

面向服务生态系统的 QoS 优化模型与求解研究

(Models and Solutions of QoS Optimization for Services Ecosystem)

报告人：李松远 (Li, Songyuan)

日期：2021 年 7 月 26 日 (July 26, 2021)



主要内容

- ❖ 引言
- ❖ 相关研究综述
- ❖ 基于公平性指标的多用户服务选择方法
- ❖ 面向市场的云资源定价与售卖机制
- ❖ QoE 感知的分布式边缘任务调度和资源管理
- ❖ 总结与展望

研究背景

❖ 服务计算 (Services Computing)

- 由软硬件共同组成的传统架构模式 → **面向服务的新兴设计模式**
- 软件/硬件资源以**预定义的接口**形式发布
- **简化**了用户对软件/硬件资源的**直接使用**过程
- **便于**上层应用对软件/硬件资源的**二次开发**
- 预约制服务订购模式 → **灵活的服务订购机制（按需服务模式）**
- 应用领域广泛：商业/金融/医疗服务等跨界信息服务领域

研究背景

❖ 服务生态 (Services Ecosystem)

- “服务生态系统” 形成背景：
 - ✓ 互联网、大数据、云计算、物联网、边缘计算等先进智能信息技术
- 服务计算领域的**研究内涵**不断发展和扩展
- 服务计算系统的**组成结构**愈发复杂
- 形成**多用户/服务供应商共同参与**的服务生态系统
- **服务生态系统会产生新的科学问题和挑战，亟需进一步的深入调研和探究！**
 - ✓ 维护多用户间服务资源分配的公平性
 - ✓ 最大化服务供应商的收益
 - ✓ 提升多用户的整体服务满意度 … …

研究背景

❖ 服务质量优化技术 (QoS Optimization)

- 众多候选服务功能相似，但是在**非功能性**方面表现参差不齐
- 服务生态系统规模庞大，对服务质量优化有更加**精细的优化要求**
- 产生新的**研究难点和挑战**：

多用户在服务生态中处于竞争和博弈关系

用户期待获取具备最优QoS的服务

VS

具备最优QoS的服务仅能供给其中一位用户

研究背景

❖ 服务质量优化技术 (QoS Optimization)

- 众多候选服务功能相似，但是在**非功能性**方面表现参差不齐
- 服务生态系统规模庞大，对服务质量优化有更加**精细的优化要求**
- 产生新的**研究难点和挑战**：

多服务供应商之间存在着相互竞争的态势

每位供应商期待将服务提供给用户以赚取尽可能高的服务收益



每位服务供应商不断提升自身的服务质量水平，以吸引用户



每位用户最终能够享受更加优质的服务质量水平

研究背景

❖ 服务质量优化技术 (QoS Optimization)

- 众多候选服务功能相似，但是在**非功能性**方面表现参差不齐
- 服务生态系统规模庞大，对服务质量优化有更加**精细的优化要求**
- 产生新的**研究难点和挑战**：

针对大规模物联网服务，边缘化的服务计算架构日益流行

服务生态系统中各用户广泛分布于不同的地理位置



亟需提出去中心化的新方式，优化多用户的边缘资源分配方案

研究内容

1

实现最大最小公平的多用户并发服务选择

2

最大化用户激励的市场化云资源定价机制

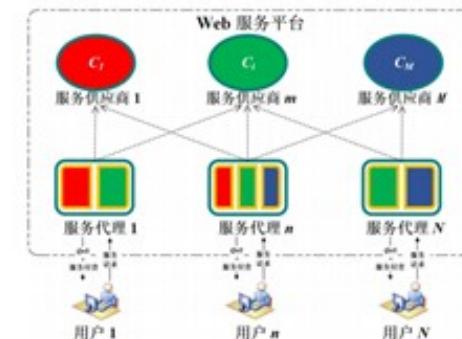
3

去中心化QoE感知的多用户边缘资源分配

研究内容及意义 (1)

❖ 实现最大最小公平的多用户并发服务选择

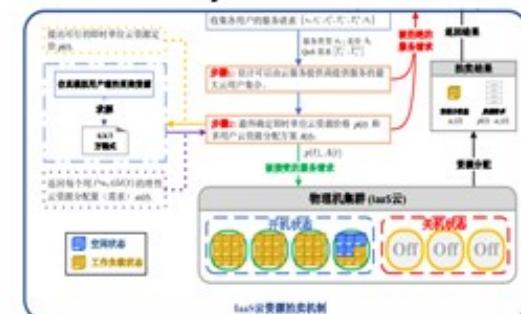
- 从公平性角度出发，研究**带约束的多用户并发服务选择**问题
 - ✓ **服务约束：** Web 服务所部署地理位置的约束、服务供应商和用户之间所达成的合约等
- **优化目标：**
 - ✓ 在多个用户请求的服务选择结果之间，满足**最大最小公平性**
 - ✓ 每位用户的 QoS 需求被得到保障，并被尽可能地提供更高的QoS 水平，使**服务供应商赚取更多的服务收益**
- 兼顾用户和服务提供商的利益需求
- 兼顾服务公平与服务质量



研究内容及意义 (2)

❖ 最大化用户激励的市场化云资源定价机制

- 构建一个面向市场的云服务生态系统
 - ✓ 云用户的资源需求量是价格敏感的，按需购买云资源
 - ✓ 云服务供应商根据动态波动的市场环境而调整即时云资源价格
 - 避免过高的云资源价格定价，导致云用户流失
 - 防止云资源定价过低，难以覆盖云资源的运营成本(能耗成本)
- **优化目标：**在保证云服务供应商最低利润率 γ 的前提下，最大化用户激励 (在云端服务的用户数量)
- **设计目标：**云资源拍卖机制 PIRA 满足预算可行性、激励相容性、无妒性



研究内容及意义 (3)

❖ 去中心化QoE感知的多用户边缘资源分配

- “边缘计算架构” 特点：针对**大规模分布式物联网服务**而设计
 - ✓ 低网络通信拥塞
 - ✓ 低服务响应时间
- **用户体验质量 (Quality of Experience, 缩写 QoE) 指标：**
 - ✓ 传统 QoS 指标(即服务响应时间)的扩展评价标准
 - ✓ 被用于量化描述物联网用户的服务满意程度
- **优化目标：**最大化系统 QoE 水平
- **设计目标：**高效、去中心化的多用户边缘资源分配解决方案



研究难点与挑战

研究内容

1

实现最大最小公平的
多用户并发服务选择

2

最大化用户激励的市
场化云资源定价机制

3

去中心化QoE感知的
多用户边缘资源分配

研究难点与挑战

兼顾服务公平与服务质量的
多目标优化设计

服务资源定价与用户资源需
求分配间的强耦合性

用户竞争博弈过程的无序性

研究难点与挑战 (1)

❖ 兼顾服务公平与服务质量的多目标优化设计

- 服务公平问题：

- ✓ 在多用户环境中，**每位用户的**服务请求被服务供应商**公平地**看待
 - ✓ **每位用户的** QoS 需求应该被服务供应商**同时兼顾**

- 服务质量问题：

- ✓ **每位用户**都希望被**最大限度地**满足其 QoS 需求，
 - ✓ 服务供应商**最大限度地**向**每位用户**提供以高 QoS 水平的服务

- 从本质上讲，**兼顾服务公平与服务质量**属于
多目标优化问题，存在诸多难题与挑战！

研究难点与挑战 (2)

❖ 服务资源定价与用户资源需求分配间的强耦合性

- 从市场化角度出发，用户对于服务资源的需求量是价格敏感的：

- ✓ 服务资源定价较高 → 服务生态的整体用户资源需求水平下降
- ✓ 服务资源定价较低 → 服务生态的整体用户资源需求水平上升



在服务生态市场里，服务资源定价方案与用户资源需求量之间具有强耦合性

- 借助动态的市场化服务资源定价策略，实现特定的市场优化目标

研究
难点
与挑
战

- ✓ 剖析服务生态市场的动态性特征
- ✓ 分析服务资源定价与用户资源需求分配之间的耦合关系
- ✓ 提出价格激励的资源拍卖机制

研究难点与挑战 (3)

❖ 多用户竞争博弈过程的无序性

- 构建**有序**的**多用户竞争/博弈**过程，促进**多用户**之间的**良性竞争**
 - ✓ 服务资源总量通常**有限**，多用户之间处于**相互竞争**的态势
 - ✓ **良性有序**的竞争/博弈关系，为服务生态系统的**改进/演化**提供基础
- 面向**大规模服务生态系统**，提供**高效可扩展**的服务资源分配方案
 - ✓ 大规模服务生态系统，e.g. 承载**大规模物联网用户**的边缘计算系统
 - ✓ 传统的**集中决策模式**难适用于**大规模服务生态系统**
 - ✓ 针对**大规模服务生态系统**，服务资源的分配决策过程：

去中心化，高效，动态有序，可扩展

主要内容

- ❖ 引言
- ❖ 相关研究综述
- ❖ 基于公平性指标的多用户服务选择方法
- ❖ 面向市场的云资源定价与售卖机制
- ❖ QoE 感知的分布式边缘任务调度和资源管理
- ❖ 总结与展望

相关研究综述

❖ 服务选择与服务推荐

- 动态系统感知：服务失效、网路失效等
- 服务冷启动

❖ 多用户间竞争与合作

- 博弈论/拍卖理论观点
- 空闲服务资源转售问题，e.g., Amazon EC2 Spot 实例

❖ 服务资源的定价策略

- **传统策略：**基于资源订阅的方法、按需付费模式
- **新兴策略：**面向市场定价策略，e.g., Amazon EC2 Spot 实例

❖ 边缘化的服务计算架构

- 边缘数据缓存优化
- 边缘服务请求调度

研究综述小结

❖ 研究现状

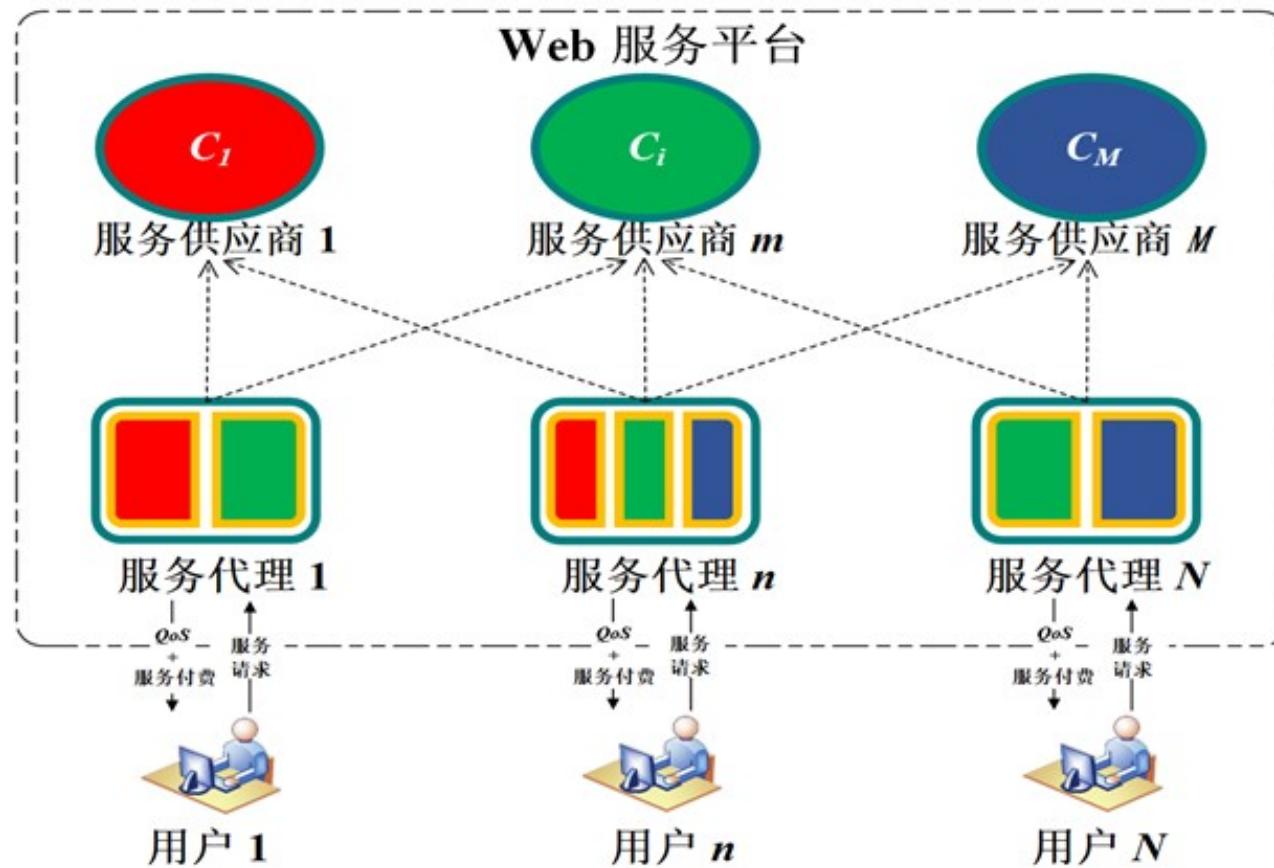
- QoS优化目标通常为响应时间、能源效率、服务成本等传统优化目标
- 鲜有将QoS与服务付费、服务定价相结合设计的服务质量优化方法
- 缺少有关多用户间的竞争博弈关系，促成多用户合作的条件和激励方法的系统性研究工作

鉴于此，此项研究工作将从理论角度出发来探索服务生态的QoS优化模型和求解，对服务生态的优化/演进具有重要的指导意义

主要内容

- ❖ 引言
- ❖ 相关研究综述
- ❖ **基于公平性指标的多用户服务选择方法**
- ❖ 面向市场的云资源定价与售卖机制
- ❖ QoE 感知的分布式边缘任务调度和资源管理
- ❖ 总结与展望

