UDC 004.75

ОЦІНКА ВИКОРИСТАННЯ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ У МЕРЕЖАХ БЛОКЧЕЙН: ПЕРЕХІД ДО НЕТРАДИЦІЙНИХ МЕТОДІВ

Хошаба Олександр Мирославович

Вінницький національний технічний університет, к.т.н., доцент кафедри програмного забезпечення, pzmag2022@gmail.com

ASSESSING THE UTILISATION OF COMPUTING RESOURCES IN BLOCKCHAIN NETWORKS: MOVING TO UNCONVENTIONAL METHODS

Khoshaba Oleksandr Myroslavovych

Vinnytsia National Technical University, Ph.D., Associate Professor, Department of Software, pzmag2022@gmail.com

The article analyses the inadequacy of traditional gas-based resource evaluation in blockchain networks, particularly in the context of computationally intensive applications such as tensor and matrix operations. A classification of computational loads and proposals for unconventional evaluation methods, utilising FLOPS, memory consumption, and execution time, are discussed.

Keywords: blockchain, resource evaluation, computational load, unconventional methods, FLOPS, matrix operations.

Blockchain systems traditionally use a gas mechanism that prioritises data storage costs while neglecting the resources consumed by data retrieval and computation. This one-dimensional

model is sufficient for simple cryptocurrency transactions but becomes inadequate for emerging applications that require intensive computation and data processing. Modern use cases such as on-chain machine learning, cryptographic proof generation, and decentralised scientific computing illustrate that computational processes (rather than data recording) now dominate resource consumption, revealing the shortcomings of current gas accounting.

Computational tasks in blockchain range from simple scalar operations to complex tensor operations. Scalar operations involve single numerical values and impose negligible computational load. In contrast, vector operations (one-dimensional arrays) and matrix operations (two-dimensional arrays) demand more memory access and can strain caching systems. Tensor operations generalise to multi-dimensional arrays and often require specialised hardware (like GPUs or TPUs) to manage their high parallelism and data throughput. Each type of operation has a distinct resource footprint, so a single storage-focused gas metric cannot account for all cases. This disparity underscores the need for a more sophisticated accounting mechanism that accurately captures the actual computational costs across diverse workloads.

Beyond basic operational counts, multiple performance metrics are necessary for a comprehensive assessment of computational resource utilisation. FLOPS (floating-point operations per second) measure raw arithmetic throughput but omit memory access patterns, communication overheads, and energy usage. Key complementary metrics include memory bandwidth utilisation (the data transfer rate between memory and processor), instruction throughput (the rate of executing all types of instructions), execution latency (delays resulting from communication or memory access), and cache performance (e.g., cache hit versus miss rates). For instance, a contract handling large data volumes may be bottlenecked by memory bandwidth despite high FLOPS, and complex control flows can reduce effective instruction throughput. Similarly, network or memory latency can undermine high theoretical throughput, and poor cache efficiency forces slower memory accesses. Incorporating these metrics into resource accounting would yield a more holistic

and fair evaluation of a contract's resource demands. In blockchain networks with varied workloads, such multidimensional accounting is crucial for performance, scalability, and fairness. By contrast, relying solely on the traditional gas model risks mispricing resource usage, leading to inefficiencies or unfair cost allocation.

The traditional gas model, designed for storage operations and simple computations, clearly fails to capture the costs of complex tasks. In response, three main directions have been proposed to improve resource accounting: extending the gas model to include computational costs; introducing an extended gas framework that separates storage and computation costs; and adopting hybrid Layer-2 models for resource accounting.

1. Extending the Gas Model to Account for Computational Costs. In the conventional gas scheme, transaction fees depend mainly on on-chain data storage and basic operations. Consequently, even compute-intensive tasks incur nearly the same fees as trivial ones, misaligning costs with actual resource use. To correct this, the gas model can be extended with direct measures of computational complexity. For example, a contract's fee could dynamically reflect the number of arithmetic operations, memory accesses, or execution time it requires. Incorporating such metrics would ensure that fees accurately represent the computational effort each transaction imposes on the network. This change promotes fairer cost distribution and incentivises developers to optimise smart contracts for computational efficiency, not just storage efficiency.

2. Implementing an Extended Gas Framework Separating Storage and Compute. A more structural improvement is to split resource accounting into separate Storage Gas and Compute Gas components. Storage Gas would quantify persistent storage usage, specifically the volume of data written, read, and stored on-chain. In contrast, Compute Gas would measure computational workload, such as the number of instructions executed or the time consumed for processing. By clearly delineating storage and computation costs, this model enhances transparency and provides fine-grained control over resource allocation. It also allows independent optimisation: storage-intensive applications can be tuned for data efficiency, while

compute-heavy ones focus on algorithmic efficiency. Furthermore, separate fee markets could emerge for storage and compute resources, letting users bid on each according to demand and achieving more efficient overall resource allocation.

3. Utilising Hybrid Models in Layer-2 Solutions. Layer-2 scaling solutions enable a hybrid resource model, where heavy computations are executed off-chain (in sidechains or semi-trusted environments), and only succinct proofs of their results are recorded on-chain. For example, an off-chain service might perform complex calculations and publish a short proof (using zk-SNARK or zk-STARK techniques) that the blockchain can quickly verify. This way, the bulk of computation is handled off-chain while the main chain performs minimal verification and storage, preserving integrity. Hybrid accounting combines off-chain computation metrics (enforced by these proofs) with on-chain metrics for verification and storage. Transaction fees reflect the proportion of the workload that is borne off-chain versus on-chain. Such Layer-2 approaches enable blockchain networks to significantly expand their computational throughput without compromising fairness or security in resource allocation.

In conclusion, incorporating extended gas models and hybrid Layer-2 approaches would be a significant step toward a scalable, economically sustainable blockchain infrastructure. Aligning fees more closely with actual resource usage will better support the diverse, computation-intensive applications emerging in decentralised finance, on-chain machine learning, and scientific computing.

References

1. Gudgeon, L., Perez, D., Harz, D., Livshits, B., & Gervais, A. (2020). The decentralized financial crisis: Attacking DeFi. Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 265–282. https://doi.org/10.1145/3372297.3423369

2. Xie, T., Cui, L., Zhou, Z., & Xu, K. (2020). A survey on machine learning in blockchain: Fundamentals, applications, and future

directions. IEEE Transactions on Knowledge and Data Engineering. https://doi.org/10.1109/TKDE.2020.3045994
3. Hassan, S., De Filippi, P., & Reijers, W. (2020). The expansion of algorithmic governance: From code is law to law is code. Philosophy & Technology, 34, 1–24. https://doi.org/10.1007/s13347-020-00418-5
4. Beller, T., Eberhardt, J., & Tai, S. (2021). Trustless machine learning contracts: Evaluating the intersection of blockchain and machine learning. Future Generation Computer Systems, 118, 174–187. https://doi.org/10.1016/j.future.2020.12.029
5. Nguyen, G. T., & Kim, K. (2022). Survey on blockchain and its impact on distributed machine learning. Electronics, 11(3), 410. https://doi.org/10.3390/electronics11030410