Harvesting Game: Strategic User Modeling, Mechanism Design and Equilibrium Analysis for Harvesting Resource

Jiawei Li[1] , **Siwei Wang**[2] , **Jessie Hui Wang**[1] and **Jilong Wang**[1]

[1]Tsinghua University
[2]Microsoft Research Asia
li-jw19@mails.tsinghua.edu.cn

Abstract

Harvesting resources have emerged as a cutting-edge cloud computing solution, aimed at optimizing machine utilization and enhancing revenue for large-scale cloud platforms. By offering dynamically sized virtual machines (VMs) at discounted prices, platforms can effectively address resource bottlenecks. However, existing pricing mechanisms are vulnerable to untruthful reporting and collusion, putting the platform's integrity and financial viability at risk. To address this issue, we formalize the ”Harvesting Game”, a leader-follower game that incorporates strategies of lying and collusion among users and optimal mechanism design by the platform. Equilibrium analysis of Harvesting Game provides useful insights for both users and cloud platforms. Further experiments based on multi-agent learning under diverse settings are being conducted.

1 Introduction

Harvesting resource is a kind of innovative cloud computing solution, specifically designed to optimize machine utilization and augment revenue for large-scale cloud platforms [Ambati *et al.*, 2020]. By offering unsold resources in the form of dynamically-sized virtual machines (VMs) at significantly reduced prices, platforms can effectively and efficiently address resource bottlenecks and improve overall performance. Furthermore, the concept of resource harvesting extends beyond cloud computing and can be applied to various other scenarios, such as power and spectrum harvesting in network systems. Modern harvesting techniques include CPU harvesting [Ambati *et al.*, 2020][Zhang *et al.*, 2021][Wang *et al.*, 2021], memory harvesting [Fuerst *et al.*, 2022], storage harvesting [Reidys *et al.*, 2022], energy harvesting [Liu *et al.*, 2020], and more. While our primary focus within this paper is on harvesting CPU resources, the insights, methodologies, and solutions we present can be readily adapted and applied to other resource types as well.

Workshop on Artificial Intelligence for Critical Infrastructure (AI4CI 2024) @ IJCAI'24 , Jeju Island, South Korea, https://sites.google.com/view/aiforci-ijcai24/, August 4, 2024. Eds: F. Silva, W. Su, R. Glatt, Y. Wang.

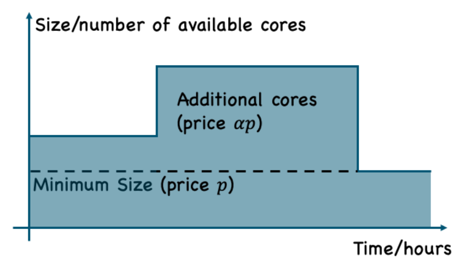


Figure 1: Sketch of a harvesting resource.

The inherent dynamic nature of harvesting resources necessitates a two-tiered pricing mechanism. The first tier involves charging a user-defined minimum size at a predetermined fixed price, p, per core hour. The second tier entails providing additional harvested cores beyond this minimum size at a further discounted rate based on the fixed price, αp. The process of selling and allocating a harvesting resource involves 3 steps: users report their minimum size requirement, m_i'; the cloud platform then allocates a harvesting resource with an actual minimum size, $\bar{m}_j \geq m_i'$, according to a specific allocation mechanism, if available; and finally, users are charged based on their reported minimum size, m_i', and the actual running time of the resource.

Despite its ability to improve resource utilization, satisfy users' service level requirements and provide attractive discounts for additional harvested cores, the current pricing mechanism is intrinsically flawed due to its lack of truthfulness. This issue arises from the discrepancy between users' reported minimum size and the actual size of allocated harvesting resources. Users can exploit this mismatch to obtain extra discounts by misreporting a lower minimum size requirement ($m_i' < m_i$) than their true needs, effectively cheating the system. Furthermore, collusion and trading among users can exacerbate this cheating behavior, reducing risks and amplifying benefits for dishonest users, ultimately undermining the integrity and financial viability of the cloud platform.

In this paper, we address these challenges and present the following contributions:

- We analyze current resource allocation and pricing mechanisms of harvesting resource and identify the cheating and collusion issue, which reduce cloud platforms' revenue.
- We first formalize the Harvesting Game, a leader-follower game compromising lying and collusion in the followers' strategies and leader's optimal mechanism design based on such strategies. Equilibrium analysis of Harvesting game gives useful insights to both users and the cloud platform.

