RainBlock: Faster Transaction Processing in Public Blockchains

Soujanya Ponnapalli[1], Aashaka Shah[1], Souvik Banerjee[1]*,
Dahlia Malkhi[3], Amy Tai[2], Vijay Chidambaram[1,2], and Michael Wei[2]

1

2

vmware®

3

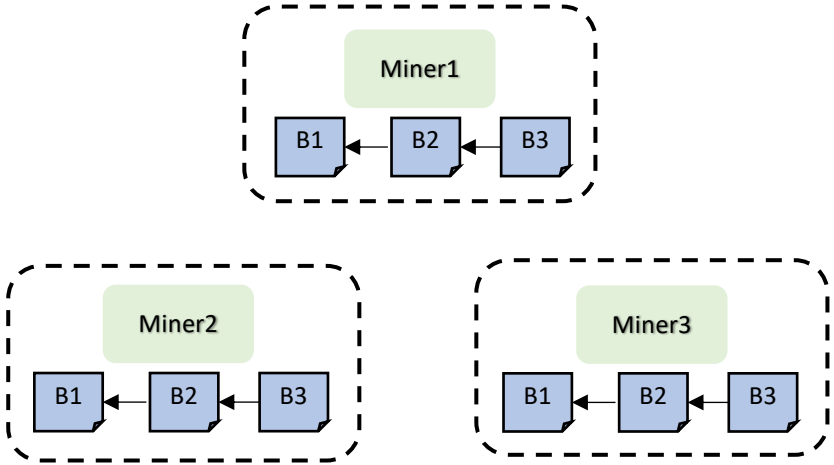
novi ≋diem

* Currently at Apple; work done while at UT Austin

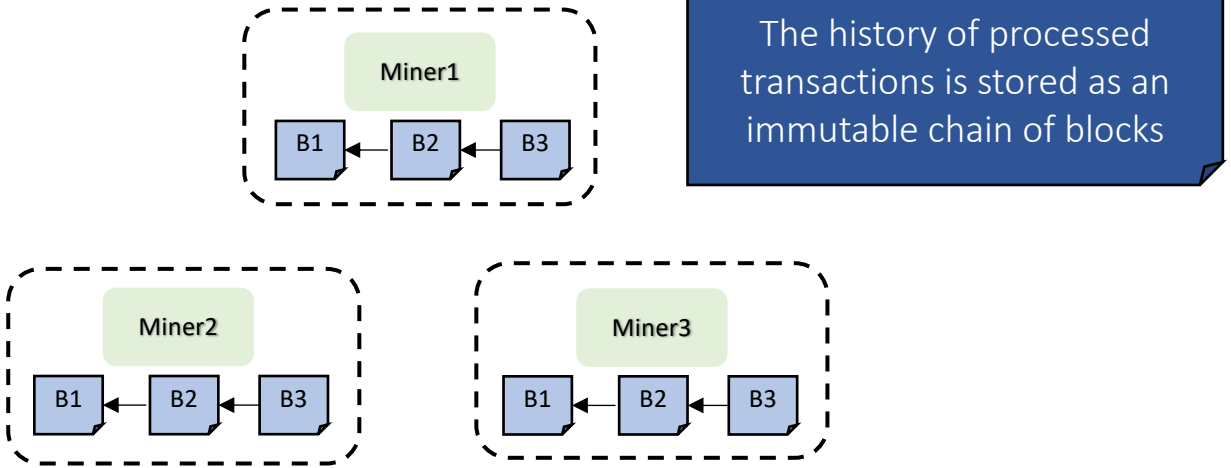
Blockchains: Decentralized Databases

Blockchains: Decentralized Databases

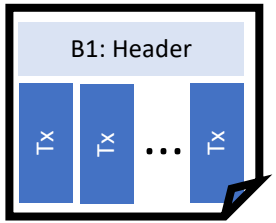
Blockchains: Decentralized Databases

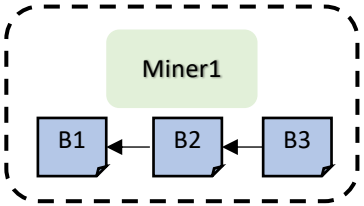


Blockchains: Decentralized Databases

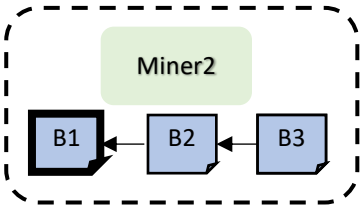


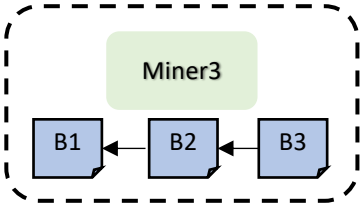
Blockchains: Decentralized Databases



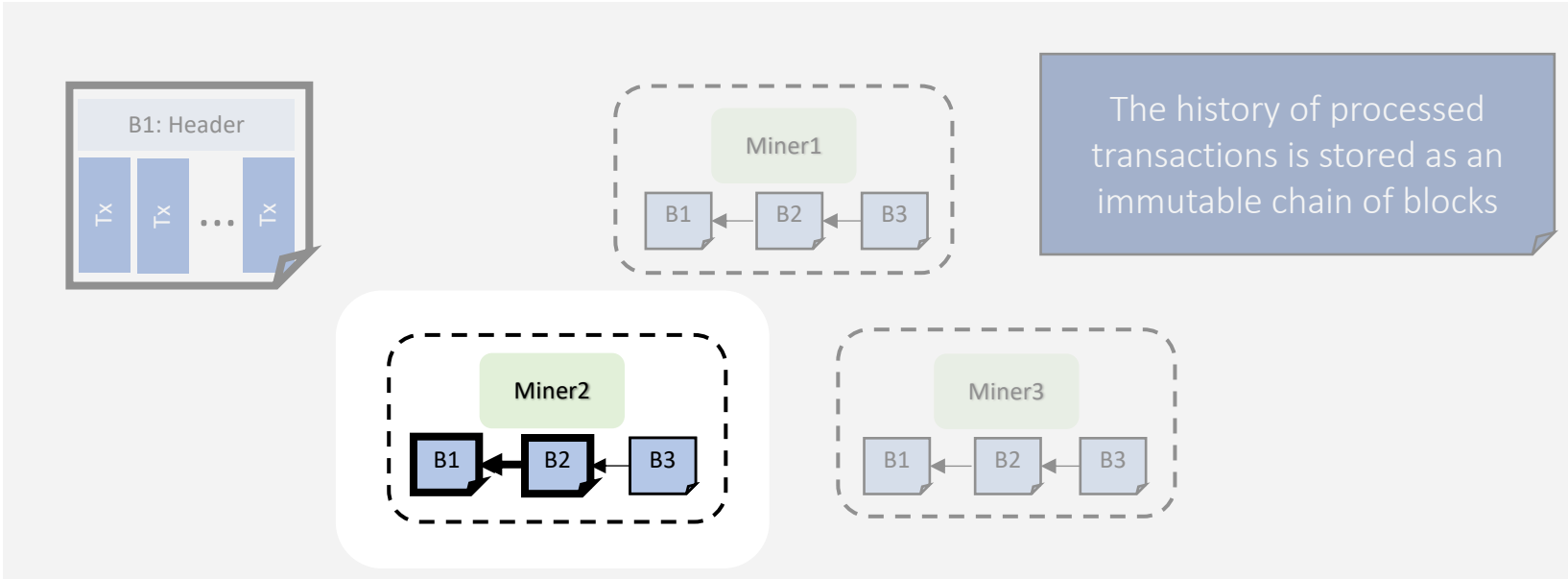


The history of processed transactions is stored as an immutable chain of blocks

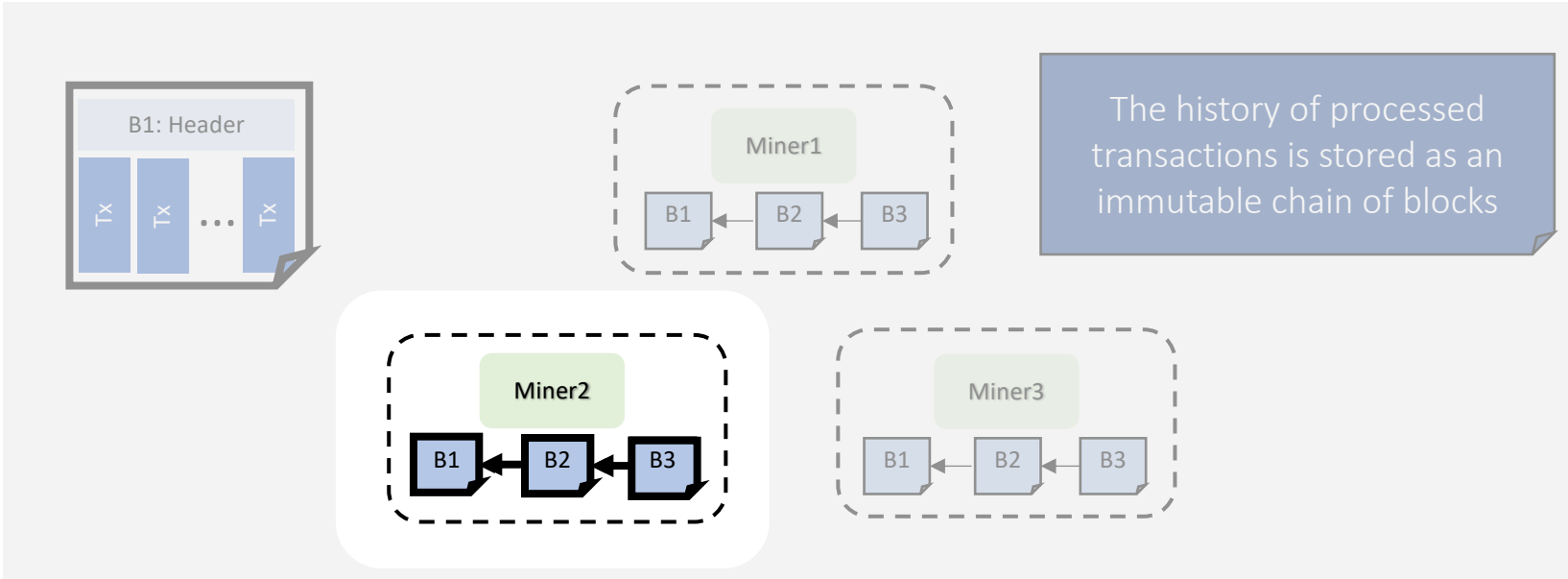




Blockchains: Decentralized Databases



Blockchains: Decentralized Databases



Public Blockchains: Proof-of-work consensus

Open networks

Public Blockchains: Proof-of-work consensus

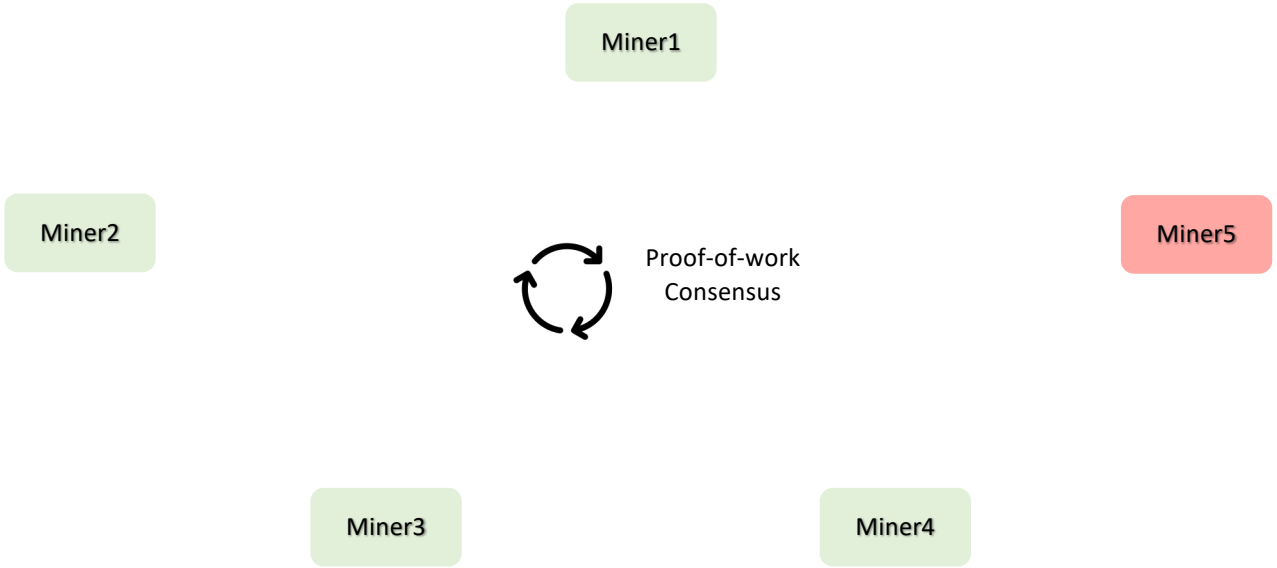
Open networks

Public Blockchains: Proof-of-work consensus

Open networks

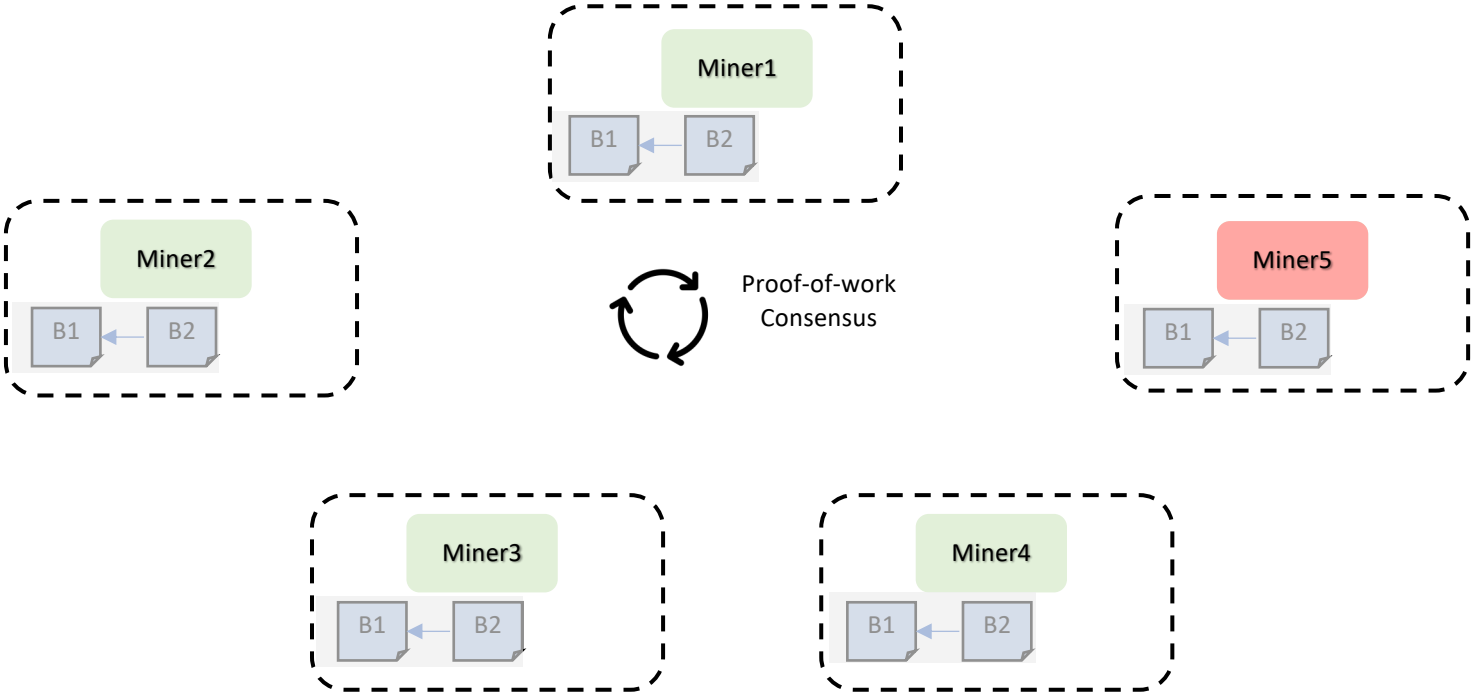
Public Blockchains: Proof-of-work consensus

Open networks



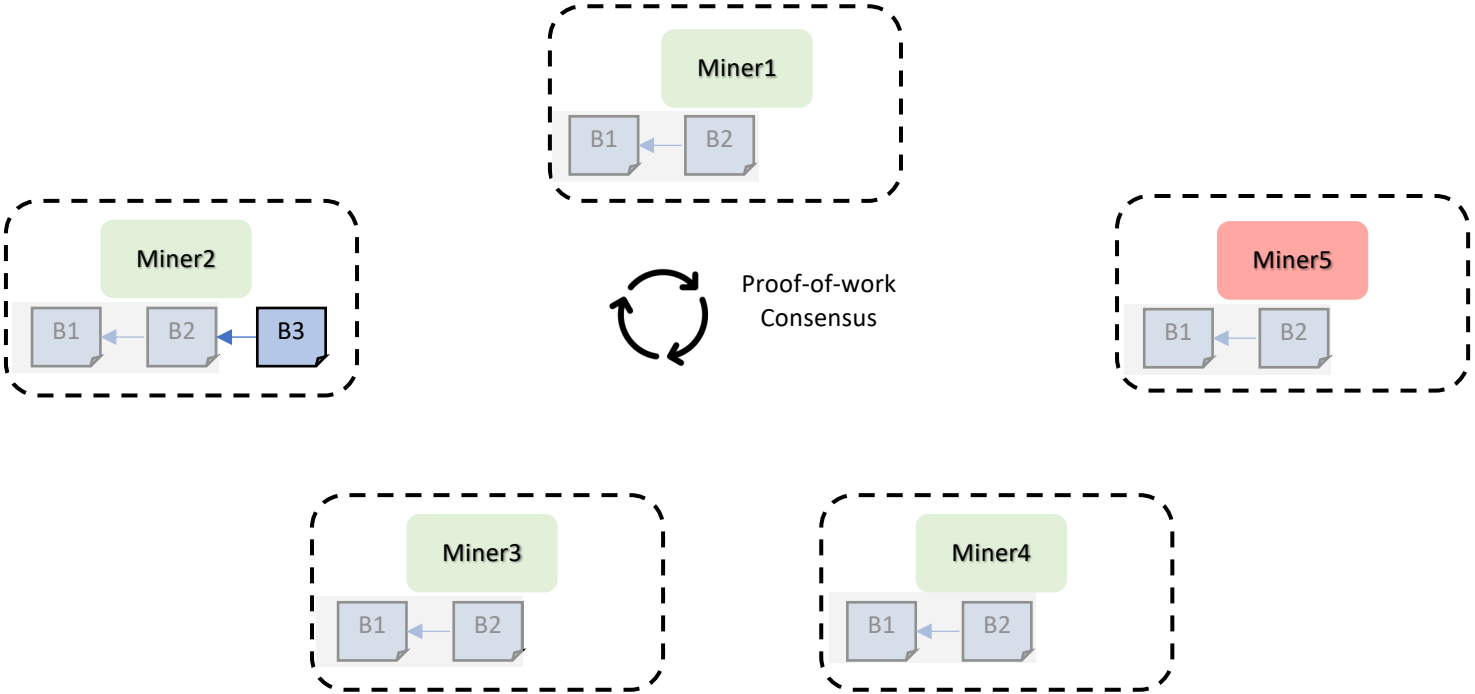
Public Blockchains: Proof-of-work consensus

Open networks



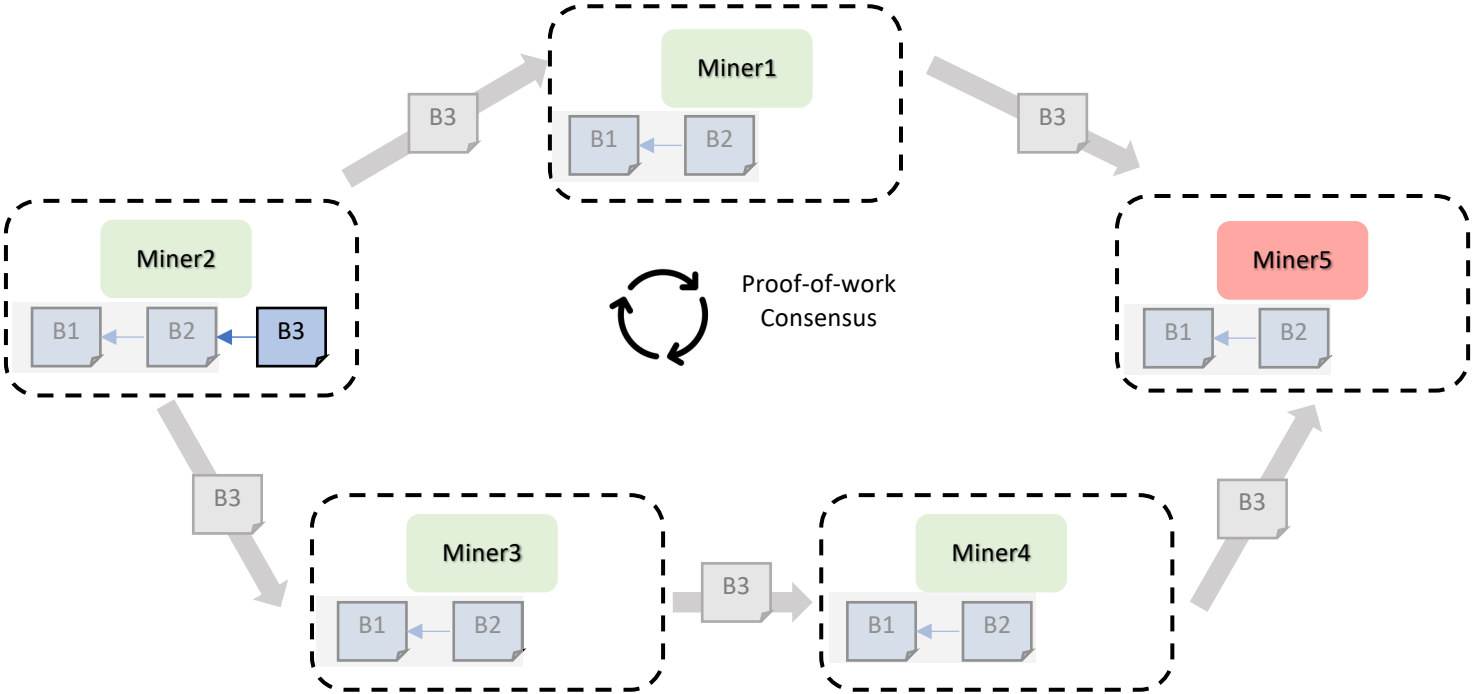
Public Blockchains: Proof-of-work consensus

Open networks



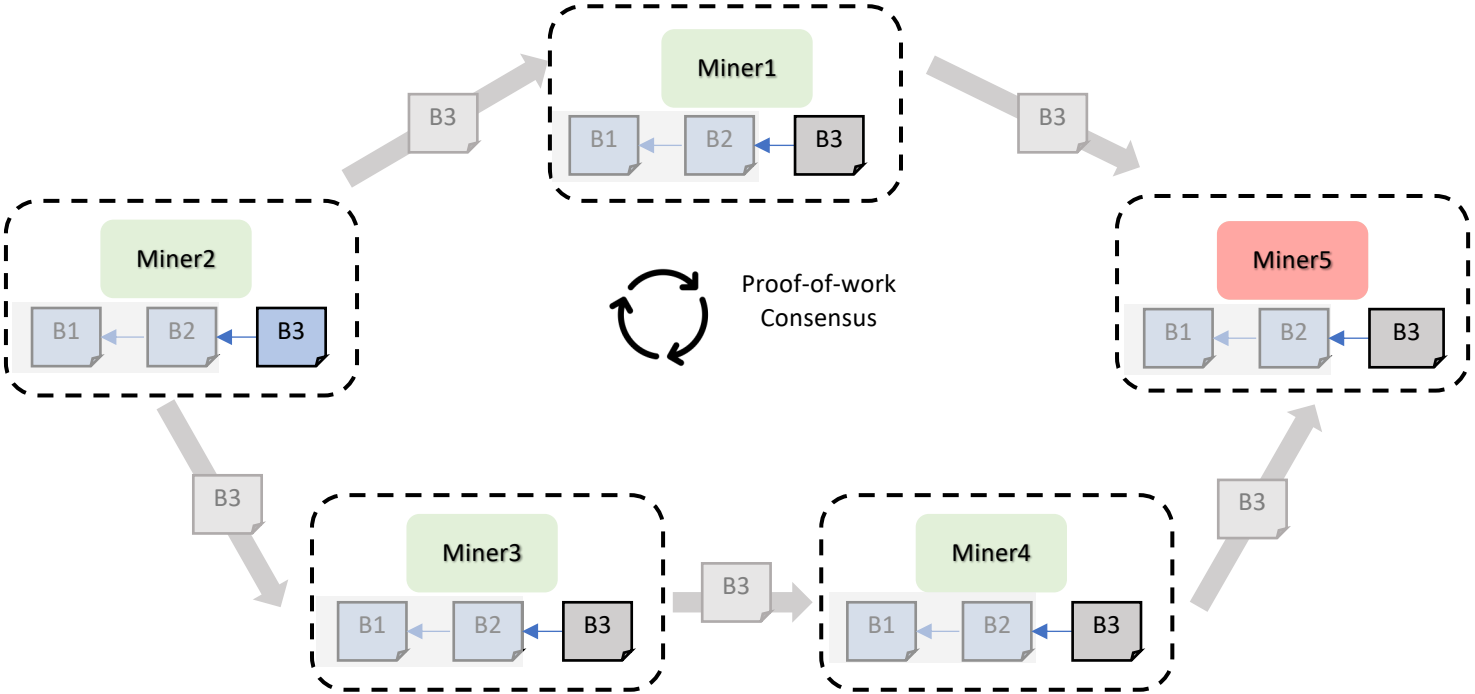
Public Blockchains: Proof-of-work consensus

Open networks



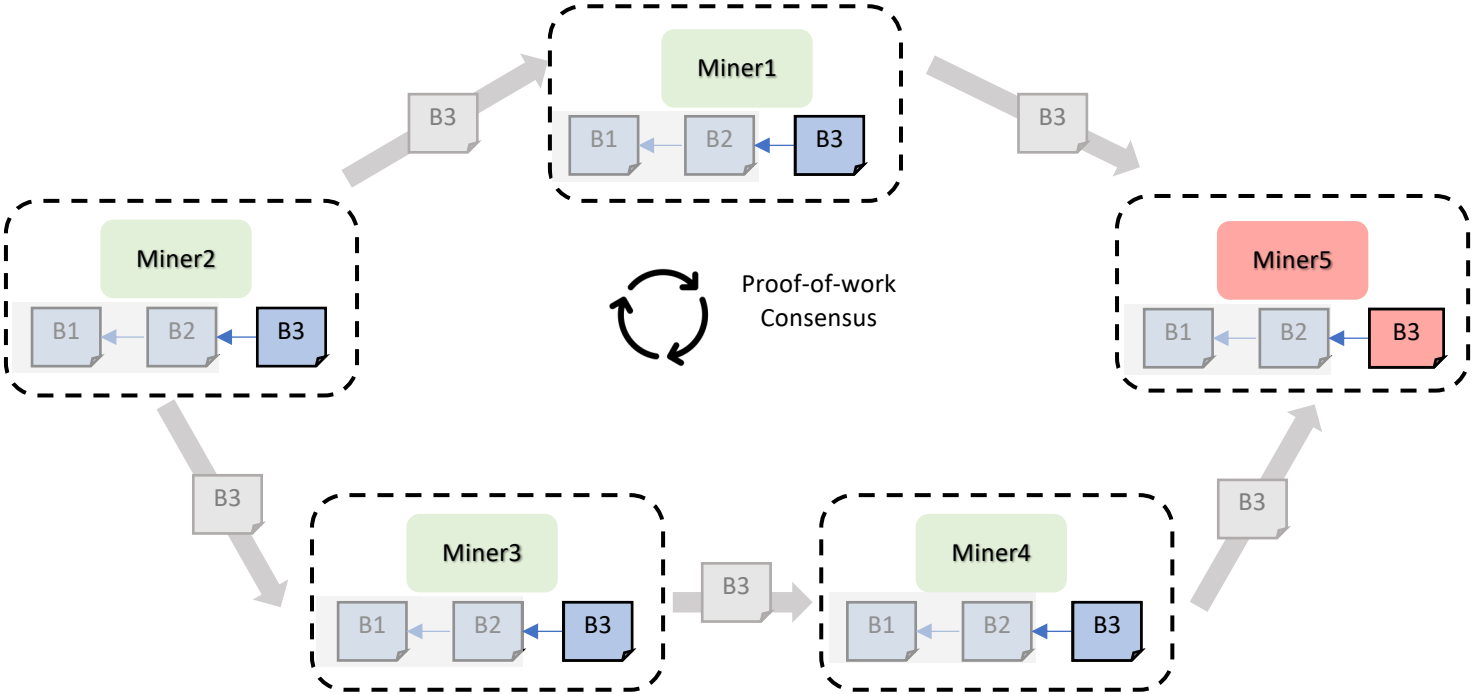
Public Blockchains: Proof-of-work consensus

Open networks



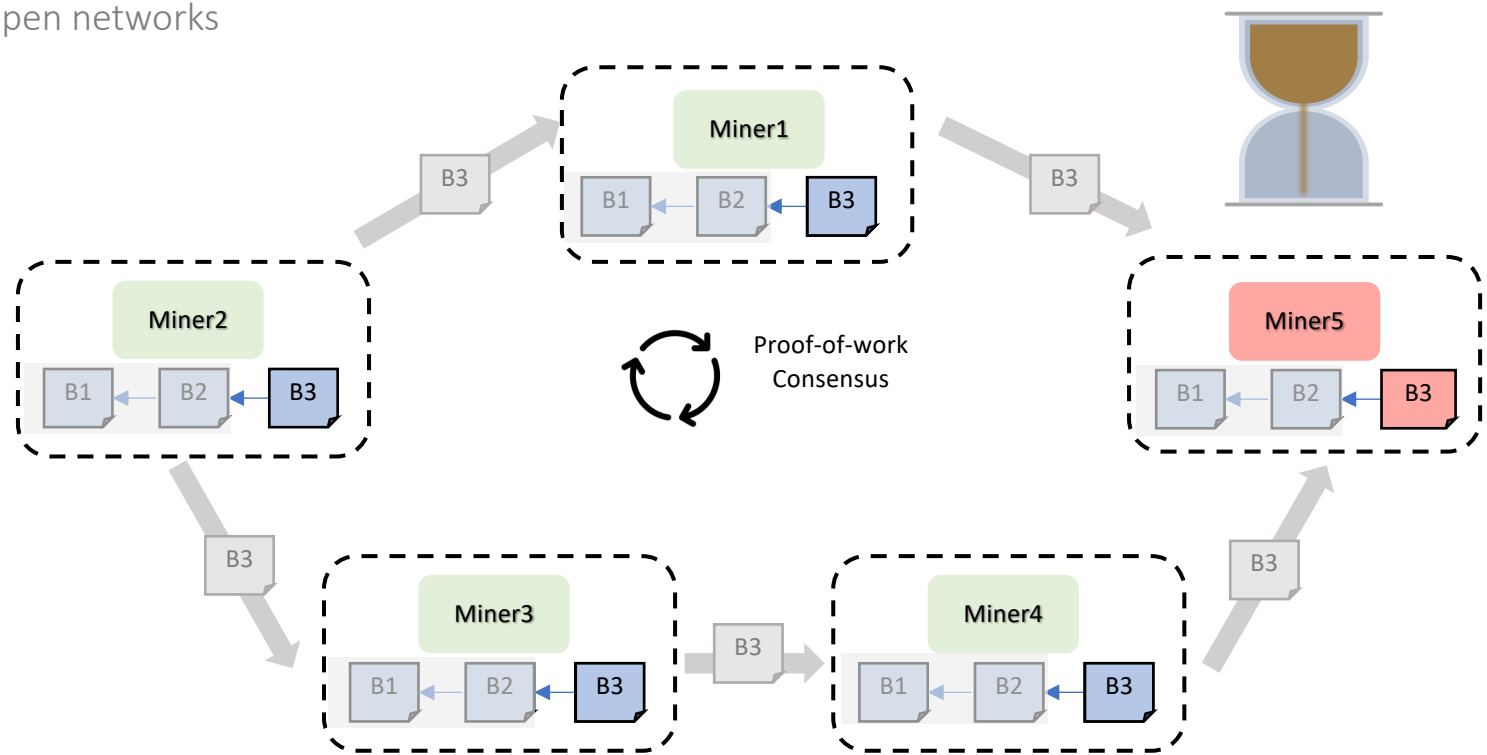
Public Blockchains: Proof-of-work consensus

