OpenGPU Network:
A True Decentralized Computing Ecosystem

by OpenGPU Research Team

Abstract

The OpenGPU Network aims to establish a *true* decentralized physical infrastructure network (DePIN) scheme and an ecosystem on top of it for efficient distribution of computation-heavy tasks, such as AI training and inference, across a network of global peers.

This whitepaper introduces the incentives, architecture, workflow, and security measures that enable an accessible and scalable ecosystem for decentralized computing. By leveraging blockchain technology, OpenGPU presents an open and collaborative environment where individuals, small enterprises, and large organizations can contribute and utilize computational resources without relying on centralized procedures or entities.

The information contained herein is subject to change as our research continues and the OpenGPU Network evolves.

1. Introduction

The rapid growth of AI and machine learning has outpaced the capabilities of traditional centralized infrastructures in terms of accessibility and efficiency. AI applications, especially those involving deep learning, natural language processing, and generative models, require immense computational power. According to recent industry reports, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud collectively control over 60% of the global cloud infrastructure market, highlighting the concentration of power in the hands of a few large entities (Synergy Research Group, 2024; Statista, 2024). The dominance of these centralized cloud systems has resulted in a market with limited competition and a hindrance for innovation. Individuals or organizations needing GPU resources face significant challenges, such as budget limitations, scalability constraints, vendor lock-in and susceptibility to single points of failure. This environment has been causing disruptions to clients seeking reliable AI computation resources and hindering innovation.

1.1. Current Landscape of AI Computation

As computational needs grow, infrastructure must scale accordingly, and the expertise and time required to manage such large systems become a significant barrier for Artificial Intelligence enterprises. Cloud services present significant advantages over maintaining a local infrastructure, particularly in terms of flexibility and process automation. However, there are inherent trade-offs to this deal that may not be immediately apparent. Cloud services are centrally managed with the majority of the market being in the control of a very few organizations. This centralized control implies serious drawbacks to the clients and hinders the innovation across the AI landscape.

One major concern when using cloud services is *vendor lock-in*, where switching providers or migrating workloads become costly and complex due to proprietary services and data formats. According to Flexera's 2020 Report on CIO priorities, 68% of CIOs express worry about vendor lock-in with cloud services (Flexera, 2020).

Building massive data centers to host cloud services is in the best interest of cloud providers economically. Capitalizing on their market dominance and vendor lock-in advantage, they have the power to influence client decisions. One obvious advantage of cloud providers is their pricing power, disrupting free market dynamics. They could easily charge more for serving on multiple locations or overrent their software services and computational capacity. This could create *central points of failure* and effectively increase the price per unit of computation for the client, resulting in *high costs* when scaling.

Large cloud providers have the *power to influence* the decisions of AI initiatives, undermining freedom and innovation. By tightly integrating certain software within their platforms, these providers steer the market and limit diversity. Furthermore, even the mainstream media have noted that Amazon, for example, does take open-source software and offer it as a paid cloud service. They reap financial benefits without necessarily contributing back to the original projects—a practice criticized as "strip-mining" open source (New York Times, 2019). Such behavior enables cloud giants to monetize community-driven efforts while providing little incentive or support to the developers and innovators.

A daring challenge of the emerging era of AI computing is maximizing the *utilization of available hardware*, both in individual devices and data centers. Despite the acceleration in manufacturing of processing units, especially GPUs, many devices owned by individuals spend the majority of time idle. Similarly, data centers often operate below optimal capacity, maintaining a headroom to accommodate peak demand but leading to inefficiencies. According to the Uptime Institute's Global Data Center Survey (2024), data centers are expanding to meet rising demand, but current infrastructure is often underutilized due to challenges in scaling cooling, power and management capabilities. Integrating idle public hardware into the broader AI effort could be a game-changer, especially for small AI enterprises that have difficulty accessing high-end hardware.

