Auction-Based Dynamic Resource Allocation in Social Metaverse

Nan Liu, Tom H. Luan, Yuntao Wang, Yiliang Liu, Zhou Su

School of Cyber Science and Engineering

Xi'an Jiaotong University

Outline

- Background & Motivation
- System Model & Problem Formulation
- Basic Idea & Solution
- Validation & Conclusion

Background

Applications of Social Metaverse

Co-Working

Virtual Music Concert

VR-Gaming

In 2021, Facebook, a well-known social media giant, rebranded itself as "Meta"

The virtual concert featuring Travis Scott in the online game Fortnite attracted more than 12 million players

…

BBC. Fortnite's travis scott virtual concert watched by millions. Accessed: Apr. 24, 2020. [Online]. Available:https://www.bbc.com/news/technology-5241064

Motivation

Features:

Immersion → Rendering avatars and virtual scenes is resource-intensive.

- Utilization of **edge servers' resources** becomes crucial.

Sociability → Interaction among users should be smooth.

- Users with social connections **form a group**, and they access **the same edge server**.

Dynamic → Users dynamically join and depart the network with dynamic requirements on resources.

Heterogeneity → Edge servers offer various levels of Quality of Experience (QoE). Users have different QoE preferences.

As an edge server may host a large-scale and dynamic body of users and social groups, there needs a distributed algorithm to achieve a global resource allocation solution while relying solely on the existing knowledge of edge servers and all groups

Outline

- Background & Motivation
- **System Model & Problem Formulation**
- Basic Idea & Solution
- Validation & Conclusion

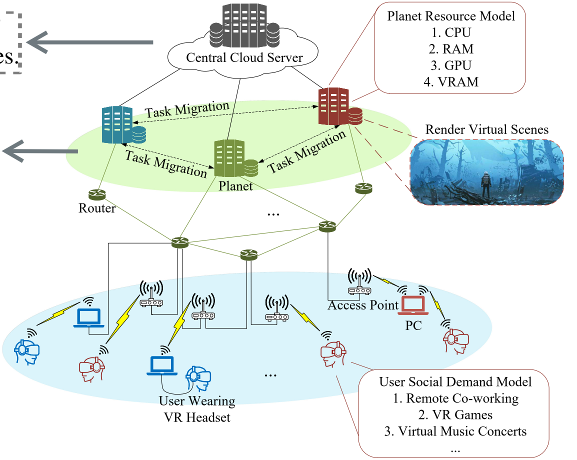
System Model

Network Model

Central Cloud Server: stores users' data, such as virtual assets and identities.

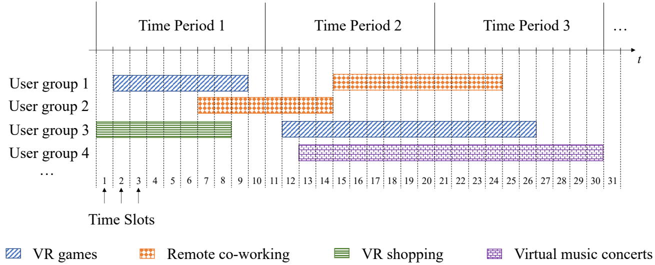
Edge Servers (Planets) $\mathcal{J} = \{1,...,J\}$**:** render virtual scenes and provide personalized spaces for users.

User groups $\mathcal{I} = \{1,...,I\}$**:** each user group $i \in \mathcal{I}$ has a set of $\mathcal{G}_i = \{1,...,G_i\}$ users.



System Model

Dynamic Social Demand Model of User Group



- Time horizon → multiple time periods, each time period → Z time slots.
- Each user group has a timetable $\mathcal{T}_{i,s} = \{A_{i,s,t} | t = 1, \ldots, Z\}$ for the next time period, where $A_{i,s,t}$ indicates whether i will engage in s in t in the next time period.
- Before the start of each time period:

  Each user group determines the Planet access scheme $\mathbf{x}_i = \{x_{i,j} | j = 1, 2, \ldots, J\}$

  Each Planet determines pricing scheme $\mathbf{p}_j = \{p_{j,r} | r = 1, 2, \ldots, R\}$

  During a time period, Planet access schemes and pricing schemes will not be changed.

