Auction-based location-aware task offloading strategy in edge cloud environment

Cui Pei
East China University of Science and Technology
Guisheng Fan (gsfan@ecust.edu.cn)
East China University of Science and Technology
Huiqun Yu
East China University of Science and Technology
Yiming Yue
East China University of Science and Technology

Research Article

Keywords: Edge cloud, Auction mechanism, Location awareness, Task offloading, Smart contract

Posted Date: April 20th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1549991/v1

Auction-based location-aware task offloading strategy in edge cloud environment

Cui Pei[1], GuishengFan[1,2*†], Huiqun Yu[1†] and Yiming Yue[1†]

[1*]Department of Computer Science and Engineering, East China University of Science and Technology, No. 130 Meilong Road, Shanghai, 200237, China.

[2]Shanghai Key Laboratory of Computer Software Evaluating and Testing, East China University of Science and Technology,Shanghai,China.

*Corresponding author(s). E-mail(s): gsfan@ecust.edu.cn; Contributing authors: 351591915@qq.com; yhq@ecust.edu.cn; 1533118717@qq.com;

†These authors contributed equally to this work.

Abstract

With the advent of 5G high-bandwidth and low-latency era, mobile data traffic has exploded. Most of the current researches on task offloading in edge cloud environments mainly focuses on minimizing latency, energy consumption and cost.But problems such as resource supply and demand relationship in the actual cloud market, participants' malicious competition behavior and task mobility will affect the actual offloading results, and the security of the offloading process cannot be guaranteed. Therefore, an Auction Based Location-aware Task Offloading (ABLTO) strategy is proposed, which uses the combined double auction model to meet users' needs for various types of resources. Then the buyer sorts according to its bid density, and the seller is ranked according to the authenticity of its bid and the distance from the buyer, and incentive/punishment measures are set to encourage the seller to deliver in good faith. The auction is deployed in the smart contract to ensure the authenticity and reliability of the process. Finally, the authenticity, fairness and economic benefits of the strategy are verified through simulation experiments.

Keywords: Edge cloud, Auction mechanism, Location awareness, Task offloading, Smart contract

1 Introduction

In recent years, with the continuous development of Internet of Things technology and information technology, the data collected by countless devices connected to the Internet is increasing exponentially, and many intelligent applications such as Virtual Reality (VR) and Augmented Reality (AR) are emerging that require intensive computing with high energy consumption. Edge cloud computing uses computing resources near the edge of devices to process device data, providing a computing environment with low latency, high bandwidth, and high stability. At present, a large number of research goals on task offloading in the edge cloud environment mainly focus on minimizing delay, energy consumption and cost, etc. There are still many shortcomings in this allocation method: (1) The amount of resources in the edge cloud environment may not exactly meet the needs of mobile devices, and competition is bound to occur, which may lead to malicious competition among users and seriously disrupt the cloud market order. (2) In the edge cloud environment, the user's demand for resources and the resources provided by the servers are changing dynamically, which affects the supply and demand relationship in the market. Unreasonable allocation is difficult to maintain the balance of supply and demand to maximize market benefits. (3) Edge cloud servers do not actively provide computing services, and may even deliberately provide low-quality resources to cause huge delays, and tasks cannot be executed on time. Appropriate incentives/punishments are necessary. Therefore, the introduction of the auction mechanism can effectively solve the above problems.

Auction has a wide range of applications in the field of economics, and auction theory has won the 2020 Nobel Prize in Economics. With the rapid development of the economy and society, the types of resources traded are becoming more and more abundant. The resource allocation in the edge cloud environment is very similar to the resource allocation in the economics market, so it is also a new trend to apply the auction theory to the resource allocation in the mobile edge cloud [1]. There has been some research work on auction-based task offloading in edge cloud environments, but there are still some problems: (1) The fairness of the offloading results obtained through the auction and the security of the offloading process cannot be guaranteed. (2) The diversity and mobility of user tasks bring challenges to the rational offloading tasks for auctions. At present, data leakage incidents are frequent. For example, in 2018, due to operational errors, Tencent Cloud's storage data in the cloud was lost; due to the vulnerability of Google's Google+, nearly 500,000 user data were leaked. Security issues have attracted great attention all over the world, and improper handling can even lead to irreversible results. Using the blockchain to store auction information in blocks ensures the security of the auction process. Blockchain 2.0 introduces smart contracts [2]. Smart contracts run on the distributed blockchain system and have the advantages of all other distributed systems. Data security and high availability are well guaranteed [3],[4]. Use smart contract technology to realize the auction process and reduce the management cost of auction [5],[6].

This paper proposes an auction-based location-aware task offloading (ABLTO) strategy for the edge cloud environment. The location of the mobile user in the edge cloud environment is not fixed, and the location also affects the choice of task offloading, which is important for reducing the service time of task offloading. Meaning, stability is also crucial for task offloading, and choosing a stable cloud server can reduce the failure rate of task offloading due to mobility. The main contributions of this paper are as follows:

(1) A combined double auction model is established to realize task offloading in edge environments. Using the combinatorial dual auction model not only improves the inefficiency of unilateral auctions, but also ensures that tasks require multiple types of resources.

(2) The ABLTO strategy is proposed, which sets priorities according to the authenticity of the seller's bid and the location distance from the buyer, and incentive/punishment measures are set to encourage the seller to deliver in good faith, so as to ensure the authenticity of the auction and effectively reduces the service time of task offloading.

(3) Deploying the auction in the smart contract ensures the security of the process and the authenticity of the unloading result, and uses the Ethereum platform as a simulated experimental environment to simulate the smart contract as the auctioneer's auction activity.

2 Related work

In recent years, there has been a lot of researches on task offloading in the edge cloud environment to reduce cost, energy consumption, time, etc. Chakroun et al. [7] proposed a stochastic network optimization based on Lyapunov optimization to optimize mobile cloud computing (MCC) resource allocation. Sarkar et al. [8] proposed a deadline-aware dynamic task placement method (DDTP) to classify deadline-sensitive tasks according to task utilization and queue allocation strategy. Mithun et al. [9] optimized the pricing model in edge computing to design prices for tasks with different deadlines, with the goal of maximizing revenue while completing tasks within their respective deadlines. Gao et al. [10] established a time and energy consumption evaluation model for computing task offloading of deep neural networks in a mobile edge computing environment. Wei et al. [11] proposed a task-separable energy-saving optimization problem for mobile devices in the scenario where multiple mobile users upload tasks to the server, and used the greedy selection method to solve the problem.

Most of the above research work ignores the dynamic changes in the supply and demand relationship in the resource market and the competition between participants in reality. As more and more scholars invest in research, auction theory is also used to solve computing offloading in the edge cloud environment. Zavodovski et al. [12] designed a DSIC dual auction mechanism with bidding, a method explicitly designed to match highly heterogeneous resources with different requests. Sun et al. [13] discussed two multi-item auction models: the

same multi-item auction and a different multi-item auction, and was also the first work on an auction mechanism that considers both privacy protection and social efficiency maximization. Kumber et al. [14] proposed a real combined dual auction mechanism for the allocation and pricing of computing resources in the cloud market to maximize market profits. Zhou et al. [15] utilized a three-stage auction for the deployment of micro-clouds in radio access networks. AP access points are sellers to users and buyers to edge servers. Reference [16] is an improvement to [15], which is also a three-stage auction, but Xia et al. [16] group mobile users in adjacent APs, and each AP can only apply for one edge server to reduce delay. Liu et al. [17] introduced a group buying mechanism in the case where the buyer moves users into groups, and there is a preferential price when the group buys, which is beneficial to improve the buyer's profit and resource utilization.

