**UNITED STATES PROVISIONAL PATENT APPLICATION**

**Title:** Novel Method for Distributed Large Language Model Computation and Verification

**Inventor:** Dylan Herbert Dunn

**ABSTRACT**

The invention is a decentralized computing system that allows clients to submit compute jobs to a network of compute nodes in exchange for payment. The system utilizes a reputation system to ensure quality of work and incentivizes correct computations with staked cryptocurrency. Compute nodes are randomly chosen based on reputation and availability. The output of a compute job is verified by a large incentive-based group of compute nodes using a machine learning model to ensure accuracy. Payments are split between the original compute node and the verifying nodes upon successful completion of a compute job. The system is designed to prevent spamming and ensure fairness while providing a fast and efficient computing network.

**FIELD OF THE INVENTION**

The invention relates generally to the field of distributed computing and more specifically to a system and method for securely distributing LLM (Large Language Model) processing using a distributed blockchain and cryptocurrency. The invention provides a decentralized and secure platform for distributing LLM processing that ensures fast, secure, and reliable data transfer.

The invention leverages blockchain technology to create a distributed network of nodes that can perform LLM processing tasks. By using cryptocurrency as a means of exchange within the network, the invention ensures that nodes are incentivized to perform LLM processing task accurately and efficiently.

The invention uses a unique and novel verification algorithm in order to quickly and efficiently verify the accuracy of computed data. This verification allows for an open method of determining the reputation of nodes and determining the distribution of cryptocurrency compensation.

The field of distributed computing is a rapidly evolving area of technology that has the potential to transform many industries. The present invention represents a significant advancement in this field, providing a novel solution to the problem of securely distributing LLM processing that is both secure, fast, and efficient. The invention allows for LLM to be outsourced to a community of compute nodes allowing for vastly more compute power and a more open access to LLM’s for the general population. The invention solves multiple problems that arise with current technology and/or more naive attempts at creating such a network.

In summary, the present invention provides a secure and decentralized platform as well as multiple novel algorithms for distributing LLM processing that leverages the power of blockchain technology and cryptocurrency. The invention represents a significant contribution to the fields of distributed computing, LLMs, and, cryptocurrencies.

**BACKGROUND OF THE INVENTION**

The field of natural language processing has seen significant advancements in recent years, with the emergence of Generative Pre-Trained Transformer (GPT) models such as GPT-3 and GPT-4. However, these models present some challenges that limit their accessibility and usability for everyday users. Two of the most significant challenges are compute power and censorship.

Compute power is a major challenge for LLMs, as these models require large amounts of computational resources to train and run effectively. Everyday users typically do not have access to the level of computing power required to use these models, and must rely on centralized providers such as OpenAI to provide the necessary resources. This creates a barrier to entry for users who may not have the financial means to access these resources, or who may be hesitant to rely on centralized providers for privacy or other concerns.

Censorship is another significant challenge for LLMs, as centralized providers of these models have control over the models they provide. This can lead to issues of censorship and control over the models, which can impact their reliability and usability. While uncensored models exist, they are often computationally burdensome and not accessible to most consumers.

Additionally, cryptocurrencies present a significant challenge to the field of natural language processing. Traditional methods of blockchain verification, such as those used by Bitcoin, rely on heavy computation that is often wasteful and inefficient. This model aims to provide a useful cause to the computation used by cryptocurrencies in securing blockchains.

While no current example of a distributed network for LLMs exists, this invention proposes to solve some of the major roadblocks one may encounter when trying to build such a network. This includes the need for a reliable and efficient verification and reputation system, which are presented here as novel solutions. The proposed system would incentivize nodes to participate in the network, and would allow users to pay for only the resources they need when they need them. This would make LLMs more accessible and affordable for everyday users, while also providing a decentralized and reliable solution.

The history of this invention started with the personal motivation of the inventor to explore GPT models on their own. However, they faced a major challenge in trying to use these models, as they either had to rely on censored models provided by centralized companies like OpenAI or run one locally, which required a significant amount of local resources. This led to the idea of distributing the network to incentivized nodes, which would allow users to pay for only what they need when they need it, making LLMs more accessible to everyday users and providing a platform for others to build applications upon.