系统描述



多用户并发服务选择模型

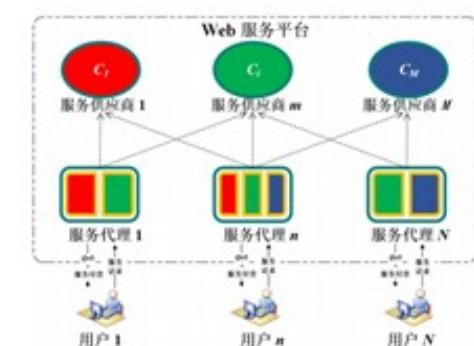
系统描述

❖ 用户服务请求 n 的执行时间 τ_n :

$$\tau_n = \sum_{i \in S_n} \sum_{j \in C_i} Q_{i,j} \cdot x_{i,j}^n$$

❖ 用户服务请求 n 的服务费用 π_n :

$$\pi_n = a_n + b_n \cdot \left(1 - \sum_{i \in S_n} \sum_{j \in C_i} \frac{Q_{i,j}}{Q_n^{(ref)}} \cdot x_{i,j}^n \right)$$



实现最大最小公平的字典序优化问题

定义 1 (最大最小公平性, 缩写 MMF): 当剔除了服务支付额高于第 n 最低服务支付额的这些用户请求后, 也无法进一步提高对应第 n 最低支付额用户的服务付费金额, $n=1,...,N$ 。这样的多用户服务选择方案符合最大最小公平性。 □

实现MMF的字典序优化问题描述:

$$\underset{x_{i,j}^n \in \Theta}{\text{lex max}} \boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$$

$$s.t. \quad \sum_{i \in S_n} \sum_{j \in C_i} x_{i,j}^n = 1, \quad \forall n \in \mathcal{N}$$

$$\sum_{n \in \mathcal{N}_i} x_{i,j}^n \leq 1, \quad \forall i \in \mathcal{M}, \forall j \in C_i$$

$$x_{i,j}^n \in \{0,1\}, \quad \forall n \in \mathcal{N}, \forall i \in \mathcal{M}, \forall j \in C_i$$



用户约束: 确保每个用户请求仅可从其可用候选服务中选择一个服务



服务供应商约束: 规定不同用户选择不同的目标服务



决策变量 $x_{i,j}^n$ 的定义域, 即 $\{0, 1\}$

多用户公平服务选择方案的求解算法 (FASS)

Algorithm 1: 实现最大最小公平的服务选择框架(FASS)

Input: 每个用户请求的服务付费参数 \mathcal{A} 和 \mathcal{B} , QoS需求信息 $\mathcal{Q}^{(ref)}$.

Output: 服务选择方案 $x_{i,j}^n, \forall n \in \mathcal{N}, i \in S_n, j \in C_i$.

1 初始化 $\tilde{\mathcal{N}} \leftarrow \mathcal{N}$;

2 **while** $\tilde{\mathcal{N}} \neq \emptyset$ **do**

3 $\mathbf{x} \leftarrow LP(\mathcal{A}, \mathcal{B}, \mathcal{Q}^{(ref)}, \mathcal{Q}, \Theta)$;

4 $\mathbf{x}_{n^*} \leftarrow \underset{n \in \mathcal{N}}{\operatorname{argmin}} \pi_n$;

5 “冻结” 用户请求 n^* 的服务选择结果 \mathbf{x}_{n^*} ;

6 对于任何 $n \neq n^*$, 设定 $x_{i,j}^n \leftarrow 0$;

7 $\Theta \leftarrow \Theta \setminus \{x_{i,j}^n \mid n = n^*\}$;

8 $\Theta \leftarrow \Theta \cap \{x_{i,j}^n \mid x_{i,j}^{n^*} = 1, i \in S_{n^*}, j \in C_i\}$;

9 $\tilde{\mathcal{N}} \leftarrow \tilde{\mathcal{N}} \setminus \{n^*\}$;

10 **return** $x_{i,j}^n, \forall n \in \mathcal{N}, i \in S_n, j \in C_i$;

对最低服务费用最大化问题的等价线性规划变换

等价线性规划变换

整数最优解保证：如果线性规划问题的系数矩阵满足完全幺模（Totally Unimodular, 缩写 TU）特性，则线性规划问题的最优解一定是整数解^[61]。在本章所定义的服务选择优化问题中，研究由约束式（3-5）和式（3-6）组成的系数矩阵，以验证系数矩阵是否满足 TU 特性。若是的话，则可以剔除整型定义域约束式（3-7），完全不影响本章服务选择优化问题的正常求解。

引理 1：由约束式（3-5）和式（3-6）组成的系数矩阵符合 TU 特性。

等价线性规划变换

整数最优解保证 - 引理 1 (完全幺模性)



等价的凸优化目标: 为了得到线性优化目标方程, 首先针对性设计出可分解的凸优化目标方程 $\xi(\varpi)$, 作为从非线性优化目标到线性优化目标转化的中间数学表达形式, 如式(3-13)所示。向量 ϖ 的第 k 个元素用 ϖ_k 表示。

$$\xi(\varpi) = \sum_{k=1}^{|\varpi|} |\varpi|^{-\varpi_k} = \sum_{k=1}^K K^{-\varpi_k} \quad \text{式 (3-13)}$$

引理 2: $\xi(\cdot)$ 颠倒了原始字典序优化问题的优化优先级 (\succeq), 即

$$\varpi(x^*) \succeq \varpi(x) \Leftrightarrow \xi(\varpi(x^*)) \leq \xi(\varpi(x))$$

等价线性规划变换

整数最优解保证 - 引理 1 (完全幺模性)



等价的凸优化目标 - 引理 2

$$\xi(\varpi) = \sum_{k=1}^{|\varpi|} |\varpi|^{-\varpi_k} = \sum_{k=1}^K K^{-\varpi_k}$$



等价的线性规划变换 - 引入 λ -技术:

- 优化目标式: $\psi_{i,j}^n(x_{i,j}^n) = \sum_{p \in \{0,1\}} K^{-\left[a_n + b_n \times \left(1 - \frac{Q_{i,j}}{Q_n^{(ref)}} \cdot p\right)\right]} \lambda_{i,j}^{n,p} = K^{-(a_n + b_n)} \lambda_{i,j}^{n,0} + K^{-\left[a_n + b_n \times \left(1 - \frac{Q_{i,j}}{Q_n^{(ref)}}\right)\right]} \lambda_{i,j}^{n,1}$
- 优化约束式: $\sum_{p \in \{0,1\}} \lambda_{i,j}^{n,p} = \lambda_{i,j}^{n,0} + \lambda_{i,j}^{n,1} = 1 , \quad x_{i,j}^n = \sum_{p \in \{0,1\}} p \cdot \lambda_{i,j}^{n,p} = \lambda_{i,j}^{n,1}$

等价线性规划变换

整数最优解保证- 引理 1 (完全幺模性)



等价的凸优化目标- 引理 2



等价的线性规划变换- 引入 λ -技术

$$\min_{x, \lambda} \sum_{n \in \mathcal{N}} \sum_{i \in S_n} \sum_{j \in C_i} K_0 \cdot \lambda_{i,j}^{n,0} + K_1 \cdot \lambda_{i,j}^{n,1}$$

$$\text{s.t. } \sum_{i \in S_n} \sum_{j \in C_i} x_{i,j}^n = 1, \forall n \in \mathcal{N} \quad \lambda_{i,j}^{n,0} + \lambda_{i,j}^{n,1} = 1 \quad \forall n \in \mathcal{N}, i \in S_n, j \in C_i$$

$$\sum_{n \in \mathcal{N}_i} x_{i,j}^n \leq 1 \quad \forall i \in \mathcal{M}, \forall j \in C_i \quad x_{i,j}^n, \lambda_{i,j}^{n,0}, \lambda_{i,j}^{n,1} \in \mathbb{R}^+ \quad \forall n \in \mathcal{N}, i \in S_n, j \in C_i$$

$$x_{i,j}^n = \lambda_{i,j}^{n,1} \quad \forall n \in \mathcal{N}, i \in S_n, j \in C_i \quad K_0 = K^{-(a_n + b_n)}, K_1 = K^{-\left[a_n + b_n \times \left(1 - \frac{Q_{i,j}}{Q_n^{(ref)}}\right)\right]}$$

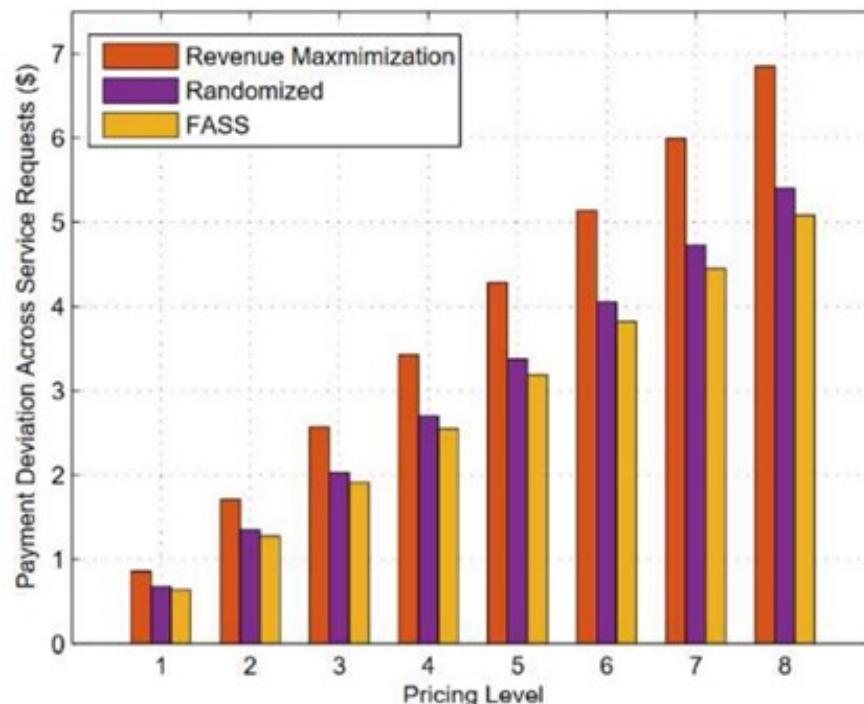
实验验证

❖ 实验设置

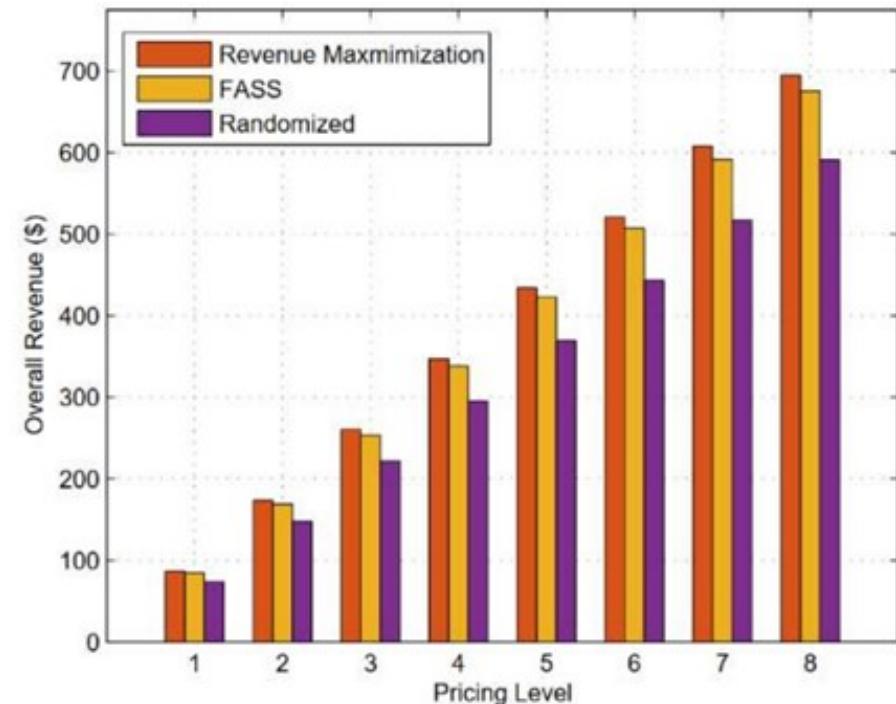
- 服务供应商数 $M = 9$; 并发用户请求数 $N = 10$
- 候选服务的 QoS 值源自 **WSDream** 数据集 [65]
 - ✓ 记录来自不同地理位置的 5825 种真实 Web 服务的响应时间
 - ✓ 从 5825 种 Web 服务中随机选择 **9 种 Web 服务** 作为服务供应商
 - ✓ 来自不同地理位置的同种 Web 服务被作为 **一个候选服务集合**
- 仿真环境
 - ✓ C++ 编程语言
 - ✓ 调用 IBM CPLEX 求解器 [64] 来求解 LP 优化问题

