Profit- Aware Computing Server Clustering and Task Scheduling in the Computing Power Network

Xiaoyao Huang（huangxy32@chinatelecom.cn), Remington R. Liu， Jie Wu, and Baoxian Zhang

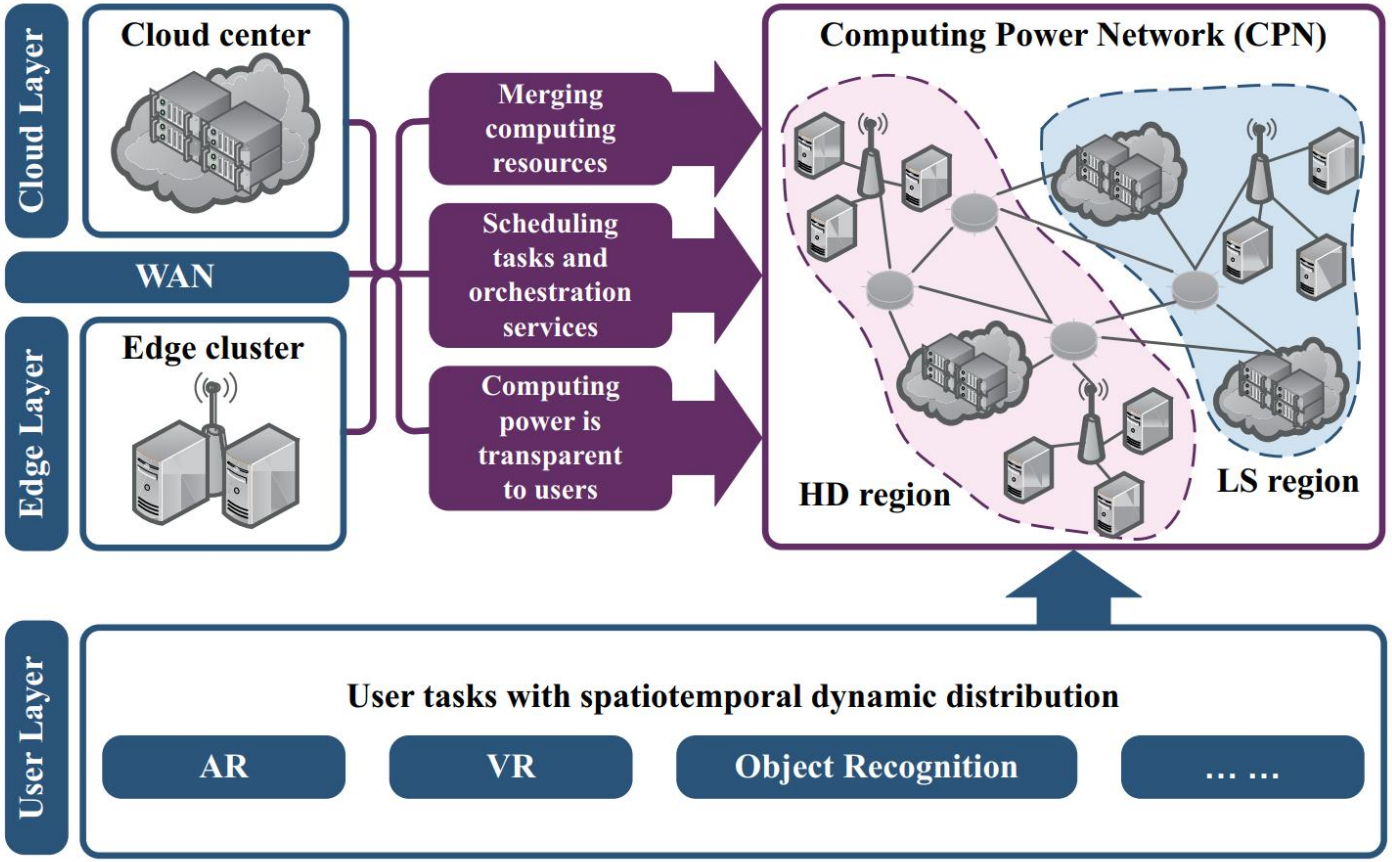
2024.12

Outline

- Background and Motivation
- System Model
- Algorithm Design
- Performance Evaluation

Background & Motivation

- The Computing Power Network (CPN) offers a flexible management and allocation of **integrated network and computing resources** to meet the increasing computation demands with improved system resource utilization performance and satisfactory quality of services.