System Model

Payoff Model of Planet

Planet's Payoff: the total revenue - the total cost

$$\Pi_j^{Pt} = \boxed{\sum_{i,s,r} p_{j,r} c_{s,r}^{j} r_{i,s} x_{i,j}} - \boxed{\sum_{i,s,r} q_{j,r} c_{s,r}^{j} r_{i,s} x_{i,j}}$$

Notation	Description
$x_{i,j}$	The amount of data that user group i plans to upload to Planet j for processing
$p_{j,r}$	The price of leasing a unit amount of resource r in Planet j
$c_{s,r}^{j}$	The resource consumption of Planet j when processing unit data in activity type s
$r_{i,s}$	The proportion of data generated by user group i during activity s to X_i
X_i	The total amount of data generated by user group i
$q_{j,r}$	The operational expenditures incurred by Planet j for utilizing unit amount of resource r

System Model

Payoff Model of User Group

User Group's Payoff: the utility - payments to Planet and routers - payment for task migration

$$\Pi_i^{ug} = \boxed{U_i\left(\sum_j x_{i,j}\right)} - \boxed{\sum_j \left(\sum_{s,r} p_{j,r} c_{s,r}^{j} r_{i,s} + \sum_s n_{i,s} r_{i,s} \sum_{u \in \mathcal{G}_i} d_{u,j}\right) x_{i,j}} - \boxed{\sum_j d_{j',j} m_i \beta sgn(x_{i,j})}$$

U_i is a continuously differentiable and strictly concave function that monotonically increases, where its derivative satisfies $U_i'(z) = (X_i/z)^{\alpha_i}$, $\alpha_i \in [1,2]$

Notation	Description
$d_{u,j}$	The network distance between user u and Planet j
$n_{i,s}$	The payment of group i buying bandwidth resource from a single router for unit data in activity s
$D_{i,s}$	The duration of activity s conducted by user group i during this time period
m_i	The amount of data that user group i needs to migrate
β	The cost per unit data migration paid by the user group to a single router

Optimization Problem Formulation

♦ ***For each Planet:*** It aims to determine a reasonable resource pricing scheme $\mathbf{p}_j = \{p_{j,r} | r = 1, 2, \ldots, R\}$ that satisfies the following three constraints:

- $\Pi_j^{Pt} \geq 0$, so each Planet should ensure that $p_{j,r} \geq q_{j,r}$;
- Capacity constraint, *i.e.*, $\sum_{i,s} x_{i,j} r_{i,s} c_{s,r}^j A_{i,t,s} / D_{i,s} \leq C_{j,r}, \forall j, r, t$;
- When satisfying the above two constraints, the resource prices should be kept as low as possible to attract more users to engage in social Metaverse activities.

♦ ***For each user group:*** After observing the resource prices announced by Planets, each user group aims to determine the optimal Planet access scheme $\mathbf{x}_i = \{x_{i,j} | j = 1, 2, \ldots, J\}$ by solving the following optimization problem.

$$\mathbf{P1}: \max_{\mathbf{x}_i} \quad \Pi_i^{ug}$$
$$s.t. \quad \begin{cases} C1: \sum_j x_{i,j} \leq X_i, \\ C2: x_{i,j} \geq 0, \ \forall j. \end{cases}$$

Outline

- Background & Motivation
- System Model & Problem Formulation
- **Basic Idea & Solution**
- Validation & Conclusion

Basic Idea

1

Planet Access Scheme of User Group

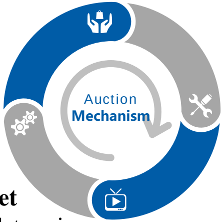
Each user group determines its Planet access scheme $\mathbf{x}_i$

- ✓ Which Planet to access
- ✓ How much data to upload

2

Bid for Resource

After determining the Planet access scheme $\mathbf{x}_i$, each user group calulates the bids, and sends the bids to the corresponding Planet.