**SUMMARY OF THE INVENTION**

The present invention provides a novel solution to the current challenges in natural language processing by leveraging a distributed blockchain network and cryptocurrency. The invention aims to address the issue of limited local compute power and censorship by creating a decentralized network of incentivized nodes that securely distributes LLM processing. The invention also solves the problem of wasteful computation in traditional blockchain verification methods by providing a useful cause for the computation used in securing blockchains.

The main features of the invention include a distributed network of incentivized nodes, a novel verification algorithm, and reputation system. These features provide benefits such as increased accessibility, improved security and reliability, and cost savings for users. The core algorithm provides a means by which to quickly check the accuracy of computed LLM output without needing to completely compute the entire set of data again. This allows for a compute node to generate output and the client to trust it’s authenticity. Additionally this allows for a transfer of funds without having to trust the compute node nor the need to spend excess resources verifying the output data. This algorithm relies on unique, unintuitive properties unique to the way that LLM’s output data.

The invention is intended for use by a wide range of industries and users, including developers, researchers, and businesses that rely on natural language processing. Furthermore, the invention has the potential to drive exponential advancements in the field by leveraging the power of the distributed network to fund training and improve the model at a later time.

**DETAILED DESCRIPTION OF THE INVENTION**

The present invention is a system for secure and efficient distributed computing using blockchain technology. The system comprises a network of nodes, including clients, compute nodes, and a distributor node. Each node generates a 256-bit key pair. Compute nodes must find a SHA-1 hash of their public key starting with N zeros and register this hash with the distributor to gain a starting reputation number. This number can be decided at the time of implementation. This provides a one-time proof of work mechanism, which stops clients from spamming the network and helps reinforce the reputation system. Additionally, the compute nodes complete a certain number of “free” jobs in order to build a reputation. Compute nodes must maintain a minimum balance for use in staking.

A “Job” in the system references a series of tokens that need to be completed via computation by the compute nodes. The client signs and sends the job to the distributor along with an optional payment instruction. The distributor maintains a list of nodes with acceptable reputation who are open to work jobs. When a job is received, the distributor signs and forwards it to any available compute nodes. The compute nodes can decide whether to pick the job based on token length and payment.

The compute nodes then compute the output tokens based on the LLM used until a <end> token is reached and/or the specified number of tokens specified by the client. They then sign the output of the job and in doing so stake a number of tokens decided at implementation. The compute job is then sent back to the distributor. The distributor then offers an incentive, a small percentage of the payment from the client, to free compute nodes to run the verification algorithm. The verification cannot be done by the original compute node. Along with the completed job the distributor also picks 3 random numbers from 0 to N, where N is the length of the output, to send to with the output for verification.

The verification algorithm involves three integers, each representing the position of a token in the output, where the integer is how far from the beginning of the output the token is starting at 0. Next, the input and output up until that point are fed into the Language Model Machine (LLM), which completes one token at a time using the previous output as input. One token is generated and verified against the output from the compute node. If it is correct, the algorithm is repeated for the next two integers. Lastly, the last token is checked to ensure it outputs the "<end>" token, which ensures that the compute node did not truncate the output. If the output exceeded the token limit set by the client the N-1 token is checked instead to verify the last token. The compute node doing the verification then sends a signed message of approval back to the distributor. Otherwise, it signs a message of rejection. In case of approval from multiple nodes, the distributor forwards the output to the client node and releases most of the payment to the original compute node and the rest to the verifying node. In case of rejection, the reputation of the original compute node is decremented, and payment is not released. The original compute job is offered to another set of random compute nodes until the job is completed.

Compute nodes will have their reputation incremented when a proper job is submitted. Additionally, they will lose their staked tokens after the distributor determines the output is invalid. The payment is decided on by the client, and a higher payment might get their compute job answered quicker. The process for choosing compute nodes is partially randomized but also based on reputation and which compute jobs are accepting jobs. The three random integers are generated by a random number generator and used to verify a few tokens that rely on the rest of the message. If the few tokens are correct, statistically, the rest of the message must be. All preceding tokens must be generated to determine each token. The payment split can be determined by the implementer of the algorithm.