2 Harvesting Resource

A harvesting resource harvests all unsold resource on a machine. According to the system design of cloud platforms [Ambati *et al.*, 2020] [Fuerst *et al.*, 2022] [Reidys *et al.*, 2022], each machine can only host one harvesting resource at most. Its size is not controllable by the cloud platform and often dynamically changes over time. In practice, users usually put those jobs that are flexible in parallelism and interruption-resilient to run on harvesting resources, such as various batch jobs. A user job running on a harvesting resource requires total M_i core hours to finish. It requires a minimum size m_i to start and can fully exploit any amount of additional cores above this threshold. Users declare their minimum sizes m'_i for accepting a harvesting resource. Cloud platforms charge a harvesting resource by its core hours in use: a fixed price p per core hour for using the resource with user declared minimum size per core hour, plus a further discounted price αp per hour for any additional resource harvested above the user declared minimum threshold, as shown in figure 1.

3 Cheating and Collusion

Current pricing and allocation mechanism improves harvesting resource utilization and is attractive to users. However, we identify that users can exploit this mechanism to get extra discount by lying. The problem becomes even more severe when collusion exists among users, which reduces the revenue of cloud platforms. Unless other stated, we consider users total demand $M_i = m_i$ and resource allocation in one hour in the following paper. We will explore more complicated situations where $M_i > m_i$ and resource allocations more than one shot in the future.

Exploiting the Price by Lying

A user's incentive to lie about their resource requirements is primarily driven by the price differential between the base price p and the discounted price αp. For example, if a user, whose true minimum size m_i is 5 core, knows that there is no harvesting resource with $\bar{m}_j = 4$. Then it could deliberately report its minimum size as $m'_i = 4$. According to the allocation mechanism, the cloud platform may allocate a larger resource that satisfies the minimum size requirement, e.g. a harvesting resource with actual minimum size $\bar{m}_j = 5$, to this user, if no other user requests for that resource. Comparing with truthfully reporting 5 core, it gets the 5th core with discount price αp instead of price p.

Exploiting the Price by Collusion

The risk of reporting a smaller minimum size $m'_i < m_i$ is that the user might be allocated a resource that is insufficient for their minimum size requirements. However, this risk decreases when there is a possibility of trading or sharing resources among users, which mitigates the consequences of being allocated an insufficient resource. For example, in a super simple case as shown in figure 2, there are 2 harvesting resource with actual minimum size $\bar{m} = (3, 2)$. There are 3 users with truthful minimum size requirements $m = (3, 2, 2)$, and total core hours $M = (3, 2, 2)$. If all 3 users report their minimum size requirements truthfully, as shown in the left part 2a, then the 3 core harvesting resource will be allocated to the 3 core user. The 2 core harvesting resource goes to one of the 2 users requiring 2 core minimum size. The total revenue of selling these 2 resources for the cloud platform per hour, i.e. the total cost of buying these resources for the 3 users, is calculated as

$$3\ cores \times p \times 1\ hour + 2\ cores \times p \times 1\ hour = 5p \quad (1)$$

However, if the 3 users form a collusion, as shown in the right part 2b, they could set an agreement and all report that they need a resource with the minimum size of 3 core. Then the 3 core harvesting resource and 2 core harvesting resource go to the 3 users, where the 3 core harvesting resource is sold to a $m' = 2$ user. The total revenue of selling these 2 resources for the cloud platform per hour, i.e. the total cost of buying these resources for the 3 users per hour, is calculated as

$$2\ cores \times p \times 1\ hour + 1\ core \times \alpha p \times 1\ hour \quad (2)$$

$$+2\ cores \times p \times 1\ hour = 4p + \alpha p \quad (3)$$

The total cost of 3 users decreases $(1 - \alpha)p$ comparing with truthfully reporting individually.

4 Harvesting Game

4.1 Strategic Harvesting Users

Players

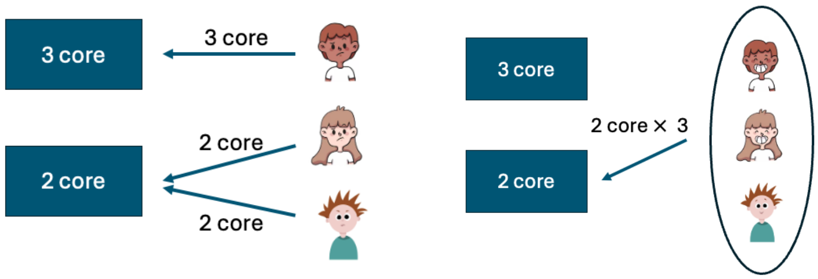
There is a set of players (users) $N = \{1, 2, ..., n\}$. Each user requires one harvesting resource with true minimum size m_i.