Leading suppliers of GPUs used in AI computations have increasingly focused on selling their most advanced chips to hyper-scale data centers and research institutions. For instance, Nvidia's strategy for its high-end H100 series chips has favored larger buyers, reducing accessibility for smaller companies and independent developers. This preferential access effectively *sidelines smaller players*, creating an uneven playing field in the AI scene (Wall Street Journal, 2024). This trend risks further consolidating computational power among large players, exacerbating inequalities in access to critical technology.

1.2. Decentralization Efforts

The current range of decentralized computing solutions uses blockchain technology as a *broker for peer-to-peer transactions*. In these systems, clients who need computing power connect with hardware providers through a blockchain. The blockchain manages transactions and contracts, allowing clients to find and rent the needed resources directly. Payments are typically made using cryptocurrencies, which provide a secure and automated way to complete the transactions. This model aims to offer a decentralized alternative to traditional cloud providers for computational resource rental. We acknowledge that these attempts are well-received and pave the way for further innovations in decentralized computing; nonetheless, they remain *unripe*. While we refrain from naming specific projects or teams to avoid any sense of rivalry, we recognize the shared goal of democratizing computing power. However, we believe that the true form of Decentralized Physical Infrastructure Network (DePIN) should go beyond and achieve decentralizing the entire end-to-end process. Then the potential of a free, competitive and equitable distribution of computing resources could be unlocked to power a democratized AI revolution.

2. The OpenGPU Network

The OpenGPU Network is an ecosystem layout designed to establish a fully democratized process for executing computational tasks across a network of diverse providers. It operates on a blockchain based, permissionless and trustless model with a low barrier to entry for all peers. Creating a free, competitive market in real-time and presenting an infinite scalability potential, it aims to address each and every problem of the current state of the industry.

Clients of the network propose tasks in predetermined frames specifying the workload, conditions, and payment function. Providers, connected to the network via an accessible suite, evaluate the feasibility and profitability of tasks and submit bids in real-time. Once an agreement is reached, tasks are executed either atomically or collaboratively. The integrity of execution and security for both parties against malicious intent are ensured by a consensus layer for computations built on top of Ethereum proof-of-stake protocol of the underlying blockchain.

2.1. Preliminary Concepts

In the rest of the paper, the parties and tasks are illustrated through practical scenarios involving AI businesses, aligned with the current state of the industry.

The ecosystem comprises three types of actors:

- **Clients** are peers in need of computational resources to run their **tasks**, such as an AI company deploying a language model to serve its customers.
- **Providers** are entities with computational resources to offer for profit. These could range from individuals with spare GPU capacity to fully operational data centers.
- **Users** subscribe to the services provided by clients, such as the customers of the aforementioned AI company.

The tasks are categorized into two types as follows:

- **Batch-tasks** have quantitative limits or targets intended for a fixed set of providers. Training an AI model with given settings for a set amount of time is a good example of a batch-task. Also, having a large language model generate a predetermined length of responses can be wrapped as a batch-task. It is worth noting that this approach is quite similar to existing decentralized solutions in structure.

- **Stream-tasks** are flexible and dynamic. With their terms accepted by a group of providers apriori, stream-tasks are then submitted by the client in real-time and on-demand. The group of providers generates responses to the streaming requests specified by the task. A very good example, again, is a large language model of a client operating under a pay-per-response agreement.

For a given task,

- **Executor** is the provider or the set of providers selected to execute the task.

- **Council** is the group of providers assigned to validate the task through the computation consensus.

The setting of the executor and the council for a batch-task differs slightly from that for a stream-task, as explained in the following subsection. The interactions between actors and tasks are orchestrated by the decentralized core protocol of the OpenGPU Network.