Necessity of Server Clustering

- Computing resources exhibit different processing capabilities, geographical locations, and costs.
- The computational capacity of a single computing node is often insufficient for some AI tasks with large computational power demands.
- Optimized computing nodes clustering within given latency circles to meet the requirements of large computational tasks is of great significance.

Objecitve: Joint optimization of computing node clustering and task scheduling considering the games between the platform and the clusters for maximized platform profit.

Contribution

01

Optimization Problem

This paper studies the problem of joint optimization of computing node clustering and task scheduling considering the games between the platform and the clusters for maximized platform profit. We formulate this problem as an integer programming, which is known to be NP- hard.

02

Proposing a Deep Reinforcement Learning Approach

To address this issue, we propose a deep reinforcement learning-based server clustering and auction- based task scheduling algorithm working at different time scales.

03

Extension Sumulations

Extensive simulations are conducted to evaluate the performance of the proposed algorithm and the results show its high performance.

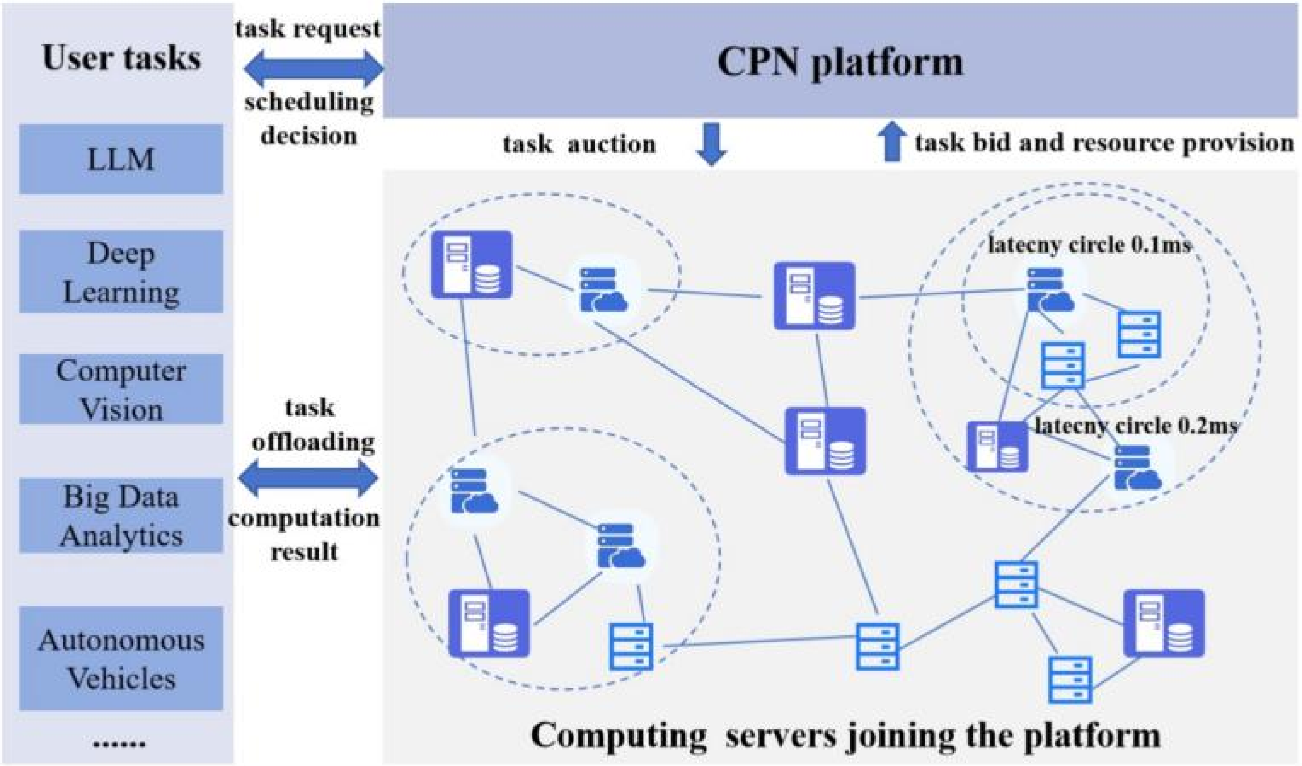
Outline

- Background
- System Model
- Algorithm Design
- Performance Evaluation

System model-network model

The CPN system consists of a **CPN platform**, **computing servers** located in different geographical locations with varying computational capabilities, and **various tasks** from user side, which have dynamic spatiotemporal characteristics and varying computational demands.