3

Resource Pricing Scheme of Planet

Receiving the bids, each Planet determines its resource pricing scheme $\mathbf{p}_j$, which should satisfy the three constraints:

- ✓ $\Pi_j^{Pt} \geq 0$, *i.e.*, $p_{j,r} \geq q_{j,r}$
- ✓ Capacity constraint
- ✓ Set prices as low as possible

4

Algorithm

- ✓ Nash Equilibrium
- ✓ Auction-Based Distributed Dynamic Resource Allocation (DDRA) algorithm

Planet Access Scheme of User Group

Let $f_{i,j} = \sum_{s,r} p_{j,r} c_{s,r}^{j} r_{i,s} + \sum_{s} n_{i,s} r_{i,s} \sum_{u \in \mathcal{G}_i} d_{u,j}$, then the simplified expression of Π_i^{ug} is shown as follows:

$$\Pi_i^{ug} = U_i(\sum_j x_{i,j}) - \sum_j f_{i,j} x_{i,j} - \sum_j d_{j',j} m_i \beta sgn(x_{i,j})$$

Group with no task migration requirement

Compute the derivative $\partial \Pi_i^{ug} / \partial x_{i,j} \rightarrow$

$$x_{i,j}^* = \begin{cases} X_i, & \text{if } f_{i,j} < U_i'(X_i), \\ U_i'^{(-1)}(f_{i,j}), & \text{if } f_{i,j} \geq U_i'(X_i). \end{cases}$$

$\rightarrow$ Substitute $x_{i,j}^*$ into Π_i^{ug}

$\rightarrow$ $f_{i,j} \nearrow, \Pi_i^{ug} \searrow$

$\rightarrow$ Access the Planet with the minimum $f_{i,j}$

Group with task migration requirement

$$\Pi_i^{ug} = \begin{cases} U_i(x_{i,j}) - f_{i,j} x_{i,j} - d_{j',j} m_i \beta, & \text{if } j \neq j', \\ U_i(x_{i,j}) - f_{i,j} x_{i,j}, & \text{if } j = j'. \end{cases}$$

$\rightarrow$ Optimal data upload amount is still $x_{i,j}^*$

$\rightarrow$ Substitute $x_{i,j}^*$ into Π_i^{ug}, and access the Planet corresponding to the maximum payoff

Bid for Resource

After obtaining the optimal Planet access scheme $\mathbf{x}_i = \{x_{i,j} | j = 1, 2, \ldots, J\}$, each user group calculates the bids:

$$w_{i,j,r}^{s} = p_{j,r} c_{s,r}^{j} r_{i,s} x_{i,j}^{*}, \forall j, s, r.$$

$w_{i,j,r}^{s}$ represents the leasing payment of resource r that group i submits to Planet j in order to carry out activity s.

Then, each user group sends the bids to the corresponding Planet

Resource Pricing Scheme of Planet

Recall three constraints:

- $\Pi_j^{Pt} \geq 0$, *i.e.*, $p_{j,r} \geq q_{j,r}$;
- Capacity constraint;
- Set prices as low as possible.

After receiving the bids, each Planet set the appropriate prices:

$$\hat{p}_{j,r} = \max\{q_{j,r}, \frac{\sum_{i,s} A_{i,t,s} \frac{w_{i,j,r}^s}{D_{i,s}}}{C_{j,r}} | t = 1, 2, ..., Z\}$$

Analysis

√ For the first constraint: $\hat{p}_{j,r} \geq q_{j,r}$;

√ For the second constraint:

After updating the resource prices, the actual data amount $\hat{x}_{i,j}$ that can be processed satisfies the following inequality:

$$\hat{x}_{i,j} \underbrace{r_{i,s} c_{s,r}^j}_{\text{actual amount of } r \text{ allocated to } i \text{ for } s} \hat{p}_{j,r} \leq x_{i,j}^* r_{i,s} c_{s,r}^j p_{j,r}, \ \forall i, j, r, s$$

$$\sum_{i,s} \frac{\hat{x}_{i,j} r_{i,s} c_{s,r}^j A_{i,t,s}}{D_{i,s}} \hat{p}_{j,r} \leq \sum_{i,s} \frac{x_{i,j}^* r_{i,s} c_{s,r}^j A_{i,t,s}}{D_{i,s}} p_{j,r}.$$

$$c_{j,r,t}^{allocated} \qquad \hat{p}_{j,r} \geq \frac{\sum_{i,s} A_{i,t^*,s} \frac{w_{i,j,r}^s}{D_{i,s}}}{C_{j,r}}$$

$$c_{j,r,t}^{allocated} \leq C_{j,r}, \ \forall j, r, t$$

√ For the third constraint: Set the lowest price that meets the constraints 1 and 2.

