Thunder Whitepaper - Powering the Future of P2P GPU Computing

Contents

I. Executive summary

A. Introduction to thunder

Thunder is a peer-to-peer (P2P) GPU computing network, enabled by a blockchain protocol. The exponential growth in GPU computing demand has strained traditional centralized cloud solutions in cost, availability, and efficiency, while millions of consumer GPUs sit idle. Thunder connects unused GPU compute capacity from otherwise idle personal computers to commercial and research users. It accomplishes this through a distributed network where "hosts" offer GPU compute resources to "users" who pay with TNR tokens.

Thunder is a cloud computing platform first and a cryptocurrency protocol second – the Thunder blockchain protocol functions as a tool to support the decentralized Thunder computing network. Thunder is designed to be as simple to use and as scalable as possible for both users and hosts, without the specialized development required when implementing solutions on traditional cloud platforms.

A crucial focus for Thunder is security, including data encryption, results validation, a trustless system design, and regular reliability tests to ensure dependable service for users. The project's roadmap sets forth an ambitious plan for the future of decentralized GPU computing, including scalability, adoption, and constant innovation.

Thunder Token (TNR), built on the Ethereum blockchain, benefits from the safe, interoperable, and scalable ERC-20 token standard. By deploying on Arbitrum, the Thunder network facilitates high transaction volume while minimizing gas fees.

Alongside utilizing the security and scalability of the Arbitrum network, Thunder also employs off-chain payment channels for each compute operation. This ensures that users are accurately billed for the resources they use, and hosts are paid for the resources they offer while minimizing gas costs

By unlocking idle GPU power, Thunder is set to reshape the GPU computing sector. It offers a more affordable, scalable, and accessible solution for users, while allowing hosts to monetize their resources securely and efficiently. Experience the technology yourself by referencing the official set-up guide.

B. Problem statement

The demand for GPU computing resources has surged in recent years due to expanding applications in areas such as artificial intelligence, machine learning, data analysis, and computer graphics. Traditional centralized cloud solutions struggle to keep up with this demand, facing limitations in terms of cost, availability, and efficiency. These solutions often require high upfront investments including specialized development tooling, lack resource flexibility, and suffer from performance bottlenecks due to centralized infrastructure. Moreover, many users face barriers to entry, such as high costs and limited access to powerful computing resources. In addition, centralized solutions ignore the vast pool of idle GPU resources available in individual systems worldwide. Consequently, there is an unmet need for a solution that addresses these challenges by connecting surging demand to this pool of idle GPUs.

C. Solution overview

The Thunder network presents a solution in the form of a decentralized peer-to-peer (P2P) GPU computing network that combines minimal technical prerequisites with extreme scalability. This network links hosts, possessing unused GPU resources, with users, delivering cost-effective and accessible computational power on-demand, all while rewarding hosts for their otherwise idle hardware. With the integration of a native cryptocurrency protocol, Thunder ensures enhanced scalability and diminished transaction costs, while maintaining privacy for all participants.

The ecosystem features a robust host reliability assessment system, a resource matching algorithm for efficient task allocation, and a secure, transparent billing and payment system underpinned by smart contracts. This system harshly penalizes malicious participants and motivates rapid computation and reliability. By unlocking idle capacity, Thunder aims to reshape the GPU computing landscape, promoting a more cost-effective, scalable, and inclusive environment for both users and hosts.

D. Use case scenarios

1. AI and Machine Learning Research: Researchers and data scientists can leverage Thunder's P2P GPU computing network to efficiently train machine learning models and execute complex AI computations, with a simple command line tool. By harnessing idle GPU resources from hosts worldwide, they can access affordable and scalable computational power, overcoming the constraints of traditional centralized solutions.

2. 3D Rendering and Computer Graphics: Small startups, graphic designers, and game developers can use Thunder's GPU computing network for rendering high-quality 3D models, animations, and visual effects. This decentralized model allows creators to access powerful GPU resources quickly and cost-effectively, speeding up the rendering process and enhancing their creative potential.[PC1]

3. Scientific Simulations: Researchers in fields such as physics, chemistry, and meteorology can utilize Thunder's network for running large-scale simulations and data analysis tasks. Decentralized GPU computing resources can expedite experiments in a cost-effective manner, accelerating scientific discoveries and breakthroughs.