实验验证

❖ 实验结果



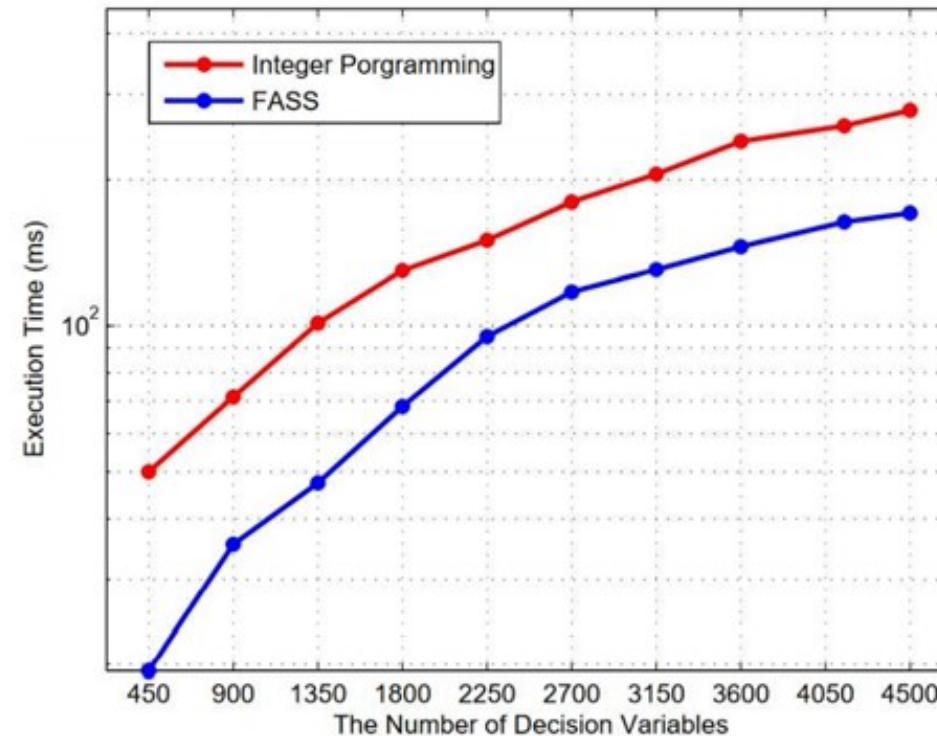
不同算法下的多用户服务付费偏差



不同算法下的服务供应商总服务收益

实验验证

❖ 实验结果

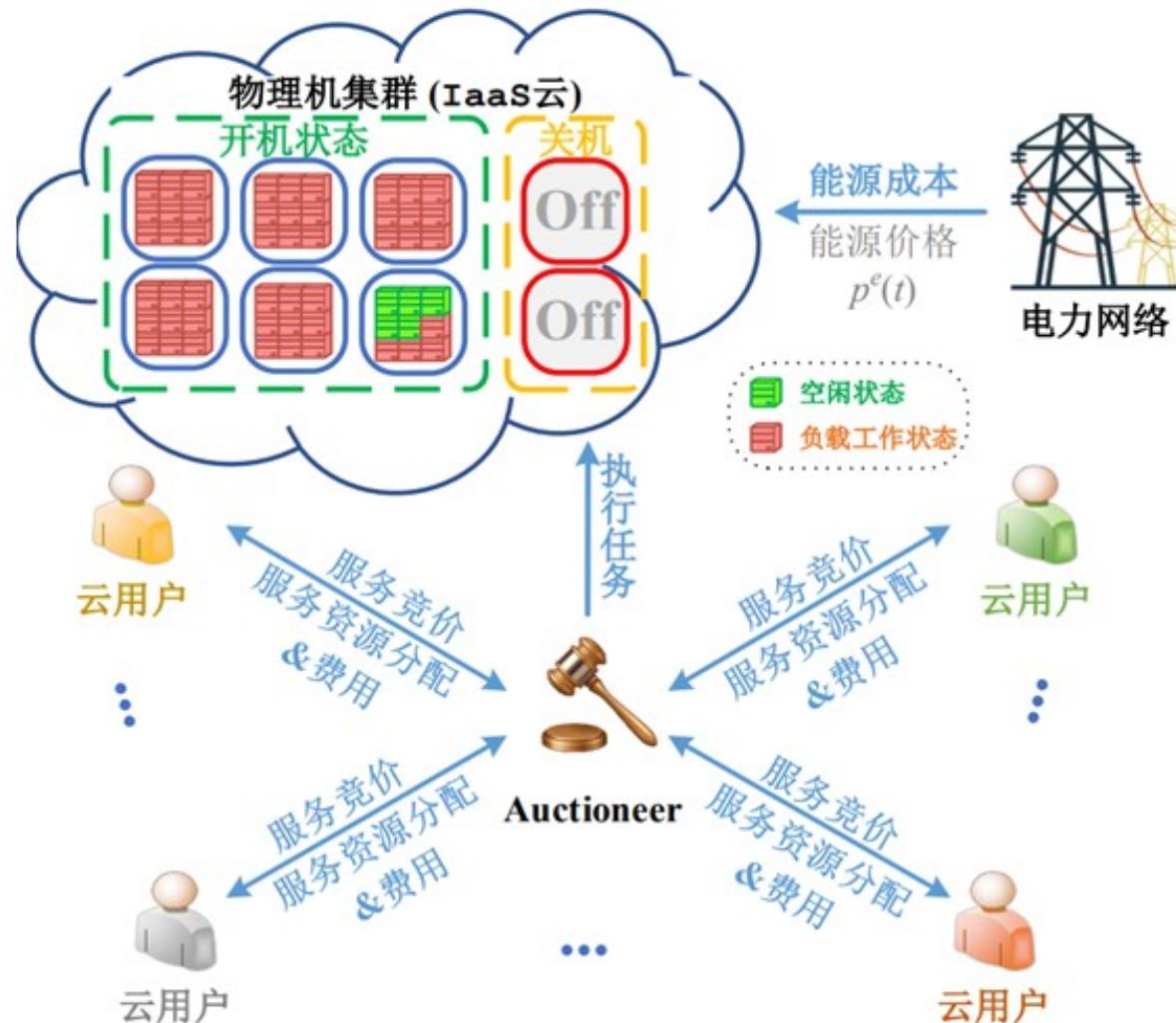


不同优化问题规模下的算法执行时间

主要内容

- ❖ 引言
- ❖ 相关研究综述
- ❖ 基于公平性指标的多用户服务选择方法
- ❖ **面向市场的云资源定价与售卖机制**
- ❖ QoE 感知的分布式边缘任务调度和资源管理
- ❖ 总结与展望

面向多用户的云资源拍卖市场



云用户服务竞价请求

- ❖ $\beta_i(t) = (s_i, b_i, \chi_i, t_i^-, t_i^+)$
 - s_i : 服务竞价请求 $\beta_i(t)$ 的服务类型
 - b_i : 服务竞价请求 $\beta_i(t)$ 的服务竞价预算
 - χ_i : 服务竞价请求 $\beta_i(t)$ 的云托管任务大小
 - $[t_i^-, t_i^+]$: 服务竞价请求 $\beta_i(t)$ 的 QoS 需求范围

云用户效用模型

- ❖ 云用户 u_i 获得的服务响应时间 $t_i(a_i(t))$ [80] :

$$t_i(a_i(t)) = \theta_{s_i,0} + \theta_{s_i,1} \cdot \frac{\chi_i}{a_i(t)} + \theta_{s_i,2} \cdot a_i(t) + \theta_{3,s_i} \cdot \log(a_i(t))$$

- ❖ $[a_i^-, a_i^+]$: 服务竞价请求 $\beta_i(t)$ 的云资源需求量范围

$$a_i^- = t_i^{-1}(t_i^+), \quad a_i^+ = t_i^{-1}(t_i^-)$$

- ❖ QoS 增益率 $\rho_i(a_i(t))$:

$$\rho_i(a_i(t)) = \frac{t_i^+}{t_i(a_i(t))}$$
 强非凸函数

云用户效用模型

- ❖ 云用户 u_i 获得的服务响应时间 $t_i(a_i(t))$
- ❖ $[a_i^-, a_i^+]$: 服务竞价请求 $\beta_i(t)$ 的云资源需求量范围
- ❖ QoS 增益率 $\rho_i(a_i(t))$

当用户 u_i 被分配到单位量为 $a_i(t)$ 的云资源时，所获得的“幸福”程度

$$g_i(a_i(t)) = \begin{cases} \rho_i(a_i^+) & \text{if } a_i(t) \in (a_i^+, +\infty) \\ \rho_i(a_i(t)) & \text{if } a_i(t) \in [a_i^-, a_i^+] \\ 0 & \text{if } a_i(t) \in [0, a_i^-] \end{cases}$$

云用户效用模型

当用户 u_i 被分配到单位量为 $a_i(t)$ 的云资源时，所获得的“幸福”程度

$$g_i(a_i(t)) = \begin{cases} \rho_i(a_i^+) & \text{if } a_i(t) \in (a_i^+, +\infty) \\ \rho_i(a_i(t)) & \text{if } a_i(t) \in [a_i^-, a_i^+] \\ 0 & \text{if } a_i(t) \in [0, a_i^-] \end{cases}$$



用户效用函数

$$v_i(p(t), a_i(t)) = \begin{cases} \rho_i(a_i^+) - \frac{p(t) \cdot a_i(t)}{b_i} & \text{if } a_i(t) \in (a_i^+, +\infty) \\ \rho_i(a_i(t)) - \frac{p(t) \cdot a_i(t)}{b_i} & \text{if } a_i(t) \in [a_i^-, a_i^+] \\ -\frac{p(t) \cdot a_i(t)}{b_i} & \text{if } a_i(t) \in [0, a_i^-] \end{cases}$$

基于用户个体理性的云资源购买策略

❖ 云用户资源需求量

$$d_i(p(t)) \triangleq \arg \max_{a_i(t)} v_i(p(t), a_i(t))$$

基于用户效用最大化

- 情况 1：云用户 u_i 的服务竞价预算 b_i 不足以承担其购买单位量为 $a_i(t) > a_i^-$ 的云资源。在这种情况下，云用户 u_i 的 QoS 需求 $[t_i^-, t_i^+]$ 不能得到满足。若云用户 u_i 决定购买云资源，则反而会产生负用户效用。因此，基于个体理性，云用户 u_i 会放弃购买云资源（即 $d_i(p(t)) = 0$ ）。
- 情况 2：云用户 u_i 有充足的服务竞价预算 b_i 来购买单位量为 $a_i(t) > a_i^-$ 的云资源。在这种情况下，基于个体理性，云用户 u_i 不会购买单位量为 $a_i(t) > a_i^+$ 的云资源，因为其 QoS 需求 $[t_i^-, t_i^+]$ 会被过度满足，徒增额外的服务费用。因此，最大化用户效用的云资源需求 $d_i(p(t))$ 应该在 $a_i(t) \in [a_i^-, a_i^+]$ 范围内搜索。

基于用户个体理性的云资源购买策略

- 情况 2: 云用户 u_i 有充足的服务竞价预算 b_i 来购买单位量为 $a_i(t) > a_i^-$ 的云资源。在这种情况下, 基于个体理性, 云用户 u_i 不会购买单位量为 $a_i(t) > a_i^+$ 的云资源, 因为其 QoS 需求 $[t_i^-, t_i^+]$ 会被过度满足, 徒增额外的服务费用。因此, 最大化用户效用的云资源需求 $d_i(p(t))$ 应该在 $a_i(t) \in [a_i^-, a_i^+]$ 范围内搜索。

利用 KKT 条件方程式 (Karush-Kuhn-Tucker Conditions) 求解下述 **凸优化** 问题

$$\max_{a_i(t)} \rho_i(a_i(t)) - \frac{p(t) \cdot a_i(t)}{b_i}$$

$$\text{s.t. } p(t) \cdot a_i(t) \leq b_i$$

$$a_i^- \leq a_i(t) \leq a_i^+$$

• 算法复杂度: $\mathcal{O}(2^K) = \mathcal{O}(1)$ ($K=3$)

基于用户个体理性的云资源购买策略

Algorithm 1: 云资源需求分配算法(RARD)

Input: 即时云资源单位价格 $p(t)$; 在时间槽 t 期间的多用户服务竞价请求 $\mathcal{U}(t)$.

Output: 多用户云资源分配方案 $A(t)$.

```
1 for 每个 $u_i \in \mathcal{U}(t)$  do
2   if  $p(t) \cdot a_i^- > b_i$  then
3      $a_i(t) \leftarrow 0$ ;  $\triangleright$  情况1
4   else
5     通过求解优化问题(即式(4-9))的KKT方
      程式, 得到最大化云用户 $u_i$ 效用 $v_i(a_i(t))$ 的云
      资源需求分配结果 $a_i(t) = d_i(p(t))$ ;  $\triangleright$  情况2
6 return  $A(t) \leftarrow \langle a_i(t) \rangle_{i=1}^{N(t)}$ ;
```

- RARD算法复杂度(串行): $\mathcal{O}(N(t))$
- RARD算法复杂度(并行): $\mathcal{O}(1)$

最大化用户激励的云资源拍卖机制 PIRA

❖ 云用户激励最大化问题

- **优化目标:** 最大化选择云端服务的用户数量

$$\max_{a_i(t), p(t)} \sum_{u_i \in \mathcal{U}(t)} I_{\{a_i(t) > 0\}}$$

$$\text{s.t. } p(t) \cdot a_i(t) \geq (1 + \gamma) \cdot p^e(t) \cdot \tilde{c} \cdot m(t) \quad \Rightarrow \quad \text{云服务供应商的最低服务利润率}\gamma\text{约束}$$