Malicious collusion between buyers and sellers in auctions and other dishonest behaviors can seriously undermine the rules and disrupt the market. A mobile blockchain-based auction guarantees a fair and secure process. Ma et al. [18] proposed a blockchain-based distributed feedback-based combined multi-unit double auction mechanism to motivate participants to provide a cloud resource market with high-quality services. Gao et al. [19] proposed an auction-based edge server resource allocation model in mobile blockchain, and used Benders decomposition to obtain a dynamic resource allocation strategy. ASIF et al. [20] introduced a multi-buyer and multi-seller dual auction mechanism in the resource allocation of the federated cloud. Liu et al [21] proposed a smart contract-based dual auction mechanism in mobile blockchain. Subtasks can be offloaded from one mobile device to heterogeneous edge servers. Desaih H et al. [22] proposed a novel hybrid blockchain architecture that bids on private blockchains and utilizes public blockchains to publish auction results. Liu et al. [23] proposed a blockchain-based MEC network architecture to handle large-scale video requests from different user devices in a decentralized and secure manner.

Using the characteristics of blockchain can effectively ensure the process security of resource allocation, but the use of blockchain technology is bound to be costly, which is also a common problem faced by the application of blockchain technology at present, but according to the above, it can be seen that data security is important for the current society, so for the more important data information, it is worth paying a certain cost for its security.

3 Auction-based location-aware task offloading strategy

3.1 System Model

The system is modeled to have similarity to the real-world environment, and the task offloading model is shown in Fig. 1. The system model consists of a central cloud (CC), an edge cloud (EC), and a mobile user (BU). User tasks are

defined by application type, which can be autonomous driving applications or infotainment applications. When the offloading process begins, consideration must be given to minimizing network latency to reduce the offloading process from start to finish, and secondly, the cloud must have sufficient resources to carry the offloaded task.

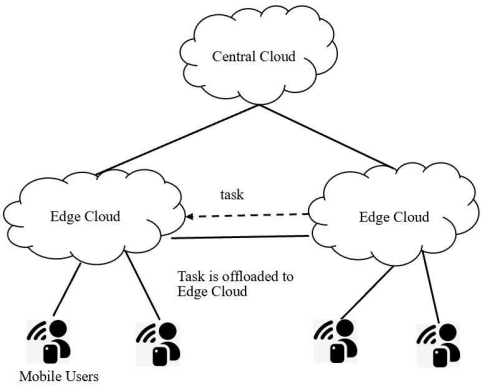


Fig. 1: Task offloading model

The sender (U) requesting task offloading and the receiver (P) of the edge cloud server move at random speeds and directions. Each mobile user contains multiple independent tasks. In short, U has several independent tasks, P performs computations by receiving tasks from U, and U can assign tasks to many P at the same time. The mobile user set is represented as $U = \{u_1, ..., u_n\}$, and for each user u_i, its task set is represented as $T_i = \{(t_1, ..., t_k)\}$, and t_j represents the jth task. The set of edge servers is denoted as $P = \{p_1, ..., p_m\}$. It is assumed that U and P can obtain their own position information through the GPS system, and Pi can receive the current position and direction vector information of U.

3.2 Moving Models

It is assumed that in a large-scale dynamic edge cloud environment, all mobile devices move in random directions. If the mobile device and the edge cloud server are oriented in similar directions, the transfer can take longer to exchange than if the directions are opposite, the edge cloud server in the same direction as the mobile device can perform more task offloading. Therefore, the edge cloud server that communicates for a long time can be found by calculating the time CT (contact time) that U and P maintain communication. U calculates the CT from the acquired current position and orientation vectors, and considers the mobility of the mobile device by choosing a longer CT. On the one hand, offloading tasks to a stable edge cloud server can reduce the number of execution failures during task offloading, and on the other hand, it can also reduce the response time. It is assumed that the moving directions of

U and P are initially fixed in randomly determined directions. The mobility model for mobile devices is shown in Fig. 2.

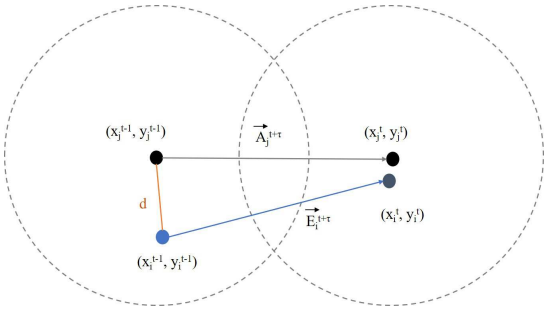


Fig. 2: Mobile model

The position of the device e_i is from $\left(x_i^{t-1}, y_i^{t-1}\right)$ to (x_i^t, y_i^t) ,and the task recipient edge server a_j is from $\left(x_j^{t-1}, y_j^{t-1}\right)$ to $\left(x_j^t, y_j^t\right)$, and the contact time is represented by τ, The direction vectors of e_i and a_j are expressed as follows:

$$z_i^t + \tau = \left(x_i d, y_i^d\right) = \left(x_i t - {x_i}^{t-1}, y_i^{t-} y_i^{t-1}\right) \tag{1}$$

$$z_j^t + \tau = \left(x_j^d, y_j^d\right) = \left(x_j^t - x_j^{t-1}, y_j^t - y_j^{t-1}\right) \tag{2}$$

Then the distance between the mobile device e_i and the edge cloud server a_j is:

$$d = (z_i + \tau)^2 + (z_j + \tau)^2 \tag{3}$$

The contact time τ of the communication between devices e_i and a_j is calculated as follows:

$$\tau \leq \frac{-(\alpha\mu + \beta\omega) \pm \sqrt{\left(\alpha^2 + \beta^2\right) R^2 - (\alpha\omega - \mu\beta)^2}}{\left(\alpha^2 + \beta^2\right)} \tag{4}$$

$$\alpha = x_i^d - x_j^d, \mu = x_i^t - x_j^t \tag{5}$$

$$\beta = y_i^d - y_j^d, \omega = y_i^t - y_j^t \tag{6}$$

3.3 Auction Model

This paper proposes a combinatorial-based dual auction model for task offloading in edge cloud environments. In this model, users give a combination of resource requirements based on the execution needs of multiple tasks, and provide a unified bid for the entire bundle of required resources. A User can obtain all resources from one supplier or from different suppliers. If all the resources the user needs are available and the supplier's price range is within the bundle bid submitted by the user, then the supplier can offer a complete resource bundle. Finally, the auction is deployed on the smart contract to ensure the authenticity of the uninstallation result. The cloud market consists of three

entities: users, suppliers, and auctioneers, where suppliers include central and edge clouds, as shown in Fig. 3.

