A Technical Analysis of Chia Proof of Space

Zack Chaffee [1]   Ioan Raicu [1]

[1]Department of Computer Science, Illinois Institute of Technology

Proof of Space

Our research provides a comprehensive analysis of Chia Network's Proof of Space implementation[2], identifying core limitations and optimization opportunities that can serve as a foundation for future cryptocurrency systems. By examining this sustainable consensus mechanism that utilizes storage rather than computation, we offer technical insights that will enable developers to build more efficient, secure, and widely-adopted Proof of Space cryptocurrencies.

Plot Architecture

Plots are optimized cryptographic data structures for allocated storage

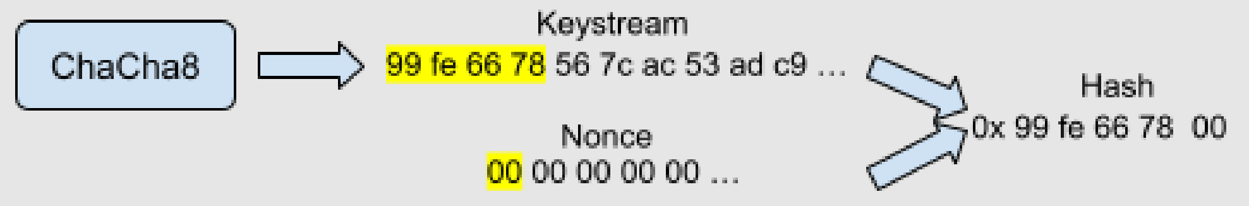
- Header Version, k-value, magic ID, memo, plot ID, compression level
- Tables 1-7 Organized in buckets for efficient lookup
- Checkpoints Optimize verification and lookup speed

The size for a typical plot, size k=32 or 2^{32}, is 101.9GB.

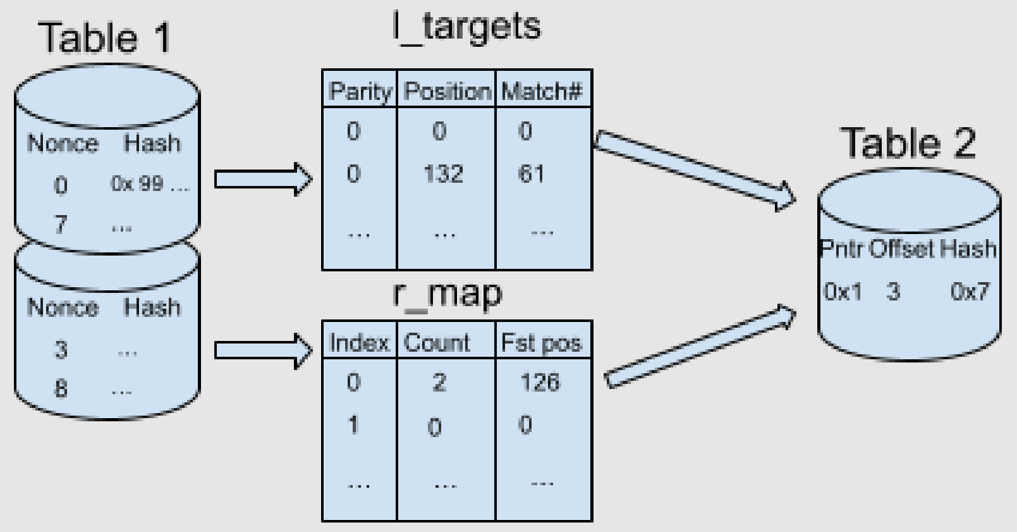
Phase 1: Forward Propagation

The process of generating a table of nonce hash pairs and finding matches in each table to form the next until there are 7 total tables.