$$a_i(t) = d_i(p(t)) \quad \forall u_i \in \mathcal{U}(t)$$

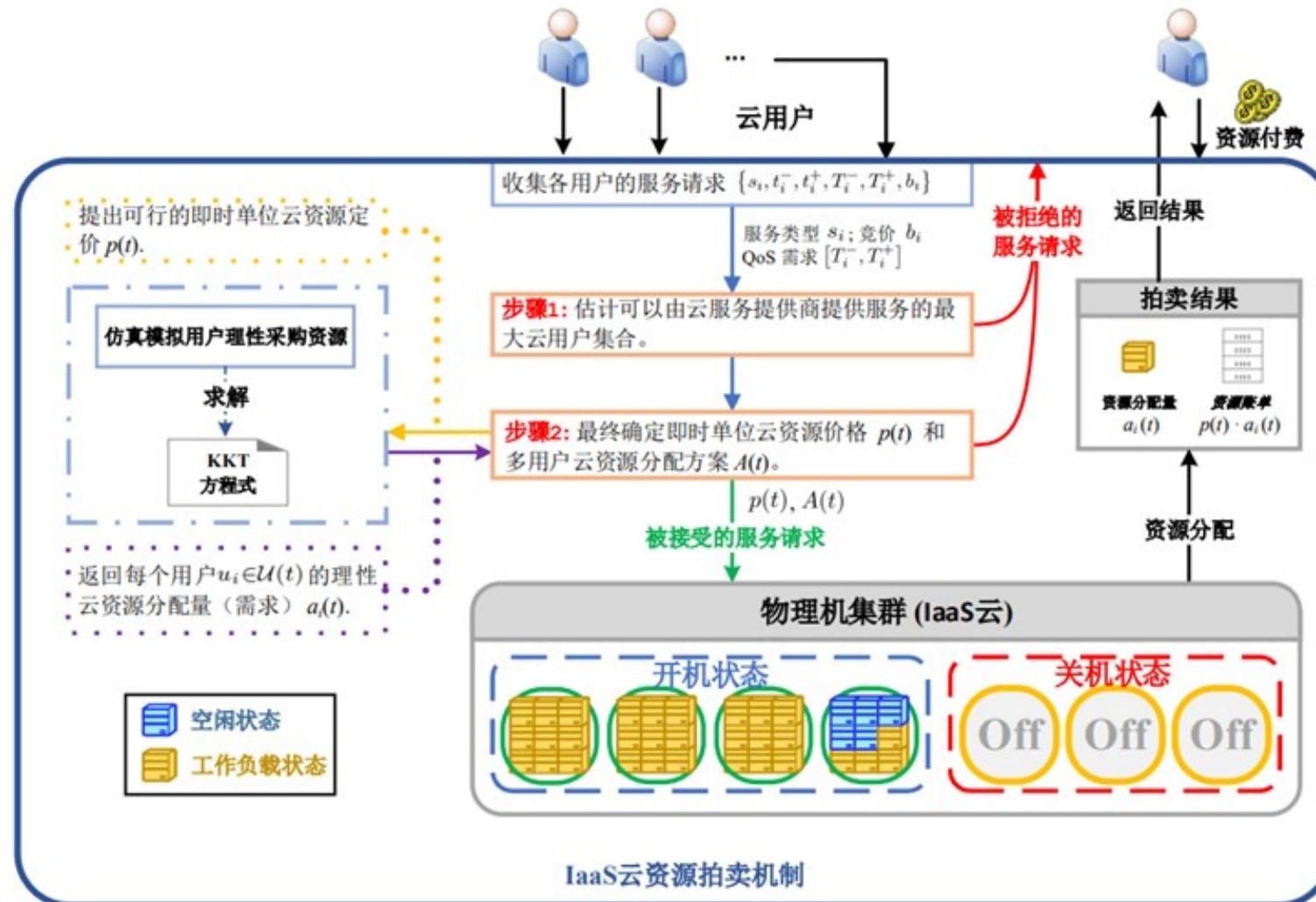
⇒ 基于用户效用最大化的云资源购买策略

云用户激励最大化问题

■ 求解难点

- 由式(4-12)定义的优化问题属于装箱问题,是NP难解的。云端部署的物理机器可视作箱子,而云用户所需求的云资源可被看作要装入箱子的物品。由于装箱问题是NP难解的,所以由式(4-12)定义的优化问题也是NP难解的。
- 另一个问题求解难点来自于用户效用函数 $v_i(\cdot)$ 的非连续性和非凸性。在由式(4-12)定义的优化问题中,每个云用户 u_i 的资源购买量 $a_i(t)$ 是基于其自身用户效用最大化策略来确定的。这会使得优化问题的解决变得更加复杂。
- 最后一个问题求解难点体现在决策变量 $p(t)$ 和 $A(t)$ 之间的复杂关联关系。对云资源价格 $p(t)$ 的决策进一步影响了云资源分配方案 $A(t)=\langle a_i(t) \rangle_{u_i \in \mathcal{U}(t)}$ 。与此同时,多用户云资源分配方案 $A(t)$ 又决定了优化结果 $\sum_{u_i \in \mathcal{U}(t)} I_{\{a_i(t)>0\}}$,并与云服务供应商所要求的最低利润率 γ 是否被满足密切相关。

最大化用户激励的云资源拍卖机制 PIRA



最大化用户激励的云资源拍卖机制 PIRA

- 步骤 1：以较粗略而高效的方式，粗估最大云用户集合 $\hat{\mathcal{U}}(t)$
 - ☞ 云服务供应商应该为最多的云用户满足其 QoS 需求 $[t_i^-, t_i^+]$ ，而不打破价格下限 $\lfloor p \rfloor$ （稍后详述）。以此为基础，估计出最大云用户集合 $\hat{\mathcal{U}}(t)$ 。

$$\lfloor p \rfloor = \tilde{c} \cdot p^e(t) \cdot (1 + \gamma) / r$$

最大化用户激励的云资源拍卖机制 PIRA

- 步骤 1：以较粗略而高效的方式，粗估最大云用户集合 $\hat{\mathcal{U}}(t)$



- 步骤 2：最终确定即时云资源单位价格 $p(t)$ 和云资源分配方案 $A(t)$

- 根据在步骤 1 中得到的粗估结果 $\hat{\mathcal{U}}(t)$ ，云服务供应商进一步最终确定最大云用户集合 $\tilde{\mathcal{U}}(t)$ 。当云用户 $u_i \in \tilde{\mathcal{U}}(t)$ 被云服务供应商服务时，最低利润率 γ 能够得到切实的保障。
- 同时，即时云资源单位价格 $p(t)$ 被优化求解，使得云服务供应商尽可能赚取更多的服务收益。基于云资源单位价格 $p(t)$ ，通过模拟每个云用户 $u_i \in \tilde{\mathcal{U}}(t)$ 的理性云资源购买过程，得到相应的云资源分配方案 $A(t) = \langle a_i(t) \rangle_{u_i \in \tilde{\mathcal{U}}(t)}$ 。

最大化用户激励的云资源拍卖机制 PIRA

■ 云资源定价算法(步骤2)

Algorithm 2: 服务收益最大化的云资源定价算法(RORP)

Input: 云资源价格范围 $[p(t)^-, p(t)^+]$.

Output: 即时云资源单位价格 $p(t)$;
多用户云资源分配方案 $A(t)$.

- 1 初始化 $\pi \leftarrow 0$, $p(t) \leftarrow \text{null}$, $A(t) \leftarrow \text{null}$;
 - 2 按照式 (4-21), 计算候选云资源价格数量 X ;
 - 3 将连续的云资源价格范围 $[p(t)^-, p(t)^+]$ 离散采样为 X 个候选价格, 其中 $\hat{p}_x \leftarrow p(t)^- \cdot (1 + \epsilon)^{x-1}$, $x = 1, 2, \dots, X$;
 - 4 **for** 每个 $x = \{1, 2, \dots, X\}$ **do**
 - 5 $A(t) \leftarrow \text{RARD}(\hat{p}_x, \mathcal{U}(t))$;
 - 6 **if** $\frac{\hat{p}_x \cdot \sum_{u_j \in \mathcal{U}(t)} a_i(t)}{(1+\gamma)} \geq \tilde{c} \cdot p^e(t) \cdot \left\lceil \frac{\sum_{u_j \in \mathcal{U}(t)} a_i(t)}{r} \right\rceil$
 - 7 **and** $\hat{p}_x \cdot \sum_{u_i \in \mathcal{U}(t)} a_i(t) > \pi$ **then**
 - 8 $\pi \leftarrow \hat{p}_x \cdot \sum_{u_i \in \mathcal{U}(t)} a_i(t)$;
 - 9 $p(t) \leftarrow \hat{p}_x$;
 - 10 $A(t) \leftarrow \langle a_i(t) \rangle_{u_i \in \mathcal{U}(t)}$;
 - 11 **return** $p(t)$ and $A(t)$;
-

最大化用户激励的云资源拍卖机制 PIRA

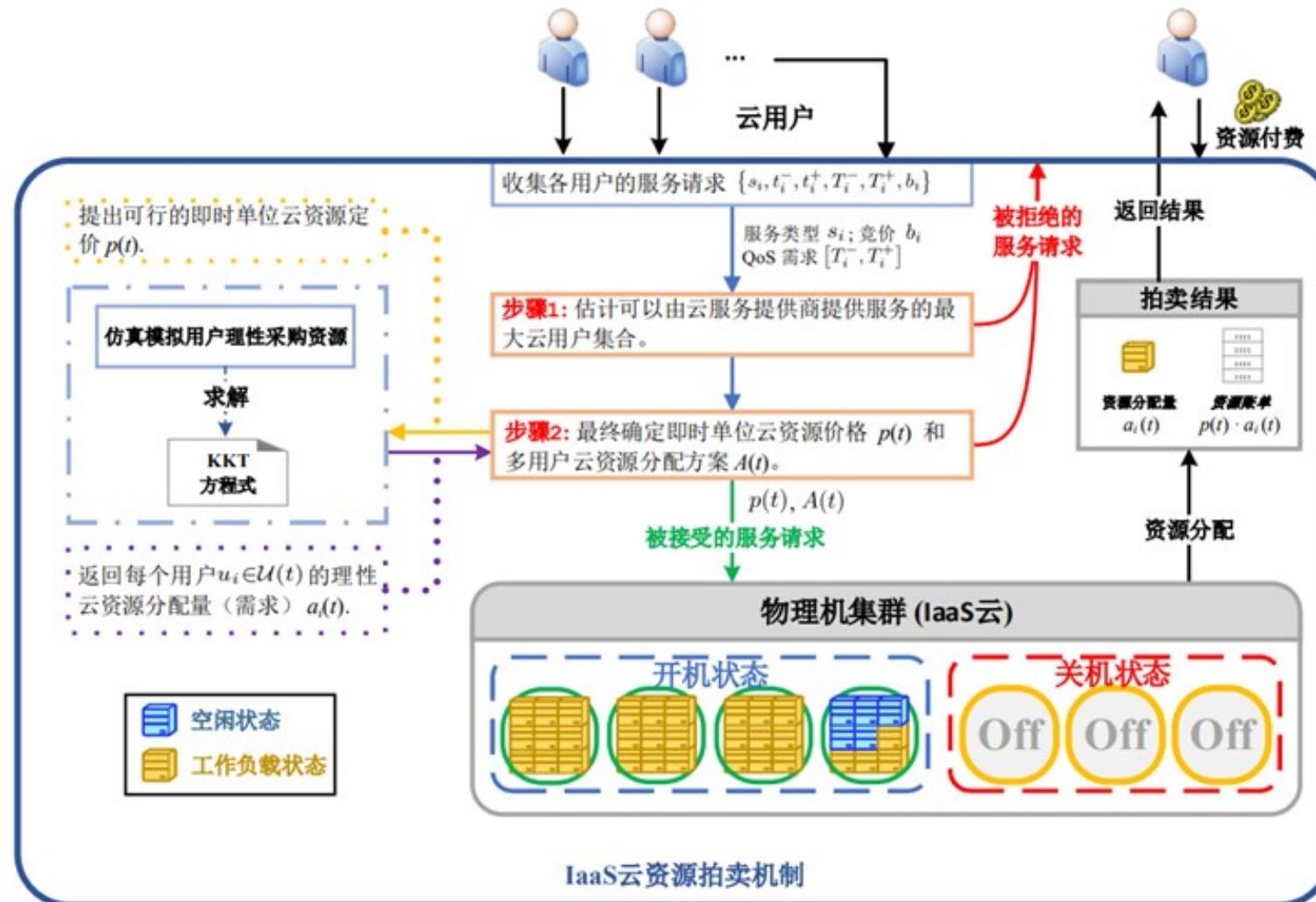
RORP 算法步骤

1. 将云资源价格范围 $[p(t)^-, p(t)^+]$ 离散采样为 X 个候选价格 \hat{p}_x , $x=1,2,\dots,X$ (第 1-3 行);
2. 在 X 个候选价格之间, 找出云资源价格的近似最优解 $\hat{p}(t) = \hat{p}_x$ ($x \in \{1,2,\dots,X\}$)。当即时云资源价格 $p(t)$ 被定为 $\hat{p}(t)$ 时, 云服务供应商可赚取 $(1+\epsilon)$ -近似比的最高服务收益 (第 4-10 行)。

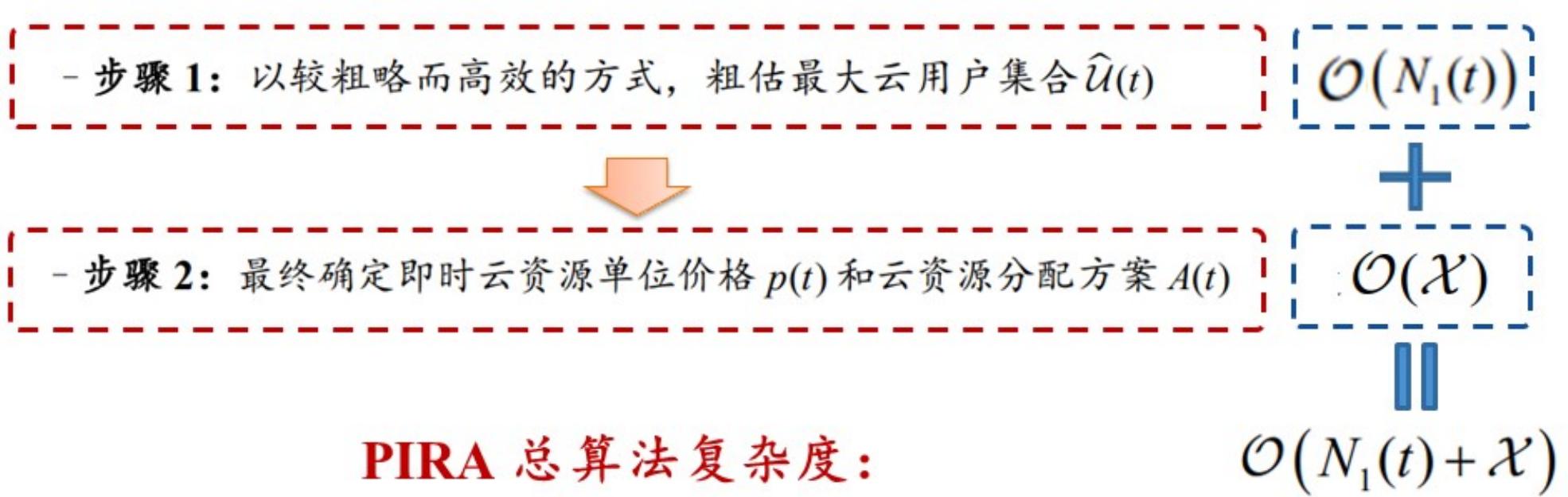
$$X = \left\lfloor \frac{\log(p(t)^+ / p(t)^-)}{\log(1+\epsilon)} \right\rfloor + 1$$

定理 1: RORP 算法提供了 $(1+\epsilon)$ -近似的服务收益最大化。当 $N(t)$ 名云用户并发地操作云资源购买过程时 (详见 RARD 算法), RORP 算法的并行计算复杂度为 $\mathcal{O}(X)$ 。

最大化用户激励的云资源拍卖机制 PIRA



最大化用户激励的云资源拍卖机制 PIRA



PIRA机制 - 重要性质分析

云资源拍卖机制 PIRA 将满足如下三个重要性质：

- **预算可行性 (Budget Fesibility)**: 规定拍卖者（云用户）的服务付费不应该超出服务竞价预算^[87]，这通常被视为拍卖机制的基本要求。
- **激励相容性 (Incentive Compatibility)**: 禁止拍卖者（云用户）的机会主义行为^[88]。无论其他竞拍者（云用户）采用何种竞价策略，拍卖者（云用户）都会上报真实的服务竞价预算，没有动机谎报。
- **无妒性 (Envy-freeness)**: 保障多用户拍卖过程的公平性。每个竞拍者（云用户）都可以得到最大化其用户效用的云资源分配量^[89]。这样的话，每个竞拍者（云用户）不会嫉妒其他竞拍者（云用户）的云资源分配结果，体现了多用户云资源分配方案的公平性。