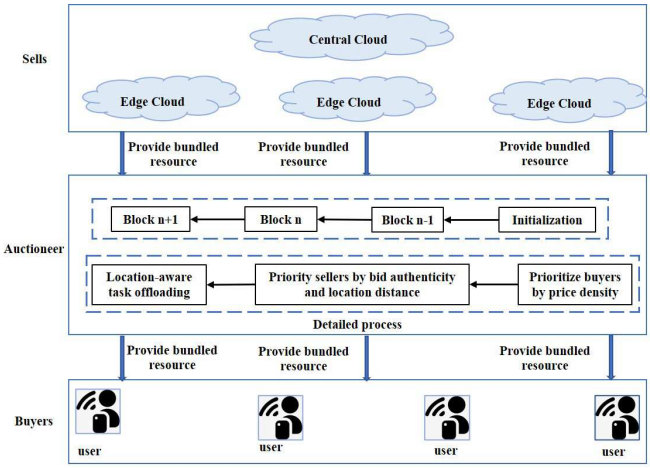


Fig. 3: System frame

This auction model considers m resource providers as sellers $P=\{p_1, ..., p_m\}$ and n mobile users as buyers $U=\{u_1, ..., u_n\}$. The virtual machine (VM) resource set provided by each resource provider i $(i = 1, ..., m)$ is represented as $Rp=\{VM_1, .., VM_A\}$, here it is assumed that the VM attributes provided by different resource providers are the same (including CPU, memory, storage and bandwidth), the amount of resources provided by each VM_l is S_{il}, the amount of bundled resources provided by p_i is expressed as $r_i^p=\{S_{i1}, , S_{iA}\}$, R_i^p means that p_i provides the total amount of resources, bid_i^p is the expected price of the bundled resources provided by p_i, bid_{il}^p is the expected price of the VM_l in the bundled resources by p_i, and the current position coordinate of p_i is (x_i^t, y_i^t).

The resource set required by each user j $(j = 1, ..., n)$ is expressed as $Ru=\{VM_1, .., VM_B\}$, and the resource amount of VM_k required by u_j to execute the task is S_{jk}. $u_j s$ demand for resources is expressed as $r_j^u=\{S_{j1}, , S_{jB}\}$, R_j^u represents the total resource demand of user u_j, bid_j^u is the highest price u_j is willing to pay to apply for bundled resources, bid_{jk}^u is the highest price that u_j is willing to pay for VM_k in the bundled resources applied for, and the current position coordinates of u_j are $\left(x_j^t, y_j^t\right)$.

The weight of each VM is equal to the capacity of the VM resource multiplied by the sum of the unit price of the resource attributes (CPU, memory, storage, and bandwidth) it contains. The weight is inherently dynamic because the resource unit price changes according to market demand. The VM weight of p_i is $w_i^p = (w_{i1}^p, \ldots, w_{iA}^p)$, and the VM weight of u_j is $w_j^u = (w_{j1}^u, \ldots, w_{jB}^u)$. The unit price of bundled resources is the price charged for bundled resources compared to the total amount of resources. $unitP_i^p$ and $unitP_j^u$ represent the unit price of the bundled resources of resource provider p_i and user u_j,

respectively.

$$\text{unit}_i^p = \frac{\text{bid}_i^p}{R_i^p} \tag{7}$$

$$\text{unitP}_j^u = \frac{\text{bid}_j^u}{R^u} \tag{8}$$

Because the configurations of VMs are not the same, and the amount of resources of different types of VMs is different for each resource provider, the unit price of each type of VM needs to be calculated for fairness. Then calculate the unit price of VM_1 and VM_k in bundled resources by users/resource providers as follows, first calculate the price weights of different types of VMs, and then calculate the unit price of the total bundled resources.

$$\text{bid}_{il}^p = unitP_i^p * W_{il}^p \tag{9}$$

$$\text{bid}_{jk}^u = unitP_j^u * W_{jk}^u \tag{10}$$

Then the transaction unit price is based on the average unit price of the VM provided by the resource provider and the VM requested by the user:

$$dp_{ij}^{kl} = \left(bid_{il}^p + bid_{jk}^u\right)/2 \tag{11}$$

The purpose of the auction is to benefit both users and suppliers. The goal of the task offloading problem is to maximize the profit of the participants. The calculation method of the profit of the winning resource provider p_i is as follows. x_{il} represents the amount of resources actually allocated by VM_l of p_i.

$$U_i^p = \sum_{l=1}^{A} \left(dp_{ij}^{kl} - bid_{il}^p\right) * x_{il} \tag{12}$$

The profit of the winning user u_j is calculated as follows:

$$U_j^u = \sum_{k=1}^{B} \left(bid_{jk}^u - dp_{ij}^{kl}\right) * s_{jk} \tag{13}$$

In addition, the task transmission time and calculation time are also considered here. The response time d_{ij} is calculated as the sum of the transmission time and the calculation time. Considering the mobility of mobile users, the contact time τ_{ij} of the task should be greater than or equal to the response time, so as to ensure the smooth completion of the task Uninstall works.

Define two decision variables win_i and win_j, if $win_i = win_j = 1$, it means that the resource provider p_i and user u_j win the auction, otherwise it means that they fail. Denote the amount of resources allocated to u_j by VMs of type

p_i k as x_{ijk}, then the problem is transformed into obtaining the maximum profit of the buyers and sellers participating in the auction as:

$$\max\left(\sum_{j=1}^{n} \mathrm{win}_j\, U_j^u + \sum_{i=1}^{m} \mathrm{win}_i\, U_i^p\right) \tag{14}$$

subject to:

$$x_{il} \leq S_{il}. \tag{15}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{A} bid_{il}^p \cdot x_{ijk} \leq win_j \cdot bid_j^u. \tag{16}$$

$$\sum_{i=1}^{m} x_{ijk} = win_j \cdot s_{jk}. \tag{17}$$

$$d_{ij} \leq \tau_{ij}. \tag{18}$$

The objective function is shown in Formula 14. Then define the constraints of the problem: Formula 15 shows that the amount of resources actually allocated by VM_1 of p_i is not greater than the total amount of resources actually owned by VM_1. Formula 16 shows that the highest price that the winner is willing to pay for the bundled resources applied for is not less than the sum of the evaluations of the resources provided by the resource providers. Formula 17 for all j and k, checking whether resource requirements of the user j for VM_k are satisfied, if satisfied, the user wins, $win_j = 1$. Formula 18 ensures that the response time of the task is less than or equal to its contact time.

A. Determine the priority of users

Prioritizing users can provide allocation efficiency and increase auction profits. So price density is introduced [24]. The price density per unit time depends on the user's bid and the amount of bundled resources required. Specifically, as shown in Algorithm 1.

Algorithm 1 Buyer User Priority

Input: bid_j^u, s_{jk}, w_j^u
Output: the sorted list of users: L_u

```
Initialization L_u=[]
for each u_j ∈ U do
    for VM_k ∈ R_u do
        W_jk^u = capacity (VM_k) /(α + β + γ + η)
        res_total = s_jk * W_jk^u
    end for
    d_j^u = b_i d_j^u / res_total
    if bid_j^u > bid_{i-1}^u then
        Update the position of bid_j^u and bid_{j-1}^u in list L_u
    end if
end for
```

$$d_j^u = \frac{bid_j^u}{\sqrt{\sum_{k=1}^{k=B} s_{jk} * W_{jk}^u}}. \tag{19}$$

B. Prioritize resource providers

For the method of determining the priority of resource providers, refer to the above method. Through price density, providers may maliciously low prices to attract customers to participate, or they may drive up prices for malicious competition. Therefore, it is difficult to guarantee the real bidding of resource providers by simply sorting by price density. In addition, considering the location awareness of tasks, the user's location may change in each round of auctions, and users prefer to offload tasks to edge cloud servers that are closer to themselves for processing, because the selection distance will inevitably lead to higher network latency, according to subsection 3.2 by choosing a longer CT to consider the mobility of mobile devices, offloading tasks to stable edge cloud servers can also reduce response time because there is no task offloading due to mobility Failure is also conducive to improving the success rate of task offloading.

