$

联合式(7)， $x$  是剩余时间，且  $x=r(t)$ ，因此 RA 的请求价格更新函数如式(11)所示。

$$a(t) = a_{\min}^i + \left[ \kappa' + (1 - \kappa') \left( \frac{r(t)}{\tau} \right)^{\frac{1}{\beta}} \right] (a_{\max}^i - a_{\min}^i) \quad (11)$$

其中， $\beta$  是请求价格参数。

为了评估请求价格更新函数，设置  $\tau = 10 \text{ min}$ ，

$a_{\min}^i$  和  $a_{\max}^i$  分别设置为 100 倍和 200 倍的基准资源节点请求价格。图 3 和图 4 中，横坐标分别代表剩余时间和拍卖时间，纵坐标反映请求价格变化趋势。从图 3 中可以看出，当  $\beta$  值较小时 ( $\beta < 0.7$ )，请求价格变化曲线是凹曲线且数值快速降低，此时销售者有较高的优先级，但收益不会太高。从图 4 可以看出， $\kappa'$  决定了请求价格降低程度。当  $\kappa'$  值较大时 ( $\kappa' = 0.90$ )，请求价格随着拍卖进程变化不大。从整体来看，建议  $\beta$  的取值最好大于 0.4， $\kappa'$  取值小于 0.1。

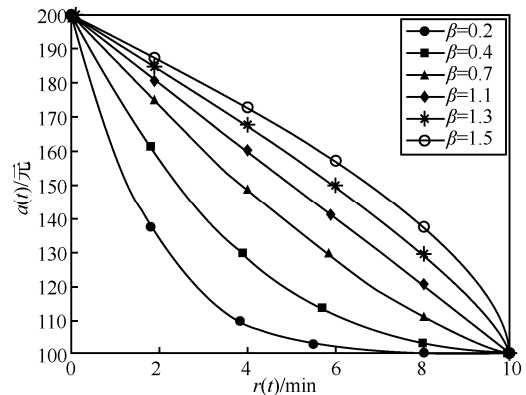


图 3 不同  $\beta$  值下随  $r(t)$  增加的 A 的  $a(t)$  变化趋势

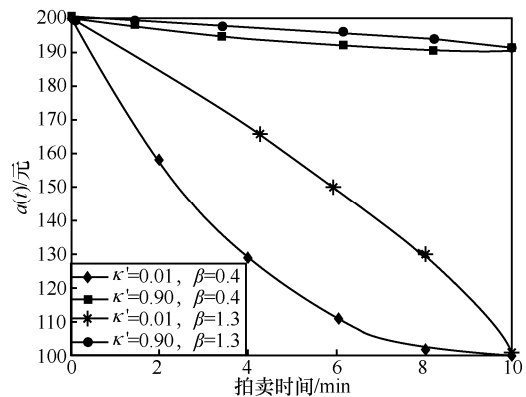


图 4 不同  $\kappa'$  和  $\beta$  值下随拍卖时间增加的请求价格变化趋势

### 3.4 支付价格和收益

在 OMRA 策略中，由资源代理和服务代理分别管理销售者和购买者的需求信息，由高性能资源节点成本核算方法计算销售者请求价格变化范围，由最佳匹配算法保证拍卖市场的最大收益，由服务请求预留算法保证在满足购买者请求条件下尽可能多地以当前的成交价格容纳更多的服务请求。上述过程运行完毕之后，交易立即执行，最终成交价格为

$$p_{ij} = \frac{\text{asking}[i] + \text{bidding}[j] \delta_{ij}}{2} \quad (12)$$

对于销售者，采用连续双边拍卖策略成交价格

为  $\frac{\text{asking}[i] + \text{bidding}[j]}{2}$ ，采用 OMRA 策略能够为销售者额外增加的费用为  $\left| \frac{(\delta_{ji} - 1)\text{bidding}[j]}{2} \right|$ 。

