STORAGE DEVELOPER CONFERENCE

SDC 22

Fremont, CA
September 12-15, 2022

BY Developers FOR Developers

A SNIA Event

RainBlock: Faster Transaction Processing for Public Blockchains

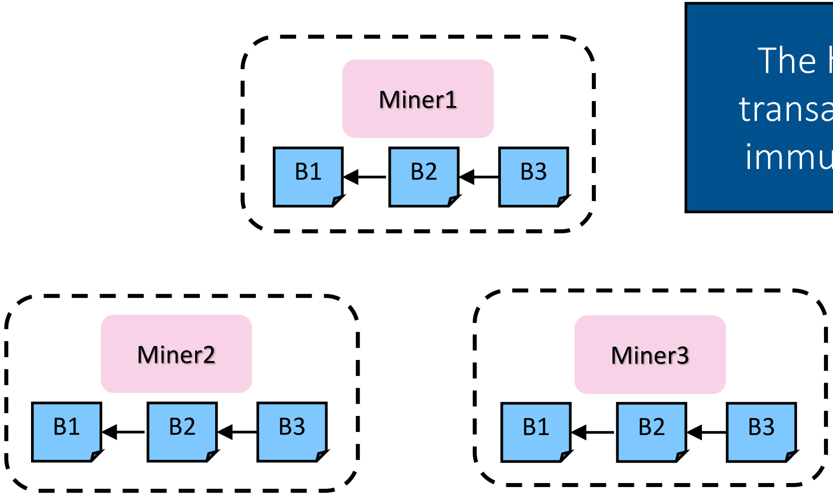
Soujanya Ponnapalli, Aashaka Shah, Souvik Banerjee, Dahlia Malkhi, Amy Tai, Vijay Chidambaram, and Michael Wei

Presented by: Soujanya Ponnapalli

Outline

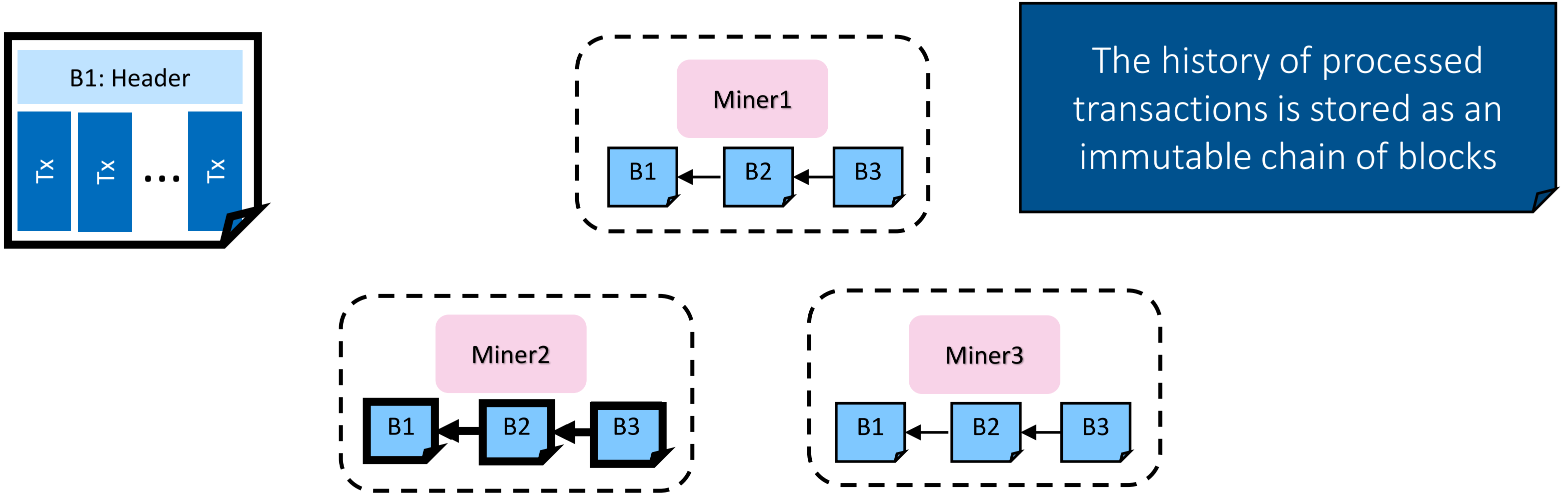
- Overview of public blockchains
- Low throughput and I/O bottlenecks
- RainBlock

Blockchains: Decentralized Databases



The history of processed transactions is stored as an immutable chain of blocks

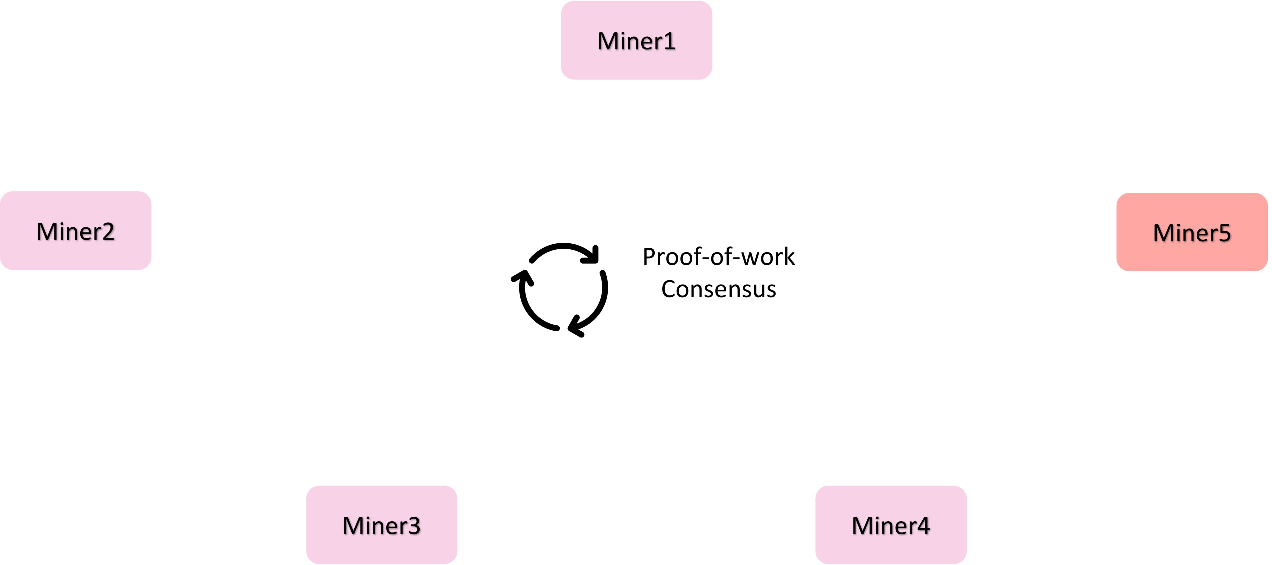
Blockchains: Decentralized Databases



The history of processed transactions is stored as an immutable chain of blocks

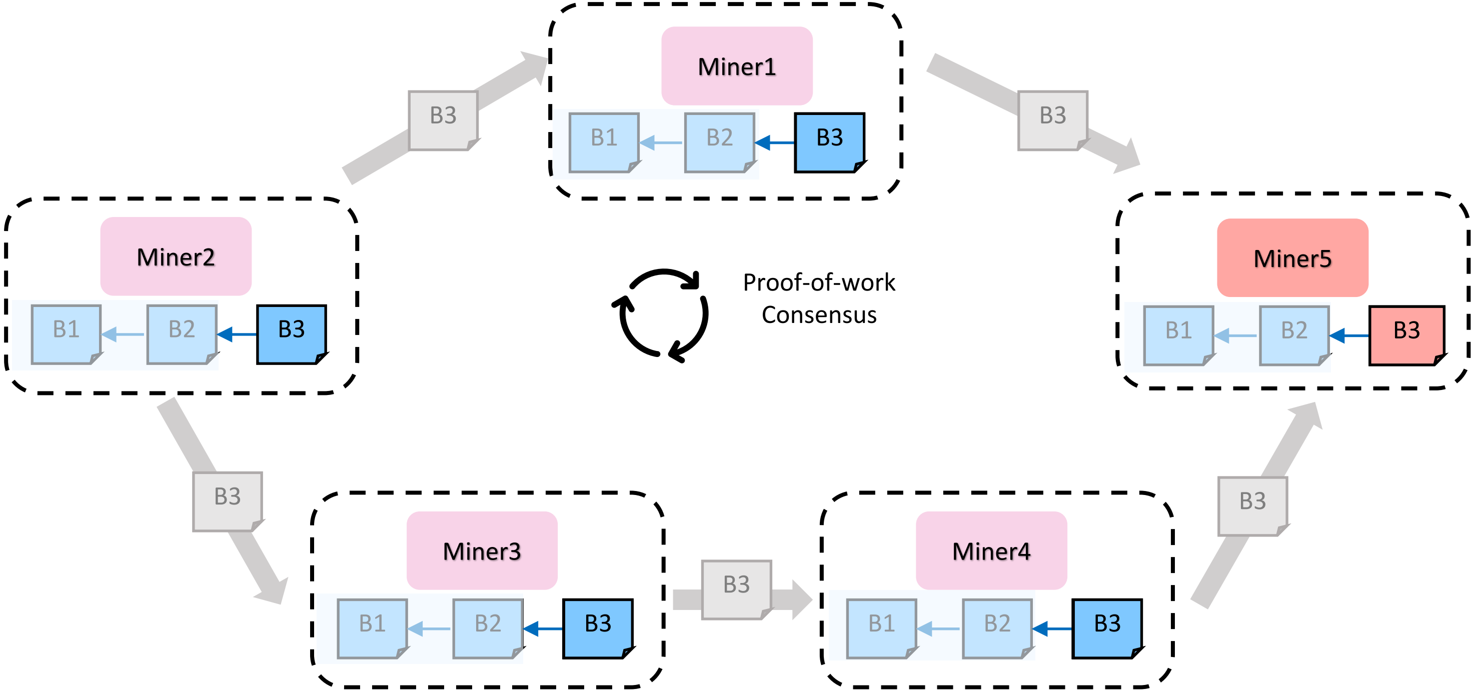
Public Blockchains: Proof-of-work consensus