2.2. Architectural Overview

At the foundation of the OpenGPU Network is a proof-of-stake blockchain serving as a distributed ledger and a decentralized state machine. While there are several viable alternatives for building such a system, Ethereum is both a pioneer and the widely adopted standard. Originally authored by Vitalik Buterin (2014), Ethereum initially used a proof-of-work consensus mechanism. However, Ethereum transitioned to proof-of-stake, a change referred to as Ethereum 2.0 (Consensys, 2020). The blockchain component enables actors to submit and trace tasks in a trustless and transparent manner. Providers take on the role of block validators of the blockchain. They ensure secure and decentralized execution of transactions and storage of public information.

Providers execute client tasks and collaboratively form a consensus to verify the honesty and integrity of their responses. This is achieved in the computation layer, implemented as a computation consensus system on top of the proof-of-stake mechanism. Verification records are stored publicly on the blockchain, so that all network peers are able to query them.

Submission and assignment of tasks, and the output of providers' executions are managed by the core protocol of the network. It is a composite decentralized application (dApp). Different from other arbitrary dApps, it has direct access to the core components of the network mentioned above, ensuring completeness of the process. Figure 1 illustrates the three main components of the network and their relationships.

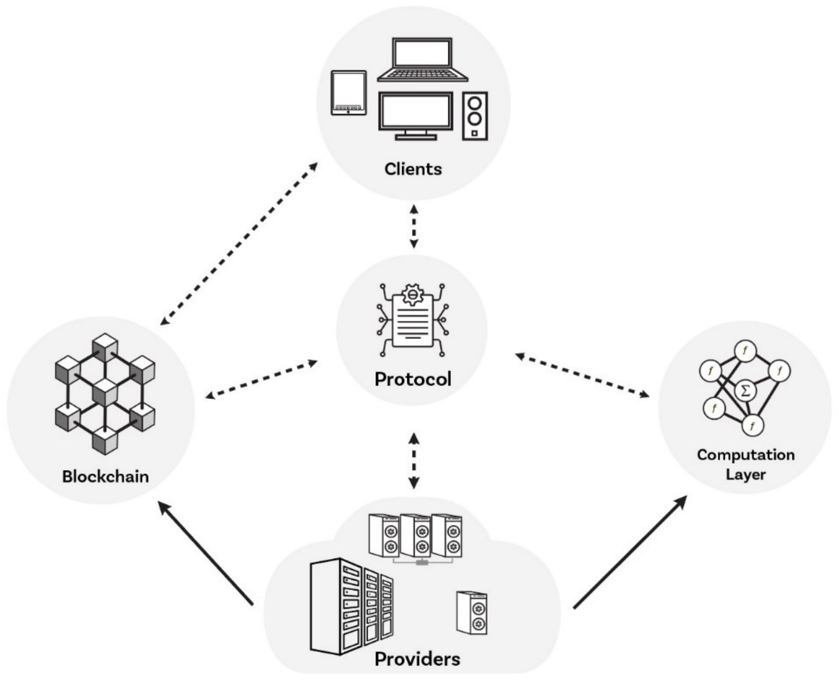

Figure 1. Solid arrows indicate providers maintaining the blockchain and computation layer, while dashed arrows represent interaction between entities.

2.3. Task Workflows

The workflow of a task from its creation and submission by a client to execution and delivery by providers, though very similar in principle, naturally depends on its type. Thus, two workflow descriptions are introduced in this subsection.

A *task prescription* is denoted as $T = \{D, M, I, f(I'), e(r, r'), H\}$, where D is the dataset related to task, M is the task model, I is the set of instructions specifying requirements and quantitative targets of execution, and $f(I')$ is a function of $I' \subseteq I$ to calculate the amount the client is willing to pay given instructions of I' are fulfilled. $e(r, r')$ is a function with a binary codomain to determine whether two responses r and r' are equivalent. It is an essential calculation for the computation consensus. H is the set of conditions to halt or finalize the process.