Therefore, the priority of each resource provider is different for different buyers. Considering the authenticity of the resource provider's bid and the location distance from the user, the priority is set, which can not only maintain the interests of the honest resource provider, but also reduce task offloading. The delays and failures also ensure the fairness and justice of the auction market. Categorize resource providers by bid:

(1) The bid is within the real bid range

(2) The bid exceeds the true bid range

(3) The bid is lower than the true bid range

The real bidding range is based on the weight of the VM service provided by itself and the average of the bids of all resource providers. Typeresource providers, that is, the real suppliers, are ranked first, ensuring that the real suppliers have the highest priority to provide resources. Then sort them in descending order of contact time with the current user. Typeandresource providers are sorted in descending order according to their proximity to the real bidding range, giving priority to the authenticity of the suppliers. If the proximity is the same, they are compared according to their locations to ensure that the higher the degree of authenticity, the contact with users The longer the resource provider, the better the chance to participate in the auction. Specifically, as shown in Algorithm 2.

C. Location-aware task offloading strategy

After determining the priorities of users and resource providers, an algorithm is proposed to maximize the profits of users and providers to achieve optimal resource allocation. The algorithm inputs a list of users, a list of resource requests, a list of resource providers, and a list of resource offerings that have been sorted by price density. Output a resource matrix and a price matrix, showing the resource allocation between users and resource providers

Algorithm 2 Seller Resource Provider Priority

Input: $bid_i^p, (x_j^t, y_j^t), (x_i^t, y_i^t), truebidscope[range_b, range_e]$
Output: the sorted list of seller: $L_p = [L_p^1, ..., L_p^n]$

```
Initialization L_u = [], a = [], b = [], c = [], real_bid = 0, total_bid = 0
for each u_j ∈ U do
    for p_i ∈ p do
        Calculate the position distance d_ij and contact time τ_ij between u_j and p_i
        if range_b ≤ bid_i^p ≤ range_e then
            if a=null then
                a.add(p_i)
            else
                Sort in descending order of the size of τ_ij
            end if
        end if
        if bid_i^p < range_b then
            ts_i = range_b − bid_i^p
            c.add(ts_i)
            Sort in ascending order of the size of ts_i
        else
            ts_i = bid_i^p − range_e
            b.add (ts_i)
            Sort in ascending order of the size of ts_i
        end if
    end for
    for p_i ∈ a do
        L_u^j.add(p_i)
    end for
    Combine list b and list c into list d in ascending order according to the size of ts_i
    if ts_i = ts_{i+1} then
        Compare τ_i in descending order
    end if
    for p_i ∈ d do
        L_u^j.add(d_i)
    end for
end for
```

and the price paid by users to resource providers, respectively. Specifically, as shown in Algorithm 3.

D. Rewards/penalties for winning sellers

If the quality of the resources provided by the winning seller is equal to or greater than the quality of the resources required by the buyer, then the buyer's task can be guaranteed to be executed smoothly. However, if it is lower than the buyer's requirement, it indicates that the seller has cheated. In order

Algorithm 3 Location-aware task offloading algorithm

Input: $L_p L_u r_j^u r_i^p, bid_j^u, bid_i^p$
Output: res[][], bid[][],profit

```
for each u_j ∈ U do
    for each p_i ∈ L_p^j do
        if bid_j^u > bid_i^p then
            determine whether to meet the current user's resource needs
            update r_j^u, r_i^p
        end if
    end for
    if u_j is winner then
        Update res[][],bid[][]
        Calculate the total profit of both parties
    end if
end for
```

to avoid the seller's deceptive behavior and encourage honest transactions, by comparing the seller's initial resource quality score with the final resource quality score after delivery, if the final score is not lower than the initial score, it means that the seller is honest, otherwise, the seller is deceived in the transaction, and the seller will be punished. The attributes that affect the quality of the seller's resources are expressed as $Q_vec = \{CPU, Memory, Storage, BW\}$, and its weight is expressed as $Q_w = \{w1, w2, w3, w4\}$, then the seller's score is calculated as follows:

$$\text{point}_i^p = \sum_{h=1}^{\text{Q}\_vec\_len} Q\_\text{vec}\ (h) \times Q\_w(h) \tag{20}$$

The actual profit of the seller in the end is as follows:

$$g_i^p = \frac{point\_final_i^p}{point\_initial_i^p} \times u_i^p \tag{21}$$

If the score calculated after the transaction is equal to 1, then the seller will get the original profit. If the score calculated after the transaction is lower than the initial score, indicating that there is deceit, then part of the actual profit obtained by the seller will be deducted as a penalty. If the score is higher than the initial score, the seller will be rewarded accordingly. Determine rewards/penalties by comparing scores, incentivizing sellers to provide high-quality resources and participate in auctions with integrity.

4 Experimental Design

4.1 Experimental environment

The experimental platform is set up as follows:

Hardware configuration: Intel$Core^{TM}$ i5-6200U CPU 2.10GHz, 16GB RAM

Development language: Matlab, Java

Experimental tools: MATLAB2018, IntelliJ IDEA 2019.3, CLOUDORADO, EdgeCloudSim,Remix, Ganache, MetaMask.

4.2 Simulation data

The prices of different virtual machines are collected from CLOUDORADO [25], a website that compares resource allocation and prices from cloud service providers (Google, Microsoft, and Amazon). Observe the pricing patterns for various types of VMs offered by all cloud service providers, and generate VM prices for the experiments based on this data. The user's mobile location coordinates are simulated from EdgeCloudSim [26], the user's mobile range is set to a radius of 500m, and the user's communication range is set to 100m [27]. The specific experimental parameter settings are shown in Table 1.

Table 1: Experimental simulation parameter setting

parameter	meaning
Type of VMs	4(VM1,VM2,VM3,VM4))
Users requested quantity	VM1[1,5] VM2[1,5] VM3[1,5] VM4[1,5]
Users bid valuation	[50,100]
Providers offered quantity	VM1[0,10] VM2[0,10] VM3[0,10] VM4[0,10]
Providers ask price	VM1[1,2] VM2[2,4] VM3[4,6] VM4[6,8]
Number of iteration	100
Number of buyers	[10,120]
Number of sellers	[10,100]

4.3 Evaluation indicators

Any market-based resource allocation combination dual auction model is effective when users and suppliers support each other. The evaluation indicators for ABLTO are as follows:

Fairness: The proposed method maintains fairness by providing equal opportunities to every participant in the auction.

Authenticity: The authenticity of the system is maintained in the above auctions by setting a higher priority to suppliers who provide real bids. When a supplier wins the auction, the supplier who bids the real price will also get more profits, which ensures the authenticity of the system.

Economic efficiency: The proposed model maximizes the profits of users and providers, and provides opportunities for more auction suppliers, increasing

the number of auction resources, thereby improving the economic efficiency of the model.

Personal rationality: To ensure that the utility of all participants is always non-negative, the proposed model constrains that the highest price that the winning user is willing to pay for the bundled resource is not less than the sum of the resource providers' valuations for the resource, ensuring that the participant's profit is non-negative.

Balanced budget: If the total payment of all buyers is equal to the total payment to the seller, the budget is balanced.

In combinatorial double auctions, where buyers bid in a bundled form, economic efficiency and computational efficiency conflict with each other[28], and authenticity and economic efficiency in combinatorial double auctions conflict with each other[29]. Therefore, the auction model mentioned in this paper ensures the authenticity of the auction as much as possible while ensuring economic efficiency.