Open networks



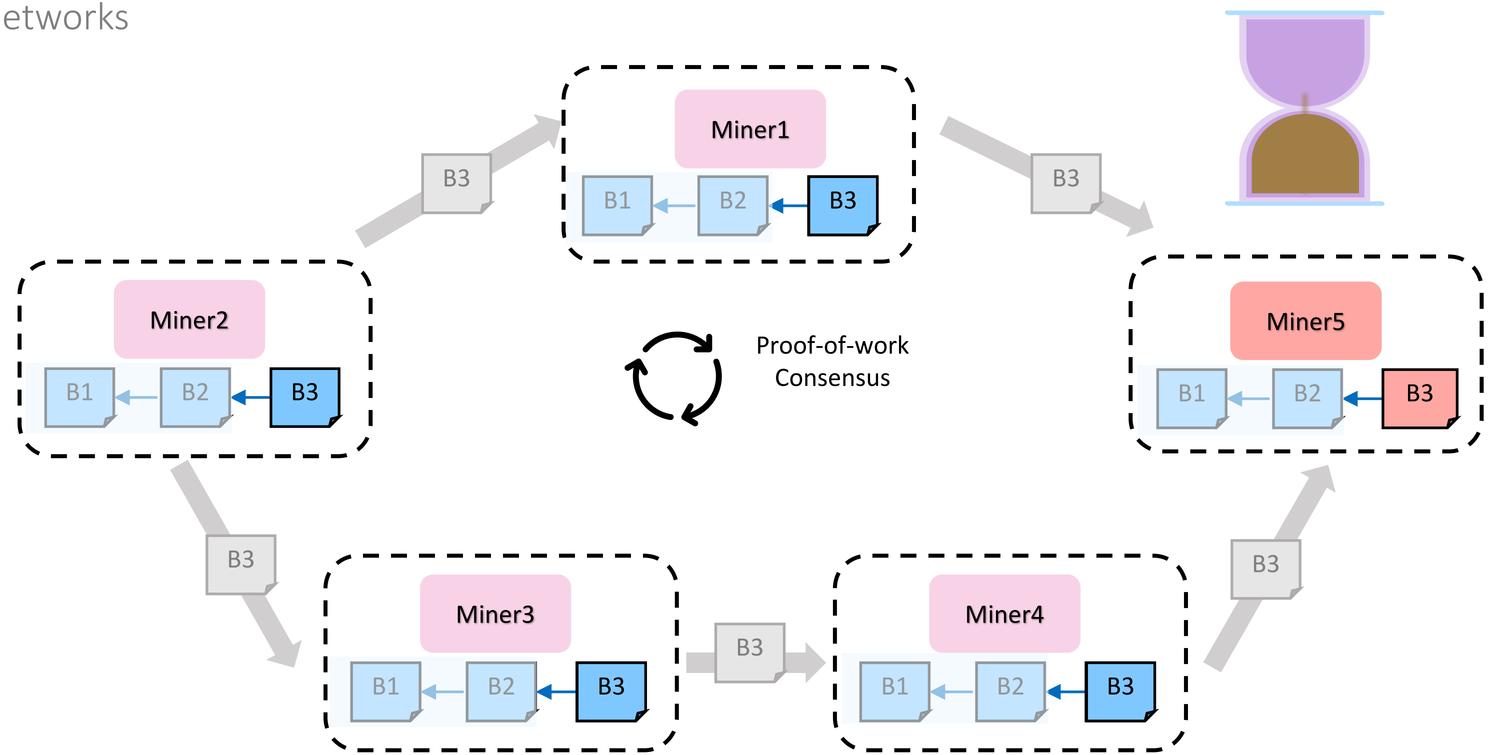
Public Blockchains: Proof-of-work consensus

Open networks



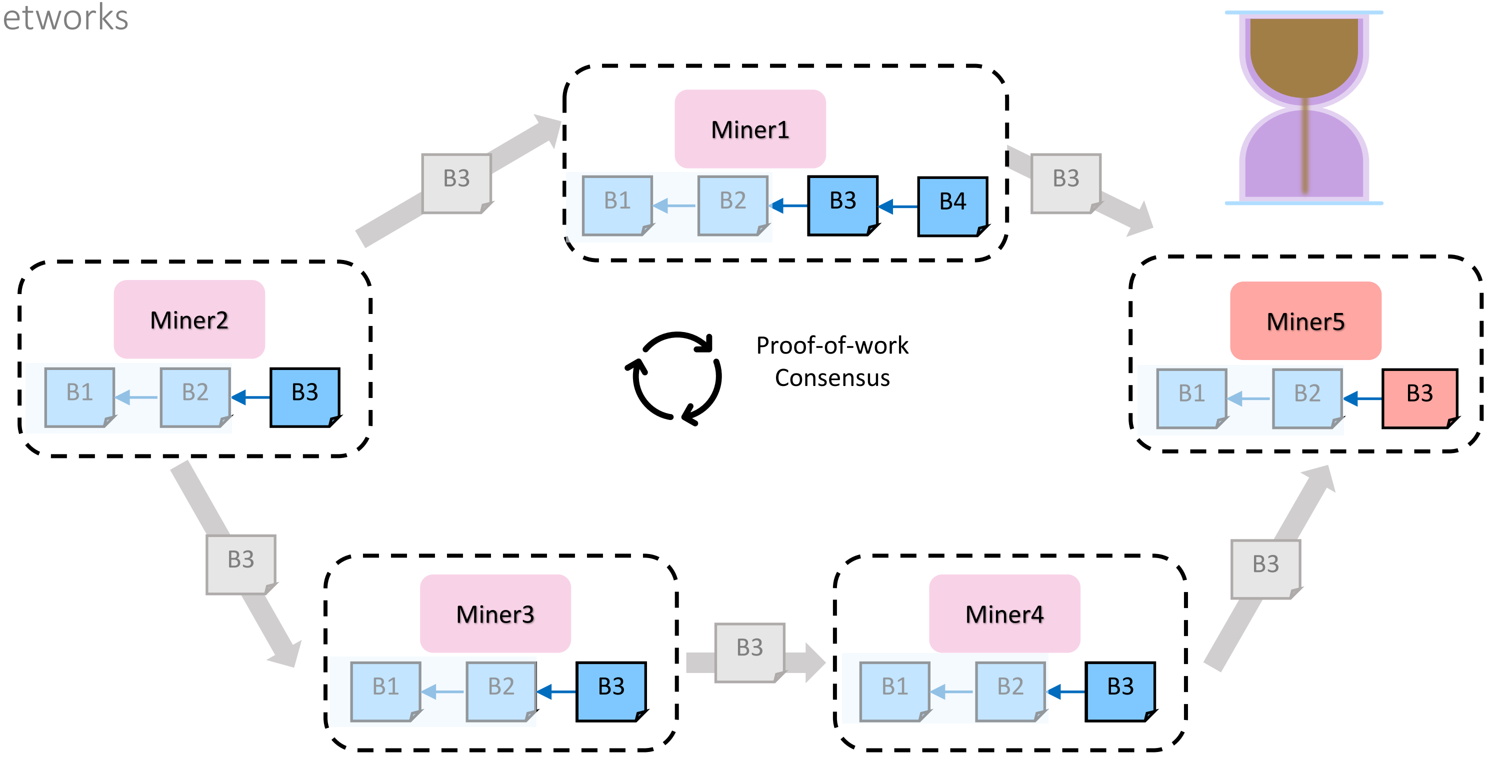
Public Blockchains: Proof-of-work consensus

Open networks



Public Blockchains: Proof-of-work consensus

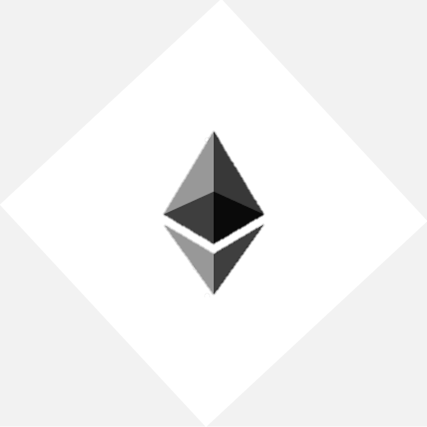
Open networks



Proof-of-work rate limits the creation of new blocks

Public Blockchains – Low throughput

Throughput: 20 tps



Throughput: 16 tps

Throughput: 24K tps
(1000x higher)

Public blockchains need to scale for wide-spread adoption

Prior work: New or Alternative Consensus

Inclusive blockchain protocols
International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2015. Lewenberg, Yoad, Yonatan Sompolinsky, and Aviv Zohar.

DAG instead of chain

Bitcoin-ng: A scalable blockchain protocol.
13th USENIX symposium on networked systems design and implementation (NSDI). 2016
Eyal, Ittay, et al.

Leader election

⋮

Algorand: Scaling byzantine agreements for cryptocurrencies.
Proceedings of the 26th Symposium on Operating Systems Principles. 2017 Gilad, Yossi, et al.

Proof-of-stake consensus

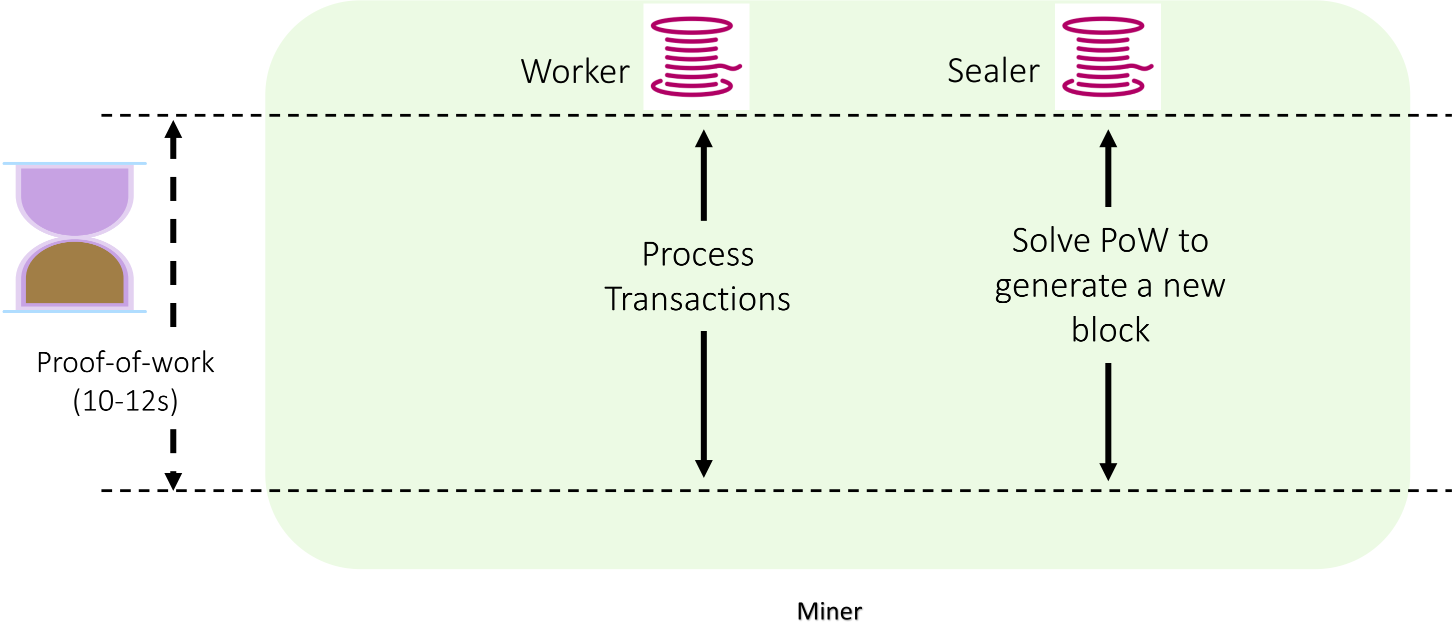
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work