4. Peak Demand capacity: Small operators, such as a research lab or start-up with 10 GPUs that experience capacity shortages during peak times, can benefit from the Thunder network. Labs can simultaneously act as users and hosts to earn TNR during periods of idle capacity, which can then be used to purchase additional resources during peak demand periods.

5. Cryptocurrency Mining: When network pricing allows, cryptocurrency miners can benefit from Thunder’s distributed GPU computing network by utilizing available GPU resources to mine various cryptocurrencies. Participation in the network provides access to existing hardware with simple set-up for hosts, mitigating the environmental impact of traditional dedicated mining setups.

E. Token utility

The Thunder blockchain protocol is supported by Thunder Token (TNR), which functions as the central utility within its peer-to-peer (P2P) GPU computing network by enabling secure transactions between users and hosts. TNR tokens act as a medium of exchange, granting users access to affordable and scalable GPU computing resources, while fairly compensating hosts for their contributed resources. Contrasted with centralized alternatives, Thunder's decentralized approach offers several benefits:

1. Cost-Effectiveness: By pooling idle GPU resources from global hosts, Thunder offers a more competitive and flexible pricing structure for users, with high levels of scalability. The network is designed without the substantial upfront investments required by traditional centralized solutions, making GPU computing resources more accessible, particularly for startups and small users.
2. Scalability: The decentralized framework ensures that the Thunder network can process a high volume of transactions concurrently, theoretically achieving infinite scalability without requiring additional centralized infrastructure. This allows the network to expand and adapt to the increasing demand for GPU computing resources.
3. Resource Availability: Thunder's decentralized GPU computing network permits users to access a diverse and distributed pool of GPU resources, providing a more reliable and readily available computing solution than centralized counterparts.
4. Privacy and Security: With its blockchain-enabled trustless system design, Thunder ensures that transactions between users and hosts are secure and private in a fully double-blind arrangement where both parties remain anonymous. Further, data encryption and containerization intensify the privacy and security of client data, while computed results are validated to ensure network reliability and prevent malicious attacks.
5. Inclusivity and Community Empowerment: Thunder democratizes access to high-performance computing resources by allowing individuals and smaller entities like startups and researchers to monetize their idle GPU resources. This results in a more inclusive and equitable computing ecosystem, empowering users and hosts to actively partake in the growing GPU computing market.

II. Background

A. The growing need for GPU computing

GPU computing has become an essential component of modern technology, driving innovation across various fields, including artificial intelligence, machine learning, big data analytics, and computer graphics. GPUs, with their parallel processing capabilities, provide an efficient solution for handling computationally intensive tasks that traditional CPUs struggle to manage. The increasing reliance on data-driven decision-making, along with the rapid development of AI and machine learning applications, has led to a surging demand for high-performance GPU resources. As industries across the globe continue to embrace digital transformation, the need for accessible and powerful GPU computing solutions is expected to grow exponentially.

B. Limitations of centralized GPU cloud solutions

Centralized GPU cloud solutions, such as those provided by major cloud computing providers, currently dominate the GPU computing market. However, these centralized solutions face several limitations:

1. High Cost: Centralized GPU cloud providers often require substantial upfront investments and may impose high usage fees, making it difficult for small businesses, startups, and individual users to access powerful GPU resources.
2. Resource Flexibility: Centralized solutions often limit scalability, particularly for smaller operators. Thunder allows for real-time capacity management with a market-based approach to capacity management
3. Development overhead: Centralized infrastructure requires substantial ongoing technical investment and engineering, as well as physical data centers, adding to their cost
4. Data Privacy and Security: Storing and processing data in centralized data centers can expose users to potential data breaches and privacy concerns.
5. Limited Accessibility: Centralized GPU cloud providers may not be available in all geographical locations, resulting in restricted access for some users.

C. The promise of decentralization

Decentralized computing offers an alternative to centralized GPU cloud solutions. By leveraging a distributed network of individual GPU resources, decentralization can overcome many of the limitations inherent in centralized systems. Decentralized GPU computing networks can provide:

1. Cost-Effectiveness: By aggregating idle GPU resources from hosts around the world, decentralized networks can offer competitive and flexible pricing structures, making GPU computing resources more accessible to a broader range of users.
2. Resource Flexibility and Efficiency: Decentralized networks can dynamically allocate resources based on user demand, ensuring optimal resource utilization and reducing waste.
3. Enhanced Performance: By distributing computing tasks across multiple nodes in a P2P network, decentralized systems can reduce performance bottlenecks and improve the overall efficiency of GPU computing tasks.
4. Improved Data Privacy and Security: Decentralized systems inherently provide greater data privacy and security, as user data is not stored in a single centralized location, reducing the risk of data breaches.
5. Global Accessibility: A decentralized GPU computing network can provide users with access to resources from various geographical locations, ensuring a more inclusive and accessible computing ecosystem.