PIRA机制 - 重要性质分析

相应地，给出详细定理证明过程：

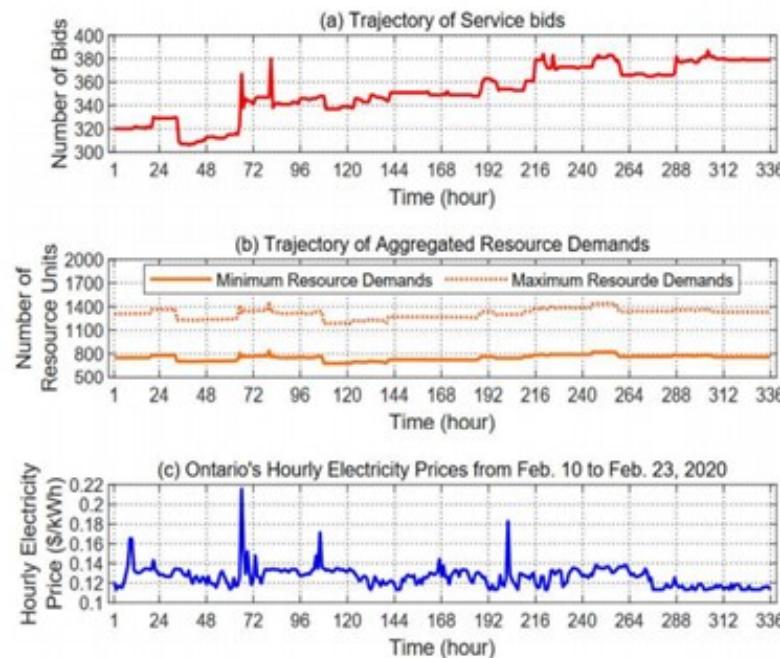
定理 2 (预算可行性): 云用户 u_i 的服务竞价预算 b_i 始终足以覆盖购买云资源 $a_i(t) = d_i(p(t))$ 所需的服务费用。

定理 3 (激励相容性): 云用户 u_i 不会向云服务供应商谎报服务竞价预算 b_i 。

定理 4 (无妒性): 云用户 u_i 更偏爱分配给自己的云资源结果 $a_i(t) = d_i(p(t))$ ，而不是嫉妒其他云用户的云资源分配量。

实验验证

❖ 实验设置



数据集概览

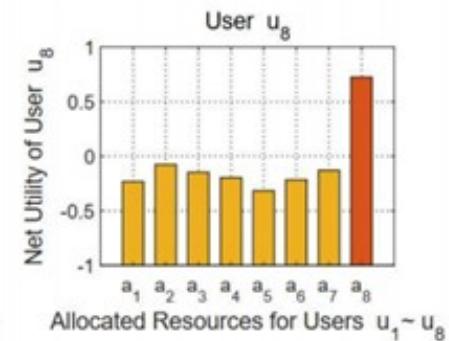
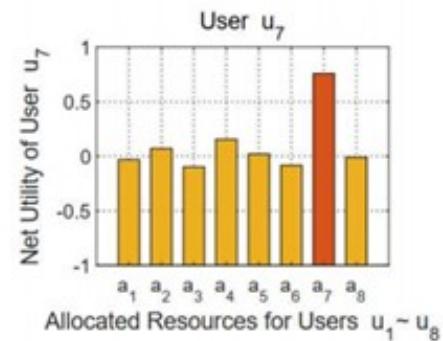
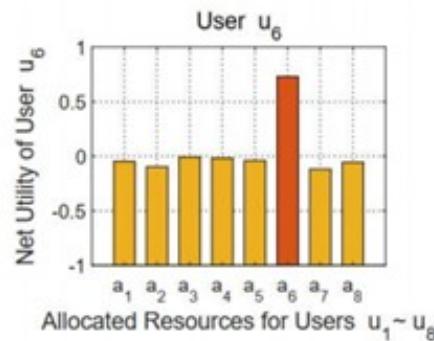
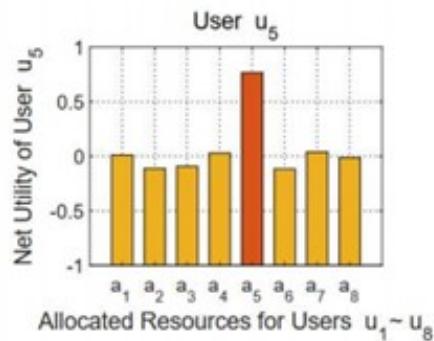
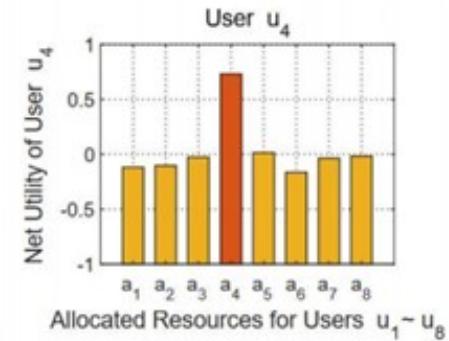
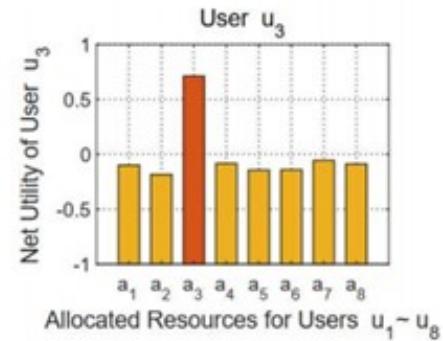
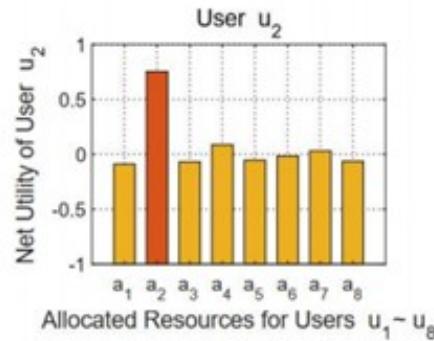
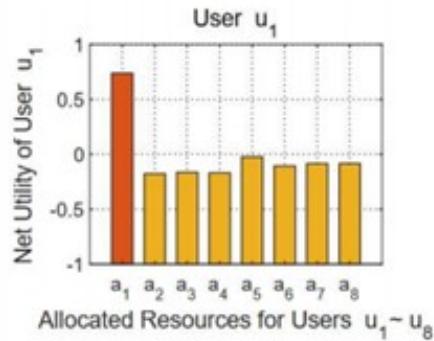
* 使用微软Azure开源数据集

实验参数设置

符号	值
τ	1 小时 [76]
\tilde{c}	600 瓦特 [77]
γ	0.2
ϵ	0.02
r	10
b_i	$((a_i^- + a_i^+)/2) \cdot \mathcal{N}(0.02, 0.015^2)$ 美元

实验验证

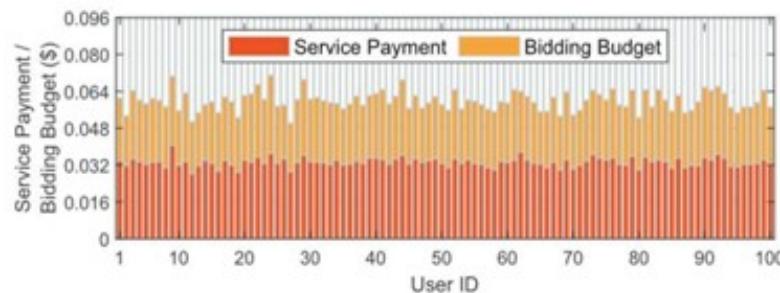
❖ 实验结果



云用户 $u_1 \sim u_8$ 对不同云资源分配结果 $a_1 \sim a_8$ 的净效用比较

实验验证

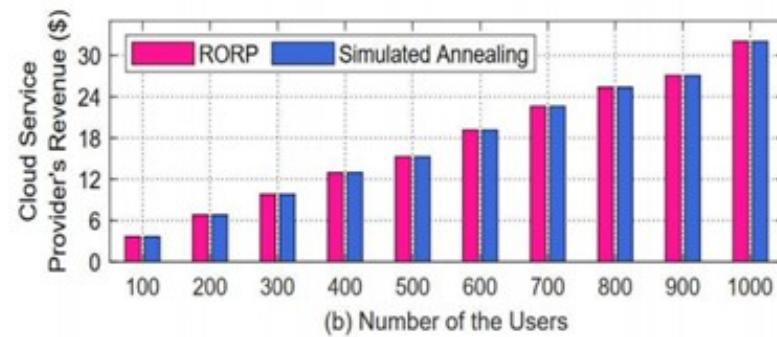
❖ 实验结果



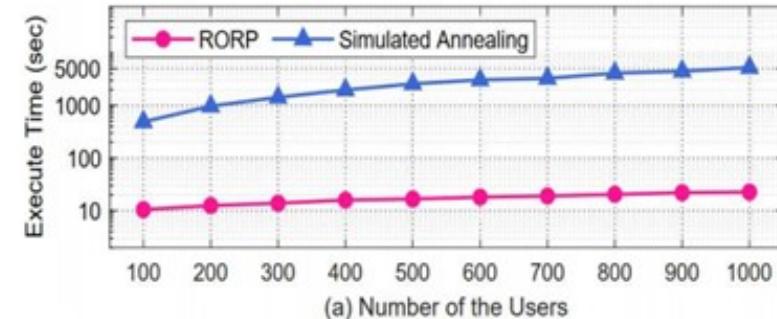
服务费用 v.s. 服务竞价预算



在不同最低利润率 γ 要求下的云资源拍卖结果



(b) Number of the Users

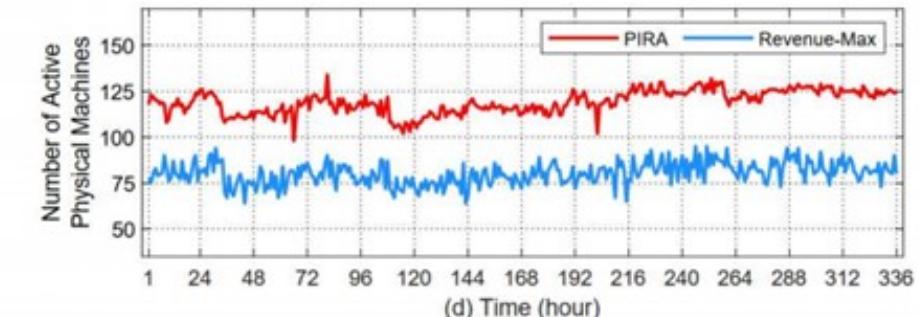
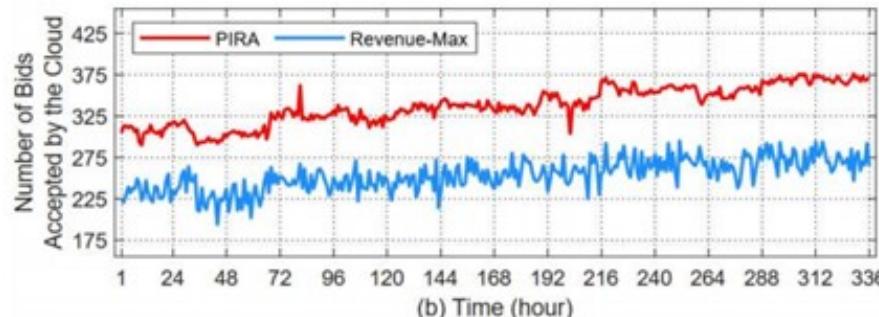
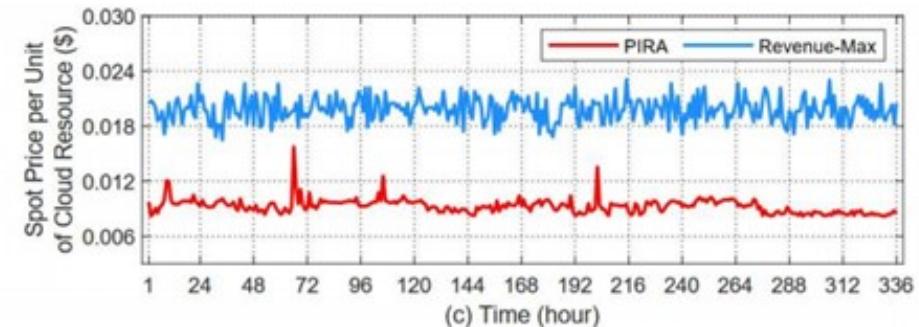
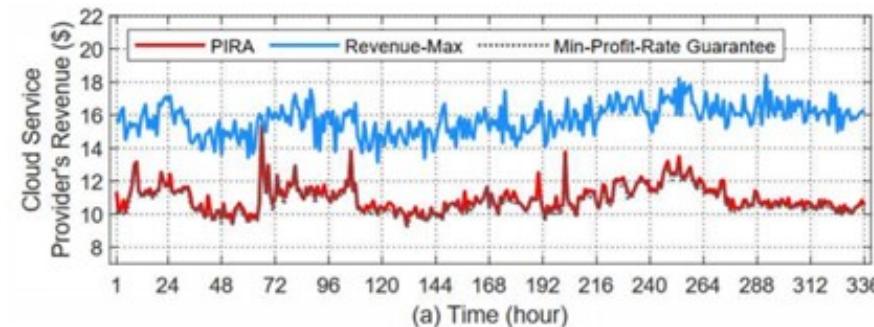


(a) Number of the Users

RORP 算法 v.s. 模拟退火算法^[58]