Miner

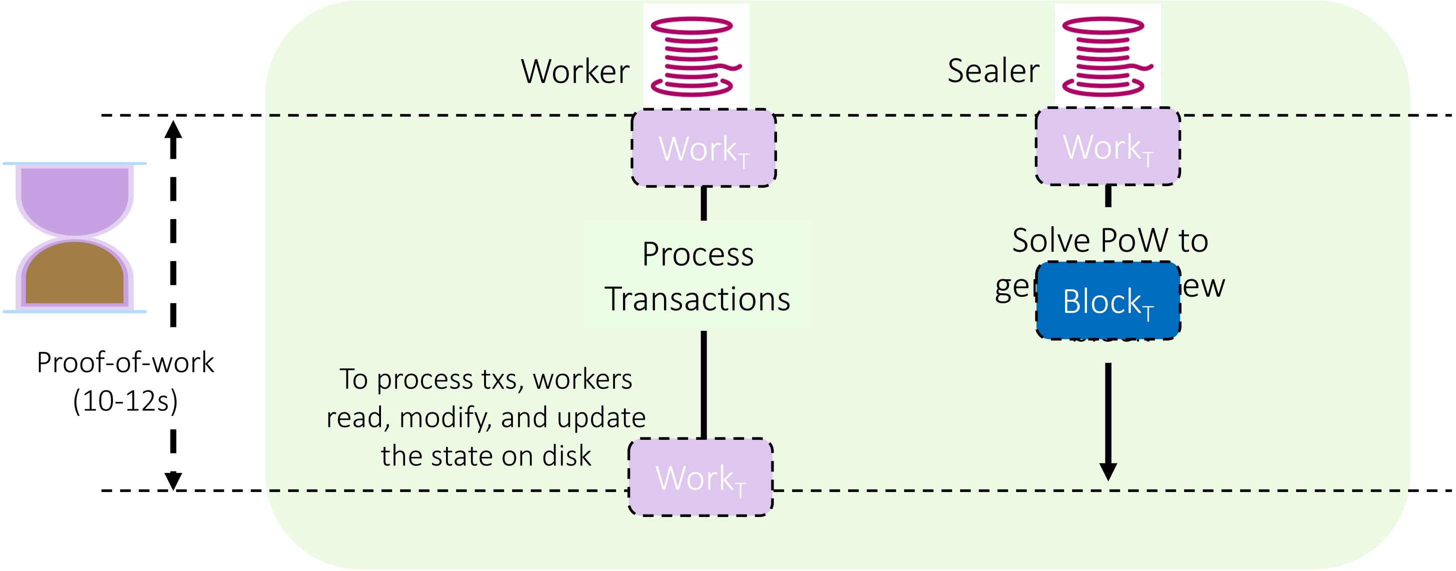
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work

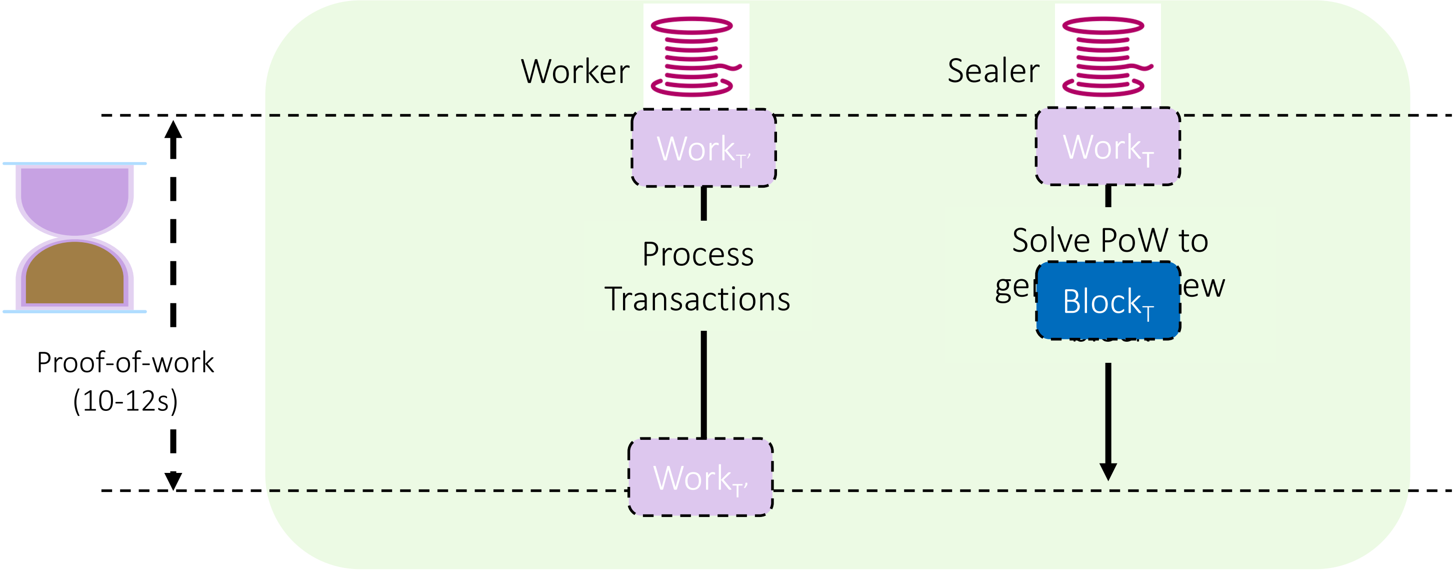


To process txs, workers read, modify, and update the state on disk

Workers perform I/O for processing txs

Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

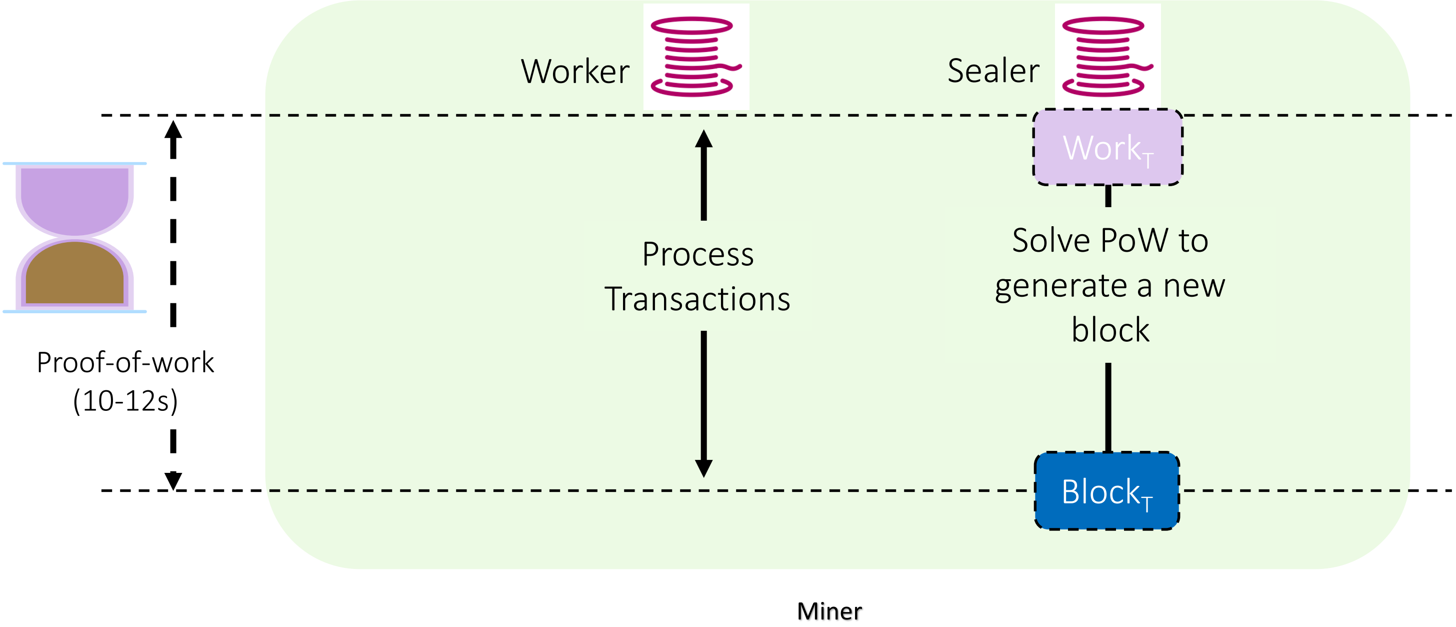
Safety and security from proof-of-work



Miner

Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



Insights

- Miners process more transactions per unit time
  - Larger blocks with more transactions

- Unmodified proof-of-work
  - Inherit safety and security properties

Faster Transaction Processing

Goal: Miners process more transactions in the same amount of time

Approach: Reducing I/O bottlenecks in transaction processing; allowing miners to safely release larger blocks

Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work

Proof-of-work rate limits the creation of new blocks, but it **does not** restrict the number of transactions in each block!

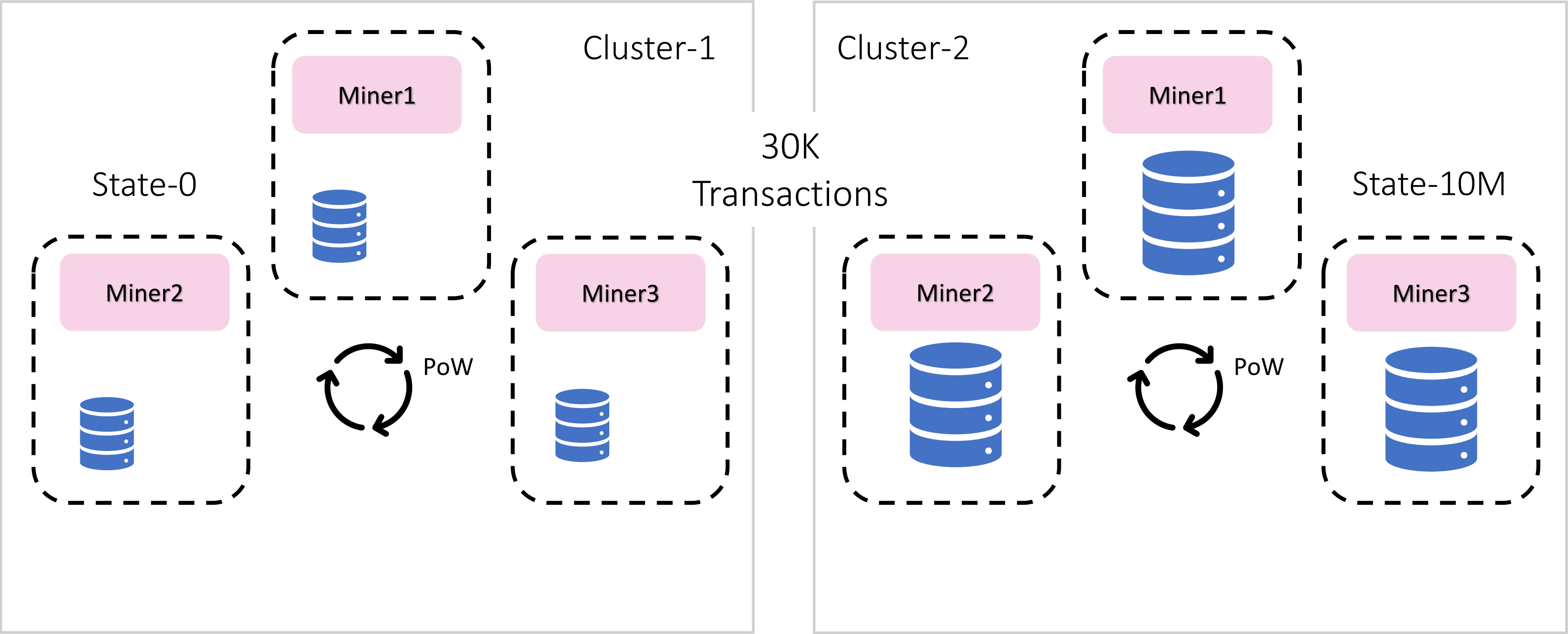
Transactions

generate a new

I/O bottlenecks limits the block size, and thereby reduce the overall throughput of public blockchains!