By harnessing the power of decentralization, Thunder’s P2P GPU computing network has the potential to revolutionize the GPU computing landscape, offering cost-effective, efficient, and accessible solutions for users and hosts alike.

III. P2P Distributed GPU Computing Network

A. Network architecture

1. Hosts
2. Users
3. Validators

B. Host onboarding process

1. System requirements
2. Verification process
3. Incentives

C. User onboarding process

1. Account creation
2. Acquiring Thunder Tokens (TNR)
3. Requesting GPU compute resources

D. Compute task allocation

1. Resource matching algorithm
2. Task distribution and execution
3. Task completion and output return

E. Trust management

1. Host scoring system

i. Build-up period

ii. Payout curve

2. Randomized testing
3. Ongoing testing

F. Billing and payment system

1. Pay-as-you-go model
2. Smart contract-based payments
3. Dispute resolution

IV. Thunder Token (TNR) - The Cryptocurrency

A. ERC-20 standard

1. Benefits of using the ERC-20 standard

Thunder Token (TNR) is built on the Arbitrum network on the Ethereum blockchain as an ERC-20 token, leveraging the benefits of the widely adopted token standard. The advantages of using the ERC-20 standard include:

a. Interoperability: ERC-20 tokens are compatible with a vast ecosystem of wallets, exchanges, and other decentralized applications (dApps) built on the Ethereum blockchain, allowing for seamless integration and ease of use.

b. Security: The Ethereum network offers a secure and reliable foundation for the Thunder network, protecting user funds and transaction data.

c. Smart Contracts: The ERC-20 standard supports the use of smart contracts, which enables Thunder to implement secure and automated decentralized billing, payments, and other functionality within its P2P GPU computing network.[PC2]

2. Token specifications

a. Name: Thunder Token

b. Ticker: TNR

c. Blockchain: Ethereum (Arbitrum network)

d. Token Standard: ERC-20

e. Initial Supply: 30,000,000 TNR

f. Inflation: 3% steady-state

B. Token distribution

1. Initial token offering

The initial supply of Thunder Token is set at 30,000,000 TNR, with the following distribution:

a. Creators' Reserve: 7,500,000 TNR (25%) are reserved for the project's creators to cover development costs, future expansions, and ongoing maintenance of the network.
b. Public Distribution: 22,500,000 TNR (75%) are allocated for public distribution through two methods:

The token will be distributed along two channels:[PC3]

a. Fixed Conversion Rate with Ethereum: A predetermined portion of the public distribution will be offered at a fixed conversion rate with Ethereum, allowing potential users and hosts to acquire TNR tokens in exchange for Ether (ETH).
b. Subsidizing Early Miners: To incentivize early participation and adoption of the network, a predetermined portion of the public distribution will be allocated as a

bonus hourly payout for early miners. This bonus will be offered in addition to the standard purchased compute rates, rewarding early miners for their contribution to the network's growth.

2. Steady-state inflation

Thunder Token has been designed with a gradual steady-state annual inflation rate of 3%. This rate is targeted to approximate expected macroeconomic inflation, incentivizing network transactions while compensating hosts for their contributions to the network. The inflation mechanism is triggered publicly every two weeks, and newly minted tokens are directly allocated to hosts through the current payment cycle.

C. Transaction fees

Fee structure: Every compute purchase within the Thunder network incurs a 7% fee, which supports the infrastructure, development, and maintenance of the P2P GPU computing network. The breakdown of the distribution of received tokens is as follows:

a. Network's share: The network creators receive 7% of the transaction which contributes to infrastructure, development, and maintenance for the network. This allocation aids in the continuous enhancement and sustainability of the Thunder ecosystem.

b. Hosts' share: The remaining 93% of the transaction is distributed to hosts within the Thunder network. These funds are allocated according to hosts' respective reliability scores and runtimes.

This fee is regulated by contract logic and disbursed every two weeks based off the total deposits in the prior period, while the remainder of the accrued balance can be claimed by the hosts. This requires a signed transaction from an oracle maintained by the network, which tracks runtime, reliability, and other metrics to determine payout.