由第 2 节和第 3 节可知，OMRA 策略对企业级网络的特征具有适应性，具体体现为：1)针对企业网络节点自私特性，设立了拍卖激励机制方案，以加速网络带宽速率和软件免费使用权为激励条件，此非严格激励机制能促使节点贡献资源，保证网络资源的可用性；2)针对企业网络节点运行任务具有时限性的特征，在设计请求价格策略时，加入了随时间变化的价格变化函数；3)针对企业级网络节点提供资源差异性特征，在运行最佳匹配算法时，确立了提供较好可用资源和任务运行时间的节点以较高优先级被选中的策略，同时关注了该类节点随着任务运行的资源数变化情况；4)最佳匹配算法的依据是保证销售者和购买者能够以双方均满意的价格达成交易，而不是偏袒某一方，因此，可以实现销售者获取比预期多的收益，购买者支付更少的费用的目标；5)为了增加本轮拍卖过程中的销售者收益，设计了服务预留算法，能够在满足任务运行时间需求的条件下，预先容纳拟运行的任务。需要注意的是，4)和 5)不是冲突的，4)是保证实现预期的交易价格，5)是实现容纳任务数的增加。

## 4 实验及结果分析

### 4.1 实验参数

为了评估 OMRA 策略的性能指标，使用 MyEclipse 平台利用 PeerSim 开发引擎形成了可加载的程序包，搭建了位于不同地理位置的由计算机和服务器组成的网络来模拟企业级网络环境。一般来说，企业级网络的搭建原则是服务器一般连接在核心交换机或者一级交换机上，其他计算机等节点连接在多级交换机上，因此，服务器具有高计算性

能和高 I/O 通信性能，全天 24 h 运行。针对企业级网络一般情况，设计了节点  $A_1 \sim A_5$  通过有线连接在同一个交换机下，相互之间的数据传输速率较快，有较低的计算能力。节点  $A_6 \sim A_8$  通过无线连接到企业网络中，由于地理位置的不同处于不同交换机下，有中等计算能力，相互之间的数据传输速率较慢。服务器  $A_9 \sim A_{10}$  通过有线连接到核心交换机上，有高计算能力和高 I/O 数据传输能力。为了反映不同需求任务的运行情况，设置服务器  $A_9$  更倾向于高计算能力， $A_{10}$  更倾向于高 I/O 通信能力。物理节点参数配置详见表 1。

设置基准资源节点的计算和带宽能力分别为 1.0 GHz 和 10 Mbit/s。节点  $A_1 \sim A_5$  提供计算资源最小值和最大值设置为  $R_{\min}=0 \times 1.0$  GHz,  $R_{\max}=2 \times 1.0$  GHz。节点  $A_6 \sim A_8$  计算资源最小值和最大值设置为  $R_{\min}=1 \times 1.0$  GHz,  $R_{\max}=2.5 \times 1.0$  GHz。 $A_1 \sim A_8$  的带宽能力计算参数为  $\Delta m = 1$ ,  $\Delta n \in [1, 10]$ 。 $A_9$  的带宽能力计算参数为  $\Delta m \in [1, 4]$ ,  $\Delta n \in [1, 100]$ 。 $A_{10}$  具有多个 CPU 核心，可提供计算资源最小值和最大值设置为  $R_{\min}=1 \times 1.0$  GHz,  $R_{\max}=12 \times 1.0$  GHz。本轮拍卖结束后的下一轮的请求价格的参数设置为  $\beta = 0.4$ ,  $\kappa' = 0.1$ 。对于购买者来讲，无论可用资源充足与否，服务请求都应该被满足。令  $a_{\min}^i = \text{成本}$ ,  $a_{\max}^i = \text{成本} \times r$ ,  $r \in [1, 4]$ ，设置可接受率来衡量服务请求的接受程度，可接受率为被接受的服务请求数与所有的服务请求数的比值。

### 4.2 结果分析

图 5 展示了服务请求需求对服务请求可接受率的影响。请求的资源数量越大，获取该资源越困难，原因在于单个的资源节点不能提供充足的资源。竞拍价格越低，在拍卖过程中和资源域匹配的概率越小，原因在于相比于接受其他高竞拍价格的服务请求，销售者得到的收益更低。预期运行时间影响占用资源的释放速度，进而影响可用资源数量。服务