Strategies

Players have 3 strategic options: Truthful Reporting (T_i), Individual Lying (L_i), and Collusion ($C_{k,i}$). All reported size requirements (cores) are integer values. Specifically, in T_i, the user i reports $m'_i = m_i$. $L_i = k$ means that the user i misreports its minimum size as $m'_i = k < m_i$. $C_{k,i} \in \{0, 1\}$ and $C_{k,i} = 1$ means that the user i joins a collusion group where all members agree to report a coordinated size k. Especially, a collusion group with coordinated report size k has a common resource pool P_k, where users with excess cores $x_i > m_i$ with users do not get enough cores $x_i < m_i$ with prices $q \in [\alpha p, p)$.

Environment

- *Resource Supply.* The total number of resources and their sizes determine the supply side of the game. Define a set R of resources, each characterized by a size $\bar{m}_j$ for $j \in R$.



(a) Individual truthful.

(b) Collusion.

Figure 2: An example of users forming a collusion and get extra discount.

Payoffs

Players' payoffs are defined on the resource allocation outcomes x_i, i.e. the number of cores allocated to user i.

- *Truthful Reporting* (T_i). The payoff function of T_i is

$$U_i(T_i, s_{-i}) = v_i(x_i) - p \cdot x_i \quad (4)$$

where $x_i = 0$ or $x_i \geq m_i$.

- *Individual Lying* (L_i). The payoff function of L_i is

$$U_i(L_i, s_{-i}) = v_i(x_i) - p \cdot m'_i \quad (5)$$

where $x_i = 0$ or $x_i \geq m'_i$. Since $m'_i < m_i$, potentially $m'_i \leq x_i < m_i$.

- *Collusion* ($C_{k,i}$). The payoff function of $C_{k,i}$ is

$$U_i(C_{k,i}, s_{-i}) = v_i(x_i) - p \cdot k \quad (6)$$

where $x_i = 0$ or $x_i \geq k$.

Equilibrium

For a game G defined by players N, strategies $S = \{T, L, C\}$, and payoffs U, a strategy profile $s^* = (s^*_1, s^*_2, ..., s^*_n)$ is a Nash Equilibrium if

$$U_i(s^*_i, s^*_{-i}) \geq U_i(s_i, s^*_{-i}) \quad \forall s_i \in S_i, \forall i \in N \quad (7)$$

where s^*_{-i} represents the strategies of all other players except player i.

4.2 Resource Allocation Mechanism

Mechanism

The resource allocation mechanism $\mathcal{A}$ takes the resource set R and users' reported minimum size requirements profile m' (comprising either truthful reports, lies, or collusion-driven reports) as input and outputs a set of resource allocation outcomes x, i.e.

$$x = \mathcal{A}(R, m') \quad (8)$$

Design Goal and Constraints

The design goal of mechanism $\mathcal{A}$ is to maximize the cloud platform's revenue $Rev(\mathcal{A})$. The design constraints are that $x_i \in \{0\} \cup [m'_i, \infty), \quad \forall i \in N$.

4.3 Leader-followers Game

We formalize the harvesting game between the cloud platform and strategic users as a 2-stage leader-followers game. In the first stage, the leader (cloud platform) announces a resource allocation mechanism $\mathcal{A}$. In the second stage, the followers (users) observe $\mathcal{A}$, and strategically report their minimum size requirements, resulting the report profile m'. If the followers' game G reaches a Nash equilibrium, and the cloud platform cannot find a mechanism $\mathcal{A}'$ with $Rev(\mathcal{A}') > Rev(\mathcal{A})$, then we say that the 2-stage Harvesting Game reaches an equilibrium.