Miner

Impact of I/O on Overall Throughput



Cluster-1

State-0

Miner1

Miner2

Miner3

PoW

30K Transactions

Cluster-2

State-10M

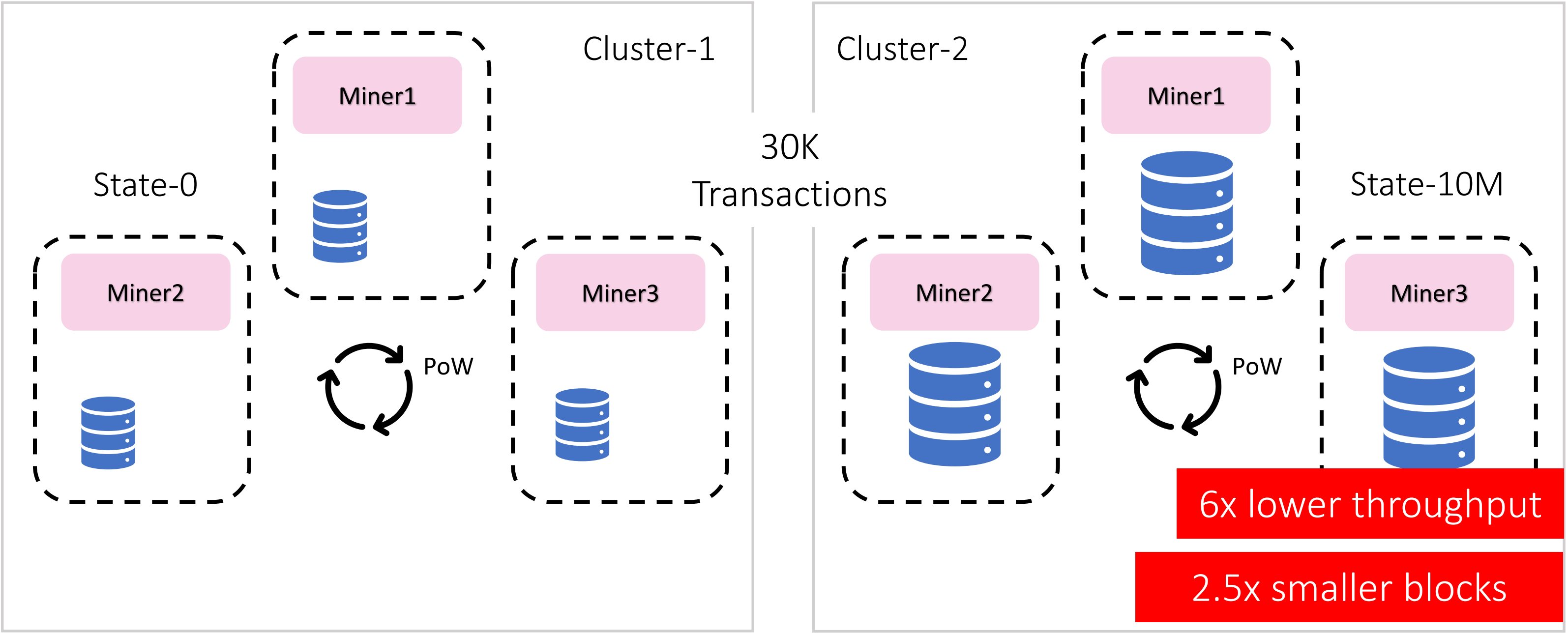
Miner1

Miner2

Miner3

PoW

Impact of I/O on Overall Throughput



6x lower throughput

2.5x smaller blocks

I/O bottlenecks limit block size and the overall throughput

RainBlock: Faster Transaction Processing

RainBlock, a new architecture for public blockchains, increases overall throughput without modifying proof-of-work consensus

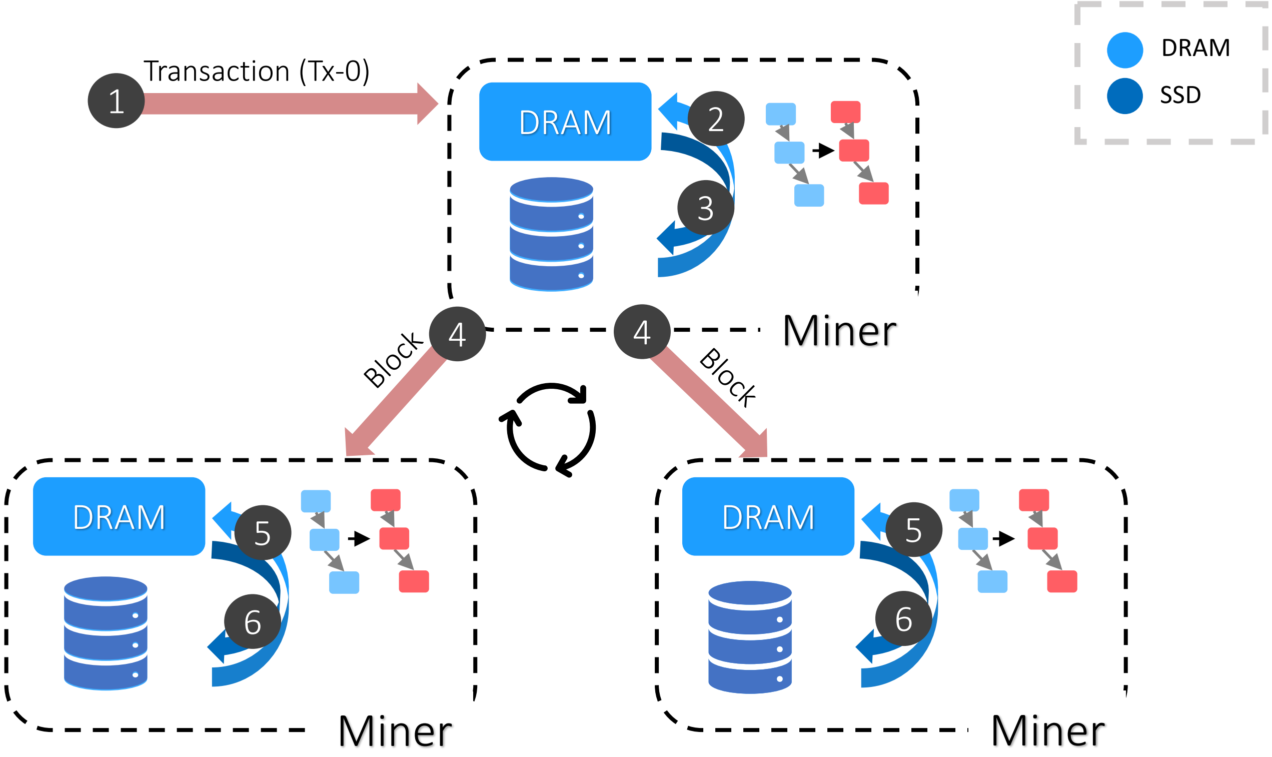
RainBlock eliminates I/O bottlenecks in transaction processing, allowing miners to process and verify more transactions in the same amount of time

RainBlock employs the novel Distributed Sharded Merkle Tree (DSM-Tree) for I/O-efficient transaction processing

In a geo-distributed setting, with 4 miners in 4 regions spread across 3 continents, RainBlock miners can process about 20000 transactions per second

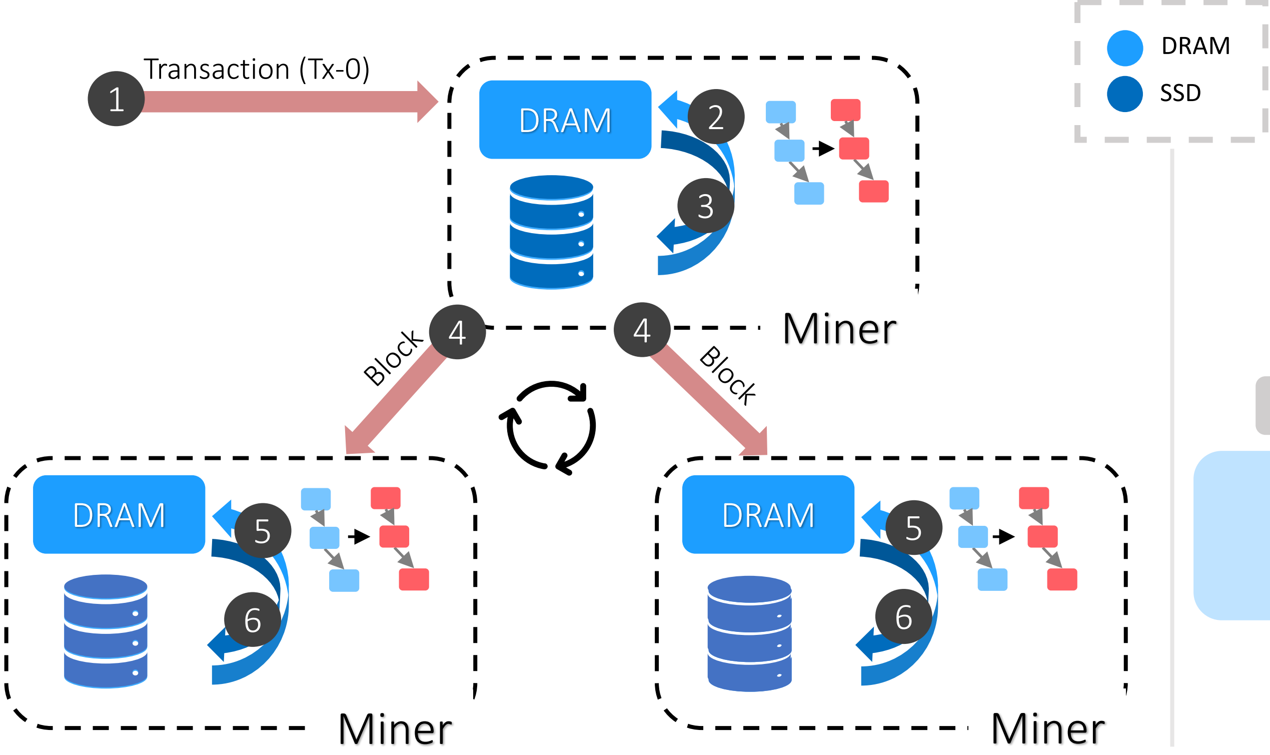
Transaction Processing in Ethereum

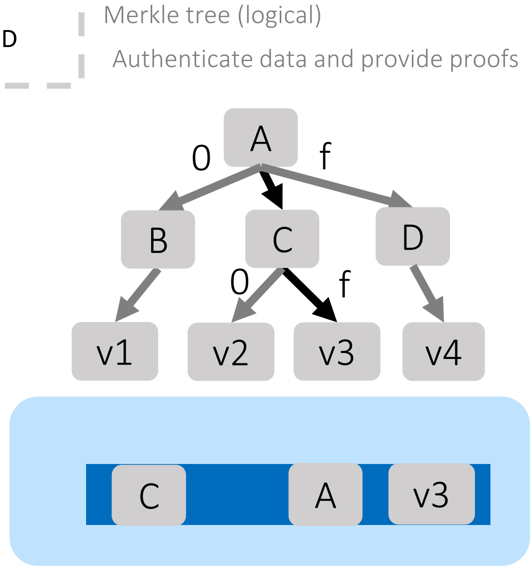
Accessing and modifying state in the critical path of processing transactions