Open networks



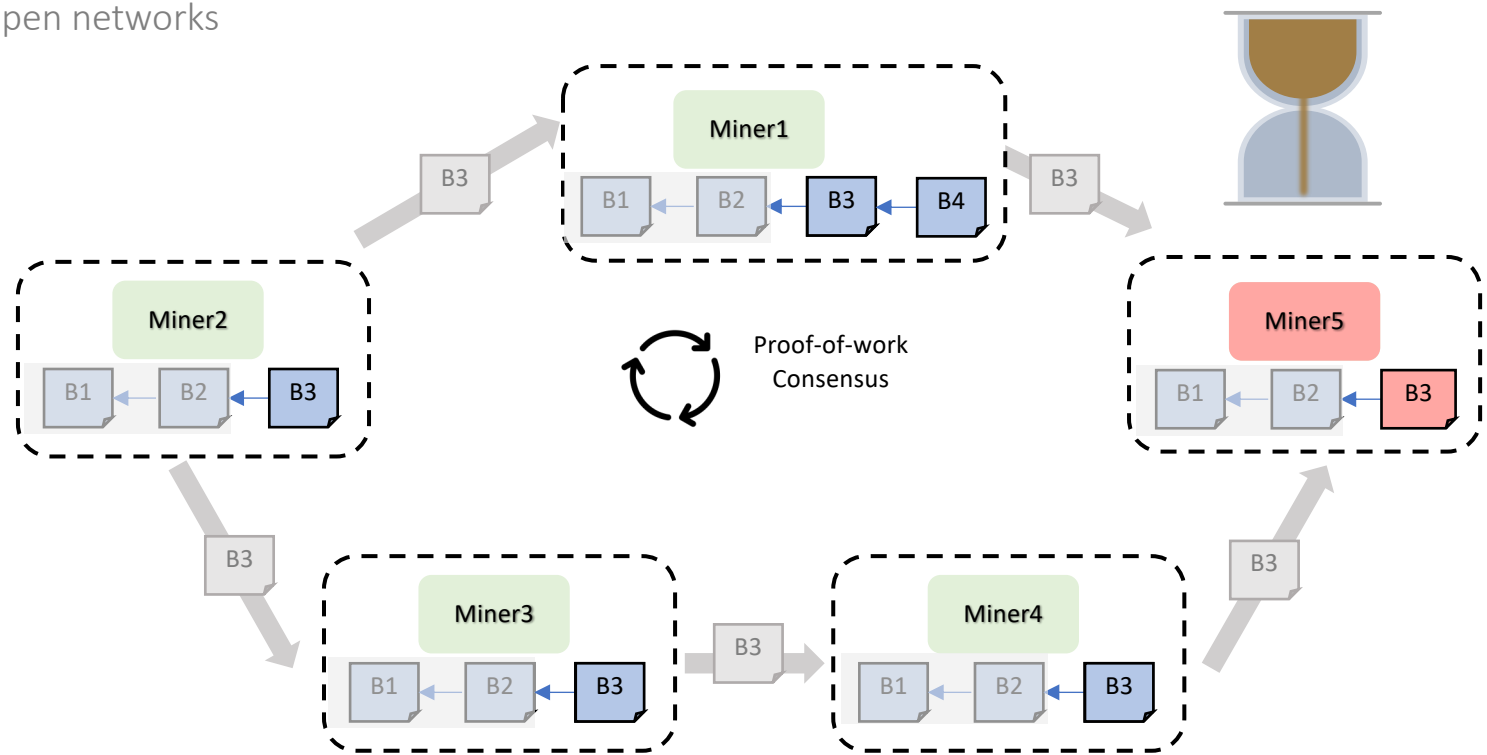
Public Blockchains: Proof-of-work consensus

Open networks



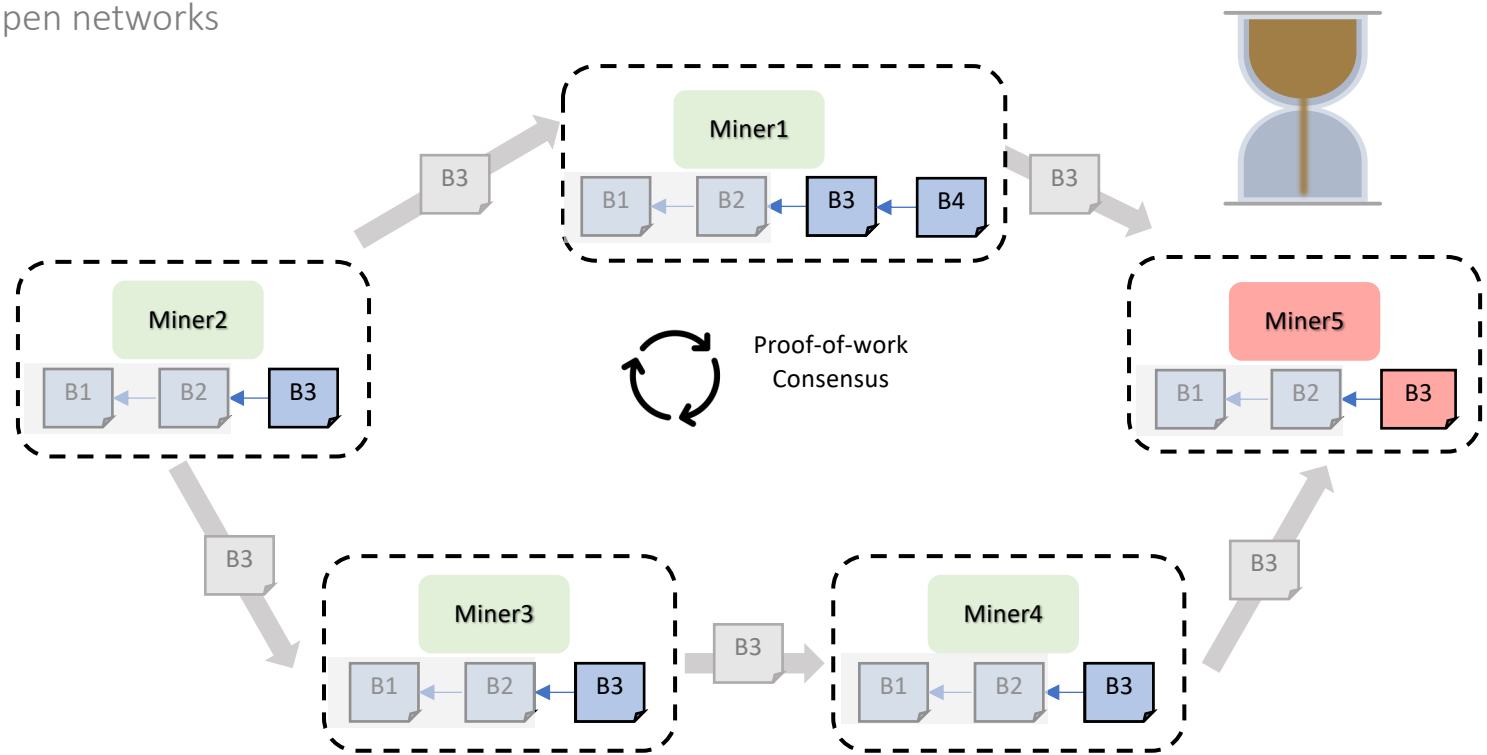
Public Blockchains: Proof-of-work consensus

Open networks



Public Blockchains: Proof-of-work consensus

Open networks



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

Public Blockchains – Low throughput

Public Blockchains – Low throughput

Throughput: 20 tps

Public Blockchains – Low throughput

Throughput: 20 tps

Throughput: 16 tps

Public Blockchains – Low throughput

Throughput: 20 tps

Throughput: 16 tps

Throughput: 24K tps
(1000x higher)

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: Modify PoW or New Consensus

Prior work: Modify PoW or New 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

Prior work: Modify PoW or New 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

Prior work: Modify PoW or New 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

⋮

Prior work: Modify PoW or New 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?

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?

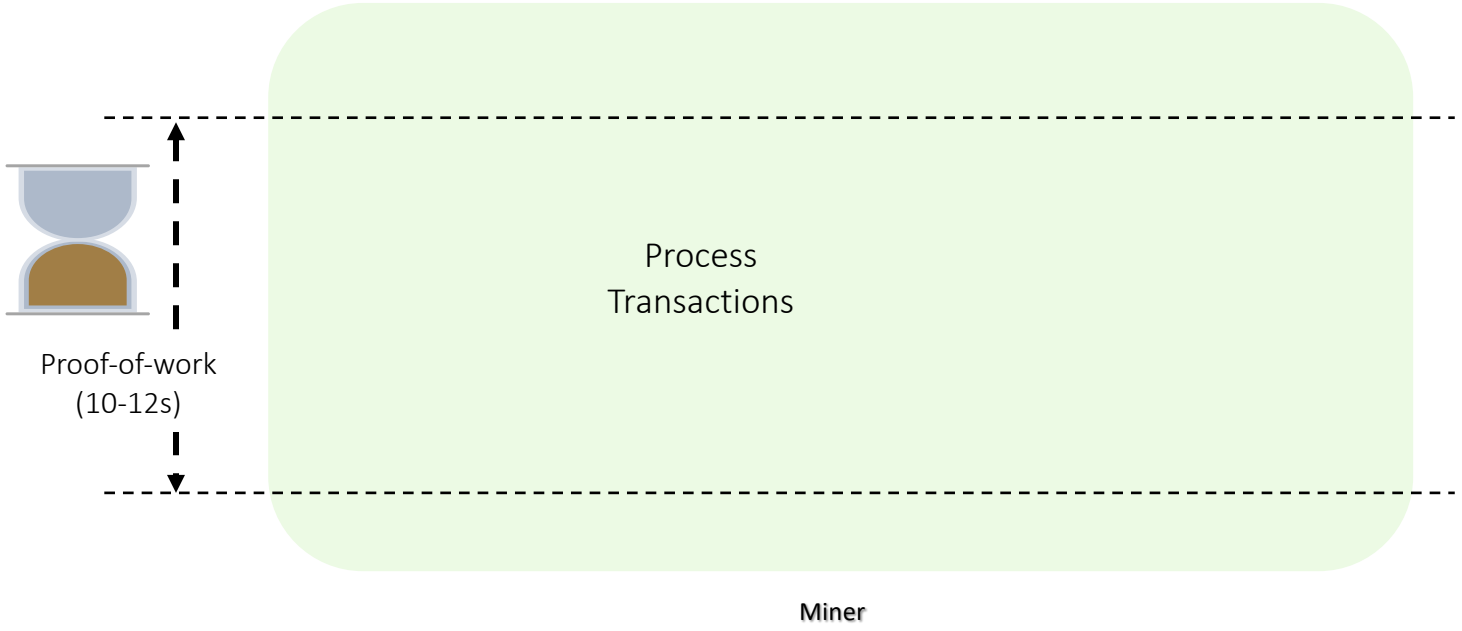
Safety and security from proof-of-work

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

Safety and security from proof-of-work

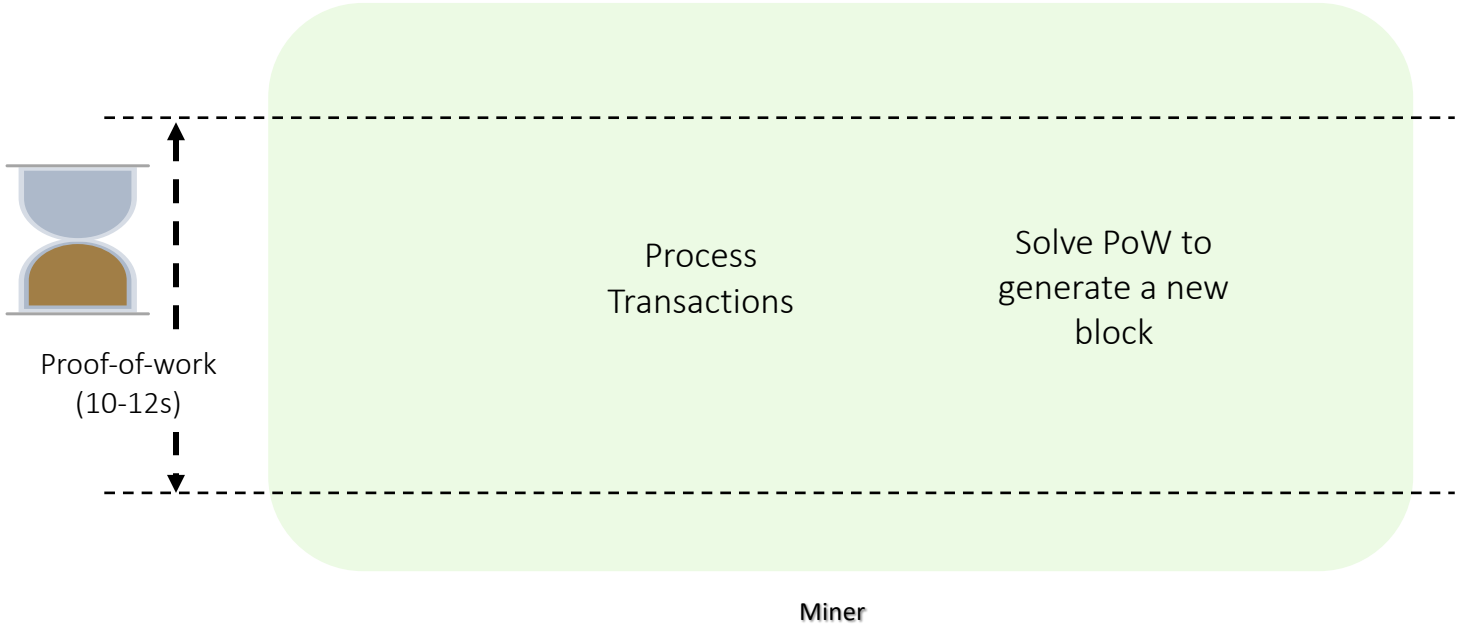
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



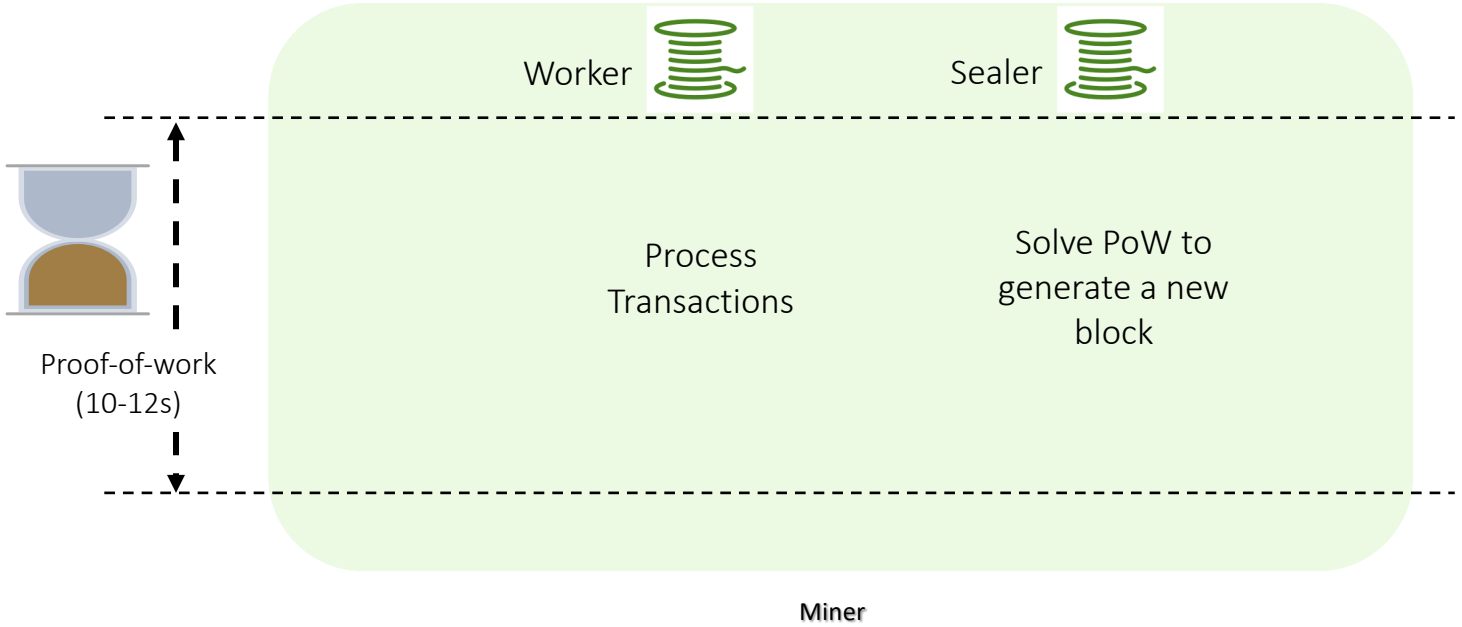
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



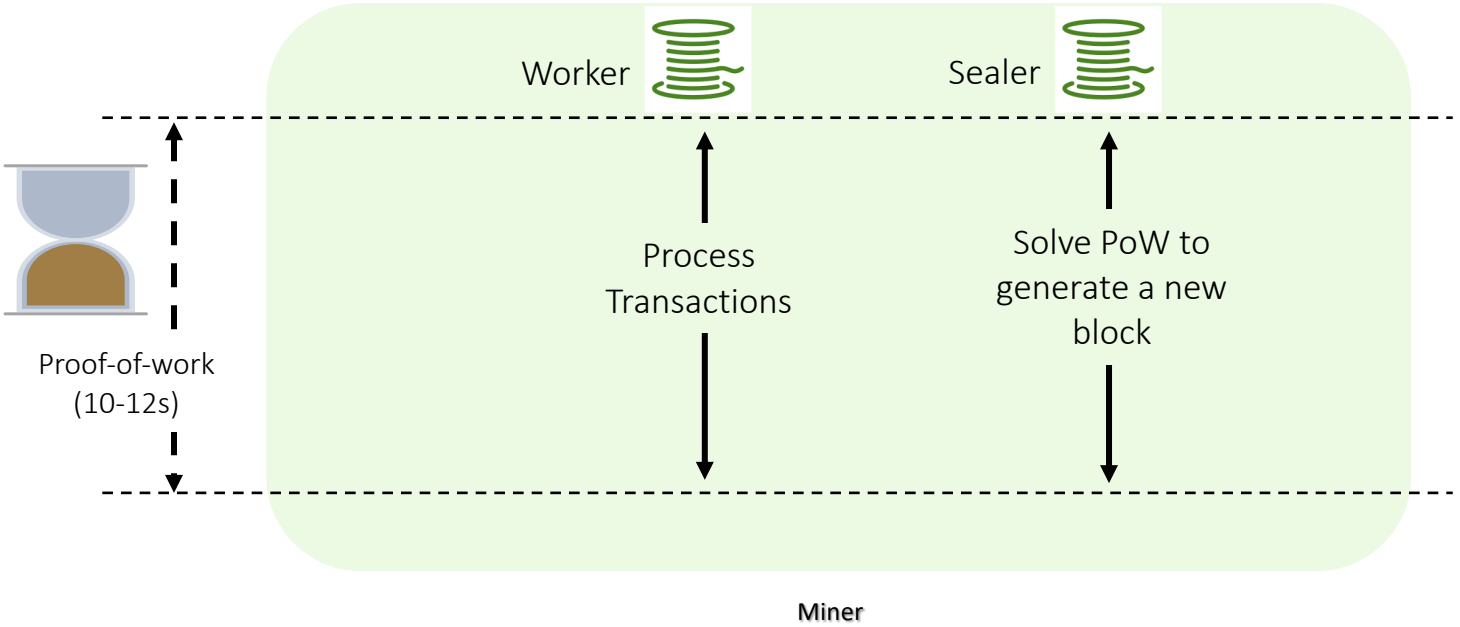
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



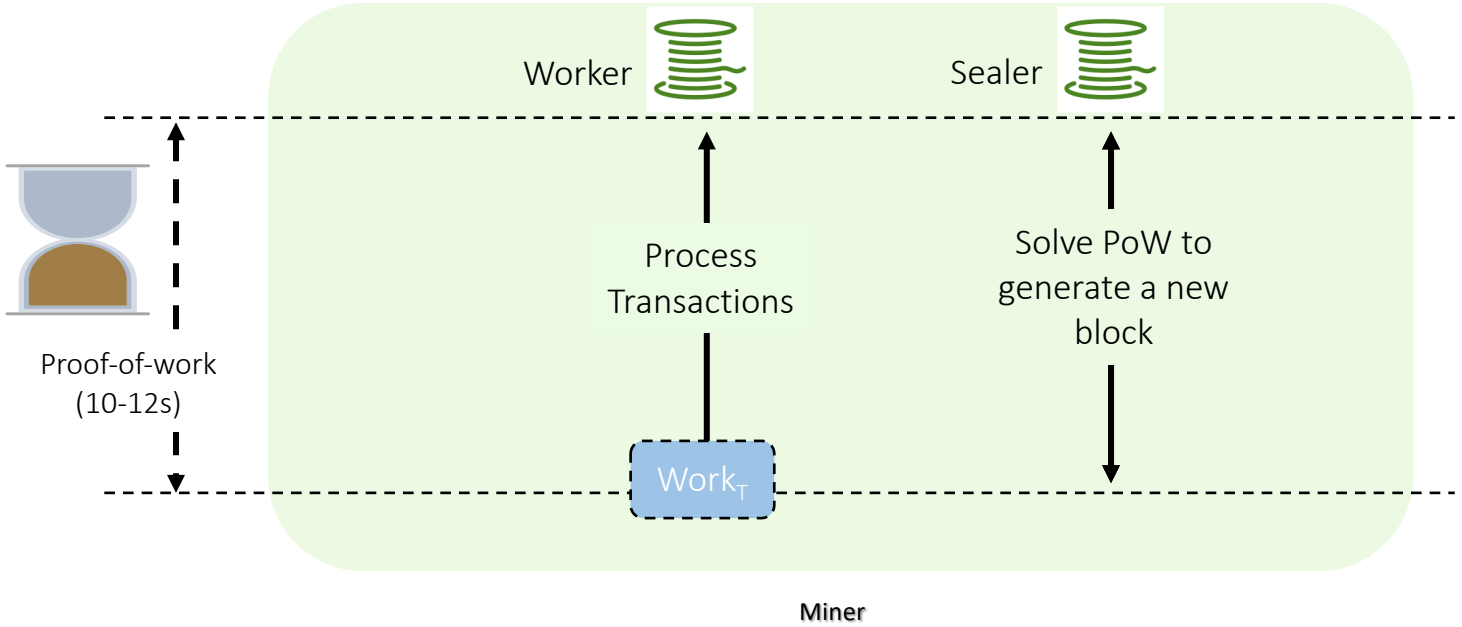
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



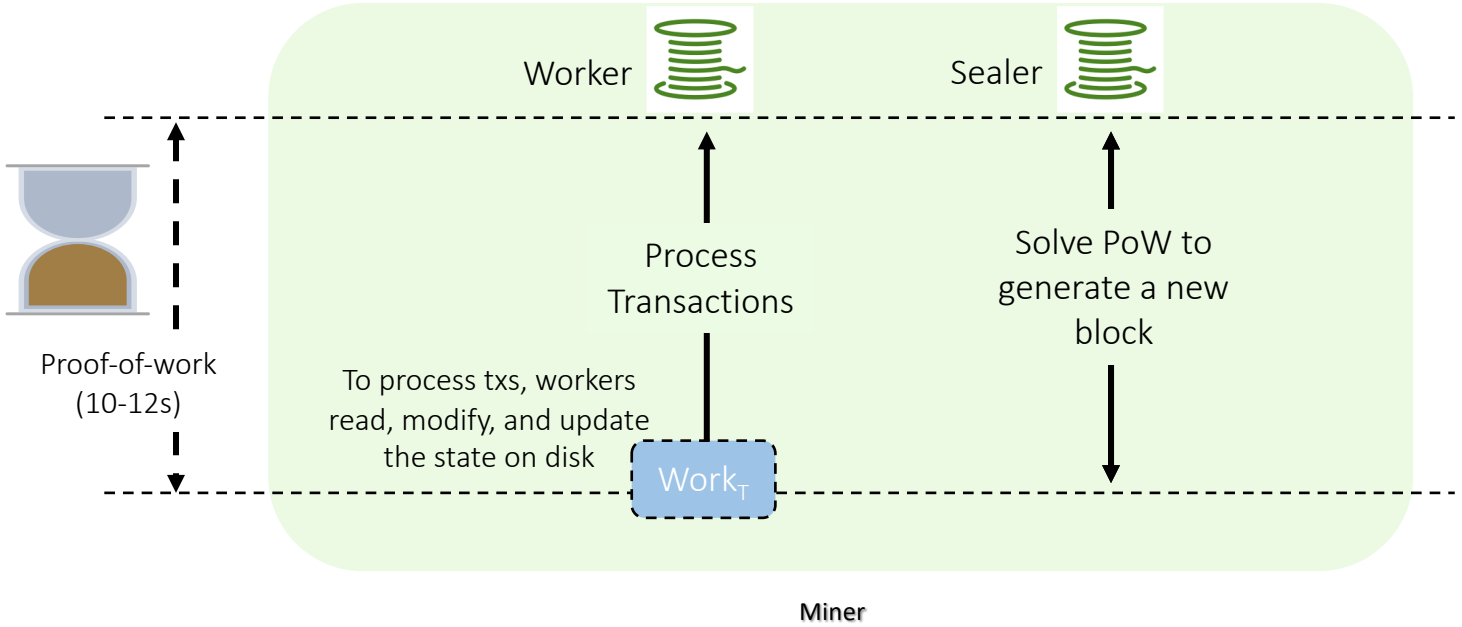
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



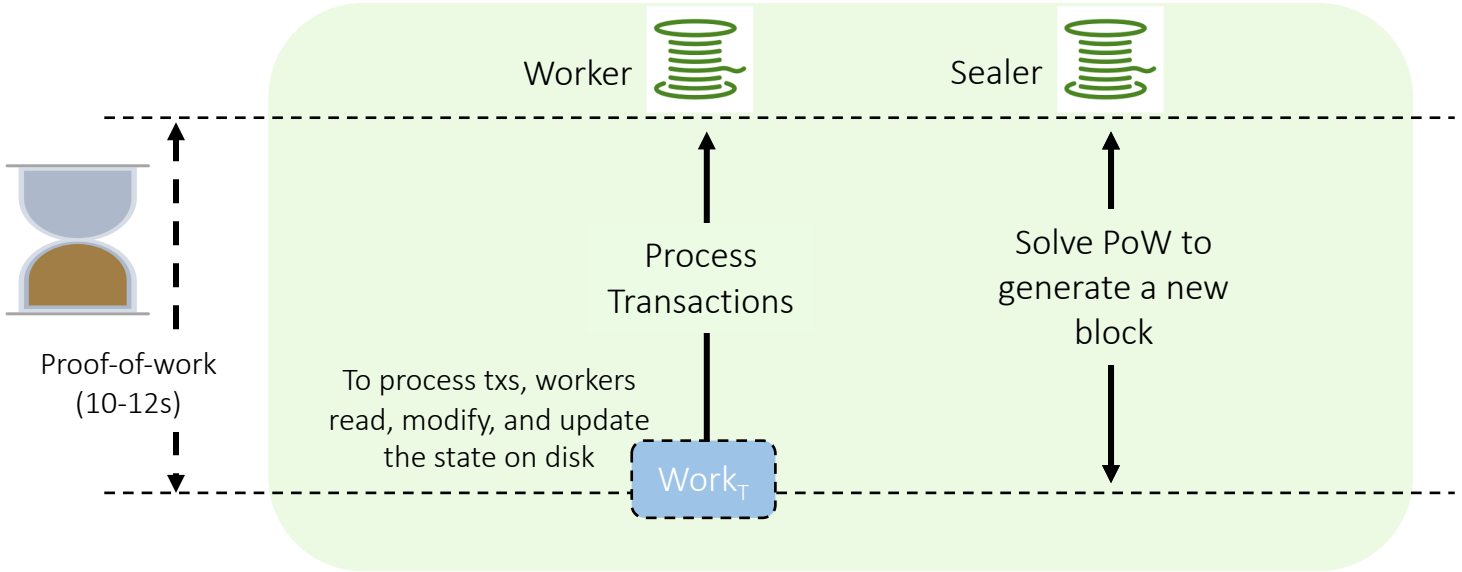
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



