Improving the Performance of Proof-of-Space in Blockchain Systems

Varvara Bondarenko[1], Renato Diaz[2], Lan Nguyen[1] (*advisor*), Ioan Raicu[1] (*advisor*)
[1]Illinois Institute of Technology, Chicago, IL, USA
[2]University of Central Florida, Orlando, FL, USA
{vbondarenko, lnguyen18}@hawk.iit.edu, renato.diaz@ucf.edu, iraicu@iit.edu

Abstract

Blockchain technologies enable the success of digital currencies by providing security, decentralization, and trustless operation. Two dominant consensus algorithms, Bitcoin's Proof-of-Work and Ethereum's Proof-of-Stake, balance security, scalability, and energy efficiency, though PoW is energy-intensive and PoS faces centralization risks. Chia's Proof-of-Space offers a middle-ground, using storage (instead of computation) for validation in the network while maintaining decentralization. PoSpace turns the compute-intensive problem into a data-intensive known as plotting. However, Chia's plotting process stresses hardware, requiring expensive setups and shortening the lifespan of solid-state drives. This work takes a clean-slate approach to implementing an efficient PoSpace system that is lightweight and operates on small nodes (e.g. Raspberry Pis with 4-cores & 2GB RAM) to large systems (HPC server with 192-cores, 768GB RAM, & multiple NVMe storage devices). Our C and Rust implementations achieve significantly higher performance than Chia in plot generation and lookup efficiency across all system sizes.

CCS Concepts

• **Software and its engineering** → *Software design engineering*.

Keywords

blockchain, proof-of-space, multi-threading, high-performance computing, memory caching & compression, I/O, storage

ACM Reference Format:
Varvara Bondarenko[1], Renato Diaz[2], Lan Nguyen[1] (*advisor*), Ioan Raicu[1] (*advisor*), [1]Illinois Institute of Technology, Chicago, IL, USA, [2]University of Central Florida, Orlando, FL, USA, {vbondarenko, lnguyen18}@hawk.iit.edu, renato.diaz@ucf.edu, iraicu@iit.edu. 2024. Improving the Performance of Proof-of-Space in Blockchain Systems. In *Proceedings of The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC '24)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

SC '24, November 17–22, 2024, Atlanta, GA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Bitcoin (BTC) [7] introduced the PoW consensus algorithm, which requires participants to perform heavy computations to prove their validity in the network [7]. These heavy computations demand significant power, conflicting with green computing principles. Since then, various blockchain technologies have introduced new consensus algorithms such as Chia's (XCH) Proof-of-Space (PoSpace), which rewards participants based on the allocation of disk storage space [4, 6]. However, Chia's plotting process stresses hardware, requiring expensive setups and shortening the lifespan of solid-state drives. [1]

Our work supports the notion of green computing by implementing a PoSpace protocol with two phases: hash-initialization and search-verification. This approach transforms a compute-intensive PoW problem into a data-intensive/storage problem, improving energy efficiency of blockchain systems. In this work, we compare the performance of our solutions – *Vault*, written in C, and *Vault-76* written in Rust – to Chia's plotting software (namely, "plotters") to demonstrate how our design outperforms in both throughput and latency. Efficient use of large HPC resources with many cores and multiple storage devices is essential for maximizing performance.

2 Architecture

2.1 Hash-Initialization

Figure 1 illustrates that the memory limit is equally shared among threads. Each thread generates hashes concurrently from random nonces with BLAKE3 library [10] and inserts them into a DashMap, a hashmap variant in Rust that supports concurrent insertions [5]. Each entry in the hashmap represents a bucket of records that start with the same prefix. Once the capacity of the hashmap is reached, each bucket is written to disk concurrently at a specific offset. This process repeats until all records are generated, resulting in a file with records sorted by prefixes.