Transaction Processing in Ethereum

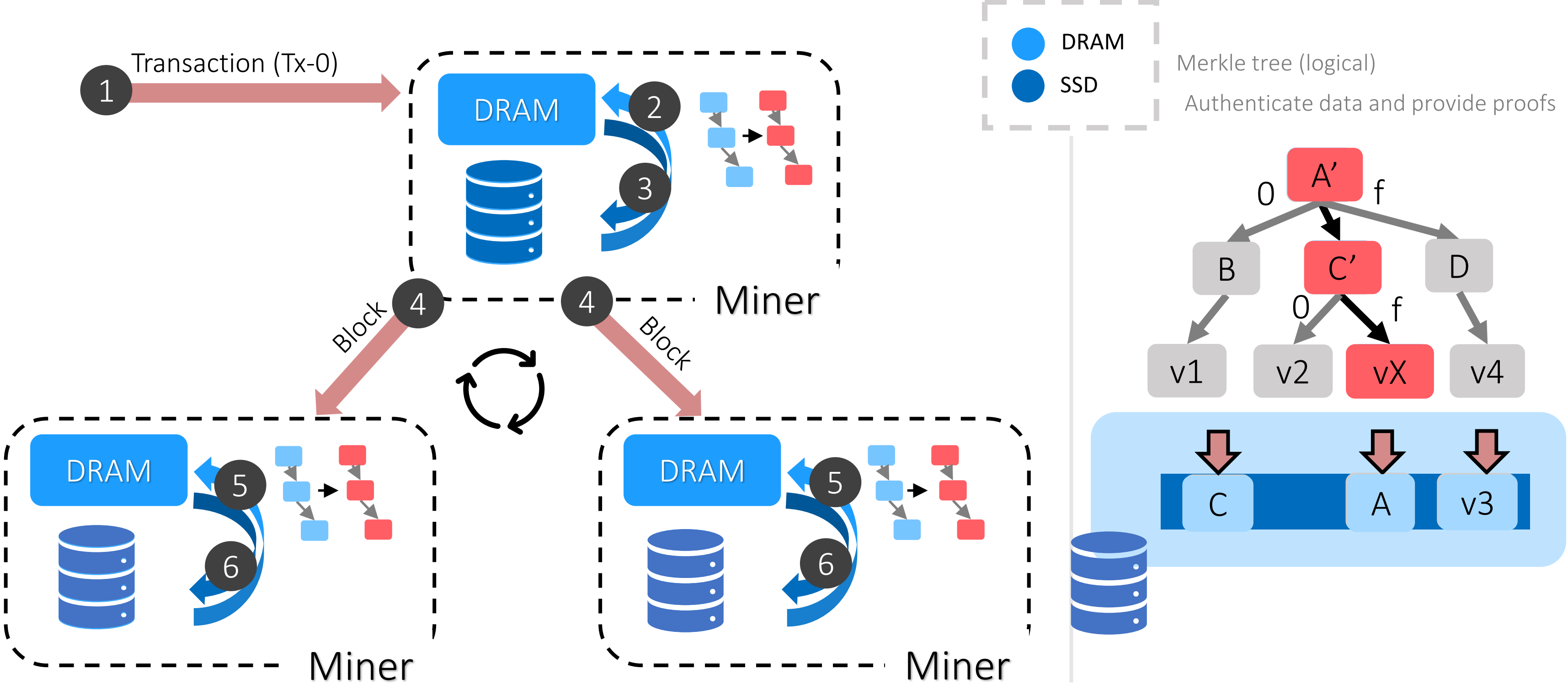
Accessing and modifying state in the critical path of processing transactions





Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



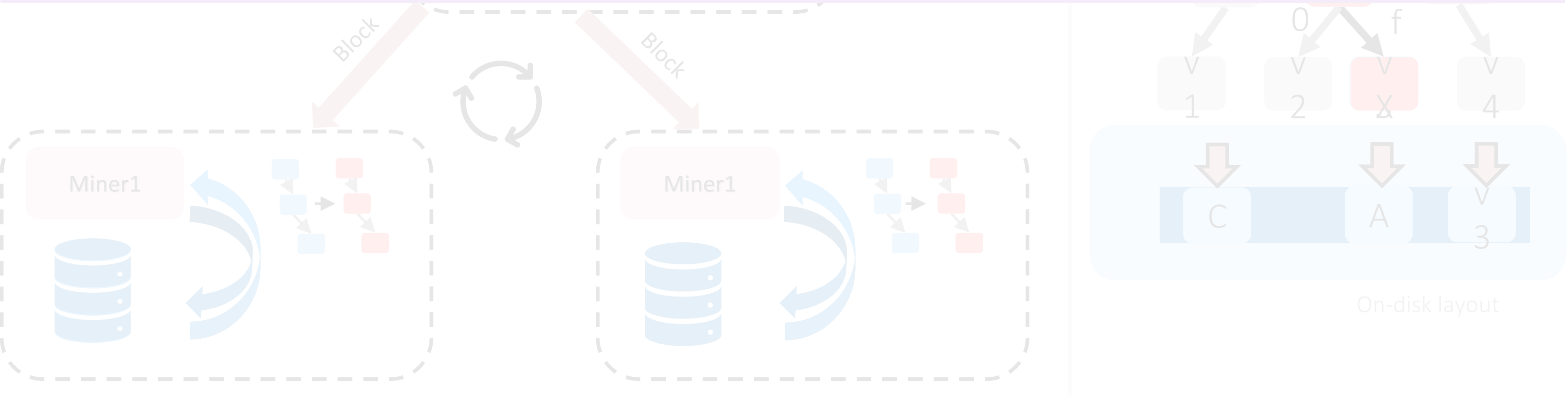
Merkle tree (logical)

Authenticate data and provide proofs

Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions

To read or update a single 100B user account, Ethereum reads above 4MB, resulting in 40-60x I/O amplification!



Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions

To read or update a single 100B user account, Ethereum reads above 4MB, resulting in 40-60x I/O amplification!

To process a single block of 100 simple transactions, Ethereum performs more than 10,000 (100x) random I/O operations!

Outline

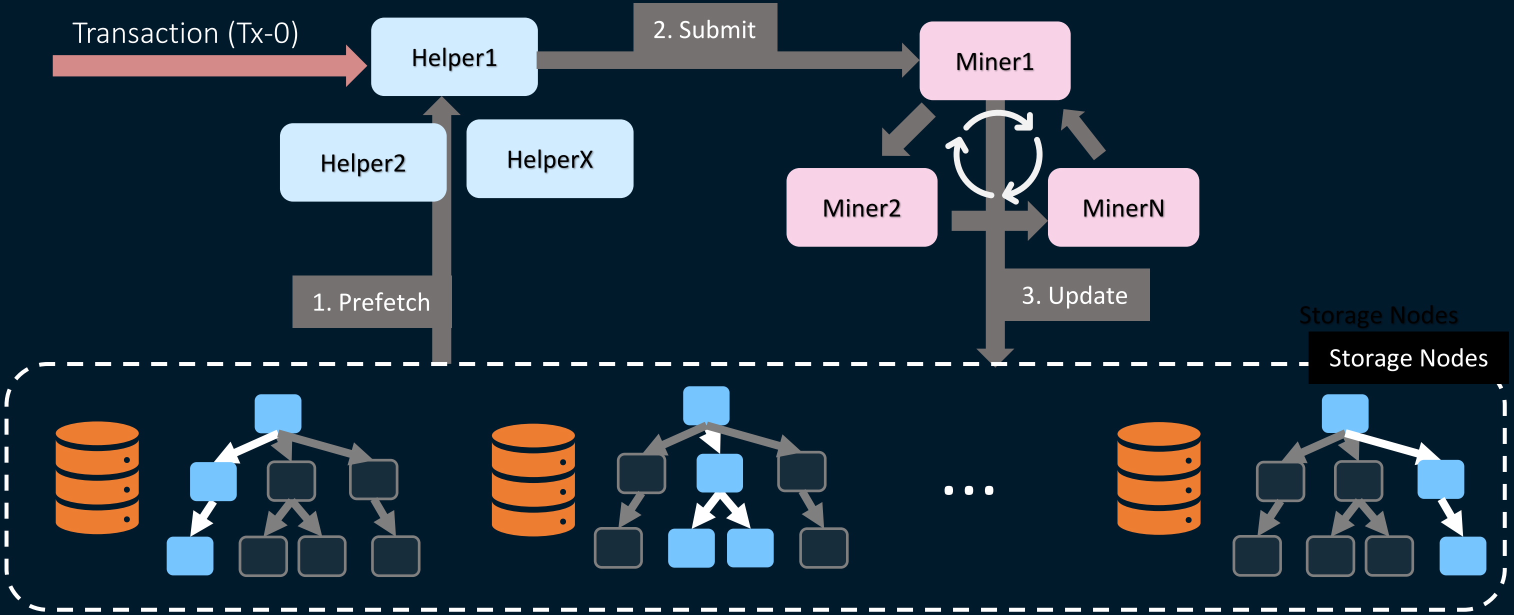
RainBlock architecture reduces I/O bottlenecks

Challenges RainBlock addresses

Life of a Transaction in RainBlock

RainBlock: Architecture for Public Blockchains

Miners do not perform I/O in the critical path



Distributed Sharded Merkle Tree (DSM-Tree)

RainBlock: Challenges

Challenge-I: Concurrent updates to storage

I/O-Helpers can prefetch from storage nodes while miners are updating them

- Storage nodes need to provide consistency in the presence of concurrency!