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

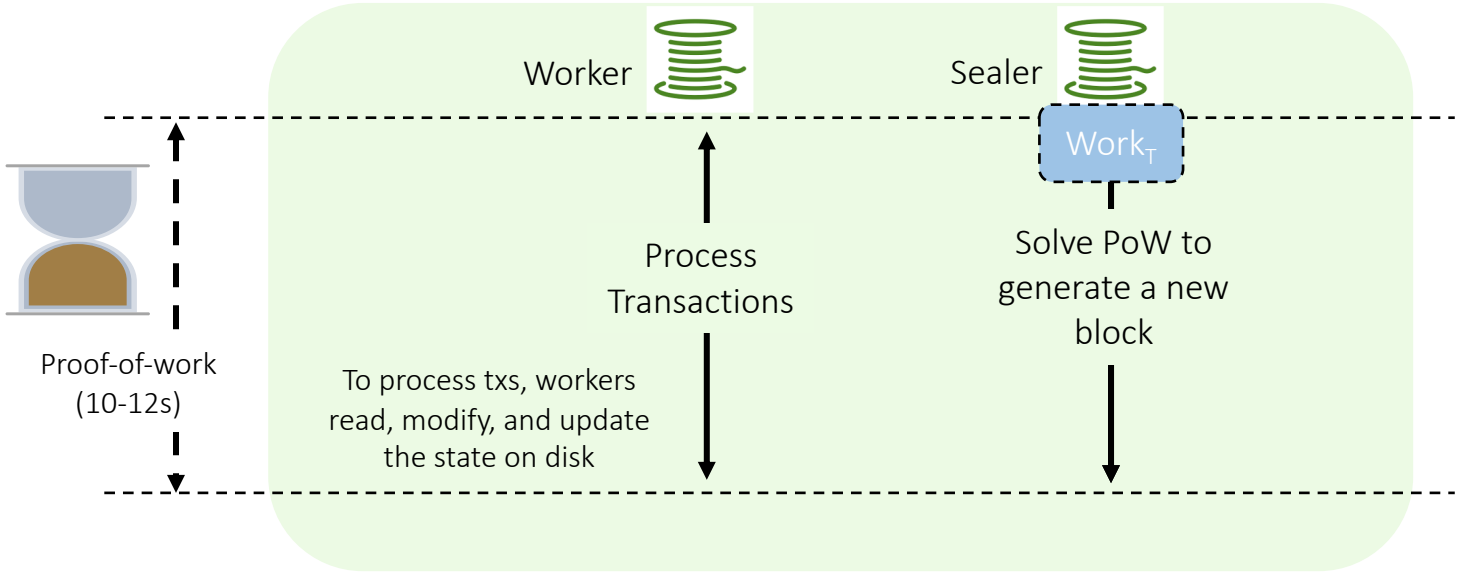
Safety and security from proof-of-work



Workers perform I/O for processing txs

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

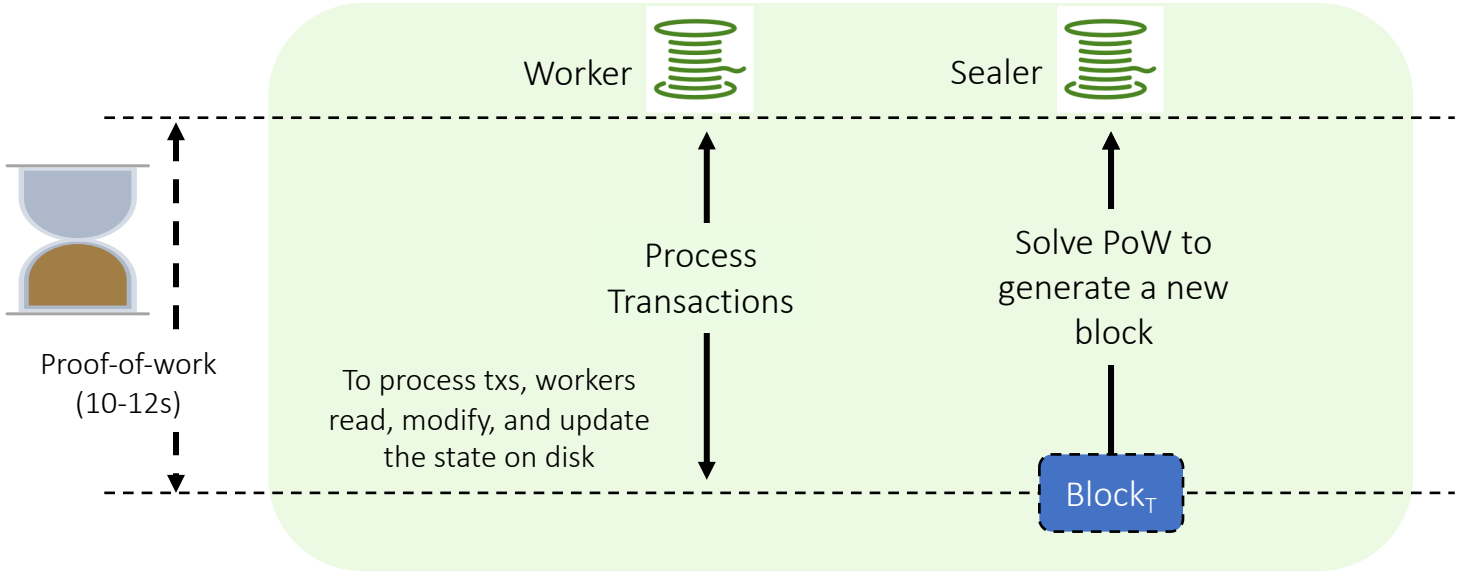
Safety and security from proof-of-work



Workers perform I/O for processing txs

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

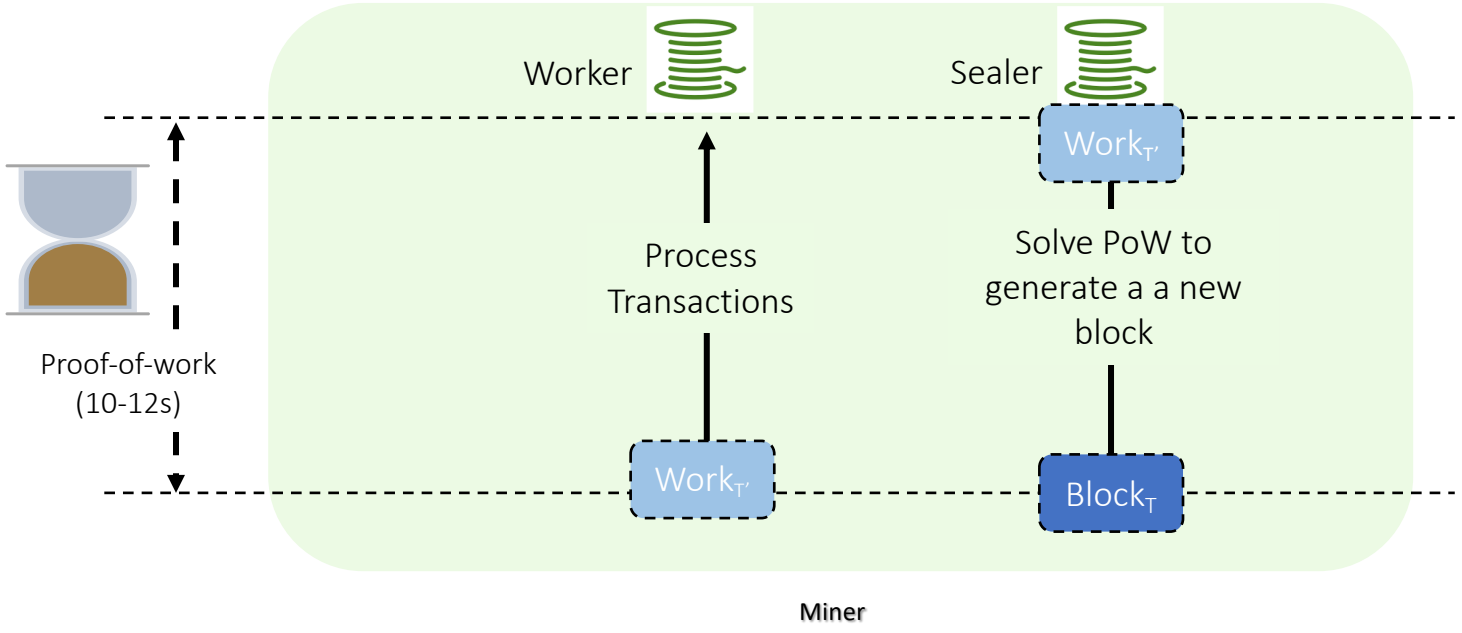
Safety and security from proof-of-work



Workers perform I/O for processing txs

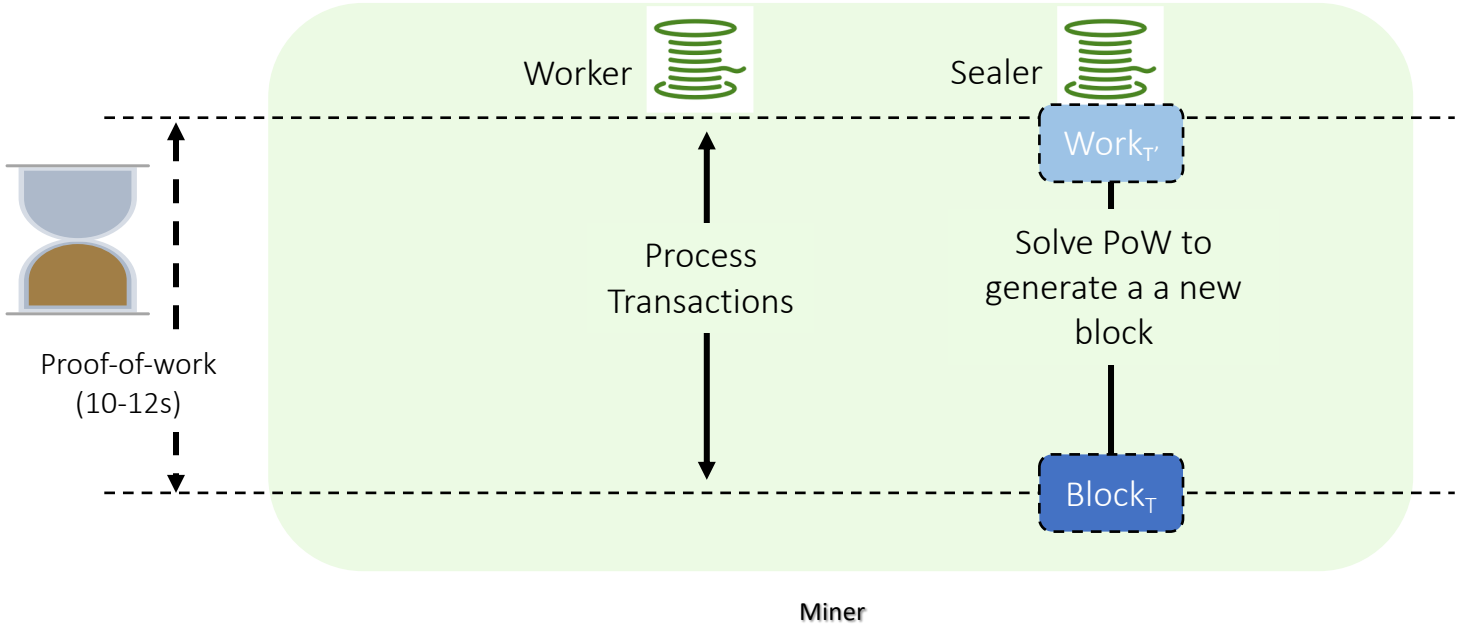
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



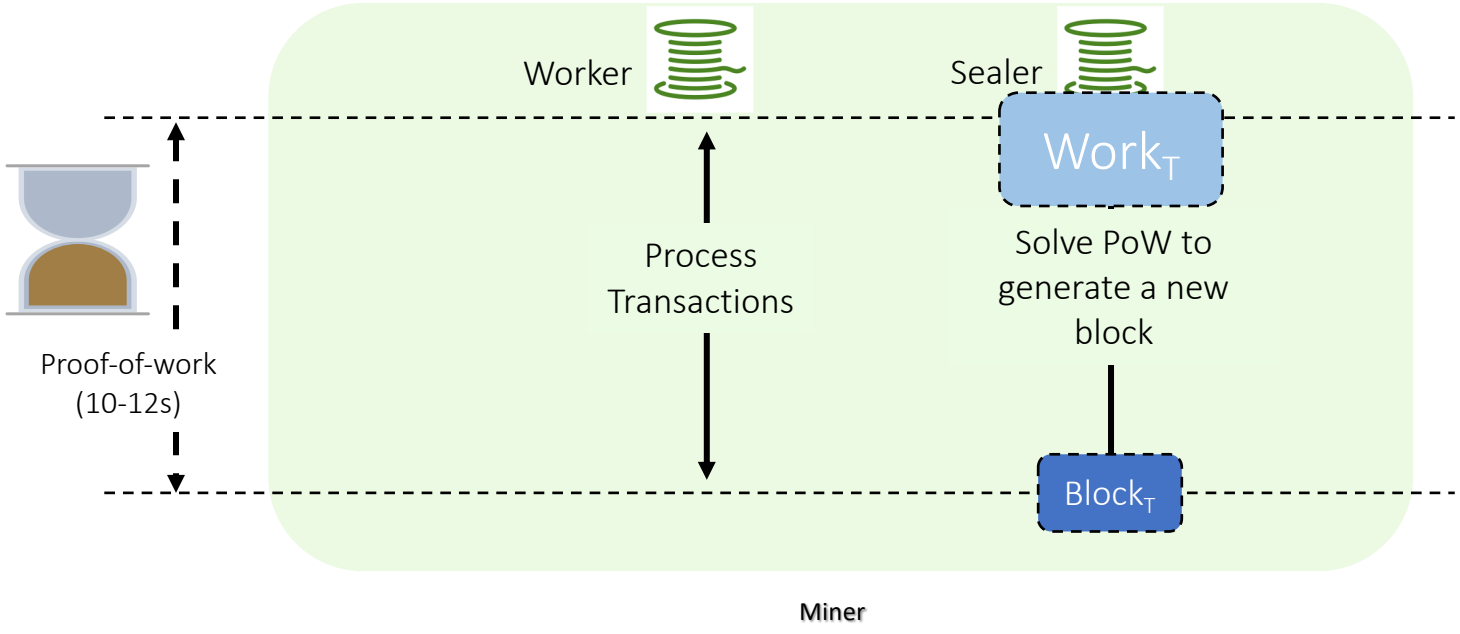
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



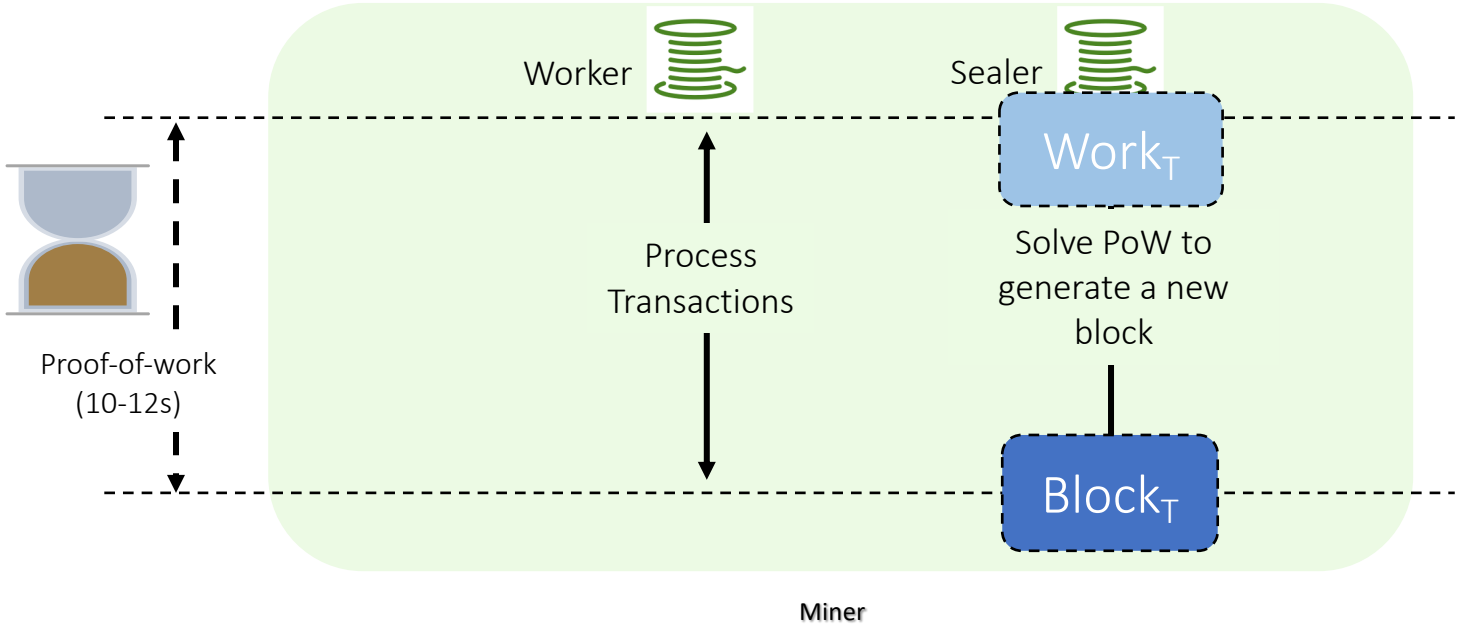
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

Safety and security from proof-of-work



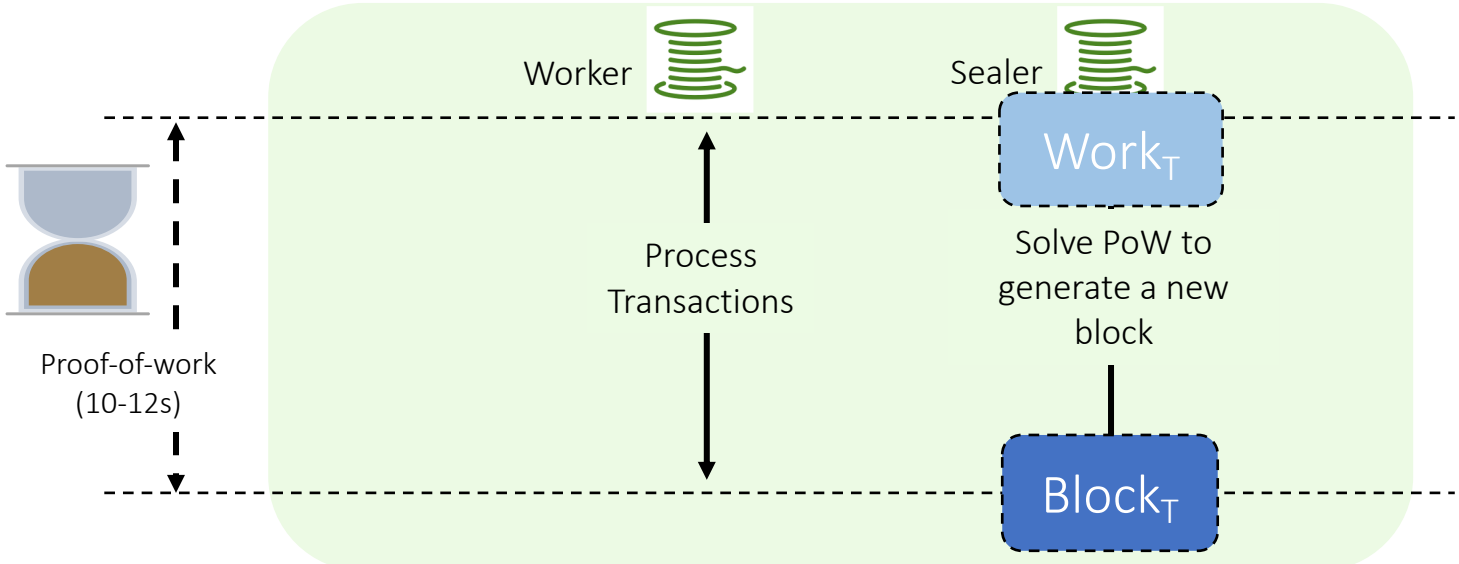
Can we Increase the Throughput of Public Blockchains Without Modifying PoW?

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



Network latency of propagating a larger block is not the dominant cost [1]

https://ethresear.ch/t/increasing-eth-s-gas-limit-what-we-can-safely-do-today/8121

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!

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

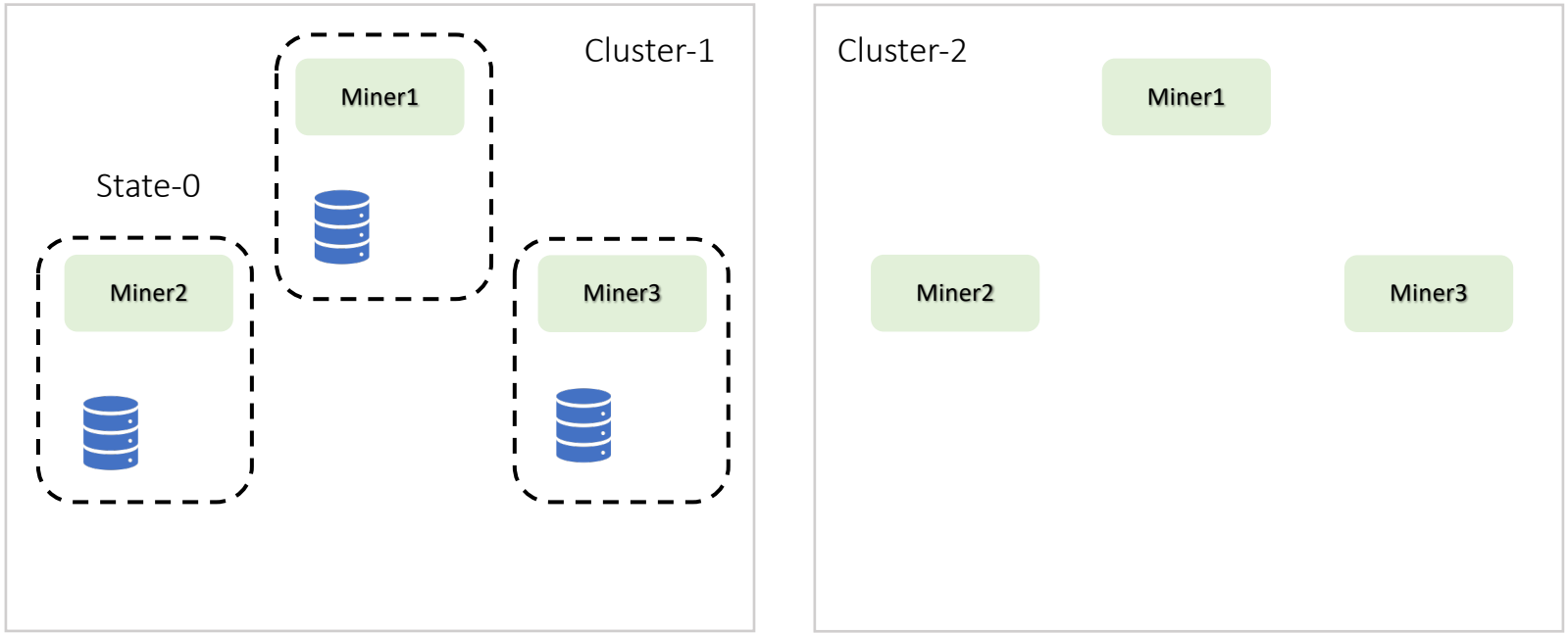
Proof-of-work rate limits the creation of new blocks, but it **does not** restrict the number of transactions in each block!