表 1 物理节点实验参数设置

节点	类型	运行速度/GHz	成本	性能	资源域数	节点	类型	运行速度/GHz	成本	性能	资源域数
$A_1$	台式机	i3 3.4	22	No	5~10	$A_6$	笔记本电脑	i7 2.9	12	No	1~10
$A_2$	台式机	i3 3.4	22	No	5~10	$A_7$	笔记本电脑	i7 2.9	12	No	1~10
$A_3$	台式机	i3 3.9	23	No	5~10	$A_8$	笔记本电脑	i7 2.9	12	No	1~10
$A_4$	台式机	i3 3.9	23	No	5~10	$A_9$	服务器	E3 3.3	45	高 I/O 性能	100
$A_5$	台式机	i5 2.8	25	No	5~10	$A_{10}$	服务器	E3 3.3	40	高 CPU 性能	100

请求预期运行时间越长，则释放资源越慢，市场上可用资源数越低。基于以上原因，在相同的服务请求到来的条件下，当可用资源数充足 ( $r=4$ ) 时，服务请求的可接受率较高，当可用资源数匮乏 ( $r=1$ ) 时，可接受率会发生突变甚至停止。

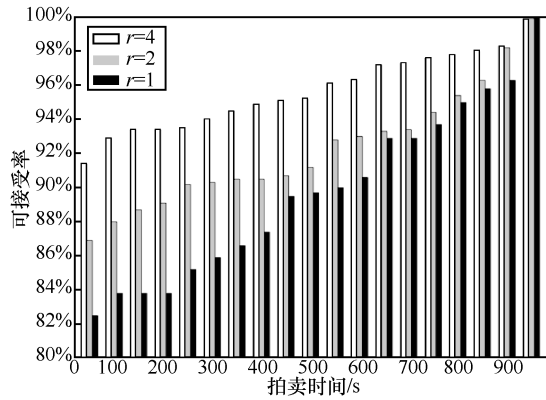


图 5 随拍卖时间增加的服务请求可接受率变化

图 6 显示了整个拍卖市场销售者收益率 (标识收益增加) 和购买者支付率 (标识费用节省) 的评估结果。定义收益率  $brate$  和支付率  $prate$  为

$$brate = \frac{\sum_{i=1}^m \sum_{j=1}^n (p_{ij} D_j)}{\sum_{i=1}^m \sum_{j=1}^n (asking[i] D_j)} \quad (13)$$

$$prate = \frac{\sum_{j=1}^n \sum_{i=1}^m (p_{ij} D_j)}{\sum_{j=1}^n \sum_{i=1}^m (bidding[j] D_i)} \quad (14)$$

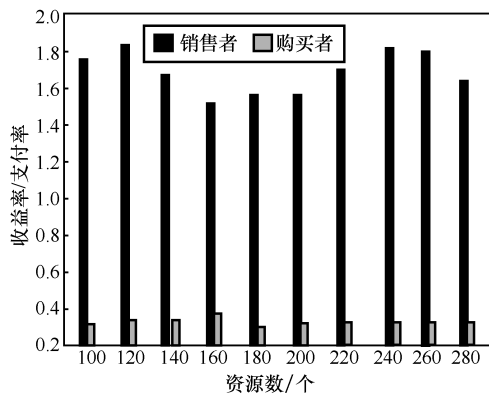


图 6 随资源数增加的收益/支付率变化

收益率  $brate$  和支付率  $prate$  评估了销售者和购买者的满意程度。

从图 5 和图 6 可以看出，1) 可接受率反映了匹配性能变化趋势，保证了资源分配的效率；2)  $r$  值

的增加不会影响匹配性能；3) 由于运行服务预留算法能够使销售者以当前的价格获取更多的服务请求，使其收益率保持在一个很窄的范围；4) 购买者支付率与收益率有相似的情形。