V. Thunder Smart Contracts

A. ERC-20 token contract

1. Overview

Thunder Token (TNR) leverages the functionality of the ERC-20 standard to provide a secure and reliable token within the Thunder Network. This token contract integrates several key features designed to protect network participants and support network functionality, such as pausability, OpenZeppelin libraries, and inflation distribution

2. Key features

a. Pausable: The Pausable feature allows network administrators to temporarily pause all token transfers in the case of an emergency. This is a security feature that can help protect user funds during unexpected situations. Only the contract owner has the authority to pause and unpause the contract.

b. OpenZeppelin: Thunder Token contract leverages OpenZeppelin libraries, which offer battle-tested implementations of the ERC-20 standard, as well as security features like Pausable and ownable.

c. Inflation: As mentioned previously, the Thunder Token contract includes a mechanism for steady-state inflation, to incentivize transactions within the network and reward hosts for their contributions. This inflation function can be called by anyone, links to the Host Payment contract, and is designed to create new tokens at a rate of 3% per annum, distributed every two weeks.

3. Contract implementation

The Thunder Token contract is implemented using Solidity programming language and deployed on the Arbitrum network on the Ethereum blockchain. The contract features a number of standard ERC-20 functions for transferring tokens, along with custom functions for pausing the contract, handling inflation, and managing the distribution of newly minted tokens. The contract leverages OpenZeppelin's security contracts to ensure a secure and reliable implementation.

4. Link to Host Payment contract for inflation

The host payment address is set by the contract owner and is used to distribute newly minted tokens as part of the steady-state inflation mechanism. The inflation mechanism is publicly triggered every two weeks, with the new tokens being directly distributed to the hosts as part of the current payment cycle.

B. Payment Channel contract

1. Overview

In the Thunder ecosystem, payment channels function as an off-chain solution that enhances transaction processing by optimizing speed and cost-efficiency, while enabling greater pricing fidelity. The network establishes a private transaction channel between a user and the Host Payment contract for each compute session, facilitating multiple transactions without the need for rapid and expensive updates to the main blockchain. This reduces network congestion and gas fees while providing precise time-based billing. Once the compute session ends, the channel is closed and the final state settled on-chain, minimizing on-chain storage requirements.

2. Benefits

Scalability: Payment channels alleviate the scalability challenges faced by traditional cloud solutions, accommodating a high volume of simultaneous transactions within the Thunder network without additional centralized overhead.

Reduced transaction costs: Payment channels lower the cost burden to users by processing transactions off-chain and only settling the final state on-chain, thus significantly cutting down on associated gas fees versus an on-chain solution.

Faster transaction times: Transactions conducted off-chain within payment channels are near-instant, ensuring an uninterrupted user experience and efficient allocation of GPU resources, while precisely tracking billing.

Improved privacy: Payment channels enhance privacy by only recording the final state on the blockchain, thereby obscuring the specifics of individual transactions.

3. Implementation in the Thunder network

Establishing a payment channel: A user, seeking GPU computing resources, initiates a payment channel with the Host Payment contract. The user locks a specified amount of TNR tokens as collateral in a dedicated payment channel smart contract on the Ethereum blockchain. This guarantees that the user has enough funds for the compute session, avoiding the need for multiple on-chain transactions.

Off-Chain transactions: Throughout the duration of a payment channel, users can securely and efficiently conduct transactions. Users transmit off-chain signed transactions reflecting the most recent balance as the session progresses. The network verifies each transaction and continues to provide computational resources in accordance with the agreed terms. The transaction may conclude once the computation is complete or upon receipt of additional signed transactions from the user.

Terminating the payment channel: After the computation task, the network submits the final off-chain transaction to the payment channel contract to close the channel. The contract then distributes the user-locked funds based on the final transaction status, recording the result on-chain. If the network does not submit the final transaction before the pre-set deadline, the user is entitled to retrieve their staked collateral.

B. Host Payment contract

1. Overview

The Host Payment contract defines the logic for payment distributions to network hosts. The contract receives payment from the payment channels opened by the users and tracks the payout schedules for each host. The payment distribution depends on various factors, including total payment received within a specific period, host reliability scores, host runtime, and a fixed fee for maintaining the network infrastructure, as described earlier. This contract also receives any slashed tokens from the Host Staking contract and tokens minted for inflation.