RainBlock: Challenges

Challenge-I: Concurrent updates to storage

I/O-Helpers can prefetch from storage nodes while miners are updating them

- Storage nodes need to provide consistency in the presence of concurrency!

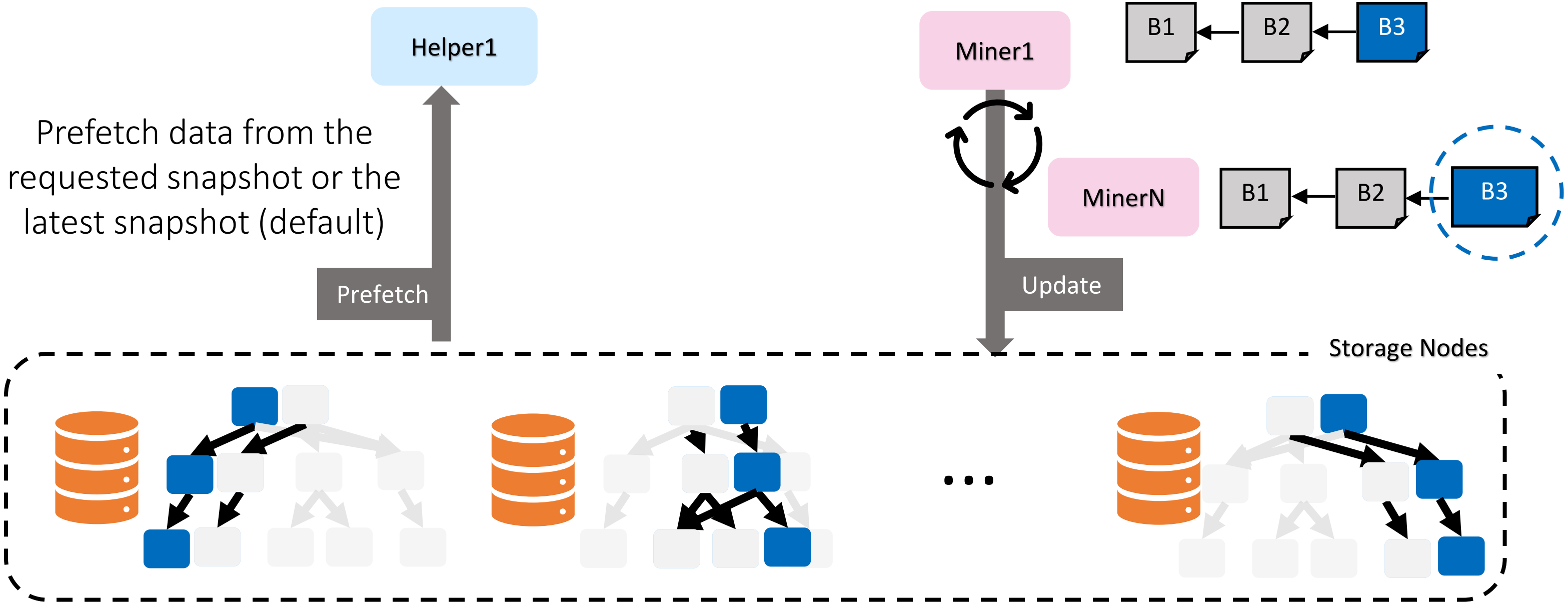
Challenge-II: Increased network traffic

RainBlock trades off local disk-I/O for network-I/O

- Data is now transmitted over the network, and is very large
- Stateless Clients proposal did not gain traction due to high network overheads

Handling Concurrent I/O operations

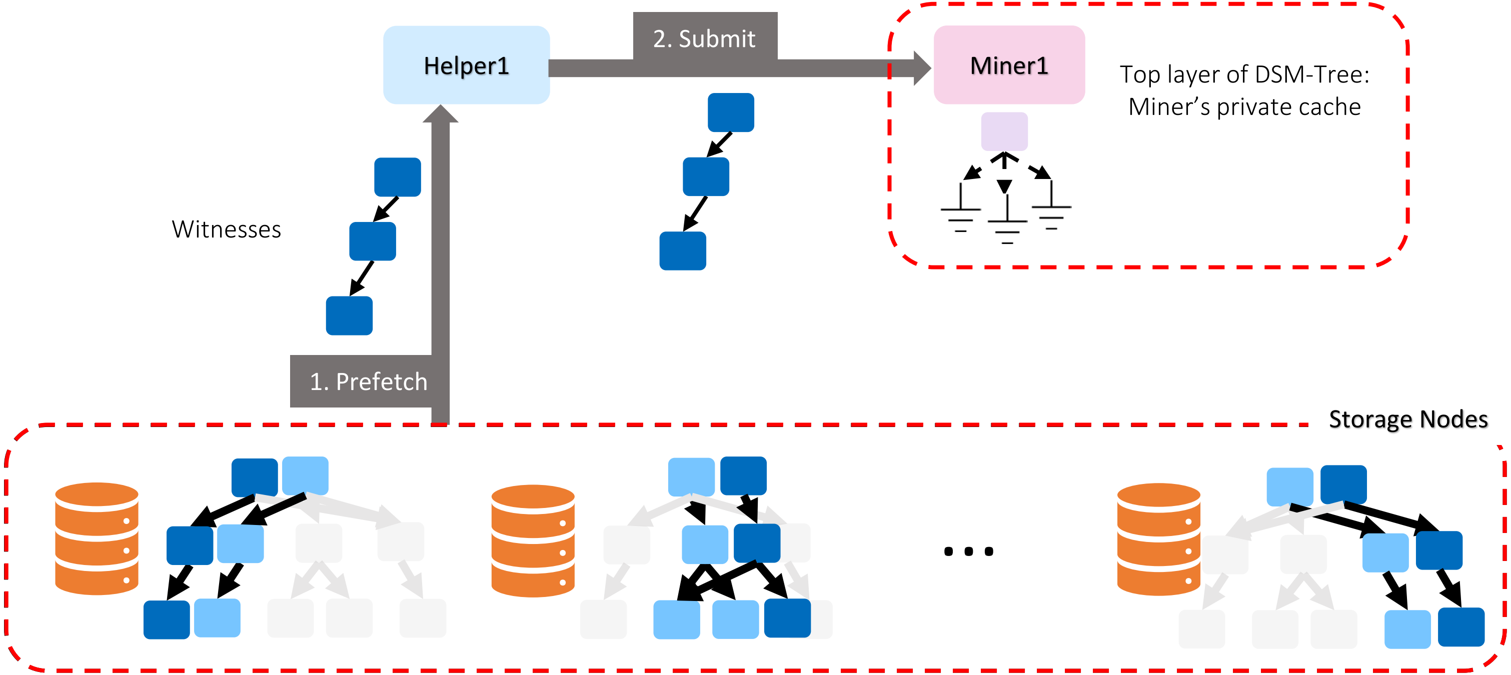
Concurrency and consistency


Helper1
Prefetch data from the requested snapshot or the latest snapshot (default)
Prefetch
Miner1
B1
B2
B3
MinerN
B1
B2
B3
Update
Storage Nodes
...

Distributed Sharded Merkle Tree (DSM-Tree)