- Table 1 generation



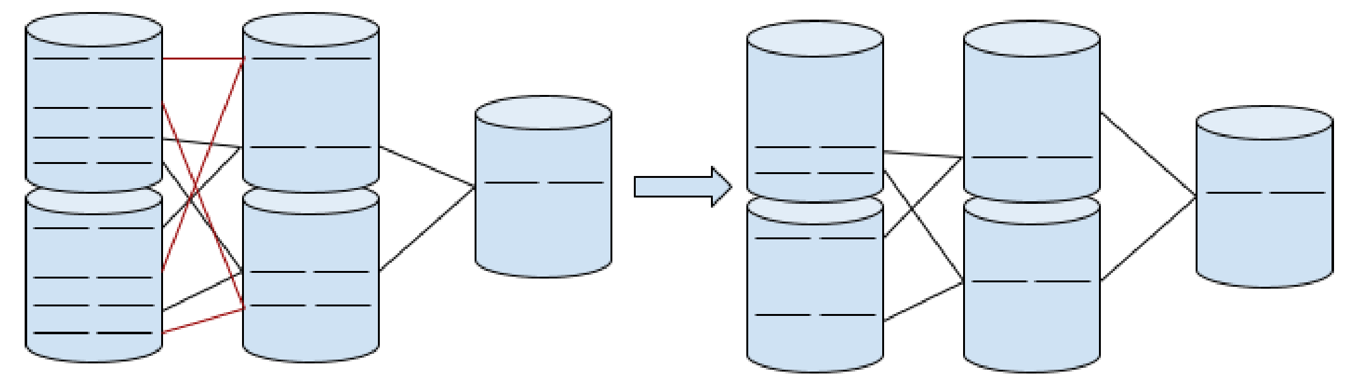
- Bucketing
  - A formula is used to put each entry in a bucket
- Matching
  - Generate 64 potential matches per left value to increase match probability



This process of bucketing and matching is repeated until there are 7 total tables generated

Phase 2: Back Propagation

Not all values will be a part of chains that contribute to a valid solution in table 7. These values are removed to reduce and optimize the plot.



Phase 3: Compression

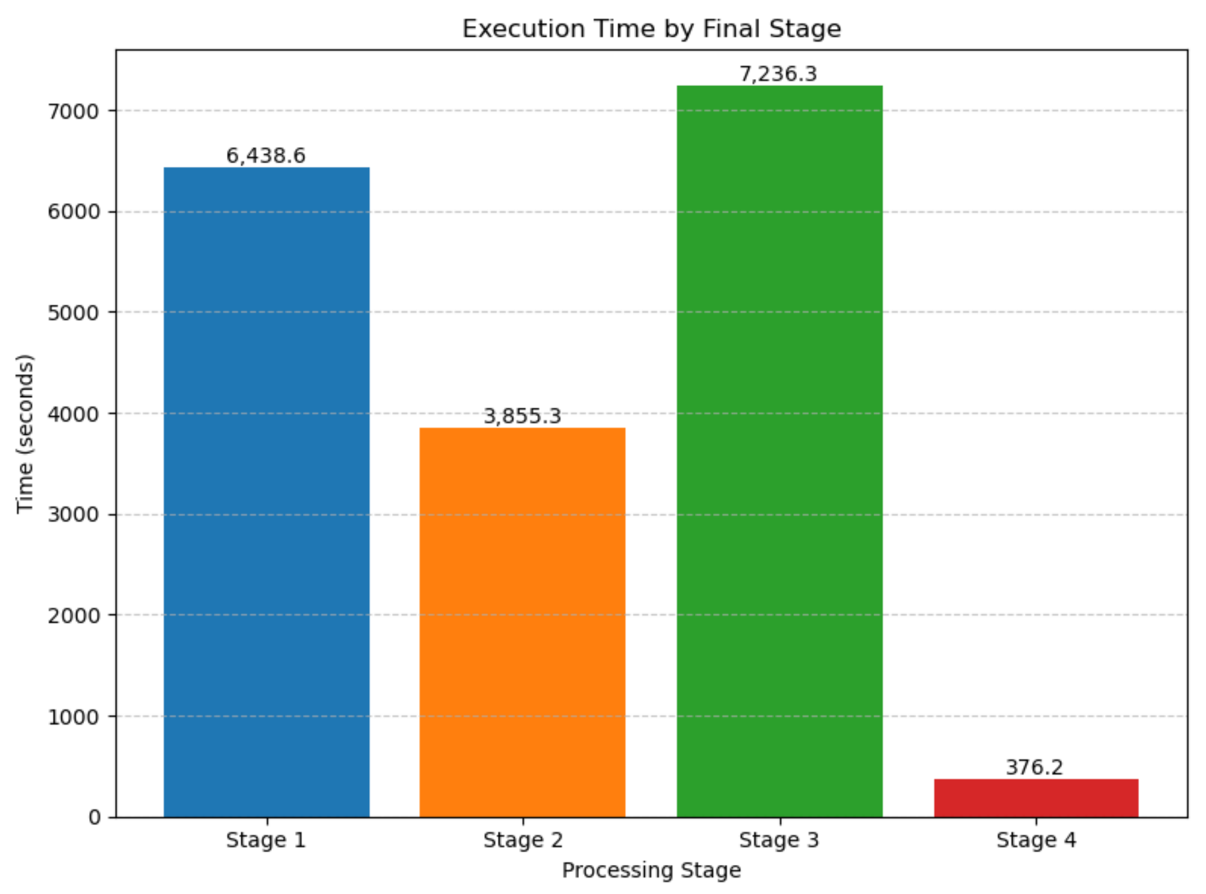
Now that the tables are fully generated and finalized their total sizes can be further reduced removing metadata and eliminating redundancy.