2. Importance

Ensuring trustworthiness: The contract verifies that hosts have staked a certain balance before receiving any payout, in addition to the network staking requirement for computing. If malicious activities such as falsifying results or runtime are detected, the staked balance is slashed, promoting ethical behavior within the network.

Flexible payouts: The Host Payment contract offers flexibility by ensuring that payments and payouts are not directly related. This increases privacy while providing flexibility for random testing, task duplication, and other quality assurance techniques without affecting host payouts.

Scalability: The contract's independence from individual payment channels allows it to scale to handle many simultaneous transactions, decoupling client growth from host growth and enabling the Thunder network to scale smoothly.

Preventing malpractice: The implementation of a "ban period" further aligns economic incentives with ethical behavior by preventing misbehaving hosts from receiving network payouts. If a host attempts to evade this ban by creating a new account, their newly staked balance may also be slashed.

3. Implementation in Thunder network

Duration of each payment cycle: Throughout each cycle, the Host Payment contract holds the sum of all payments received from the client's payment channels. It keeps track of the total amount of tokens withdrawn throughout the current cycle.

End of cycle: At the end of each payment cycle, the contract calculates the total deposits and deducts a predetermined fee, which is then transferred to the designated fee address. The remaining balance is allocated for payout to the hosts. The contract resets for the next payment cycle.

Payouts to hosts: Hosts can withdraw their payment by retrieving a signed confirmation from an oracle, after the contract validates the payment amount and the block number. The Host Payment contract verifies the oracle's signature and checks the host's staked balance before initiating the payout. The exact calculation is centralized and concealed to prevent malicious behavior; however, it is generally a function of runtime and host reliability, [PC4]resulting in a percentage of total post-fee deposits within a cycle.

Adjustments: The contract allows the network owner to change the fee percentage, oracle address, and fee address, providing flexibility to adapt to changing network needs.

C. Host Staking contract

1. Overview

The staking mechanism requires hosts stake a pre-determined amount of TNR before providing compute to or withdrawing payment from the network. This staked balance can be slashed if malicious behavior is detected, creating a financial incentive to provide accurate results.

2. Staking and unstaking

Hosts must stake a specific amount of TNR to participate in the network. When they wish to stop participating, they can initiate an unstaking process. The contract includes a wait period of 28 days for unstaking, during which malicious actors can still have their balances slashed.

3. Slashing distribution

As mentioned previously, the network administrators can 'slash' or penalize hosts who violate the network's rules. The slashed tokens are transferred to the HostPayment contract, aligning incentives of network participants and operators

D. Functionality and interaction

The Thunder protocol is defined by the interactions between four Ethereum-based smart contracts: the Thunder ERC-20 Token, Payment Channel, HostPayment, and HostStaking contracts, as described individually above. Interactions between them are as follows:

1. Host Staking and Thunder ERC-20 token

In the Thunder Network, the HostStaking contract acts as the gatekeeper for hosts wishing to join. Hosts are required to stake a specific amount of TNR prior to providing compute or initiating payout. The staking mechanism ensures commitment and incentivizes hosts to adhere to network rules, thereby contributing to the overall security and reliability of the network.

To initiate the staking process, hosts first need to have the required number of TNR in their possession, as well as Ethereum to cover gas fees. They then approve the HostStaking contract to transfer tokens on their behalf. Following the approval, hosts can call the stake() function of the HostStaking contract, which transfers the required number of tokens from the host's account to the contract. The staked tokens are held within the contract until the host decides to unstake or their stake is slashed due to non-compliance with the network rules.

2. Host unstaking and withdrawal

If a host decides to stop participating in the network, they can initiate the unstaking process by calling the unstake() function on the HostStaking contract. An unstaking request triggers a waiting period, during which the host cannot participate in network activities. Once the waiting period is over, the host can call the disburse() function to retrieve their staked tokens from the contract.

3. Payment channels and host payment

For ongoing transactions between users and hosts, payment channels can be established. These channels allow for efficient, low-cost, off-chain transactions while preserving the security and transparency offered by the blockchain. All payment channels in the thunder protocol interact with the host payment contract, which manages all finalized transactions.

4. Host Payment contract

In addition to managing funds received through payment channels, The HostPayment contract is also the recipient of any slashed tokens from the HostStaking contract, and minted tokens from inflation. The transfer of funds allows for the redistribution of the slashed tokens back into the Thunder ecosystem, further incentivizing network hosts.