On updates, shards create new copies of data

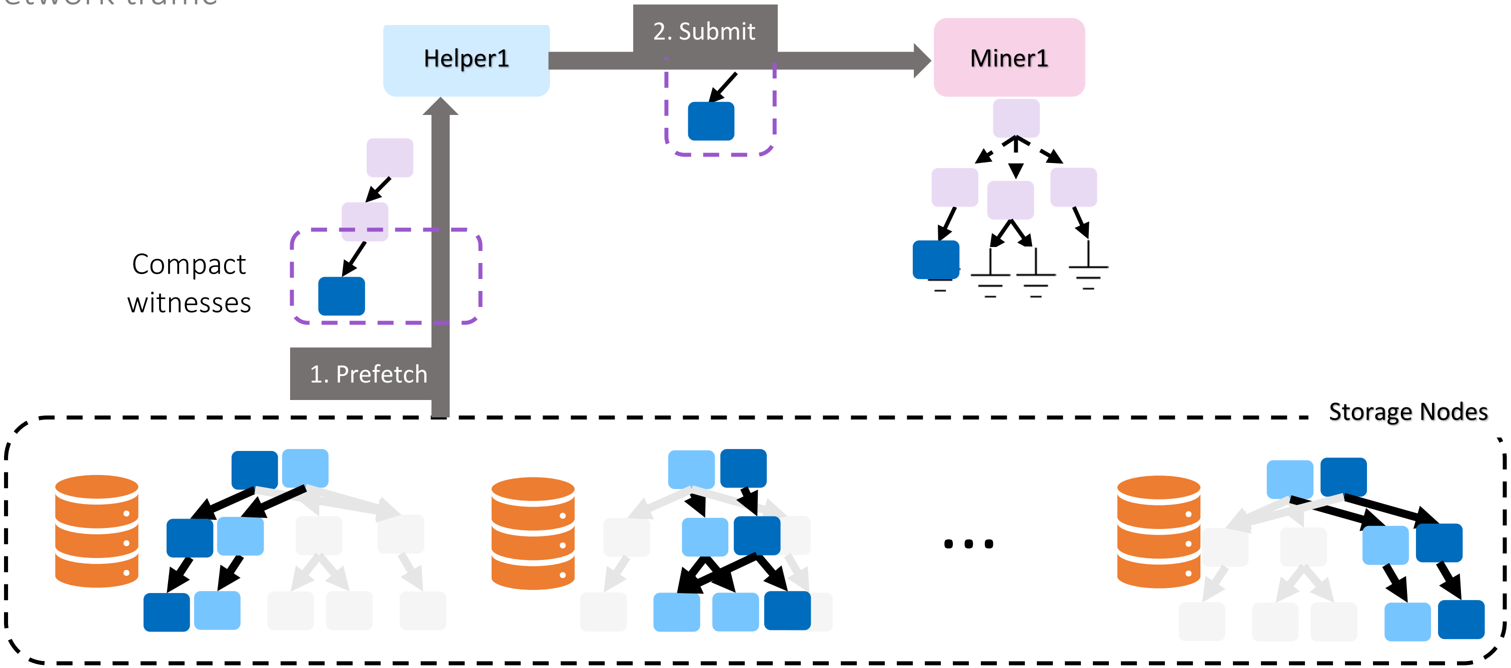
Two Layered DSM-Tree



The DSM-Tree layers collaborate with each other to reduce network traffic

Two Layered DSM-Tree

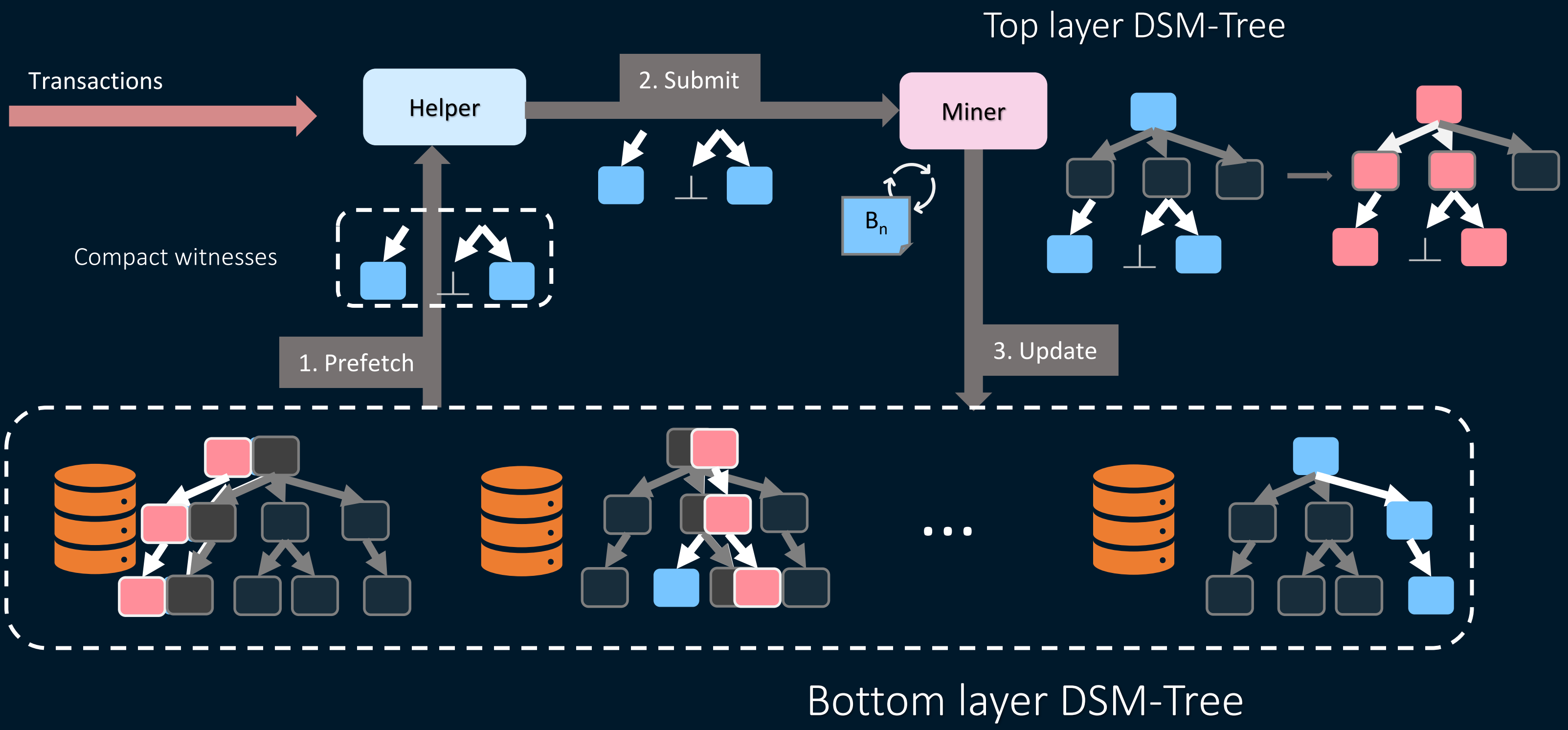
Reduce network traffic



Cross-layer optimizations reduce network traffic by up to 95%

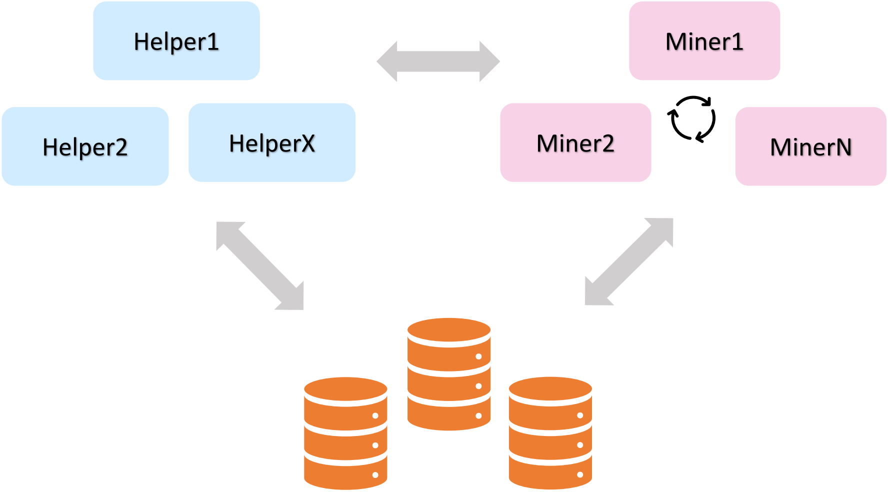
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



Trust, Safety, and Security

- Trust assumptions
  - All components work **without** trust
- Safety and Security
  - PoW remains unchanged
- RainBlock architecture
  - Better throughput and scalability

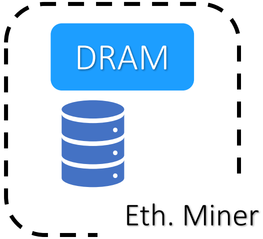


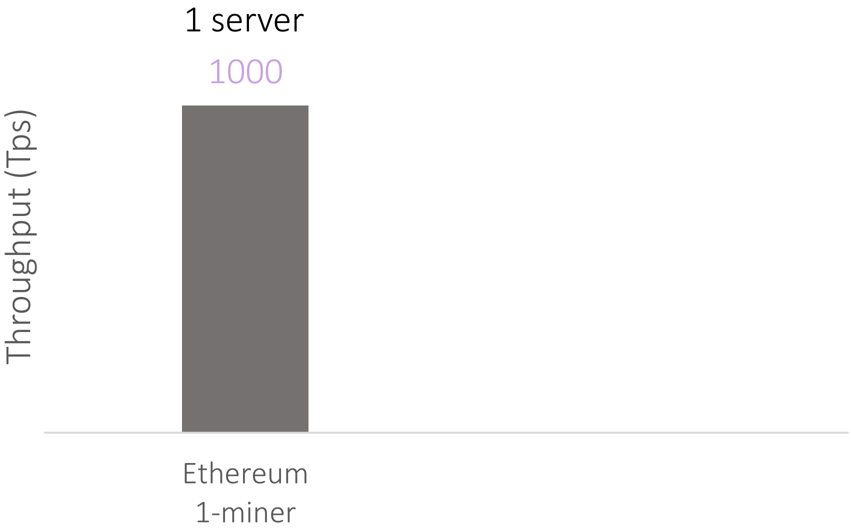
Evaluation

Experimental Setup

- Amazon EC2 m4.2xlarge instances
  - 32GB RAM
  - 48 threads per machine
- Storage nodes, miners, and I/O-Helpers are deployed on their own instance
- Workloads reflecting transactions in the public Ethereum network