4.4 Experimental scheme

According to the above introduction to the experimental environment, data simulation and evaluation indicators, the experimental analysis of the ABLTO strategy is carried out to verify the correctness and feasibility of the strategy.

(1) Verify that the model satisfies the basic properties of auctions: individual rationality, authenticity, and economic efficiency. Set the number of buyers and sellers participating in the auction respectively, and observe the profit changes of users and resource providers.

(2) Comparing the proposed ABLTO strategy with three other allocation mechanisms: CDARA[30], DS-VRAP[31] and OPTIMAL. Compare the total profit and total cost of its auction participants.

The CDARA resource allocation model adopts an average pricing mechanism to match a user's request for a specific type of VM with all types of VMs available to resource providers by comparing their respective resource attribute values, and does not consider resource providers that are not true bidders. The ABLTO proposed in this paper abstracts resource attribute values, and calculates a weight value for different types of VMs through one or more resource attribute values. According to dynamic changes in market conditions, users request a specific type of VM to match the same type of VM provided by the resource provider. Resource providers with non-real bids participate in the auction according to their proximity to the real bid range, increasing the participation rate of users and resource providers.

DS-VRAP proposes a combined greedy allocation scheme, which sorts the auction participants according to the residual density value (the amount of residual resources caused by the participants trading resources), and allocates the surplus generated by the market to all the winners according to the contribution ratio. Seller, which does not take into account the authenticity of the seller. In this model, buyer users can only obtain resources from one resource provider, and one resource provider can allocate resources to multiple users.

The ABLTO proposed in this paper allows the buyer to obtain the resources of different sellers and at the same time guarantees the authenticity of both buyers and sellers.

OPTIMAL is an optimal allocation mechanism based on Integer Linear Programming (ILP) to solve the market profit maximization problem in an optimal way [27]. Both the user and the resource provider use the first price, and the user wins and pays his offer. The winning amount of the provider is the price multiplied by the amount of resources provided. This mechanism optimizes the allocation of resources, but cannot guarantee the authenticity of buyers and sellers.

4.5 Experimental results and analysis

4.5.1 Verification of evaluation indicators

Fig. 4 shows the bid, bid density, and profit information for 20 sellers in a market with 10 sellers.

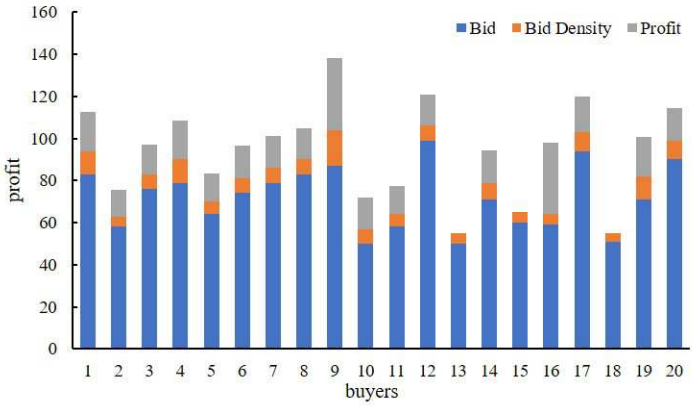


Fig. 4: Relationship between buyer's bid density and profit

It can be seen from the Fig. 4 that the buyer's profit is sorted in descending order according to their bidding density. For example, user 9 with higher bidding density obtains more profit, and users 13, 15, and 18 have lower priority due to lower bidding density, resulting in failure of the auction and unable to obtain resources to perform the task, that is, the profit is 0. So a user's profit depends on their bidding density. If the user is dishonest, it may happen that the bundled bid of user (1) is lower than its real expected value, its bidding density will be reduced, the priority will be reduced, and the probability of winning will also be reduced, and it may not be possible to win and gain profits. (2) User's bundle bid is higher than its true valuation, and its bid density increases. Although the probability of winning increases, the transaction price may increase accordingly, resulting in a decrease in the user's profit, so the user will be more willing to make a bid equal to its true valuation. In addition,

it can be seen from the figure that the profits of all buyers are non-negative, which proves that ABLTO guarantees the fairness, authenticity and personal rationality of the auction sellers.

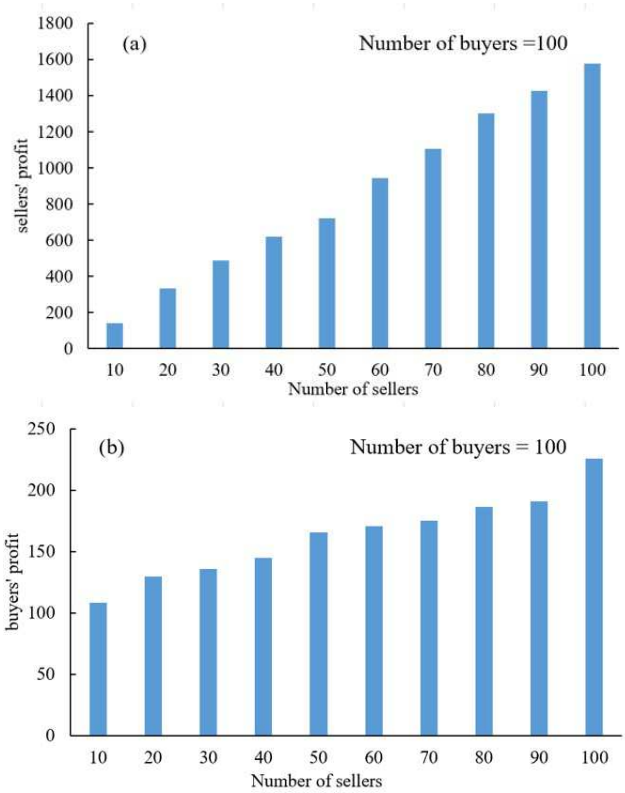


Fig. 5: The effect of quantity of seller on profit of buyer and seller (a) Seller's profit (b) Buyer's profit

Fig. 5 shows the effect of the number of sellers on the profit of the participants in the ABLTO strategy. The number of buyers is set to 100, and the number of sellers increases from 10 to 100. It can be seen from the Fig.5 that the profits of both buyers and sellers increase with the increase of the number of sellers. Because as the number of resource providers increases, there will be more resources in the market, and the resource needs of more users will be met, so there will be more winning users and resource providers, so the profits of buyers and sellers will also increase.

4.5.2 Algorithm comparative analysis

Fig. 6 and Fig. 7 show the total seller cost and buyer total payment for different numbers of resource providers, respectively. It is also set that there are 100 buyers participating in the auction, and the number of sellers increases from 10 to 100 in turn. Observe the specific information of the buyers and sellers.

As shown in Fig. 6, the total seller cost of CDARA and OPTIMAL first increases and then decreases as the number of sellers increases. Initially, resources are in short supply. As the number of sellers increases, more resources are provided to participate in the transaction, so the total cost increases sequentially. When the resources in the market have met the needs of all users, CDARA and OPTIMAL will give preference to sellers with lower bids, resulting in a reduction in the total cost. In DS-VRAP, since the user's request cannot be split, there will be more winning sellers when the number of sellers increases, and the total cost of sellers will increase accordingly. This paper proposes ABLTO, the priority of sellers is to sort according to the authenticity of their bids and location information instead of blindly choosing the lowest. Therefore, the total cost of the seller increases with the increase of its number, and tends to be stable when the user's resource requirements are all satisfied, and the total cost of ABLTO will be slightly higher than the other three mechanisms.