In summary, together these smart contracts create a seamless and secure system for transactions between hosts and users in the Thunder Network. The staking and payment mechanisms ensure hosts are incentivized to provide high-quality services while allowing users to pay securely and efficiently.

VI. Security measures

Ensuring the security and integrity of the Thunder network is a top priority. The network employs various security measures to protect user and host data and maintain the stability and reliability of the system.[PC5]

A. Data encryption

Thunder uses state-of-the-art encryption algorithms to secure user and host data during transmission and storage. By encrypting sensitive data, the network safeguards against unauthorized access, data breaches, and other security threats. This ensures that user and host information remains confidential and secure within the Thunder ecosystem.

B. Trustless system design

The Thunder network is designed with a trustless architecture, which minimizes the need for trust between users, hosts, and other parties within the system. Smart contracts, multi-signature wallets, and consensus mechanisms are employed to automate transactions, enforce agreements, and ensure the fairness and transparency of the network. This trustless design reduces the risk of fraud and other malicious activities within the Thunder ecosystem, minimizing overhead as compared to a traditional cloud provider.

C. Results verification

As discussed in the previous section, payment channels are employed within the Thunder network to enable efficient and scalable transaction processing. The implementation of payment channels is secured through multi-signature smart contracts, collateral deposits, and a trusted third party or decentralized validator system. These security measures ensure that transactions within the payment channels remain secure, private, and tamper-proof, while also providing a robust dispute resolution mechanism.

D. Containerization[PC6]

By employing these security measures, Thunder aims to provide a secure and reliable P2P GPU computing network that users and hosts can trust. The network's commitment to security, combined with its innovative technology and robust infrastructure, ensures a stable and secure ecosystem for all participants.

VII. Roadmap

A. Project milestones

B. Future developments

VIII. Conclusion

A. Thunder's vision for decentralized GPU computing

Thunder envisions a world where powerful, accessible, and cost-effective GPU computing resources are available to users and organizations globally through a decentralized P2P network. By leveraging the power of distributed computing, idle GPU power, and blockchain technology, Thunder aims to revolutionize the GPU computing landscape, breaking down barriers to entry and democratizing access to high-performance computing resources.

B. Scaling and adoption

As the Thunder network grows and evolves, a focus on scaling and adoption is crucial to ensure its long-term success. The Thunder team is committed to continuous improvement and innovation, ensuring that the network remains secure, efficient, and user-friendly. By implementing a scalable network architecture built on a flexible blockchain foundation, Thunder aims to provide a scalable and sustainable solution that can adapt to the ever-changing demands of the GPU computing market, and serve customers and their individual needs.

In addition to technological advancements, fostering a strong community of users, hosts, and developers is vital to the network's growth and adoption. The Thunder team will actively engage with the community, seeking feedback, addressing concerns, and collaborating on new features and improvements.

C. Invitation to the community

The Thunder team invites users, hosts, developers, and other stakeholders to join the Thunder ecosystem and contribute to the development and growth of the network. By participating in the network, users can access powerful GPU computing resources, hosts can monetize their idle GPU capacity, and developers can build innovative applications that leverage the power of decentralized GPU computing.

Together, we can drive the adoption of decentralized GPU computing, create new opportunities for collaboration and innovation, and reshape the future of high-performance computing for the benefit of all. Join us in our mission to democratize access to GPU resources and build a more inclusive, secure, and efficient computing ecosystem with Thunder.

IX. Frequently asked questions

1. What is Thunder Token (TNR)?

Thunder Token is a utility token for the Thunder network, a blockchain-based peer-to-peer (P2P) network for sharing GPU computing resources.

2. What are the key features of the Thunder network?

Key features include access to cost-effective, decentralized GPU computing resources, a tokenomics model, payment channel contracts, host payment contracts, and host staking contracts. The network also focuses on security, scalability, and user-friendly interfaces.

3. How does the payment channel function within the Thunder ecosystem?

Payment channels act as an off-chain solution for transaction processing. They establish a private conduit between users and hosts, facilitating multiple transactions without each transaction needing to be registered on the main blockchain. This results in optimized speed, cost-efficiency, and privacy.

4. What are the benefits of using the Thunder network over traditional cloud solutions?

The Thunder network offers improved scalability, reduced transaction costs, faster transaction times, and enhanced privacy. It also leverages blockchain technology to offer a more secure, transparent, and efficient environment for P2P GPU computing.

