An Efficient Design and Implementation of Blockchain Architecture

Final report for the OUR project

Anish Gupta(Student-2nd year)
Btech-Computer Science and Engineering
SoE, Shiv Nadar University
2310110047
ag801@snu.edu.in

Dr. Sweta Kumari(Faculty Advisor)
Assistant professor
Dept-Computer Science and Engineering
Shiv Nadar University
sweta.kumari@snu.edu.in

***Abstract*—Blockchain technology has revolutionized how we manage secure, decentralized transactions across various industries, from finance to supply chain management. However, traditional blockchains, which use basic structures like linked lists (used in bitcoin [12]) and Merkle trees (used in hyperledger [11]), face issues with transaction speed, scalability, and energy efficiency. Directed Acyclic Graph–based blockchains mark a shift from these designs, enabling significantly higher throughput through concurrent storage and execution. In our work, we refine MorphDAG [10]'s transaction classification by replacing its coarse operation-count threshold with a per-account read/write frequency analysis. Transactions touching only low-frequency ("cold/read dominated") accounts are executed fully in parallel , while those involving high-frequency ("hot/write dominated") accounts incur minimal dependency checks. Integrated into the MorphDAG prototype and tested on a 5 iterations of the test applications with a randomly generated set of transactions based on synthethic workload of smallbank benchmark, this refinement yields a 12% reduction in block-processing latency (819ms→722ms) and a 13% increase in throughput, with no data inconsistencies over a variable number of transactions.**

I. Introduction

Blockchain technology underpins a wide range of decentralized applications (dApps), from permissionless cryptocurrencies to permissioned ledgers for supply chain and healthcare. Its core innovations are immutability, transparency, and decentralized consensus derive from chaining cryptographically linked blocks, as in Bitcoin's linked-list structure [12], or Merkle tree based designs such as Hyperledger Fabric [11]. However, these traditional architectures suffer from inherent limitations:

- **Throughput bottlenecks**: Sequential block creation and strict ordering limit transactions per second (TPS).
- **Energy inefficiency**: Proof of Work (PoW) mining consumes vast electricity, even in permissioned contexts.
- **Poor scalability**: Network and storage overheads grow linearly with chain length and transaction volume.

Directed Acyclic Graph (DAG) based blockchains address these concerns by allowing multiple blocks (or "events") to be appended concurrently, yielding higher parallelism and throughput. Notable DAG systems include IOTA's Tangle, Nano's block-lattice, and hybrid approaches such as LightDAG [14] and MorphDAG [10]. In particular, MorphDAG combines an *elastic storage concurrency* model dynamically tuning the number of parallel block producers via PoS-based VRF sortition with a dual-mode transaction processor as made by MorphDAG developers that handles "hot" and "cold" data differently to mitigate conflicts under skewed workloads.

In its original form, (old) MorphDAG classifies transactions as hot or cold based on a simple threshold of total read/write operations per transaction. While this approach improves over uniform execution, Transactions with both reads and writes can end up on the "hot" path simply because their total operation count is high even if they touch only rarely used accounts and would never actually conflict. This causes them to be serialized unnecessarily. Our per-account frequency approach fixes this by tagging a transaction as "hot" only if it really uses a busy (high-frequency) account, and otherwise running it fully in parallel., we propose a *per-account read/write frequency* refinement:

- We compute `writeFreq` and `readFreq` for each account over the current block's transaction set.
- We designate high-frequency accounts (*hot*) and low-frequency accounts (*cold*) via a tunable percentile threshold.
- Transactions that touch only cold accounts (read-dominated) execute fully in parallel ; those involving any hot accounts incur lightweight dependency checks.

We integrate this refinement into the MorphDAG prototype and evaluate it using a Smallbank synthetic benchmark tested on a local machine, Across five iterations and a variable set of transactions, our method achieves a **12% reduction in block-processing latency** (819ms→722ms) and a **13% increase in throughput**, with no observed state inconsistencies. These results demonstrate that per-account frequency based classification can further unlock the parallelism inherent in DAG-based blockchains while preserving security and correctness.

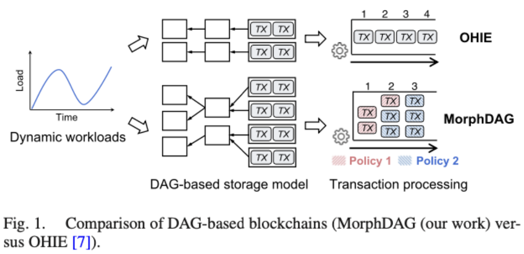

Fig. 1. Your caption here.