实验验证

❖ 实验结果

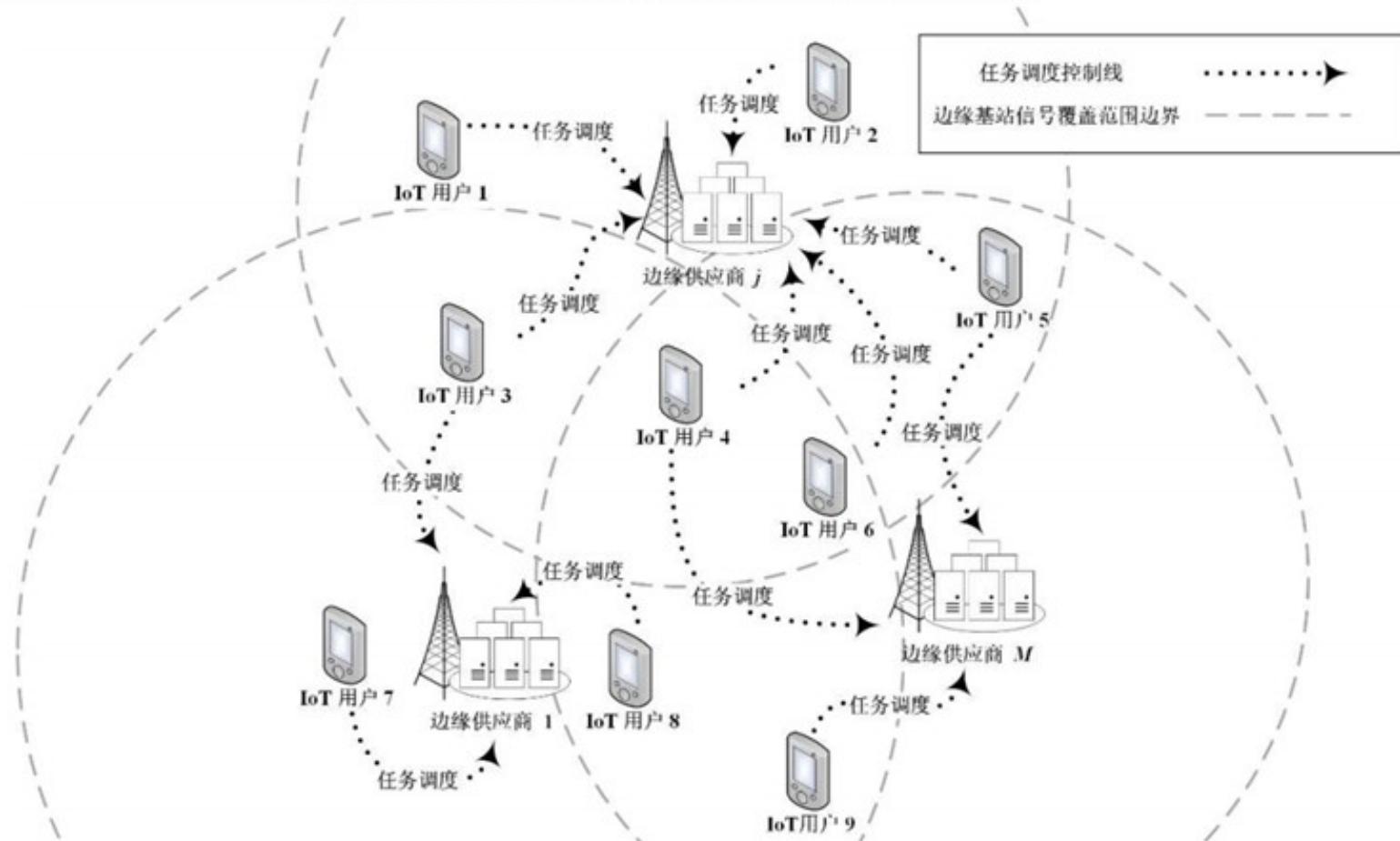


基于实际轨迹数据的对比仿真结果

主要内容

- ❖ 引言
- ❖ 相关研究综述
- ❖ 基于公平性指标的多用户服务选择方法
- ❖ 面向市场的云资源定价与售卖机制
- ❖ **QoE 感知的分布式边缘任务调度和资源管理**
- ❖ 总结与展望

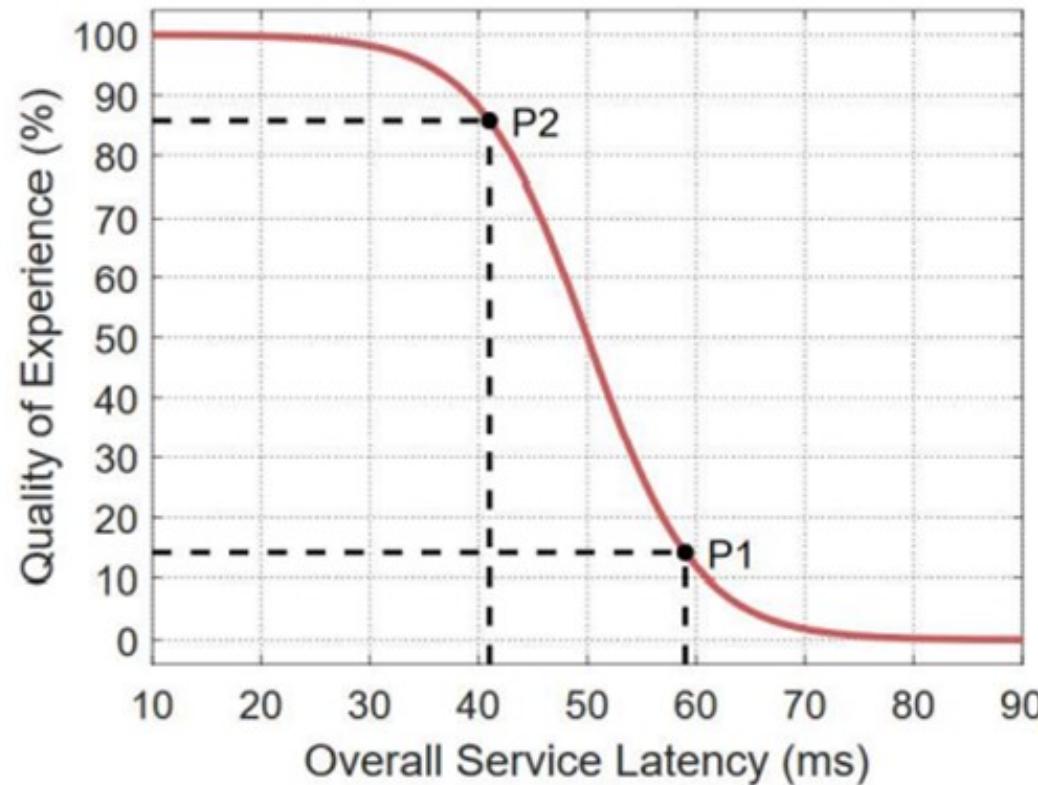
系统模型



边缘计算架构下的 IoT 任务调度和资源管理

用户体验质量 (QoE) 模型

❖ QoS / QoE 指标间的关联性分析



用户体验质量 (QoE) 模型

❖ 用户体验质量 (QoE) 指标

物联网用户 u_i 在本地设备上处理服务请求

$$QoE_i^L = \frac{1}{1 + e^{\alpha_i(t_i^L - \beta_i)}}$$

Sigmoid 函数

物联网用户 u_i 按照策略 $\mathbf{s}_i = (x_i, a_i)$ 将服务请求置于边缘服务器处理

$$QoE_i^E(\mathbf{s}_i) = \frac{1}{1 + e^{\alpha_i(t_{i,x_i}^E(a_i) - \beta_i)}}$$

Sigmoid 函数

优化问题

❖ 边缘资源分配策略

$$\pi_i(s_i) = \begin{cases} QoE_i^E & \text{if } QoE_i^E > QoE_i^L \\ QoE_i^L & \text{otherwise} \end{cases}$$

❖ 边缘资源分配(ERA) 优化问题

$$\max_{x_i, a_i} \sum_{i \in \mathcal{U}} \pi_i(s_i)$$



优化目标：最大化系统 QoE 水平

$$\sum_{i \in \mathcal{U}_j} a_i \cdot I_{\{x_i=j\}} \leq c_j \quad \forall j \in \mathcal{E}$$



边缘服务器资源容量约束

$$x_i \in \{0\} \cup \mathcal{E}_i, \quad a_i \geq 0 \quad \forall i \in \mathcal{U}$$



物联网用户服务请求约束

多用户边缘资源博弈 (ERAGame)

定义 1 (ERAGame) : 策略博弈 \mathcal{G} 被用于描述 N 位物联网用户之间对有限边缘服务器资源的竞争态势，由三元组 $\langle \mathcal{U}, (\mathbf{S}_i)_{u_i \in \mathcal{U}}, (\pi_i)_{u_i \in \mathcal{U}} \rangle$ 定义。

- \mathcal{U} 是参与策略博弈 \mathcal{G} 的局中人 (Player) 集合；具体指边缘计算系统中的 N 位物联网用户。多位局中人 (物联网用户) 之前相互竞争，以期被分配到更多边缘服务器资源，从而获得更高的 QoE 水平。
- \mathbf{S}_i 是物联网用户 u_i 的博弈策略集合；每位物联网用户 u_i 的所有可行策略 $\mathbf{s}_i = (x_i, a_i)$ 组成了其博弈策略集合 \mathbf{S}_i 。物联网用户 u_i 的博弈策略 $\mathbf{s}_i \in \mathbf{S}_i$ 指定了其服务请求被调度到的边缘服务器 e_{x_i} ，以及相应被分配的边缘服务器资源量。
- π_i 是物联网用户 u_i 的效用函数，由式 (5-8) 或者式 (5-9) 表示。物联网用户 u_i 效用 π_i 被用于评估博弈策略 $\mathbf{s}_i = (x_i, a_i)$ 所获得的 QoE 水平。□

多用户边缘资源博弈 (ERAGame)

定义 1 (ERAGame) : 策略博弈 \mathcal{G} 被用于描述 N 位物联网用户之间对有限边缘服务器资源的竞争态势, 由三元组 $\langle \mathcal{U}, (\mathbf{S}_i)_{u_i \in \mathcal{U}}, (\pi_i)_{u_i \in \mathcal{U}} \rangle$ 定义。



定义 2 (纳什均衡, 缩写 NE) : ERAGame 博弈 $\mathcal{G} = \langle \mathcal{U}, (\mathbf{S}_i)_{u_i \in \mathcal{U}}, (\pi_i)_{u_i \in \mathcal{U}} \rangle$ 的纳什均衡是指策略组合 \mathbf{s}^* , 其满足对于每位局中人 (物联网用户) $u_i \in \mathcal{U}$,

$$\pi_i(\mathbf{s}_i^*, \mathbf{s}_{-i}^*) \geq \pi_i(\mathbf{s}_i, \mathbf{s}_{-i}^*), \quad \forall \mathbf{s}_i \in \mathbf{S}_i \quad \text{式 (5-13)}$$

其中 \mathbf{s}_{-i} 表示除物联网用户 u_i 以外其他用户所组成的策略组合, 即 $\mathbf{s}_{-i} = \mathbf{s} - \{\mathbf{s}_i\}$; 而 π_i 从原有 $\pi_i(\mathbf{s}_i)$ 扩展为 $\pi_i(\mathbf{s}_i, \mathbf{s}_{-i})$, 被用于描述不同物联网用户间的边缘资源竞争态势。 \square

面向 QoE 改进的多用户博弈协调机制

定义 3（边缘资源抢占机制）：通过本轮边缘资源抢占，如果系统 QoE 水平得到提升，即

$$\Delta \pi_i > \Delta \pi_{-i} \quad \text{式 (5-14)}$$

则物联网用户 u_i 会抢占其他物联网用户 $u_k \in p(u_i)$ 所占用的部分边缘服务器资源，从而提升系统 QoE 水平。 \square

面向 QoE 改进的多用户博弈协调机制

定义 3（边缘资源抢占机制）：通过本轮边缘资源抢占，如果系统 QoE 水平得到提升，即

$$\Delta \pi_i > \Delta \pi_{-i} \quad \text{式 (5-14)}$$

则物联网用户 u_i 会抢占其他物联网用户 $u_k \in p(u_i)$ 所占用的部分边缘服务器资源，从而提升系统 QoE 水平。 \square



抢占式 QoE 改进算法 PRIM

抢占式 QoE 改进算法 PRIM

Algorithm 1: 抢占式QoE改进算法(PRIM)

Input: 物联网用户 u_i 所提出的改进策略 s'_i ,
当前策略组合 \mathbf{s} .