- **Users** subscribe computing services from the platform on a **paid basis** and offload tasks to the servers according to the platform's scheduling decisions.
- The **computing servers** (denoted by S) join the system to provide resource and earn revenue via task processing.
- The **platform** works to form clusers of servers that are closely located with low enough latency for meeting the synchronization requirement of processing large tasks and schedule the tasks to proper participating servers while satisfying the tasks' QoS requirements to maximize its own profit.



System model –task delay model

Total delay of a task

$$t_{n,k}^{total} = t_{n,k}^{trans} + t_{n,k}^{wait} + t_{n,k}^{proc}$$

Transmission delay

$$t_{n,k}^{trans} = \sum_{s \in P_{n,k}} \frac{m_n}{v_s} + \sum_{s \in P_{n,k}} \frac{d_s}{V_s},$$

Queueing delay

$$t_{n,k}^{wait} = \frac{L_{n,k}}{f_{C_k^t}}.$$

Processing delay

$$t_{n,k}^{proc} = \frac{l_n}{f_{C_k^t}}.$$

System model –Utility of Different Parties

The utility of platform

The fee for processing task

$$p_n = p(1 + \frac{\mu}{\tau_n}) l_n,$$

$$U_c = \sum_{\delta \in \Delta} \sum_{r_n \in \mathcal{R}^\delta} \left(p_n - \sum_{\mathcal{C}_k^t \in \mathcal{C}} y_{n,k} g_{\mathcal{C}_k^t, n} \right).$$

The utility of a user

$$U_n = \sum_{\mathcal{C}_k^t \in \mathcal{C}} (y_{n,k} v_n - p_n).$$

The utility of a cluster

$$c_{n,k} = \sum_{i \in \mathcal{C}_k^t} \left(\varepsilon_e l_{n,i} f_i^2 + \varepsilon_t \frac{l_{n,i}}{f_i} \right),$$

$$U_k = \sum_{\delta \in \Delta} \sum_{r_n \in \mathcal{R}^\delta} y_{n,k} \left(g_{\mathcal{C}_k^t, n} - c_{n,k} \right) \frac{l_{n,i}}{l_n},$$

System model –Problem Formulation

NP—hard

$$\textbf{P1:} \max_{\boldsymbol{x},\boldsymbol{y}} \sum_{t\in\mathcal{T}} \sum_{\delta\in\Delta} \sum_{r_n\in\mathcal{R}^\delta} \left(p_n - \sum_{\mathcal{C}_k^t\in\mathcal{C}} y_{n,k} g_{c_k^t,n} \right) \tag{10}$$

profit maximization

$$\text{s.t.} \sum_{\mathcal{C}_k^t\in\mathcal{C}^t} y_{n,k} = 1, \forall r_n \in \mathcal{R}^\delta \tag{10a}$$

a task can only be assigned to one cluster

$$\sum_{\mathcal{C}_k^t\in\mathcal{C}^t} y_{n,k} t_{n,k}^{total} \leqslant \tau_n, \forall r_n \in \mathcal{R}^\delta \tag{10b}$$

delay requirement of a task

$$\max_{p,q\in\mathcal{C}_k^t} \|\boldsymbol{L}_p - \boldsymbol{L}_q\|_2 \leqslant R_{\max}, \forall \mathcal{C}_k^t \in \mathcal{C}^t \tag{10c}$$

limit the size of a cluster

$$y_{n,k} \in \{0,1\}, \forall r_n \in \mathcal{R}^\delta, \mathcal{C}_k^t \in \mathcal{C}^t \tag{10d}$$

$$x_{i,j} \in \{0,1\}, \forall i,j \in \mathcal{S} \tag{10e}$$

$$U_c \geqslant 0, U_n \geqslant 0, U_i \geqslant 0, \forall r_n \in \mathcal{R}^\delta, i \in \mathcal{C}_k^t \in \mathcal{C}^t \tag{10f}$$

rationality of all parties

Outline

- Background
- System Model
- Algorithm Design
- Performance Evaluation

Algorithm design-two stage optimization

Due to the huge and complex optimization space due to the large amount of tasks with spatiotemporal dynamics and heterogeneous resources, we propose a DRL based server Clustering Algorithm and an auction-based Task scheduling algorithm (DCAT) to address this problem.