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

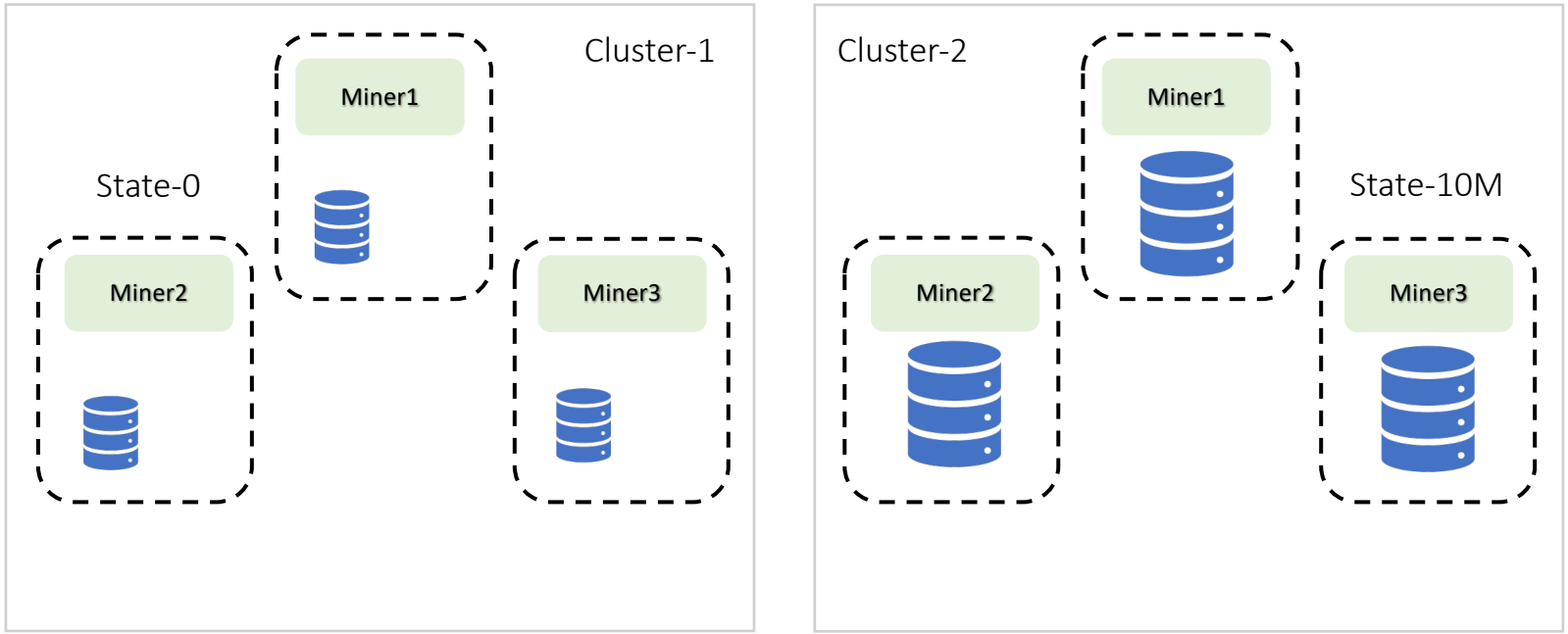
Impact of I/O on Overall Throughput

Impact of I/O on Overall Throughput

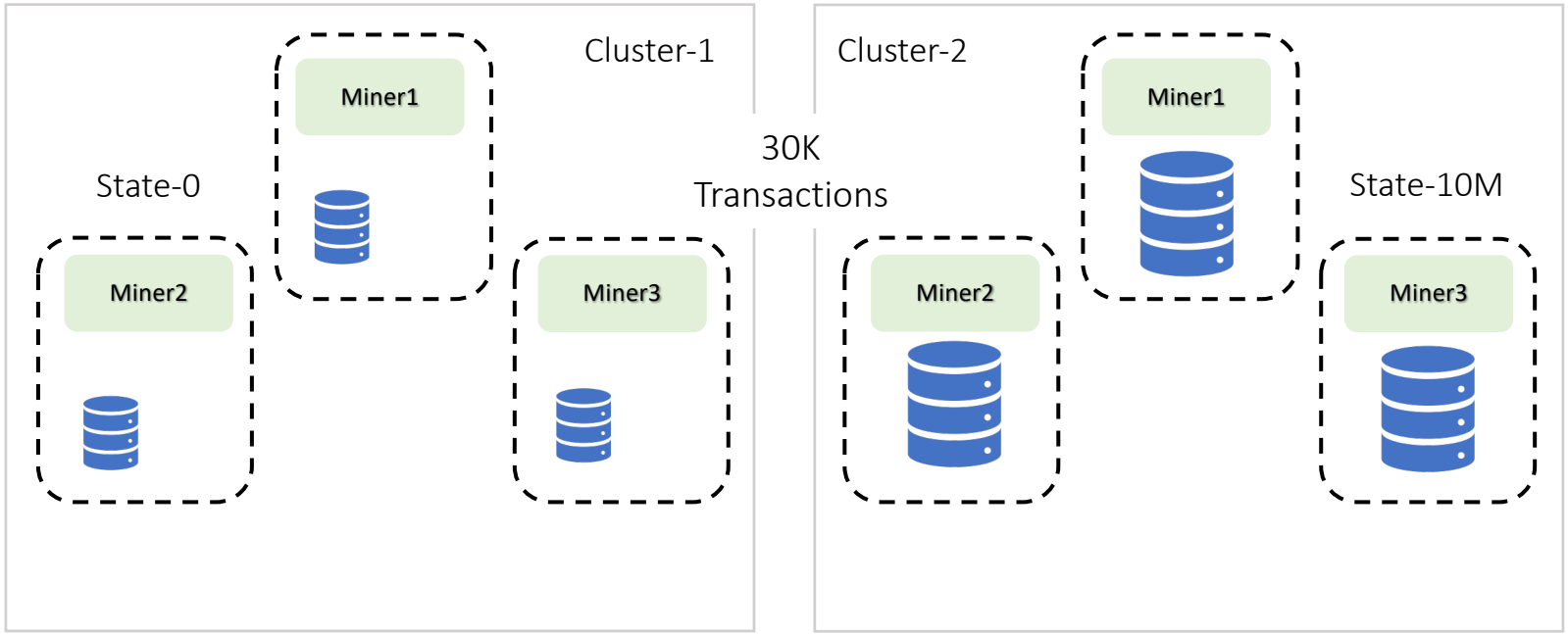
Impact of I/O on Overall Throughput



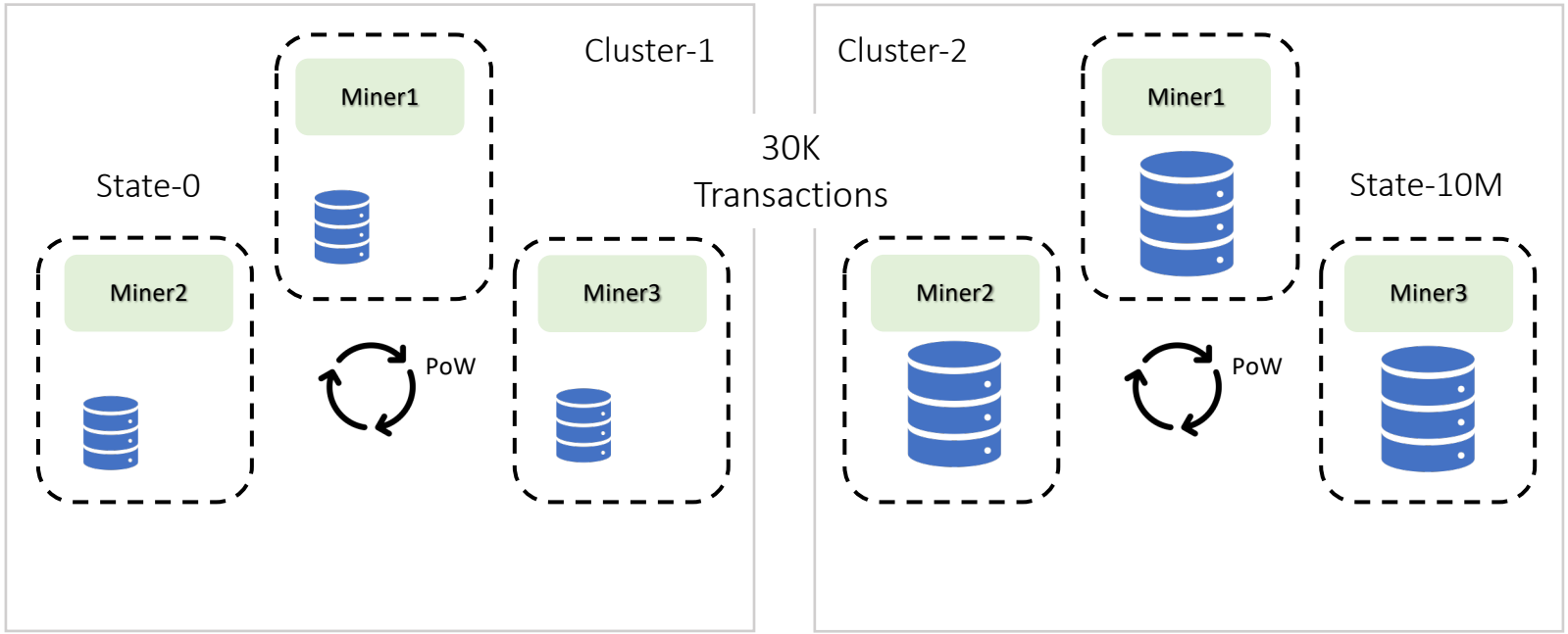
Impact of I/O on Overall Throughput



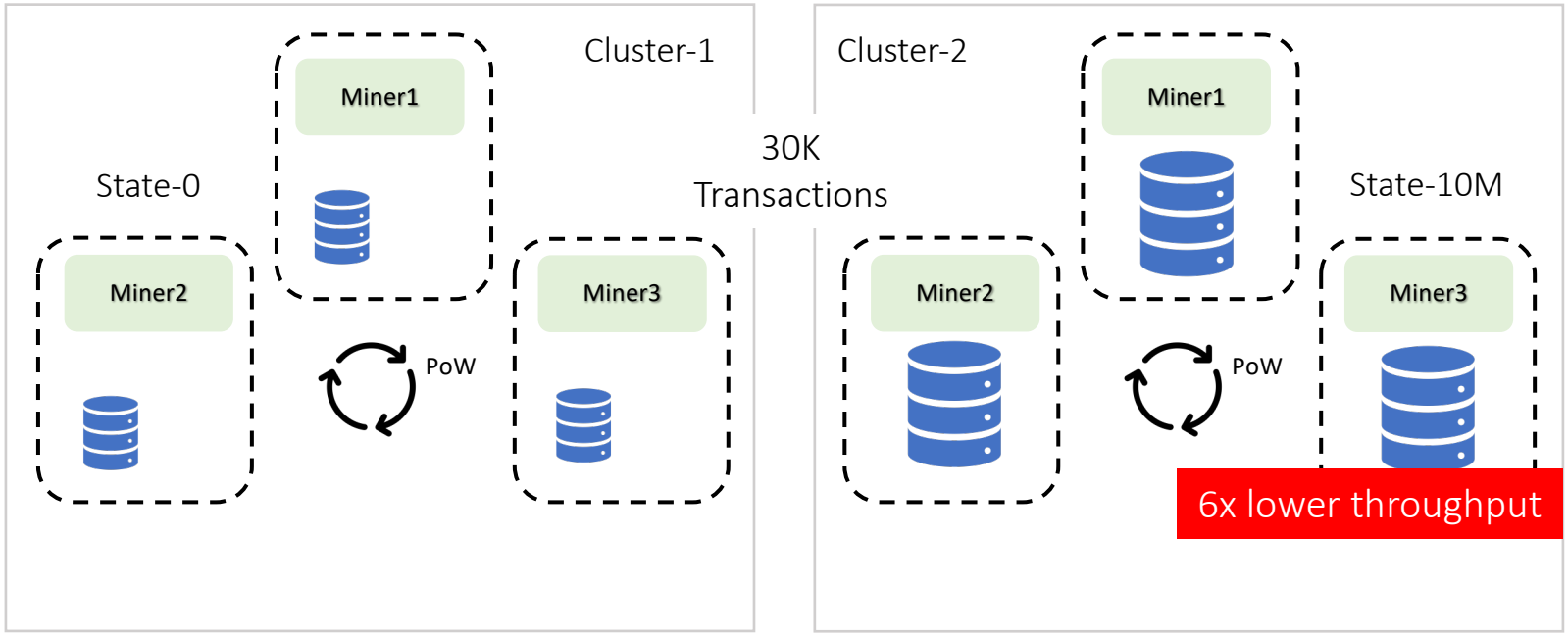
Impact of I/O on Overall Throughput



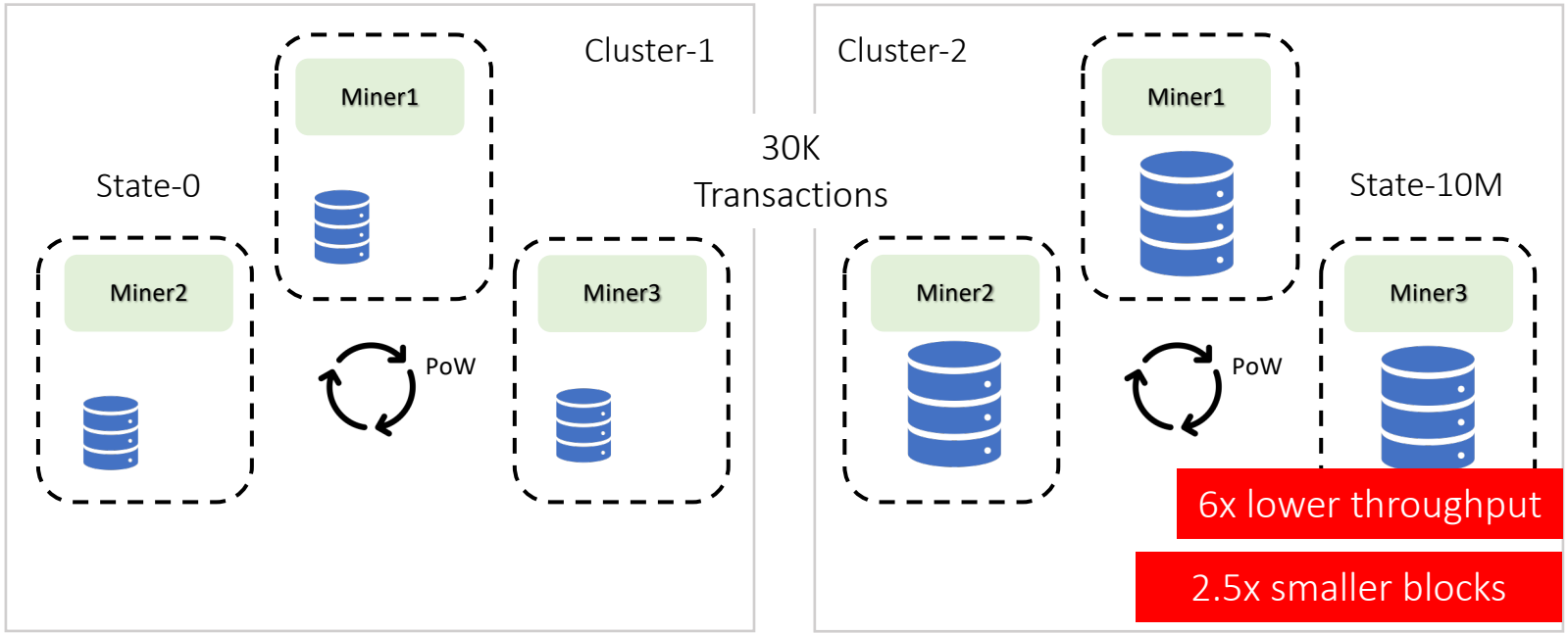
Impact of I/O on Overall Throughput



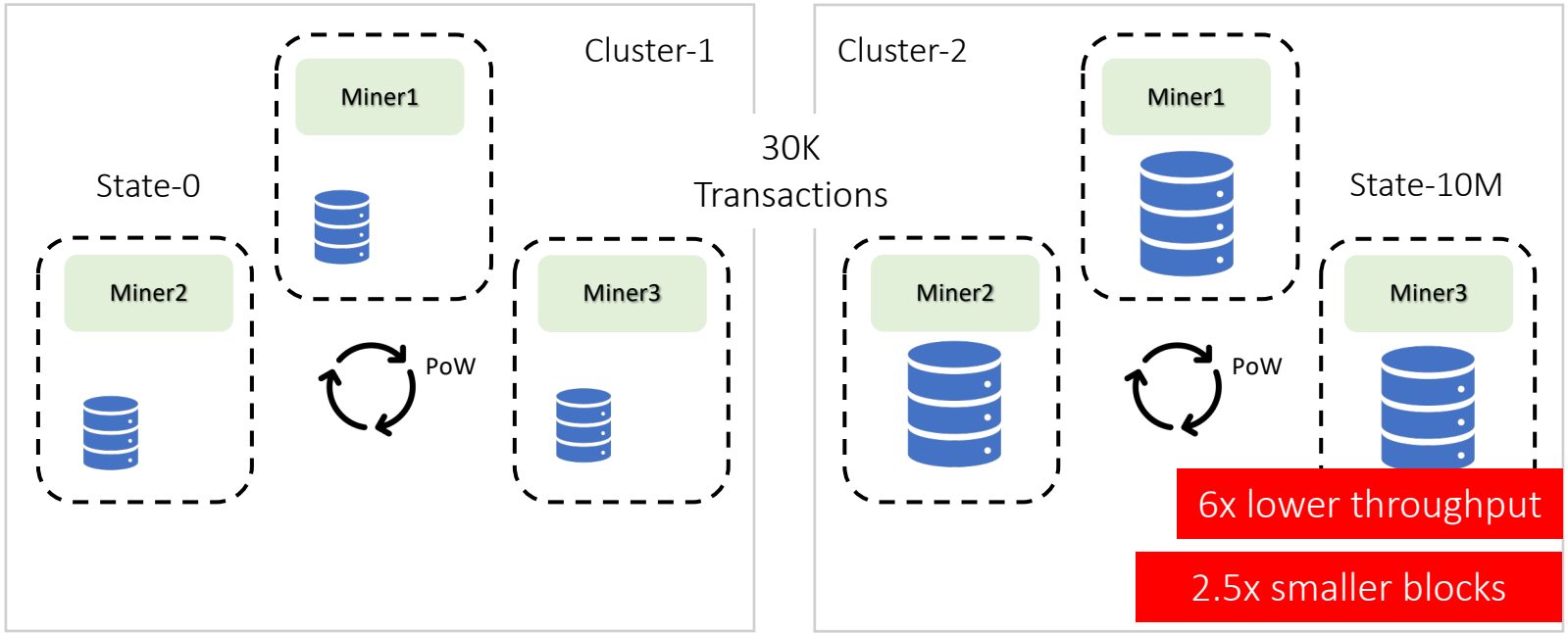
Impact of I/O on Overall Throughput



Impact of I/O on Overall Throughput



Impact of I/O on Overall Throughput



I/O bottlenecks limit block size and the overall throughput

Faster Transaction Processing

Faster Transaction Processing

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

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

RainBlock: Faster Transaction Processing

RainBlock: Faster Transaction Processing

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

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: 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

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

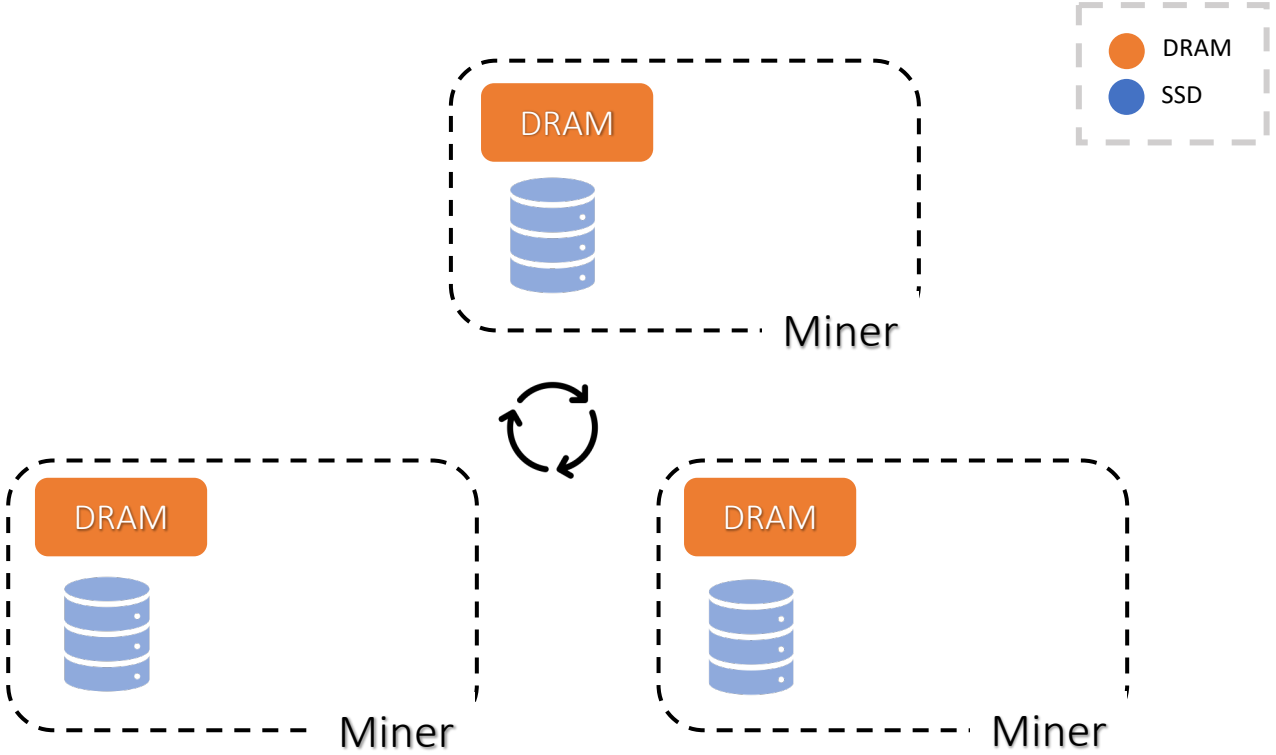
In a geo-distributed setting, with 4 miners in 4 regions spread across 3 continents, RainBlock miners can process about 20000 transactions per second

Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions

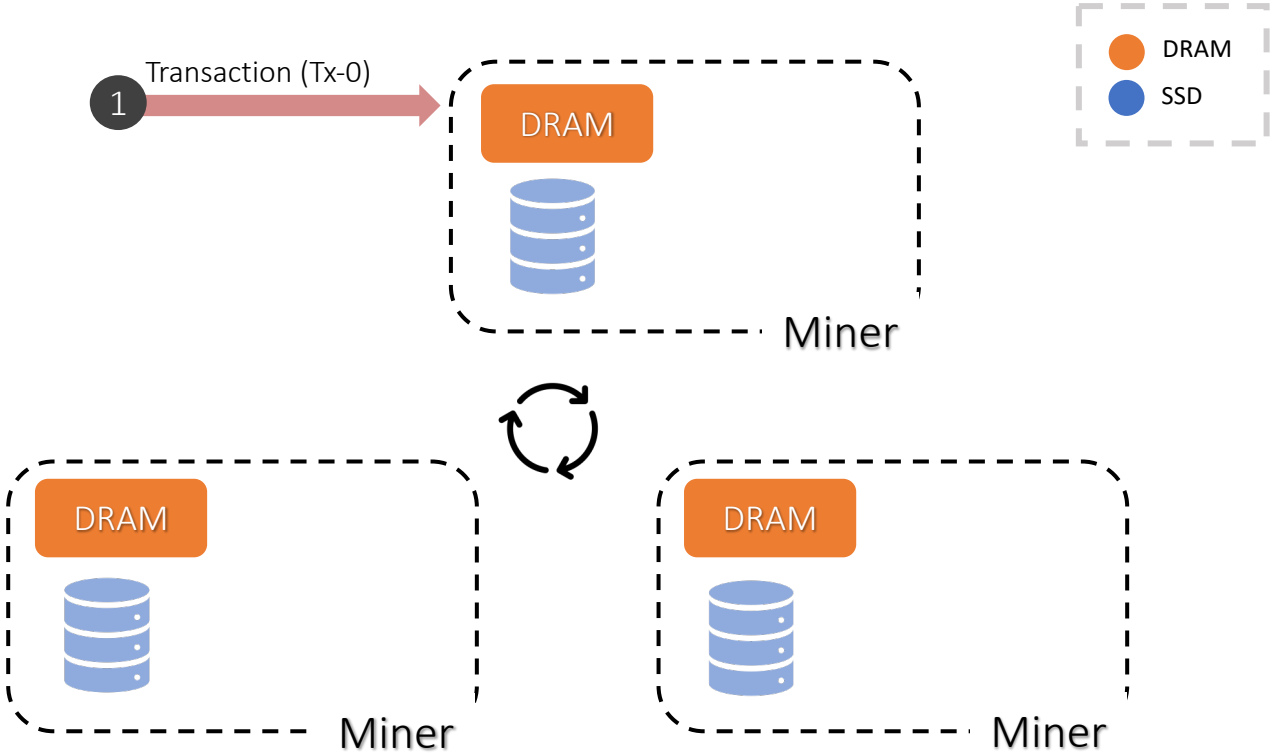
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



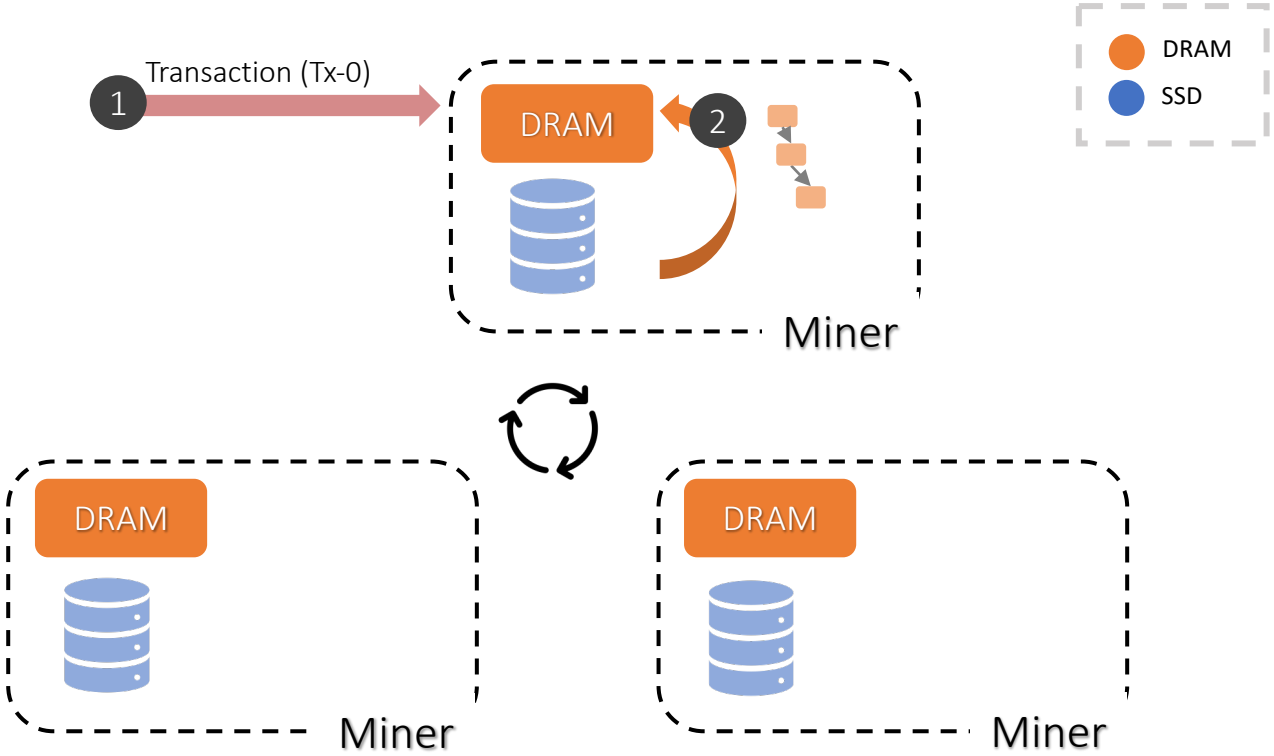
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



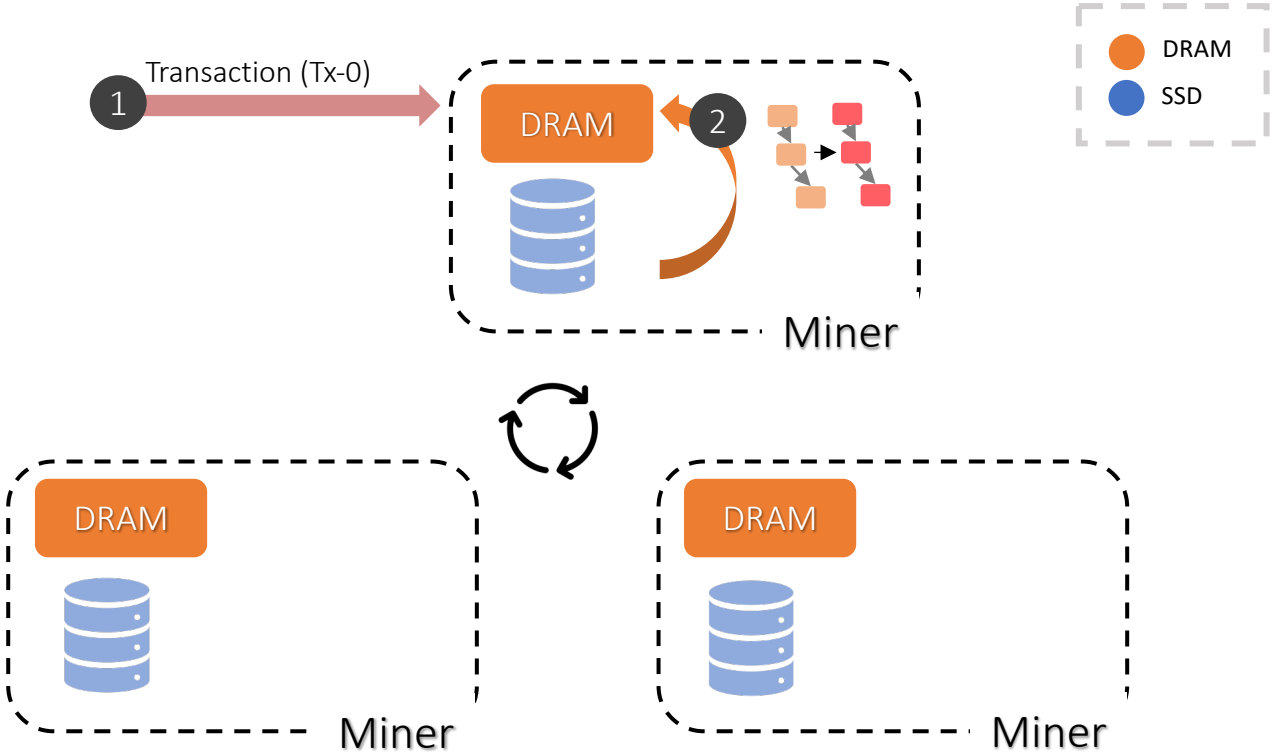
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



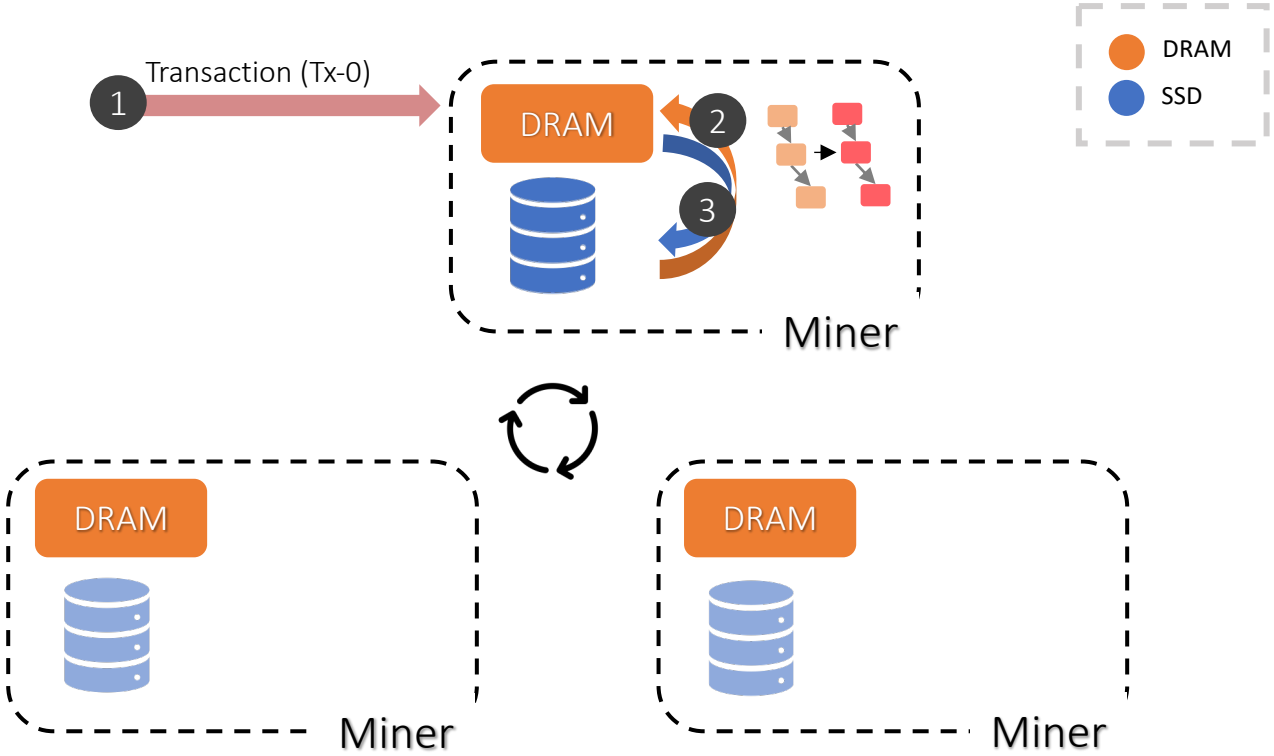
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



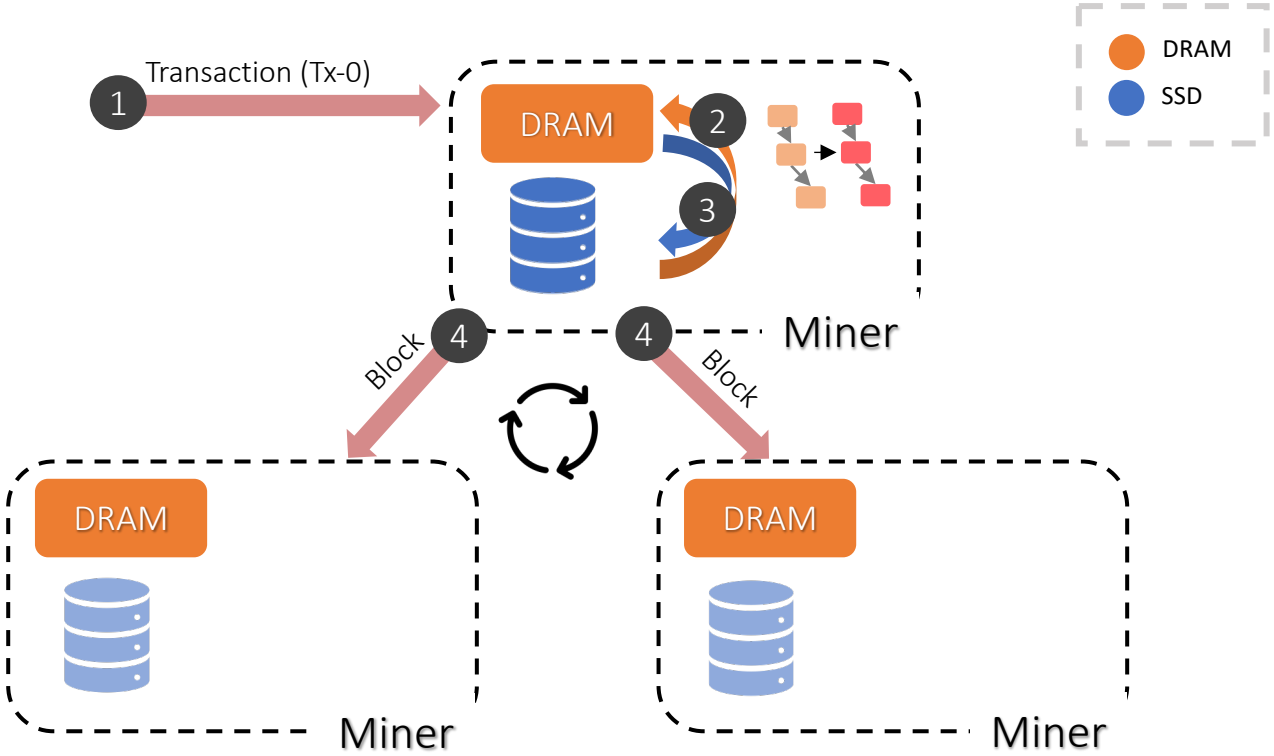
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



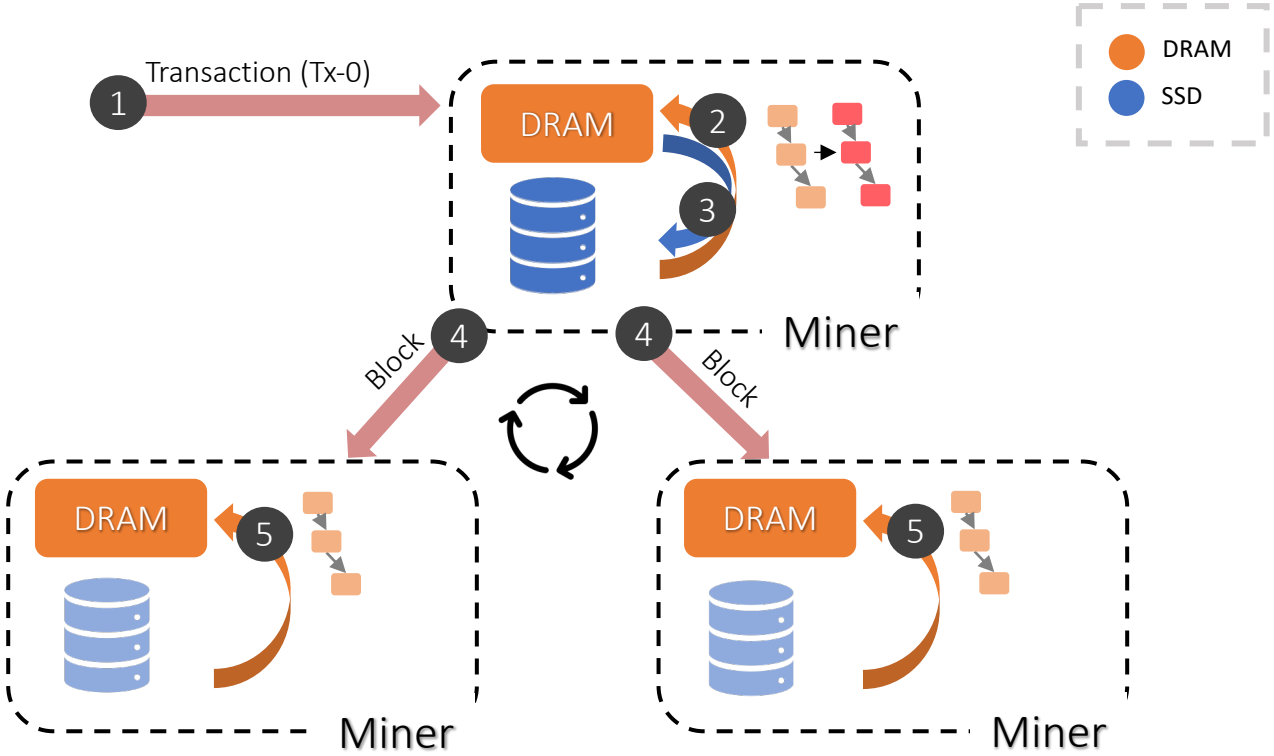
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



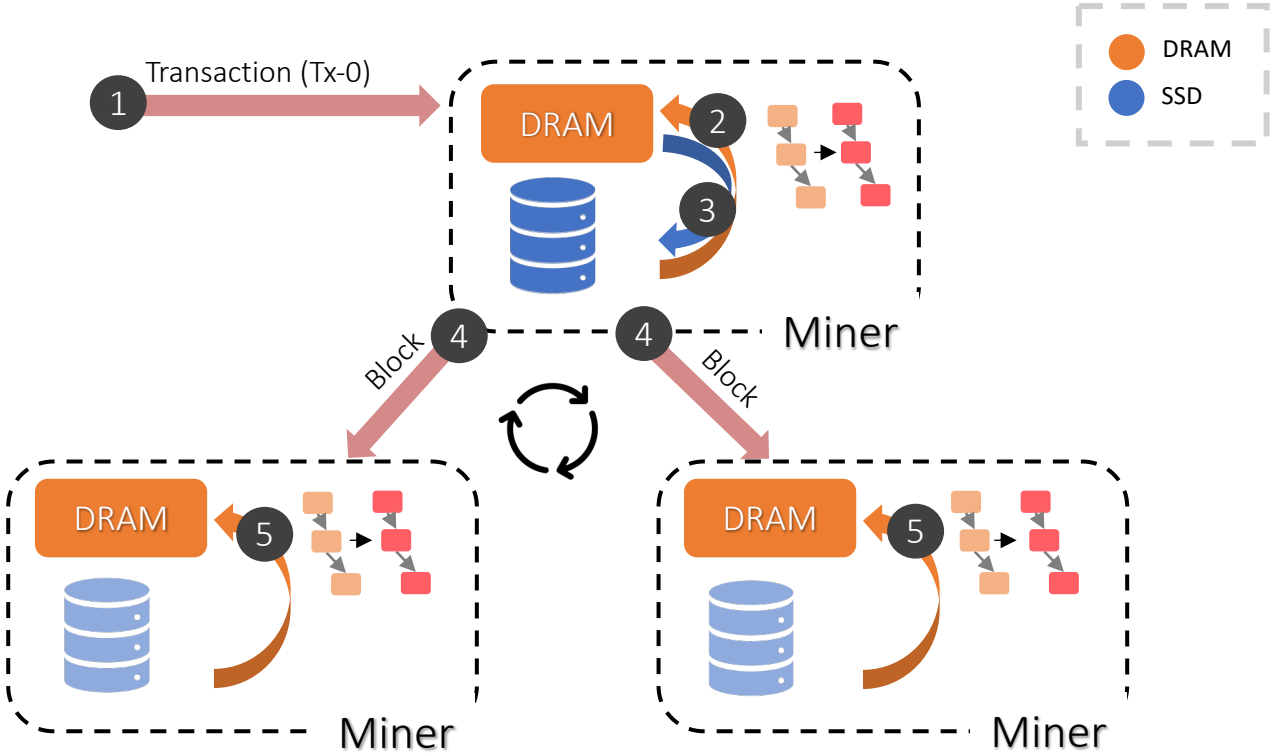
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



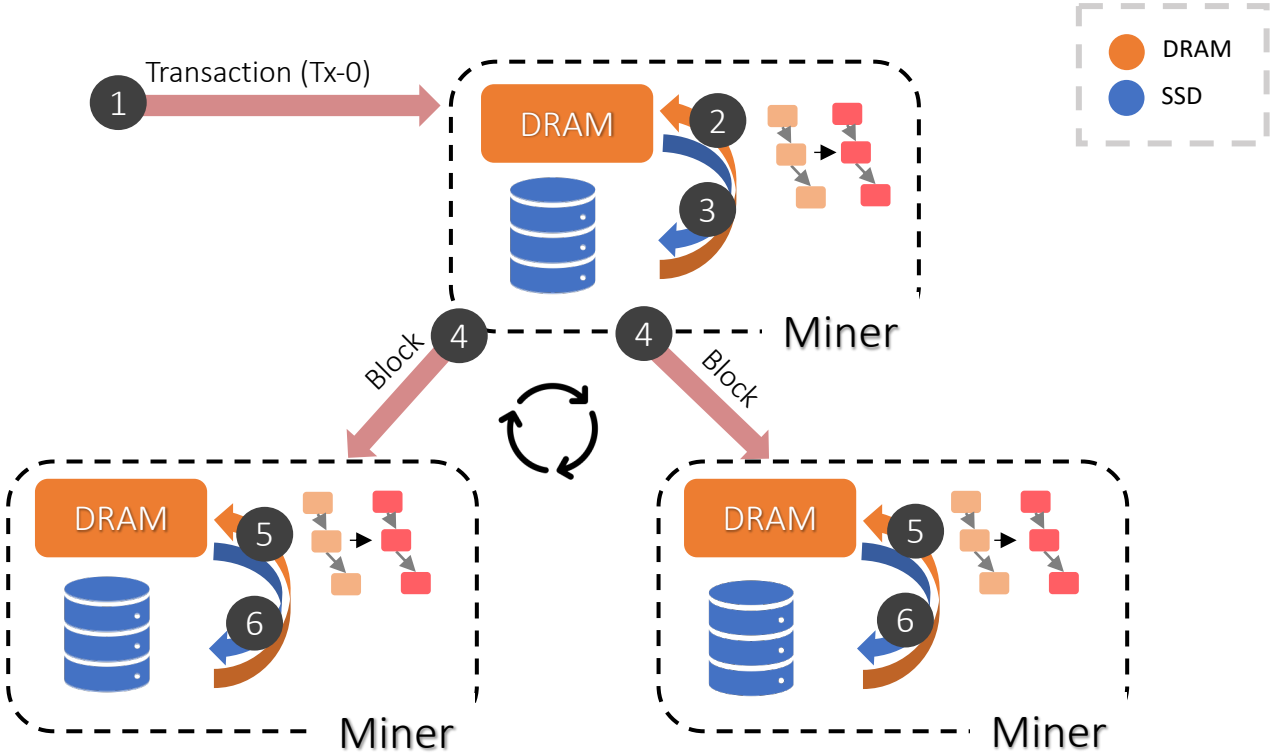
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