Training an AI model, such as a neural network (NN), can easily be represented as a batch-task. Then, D is the dataset to train and evaluate the model. M is the specifications of said AI model such as the type of NN and a set of hyperparameters. I may involve number of epochs, time limits, hardware requirements, maximum payment amount and so on. A plausible choice of f would be a logarithmically increasing function that is inversely proportional to time, with council members sharing a portion of the amount. Since the NN steps are deterministic, each checkpoint must output the same model weights. Thus, e

simply compares two weight maps at a given epoch. Finally, H can be, as a mere example, the best of the first 10 bids to be picked as the executor.

The workflow for a batch-task is as follows.

1. Client submits a task prescription T_{batch} via the protocol onto the blockchain.
2. Available providers read the submission event and evaluate the prescription. If the execution is feasible, profitable and there are no better bids placed by another provider, provider p places a bid f_p.
3. When the halting conditions are met, the protocol announces the executor and the council. (In a straightforward implementation, the implied executor is the lowest bidder.)
4. The executor executes the prescribed task while the council validates the work at checkpoints designated by the protocol.
5. The final response to the batch-task is sent to the client and the process is finalized upon mutual confirmation.

This batch-task workflow happily resembles the process of block validation in a blockchain. A task is similar to a transaction and a batch is to a block. Various aspects of the workflow, such as council assignment and security measures, are discussed later in the paper.

Stream-tasks are handled with a subscription method. Providers subscribe to the task and receive streaming requests upon agreement. Responses are sent to the client upon completion. In this case, the council is a subset of size c of all the subscribed executors. Completion of a task is subject to the preferred response methods out of the following:

- **Optimistic response** is where the first submitted response is sufficient to mark the task completion. The later responses from the other providers in the council asynchronously validate the first response via consensus.
- **Weighted response** is a generalization of the former. A weight $w \in [0, 1]$ is assigned to the subscription. The task is complete when $w \cdot c$ responses are submitted by the executors, where c is the number of council members. If there are different responses, the most common response is sent. Consensus is worked out afterwards.
- **Unanimous response** is where a task is complete only after all the providers in the council submit responses and a consensus is reached.

The unanimous response method is the natural choice if correctness of each task is critical. On the contrary, the optimistic response method is the fastest. An illustration of the response methods is given with Figure 2.

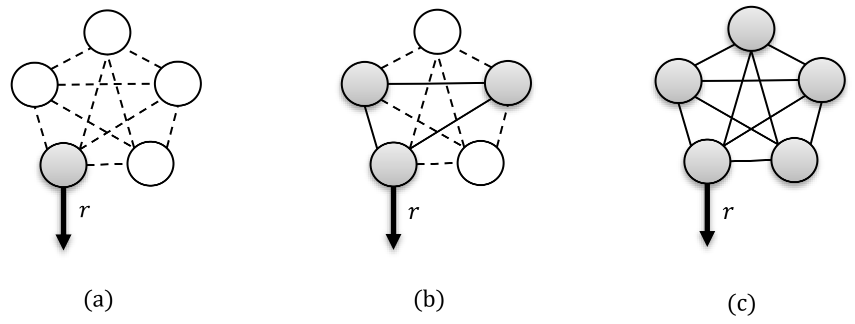

Figure 2. (a) Optimistic response (b) Weighted response (c) Unanimous response Solid lines represent validation occurring between providers before the response is sent, while dashed lines represent validation occurring after the response is sent.

Stream-tasks are tailored for arbitrarily continuous executions, such as running a large language model (LLM) per user prompt. In that context, D may be empty or some initial prompts for the model. M is a description of the model with its parameters. I must specify the response method and the setting of the council. Additionally, it may involve the maximum acceptable response time proportional to the number of generated characters. Then f could be a linear function with respect to the output length. The choice of e is simple, if the model only picks the highest weighted output. Otherwise, the client needs to specify a procedure checking whether an output is possible with the given settings. While H can be empty as well, leaving the halting condition to the protocol, the client could specify a maximum downtime allowed.

The workflow for a stream-task is as follows.