.

DCAT

DRL based server Clustering Algorithm and an auction-based Task scheduling

Large time scale

clustering decision

Deep-reinforcement learning

clustering decision:including the sizes of formed clusters as well as their member nodes

Small time scale

scheduling decision

Auction - based scheduling algorithm

scheduling decision: scheduling the tasks to the clusters

Algorithm design- Markov Decision Process for clustering decision

The problem can be modeled as a Markov Decision Process (MDP), which can be expressed as a 5-tuple $M = \langle S_{rl}, A_{rl}, R_{rl}, P_{rl}, \gamma \rangle$.

State (S_{rl})

The CPN needs to determine which servers for cluster formations based on the computing power of each server and the spatial and temporal distribution of task arrivals and servers. Therefore, the status set is defined as the computing power of each server, the task arrival rate of each area, and the task workload, data size,as well as deadline

Action (A_{rl})

The action set is the clustering decision of CPN

Reward ($R_{rl}(\cdot,\cdot)$)

The reward function is the objective function of the platform profit maximization problem

Discount rate γ

$\gamma \in (0,1]$

Algorithm design-Procedure for clustering decision

The primary purpose of Algorithm 1 is to learn an effective clustering strategy that maximizes the platform's profit by optimizing the sizes and compositions of different clusters.

Algorithm 1: A2C-based Server Clustering Algorithm.

```
Initialize Q function approximator Q_w(s, a), initialize policy
  approximator π_θ(s, a|θ);
for t = 0,1,2,... do
    Sampling clustering strategy a_t ~ π_θ(s_t, a_t|θ) and make
      a_t satisfy the constraints, and then execute it;
    for δ = 0,1,2,... do
        for task r_n ∈ R^δ do
            Algorithm 2 is executed to assign the task to a
              certain cluster and reward r^δ_{t,n} is returned;
        end
    end
    Calculate r_t = Σ_δ Σ_{r_n ∈ R^δ} r^δ_{t,n} and observe s_{t+1};
    δ ← r_{t+1} + γQ_w(s_{t+1}, a_{t+1}) − Q_w(s_t, a_t);
    w ← w + βδ∇_w Q_w(s_t, a_t);
    θ ← θ + α∇_θ log π_θ(s_t, a_t) Q_w(s_t, a_t);
    t ← t + 1;
end
```

Algorithm 3: Procedure of DCAT Algorithm.

```
Train the initial model according to Algorithm 1 and deploy
  at the platform;
for t = 0,1,2,... do
    Obtain the clustering decision x based on the
      observation according to Algorithm 1;
    for δ = 0,1,2,... do
        Execute Algorithm 2 for task scheduling.
    end
end
```

Algorithm design-Procedure for scheduling decision

Algorithm 2: REA Algorithm.

Input: Cluster set $\mathcal{C}^t$, task r_n, current workload of each server, the bid reduction coefficient γ.
Output: The income after task assignment.

$k \leftarrow 0$, $p_n^{(k)} \leftarrow p_n$;
According to the cluster set $\mathcal{C}^t$ and the running status and workload of each server, candidates that meet the offloading task conditions are obtained and a candidate set $\mathcal{C}_a^{(k,t)}$ is formed;
while $k \leqslant K$ **do**
  if $\mathcal{C}_a^{(k,t)} = \emptyset$ **then**
    Task r_n's assignment failed;
    return 0;
  end
  if $|\mathcal{C}_a^{(k,t)}| = 1$ **then**
    Assign task r_n to the server $\mathcal{C}_i^t \in \mathcal{C}_a^{(k,t)}$;
    return $p_n - p_n^{(k)}$;
  end
  $k \leftarrow k + 1$;
  $p_n^{(k)} \leftarrow \gamma p_n^{(k-1)}$;
  The clusters that will continue to participate in the auction form a new candidate set $\mathcal{C}_a^{(k,t)}$;
end

Algorithm 2 is an auction-based task assignment mechanism that operates at a small timescale. It is used to assign tasks to clusters in a way that satisfies the rationalities of all parties involved and maximizes the platform's profit.