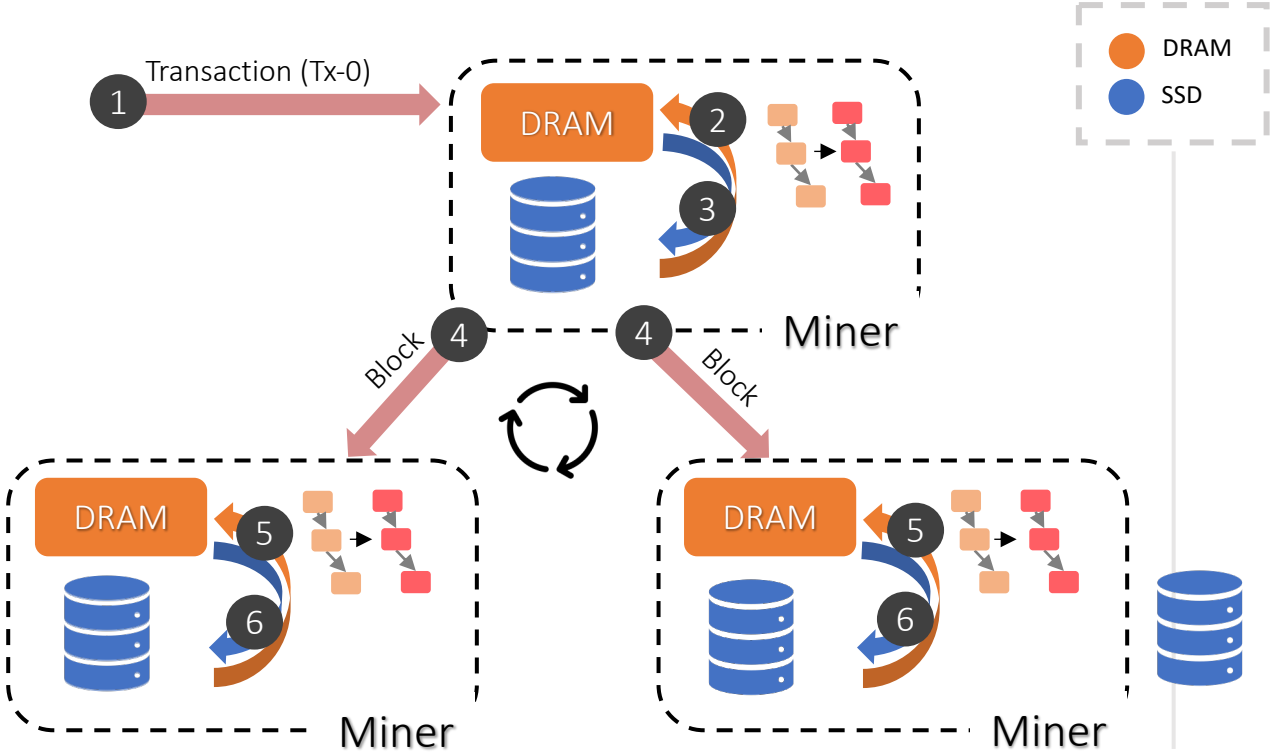
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



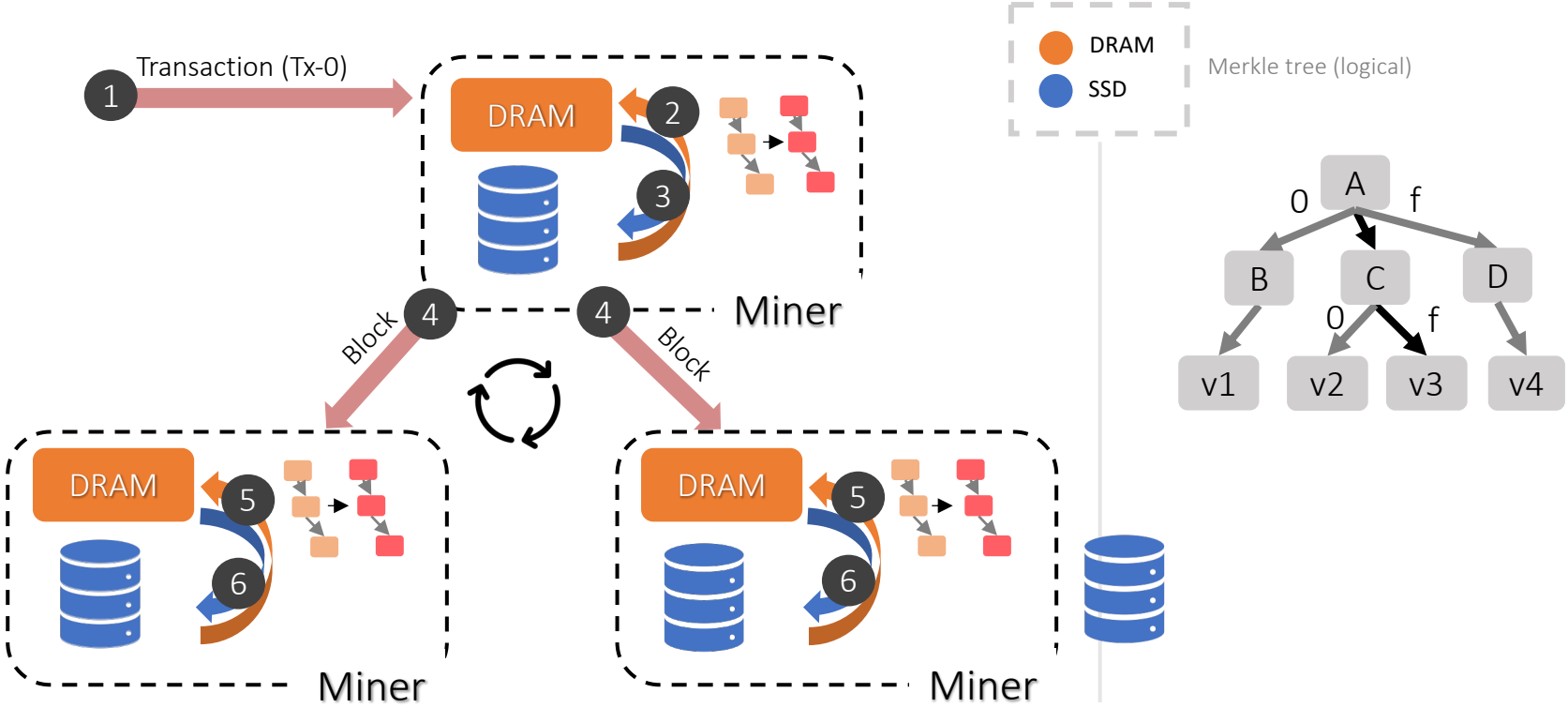
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



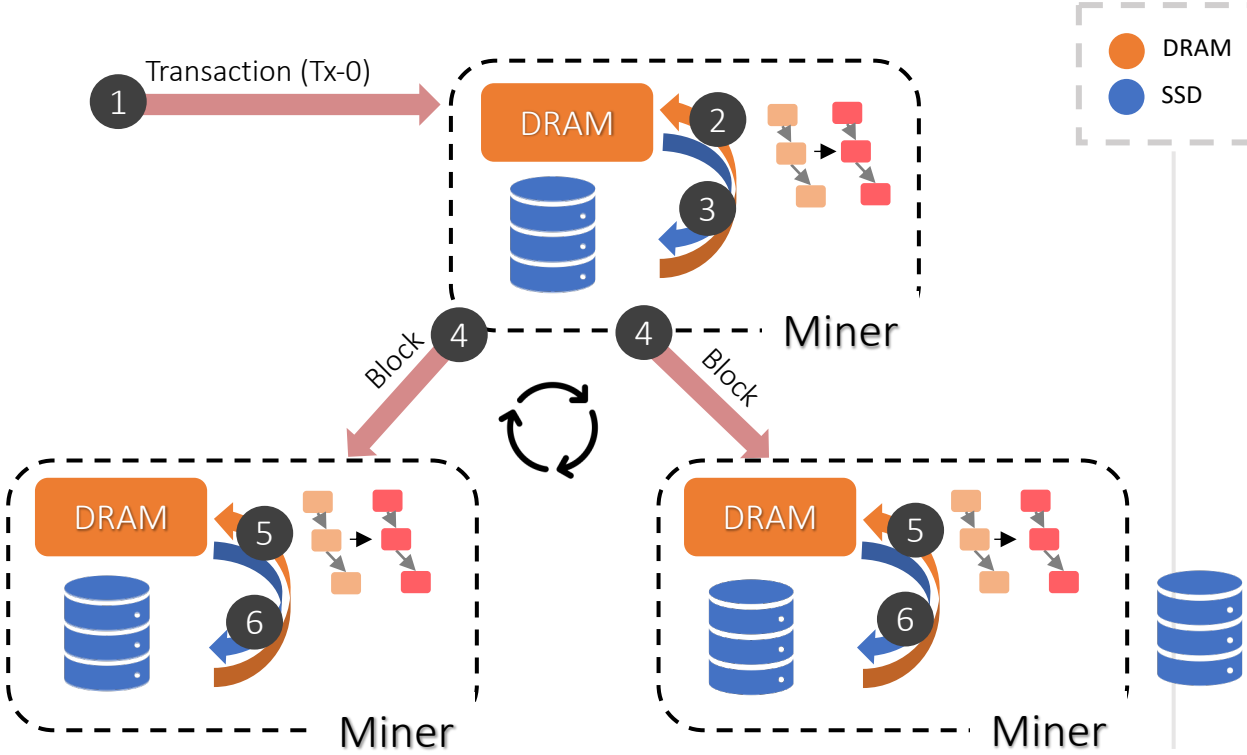
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions

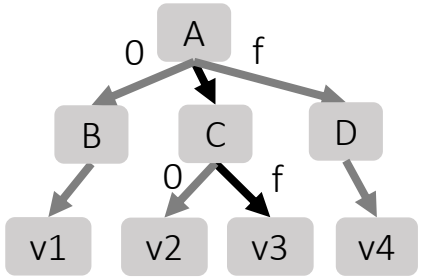


DRAM

SSD

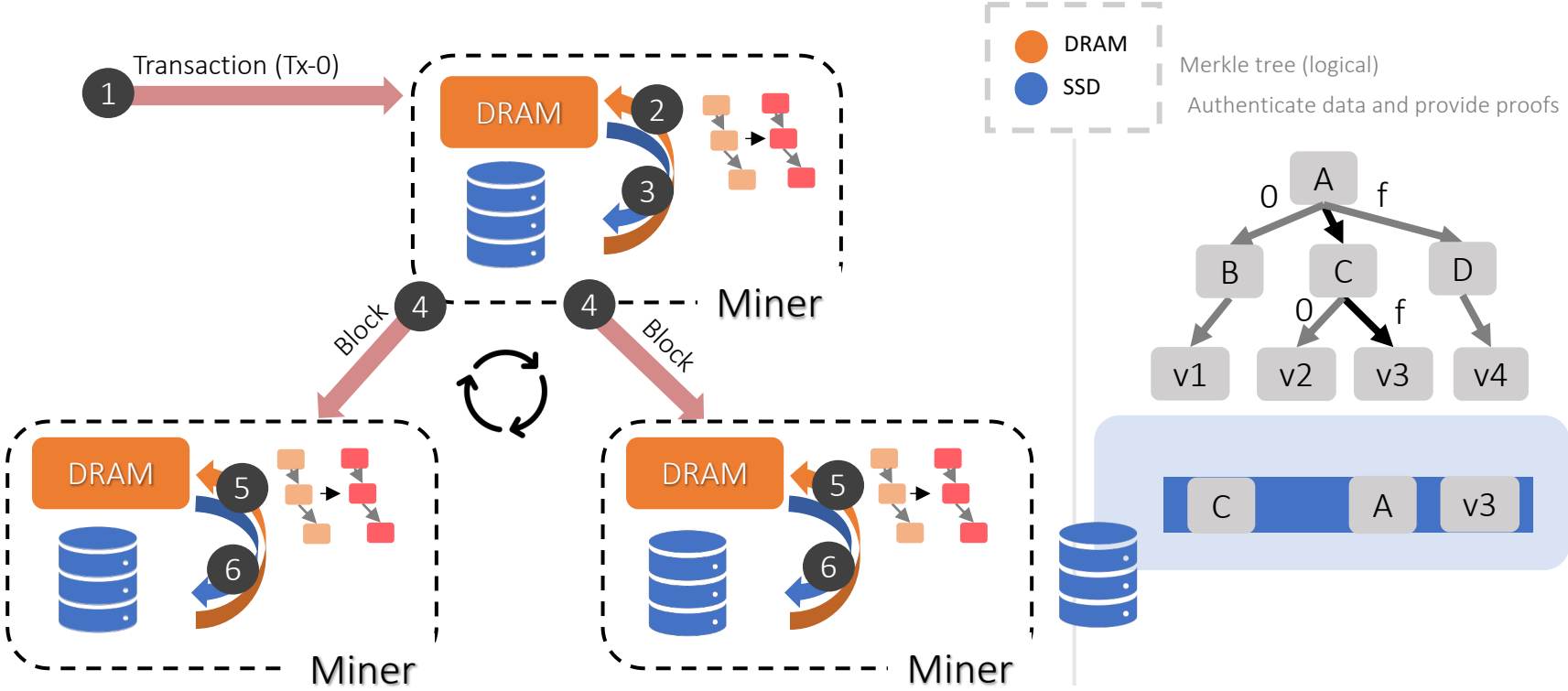
Merkle tree (logical)

Authenticate data and provide proofs



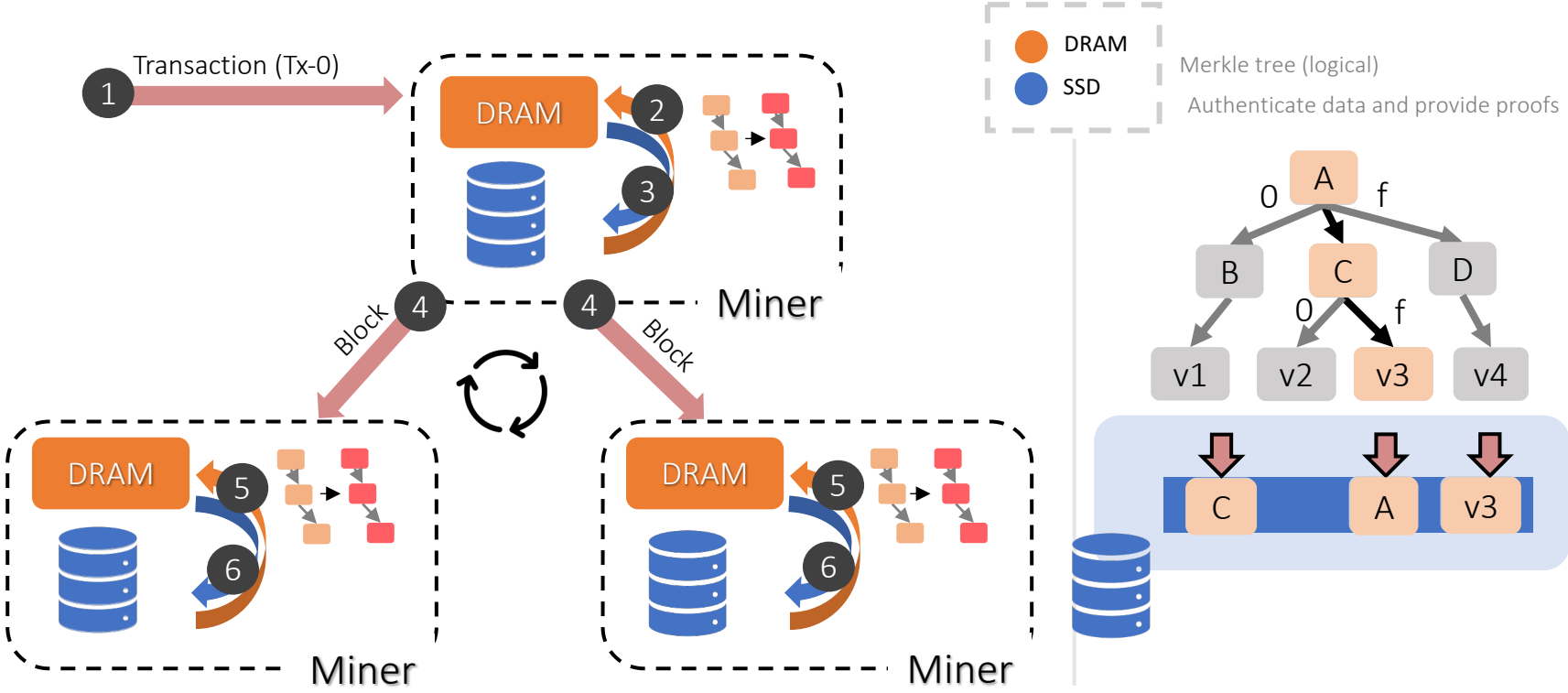
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



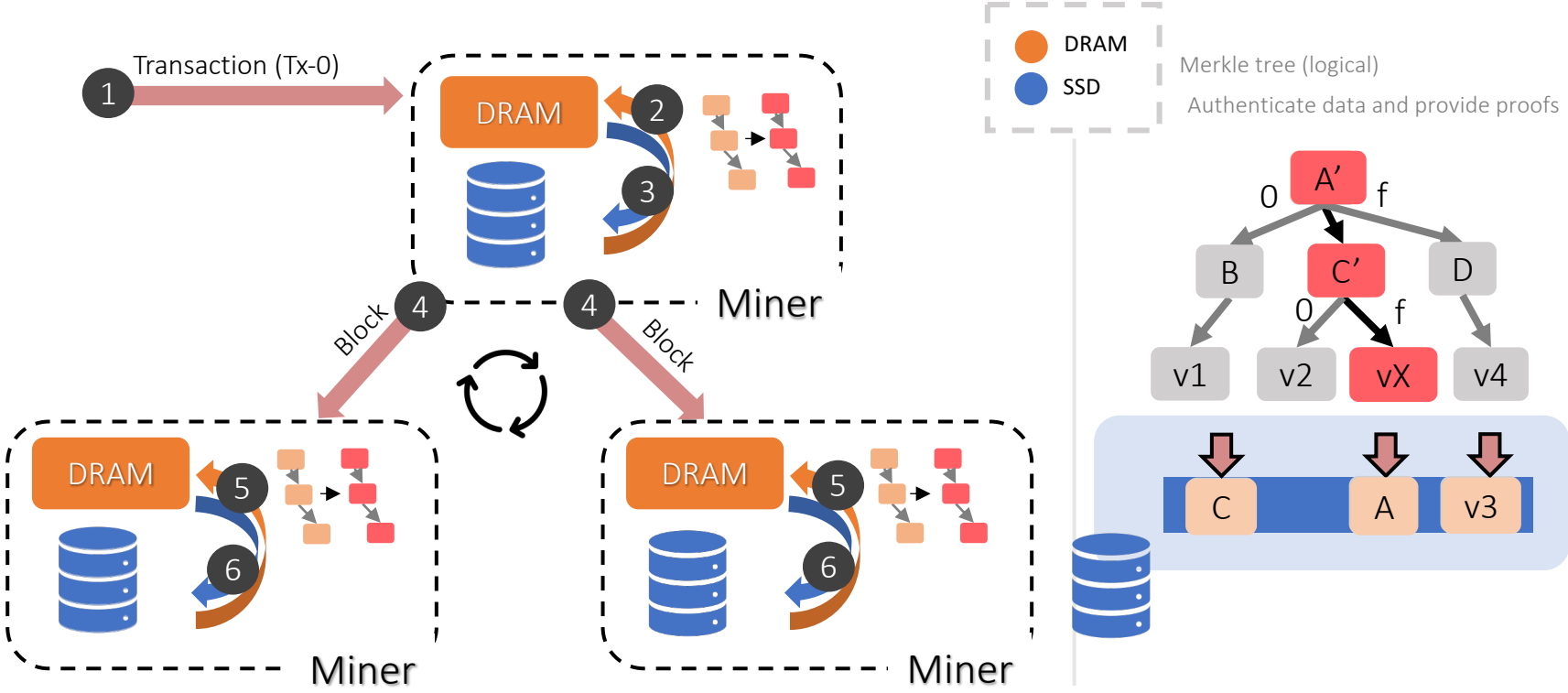
Transaction Processing in Ethereum

Accessing and modifying state in the critical path of processing transactions



Transaction Processing in Ethereum

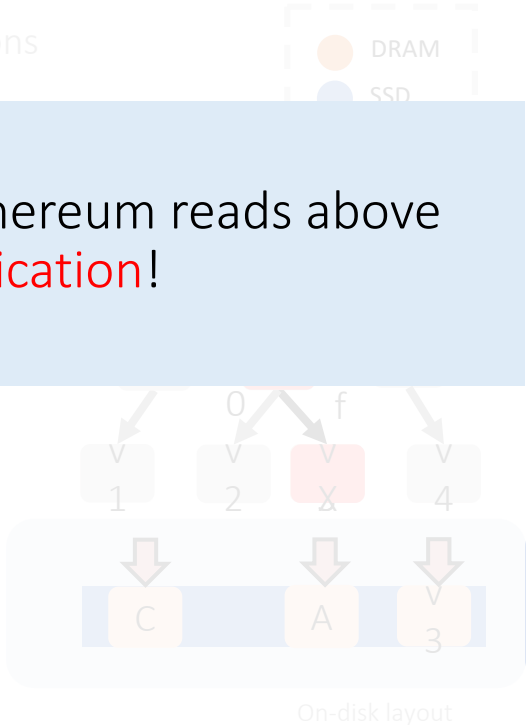
Accessing and modifying state in the critical path of processing transactions



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 40MB, 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 40MB, 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

Outline

RainBlock architecture reduces I/O bottlenecks

Outline

RainBlock architecture reduces I/O bottlenecks

Challenges RainBlock addresses

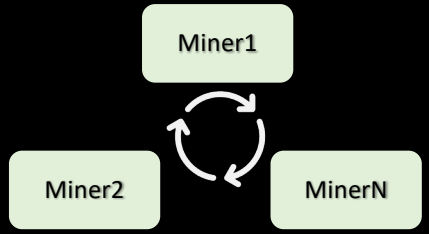
Outline

RainBlock architecture reduces I/O bottlenecks

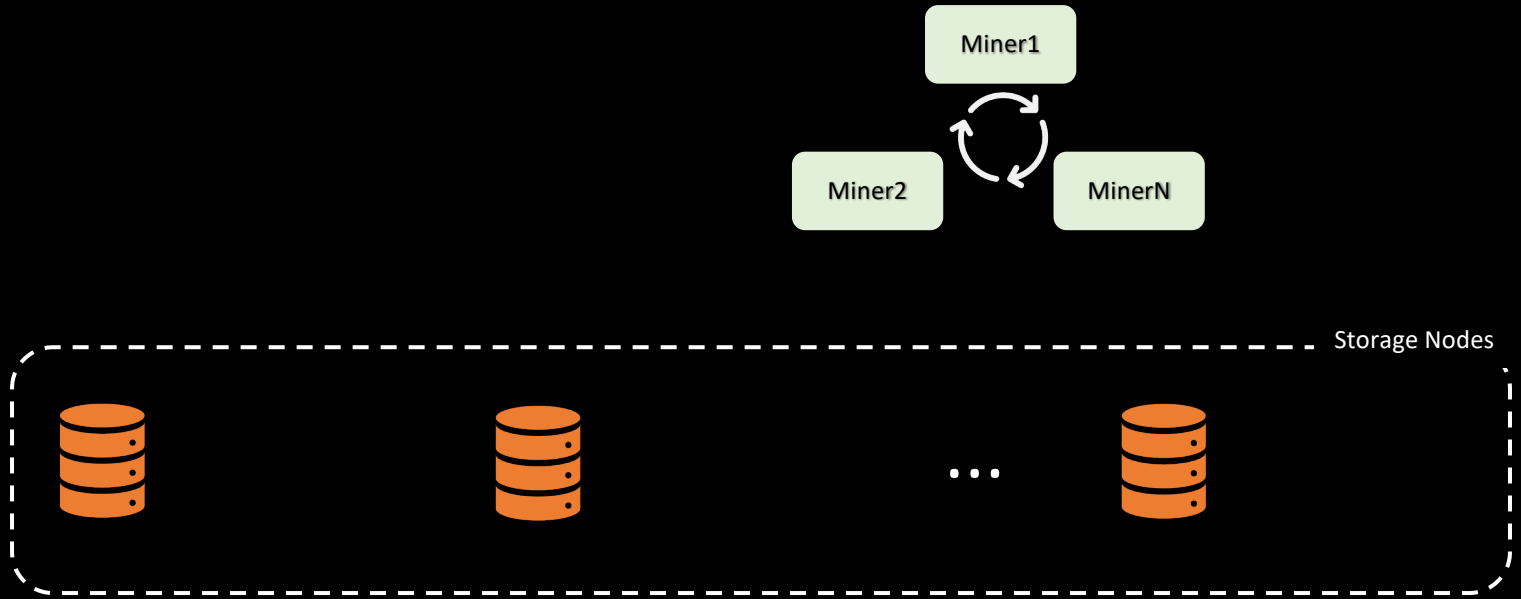
Challenges RainBlock addresses

Life of a Transaction in RainBlock

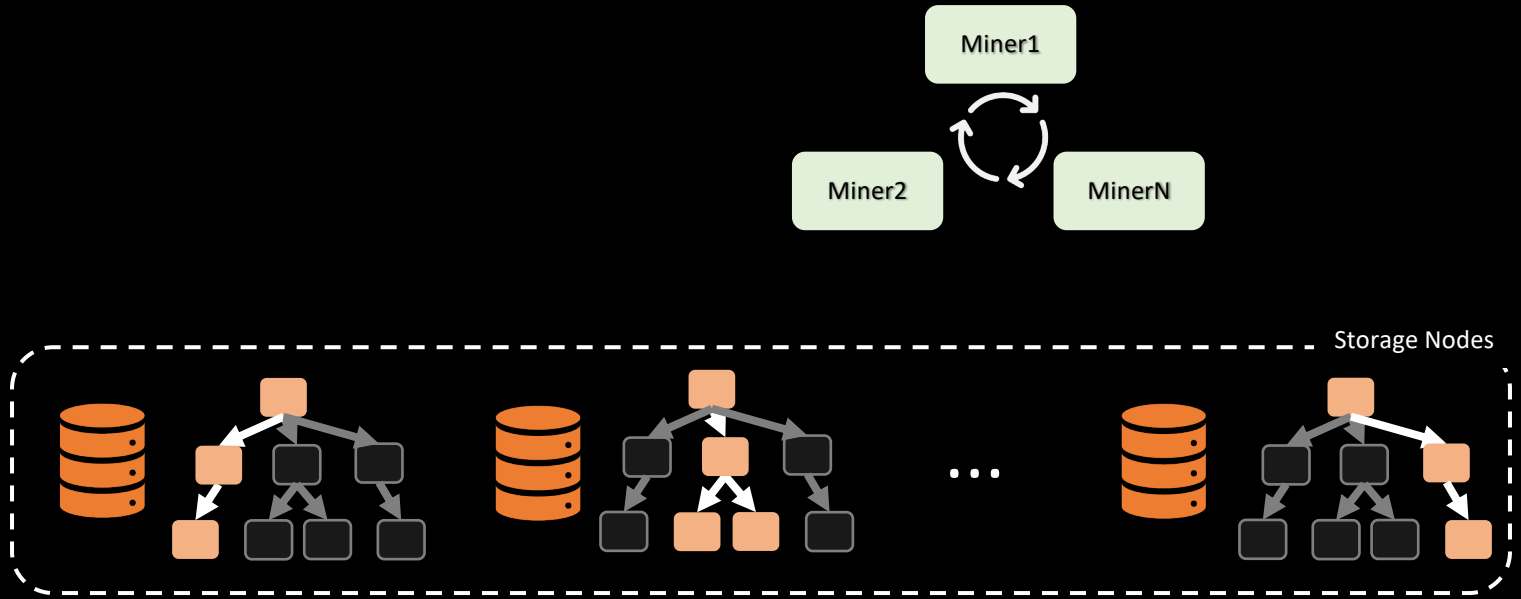
RainBlock: Architecture for Public Blockchains



RainBlock: Architecture for Public Blockchains

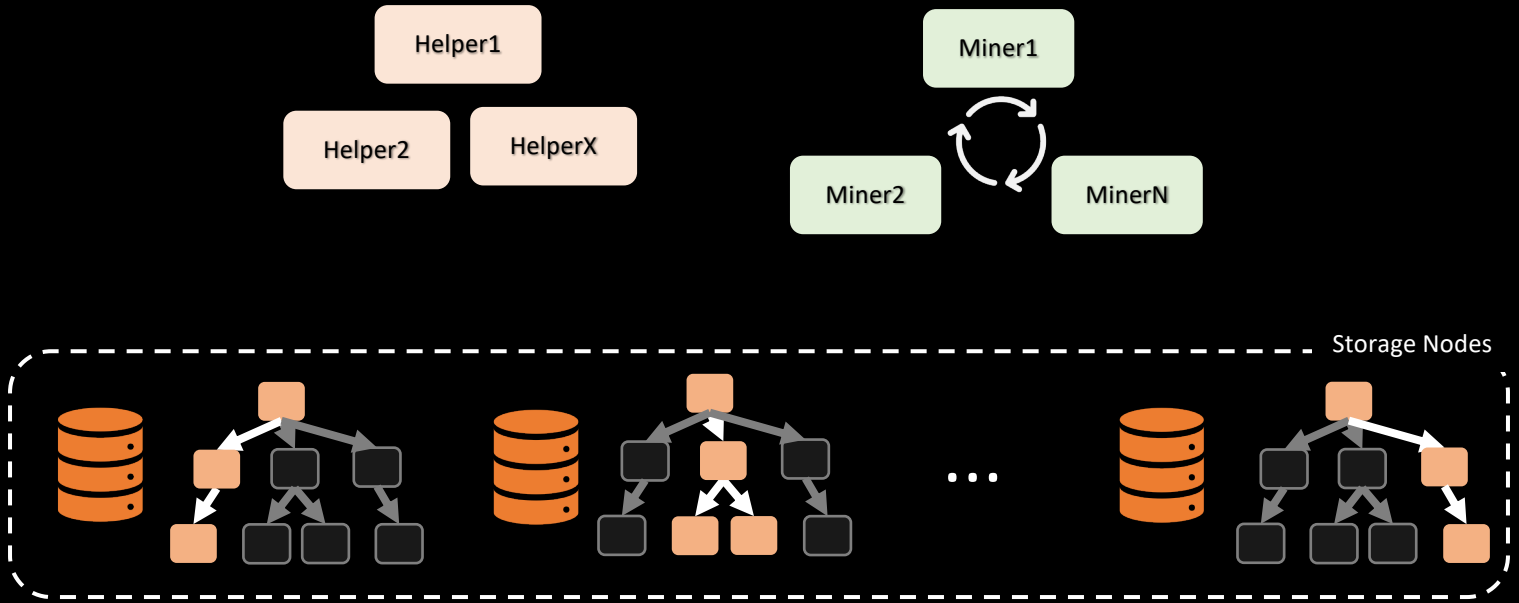


RainBlock: Architecture for Public Blockchains



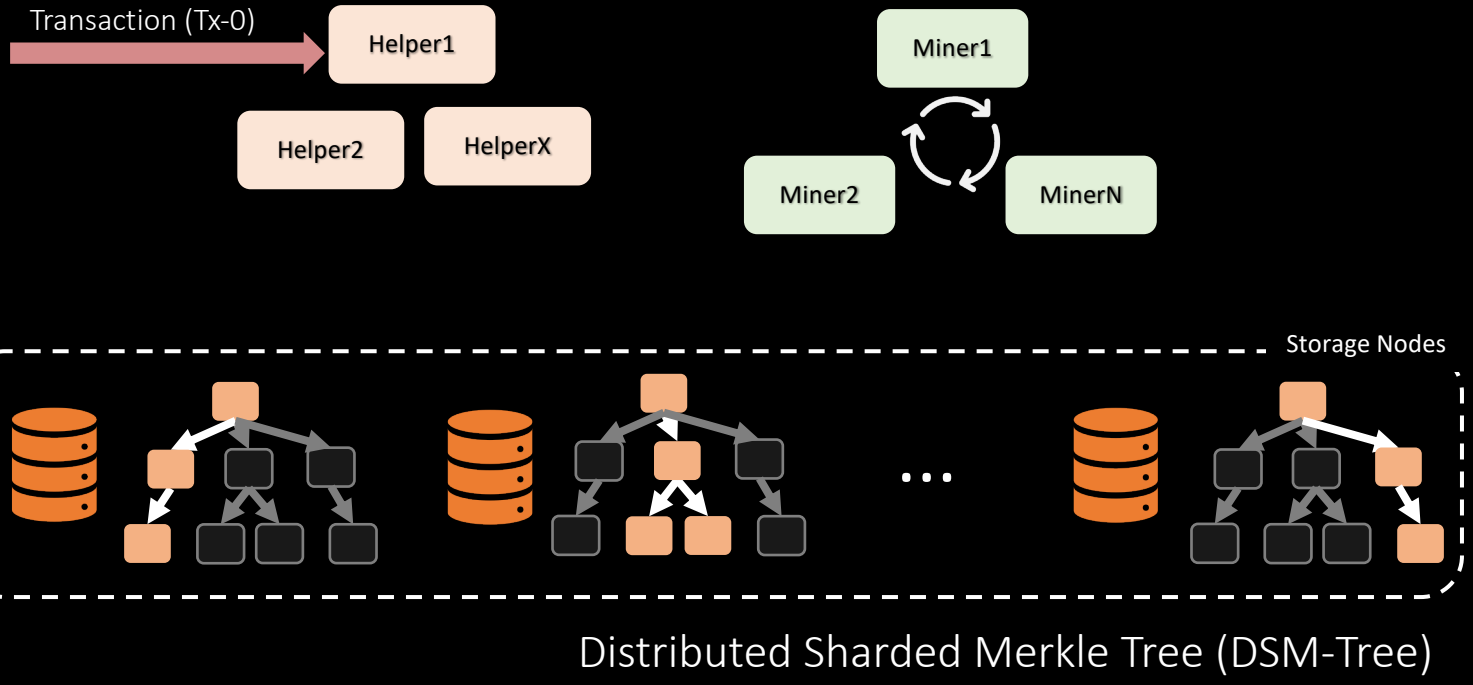
Distributed Sharded Merkle Tree (DSM-Tree)