Output: 经过更新后的策略组合 \mathbf{s}' .

```

1 if  $\pi_i(s'_i) > \pi_i(s_i)$  then
2   if  $a'_i + \sum_{u_k \in \mathcal{U}: x_k = x'_i, k \neq i} a_k \leq c_{x'_i}$  then
3     return  $\mathbf{s}' \leftarrow \{s'_i, s_{-i}\}$ ;
4   else
5      $\Delta a_i \leftarrow a'_i - c_{x'_i} + \sum_{u_k \in \mathcal{U}: x_k = x'_i, k \neq i} a_k$ ;
6     按照 $\hat{\pi}_k$ 对物联网用户 $u_k \in \mathcal{U} : x_k = x'_i$  and  $k \neq i$   
进行非降序排序, 其经过排序的用户排列为 $P$ ;
7     初始化 $\mathbf{s}'_{-i} \leftarrow \mathbf{s}_{-i}$ ;
8     while  $\Delta a_i > 0$  do
9       选取用户排列 $P$ 中的头部用户 $u_v$ ;
10      尝试更新物联网用户 $u_v$ 的策略 $\mathbf{s}'_v \leftarrow (x_v, a_v - 1)$ ;
11      重新按照 $\hat{\pi}_k$ 对物联网用户 $u_k \in \mathcal{U} : x_k = x'_i$  and  
 $k \neq i$  进行非降序排序;
12       $\Delta a_i \leftarrow \Delta a_i - 1$ ;
13       $\Delta \pi_i \leftarrow \pi_i(\mathbf{s}'_i, \mathbf{s}'_{-i}) - \pi_i(\mathbf{s}_i, \mathbf{s}_{-i})$ ;
14       $\Delta \pi_{-i} \leftarrow \pi_{-i}(\mathbf{s}_i, \mathbf{s}_{-i}) - \pi_{-i}(\mathbf{s}'_i, \mathbf{s}'_{-i})$ ;
15      if  $\Delta \pi_i > \Delta \pi_{-i}$  then
16        return  $\mathbf{s}' \leftarrow \{s'_i, s'_{-i}\}$ ;
17      else
18        return  $\mathbf{s}' \leftarrow \mathbf{s}$ ;
19    else
20      return  $\mathbf{s}' \leftarrow \mathbf{s}$ ;

```

PRIM 算法重要变量:

$$1) \quad \hat{\pi}_k = \begin{cases} \pi_k(\mathbf{s}_k) - \pi_i(x_k, a_k - 1) & \text{if } a_i \geq 2 \\ \pi_k(\mathbf{s}_k) - \pi_i(0, 0) & \text{otherwise} \end{cases}$$

$$2) \quad \Delta a_i \leftarrow a'_i - c_{x'_i} + \sum_{u_k \in \mathcal{U}: x_k = x'_i, k \neq i} a_k$$

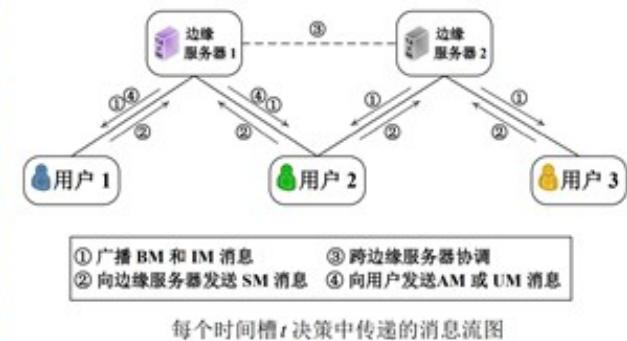
PRIM 算法复杂度:

$$\mathcal{O}(m(\log m + \Delta a_i - 1))$$

基于 ERAGame 的分布式算法QoE-DEER

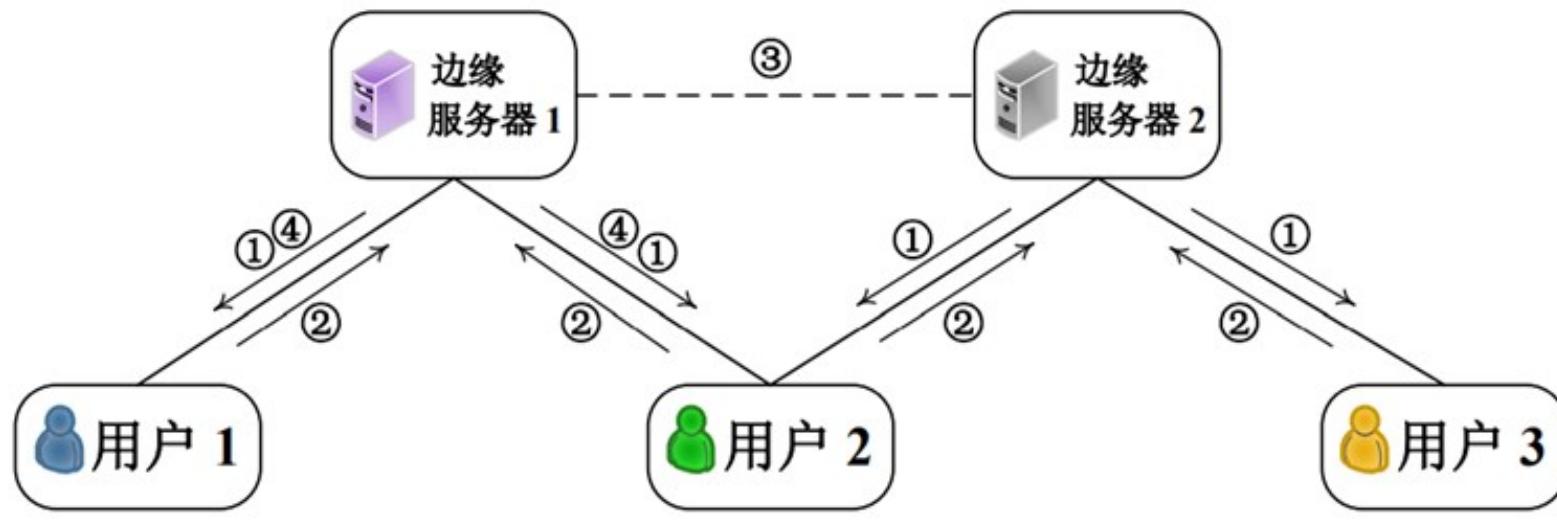
❖ 基于合作的消息传递机制

- **开始消息 (Begin Message, 简称 BM):** 只要存在请求更新的策略, 则 ERAGame 博弈就未达到纳什均衡状态。这时, 每个边缘服务器 e_j 所属的物联网用户 $u_i \in \mathcal{U}_j$ 广播 BM 消息, 表示继续 ERAGame 博弈。
- **信息消息 (Information Message, 简称 IM):** 在每个边缘服务器 e_j 向物联网用户广播 BM 消息的同时, 也通过发送 IM 消息来向物联网用户通知当前的边缘服务器资源分配状态。
- **策略消息 (Strategy Message, 简称 SM):** 收到 IM 消息后, 每个物联网用户 u_i 会决定是否请求更新当前的边缘资源分配策略。如果需要更新, 则物联网用户 u_i 发送 SM 消息给有关的边缘服务器, 以期获得策略更新的许可。
- **允许消息 (Allow Message, 简称 AM):** 在 ERAGame 博弈模型中, 每个时间槽 t 里仅能有一个用户 SM 消息所披露的策略更新请求被允许。因此, 所有边缘服务器将互相协商, 然后仅发送出一条 AM 消息给被允许执行策略更新的 AM 请求用户。该协商过程是基于全局 QoE 改进的。
- **更新消息 (Update Message, 简称 UM):** 一旦确定被批准的策略更新请求后, 所有边缘服务器也应该向其边缘资源被抢占的物联网用户发送 UM 消息, 从而告知新实施的博弈策略。



基于 ERAGame 的分布式算法 QoE-DEER

❖ 基于合作的消息传递机制



- | | |
|-------------------------------------|----------------------------------|
| ① 广播 BM 和 IM 消息
② 向边缘服务器发送 SM 消息 | ③ 跨边缘服务器协调
④ 向用户发送 AM 或 UM 消息 |
|-------------------------------------|----------------------------------|

每个时间槽 t 决策中传递的消息流图

基于 ERAGame 的分布式算法QoE-DEER

Algorithm 2: QoE感知分布式边缘资源分配算法(QoE-DEER)

```
Input: 当前时间槽 $t$ 的策略组合 $\mathbf{s}(t)$ .  
Output: 下一时间槽 $t + 1$ 的策略组合 $\mathbf{s}(t + 1)$ .  
1 初始化设置 $\mathbf{s}(t + 1) \leftarrow \mathbf{s}(t)$ ;  
2 步骤1: 继续博弈  
3 每台边缘服务器向所属物联网用户广播BM和IM消息;  
4 步骤2: 获取QoE改进的新博弈策略  
5 for 每位物联网用户 $i \in \mathcal{U}$  do  
6   设置临时变量 $\mathbf{s}' \leftarrow \mathbf{s}(t)$ ;  
7   for 每个可行策略 $\hat{\mathbf{s}}_i \in S_i$  do  
8      $\mathbf{s}' \leftarrow PRIM(\hat{\mathbf{s}}_i, \mathbf{s}')$ ;  
9   if  $\mathbf{s}' \neq \mathbf{s}(t)$  then  
10    向新策略 $\mathbf{s}'_i$ 所指明的边缘服务器发送SM消息;  
11 步骤3: 选出并实施一条QoE改进的新策略组合  
12 if 任一边缘服务器在 $\Delta t$ 时间内收到SM消息 then  
13  拣选出系统QoE提升幅度最大的新策略组合 $\mathbf{s}^*$ ;  
14  采用新策略组合 $\mathbf{s}(t + 1) \leftarrow \mathbf{s}^*$ ;  
15  向请求实施新策略组合 $\mathbf{s}^*$ 的物联网用户发送AM消息;  
16  向被抢占的物联网用户发送UM消息;  
17 else  
18  return  $\mathbf{s}(t + 1)$ ; ▷ 达到ERAGame纳什均衡状态.
```

步骤1 (第1-3行): 如果 ERAGame 博弈尚未达到纳什均衡状态, 则各个边缘服务器将向所属的物联网用户广播 BM 消息, 表明“继续博弈”。与此同时, 每个边缘服务器都会更新记录当前的边缘资源分配状态, 并将其打包为 IM 消息、再发送到每个边缘服务器所下属的物联网用户, 为各用户提供策略更新的参考。



步骤2 (第4-10行): 收到各边缘服务器发来的 IM 消息后, 每位物联网用户 u_i 将查找是否存在一个 QoE 改进的策略 s' 。如果存在, 则物联网用户 u_i 将发送 SM 消息到 s' 指定的边缘服务器。



步骤3 (第11-18行): 如果有物联网用户请求策略更新, 则边缘服务器将不会晚于 Δt 收到 SM 消息。在等待 Δt 时间后, 边缘服务器应接收到所有 SM 消息; 然后, 开始协商并决定哪个博弈策略更新请求被允许, AM 消息被作为允许策略更新的消息响应。相反, 如果在 Δt 时间范围内未收到任何 SM 消息, 则表明 ERAGame 博弈达到纳什均衡状态, 从而 QoE-DEER 算法终止。

QoE-DEER 算法分析

❖ 算法收敛性分析

- 利用潜在博弈的有限改进(Finite Improvement)性质

定义 4 (潜在博弈, Potential Game): 如果存在潜在函数 $\Phi: S \rightarrow \mathbb{R}$, 使得对于每位局中人 $u_i \in \mathcal{U}$, 从 $s = (s_i, s_{-i})$ 到 $s' = (s'_i, s'_{-i})$ 的博弈策略改进, 满足

$$\pi_i(s'_i) > \pi_i(s_i) \Rightarrow \Phi(s') > \Phi(s) \quad \text{式 (5-16)}$$

则该博弈模型 \mathcal{G} 属于潜在博弈。 □



定理 1 (ERA 潜在博弈): ERA Game 博弈 $\mathcal{G} = \langle \mathcal{U}, (S_i)_{u_i \in \mathcal{U}}, (\pi_i)_{u_i \in \mathcal{U}} \rangle$ 属于潜在博弈, 其潜在函数 $\Phi(s)$ 由式 (5-17) 表示。

QoE-DEER 算法分析

❖ 算法最优化分析

定理 2: 分布式 QoE-DEER 算法收敛于最大化系统 QoE 水平的纳什均衡解。

边缘资源分配(ERA)优化问题

$$\max_{x_i, a_i} \sum_{i \in \mathcal{U}} \pi_i(s_i)$$

优化目标: 最大化系统 QoE 水平

$$\sum_{i \in \mathcal{U}_j} a_i \cdot I_{\{x_i=j\}} \leq c_j \quad \forall j \in \mathcal{E}$$

边缘服务器资源容量约束

$$x_i \in \{0\} \cup \mathcal{E}_i, \quad a_i \geq 0 \quad \forall i \in \mathcal{U}$$

物联网用户服务请求约束

实验验证

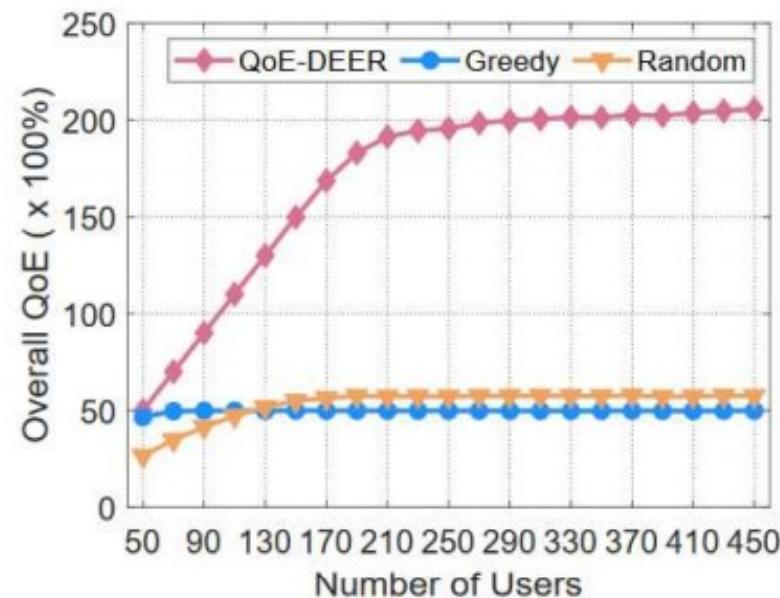
❖ 实验设置

■ 边缘计算 EUA 数据集 [122]

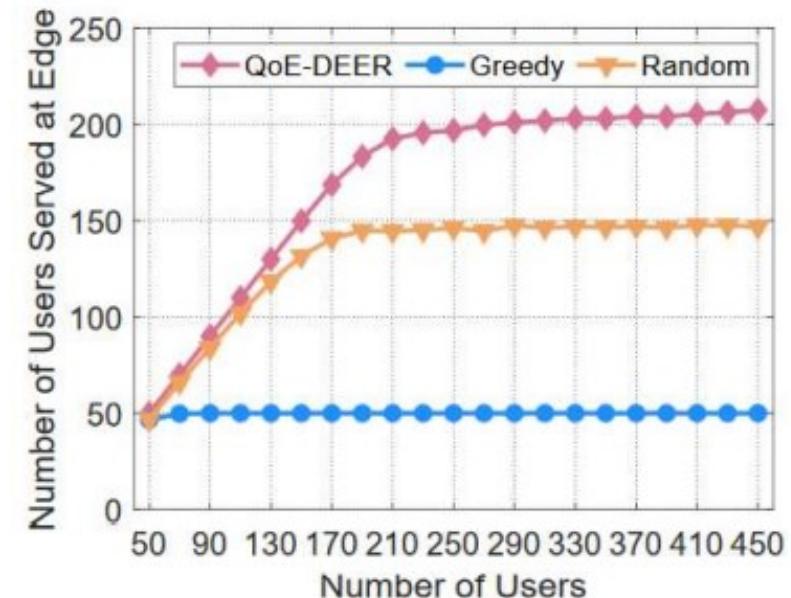
- ✓ 记录了澳大利亚墨尔本中央商务区内移动蜂窝基站、移动终端用户的经/纬度位置信息
- ✓ 将**移动蜂窝基站的位置信息**作为**边缘服务器**的**部署位置**
- ✓ 将**移动终端用户的位置信息**作为**物联网用户的所处位置**