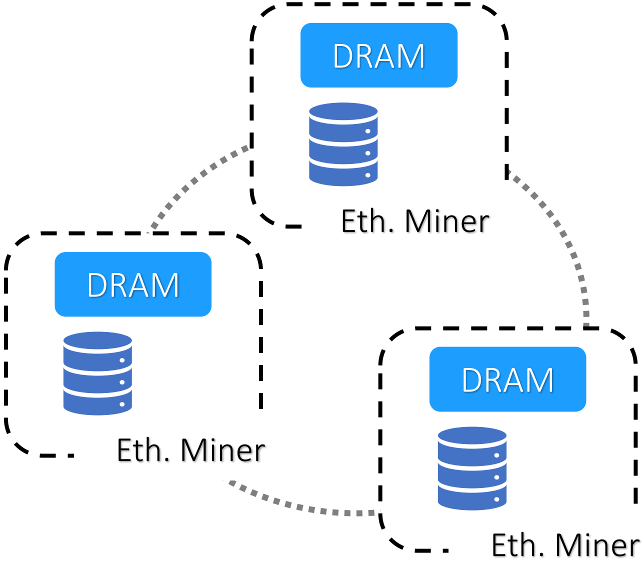
Evaluation

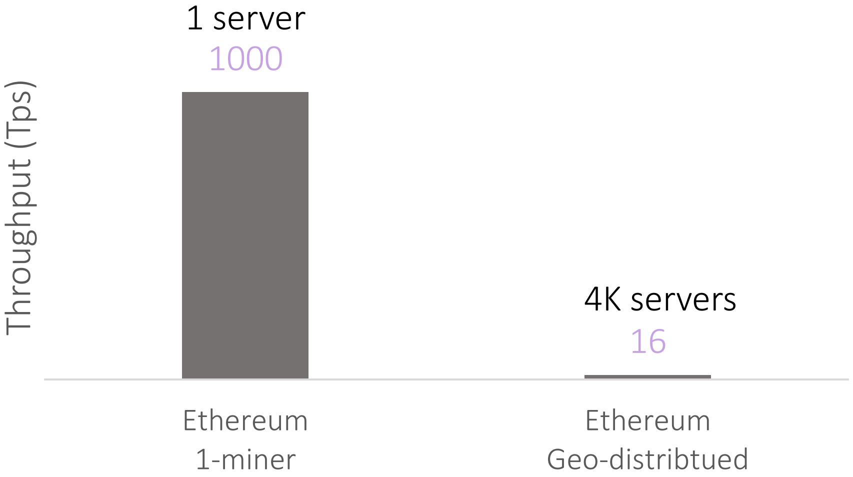




Ethereum 1-miner | Performance Breakdown

Evaluation

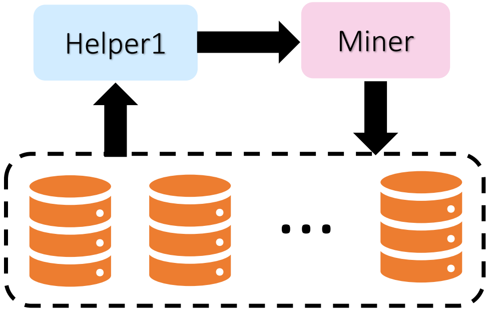


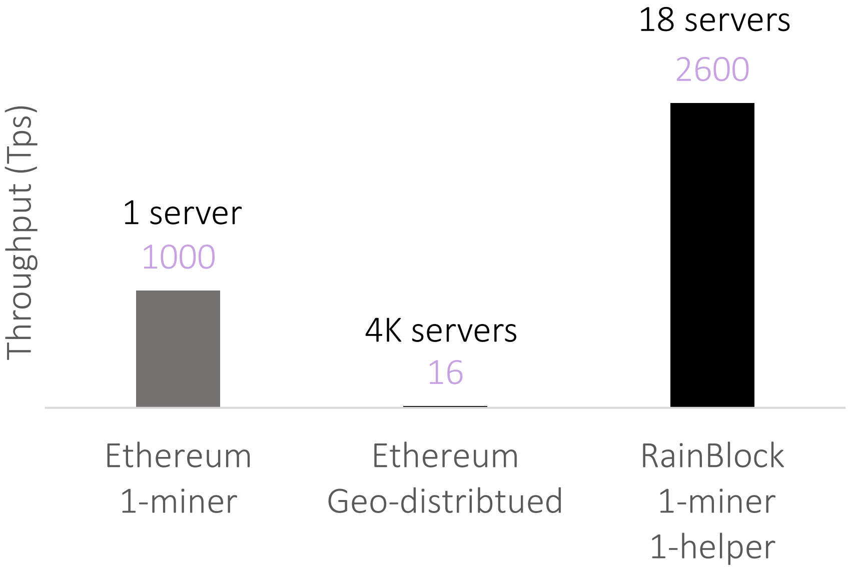


Geo-distributed Ethereum

Performance Breakdown

Evaluation

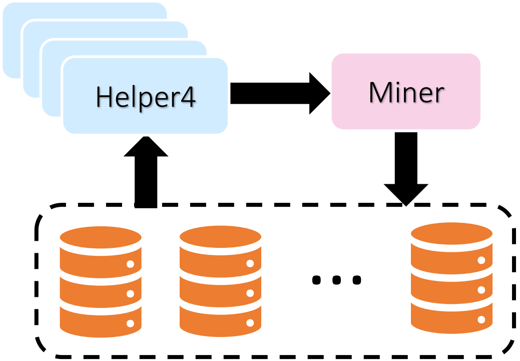


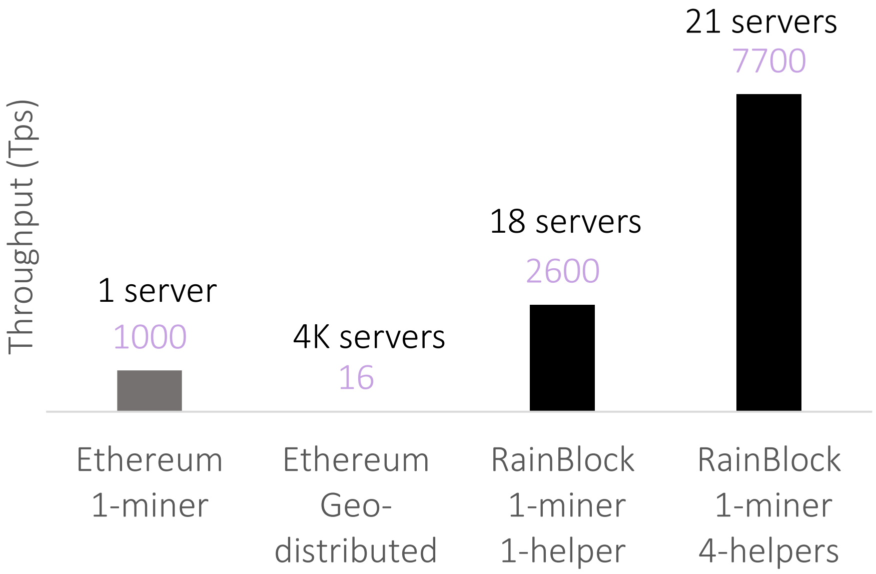


RainBlock 1-miner, 1-helper, 16-storage nodes

Performance Breakdown

Evaluation

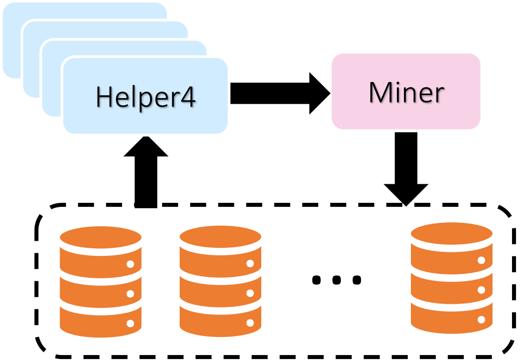




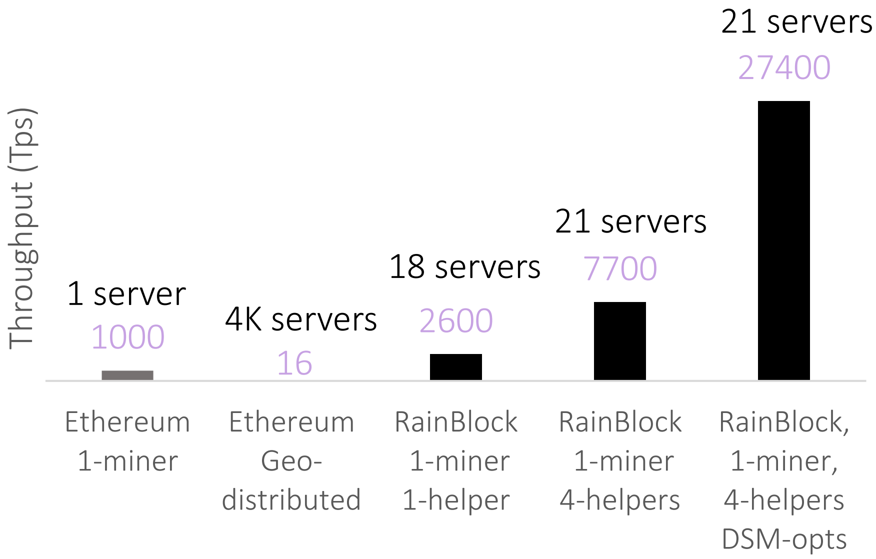
RainBlock 1-miner, 4-helpers, 16-storage nodes

Performance Breakdown

Evaluation



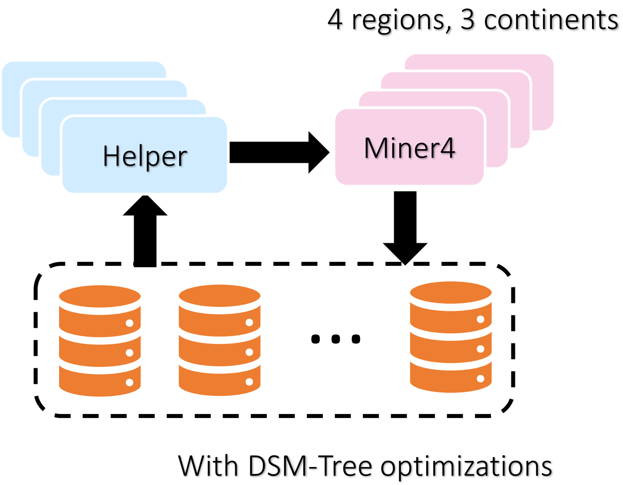
With DSM-Tree optimizations

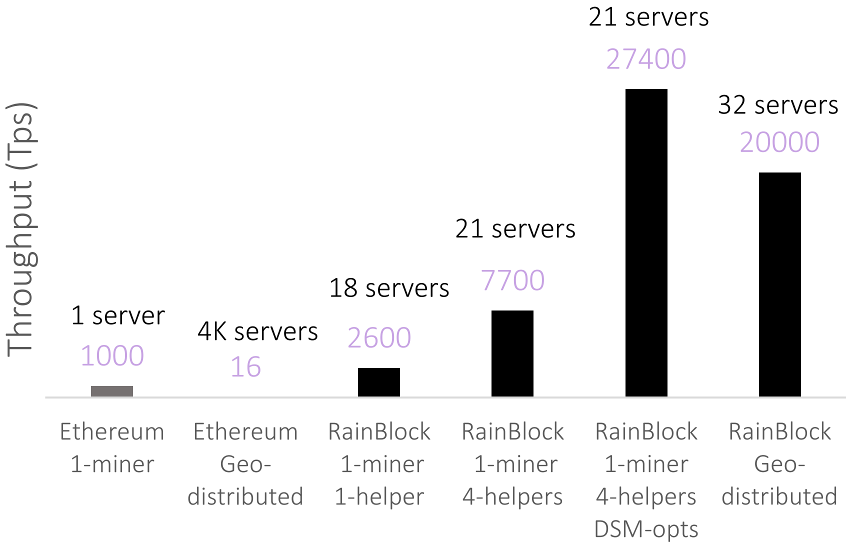


RainBlock 1-miner, 4-helpers, 16-storage nodes

Performance Breakdown

Evaluation





Geo-distributed RainBlock

Performance Breakdown

Summary

I/O bottlenecks limit the block size not proof-of-work consensus

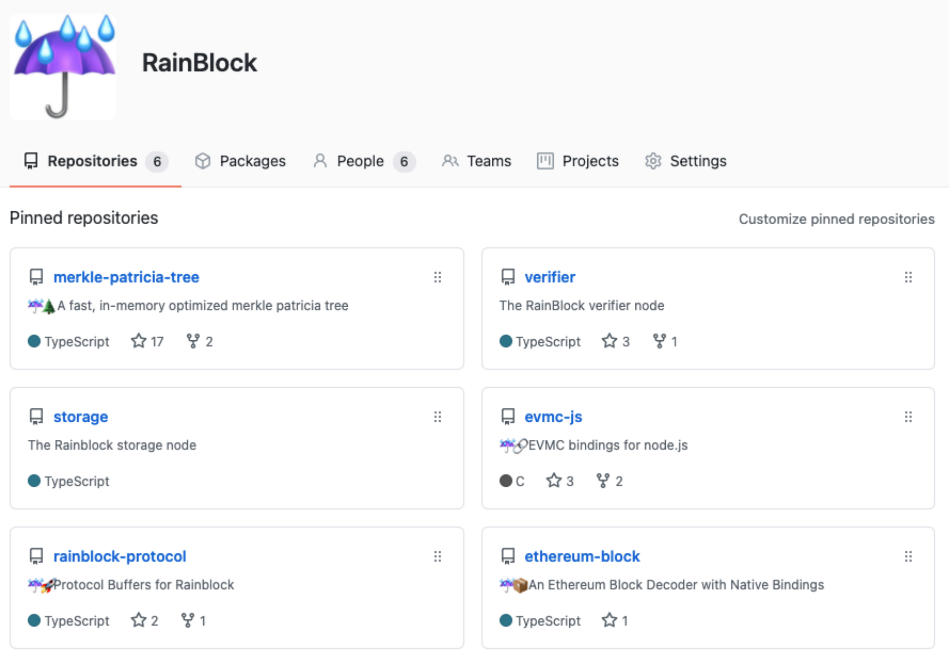
RainBlock avoids I/O in the critical path with I/O-Helpers and storage nodes

RainBlock uses DSM-Trees to reduce network traffic

RainBlock processes 20K tps in a geo-distributed setting

RainBlock: Faster Transaction Processing in Public Blockchains

https://github.com/RainBlock



RainBlock

Repositories 6 | Packages | People 6 | Teams | Projects | Settings

Pinned repositories

Customize pinned repositories

merkle-patricia-tree
A fast, in-memory optimized merkle patricia tree
● TypeScript ☆ 17 2

verifier
The RainBlock verifier node
● TypeScript ☆ 3 1

storage
The Rainblock storage node
● TypeScript

evmc-js
EVMC bindings for node.js
● C ☆ 3 2

rainblock-protocol
Protocol Buffers for Rainblock
● TypeScript ☆ 2 1

ethereum-block
An Ethereum Block Decoder with Native Bindings
● TypeScript ☆ 1

soujanya@cs.utexas.edu