II. Literature Review

Here are a few things I learnt from online resources and research papers available in this field (briefing some of them) to explore more about this topic :-

A. Basic Blockchain terminologies and concepts

Before diving into the world of Blockchain I had the opportunity to learn about basic blockchain working, terminologies and what are all the types of blockchain available in the market.

B. Research Paper-Implementation of directed acyclic graph in blockchain network to improve security and speed of transactions

This paper explores how DAGs type structures improve blockchain networks by enhancing transaction speed and security. Unlike traditional blockchains, they allow parallel transaction processing, reducing delays and fees. This structure eliminates mining, making transactions faster, scalable, and fee-less, ideal for IoT and financial applications.

C. Research Paper-A Comparative Analysis of DAG-based Blockchain Architectures

Since we knew how DAG is very promising from previous paper, this one analyzes DAG-based blockchains like IOTA, Nano, and Byteball, highlighting their scalability, speed, and efficiency over traditional blockchains. It compares their consensus mechanisms, transaction validation, and security to identify best practices and proposes an optimized hybrid DAG model for improved performance.

D. The Merkle Tree structure

A Merkle tree is a data structure used in blockchains to organize and verify transaction data efficiently. it follows a binary tree format, where each leaf node contains a cryptographic hash of a transaction, and each non-leaf node stores the hash of its two child nodes. this process continues recursively until a single hash, known as the Merkle root, is formed at the top. The Merkle root is stored in the block header, allowing efficient verification of transaction integrity without requiring the entire dataset. bitcoin and ethereum use this structure to secure transactions.

E. Why DAG, and not Merkle tree

- **Higher scalability** – DAG enable parallel transaction processing, whereas merkle trees follow a sequential validation approach.
- **Greater throughput**– DAG efficiently process high transaction volumes, while merkle trees may create bottlenecks in large networks.
- **Faster confirmations** – Transactions in a DAG validate each other, reducing delays compared to merkle tree-based proof-of-work systems.
- **No reliance on mining** – Some DAG-based networks remove the need for mining, making them more energy-efficient.

F. Research Paper-Jointgraph: A DAG-based efficient consensus algorithm for consortium blockchains

This paper helped us to understand and learn how combining consensus with DAG strucutre can be a optimal solution as unlike traditional blockchains, JointGraph allows parallel transaction processing, reducing confirmation delays and improving efficiency. It addresses Byzantine faults by isolating malicious nodes and optimizes memory through periodic snapshots.

G. Research paper- Tangle the Blockchain:Towards Connecting Blockchain and DAG

Tangle is a hybrid Blockchain-DAG integration to improve scalability and efficiency, particularly for IoT. It introduced a connector component that facilitates seamless interaction, using Blockchain for data storage and Tangle for fast, scalable transactions.This system enhances flexibility, reliability in decentralized networks.

H. Research paper- MorphDAG: A Workload-Aware Elastic DAG-Based Blockchain

This is the technology I am currently exploring. I am breifly explaining it here:-

- MorphDAG is a workload-aware DAG-based blockchain designed to improve scalability and storage efficiency.
- It dynamically adjusts its DAG structure based on transaction patterns, optimizing performance for hotspot and cold transactions.
- By implementing elastic storage techniques, MorphDAG reduces memory usage while maintaining fast consensus.
- The system is tested under real-world workloads, demonstrating high throughput, low latency, and adaptive resource allocation, making it well-suited for large-scale decentralized applications.

III. Proposed Methodology/future work

We plan to work on the structural code of MorphDAG and propose a new blockchain structure which would have some additional features which could potentially increase throughput and transaction speed as MorphDAG as workload aware, the new structure could give very promising results. We plan to :-

TABLE I
DIFFERENT DAG TECHNOLOGIES

Comparing Factor	DAG Technologies		
	MorphDAG	***Tangle***	***JointGraph***
Scalability	High (Dynamically adjusts DAG structure)	High, but may suffer from orphaned transactions	Moderate (designed for consortium blockchains)
Transaction Speed	Very High (adaptive transaction processing)	Fast, but can slow down in low-participation scenarios	High (but limited by supervisory node bottlenecks)
Storage Efficiency	Elastic storage reduces redundant data	Prone to unbounded DAG growth	Snapshots reduce memory load
Latency	Lowest (adaptive consensus process)	Low, but varies based on network activity	Low (due to direct voting mechanism)
Security Model	Stronger (adaptive conflict resolution)	Secure, but susceptible to network attacks	Strong (malicious nodes removed)
Ideal Use Case	Large-scale decentralized applications	IoT and microtransactions	Consortium blockchains