图 7 展示了在资源分配率参数上与 CRAA/FA (cloud resource allocating algorithm via fitness-enabled auction) 策略 [16] 对比情况。假设资源域接受服务请求  $i$ ，相对基准资源节点的归一化参数为  $\delta_i$ ， $i$  请求的资源数为  $D_i$ 。定义资源分配率  $ralloc$  为

$$ralloc = \frac{\sum_{i=1}^m \delta_i D_i}{\sum_{i=1}^m D_i} \quad (15)$$

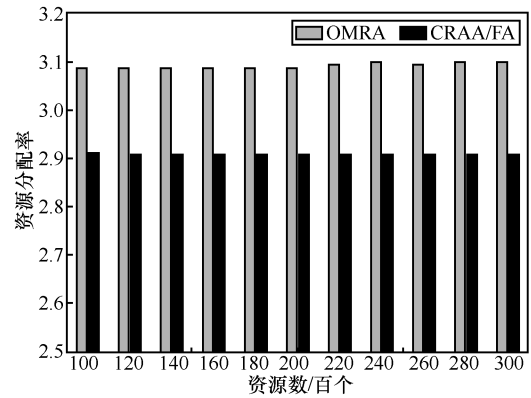


图 7 OMRA 与 CRAA/FA 机制在资源分配率的比较

资源分配率反映了当资源数增加或者降低时的资源分配情况，同时，在资源数一定的情况下，影响当前购买者的竞拍价格参数设置，即评估资源域和服务请求的匹配程度。从图 7 可以看出，当  $\delta_i$  确定之后，调动更多的资源到不同的资源域上对资源分配率不会造成太多的影响。

图 8 展示了 OMRA 策略与 CRAA/FA 策略在收益率的比较情况。从图 8 中可以看出，随着  $r$  的增大并进入到非合理的区间，收益率会骤降，直至降至 0。这种情况是合理的且能引导资源域诚信地给出请求价格，同时能够避免销售者之间的联盟。服务请求预留算法促使资源域能够提前接收到服务请求，因此 OMRA 策略的收敛速度要快于 CRAA/FA 策略。

从图 7 和图 8 中可以看出：1) 当资源数量或者请求价格在合理区间时，资源分配率曲线是稳定的；2) 当请求价格增长时，整个拍卖市场范围内的收益率不是一直增加的，尤其是当  $r$  处于非合理

的范围时，收益率会快速下降。

$RA_n = \bigcup_{i=1}^n A_i$  是  $n$  个资源节点的集合， $D_j$  是被  $A_i$  接受的服务请求  $j$  所需求的资源数。设立市场共享率  $msrate$  来表明价格和分配资源之间的关系，如式(16)所示。