5 Equilibrium Analysis

In the simple case in figure 1, the equilibrium point depends on users' valuations of their jobs. Informally, when the user 1's job value is high, it is very sensitive to the risk of getting no resources. If users 2 and 3 also have high job values, they are unwilling to trade resources, which increases the chance of user 1 getting no resource in the collusion. In such cases, user 1 reports truthfully $m'_1 = 3$ to make sure it can get the 3 core resources. User 2 and 3 complete for one 2 core resource and the equilibrium strategy is to truthfully report $m'_2 = 2$, $m'_3 = 2$. When both user 2 and 3 have low job values, or all 3 users have low job values, the 3 users form a collusion and negotiate a trading price $q \in [\alpha p, p)$. The optimal coordinated misreport in their collusion $k = 0$ if mechanism $\mathcal{A}$ allocates resources to users with $m'_i = 0$ or $k = 1$ if mechanism $\mathcal{A}$ refuse to allocate resources to users with $m'_i = 0$.

Finding equilibria in the Harvesting Game becomes hard when there are more diverse resources and users. Inspired by recent advancements in multi-agent learning-based solutions for searching equilibria in games [Foxabbott *et al.*, 2023][Brero *et al.*, 2022], we use learning agents to simulate strategic users and a neural network to represent the cloud platform's mechanism $\mathcal{A}$. Especially, each agent (user) adopts Q-learning, as in [Brero *et al.*, 2022], which shares the same architecture as the more sophisticated programs and has clear economic interpretation [Calvano *et al.*, 2020]. Simulation experiments with more resources and users in diverse settings are being conducted.

6 Conclusion

In this paper, we identify the cheating and collusion issues in the current pricing and allocation mechanism of harvesting resources. We formalize the Harvesting Game, where users can lie or form collusions to maximize their payoffs. Further experiments for equilibrium simulations under diverse settings are being conducted. We will explore repeated interactions and partial observations in the game in the future.

References

[Ambati *et al.*, 2020] Pradeep Ambati, Íñigo Goiri, Felipe Frujeri, Alper Gun, Ke Wang, Brian Dolan, Brian Corell, Sekhar Pasupuleti, Thomas Moscibroda, Sameh Elnikety, et al. Providing {SLOs} for {Resource-Harvesting}{VMs} in cloud platforms. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pages 735–751, 2020.

[Brero *et al.*, 2022] Gianluca Brero, Eric Mibuari, Nicolas Lepore, and David C Parkes. Learning to mitigate ai collusion on economic platforms. *Advances in Neural Information Processing Systems*, 35:37892–37904, 2022.

[Calvano *et al.*, 2020] Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolo, and Sergio Pastorello. Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–3297, 2020.

[Foxabbott *et al.*, 2023] Jack Foxabbott, Sam Deverett, Kaspar Senft, Samuel Dower, and Lewis Hammond. Defining and mitigating collusion in multi-agent systems. In *Multi-Agent Security Workshop@ NeurIPS'23*, 2023.

[Fuerst *et al.*, 2022] Alexander Fuerst, Stanko Novaković, Íñigo Goiri, Gohar Irfan Chaudhry, Prateek Sharma, Kapil Arya, Kevin Broas, Eugene Bak, Mehmet Iyigun, and Ricardo Bianchini. Memory-harvesting vms in cloud platforms. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 583–594, 2022.

[Liu *et al.*, 2020] Xiaoying Liu, Kechen Zheng, Kaikai Chi, and Yi-Hua Zhu. Cooperative spectrum sensing optimization in energy-harvesting cognitive radio networks. *IEEE Transactions on Wireless Communications*, 19(11):7663–7676, 2020.

[Reidys *et al.*, 2022] Benjamin Reidys, Jinghan Sun, Anirudh Badam, Shadi Noghabi, and Jian Huang. {BlockFlex}: Enabling storage harvesting with {Software-Defined} flash in modern cloud platforms. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 17–33, 2022.

[Wang *et al.*, 2021] Yawen Wang, Kapil Arya, Marios Kogias, Manohar Vanga, Aditya Bhandari, Neeraja J Yadwadkar, Siddhartha Sen, Sameh Elnikety, Christos Kozyrakis, and Ricardo Bianchini. Smartharvest: Harvesting idle cpus safely and efficiently in the cloud. In *Proceedings of the Sixteenth European Conference on Computer Systems*, pages 1–16, 2021.

[Zhang *et al.*, 2021] Yanqi Zhang, Íñigo Goiri, Gohar Irfan Chaudhry, Rodrigo Fonseca, Sameh Elnikety, Christina Delimitrou, and Ricardo Bianchini. Faster and cheaper serverless computing on harvested resources. In *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*, pages 724–739, 2021.