5. How does the Host Payment contract work?

The Host Payment contract outlines the logic for distributing payments to network hosts. It takes into account various factors such as total payments received within a period, host reliability scores, host runtime, and a fixed fee for maintaining the network.

6. What is the purpose of the Host Staking contract?

The Host Staking contract requires hosts to stake a predetermined amount of TNR tokens before they can provide computing services or withdraw payment from the network. This staked balance can be slashed if malicious behavior is detected, creating a financial incentive to provide accurate results.

7. How does Thunder ensure the security of its network?

Security measures include state-of-the-art data encryption, a trustless system design, results verification using payment channels and containerization. Together, these measures protect user and host data, and maintain the stability and reliability of the system.

8. What is Thunder's vision for the future?

Thunder envisions a world where powerful, accessible, and cost-effective GPU computing resources are available to users and organizations globally through a decentralized P2P network. It aims to democratize access to high-performance computing resources.

9. How can I get involved with Thunder?

The Thunder team invites users, hosts, developers, and other stakeholders to join the Thunder ecosystem and contribute to the development and growth of the network. By participating, you can access powerful GPU computing resources, monetize idle GPU capacity, or develop innovative applications leveraging decentralized GPU computing.

X. Appendix

A. Glossary

1. Blockchain: A decentralized and distributed digital ledger used to record transactions across multiple computers, ensuring the data's security, transparency, and immutability.
2. Cryptocurrency: A digital or virtual currency that uses cryptography for security and operates independently of a central authority, such as a bank or government.

3. Decentralization: The process of distributing or dispersing functions, powers, or resources away from a central authority or location.
4. ERC-20: A technical standard used for implementing tokens on the Ethereum blockchain. It provides a set of rules and functions that allow for the creation of new tokens with interoperability and compatibility within the Ethereum ecosystem.
5. Ethereum: A decentralized, open-source blockchain platform that enables the creation and execution of smart contracts and decentralized applications (dApps).
6. Gas fees: Transaction fees required to perform operations on the Ethereum network, including executing smart contracts, sending tokens, and interacting with dApps.
7. GPU (Graphics Processing Unit): A specialized electronic circuit designed to rapidly process and manipulate complex images and data, commonly used for rendering graphics in video games and other computationally intensive tasks.
8. Multi-signature wallet: A type of digital wallet that requires multiple signatures to authorize a transaction, providing increased security and control over the wallet's assets.
9. P2P (Peer-to-Peer): A decentralized network architecture in which participants directly interact with each other without the need for intermediaries or centralized servers.
10. Smart contract: A self-executing contract with the terms of the agreement directly written into code, which automatically executes and enforces the contract's conditions when predetermined conditions are met.
11. Payment channel: An off-chain scaling solution that allows participants to conduct transactions off the blockchain, with only the final state being recorded on the main blockchain, resulting in reduced transaction costs and increased scalability.
12. Staking: The process of participating in the proof-of-stake consensus mechanism by locking up a certain amount of cryptocurrency in a wallet, supporting the network's security and stability in return for rewards.
13. Tokenomics: The study of the economic systems, structures, and incentives surrounding the creation, distribution, and management of a digital token within a blockchain ecosystem.

B. References

1. Buterin, V. (2013). Ethereum White Paper. Ethereum Foundation. Retrieved from https://ethereum.org/en/whitepaper/
2. Dziembowski, S., Faust, S., Kolmogorov, V., & Pietrzak, K. (2018). Proofs of Space. In Annual International Cryptology Conference (pp. 585-605). Springer, Cham. https://eprint.iacr.org/2013/796.pdf
3. Fabian, V. (2015). ERC-20 Token Standard. Ethereum Improvement Proposals. Retrieved from https://eips.ethereum.org/EIPS/eip-20
4. Merkle, R. C. (1987). A digital signature based on a conventional encryption function. Advances in Cryptology—CRYPTO’87 (pp. 369-378). Springer, Berlin, Heidelberg.

5. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Bitcoin.org. Retrieved from https://bitcoin.org/bitcoin.pdf

6. Poon, J., & Buterin, V. (2017). Plasma: Scalable Autonomous Smart Contracts. White Paper. Retrieved from https://plasma.io/plasma.pdf

7. Poon, J., & Dryja, T. (2016). The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. White Paper. Retrieved from https://lightning.network/lightning-network-paper.pdf

8. Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. First Monday, 2(9). https://firstmonday.org/ojs/index.php/fm/article/view/548/469