Towards a Blockchain Database for Massive IoT Workloads

Panagiotis Drakatos, Erodotos Demetriou, Stavroulla Koumou,
Andreas Konstantinidis*, Demetrios Zeinalipour-Yazti

Department of Computer Science, University of Cyprus, 1678 Nicosia, Cyprus
* Department of Computer Science, Frederick University, 1036 Nicosia, Cyprus
{pdraka01, edemet01, skoumo01, akonstan, dzeina}@cs.ucy.ac.cy

***Abstract*—The Internet of Things (IoT) revolution has massively introduced sensor-rich devices to an ever growing landscape of smart environments. A key component in the IoT scenarios of the future is the requirement to utilize a shared database that allows all participants to operate collaboratively, transparently, immutably, correctly and with performance guarantees. Blockchain databases have been proposed by the community to alleviate these challenges, however existing blockchain architectures suffer from performance issues. In this vision paper we propose Triabase, a novel permissioned blockchain database system that carries out machine learning on the edge, abstracts machine learning models into primitive data blocks that are subsequently stored and retrieved from the blockchain. As such, it does not store detailed records on a medium, like blockchains, which is fundamentally very slow due to the expensive verification process. We lay out the primitive architectural blocks of our design, the requirements and the inherent challenges. Triabase employs technical novelties in respect to its consensus protocol, namely the notion of Proof-of-Federated-Learning (PoFL). The Triabase prototype system is implemented in the Hyperledger Fabric blockchain framework, upon which encouraging preliminary findings have been drawn.**

***Index Terms*—blockchain, IoT, federated-learning, databases.**

I. Introduction

Internet of Things (IoT) refers to a large number of physical devices being connected to the Internet that are able to see, hear, think, perform tasks, as well as communicate with each other using open protocols [1]–[4]. IoT devices are connected to Cloud and Edge computing appliances through massively parallel I/O channels (e.g., 5G, WiFi 6) with milliseconds latency offering new opportunities in industrial optimization, human health and well-being as well as safety. In absolute numbers, the IoT revolution is expected to bring the number of such devices close to a staggering 40 billion in 2020, more than double from 2019 [5]. This will procreate tremendous opportunities for IoT applications between multiple parties, such as collaborative multitasking techniques [6], machine learning [7], cooperative benchmarking [8], and augmented reality technology [9].

A key component in the IoT scenarios of the future is the requirement to utilize a shared database that allows all participants to operate collaboratively with more functionality. The shared database can bridge the actual gap between the data generated from the IoT applications [10] and the rate that these are processed and analyzed in real-time. The objective is to enable users execute updates and queries on the collaborative database while preserving a consistent view among all users maintaining the system consistency and transparency. Moreover, it is essentially common to be compromised by malicious outsources. An innovative design of a shared database with high performance, it is therefore required for all the participants, in order to collaborate among each other with trust. Blockchain databases have been recently proposed by the community to alleviate these challenges. However, existing blockchain architectures suffer from performance issues measured in terms of throughput and latency, mainly because the transactions are executed in a sequential manner. The latter in conjunction with confidentiality issues, does not leave much space for scaling.

It is imperative to devise a database architecture that can withstand billions of transactions per second, as opposed to thousands transactions per second that is currently the case for typical blockchains due to the expensive verification cost.

In this paper, we propose *Triabase* (inspired from Greek "Tria", meaning "three", being a Database for the Web 3.0 era), a permissioned blockchain database system that carries out machine learning at the edge, abstracts machine learning in primitive blocks that are subsequently stored and retrieved from the blockchain. In Triabase, we have two types of nodes those that store the entire shared database, and the others that use the database for their own operations, such as sending query and update requests to the blockchain shared ledger. We expect the blockchain nodes to be synchronized under the decentralized blockchain network. The clients that utilize blockchain for database operations will store the appropriate block header only, as opposed to the full nodes that will store the entire blockchain ledger.

For this purpose, the key challenge is to find a robust design that is able to: *i) Execute machine learning algorithms at the edge; ii) Operate on a distributed environment; and iii) Mitigate issues related to data privacy protection.* Our main goal is to guarantee the following aspects:

- **Immutability:** We want to ensure that any update committed to the blockchain is immutable and will not be tampered by any malicious node;

- **Transparency:** We oblige the shared database to strictly update according to the committed transactions. All database operations e.g., insertions, deletions, updates are transparent to nodes because users are able to get all historical data of the transactions committed on the blockchain, at any time;
- **Correctness:** Performing all the required operations with minimal computational requirements and without the excessive energy consumption, when a client receives a query and/or results from a server node;
- **Performance:** Our system must support a wide range of queries and indexes. As a result, we should allow Triabase to achieve better performance to scale; and thus improve the overall throughput of the network in order to minimize any unnecessary overhead that causes latency;
- **Privacy:** Centralized artificial intelligence algorithms demand from the clients to provide whole trained models, which incurs high data leakage risks, something that must be taken into account by Triabase.