1. Client submits a task prescription T_{stream} via the protocol onto the blockchain.
2. Available providers read the submission event and evaluate the prescription. If the execution is feasible and potentially profitable, provider p sets the given model up and subscribes to the stream.
3. Having reached enough subscribers to form a council, streaming starts with the preferred response method. Stream is interrupted, meaning the service is down, if
    - 3.1. The number of executors drops below c due to idle or unsubscribed providers. If the council reaches the required number with new subscribers, the stream continues.
    - 3.2. Consensus is failed. If the failure is not critical and the council is reformed in time, the stream continues.
    - 3.3. The client retracts.

The stream-task workflow allows clients and providers to dynamically adjust terms in real-time according to market conditions. Providers can choose to stop responding to streaming requests, most likely when they find better uses for their resources. Subscription is open at all times, so a provider with a better bid can join the competition at will. On the other hand, clients can adjust the requirement instruction set I and payment function $f(I')$, allowing them to attract more providers when necessary.

3. Implementation

Let us discuss a blueprint for implementing the OpenGPU Network.

3.1. The Blockchain

At the core of the OpenGPU Network lies a blockchain infrastructure, which can be implemented using any blockchain stack equipped with robust crypto-economic security mechanisms. Among the available options, Ethereum stands out as the standard-setter and the most widely adopted blockchain for decentralized applications. Its proven technology, modular architecture, and mature ecosystem make it a natural choice.

Ethereum serves as the decentralized ledger and state machine, providing the foundation for task submission, assignment, and verification. By leveraging Ethereum's proof-of-stake consensus mechanism, the network ensures both scalability and energy efficiency. Tasks submitted to the blockchain are immutably recorded, which creates a trustless and transparent environment for participants. Moreover, having such a well-equipped blockchain infrastructure allows the OpenGPU Network to match the full capabilities of other Layer-1 blockchains. Applications from other ecosystems can easily be migrated or extended to the OpenGPU Network, boosting the vibrancy of the ecosystem as a whole.

The native currency of the blockchain, **$oGPU**, is integral to the network's operation. It is used to pay gas fees for blockchain transactions and serves as the default currency for task payments. This dual functionality simplifies interactions within the network and highlights a compelling use case for $oGPU.

3.2. The Computation Layer

The computation layer is implemented as a system atop the Ethereum stack, serving as an intermediary between the blockchain and task execution workflows. This layer handles task-specific computation requirements, including consensus validation for batch-tasks and stream-tasks. By abstracting the complexity of computation consensus from the underlying blockchain, the computation layer enables scalability and ensures seamless interaction between clients, providers, and councils.

The integrity and reliability of the task workflows are underpinned by crypto-economic incentives. Clients are required to lock-in safety deposits to get their tasks published. These deposits are used for task payments upon finalization and for penalizing clients in the event of malicious activity. Meanwhile, in addition to execution revenues earned as executors and council members, providers are incentivized through mechanisms that reward honest behavior and penalize malicious intent.

- **Staking:** Providers and council members must stake $oGPU to participate in task execution and validation. This ensures they have a vested interest in maintaining the network's integrity.

- **Penalties:** Malicious actors attempting to submit fraudulent responses or failing to meet task requirements face penalties, including slashing of their stakes. Likewise, clients are subject to enforcements through their safety deposits.

- **Rewards:** Honest execution and validation are awarded proportionally based on the client-supplied payment functions, ensuring fair compensation for computational resources and validation efforts. Providers that participate in the blockchain consensus are additionally rewarded.

These mechanisms align the incentives of all participants, creating a self-sustaining ecosystem that resists exploitation.

3.3. The Protocol Suite

The core protocol serves as the intermediary through which clients and providers interact with the OpenGPU Network. It is implemented as a decentralized application suite, comprising: A **client application**, enabling clients to define task prescriptions, specify requirements, and monitor task progress. It provides an interface for interacting with the blockchain and computation layer to ensure transparency and control. A **provider application**: Allows providers to register their resources, evaluate task prescriptions, and participate in the bidding process. The app integrates with the computation layer to facilitate task execution and response validation, ensuring secure and efficient operations. A system of **smart contracts** that serve as the backbone of the protocol by automating task submissions, bidding processes, payment distributions, and dispute resolutions. These contracts ensure trustless execution of all agreements, immutably recording outcomes and enforcing penalties or rewards based on the consensus layer's validations.