RainBlock: Architecture for Public Blockchains



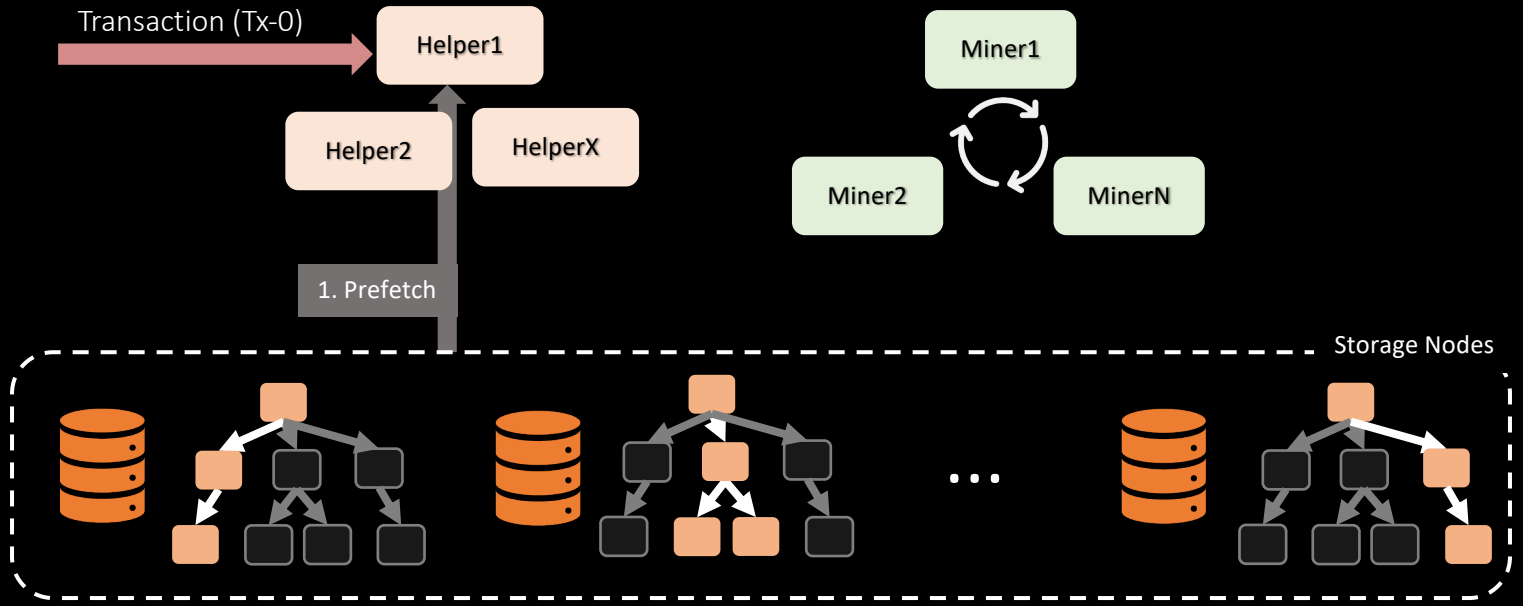
Distributed Sharded Merkle Tree (DSM-Tree)

RainBlock: Architecture for Public Blockchains



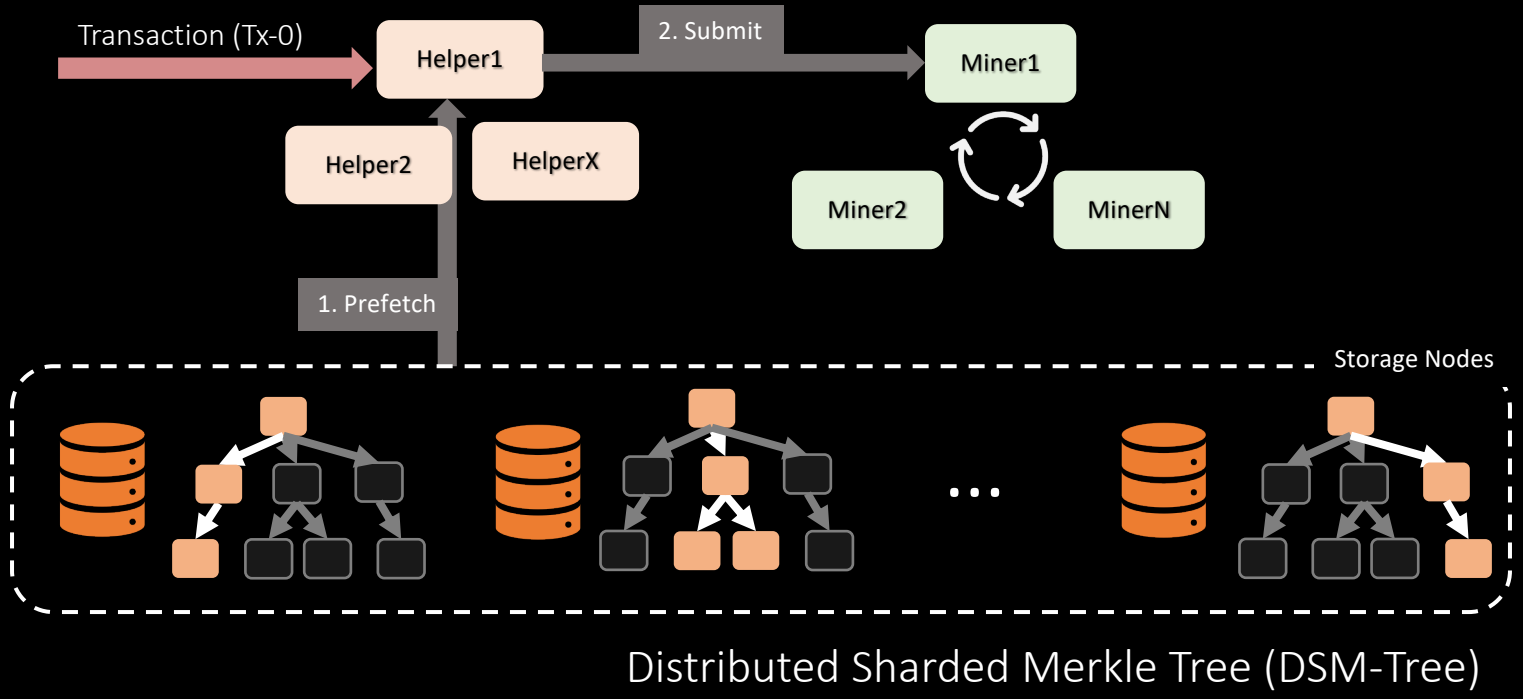
Distributed Sharded Merkle Tree (DSM-Tree)

RainBlock: Architecture for Public Blockchains



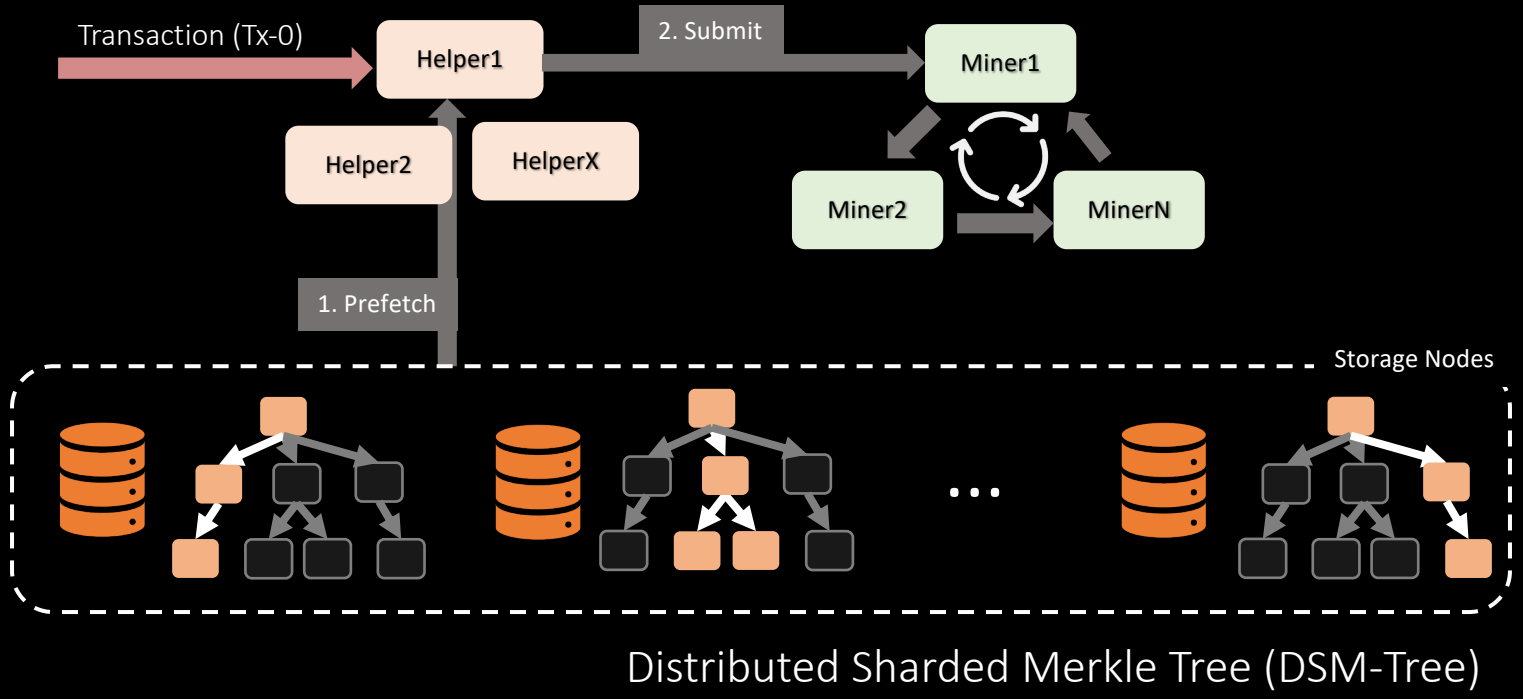
Distributed Sharded Merkle Tree (DSM-Tree)

RainBlock: Architecture for Public Blockchains



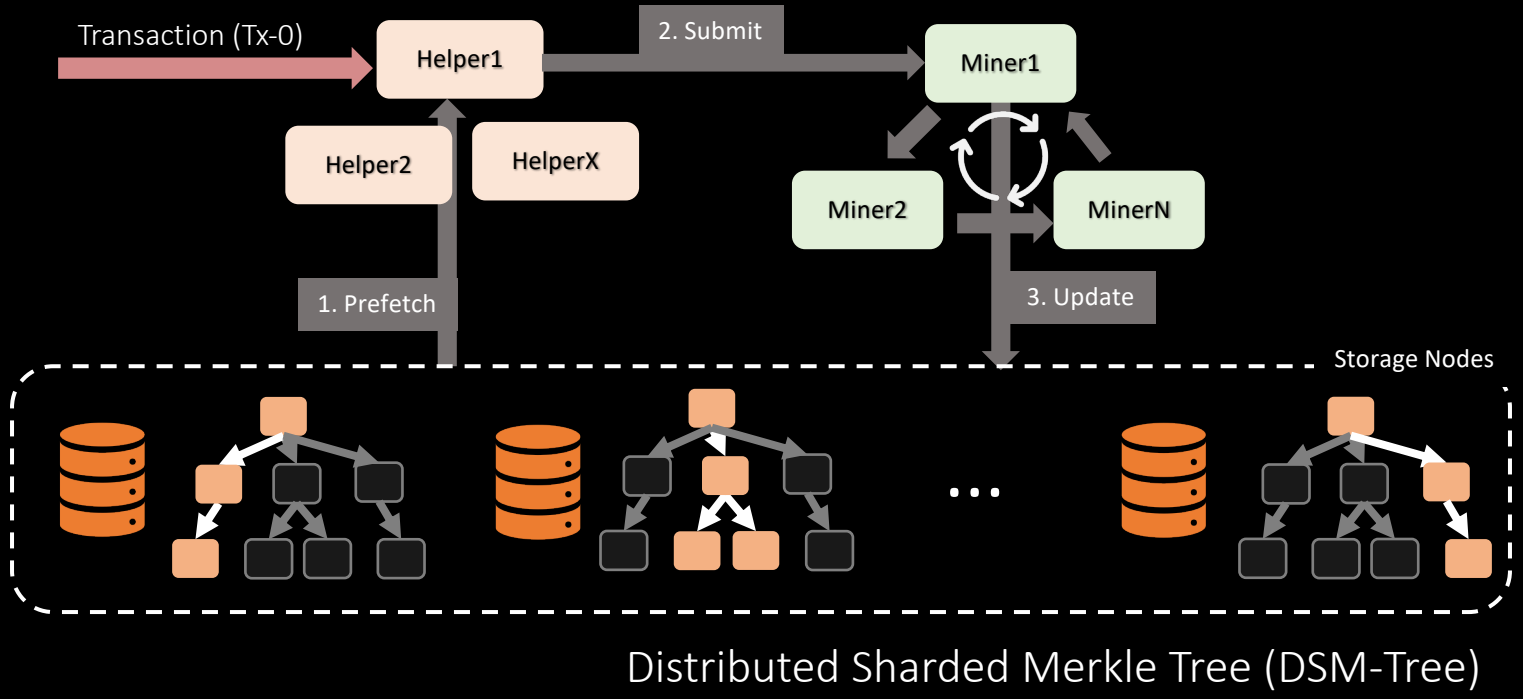
Distributed Sharded Merkle Tree (DSM-Tree)

RainBlock: Architecture for Public Blockchains



Distributed Sharded Merkle Tree (DSM-Tree)

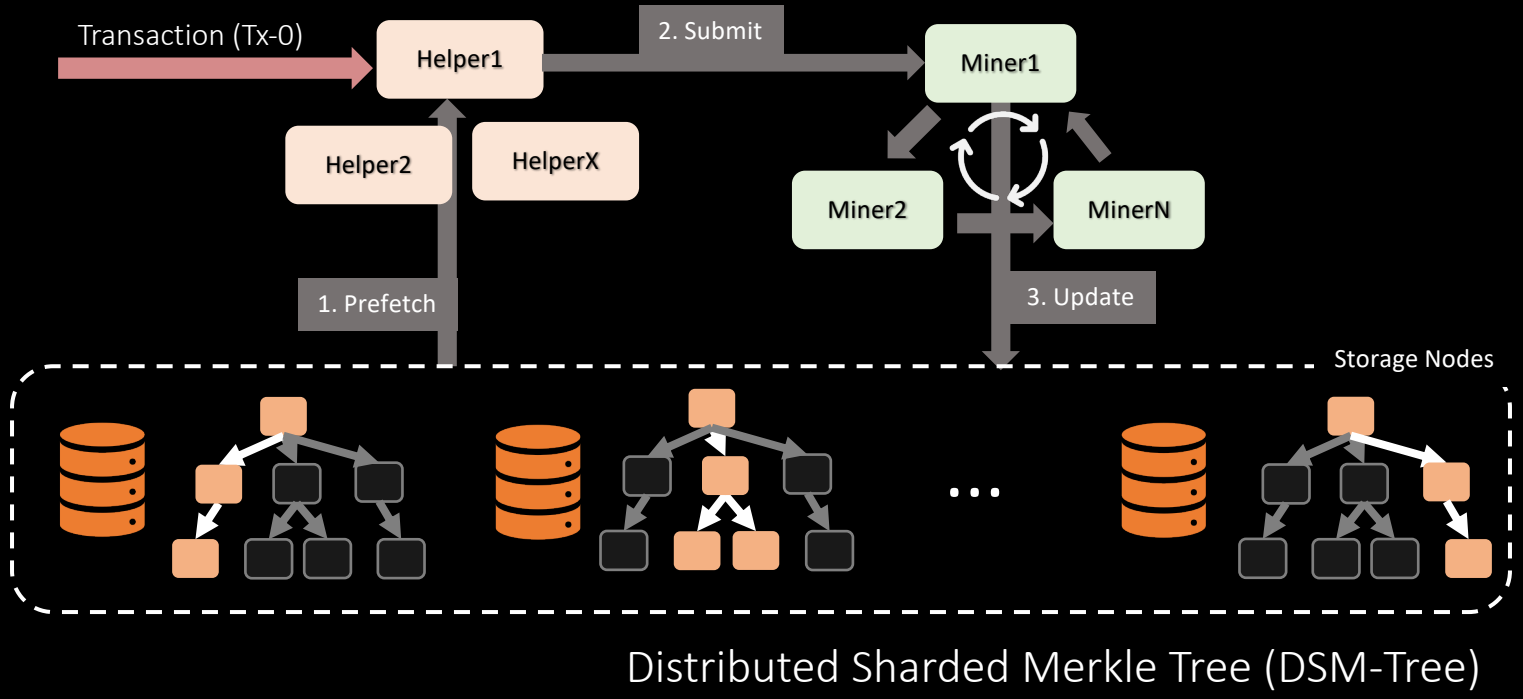
RainBlock: Architecture for Public Blockchains



Distributed Sharded Merkle Tree (DSM-Tree)

RainBlock: Architecture for Public Blockchains

Miners do not perform I/O in the critical path



Distributed Sharded Merkle Tree (DSM-Tree)

RainBlock: Challenges

RainBlock: Challenges

Challenge-I: Concurrent updates to storage

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

RainBlock: Challenges

Challenge-I: Concurrent updates to storage

I/O-Helpers can prefetch from storage 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

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

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!