实验验证

❖ 纳什均衡解的最优性



(a) 多用户的系统 QoE 水平

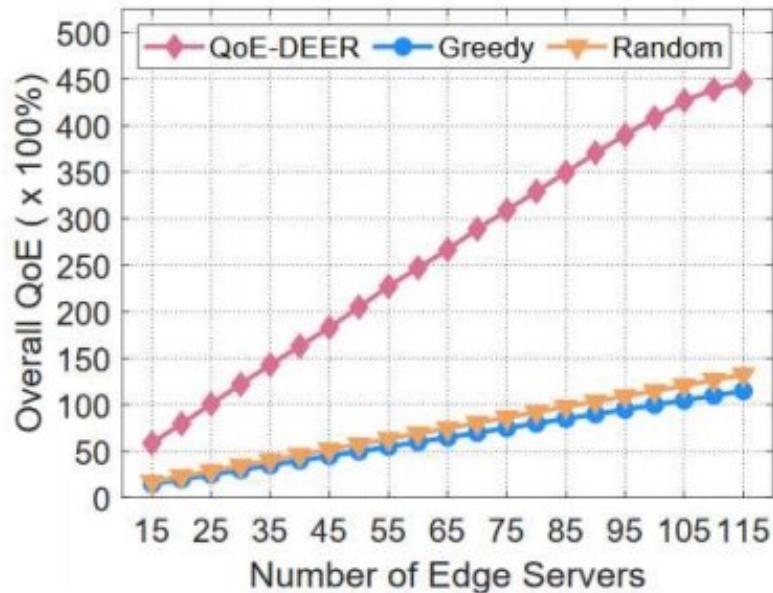


(b) 在边缘端被服务的物联网用户数

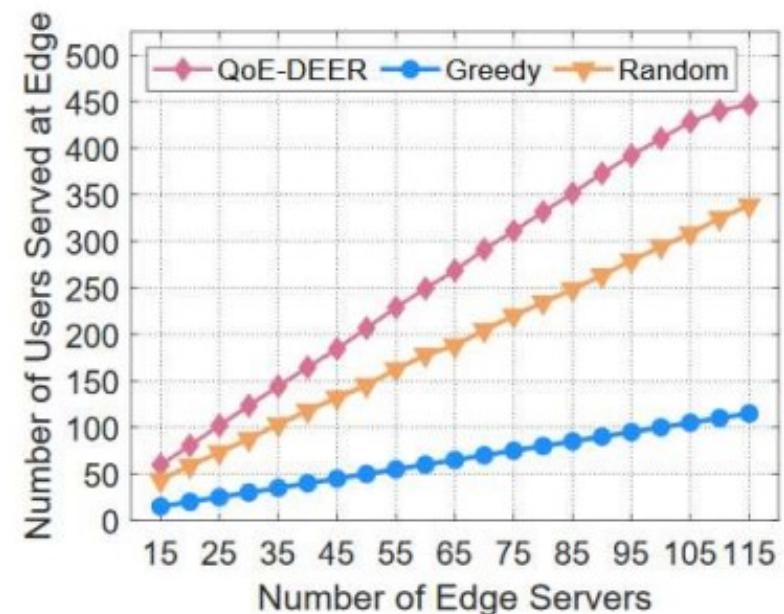
QoE-DEER 纳什均衡解的最优性 v.s. 物联网用户数量 N

实验验证

❖ 纳什均衡解的最优性



(a) 多用户的系统 QoE 水平

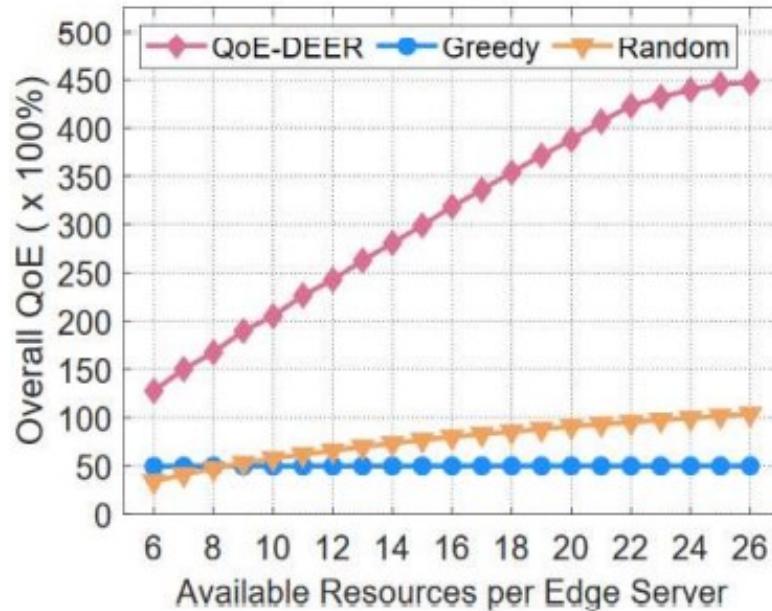


(b) 在边缘端被服务的物联网用户数

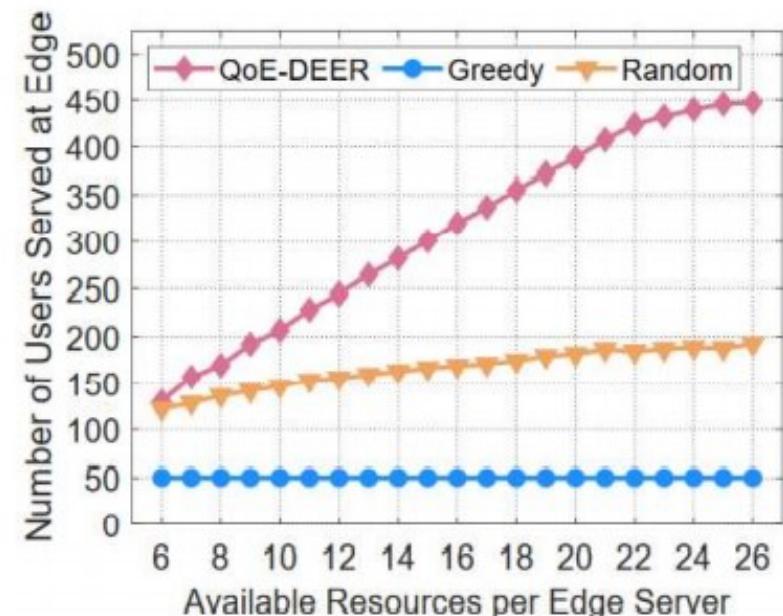
QoE-DEER 纳什均衡解的最优性 v.s. 边缘服务器数量 M

实验验证

❖ 纳什均衡解的最优性



(a) 多用户的系统 QoE 水平

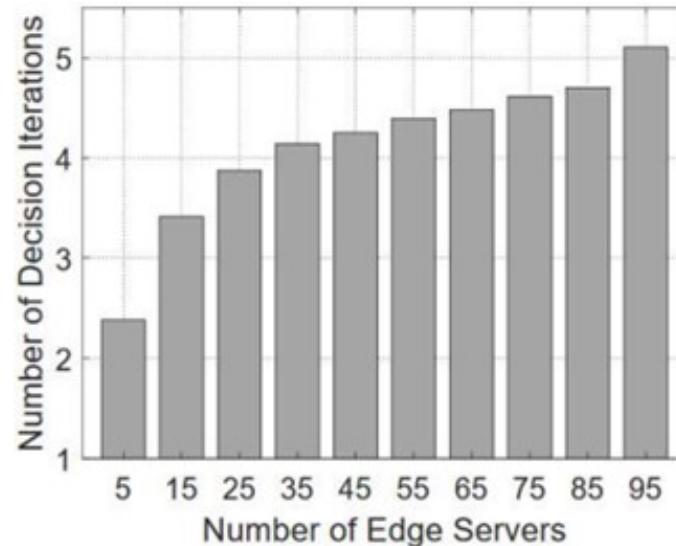


(b) 在边缘端被服务的物联网用户数

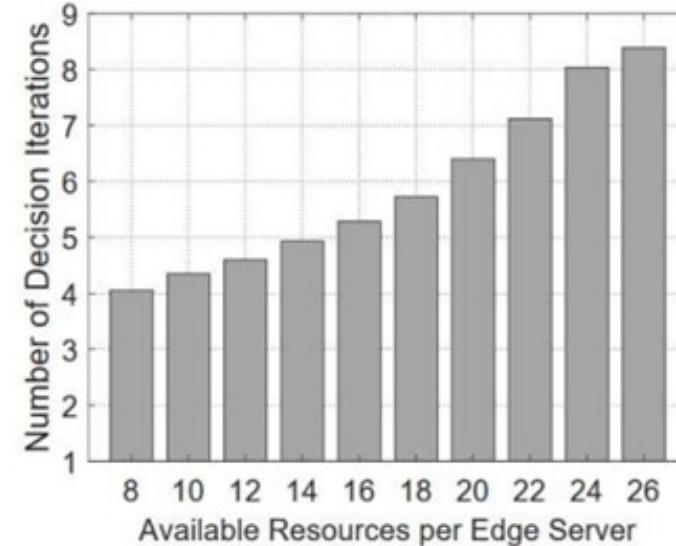
QoE-DEER 纳什均衡解的最优性 v.s. 边缘服务器资源容量 c_j

实验验证

❖ QoE-DEER 算法的收敛性



(a) QoE-DEER 算法收敛性 v.s. M



(b) QoE-DEER 算法收敛性 v.s. c_j

分布式 QoE-DEER 算法收敛性 v.s. 物联网用户数量 N

物联网用户数量 N	50	130	210	290	380	450
博弈回合次数	4.629	6.419	5.646	4.643	4.455	4.337

主要内容

- ❖ 引言
- ❖ 相关研究综述
- ❖ 基于公平性指标的多用户服务选择方法
- ❖ 面向市场的云资源定价与售卖机制
- ❖ QoE 感知的分布式边缘任务调度和资源管理
- ❖ **总结与展望**

论文总结

- ❖ 从服务生态系统的角度出发，设计了一种实现最大最小公平性的多用户服务选择算法 (**FASS**)，为用户提供了一种符合用户公平性的服务提供平台；
- ❖ 从服务供应商的角度出发，设计了市场化的用户服务激励机制 (**PIRA**)，旨在吸引大量用户来购买服务资源，帮助服务供应商占据较大的市场份额、赚取较高的服务收益；
- ❖ 从用户的角度出发，设计了 **QoE** 感知的去中心化边缘资源博弈算法 (**QoE-DEER**)，有效地管理了多用户之间的竞争博弈关系，促成了服务生态系统中的良性竞争态势。

研究展望

- ❖ 多服务供应商之间的服务公平性
 - 多家服务供应商拥有同等公平的竞争机会
 - 跨多家服务供应商的公平服务提供机制
- ❖ 基于次高价投标拍卖模型的服务价格激励机制
 - 实现更好的价格激励能力
 - 赚取更高的服务收益
- ❖ 动态博弈模型在服务生态系统中的应用
 - 感知服务供应商/用户的实时动态变化
 - Stackelberg 博弈、马尔可夫博弈等

上述研究工作依托的科研项目

- [1] 国家重点研发计划“面向服务的群智化生态化软件开发方法与环境”，课题4：“人-机-服务”协同学习的软件生态系统自适应机制研究（No. 2018YFB1003804）
- [2] 国家自然科学基金面上项目“基于边缘计算架构的物联网服务系统性能评价与优化”（No. 61972414）
- [3] 国家自然科学基金青年项目“面向大规模动态服务环境的 QoS 评价方法研究”（No. 61502043）
- [4] 北京市自然科学基金面上项目“农产品质量追溯的形式化建模理论和方法研究”（No. 4202066）

参考文献

- [1] Songyuan Li, Jiwei Huang, Bo Cheng. A Price-Incentive Resource Auction Mechanism Balancing the Interests Between Users and Cloud Service Provider. *IEEE Transactions on Network and Service Management (TNSM)*. (CCF推荐C类期刊; 中科院JCR小类Q1区, 近三年影响因子: 3.949; EI收录, 检索号: 20204709518052; SCI刊源, 已录用, DOI: 10.1109/TNSM.2020.3036989)
- [2] Songyuan Li, Jiwei Huang, Bo Cheng, Lizhen Cui, Yuliang Shi. FASS: A Fairness-Aware Approach for Concurrent Service Selection with Constraints. *The 2019 IEEE International Conference on Web Services (IEEE ICWS 2019)*, Milan, Italy, pp. 255-259, July 2019. (服务计算顶级会议, CCF推荐B类会议, EI收录, 检索号: 20194007491243)
- [3] Songyuan Li, Jiwei Huang, Bo Cheng. Resource Pricing and Demand Allocation for Revenue Maximization in IaaS Clouds: A Market-Oriented Approach. *IEEE Transactions on Network and Service Management (TNSM)*. (CCF推荐C类期刊; 中科院JCR小类Q1区, 近三年影响因子: 3.949; SCI、EI刊源, 已录用, DOI: 10.1109/TNSM.2021.3085519)
- [4] Songyuan Li, Jiwei Huang, Bo Cheng. QoE-DEER: A QoE-Aware Decentralized Resource Allocation Scheme for Edge Computing. *IEEE Transactions on Cognitive Communications and Networking (TCCN)*. (中科院JCRQ2区, 近三年影响因子: 4.574; SCI、EI刊源, 初审)
- [5] Jiwei Huang, Songyuan Li, Ying Chen. Revenue-Optimal Task Scheduling and Resource Management for IoT Batch Jobs in Mobile Edge Computing. *Peer-to-Peer Networking and Applications (PPNA)*, vol. 13, no. 5, pp. 1776–1787, September 2020. (CCF推荐C类期刊; 中科院JCRQ3区, 近三年影响因子: 2.235; EI收录, 检索号: 20201208313908; SCI收录, 检索号: 000562295700002)

(Thanks!)

敬请各位专家指正

Q&A

报告人：李松远 (Li, Songyuan)

2021 年 7 月 26 日