For REA strategy, we give the following theorem.

***Theorem** 1: REA satisfies IC*

Theorem 2: If $\gamma > 1 - \frac{\min_{p\in \mathcal{C}^t\backslash q}\{c_{n,p}-\min_{q\in \mathcal{C}^t} c_{n,q}\}}{p_n^{(0)}}$,

then the task will be assigned and the difference between the bid and the minimum cost is no greater than $\min_{p\in Ct\backslash q}\{c_{n,p} - \min_{q\in Ct}c_{n,q}\}$.

Outline

- Background
- System Model
- Algorithm Design
- **Performance Evaluation**

Performance Evaluation-simulation setting

Parm	Values	Unit	Parm	Values	Unit
$\lvert \mathcal{S} \rvert$	20+i10, $i \in [0, 4]$	-	f_i	[200, 300]	GHz
l_n	[500, 900]	G	m_n	20	Mb
τ_n	[2, 3]	s	$R_{\max}$	100+i50,$i \in [0, 4]$	km
v_s	[100, 500]	Mb/s	$P_{n,k}$	[1, 24]	-
V	3×10^8	m/s	μ	2	-
p	10	-	γ	0.8	-
ϵ_e	10^{-28}	-	ϵ_t	20	-
λ	20, 30, 40, 50	-	ξ	0.99	-

- We implemented the proposed algorithm using PyTorch 2.1 and CUDA 12.1, leveraging gym as the environment framework.
- The server locations and task arrival patterns follow the dataset [18] with computing resources data published by Huawei Group.

[18]J. Shi, K. Fu, Q. Chen, C. Yang, P. Huang, M. Zhou, J. Zhao, C. Chen, and M. Guo, “Characterizing and orchestrating vm reservation in geo-distributed clouds to improve the resource efficiency,” in *Proceedings of the 13th Symposium on Cloud Computing*, 2022, pp. 94–109.

Performance Evaluation-baseline algorithms

For the **server clustering** problem, the following baseline algorithms are simulated:

• Random Alg. (Rand): Servers are randomly divided into clusters, and each cluster satisfies the constraints.(This baseline is used to illustrate the results of no optimization.)

• K-Means based Alg. (KM): The number of clusters n is given in advance, and the K-Means algorithm is executed for clustering according to the geographic locations of the servers and delete illegal connections between servers according to the constraints. (This baseline is used to illustrate the results of optimization based on geographic location information only.)

For the **task allocation** problem, the following baseline algorithm is simulated.

• One-off Bid REA (OB-REA): The CPN platform only bids once and then assigns the task to a random cluster in the set of qualified candidates. (This baseline is used to illustrate the impact of multiple bids and one-off bid on results.)

Performance Evaluation

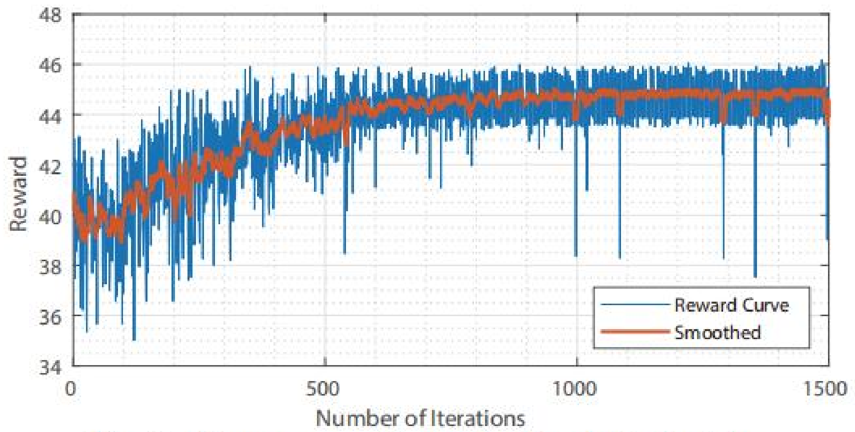


Fig. 2. Convergence of reward value during training.