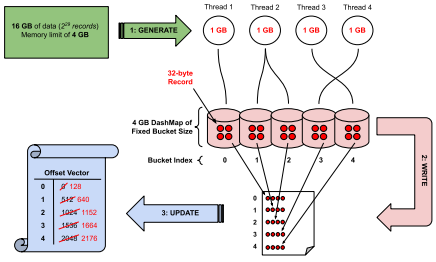

Figure 1: Hashes inserted into hashmap and flushed to disk

In Figure 2, the records are read into memory and sorted within their buckets using an optimized combination of Quicksort and Heapsort algorithms [11]. The records are then written back to disk in a fully sorted order.

2.2 Search-Verification

This stage is intended to verify whether a network participant stores a record file by searching for a particular hash. In Figure 3, a binary search algorithm is employed, resulting in fast lookup performance.

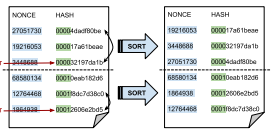

Figure 2: Sorting records

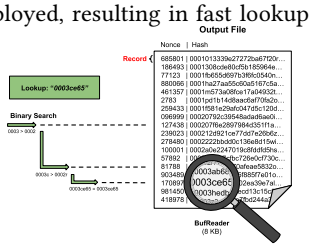

Figure 3: Looking up records

3 Performance Evaluation

In order to evaluate the performance of *Vault-76*, we generated files with 2^k 32-byte records and collected throughput data on machines with varying compute capabilities [3] and different types of drives. Table 1 lists all Mystic nodes used for running benchmarks [9]. *Vault-76* and *Vault* tests were conducted with a file size-to-memory limit ratio of 2 and a maximum memory limit of 16GB.

MACHINES	CPU	CORES	RAM	STORAGE	ISA
8Socket	Intel Xeon Platinum 8160 @2.10GHz	192	768GB	SATA, NVMe SSD	x86_64
EpycBox	AMD EPYC 7501 @2.00GHz	64	320GB	SATA, NVMe SSD, SATA HDD	x86_64
Raspberry Pi 4	Broadcom BCM2711, ARM Cortex-A72 @1.8GHz	4	2GB	SATA SSD, SATA HDD	aarch64

Table 1: Machines used for experiments

3.1 Comparison to Chia Plotters

The current Chia plotters present significant speedup opportunities due to their complex algorithm, which creates seven tables of SHA256 hash data. *Vault-76*, on the other hand, uses BLAKE3 hash function, which is four times faster than SHA256 [2]. Additionally, current Chia plotters can only create file sizes with a minimum k of 32 (101.4 GiB), limiting their use on smaller systems. While Chia plans to introduce smaller plot sizes in the coming years [8], our solution already surpasses Chia plotters in this regard.

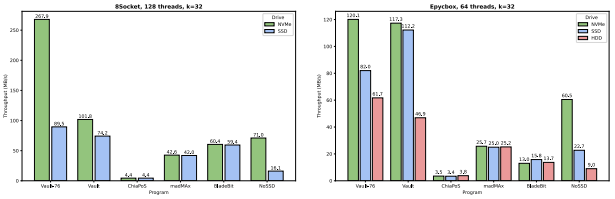

Figure 4: Throughput comparison of various plotters

Search Performance: Table 2 presents the lookup latency measured on the Epycbox machine (Table 1) after generating a file with k=32 records.

DRIVE	MIN (ms)	AVG (ms)	MAX (ms)
NVMe SSD	1.25	2.31	3.71
SATA SSD	2.10	31.86	155.08
SATA HDD	33.83	195.57	393.89

DRIVE	MIN (ms)	AVG (ms)	MAX (ms)
NVMe SSD	8.00	17.14	118.00
SATA SSD	10.00	87.28	346.00
SATA HDD	224.00	310.66	411.00

Table 2: Search-Verification phase latency: *Vault-76 (left)*; *ChiaPoS (right)*;