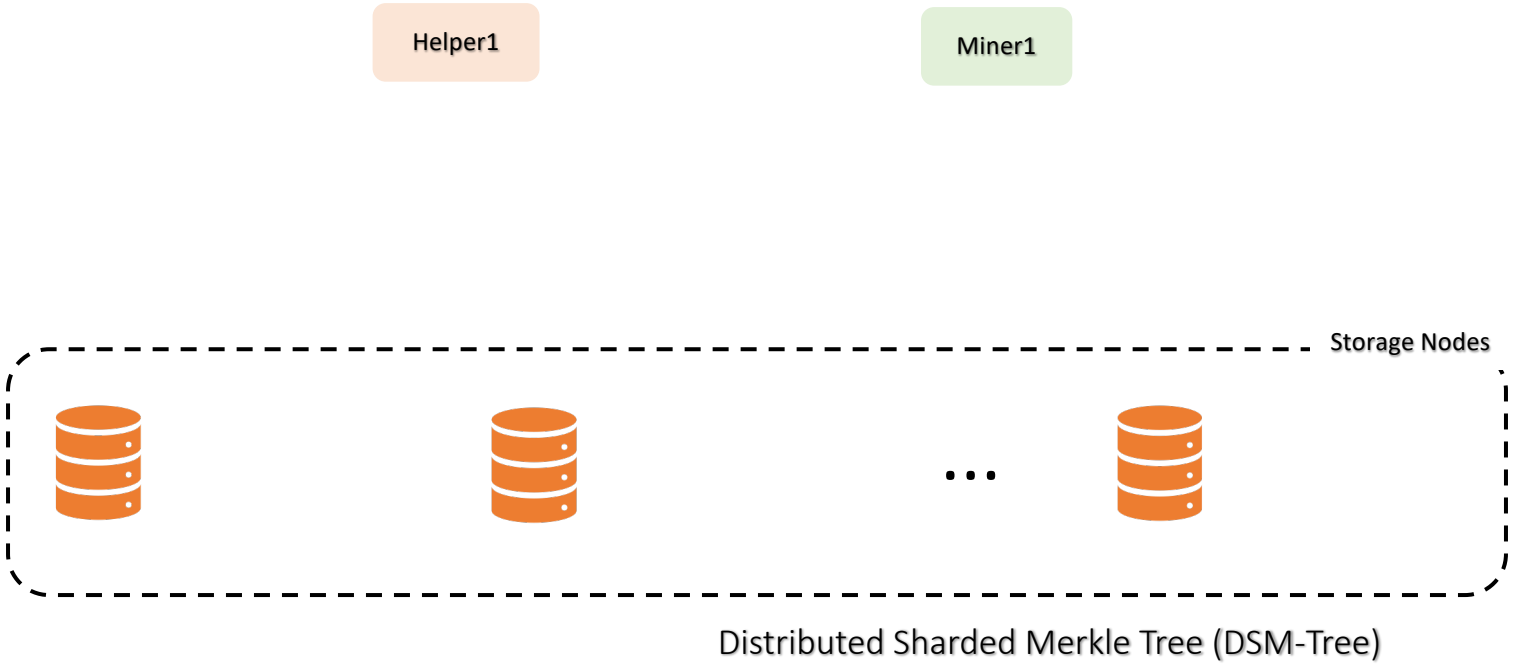
Challenge-II: Increased network traffic

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

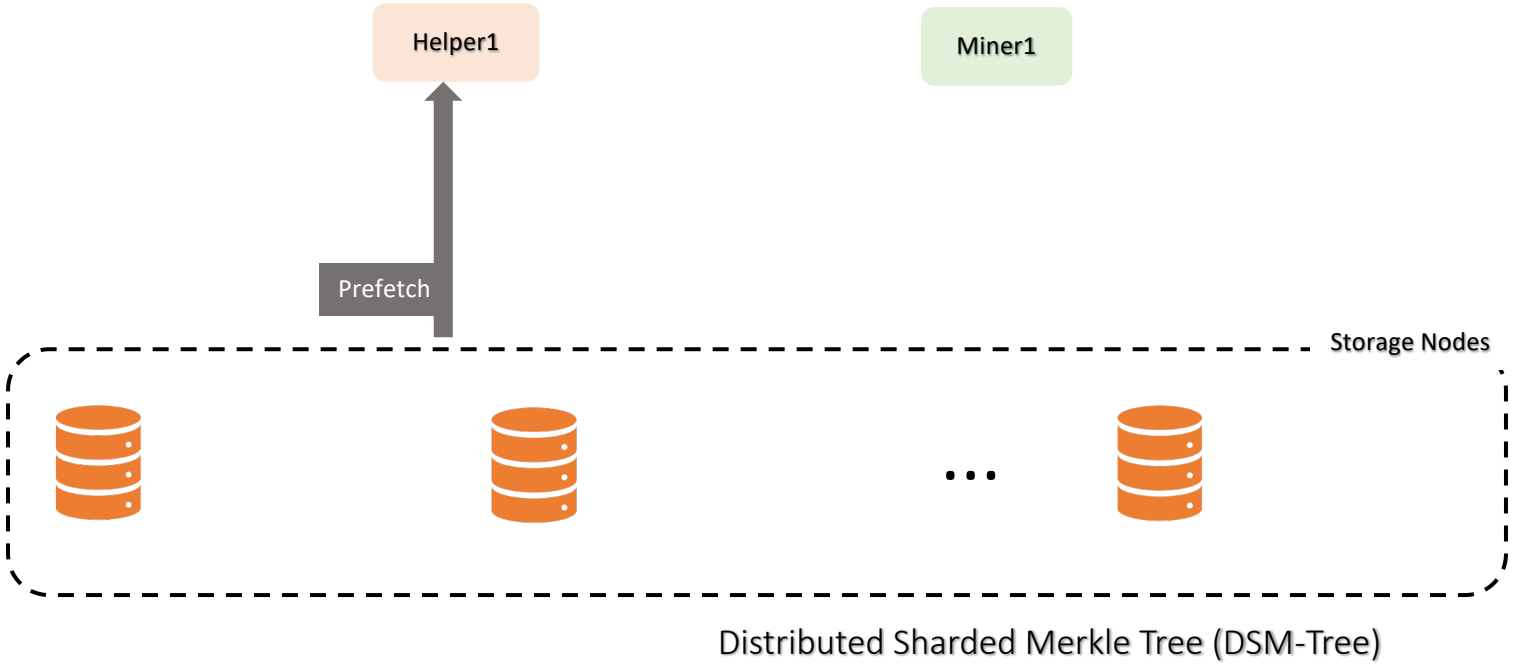
Handling Concurrent Operations

Concurrency and consistency



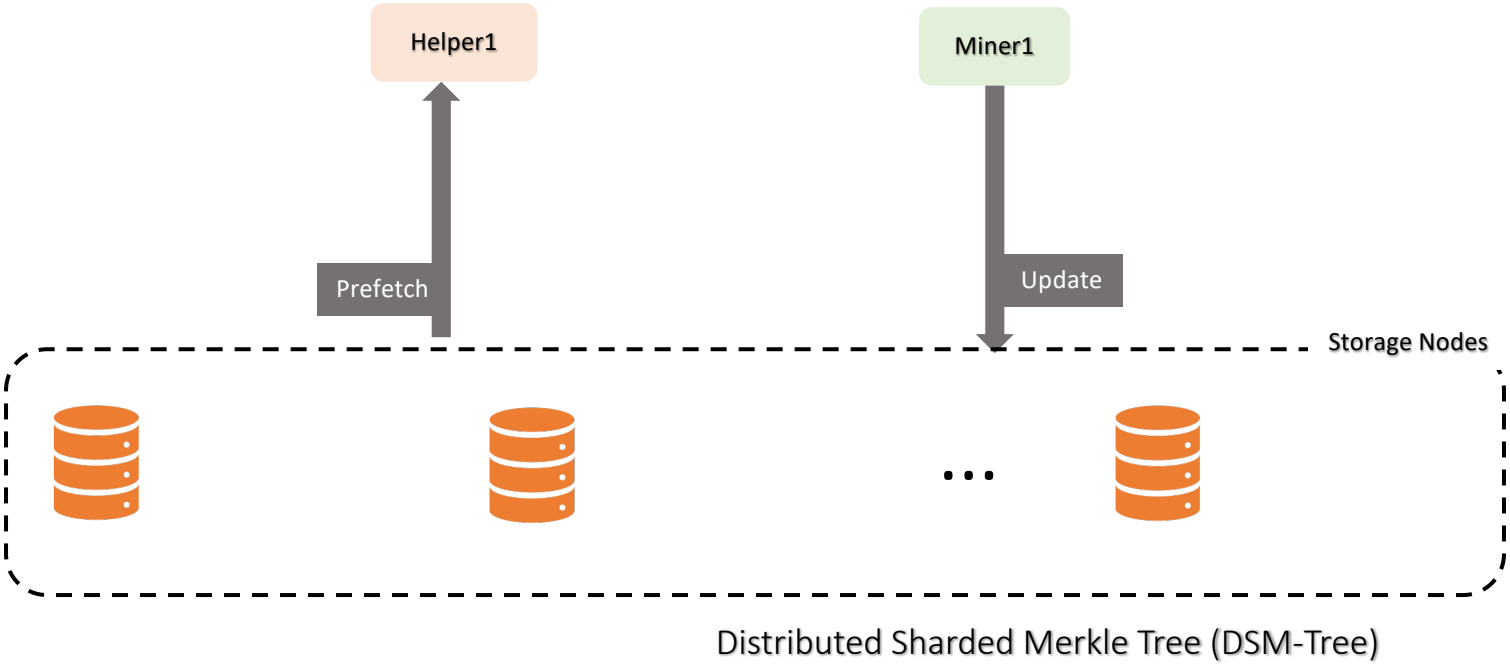
Handling Concurrent Operations

Concurrency and consistency



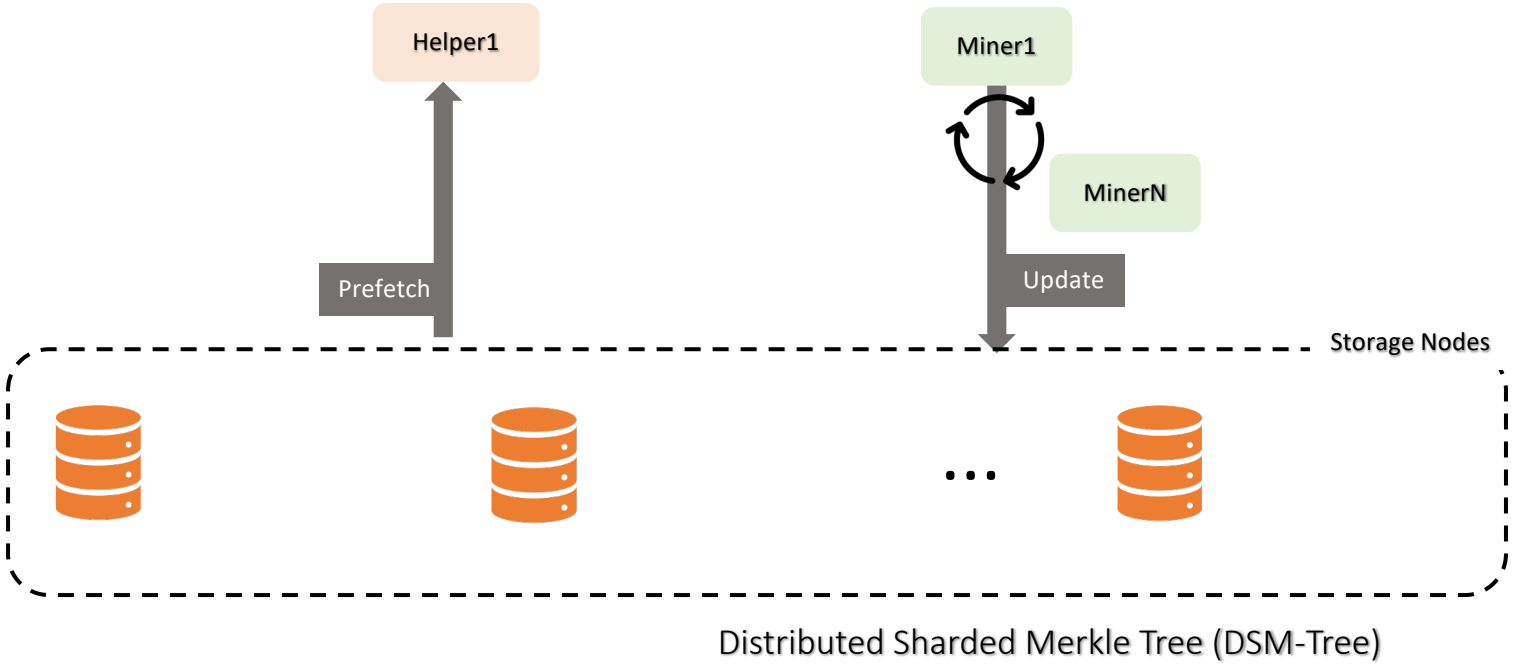
Handling Concurrent Operations

Concurrency and consistency



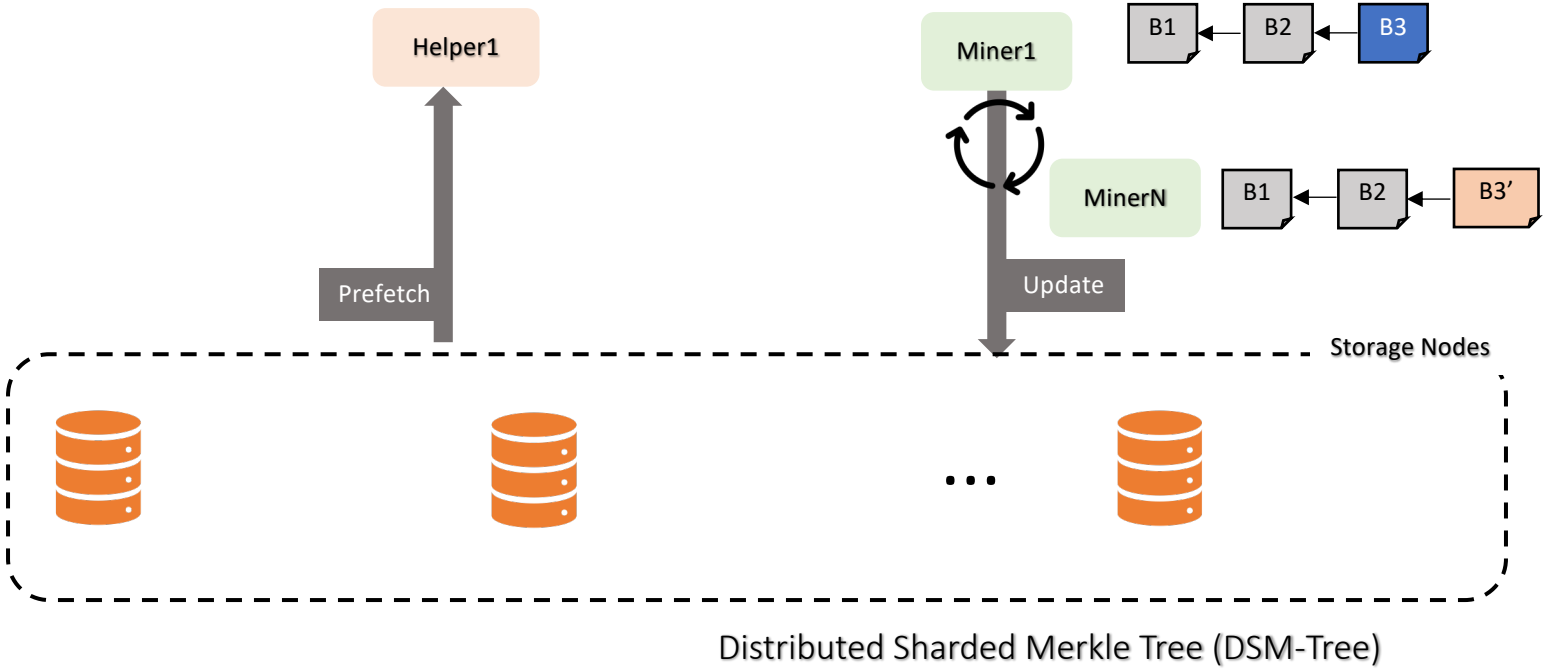
Handling Concurrent Operations

Concurrency and consistency



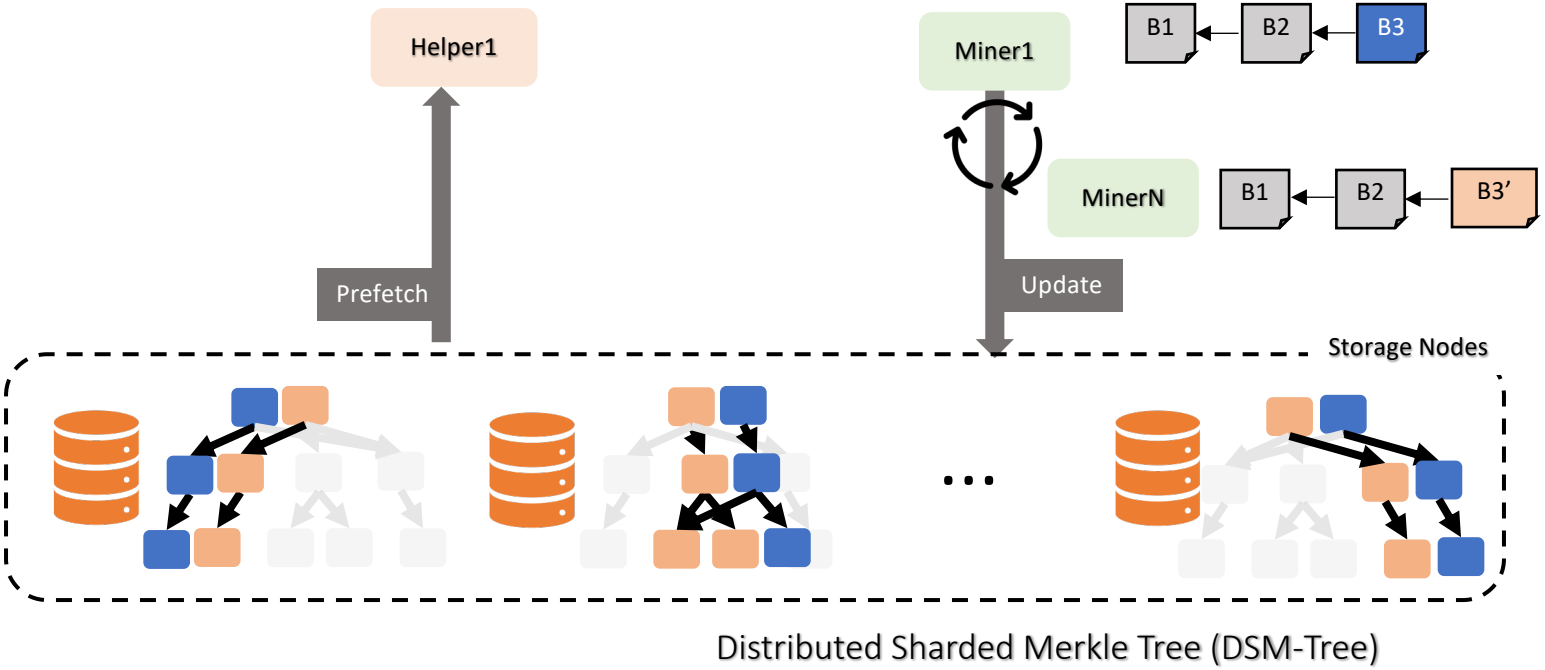
Handling Concurrent Operations

Concurrency and consistency



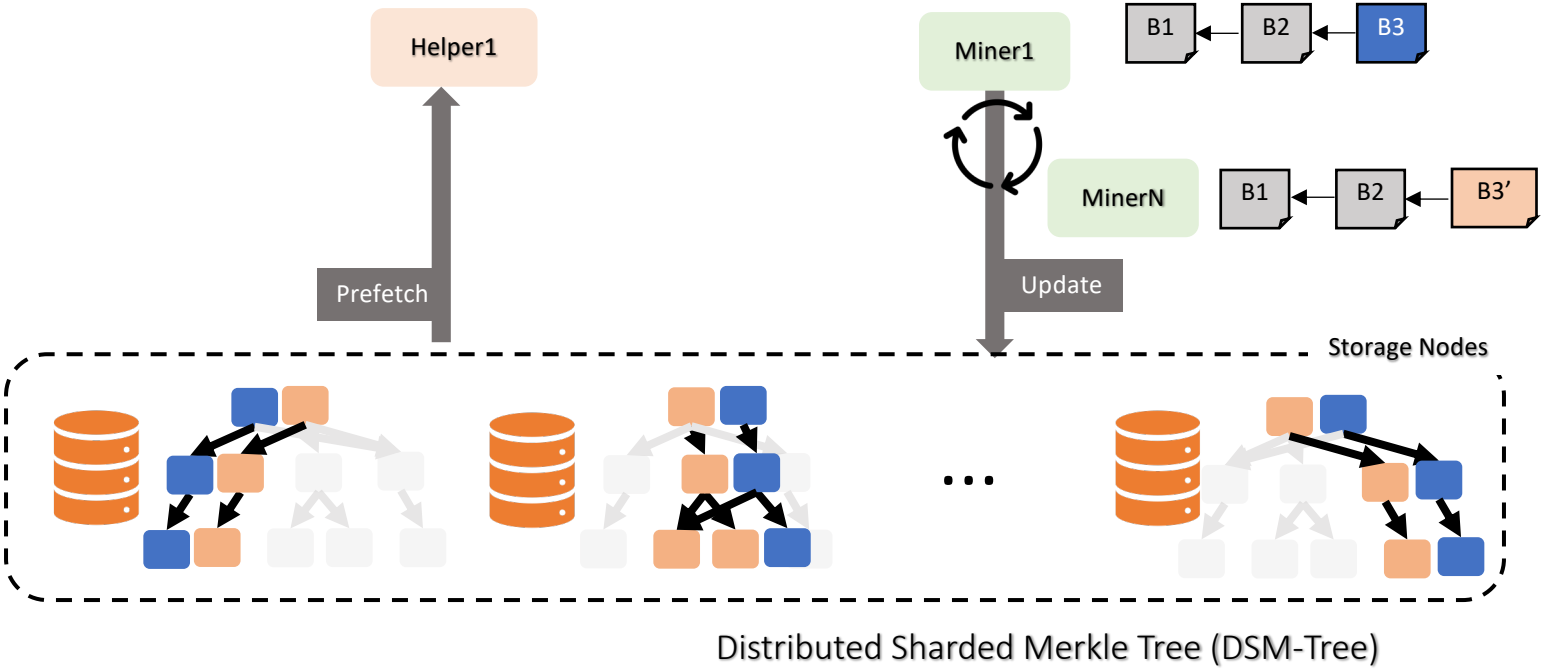
Handling Concurrent Operations

Concurrency and consistency



Handling Concurrent Operations

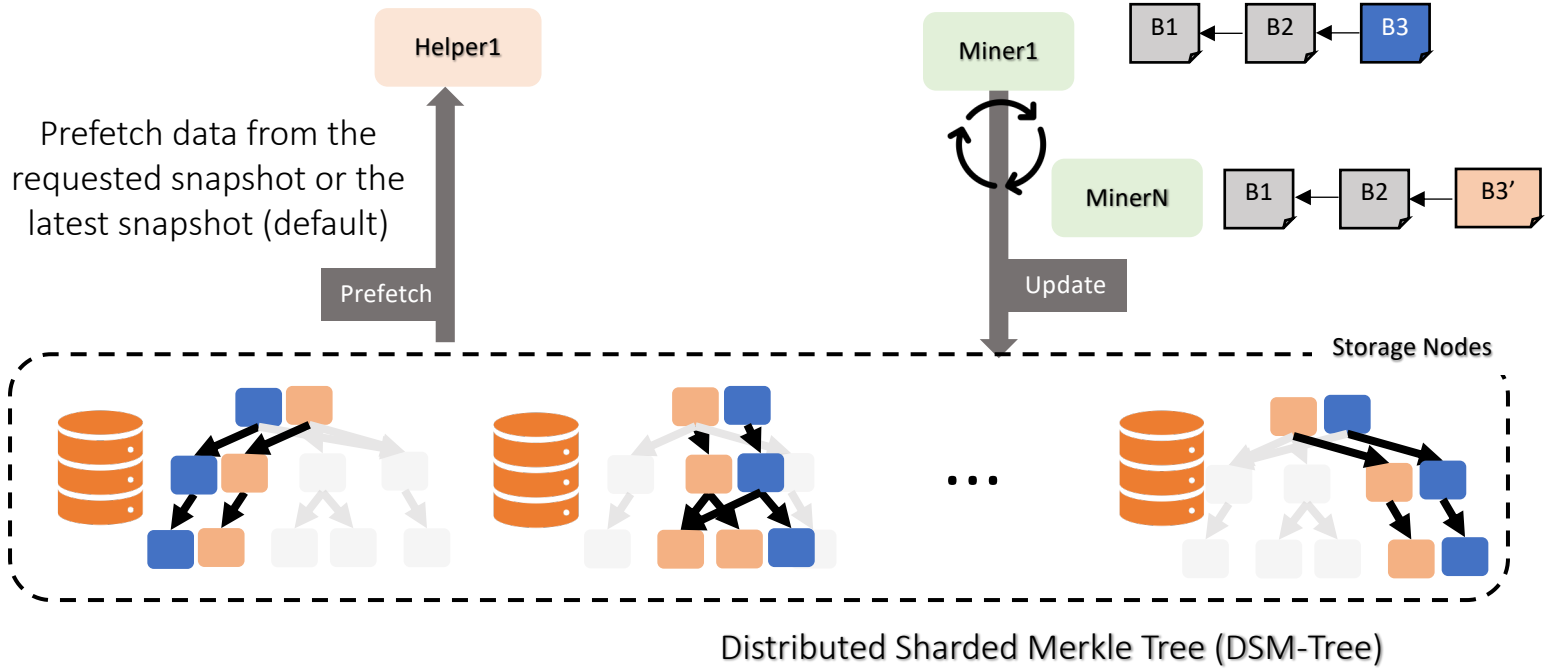
Concurrency and consistency



On updates, shards create new copies of data

Handling Concurrent Operations

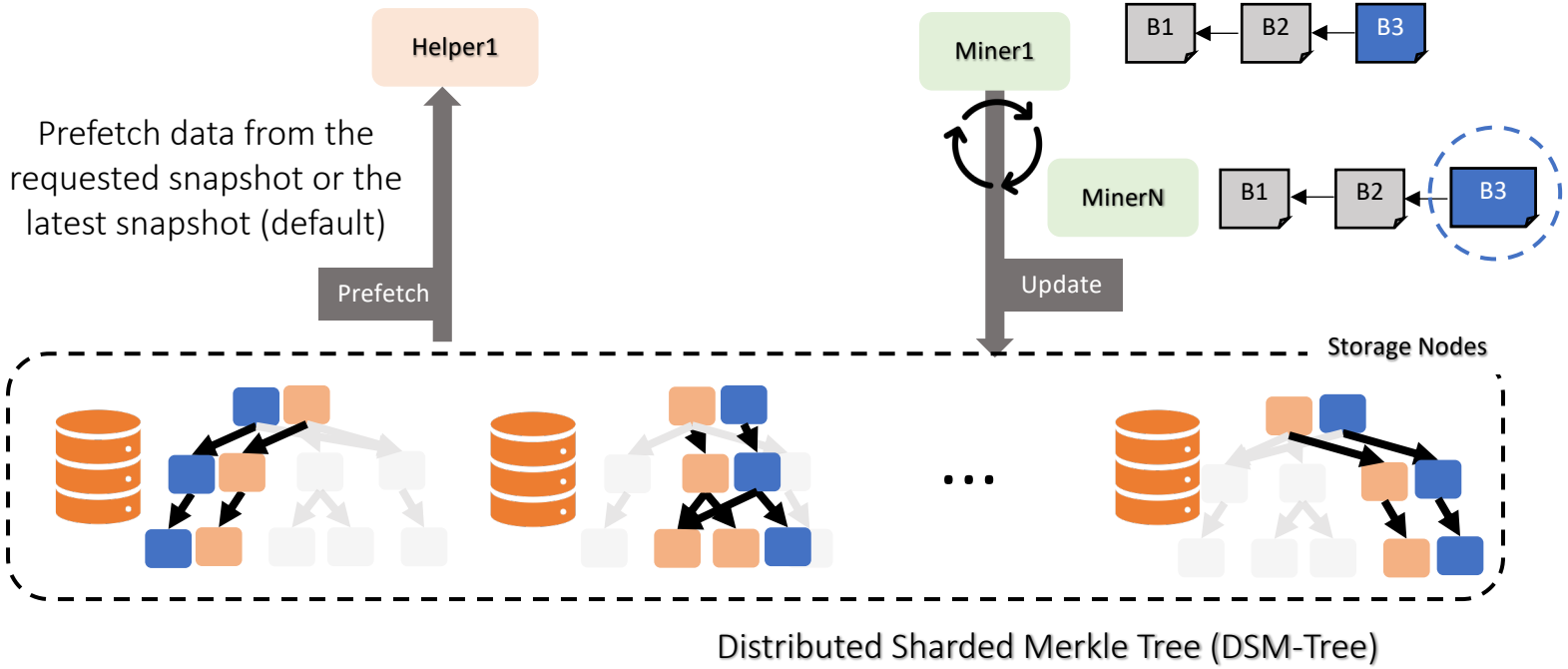
Concurrency and consistency



On updates, shards create new copies of data

Handling Concurrent Operations

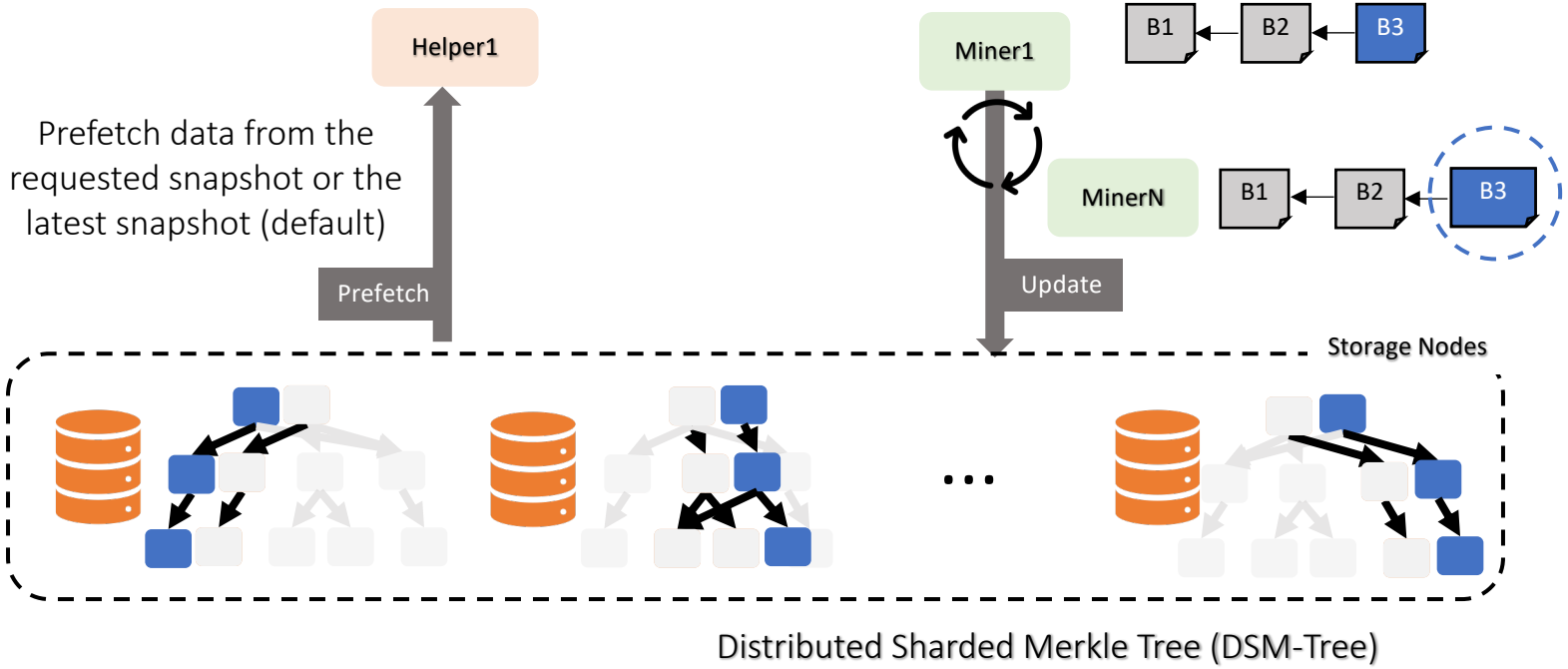
Concurrency and consistency



On updates, shards create new copies of data

Handling Concurrent Operations

Concurrency and consistency



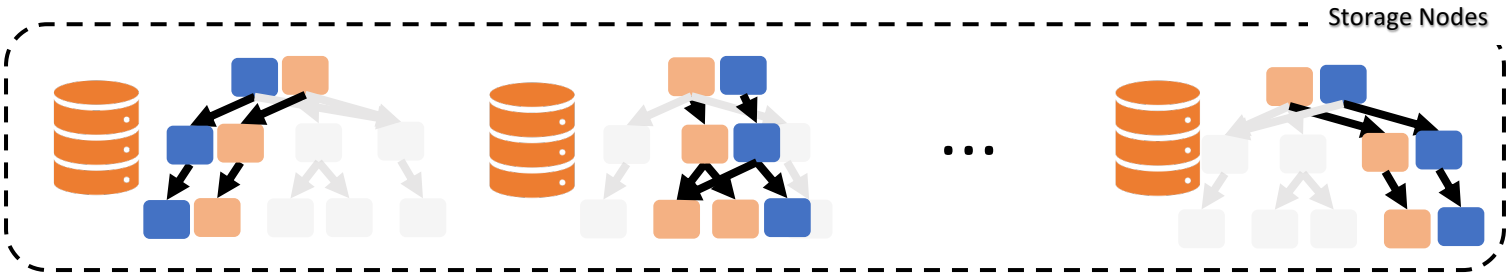
On updates, shards create new copies of data