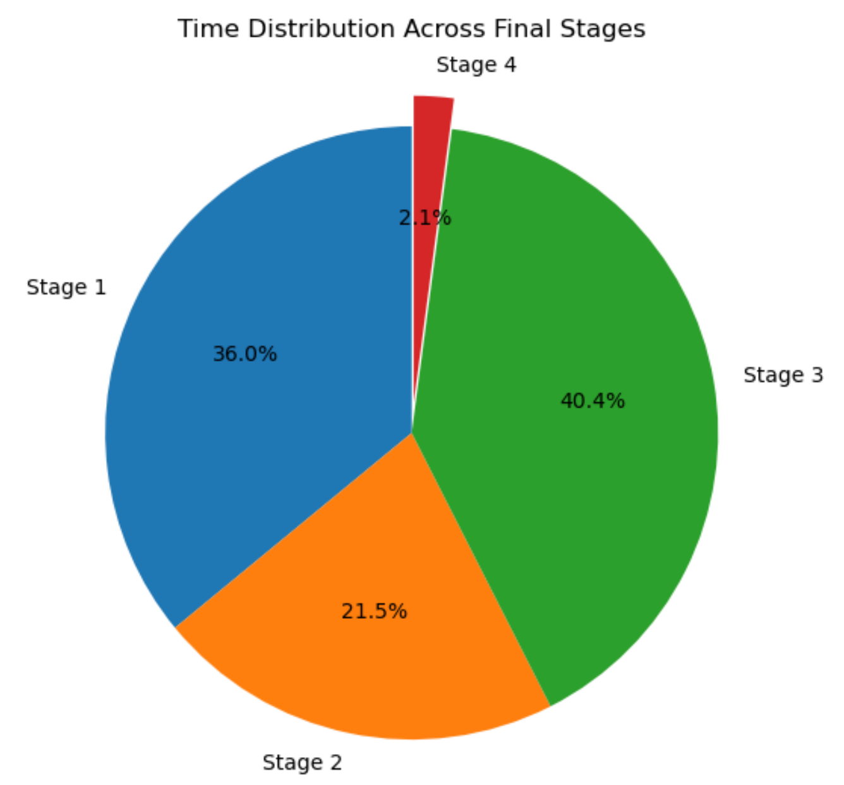
- Park Bit Packing Parks contains relationships between multiple entries
- Delta Encoding Store offset of consecutive values instead
- Line Point Format Don't store information identifying bucket

Phase 4: Check-pointing & Optimization

- Strategically places full-value park pointers at fixed intervals (typically every 2048 entries)
- Reorganizes compressed data for optimal sequential read patterns
- Generates C1-C3 indexed lookup tables that enable binary search
- Constructs proof quality map for rapid filtering of potential solutions
- Balances checkpoint frequency against storage overhead

Execution Time





Bucketing & Sorting Optimization

Targets

Generates 3D table of targets for a bucket. Uses bit-packed format and cache-aligned storage and enables parallel evaluation via SIMD operations

Right Map

Normalizes y-values to bucket positions with each entry representing potential matches requiring verification. It uses early rejection filters to minimize full validation overhead and optimizes memory layout for efficient collision handling

Sorting

Different techniques used depending on the phase

- Phase 1 bucket sort, two-phase sorting, sort on disk
- Phase 2 backpropogation-optimized sort, uniform sort
- Phase 3 quicksort
- Phase 4 uniform sort, quicksort

Impact

This analysis reveals critical optimization techniques in Chia's Proof of Space implementation that can benefit the broader cryptocurrency ecosystem:

- Provides a technical foundation for future Proof of Space coins
- Identifies specific compression and checkpointing strategies that balance storage efficiency with access speed
- Already informing VaultX's[1] enhanced Proof of Space implementation

References

[1] Varvara Bondarenko and Ioan Raicu.
Securing proof-of-space against hellman attacks.
In *Proceedings of the Conference on Blockchain Technology*, 2023.
Poster Presentation.

[2] Chia.
Chia proof of space construction.

GCASR 2025, Chicago