**BRIEF DESCRIPTION OF THE DRAWINGS**

1. Client sends job to distributor

2. Distributor sends job to free compute node

3. Compute node sends output back to distributor

4. Distributor sends output for verification to random free compute node

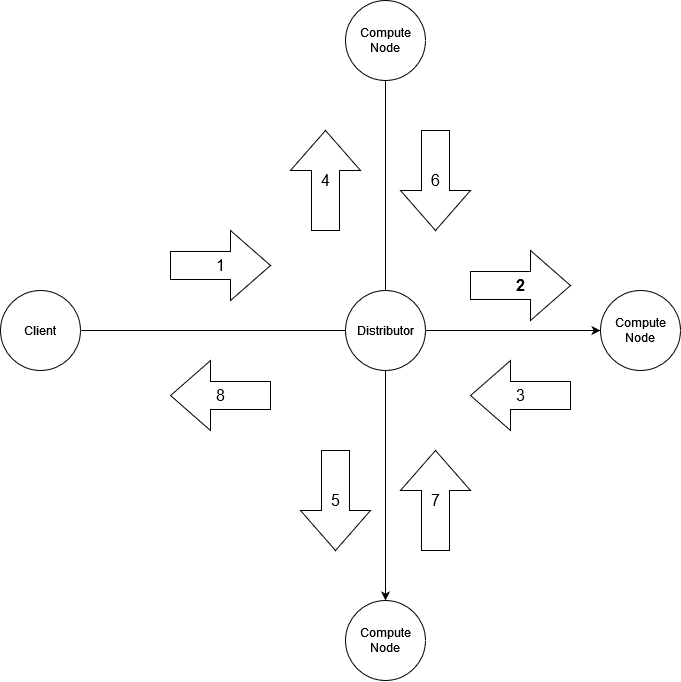
5. Distributor sends output for verification to another random free compute node

6. Verification is confirmed by compute node

7. Verification is confirmed by another compute node

8. Verified output is sent back to client

**CLAIMS**1. A decentralized computing system comprising a plurality of compute nodes, a distributor node, and a client node, wherein the compute nodes perform computation tasks and are incentivized through a staking mechanism and reputation system.  
2. The system of claim 1, wherein each compute node generates a 256- bit key pair and uses the public key to register with the distributor node, which assigns a starting reputation number to the compute node based on a proof-of-work algorithm.  
3. The system of claim 1, wherein the distributor node maintains a list of compute nodes with acceptable reputation who are available to work on computation tasks submitted by the client node.  
4. The system of claim 1, wherein the client node submits computation tasks to the distributor node, which signs and forwards the tasks to one or more compute nodes, and the compute nodes decide whether to accept the task based on payment and computational resources available.  
5. The system of claim 1, wherein the compute nodes perform the computation tasks and return the results to the distributor node, which verifies the correctness of the results using a verification algorithm and incentivizes other compute nodes to perform the verification.  
6. The system of claim 5, wherein the verification algorithm uses three random integers and a language model to verify a subset of tokens in the result, and the compute node performing the verification signs a message of approval or rejection based on the correctness of the result.  
7. The system of claim 5, wherein the distributor node releases payment to the compute node and verification node upon successful verification and decrements the reputation and staked tokens of the compute node upon unsuccessful verification.  
8. The system of claim 1, wherein the payment for computation tasks is determined by the client node and a higher payment results in faster completion of the task by the compute nodes.  
9. The system of claim 1, wherein the selection of compute nodes to perform computation tasks is partially randomized and based on reputation and availability of the compute nodes.  
10. The system of claim 1, wherein the language model used in the verification algorithm is any language model that completes one token at a time using the previous output as input.  
11. The system of claim 1, wherein the compute nodes use a staking mechanism to incentivize correct completion of computation tasks and the loss of staked tokens upon unsuccessful verification by the distributor node.  
12. A method of decentralized computation using a plurality of compute nodes, a distributor node, and a client node, comprising submitting computation tasks from the client node to the distributor node, forwarding the tasks to compute nodes with acceptable reputation, performing the computation tasks, verifying the correctness of the results, and incentivizing the compute nodes using a staking mechanism and reputation system.   
13. The method of claim 12, wherein the verification of the results uses three random integers and a language model to verify a subset of tokens in the result and incentivizes other compute nodes to perform the verification.  
14. The method of claim 12, wherein the payment for computation tasks is determined by the client node and a higher payment results in faster completion of the task by the compute nodes.  
15. The method of claim 12, wherein the selection of compute nodes to perform computation tasks is partially randomized and based on reputation and availability of the compute nodes.

****