To enhance user security and privacy in Triabase, we propose federating learning [11] that is a new wise choice for distributed machine learning. Federating learning differs from the traditional artificial intelligence algorithms, since it trains a global federated learning model at the server side, by using only appropriate parameters from the locally trained models, keeping the full amount of data at the user endpoint, mitigating in this way several security and privacy risks.

In order to test the validity of the system, we have implemented an initial version of our architecture using the hyperledger fabric technology, which enables us to measure the latency, as well as the throughput of different parts of our implementation during the ingestion load and during the searching query process. Our preliminary results are very encouraging as they reveal that in our proposed architecture, the tradeoff between the learning accuracy and the efficiency of the trained models from the federated learning approach achieve comparable results.

The main contributions of our vision paper are as follows:

- We introduce Triabase, a permissioned blockchain database system enhanced with federating learning approach, which contains the running states and behavior models of the blockchain nodes to ensure the security and data privacy of users;
- We propose a new consensus empowering collaborative mechanism, namely Proof of Federated learning (PoFL), to share parameters over distributed multiple parties to reduce the risk of data leakage and to protect federated nodes from being tampered;
- We also implement our proposal with the integration of the fabric open-source platform to provide a more realistic blockchain scenario.

The rest of the paper is organized as follows. Related work is presented in Section II while Section III presents the proposed system architecture. Finally, Section IV discusses the current limitations of the proposed system and future work.

II. Background and Related Work

Blockchain architecture is mainly used to keep records on an immutable chain of blocks, where nodes agree on the shared state across a network of untrusted participants. This forms the blockchain platform that can be viewed as a distributed (transaction-log or) database system. The blocks are agreed by the majority of validators according to the consensus protocol that tolerates Byzantine faults. The most well-known platforms include Capera [12], Hyperledger [13], Monoxide [14]. This architecture does not require a centralized server and operates in untrusted environments of arbitrary nodes.

The authors of [12] introduce a system named Caper, a permission blockchain architecture based on an acyclic graph and three consensus protocols to support internal and cross-application transactions. Moreover, [15] introduces a novel framework, called vChain, which is able to improve the storage and computing costs of the user and employs verifiable queries to ensure the system integrity.

Artificial intelligence along with the integration of blockchain technology is a great promise to solve various resource optimization problems. For instance, the merit of the two technologies is proposed in [16] providing a secure resource sharing scheme by developing a caching mechanism with the usage of DRL. Reyna et al. [17] introduced how blockchain may potentially improve IoT environments and how blockchain can overcome IoT security challenges. However, AI algorithms, which are vulnerable to security threats, depend much on centralization approaches, a fact that has a negative impact on improving efficiency, because it consumes a large number of communication resources.

Moreover in the literature, several research studies aim at improving the scalability and performance of blockchain networks. Algorand [18] and RandHound [19] achieve high scalability by randomly selecting a subset of validators to participate in the consensus, while maintaining the security level. The study in [20] use directed acyclic graphs, instead of a blockchain structure, to reduce the average amount of time for each transaction. Blockbench [21] proposed the permissioned blockchain, an approach for comparing the performance of different platforms including Ethereum Parity, and Hyperledger Fabric by using a set of micro and macro benchmarks. Furthermore, [13] introduces the architecture of fabcoin, which presents the performance of bitcoin in the fabric network.

III. The TriaBase architecture

In this section, we introduce the TriaBase system architecture in a bottom-up manner.

A. Blockchain Layer

The Bitcoin protocol uses a PoW (Proof-of-Work) consensus mechanism to validate users' transactions in the blockchain. PoW, however, is a high-cost algorithm that leads to excessive energy consumption. Therefore, a low-cost consensus mechanism is required for the Triabase architecture that will provide security with respect to Sybil attacks, at the

same time. In this paper, we propose the *Proof-of-Federated-Learning process (PoFL)*, which brings new technological advantages with respect to user security level.

TriaBase blockchain nodes maintain a separate permission blockchain, where the local models of federated learning are stored in the blockchain distributed ledger. Hence, we detach two types of records: the training model records and the IoT dataset records. We distinguish only the first record in the block that contains the aggregate model for the r round difficulty. We consider every block generation as a separate round of the federated learning process, where clients use the shared model from block $b-1$ and their existing local models to generate the next round local models.

The process starts with training local models by using local data at the user side. Then, the communication process takes place where all users broadcast and upload the trained models to the blockchain nodes and store them as transactions to the distributed ledger. The blockchain node that was the winner from the previous round (depends on the blockchain difficulty) is responsible for initiating the 2-step consensus protocol and constructs the blocks with all the cached transactions that are not yet validated.