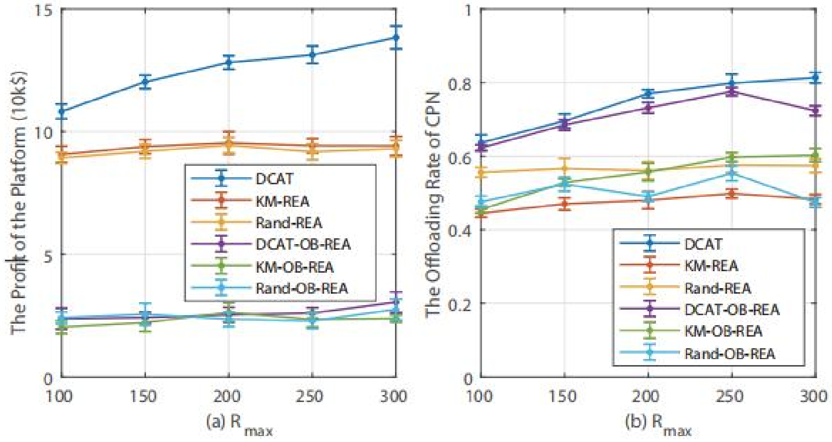


Fig. 4. Impact of cluster size on the profit and offloading rate.

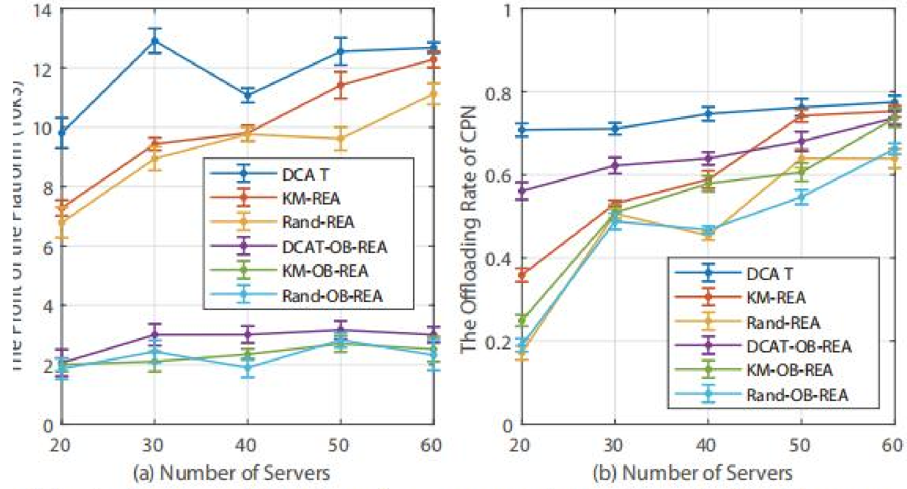


Fig. 3. Impact of number of servers on the profit and offloading rate.

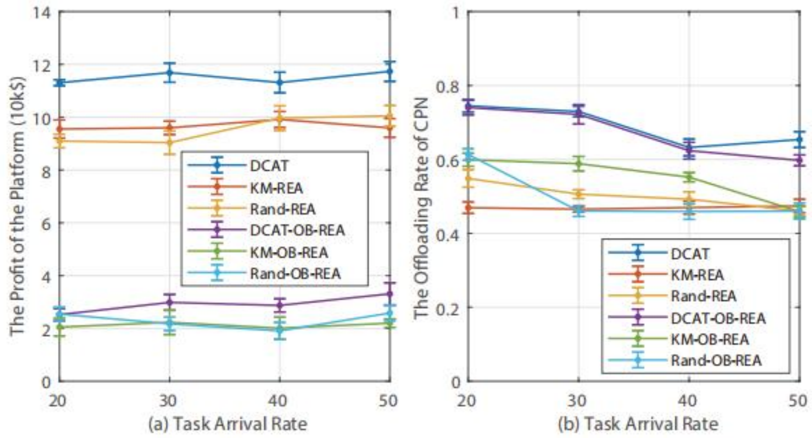


Fig. 5. Impact of task arrival rate on the profit and offloading rate.

Thanks !