$$msrate = \frac{\sum_{D_j \in A_i} D_j}{\sum_{D_j \in RA_n} D_j} \quad (16)$$

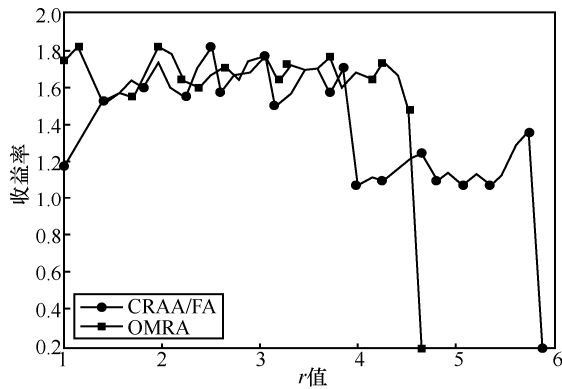


图 8 OMRA 与 CRAA/FA 机制在收益率的比较

图 9 中，资源域有 3 种类型。第一种以  $\bigcup_{i=1}^5 A_i$  为代表，其特点是运行速率较低，只在销售者工作时间提供资源。第二种以  $\bigcup_{i=6}^8 A_i$  为代表，其特点是较快的运行速率，只在销售者工作时间提供资源。第三种以  $\bigcup_{i=9}^{10} A_i$  为代表，其特点是最快的运行速率，24 h 不间断工作，具有高性能 CPU 和 I/O 通信能力。从图中可以看出，随着资源数的增加， $A_1$  的市场共享率在稳步增加，直至达到顶点。相比于 CRAA/FA 策略，OMRA 策略的市场共享率偏低。原因在于资源是由企业级网络设备而不是云平台内的节点提供，能力偏低，可用资源数偏少，同时  $A_2 \sim A_5$  分散了服务请求数，导致  $A_1$  赢得的服务请求数偏少。得益于服务预留和最佳匹配算法，OMRA 策略收敛速度要快于 CRAA/FA 策略。其他曲线与  $A_1$  类似。

从图 9 可以看出：1) 没有特殊需求的服务（非实时应答、无高性能要求、预期截止时间很长）希望租用仅仅满足其需求的资源，以节省费用支出；2) 当拍卖开始时， $A_1$  给出的初始请求价格很低，则最佳匹配算法和成本计算方法能保证很多服务都分配到  $A_1$

上，但当  $A_1$  的市场共享率到达了峰值，再增加额外的资源数或者降低  $r$  值都不会增加其收益。

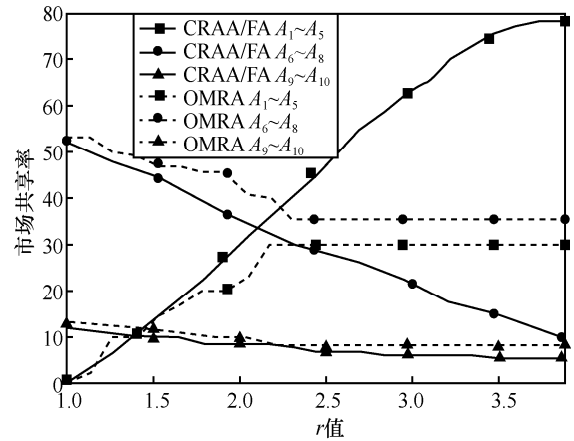


图 9 OMRA 与 CRAA/FA 机制在市场共享率的比较

### 4.3 应用及讨论

物理网络资源优化分配是云平台中的典型研究问题，已有的研究是基于资源提供节点地理位置相对集中，且资源提供节点能力相对一致的场景下。与之相对，OMRA 策略可应用于网络中提供资源的计算机节点位于不同的地理位置，企业数据中心服务器相对集中，提供的资源数目和持续在网时长多样化的企业级网络场景下。OMRA 策略也可应用于需要抑制节点自私性的网络资源分配场景，提升物理资源提供的数量和质量。基于资源分配框架的思想，OMRA 策略可用“包”的形式与其他研究者的算法同时集成，部署到企业级网络的服务器和计算机节点上。

### 5 结束语

本文提出了一种新的基于最优拍卖理论的资源分配机制，能够适应企业级网络资源节点的特征，将服务请求分配到不同的资源域处，达到销售者获取比预期多的收益，购买者支付更少的费用的目标。该机制包含成本计算方法，最佳匹配拍卖算法、服务预留算法、请求价格更新算法最终保证了整个拍卖市场的效率。实验结果表明，在企业级网络规模不大的情况下，OMRA 能有效地提高网络资源的分配效率，同时提升拍卖市场的收益率。

现有的实验模拟是在小范围内运行的，限于已有的资源限制和技术手段，未开展在大规模企业级网络环境下的适应性研究。由于资源域分散在不同的地理位置，如何穿过路由器发现更多资

源提供节点是进一步需要研究的问题。另外，资源域之间存在竞争关系，但垄断也可能存在于拍卖过程中，原因在于相似的资源节点可能位于同一个办公室，会形成联盟来垄断某些类型的资源域，例如高性能 I/O 资源域，研究的资源分配算法能否用于有此种情况下的资源市场是需要进一步研究的问题。