In addition, the winner node is in charge for aggregating the users local models and producing the shared model. Then the shared model is added as the first transaction in the block in order for the federated learning nodes to access it, in the next round $r+1$. Our PoFL consensus protocol contemplates that users who participate in the blockchain process will be rewarded with training coins. Users are awarded coins according to their performance in the training process, that is whether the federated nodes training algorithm converges faster and achieves higher accuracy. The winner node of each round r is the one that achieves the highest accuracy, considering the difficulty of the block. Furthermore, in every training round the coins will be adjusted to the users depending on their contribution.

Nevertheless, to secure our protocol and to ensure that every user will obey the protocol we introduce a new hierarchy of nodes simply called peacemaker entity. The peacemaker's entity is responsible to observe the correctness of the protocol followed by all the federating nodes. For example, users that refuse to cooperate with the protocol will get no payment for their work. Moreover, users that will try to get more rewards and try to counterfeit the correctness of the whole process will get a punishment by the peacemaker entity. The peacemaker will then claim the adjusted coins as their own reward for their effort in the protocol correctness.

B. Storage Layer

This layer is responsible to store the incoming data in the Triabase blockchain, in an online or offline mode, and to provide access methods to various input sources. Furthermore, the incoming data will be aggregated and used by the federated learning nodes to replicate the information and therefore provide failure recovery, as well as, availability and increased performance. This component is inspired by our earlier Spate [22] and the work on data postdiction.

The federated learning process aims to train a global shared model by aggregating local models from all clients, and consequently allow each client to maintain its own data locally. We follow the federated averaging model introduced in [23], in which a fixed set of N clients, each one holding n_k data points and compute the average gradient on their local data by having a current model w_t with a fixed learning rate η $g_k = \nabla F_k(W_t)$, where $F_k(W_t) = \frac{1}{n_k}\sum_{i \in P_k} fi(W_t)$ is the loss equation of the prediction on local model. Hence, each federated client can update its local model as $W_{t+1} = W_t - \eta g_k$ while the server is responsible to aggregate the received local model of all clients as $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} W_{t+1}$.

The primary objectives of Triabase are to protect the local data from any external entity by not allowing them to be extracted from the devices, and protecting the local data from the server by never leaking private information of the local model while updating the global model at the server side. In addition, our research goals are to provide security to the users and ensure confidentiality and integrity of the local models from malicious nodes that will try to counterfeit the protocol.

C. Query/Indexing Layer

The main purpose of this layer is to minimize the query response time that requires storage space with efficient algorithms and with minimal overhead. We plan to use compression techniques to achieve more high compression ratios. In addition, we are going to use a specific type of queries in the Triabase architecture to optimize the response time. More specifically the query types supported by Triabase include *standard queries*, where only the newest database version is queried, *full historical queries* on a particular predicate, *range historical queries* on all updates in a specific time range, and *delta query* that helps the users to query the changes made by the transactions committed at any particular block.

Indexing is an essential task that intends to increase execution queries' performance. Therefore in TriaBase architecture, we need to index machine learning models according to their temporal and spatial characteristics. This is important since different models might refer to data from distinct time frames and geographical locations. To eliminate this problem and optimize the performance of the queries, we introduce an unclustered B+ Tree index. Specifically, when a new block arrives at a client, the B+ Tree index is locally updated according to the new models. This temporal index can be divided into multiple levels, according to our needs. The tree lower layer (leaves) holds the pointers to specific models stored throughout the Blockchain. Using this simple data structure, we don't need to scan the whole database to find a specific model. Instead, we use the index to navigate to a specific block and thus we improve the asymptotic complexity to logarithmic time.

IV. Limitations and Future Work

Privacy is critical and needs to be carefully handled in IoT environments. Location data is inherently highly sensitive data because it is easy to extract user activities from trajectories. To tackle these issues, we aim at investigating techniques and algorithms to achieve strong privacy guarantees. For example, zk-Snarks [24] can be utilized to protect the privacy of our infrastructure. Furthermore, we plan to use bloom filters [25], which offer an efficient way to describe a search pattern without specifying and revealing information for the entities and thus enhance the privacy of our system.

In addition, a key challenge is how to achieve incremental scalability to improve the performance of TriaBase and make our approach more efficient. To protect the data integrity of transactions in the network we are going to use binary hash trees, also known as merkle trees, that constitute data structures to efficiently verify the integrity of large datasets. Additionally, we aim at using only byte hashes to construct a merkle path from the root to reach a specific transaction. Finally, theoretical analysis and empirical evaluation will be conducted to validate the performance of our propositions.