Nash Equilibrium

The auction game is said to be in a Nash Equilibrium if there exists an allocation $\mathbf{x}^*$ and a set of resource prices $\mathbf{p}^* = \mathbf{p}(\mathbf{x}^*)$, such that no user group can unilaterally deviate from its current Planet access scheme to obtain a higher payoff, i.e.,

$$\Pi_i^{ug}(\mathbf{x}_i^*; \mathbf{x}_{-i}^*; \mathbf{p}^*) \geq \Pi_i^{ug}(\mathbf{x}_i; \mathbf{x}_{-i}^*; \mathbf{p}^*), \forall i,$$

where $\mathbf{x}_i$ represents any Planet access scheme of group i that satisfies $\mathbf{x}_i \neq \mathbf{x}_i^*$ and $\mathbf{x}_{-i}^*$ are optimal strategies of all user groups except for group i.

Algorithm

Algorithm 1: The DDRA Algorithm

Input: Resource prices $\mathbf{p} = \{p_{j,r}|\forall j, r\}$ and Planet access scheme of each group $\mathbf{x}_i = \{x_{i,j}|\forall j\}$ in the current time period, timetable $\mathcal{T}$ of each group for the next time period;

Output: Resource prices $\hat{\mathbf{p}} = \{\hat{p}_{j,r}|\forall j, r\}$ and Planet access scheme of each group $\hat{\mathbf{x}}_i = \{\hat{x}_{i,j}|\forall j\}$ in the next time period;

Initialization: ϵ, $\mathbf{x}_i^*$, $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$

while $\|\hat{\mathbf{x}} - \mathbf{x}^*\| > \epsilon$ **do**

- **for** *user group* $i = 1$ *to* I **do**
  - **if** i *have no task migration requirement* **then**
    - Access the Planet with the minimum $f_{i,j}$;
  - **else**
    - Access the Planet with the maximum Π_i^{ug}, according to (8);
  - Obtain $x_{i,j}^*$ via (6);
  - To prevent oscillations, let

$$x_{i,j} = \gamma x_{i,j}^* + (1-\gamma)\hat{x}_{i,j}, 0 < \gamma < 1; \quad (16)$$

  - Make bids: $w_{i,j,r}^s = p_{j,r} c_{s,r}^j r_{i,s} x_{i,j}$;
- **for** *Planet* $j = 1$ *to* J **do**
  - Receive bids from user groups and calculate (9) to obtain the updated resource price $\hat{p}_{j,r}$;
  - Broadcast $\hat{p}_{j,r}$ to all groups;
- **for** *user group* $i = 1$ *to* I **do**
  - According to (10), calculate the actual amount of data that can be processed:

$$\hat{x}_{i,j} = \min_{\mathbf{r}} \frac{p_{j,r}}{\hat{p}_{j,r}} x_{i,j}^*; \quad (17)$$

- Calculate $\|\hat{\mathbf{x}} - \mathbf{x}^*\|$.

Each user group determines its Planet access scheme

Bid for resource

Each Planet determines its Resource pricing scheme

Whether or not reach the Nash equilibrium

Outline

- Background & Motivation
- System Model & Problem Formulation
- Basic Idea & Solution
- **Validation & Conclusion**

Validation

Simulation Settings

Scenario Description

- 1 central cloud server
- 3 Planets
- 200 user groups
- 1 group has 2~50 members
- 1000 × 1000 grid
- 1 time period = 10 time slots
- 1 time slot = 1s

01

03

Planet Parameter

- CPU, GPU, RAM, VRAM
- $c_{s,r}^{j}$, $q_{j,r}$ follow uniform distribution
- $C_{j,r}$:

Planets	CPU (cores)	RAM (GBytes)	GPU (SMs)	VRAM (GBytes)
Planet 1	96	256	420	120
Planet 2	128	384	420	120
Planet 3	128	384	336	96

User Group Parameter

- The amount of data generated by a user in one time slot follows a Gaussian distribution
  mean~[30,50]Mb
  standard deviation~[0.5,2]
- $n_{i,s}$ ~[1,10]× 10^{-4}¢

02

04

Auction Setting

- Auction Type-1:

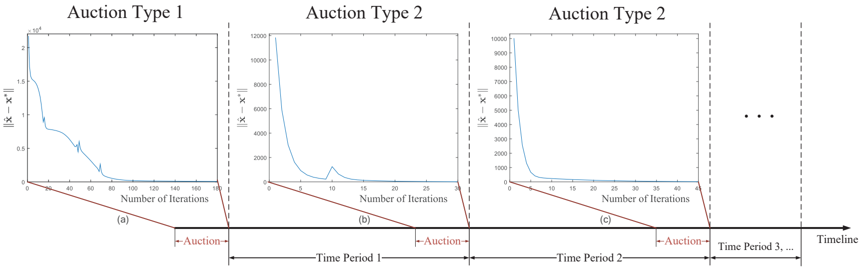
  All groups do not have task migration requirements

- Auction Type-2:

  At least one group has task migration requirement

Validation

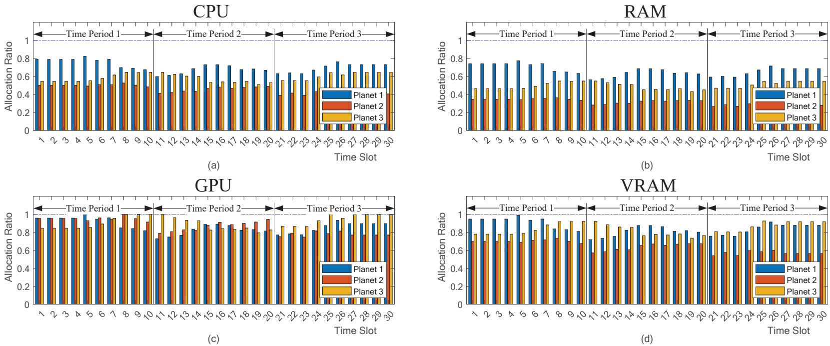
◻ Algorithm Convergence



- In each auction process, number of iterations ↗, $\|\hat{\mathbf{x}} - \mathbf{x}^*\|$ ↘
- The algorithm converges faster in the auction processes(b) and (c) than (a)
  - ❖ in the latter two auction processes, some user groups need to consider the problem of task migration, and they are more inclined to not change the access Planet, which helps the algorithm converge.

Validation

- Resource allocation ratios of 4 resources in Planets 1∼3 over 30 time slots



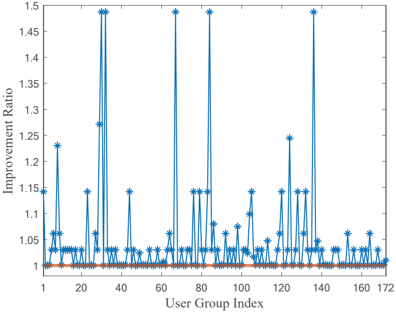
Resource allocation ratio = allocated resource / resource capacity

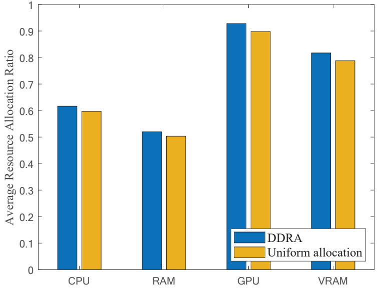
- The allocation ratios change dynamically → system dynamics
- The allocation ratios ≤ 1 → resource capacity constraint ✓

Validation

Allocation Effectiveness of Multi-dimensional Resources

Uniform allocation: a fraction $\sum_{r,s} w_{i,j,r}^{s} / \sum_{i,r,s} w_{i,j,r}^{s}$ of 4 types of resources of Planet j is allocated to user group i at the NE





Improvement ratio: the data amount each user group can be processed by Planet under DDRA algorithm / under uniform allocation

Average resource allocation ratio: resource allocation ratios when averaged over time slot 1~30 and over all Planets

Conclusion

01

02

03

04

✓ We devise a three-layer network architecture of the social Metaverse wherein resource-scarce users can lease resources from Planets to enhance their QoE.

✓ We consider the system dynamic feature and analyze the dynamic social demand model of user groups.

✓ We propose the DDRA algorithm to dynamically allocate the various types of limited resource in Planets to user groups.

✓ Simulation results show that the proposed algorithm is able to dynamically and efficiently allocate the resources and outperforms the benchmark scheme.

Thank you!

Nan Liu
ln201225@stu.xjtu.edu.cn