### 参考文献:

- [1] KENT A D, LIEBROCK L M, NEIL J C. Authentication graphs: analyzing user behavior within an enterprise network[J]. *Computers & Security*, 2015, 48(7):150-166.
- [2] DOMEIJ R, LUONG R. Estimation of resource allocation based on disturbance prediction data with use of statistics, machine learning and data analysis[D]. Lund: Lund University, 2017: E19.
- [3] SINGH S, CHANA I, BUYYA R. STAR: SLA-aware autonomic management of cloud resources[J]. *IEEE Transactions on Cloud Computing*, 2017, PP(99):1.
- [4] YAO Y, CAO J, JIANG Y S. A pattern prediction-based elastic workflow service in the cloud[C]// *IEEE International Conference on Web Services*. IEEE, 2017:784-791.
- [5] SHIRVASTAVA S, DUBEY R, SHRIVASTAVA M. Best fit based VM allocation for cloud resource allocation[J]. *International Journal of Computer Applications*, 2017,158(9):25-27.
- [6] PRASAD G V, PRASAD A S, RAO S. A combinatorial auction mechanism for multiple resource procurement in cloud computing[J]. *IEEE Transactions on Cloud Computing*, 2018, 6(4):904-914.
- [7] MARUFU A M C, KAYEM A V D M, WOLTHUSEN S D. Circumventing cheating on power auctioning in resource constrained micro-grids [C]// *2016 IEEE 18th International Conference on High Performance Computing and Communications*. IEEE, 2016: 137-149.
- [8] FAN Q, ZEITOUNI K, XIONG N, et al. Nash equilibrium-based semantic cache in mobile sensor grid database systems[J]. *IEEE Transactions on Systems Man & Cybernetics Systems*, 2017, 47(9): 2550-2561.
- [9] SUN D, CHANG G, WANG C, et al. Efficient Nash equilibrium based cloud resource allocation by using a continuous double auction[C]// *International Conference on Computer Design & Applications*. IEEE, 2010: 94-99.
- [10] FARAJIAN N, ZAMANIFAR K. Intelligent continuous double auction method for service allocation in cloud computing[J]. *Brain Broad Research in Artificial Intelligence & Neuroscience*, 2017(10):74-83.
- [11] CONG X, SHUANG K, SU S, et al. LBAS: an effective pricing mechanism towards video migration in cloud-assisted VoD system[J]. *Computer Networks*, 2014, 64(8):15-25.
- [12] ZHANG H, JIANG D, LI F, et al. Cluster-based resource allocation for spectrum-sharing femtocell networks[J]. *IEEE Access*, 2017,4(99): 8643-8656.
- [13] 肖丁,贾亚璞,石川.云计算环境下的非线性能耗建模方法[J]. *北京邮电大学学报*, 2016,39(1):107-111.  
XIAO D, JIA Y P, SHI C. Non-linear energy consumption model for cloud computing[J]. *Journal of Beijing University of Posts and Telecommunications*, 2016, 39(1):107-111.
- [14] GUENTER B, JAIN N, WILLIAMS C. Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning[C]// *International Conference on Computer Communications (INFOCOM)*. IEEE, 2011: 1332-1340.
- [15] ANTHONY P, JENNINGS N R. Developing a bidding agent for multiple heterogeneous auctions[J]. *ACM Transactions on Internet Technology*, 2003,2(3):185-217.
- [16] WANG H B, KANG Z, WANG L. Performance-aware cloud resource allocation via fitness-enabled auction[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2016,27(4):1160-1173.

### [作者简介]



丛鑫 (1982- )，男，辽宁阜新人，博士，辽宁工程技术大学高级工程师、硕士生导师，主要研究方向为 P2P 技术、云计算、流媒体和虚拟网络映射。



訾玲玲 (1981- )，女，辽宁阜新人，博士，辽宁工程技术大学副教授、硕士生导师，主要研究方向为图形图像与多媒体。



沈学利 (1969- )，男，江苏连云港人，辽宁工程技术大学教授、硕士生导师，主要研究方向为网络及信息安全。