References

[1] L. Yao, Q. Z. Sheng, and S. Dustdar, "Web-based management of the internet of things," *IEEE Internet Computing*, vol. 19, no. 4, pp. 60–67, 2015.

[2] A. A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Comm. Surv. Tutor., vol. 17, no. 4, pp. 2347–2376*, 2015.

[3] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[4] L. Atzori, A. Iera, and G. Morabito., "The internet of things: A survey," *Comput. Netw., vol. 54, iss. 15, pp. 2787–2805*, 2010.

[5] Juniper Research, "IoT connected devices to almost triple to over 38 billion units by 2020," 2019. [Online]. Available: https://tinyurl.com/juniperresearchIoT

[6] Wenliang Du and M. J. Atallah, "Privacy-preserving cooperative scientific computations," in *Proceedings. 14th IEEE Comp Security Foundations Works., 2001.*, Jun. 2001, pp. 273–282, iSSN: 1063-6900.

[7] D. Billsus and M. J. Pazzani, "Learning Collaborative Information Filters," in *Proceedings of the Fifteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., Jul. 1998, pp. 46–54.

[8] M. Atallah, M. Bykova, J. Li, K. Frikken, and M. Topkara, "Private collaborative forecasting and benchmarking," in *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*. Association for Computing Machinery, Oct. 2004, pp. 103–114. [Online]. Available: https://doi.org/10.1145/1029179.1029204

[9] J. Li, C. Wang, X. Kang, and Q. Zhao, "Camera localization for augmented reality and indoor positioning: a vision-based 3D feature database approach," *International Journal of Digital Earth*, vol. 13, no. 6, pp. 727–741, Jun. 2020, publisher: Taylor & Francis. [Online]. Available: https://www.tandfonline.com/doi/10.1080/17538947.2018.1564379

[10] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349–359, Aug. 2014, conference Name: IEEE Internet of Things Journal.

[11] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated Multi-Task Learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] M. J. Amiri, D. Agrawal, and A. E. Abbadi, "CAPER: a cross-application permissioned blockchain," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1385–1398, Jul. 2019. [Online]. Available: https://doi.org/10.14778/3342263.3342275

[13] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. Association for Computing Machinery, Apr. 2018, pp. 1–15. [Online]. Available: https://doi.org/10.1145/3190508.3190538

[14] J. Wang and H. Wang, "Monoxide: Scale out Blockchains with Asynchronous Consensus Zones," 2019, pp. 95–112. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/wang-jiaping

[15] C. Xu, C. Zhang, and J. Xu, "vChain: Enabling Verifiable Boolean Range Queries over Blockchain Databases," in *Proceedings of the 2019 International Conference on Management of Data*. Association for Computing Machinery, Jun. 2019, pp. 141–158. [Online]. Available: https://doi.org/10.1145/3299869.3300083

[16] Y. Dai, D. Xu, S. Maharjan, Z. Chen, Q. He, and Y. Zhang, "Blockchain and Deep Reinforcement Learning Empowered Intelligent 5G Beyond," *IEEE Network*, vol. 33, no. 3, pp. 10–17, May 2019.

[17] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, Nov. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17329205

[18] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling Byzantine Agreements for Cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*. Association for Computing Machinery, Oct. 2017, pp. 51–68. [Online]. Available: https://doi.org/10.1145/3132747.3132757

[19] E. Syta, P. Jovanovic, E. K. Kogias, N. Gailly, L. Gasser, I. Khoffi, M. J. Fischer, and B. Ford, "Scalable Bias-Resistant Distributed Randomness," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 444–460, iSSN: 2375-1207.

[20] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, "Scaling Nakamoto Consensus to Thousands of Transactions per Second," *arXiv:1805.03870 [cs]*, Aug. 2018, arXiv: 1805.03870. [Online]. Available: http://arxiv.org/abs/1805.03870

[21] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "BLOCKBENCH: A Framework for Analyzing Private Blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*. Association for Computing Machinery, May 2017, pp. 1085–1100. [Online]. Available: https://doi.org/10.1145/3035918.3064033

[22] C. Costa, G. Chatzimilioudis, D. Zeinalipour-Yazti, and M. F. Mokbel, "Efficient Exploration of Telco Big Data with Compression and Decaying," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, Apr. 2017, pp. 1332–1343, iSSN: 2375-026X.

[23] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Artificial Intelligence and Statistics*. PMLR, Apr. 2017, pp. 1273–1282, iSSN: 2640-3498. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[24] M. Petkus, "Why and How zk-SNARK Works," *arXiv:1906.07221 [cs, math]*, Jun. 2019, arXiv: 1906.07221. [Online]. Available: http://arxiv.org/abs/1906.07221

[25] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters: A Survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, Jan. 2004. [Online]. Available: https://doi.org/10.1080/15427951.2004.10129096