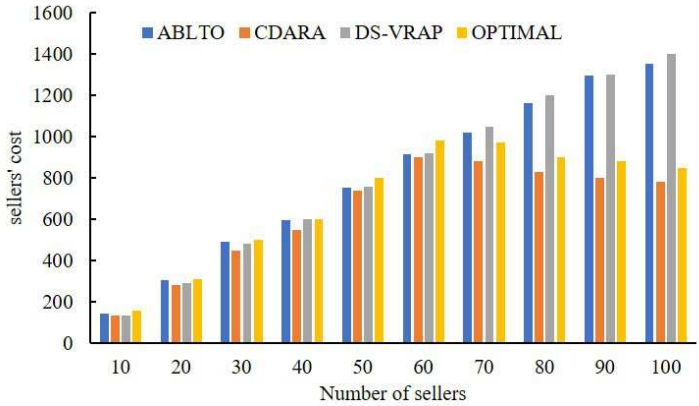


Fig. 6: The change in seller's cost as the number of sellers increases

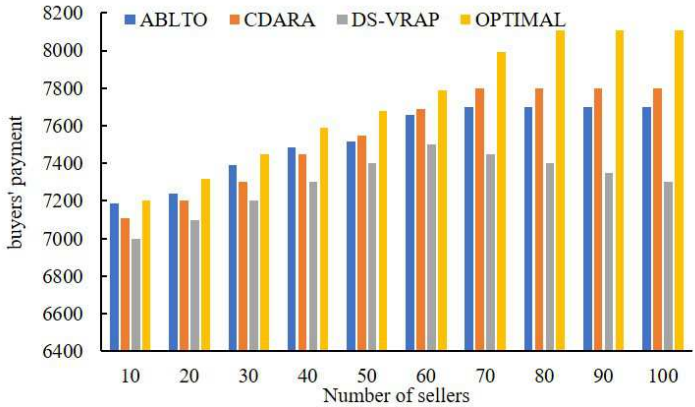


Fig. 7: The change in buyer's payment as the number of sellers increases

As shown in Fig. 7, OPTIMAL has the highest payment due to choosing the first price, ABLTO, CDARA and DS-VRAP satisfy the authenticity, and the user's actual payment is lower than its real valuation. When the supply of resources in the market is lower than the demand for resources, according to the above-mentioned ABLTO's introduction to the selection of resource providers, the one with the lowest bid is not preferentially selected, so the highest payment will be generated. As the number of sellers increases, more buyers are satisfied, and more buyers win, so the total payment increases. When all user resource requirements in the market are met, the buyer's total payment does not change much with the increase in the number of sellers until it no longer increases.

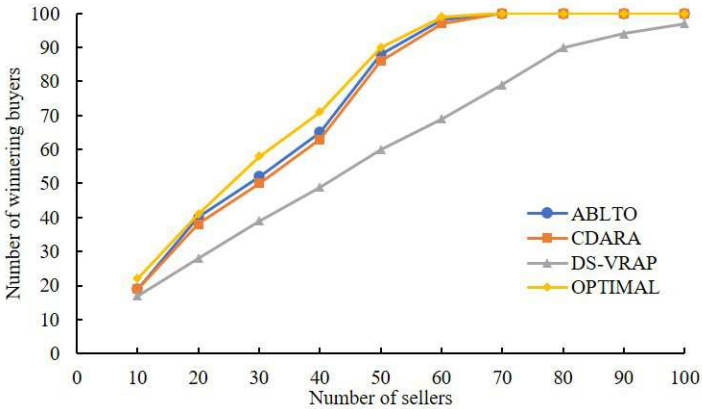


Fig. 8: The number of buyer winners with increasing number of sellers

Fig. 8 shows 100 buyers participating in the auction, with the number of sellers increasing from 10 to the number of winning buyers in 100. DS-VRAP, the user request described cannot be split, and a seller must satisfy all its needs before it can be traded. Therefore, it has the least number of winning buyers compared to the other three mechanisms. ABLTO, CDARA and OPTIMAL have similar number of winning buyers in each auction, and OPTIMAL is the best. Since CDARA completely rejects non-real bid sellers and reduces the participation rate of resource providers, the number of winning buyers is slightly lower than that of ABLTO.

Fig. 9 shows user satisfaction, calculated as the ratio of the winning user's total valuation to the total valuation of all users. As can be seen from Fig. 9, OPTIMAL solves the social welfare maximization problem in an optimal way, which yields the highest profit among all mechanisms. Buyer satisfaction of ABLTO and CDARA is similar and better than DS-VRAP. DS-VRAP users can only acquire resources from a single resource provider, so some possible allocations may be rejected. User satisfaction for CDARA, OPTIMAL and ABLTO does not change when resource providers are increased from 60 to 100. Because when the number of sellers is 60, the demand for the resource equals the supply, resulting in no further allocation. DS-VRAP resources are

not fully utilized due to the indivisible constraint of requests, and when the number of suppliers in the market increases, more allocations are generated.

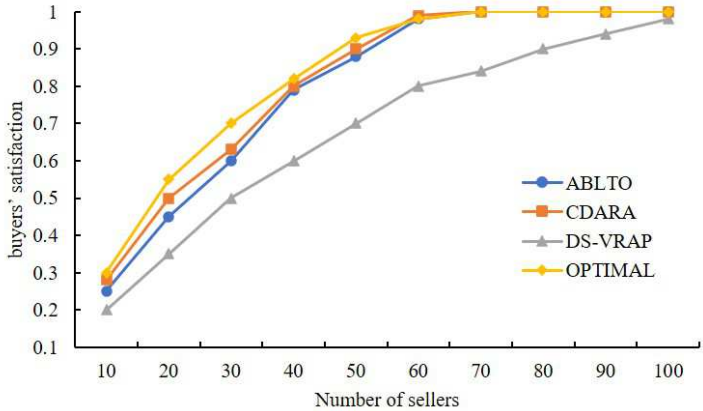


Fig. 9: The change of buyer satisfaction with increasing number of sellers

Fig. 10 shows the task response time. As the number of sellers increases, more buyers' needs are met, so the response time increases. Compared with ABLTO, the other three algorithms do not consider user mobility. If the user is out of communication range due to mobility, then task offloading fails, resulting in more response time. ABLTO considers the distance between the user and the edge server. The short distance will inevitably reduce the network delay. In addition, the long contact time is preferred to ensure the successful execution of task offloading and minimize the response time. From the Fig. 10, it is concluded that OPTIMA has the longest response time because the optimal allocation has the most winners and the most possible unloading failures.

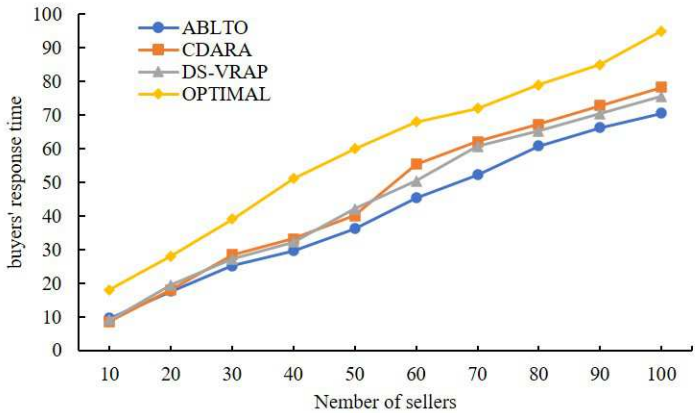


Fig. 10: Response time as the number of sellers increases

Fig. 11 shows the total profit for several distribution mechanisms. As the number of sellers increases, the amount of resources involved in transactions in the market increases, more and more user resource demands are met, and ultimately the total profit increases. The OPTIMAL mechanism solves the profit maximization problem in an optimal way, using the first price so it has the highest market profit compared to all mechanisms. DS-VRAP, since the user's request cannot be split, it is difficult for a seller to satisfy the buyer's demand in terms of price, VM specific attribute value and resource amount at the same time, so the total profit is the lowest. Because CDARA directly rejects resource providers with non-real bids, it reduces the participation rate of providers, and also affects users' inability to obtain the required resources to win the auction, so its total profit is lower than that of ABLTO. Because ABLTO considers all resource providers and prioritizes them according to the authenticity and location of bids, it increases the participation rate of users and providers in the market, and is the closest allocation mechanism to OPTIMAL.