Unlike general-purpose dApps, the core protocol is tightly integrated with the blockchain stack and the computation consensus layer. This direct connection allows the protocol to orchestrate the tasks by setting boundaries and enforce security. The governance of the protocol should ideally be managed by a decentralized autonomous organization (DAO) as the ecosystem matures.

4. Domain of AI Agents

Engaging with the task workflows in real-time is highly impractical for a human. Within the OpenGPU Network, AI agents bridge the gap between the intricate workings of the decentralized ecosystem and its human peers. Those autonomous and intelligent digital entities actively assist clients in defining task prescriptions, evaluating feasibility, and optimizing requirements. They can predict task costs based on market conditions, recommend adjustments to maximize value, and ensure security by aligning task configurations with network standards. For providers, AI agents manage bidding strategies, assess profitability based on real-time market conditions, and computational costs of tasks. Also, they ensure adherence to agreed terms and maintain high levels of reliability.

AI agents could easily play an active role in the overall security of the ecosystem, with their focus on the safety of tasks. Tasks submitted by clients or responses of providers may, intentionally or not, contain harmful content that poses risks to other peers. While crypto-economic security is the main protection against bad actors, AI agents act as vigilant overseers, detecting anomalous activities. By doing so, they can also prevent participants from getting penalized for honest mistakes.

Operating within the framework of the protocol suite, AI agents could leverage privileged access to blockchain records, computation consensus outputs, and smart contract functionalities. They act autonomously yet remain configurable by users, aligning their operations with specific goals and preferences.

As the OpenGPU Network evolves, AI agents will presumably become integral to its decentralized infrastructure; while being powered by the network itself.

5. Conclusion

The OpenGPU Network is a transformative step towards the true decentralization of computing power at a global scale. It overcedes previous decentralization approaches by achieving a complete workflow integration onto the blockchain infrastructure while maintaining a very low barrier to entry for peers. The realization and adoption of the OpenGPU Network will undoubtedly accelerate the coming AI revolution.

References

Statista, 2024, *Cloud computing - statistics & facts* by Lionel Sujay Vailshery
https://www.statista.com/topics/1695/cloud-computing/
Last accessed on 15Nov2024

Synergy Research Group, 2024, *Cloud market growth stays strong in q2 while amazon, google, and oracle nudge higher*
https://www.srgresearch.com/articles/cloud-market-growth-stays-strong-in-q2-while-amazon-google-and-oracle-nudge-higher
Last accessed on 15Nov2024

Flexera, 2020, *The Flexera 2020 CIO Priorities Report*
https://info.flexera.com/FLX1-REPORT-CIO-Priorities-2020
Last accessed on 18Nov2024

New York Times, 2019, *Prime Leverage: How Amazon Wields Power in the Technology World*
https://www.nytimes.com/2019/12/15/technology/amazon-aws-cloud-competition.html
Last accessed on 18Nov2024

Uptime Institute, 2024, *Global Data Center Survey Results*
https://uptimeinstitute.com/resources/research-and-reports/uptime-institute-global-data-center-survey-results-2024
Last accessed on 18Nov2024

WSJ (Wall Street Journal), 2024, *AI's Future and Nvidia's Fortunes Ride on the Race to Pack More Chips Into One Place*
https://www.wsj.com/tech/ai/nvidia-chips-ai-race-96d21d09
Last accessed on 18Nov2024

Vitalik Buterin, 2014, *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*
https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf
Last accessed on 01Dec2024

Consensys, 2020, *What is Proof of Stake?* by Everett Muzzy
https://consensys.io/blog/what-is-proof-of-stake
Last accessed on 01Dec2024