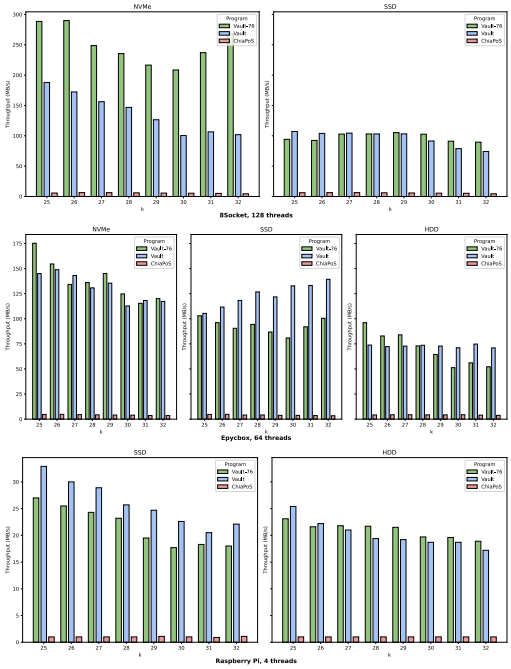

Figure 5: Throughput comparison for k25-32 on various Mystic machines

4 Contributions

Our work has the following contributions:

- Improved PoSpace performance (both write throughput and search time) by up to an order-of-magnitude across a variety of hardware.
- Reduced I/O requirements and improved non-volatile memory longevity by decreasing I/O writes to two and I/O reads to one.

5 Conclusion

The experiments demonstrate that *Vault-76* and *Vault* significantly outperform Chia plotters across all system sizes. Both programs benefit from an increased thread count; however, with HDDs, I/O operations represent a bottleneck, resulting in similar performance for both Rust and C implementations. With NVMe drives, hash generation becomes the bottleneck, where Rust shows superior performance due to its native BLAKE3 library, which supports concurrent hash generation more effectively under high parallelism. Additionally, *Vault-76* has lower latency in the search-verification phase due to its simplified file traversal, whereas Chia's lookup involves seven tables.

Future work will focus on enhancing *Vault-76*'s performance through the implementation of caching and compression. Caching will involve adding an NVMe layer to store frequently accessed data, while compression will employ a lossless technique to reduce file size.

6 Acknowledgments

This work is supported in part by the National Science Foundation OAC-2150500 award.

References

[1] [n. d.]. https://docs.chia.net/ssd-endurance/
[2] Aahad Abubaker, Tanmay Anand, Sonal Gaikwad, Mahad Haider, Jacklyn McAninch, Lan Nguyen, Alexandru Iulian Orhean, and Ioan Raicu. 2023. Exploring Green Cryptographic Hashing Algorithms for Eco-Friendly Blockchains. (2023).
[3] AWS. 2024. What is Compute? https://aws.amazon.com/what-is/compute/ [Online; 2024].
[4] Bram Cohen and Krzysztof Pietrzak. 2019. The chia network blockchain. *White Paper, Chia. net* 9 (2019).
[5] Docs.rs [n. d.]. DashMap. https://github.com/xacrimon/dashmap?tab=readme-ov-file
[6] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. 2015. Proofs of space. In *Annual Cryptology Conference*. Springer, 585–605.
[7] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system.
[8] Chia (XCH) Network. 2024. Approaching the Next Generation of Proof of Space. https://www.chia.net/2024/08/08/approaching-the-next-generation-of-proof-of-space/ [Online; posted 8-August-2024].
[9] AI Orhean, A Ballmer, T Koehring, K Hale, XH Sun, O Trigalo, N Hardavellas, S Kapoor, and I Raicu. 2019. Mystic: Programmable systems research testbed to explore a stack-wide adaptive system fabric. In *8th Greater Chicago Area Systems Research Workshop (GCASR)*.
[10] Jack O'Connor, Jean-Philippe Aumasson, Samuel Neves, and Zooko Wilcox-O'Hearn. 2020. one function, fast everywhere. (2020).
[11] Orson Peters. [n. d.]. ORLP/pdqsort: Pattern-defeating quicksort. https://github.com/orlp/pdqsort