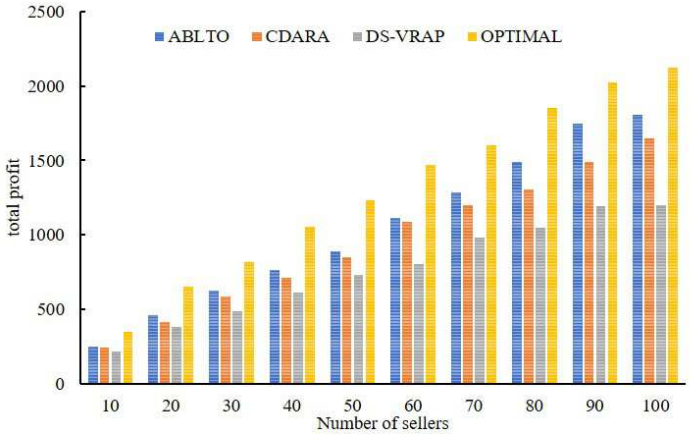


Fig. 11: The change in total profit as the number of sellers increases

4.5.3 Deploy the Ethereum platform

Use Remix for contract development, use Ganache client to initialize 10 accounts, each account has 100 bitcoins, any account is added to MetaMask through private key, it can customize RPC network http://127.0.0.1: 7545 is consistent with Ganache's RPC Severe, otherwise it will not be possible to connect to the Ganache private chain.

After configuring Ganache and MateMask, compile and execute Remix's Auction contract. Here, you need to select the injected web3 object in the execution environment, and you can use the logged-in account on MateMask to conduct transactions. The deployment is successful, and the result is shown in Fig. 12. Regardless of whether the auction is successful or not, as long as the contract

is successfully executed, the account will consume a certain amount of gas. Check the balance information of Ganache's corresponding account as shown in Fig. 13. The transaction information is packaged into blocks and the chain is completed.

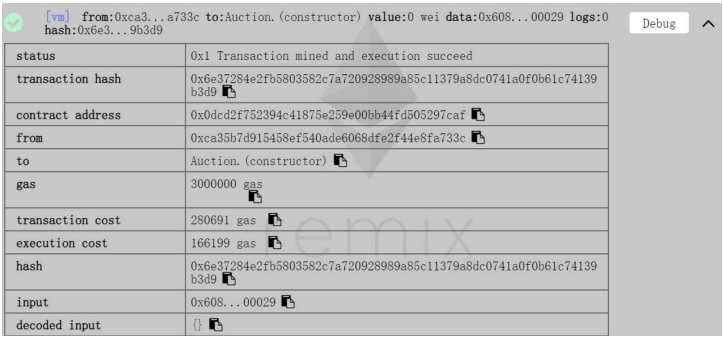


Fig. 12: Deploy smart contracts

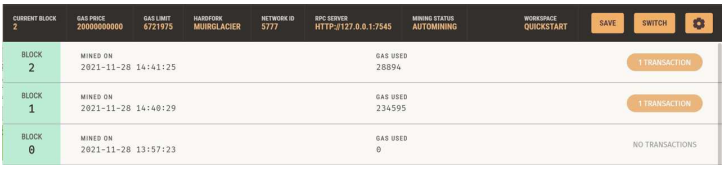


Fig. 13: Auction results are made into blocks and put on the chain

5 Conclusion

This paper considers the influence of the location and mobility of mobile users on the auction in the auction process, and considers the user's demand for resource categories, using the combined double auction model to improve the inefficiency of unilateral auctions. An auction-based location-aware task offloading strategy is proposed. Buyers and sellers bundle and bid on resources, set seller priorities based on bid density, divide sellers according to the real bidding range, and then calculate contact time settings for different users according to the distance of sellers. The seller's priority allows users to choose the resource provider with the highest bid authenticity and the closest to themselves, which not only ensures the authenticity of the auction, but also effectively reduces the service time of task offloading. ABLTO finally transforms the problem into maximizing the profits of auction participants, and verifies the authenticity, fairness, economic benefits and personal rationality of the model through comparative experiments.

The ABDTO strategy studied in this paper is to use the smart contract as an auctioneer to ensure the fairness and impartiality of the auction. However, there are still loopholes in the smart contract, which may lead to the failure of the auction. In practical applications, it may be maliciously attacked and

the auction information may be leaked. Therefore, future work can consider solving the security vulnerability of smart contracts, so that smart contracts can more safely serve the task offloading problem.

Statements and Declarations

Ethical Approval and Consent to participate. Not applicable

Human and Animal Ethics. Not applicable

Consent for publication. Not applicable

Availability of supporting data. All data generated or analyzed during this study are included in this published paper.

Competing interests. The authors have no relevant financial or non-financial interests to disclose. No conflict of interest exists in the submission of this manuscript. We declare that we dont have competing interests.

Funding. No funding was received for conducting this study.

Authors' contributions. CP was the experimental designer of this study and the executor of the experimental study and wrote the first draft. YY participated in experimental design and analysis of experimental results. GF and HY guide the experimental design, thesis writing and revision. Final manuscript read and approved by all authors.

Acknowledgments. At the point of finishing this manuscript, the authors would like to express sincere thanks to all those who have lent us hands in the course of our writing this manuscript. At the same time, the authors would like to acknowledge other researchers for assisting in providing references and information during the conduct of this research and manuscript.

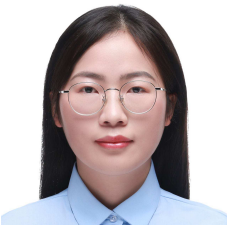
Authors' information. Cui Pei, born in 1996, postgraduate. Her main research interests include edge computing, task unloading method and smart contract.

Guisheng Fan received the B.S. degree in computer science from the Anhui University of Technology in 2003, and the M.S. and Ph.D. degrees in computer

science from the East China University of Science and Technology in 2006 and 2009, respectively, where he is currently a Research Assistant with the Department of Computer Science and Engineering. His research interests include formal methods for complex software system, service oriented computing, and techniques for analysis of software architecture.

Huiqun Yu (Senior Member, IEEE) received the B.S. degree in computer science from Nanjing University in 1989, the M.S. degree in computer science from the East China University of Science and Technology (ECUST) in 1992, and the Ph.D. degree in computer science form Shanghai Jiaotong University in 1995. He is currently a Professor of computer science with the Department of Computer Science and Engineering, ECUST. His research interests include software engineering, high confidence computing systems, cloud computing, and formal methods.

Yiming Yue, born in 1997, postgraduate. Her main research interests include edge computing, trusted allocation of resources and smart contract.