- To work on the code so that it can **allow cyclic dependencies** in isolated segments of the DAG for temporary local consensus which could increase speed
- Divide the DAG into priority zones like **High-priority zones** process time-sensitive transactions (e.g., real-time payments) and **Low-priority zones** handle bulk, non-urgent transactions (e.g., file storage).
- Allow nodes to create additional dynamic edges between non-parent events based on: **Similarity** (e.g., same sender, timestamp proximity), **Dependencies** (e.g., transactions involving the same assets or participants).
- Optimize the DAG structure for energy-efficient validation so, Nodes calculate the energy cost of validating specific transactions and **prioritize low-cost paths**. Transactions with higher costs are deferred or aggregated for batch processing.
- Also plan on checking that which blockchain **consensus algorithm** can result in highest throughput not compromosing the basic security of the system
- I am attaching the MorphDAG code link here for reference [https://github.com/CGCL-codes/MorphDAG/blob/main/README.md]

Some existing plans might be modified/ more plans be added depending on our understanding of this code

IV. CONCLUSION

The DAG blockchain structure shows a lot of promise and once we have completed the code modification and testing under real-life workloads like Etherium we would be able to present a blockchain structure which would result in a higher throughput and wont compromise safety too. For now we continue to work on the code of MorphDAG which is a latest technology we found and has a lot of caliber.

ACKNOWLEDGMENT

I Would like to thank Dr Sweta Kumari (my faculty Advisor) for letting me work on this project, I am learning a lot from this opportunity and hope to continue doing so, I would also like to thank the OUR team for giving me the wonderful opportunity to take up this research in my second year.

REFERENCES

[1] P. S. Anjana, S. Kumari, S. Peri, S. Rathor and A. Somani, ”An Efficient Framework for Optimistic Concurrent Execution of Smart Contracts,” 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Pavia, Italy, 2019, pp. 83-92

[2] Fu, Xiang et al. “Jointgraph: A DAG-based efficient consensus algorithm for consortium blockchains.” Software: Practice and Experience 51 (2019): 1987 - 1999.

[3] H. Hellani, L. Sliman, A. E. Samhat and E. Exposito, ”Tangle the Blockchain:Towards Connecting Blockchain and DAG,” 2021 IEEE 30th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Bayonne, France, 2021, pp. 63-68

[4] H. Pervez, M. Muneeb, M. U. Irfan and I. U. Haq, ”A Comparative Analysis of DAG-Based Blockchain Architectures,” 2018 12th International Conference on Open Source Systems and Technologies (ICOSST), Lahore, Pakistan, 2018, pp. 27-34

[5] F. M. Benčić and I. Podnar Žarko, ”Distributed Ledger Technology: Blockchain Compared to Directed Acyclic Graph,” 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2018, pp. 1569-1570

[6] Kotilevets, I. D. et al. “Implementation of directed acyclic graph in blockchain network to improve security and speed of transactions.” IFAC-PapersOnLine 51 (2018): 693-696.

[7] Tokhmetov, A., Lee, V. ., and Tanchenko, L. . (2023). DEVELOPMENT OF DAG BLOCKCHAIN MODEL. Scientific Journal of Astana IT University, 16(16).

[8] Qin Wang, Jiangshan Yu, Shiping Chen, and Yang Xiang. 2023. SoK: DAG-based Blockchain Systems. ACM Comput. Surv. 55, 12, Article 261 (December 2023), 38 pages.

[9] Dai, Xiaohai and Wang, Guanxiong and Xiao, Jiang and Guo, Zhengxuan and Hao, Rui and Xie, Xia and Jin, Hai. (2024). LightDAG: A Low-latency DAG-based BFT Consensus through Lightweight Broadcast. 998-1008. 10.1109/IPDPS57955.2024.00093.

[10] S. Zhang et al., ”MorphDAG: A Workload-Aware Elastic DAG-Based Blockchain,” in IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 10, pp. 5249-5264, Oct. 2024

[11] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Weed Cocco, and Jason Yellick. 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference (EuroSys ’18). Association for Computing Machinery, New York, NY, USA, Article 30, 1–15. https://doi.org/10.1145/3190508.3190538

[12] (2009). Bitcoin: A Peer-to-Peer Electronic Cash System.

[13] Wang, Tianyu, et al. ”Understanding characteristics and system implications of DAG-based blockchain in IoT environments.” IEEE Internet of Things Journal 9.16 (2021): 14478-14489.

[14] Park, Seongjoon, Seounghwan Oh, and Hwangnam Kim. ”Performance analysis of DAG-based cryptocurrency.” In 2019 IEEE International Conference on Communications workshops (ICC workshops), pp. 1-6. IEEE, 2019.

[15] Yang, Wenhui, Xiaohai Dai, Jiang Xiao, and Hai Jin. ”LDV: A lightweight DAG-based blockchain for vehicular social networks.” IEEE Transactions on Vehicular Technology 69, no. 6 (2020): 5749-5759.