References

[1] Koutsopoulos I, Iosifidis G. Auction mechanisms for network resource allocation[C]. // 8th International Symposium on Modeling and Optimization in Mobile Auction mechanisms for network resource allocation. IEEE, 2010: 353-362.

[2] Kemmoe V Y, Stone W, Kim J, et al. Recent advances in smart contracts: A technical overview and state of the art[J]. IEEE Access, 2020, 8: 117782-117801.
DOI: 10.1109/ACCESS.2020.3005020

[3] Wei X, Li X. Data trading certification based on consortium blockchain and smart contracts[J]. IEEE Access, 2021, 9: 3482-3496.
DOI: 10.1109/ACCESS.2020.3047398

[4] Lin S Y, Zhang L, Li J, et al. A survey of application research based on blockchain smart contract[J]. Wireless Networks, 2022, 28(2): 635-690.
DOI: 10.1007/s11276-021-02874-x

[5] Wu S, Chen Y, Wang Q, et al. CReam: A smart contract enabled collusion-resistant e-auction[J]. IEEE Transactions on Information Forensics and Security, 2019, 14(7): 1687-1701.
DOI: 10.1109/TIFS.2018.2883275

[6] Chen Y H, Chen S H, Lin I C. Blockchain based smart contract for bidding system[C]// 2018 IEEE International Conference on Applied System Invention (ICASI). IEEE, 2018: 208-211.
DOI: 10.1109/ICASI.2018.8394569

[7] Chakroun O, Cherkaoui S. Resource allocation for delay sensitive applications in mobile cloud computing[C]// 41st IEEE Conference on Local Computer Networks (LCN). IEEE, 2016: 615-618.
DOI:10.1109/LCN.2016.105

[8] Sarkar I, Adhikari M, Kumar N, et al. Dynamic task placement for deadline-aware IoT applications in federated fog networks[J]. IEEE Internet of Things Journal, 2022, 9(2): 1469-1468.
DOI: 10.1109/JIOT.2021.3088227

[9] Mithun M, Vikas K, Qi Z, et al. Optimal pricing for offloaded hard- and soft-deadline tasks in edge computing[J]. IEEE Transactions on Intelligent Transportation Systems, 2021: 1-11.
DOI: 10.1109/TITS.2021.3117973

[10] Gao H, Li X J, Zhou B W, et al. Energy efficient computing task offloading strategy for deep neural networks in mobile edge computing[J]. Computer Integrated Manufacturing Systems, 2020, 26(6): 1607-1615. (in Chinese)
DOI: 10.13196/j.cims.2020.06.017

[11] Wei F, Chen S X, Zou W X. A greedy algorithm for task offloading in mobile edge computing system[J]. China Communications, 2018, 15(11): 149-157.
DOI: 10.1109/CC.2018.8543056

[12] Zavodovski A, Bayhan S, Mohan N, et al. DeCloud: Truthful decentralized double auction for edge clouds[C]// International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019, 2157-2167.
DOI: 10.1109/ICDCS.2019.00212

[13] Sun Y E, He H, Li X Y, et al. Privacy-preserving strategyproof auction mechanisms for resource allocation[J]. Tsinghua Science and Technology, 2017, 22(2): 119-134.
DOI: 10.1007/978-3-319-42553-5_2

[14] Kumber D, Baranwal G, Raza Z, et al. A truthful combinatorial double auction-based marketplace mechanism for cloud computing[J]. Journal of Systems and Software, 2018, 140: 91-108.
DOI:10.1016/j.jss.2018.03.003

[15] Zhou G Q, Wu J G, Chen L. TACD: A three-stage auction scheme for cloudlet deployment in wireless access network[C]// 12th International Conference on Wireless Algorithms, Systems, and Applications (WASA). 2017: 877-882.
DOI: 10.1007/978-3-319-60033-8_76

[16] Xia C P, Chen H, Liu X L, et al. ETRA: Efficient three-stage resource allocation auction for mobile blockchain in edge computing[C]// 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2018: 701-705.
DOI:10.1109/ICPADS.2018.00096

[17] Liu X L, Wu J G, Chen L, et al. Efficient auction mechanism for edge computing resource allocation in mobile blockchain[C]// 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2019: 871-879.
DOI: 10.1109/HPCC/SmartCity/DSS.2019.00127

[18] Ma X Y, Xu D, Wolter K. Blockchain-enabled feedback-based combinatorial double auction for cloud markets[J]. Future Generation Computer Systems the International Journal Of Escience, 2022, 127: 225-239.
DOI:10.1016/j.future.2021.09.009

[19] Gao Z P, Lin B, Xiao K L, et al. A dynamic resource allocation algorithm based on auction model in mobile blockchain network[C]// 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE). IEEE, 2019: 1605-1610.
DOI: 10.1109/EITCE47263.2019.9095058

[20] Asif I M, Benay R, Sarbant R. Auction-based resource allocation mechanism in federated cloud environment: TARA[J]. IEEE Transactions on Services Computing, 2022, 15(1): 470-483.
DOI: 10.1109/TSC.2019.2952772

[21] Liu T L, Wu J G, Chen L, et al. Smart contract-based long-term auction for mobile blockchain computation offloading[J]. IEEE Access, 2020, 8(99):

36029-36042.
DOI: 10.1109/ACCESS.2020.2974750

[22] Desai H, Kantarcioglu M, Kagal L. A hybrid blockchain architecture for privacy-enabled and accountable auctions[C]// 2019 IEEE International Conference on Blockchain (Blockchain).IEEE, 2019:34-43.
DOI: 10.1109/Blockchain.2019.00014

[23] Liu Y M, Yu R, Li X, et al. Decentralized resource allocation for video transcoding and delivery in blockchain-based system with mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2019, 68(11): 11169-11185.
DOI: 10.1109/TVT.2019.2937351

[24] Huang H, Ye Q, Zhou Y T. Deadline-aware task offloading with partially-observable deep reinforcement learning for multi-access edge computing[J]. IEEE Transactions on Network Science and Engineering, 2021.
DOI:10.1109/TNSE.2021.3.311054.

[25] Cloud computing price comparison,2021, Cloudorado. [Online].
Available: https://www.cloudorado.com/.

[26] Sonmez C, Ozgovde A, Ersoy C. EdgeCloudSim: An environment for performance evaluation of edge computing systems[C]// The 2nd International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, 2017: 39-44.
DOI: 10.1109/FMEC.2017.7946405

[27] Jeon Y, Baek H, Pack S, et al. Mobility-aware optimal task offloading in distributed edge computing[C]// 35th International Conference on Information Networking (ICOIN), 2021: 65-68.
DOI: 10.1109/ICOIN50884.2021.9334008

[28] Mu X, Stallaert J, Whinston A B. Solving the combinatorial double auction problem[J]. European Journal of Operational Research, 2005, 164(1): 239-251.
DOI: 10.1016/j.ejor.2003.11.018

[29] Mcafee R P. A dominant strategy double auction[J]. Journal of Economic Theory, 1992, 56:434-450.
DOI: 10.1016/0022-0531(92)90091-U

[30] Samimi P, Teimouri Y, Mukhtar M. A combinatorial double auction resource allocation model in cloud computing[J]. Information Sciences, 2016: 357.
DOI: 10.1016/j.ins.2014.02.008

[31] Chichin S, Vo Q B, Kowalczyk R. Double-sided market mechanism for trading cloud resources[C]// 2015 IEEE / WIC / ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). ACM, 2015: 198-205.
DOI: 10.1109/WI-IAT.2015.85