Two-layered DSM-Tree

Concurrency and consistency

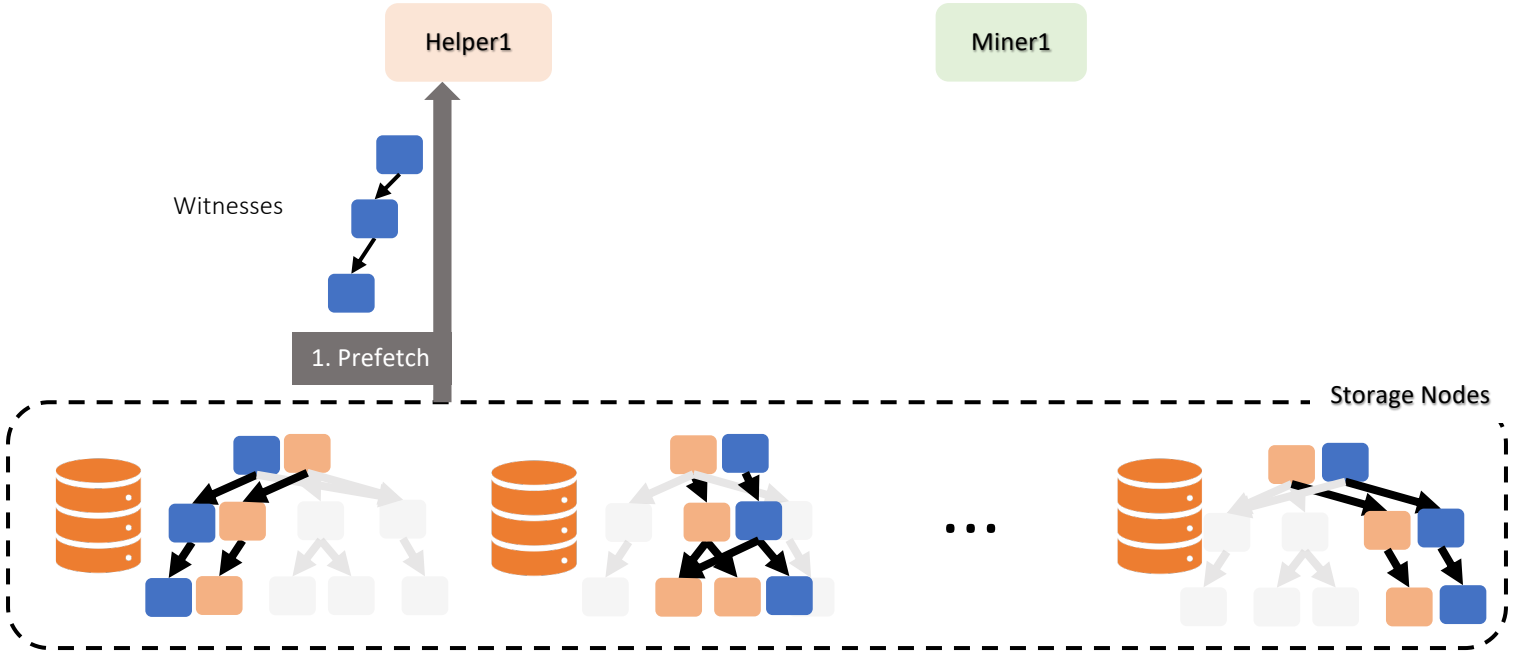
Helper1

Miner1



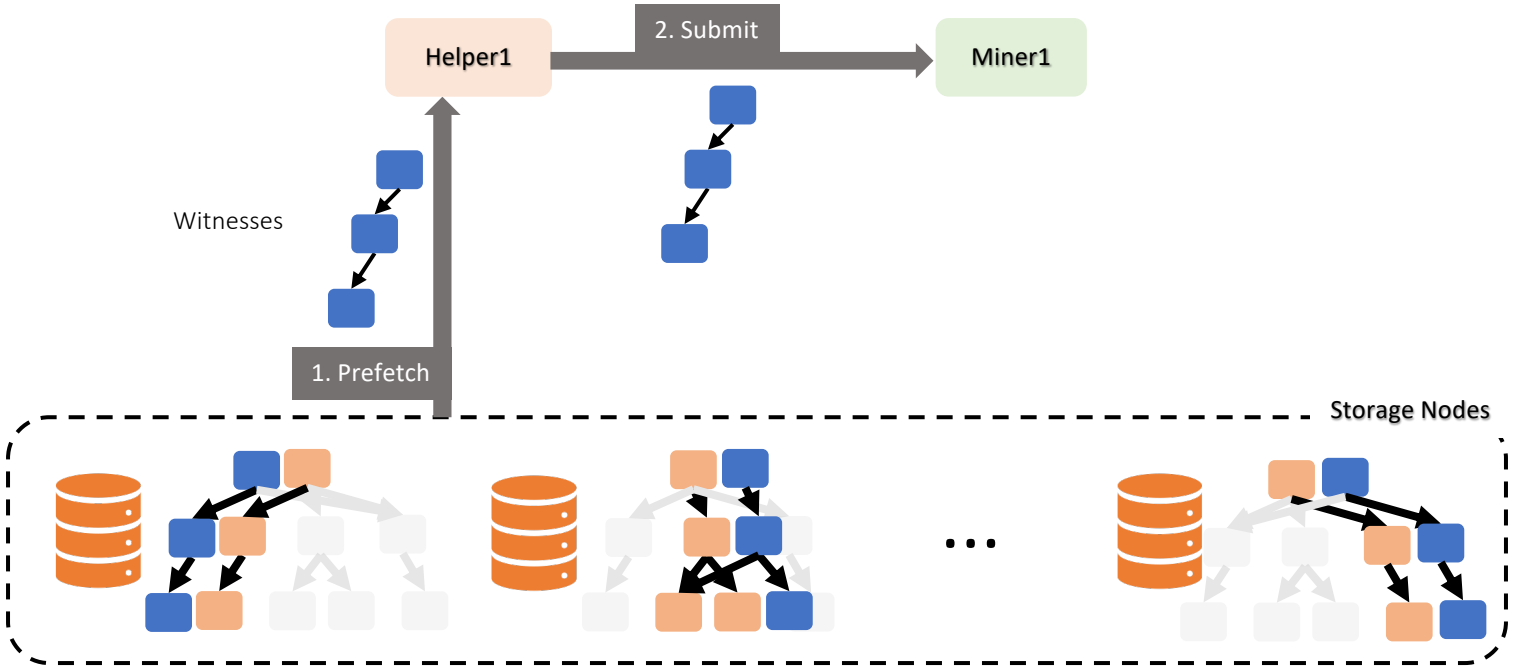
Two-layered DSM-Tree

Concurrency and consistency



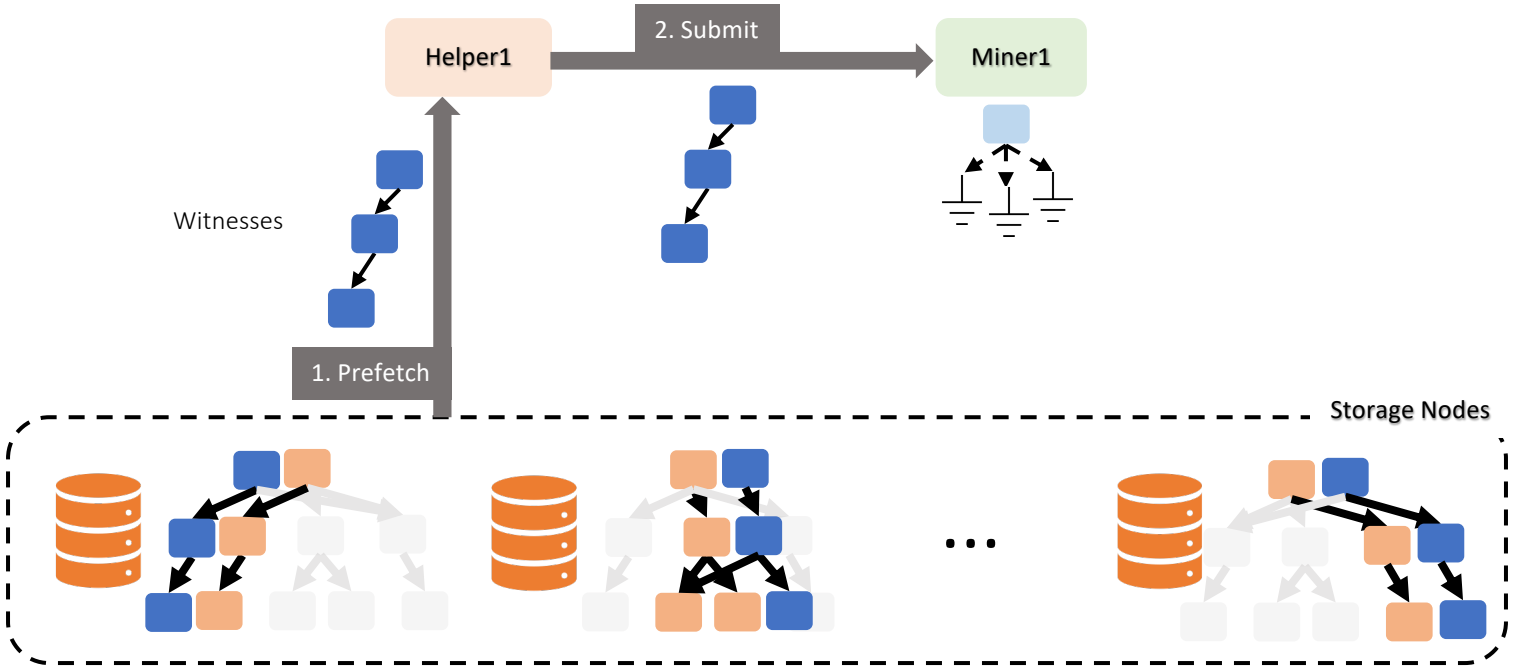
Two-layered DSM-Tree

Concurrency and consistency



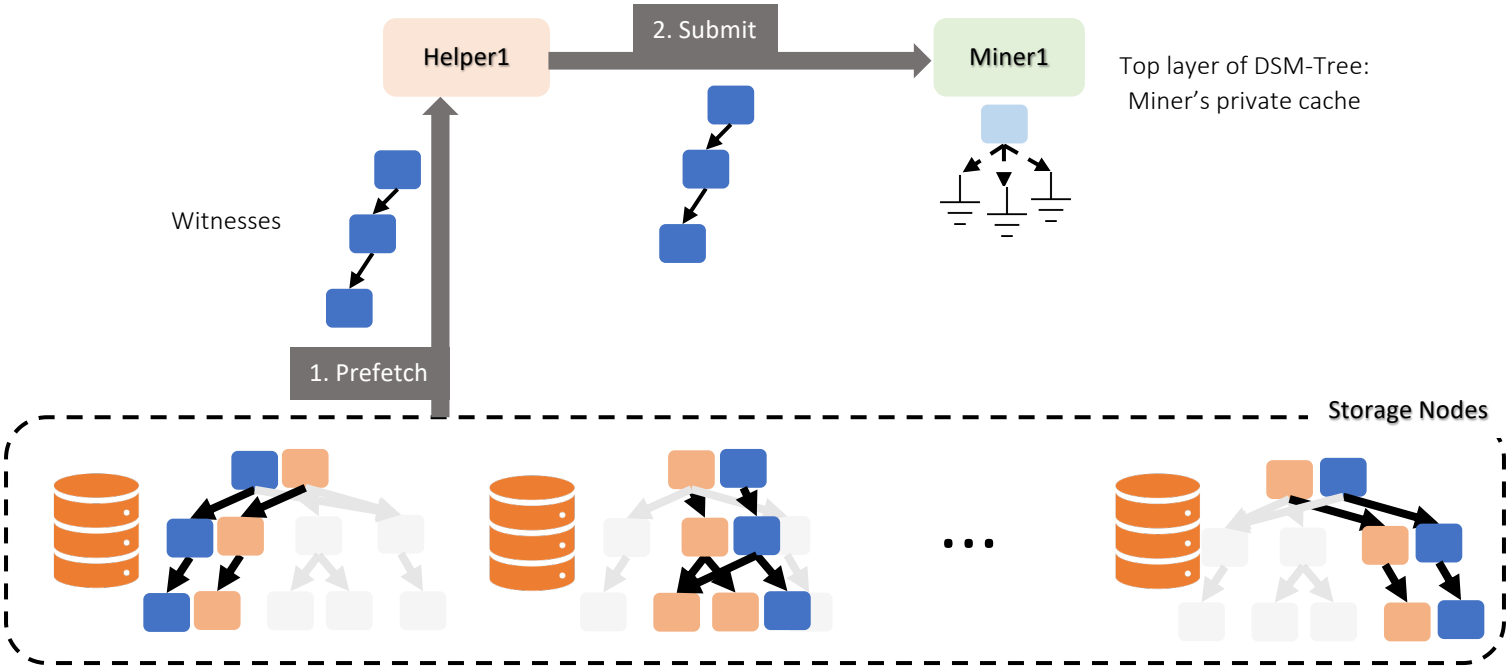
Two-layered DSM-Tree

Concurrency and consistency



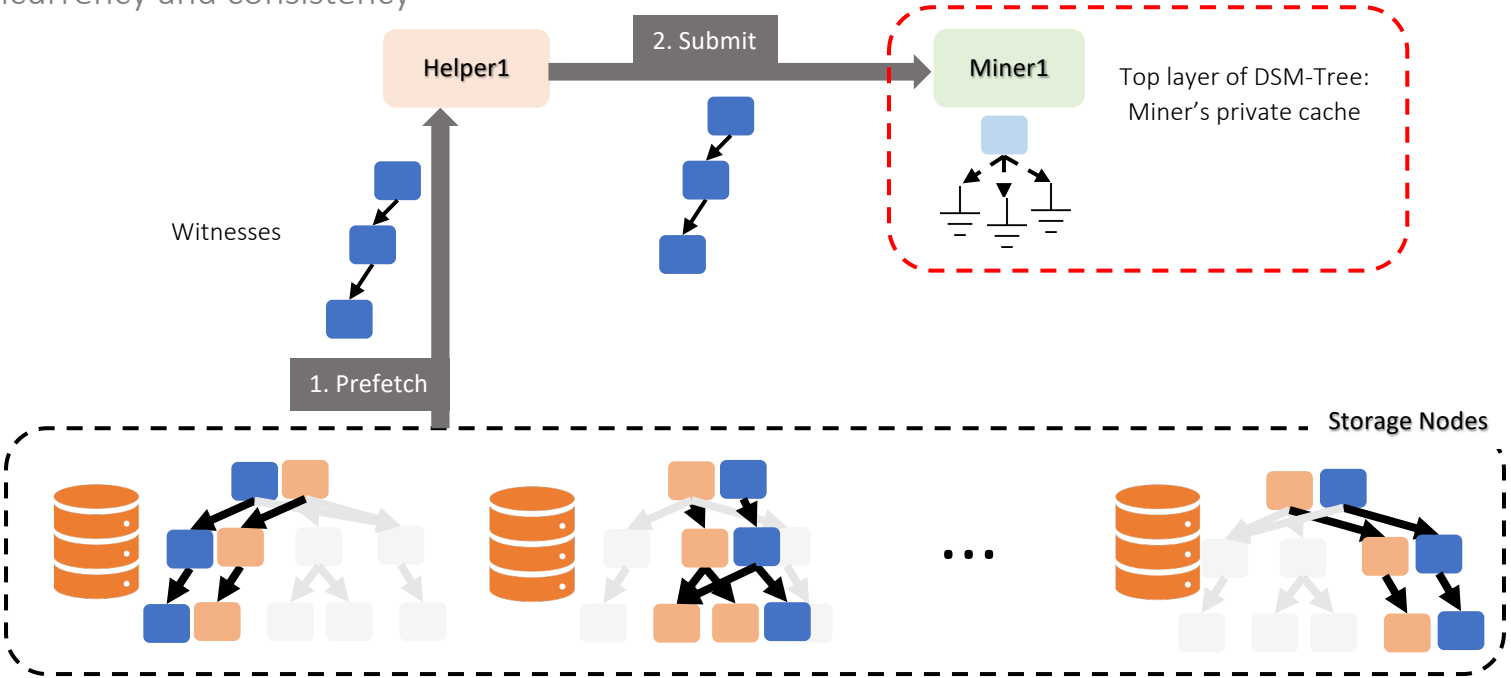
Two-layered DSM-Tree

Concurrency and consistency



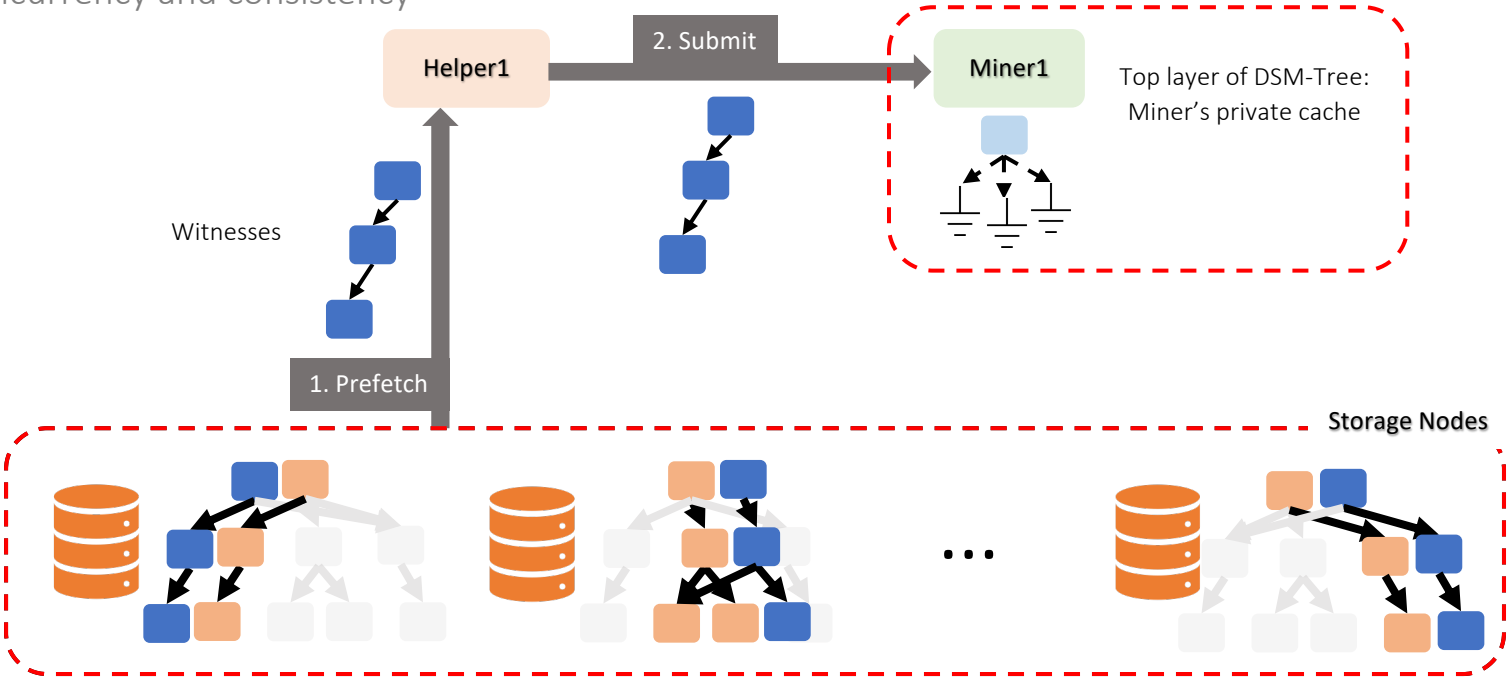
Two-layered DSM-Tree

Concurrency and consistency



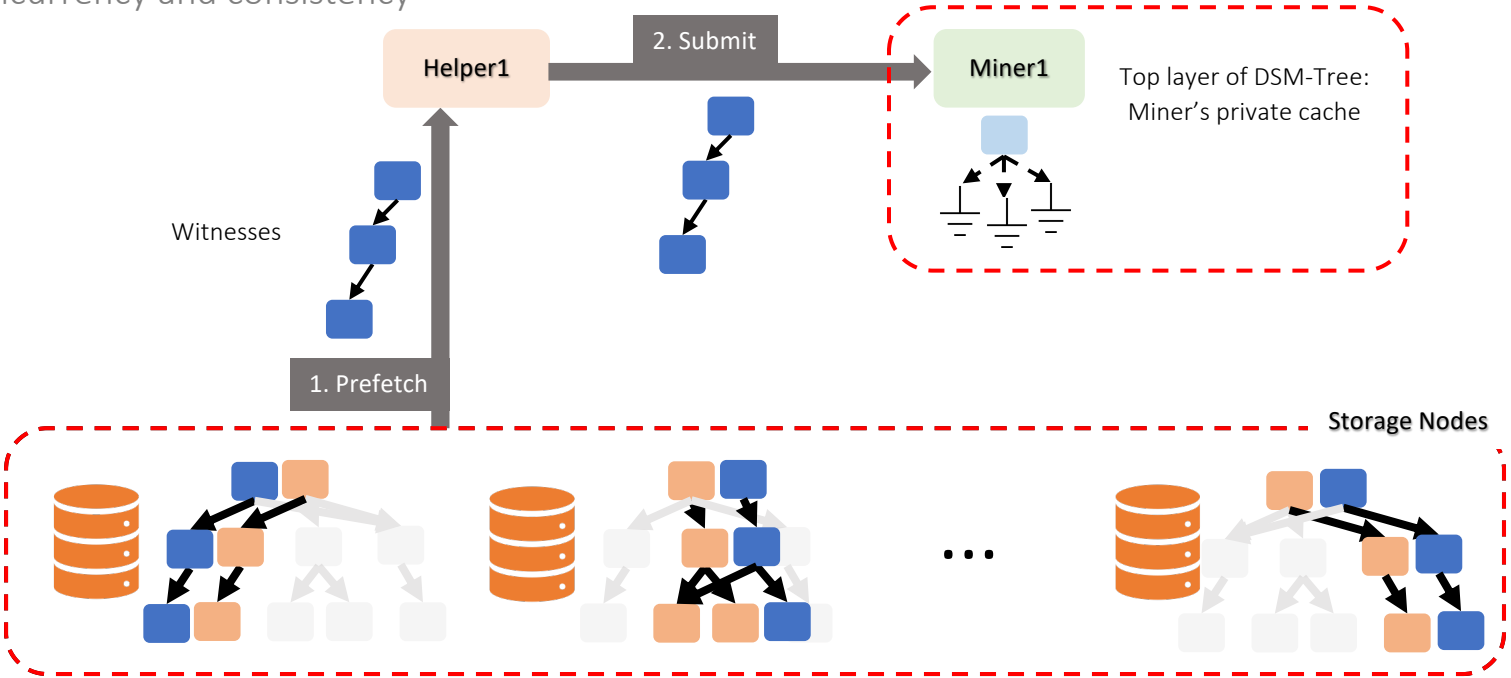
Two-layered DSM-Tree

Concurrency and consistency



Two-layered DSM-Tree

Concurrency and consistency



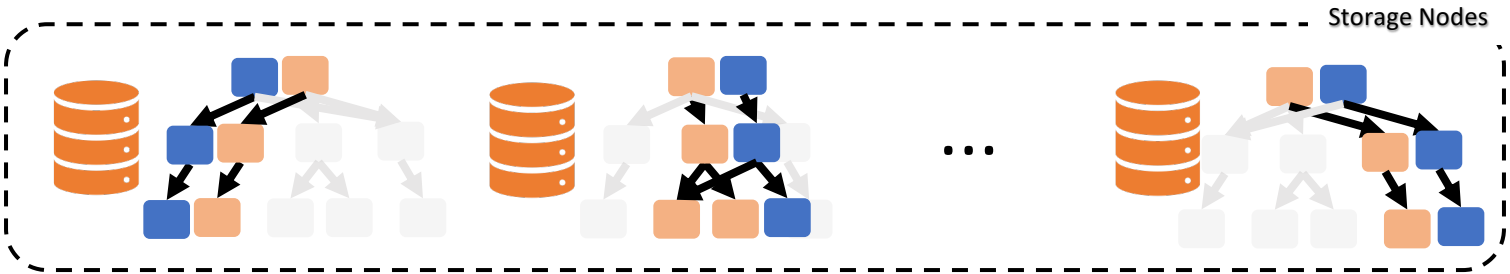
The DSM-Tree layers collaborate with each other to reduce network traffic

Two-layered DSM-Tree

Reduce network traffic

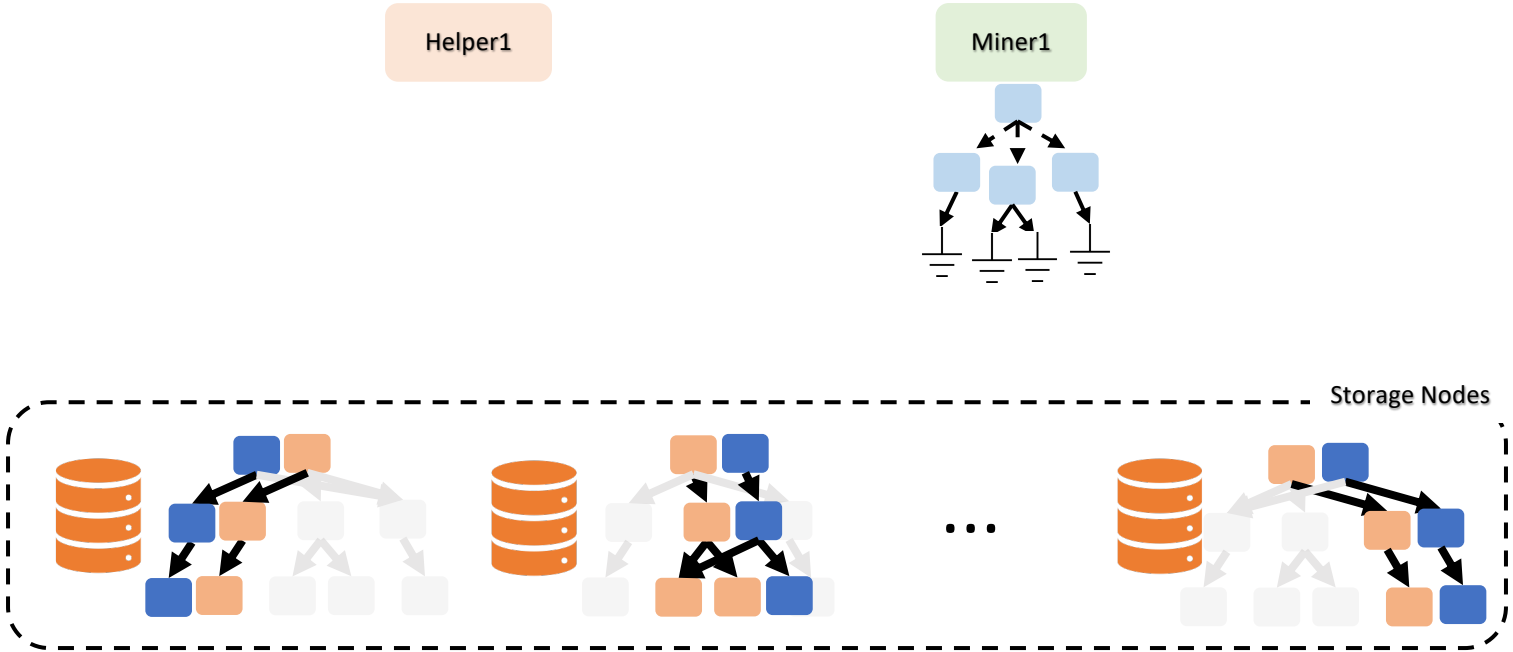
Helper1

Miner1



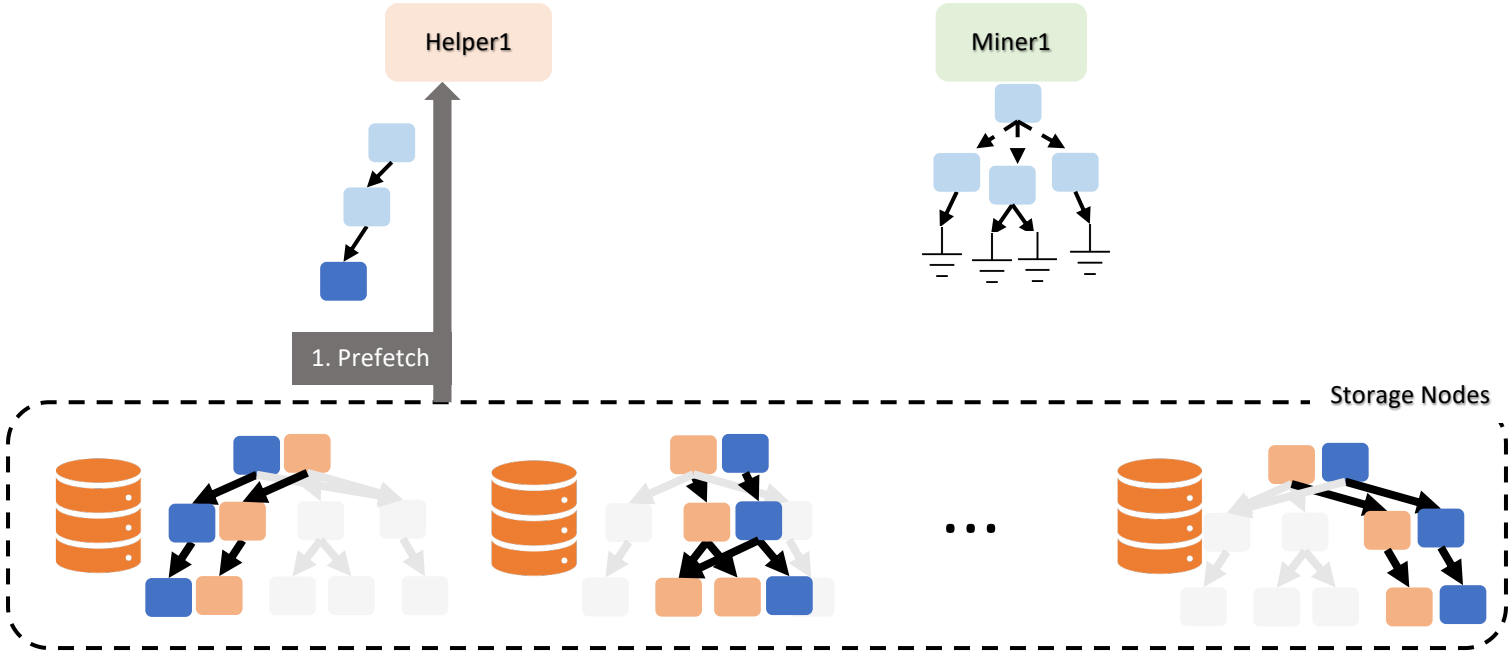
Two-layered DSM-Tree

Reduce network traffic



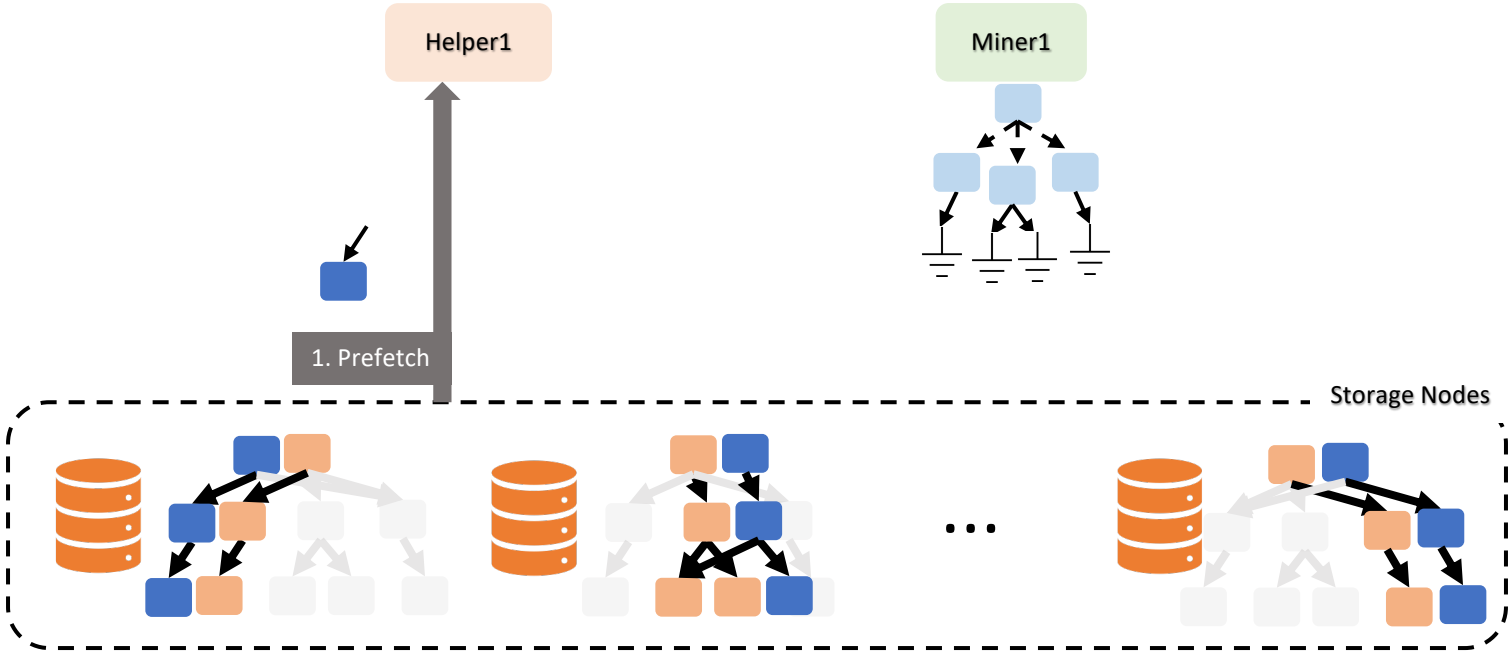
Two-layered DSM-Tree

Reduce network traffic



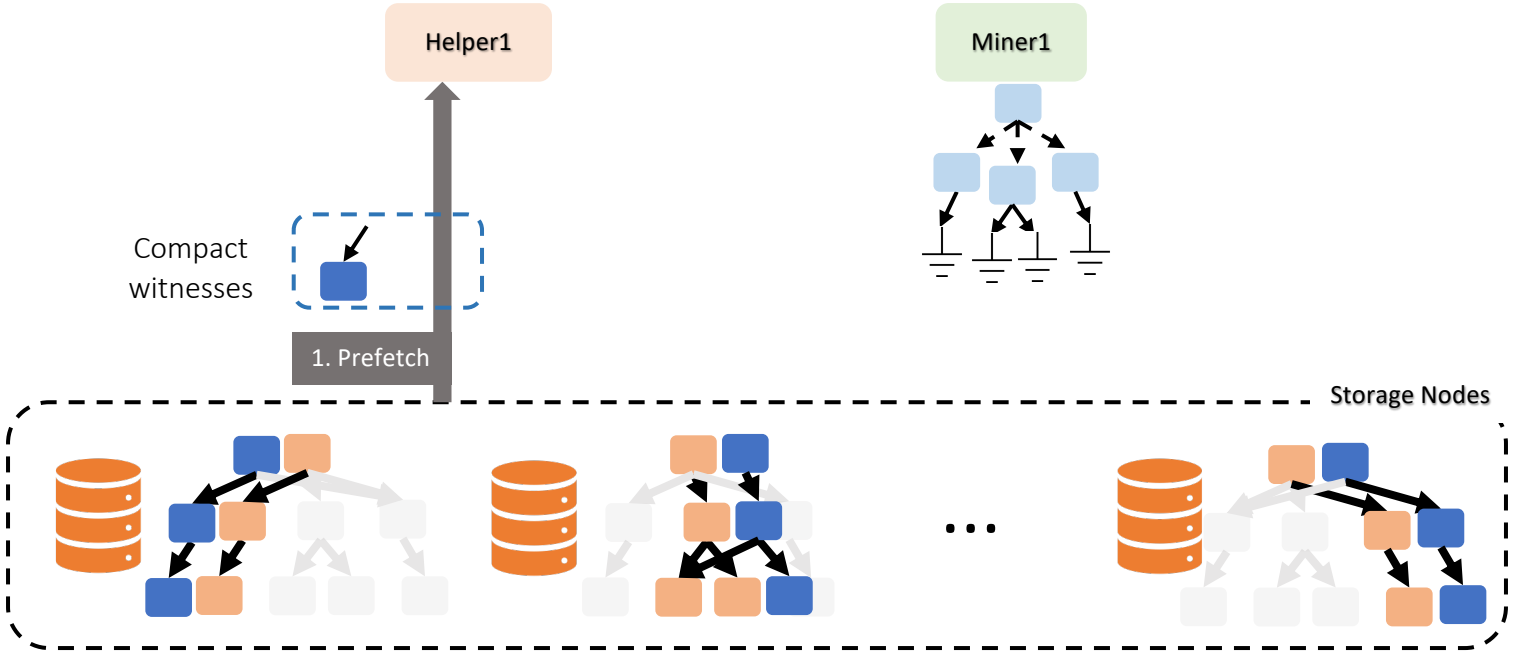
Two-layered DSM-Tree

Reduce network traffic



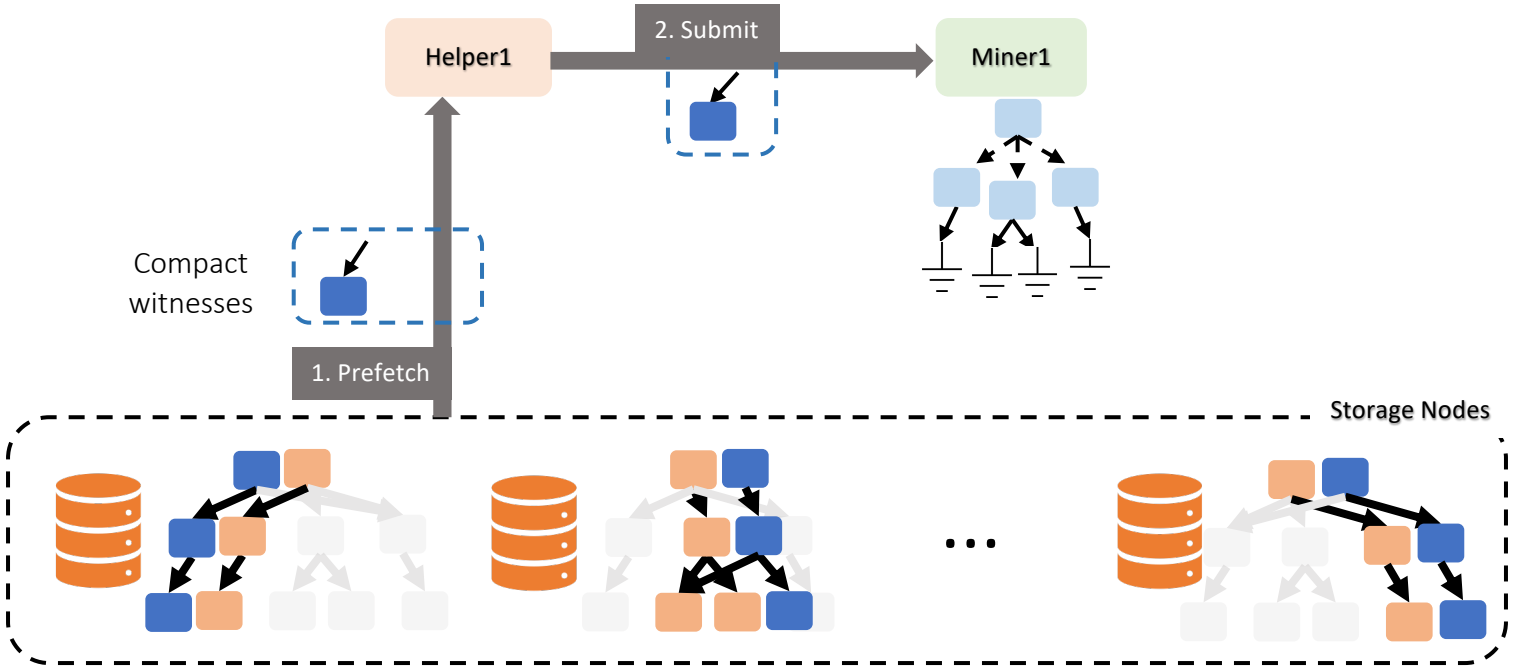
Two-layered DSM-Tree

Reduce network traffic



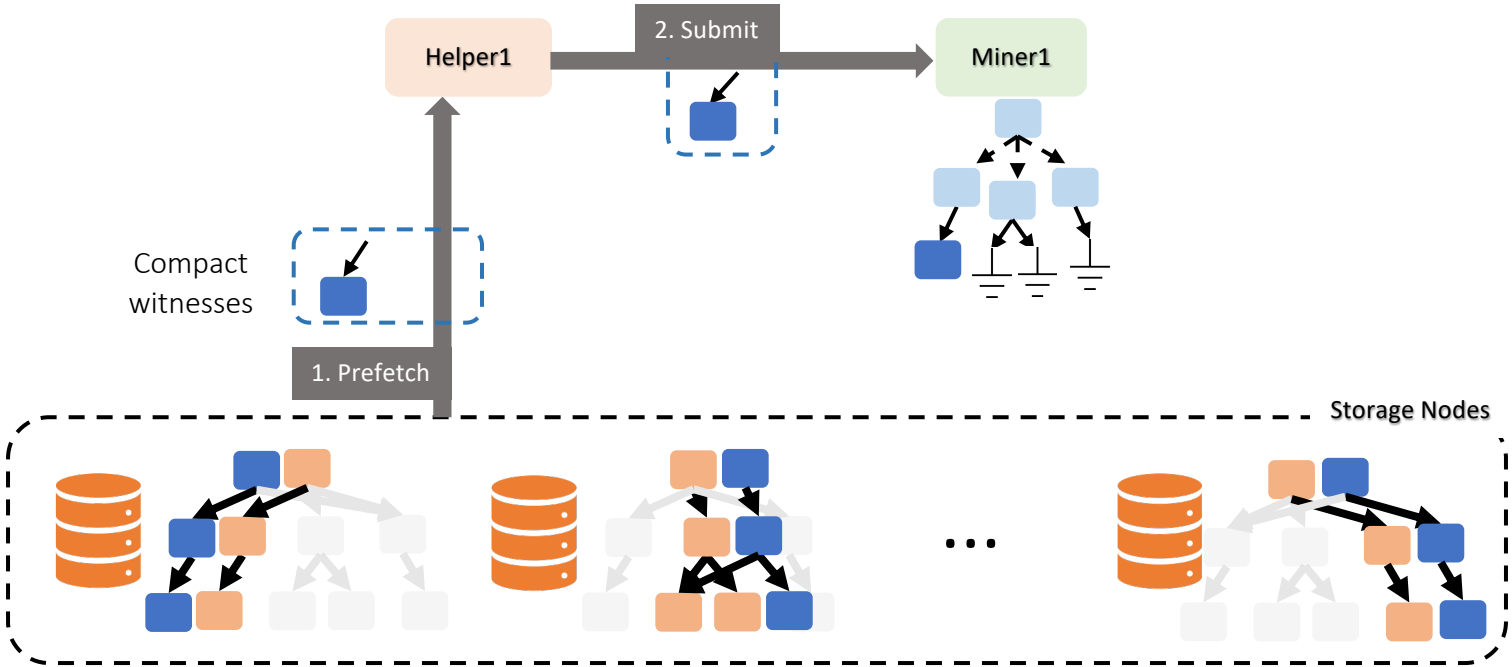
Two-layered DSM-Tree

Reduce network traffic



Two-layered DSM-Tree

Reduce network traffic



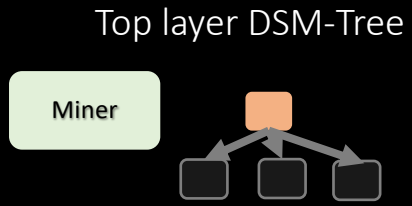
Cross-layer optimizations reduce network traffic by up to 95%

Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path

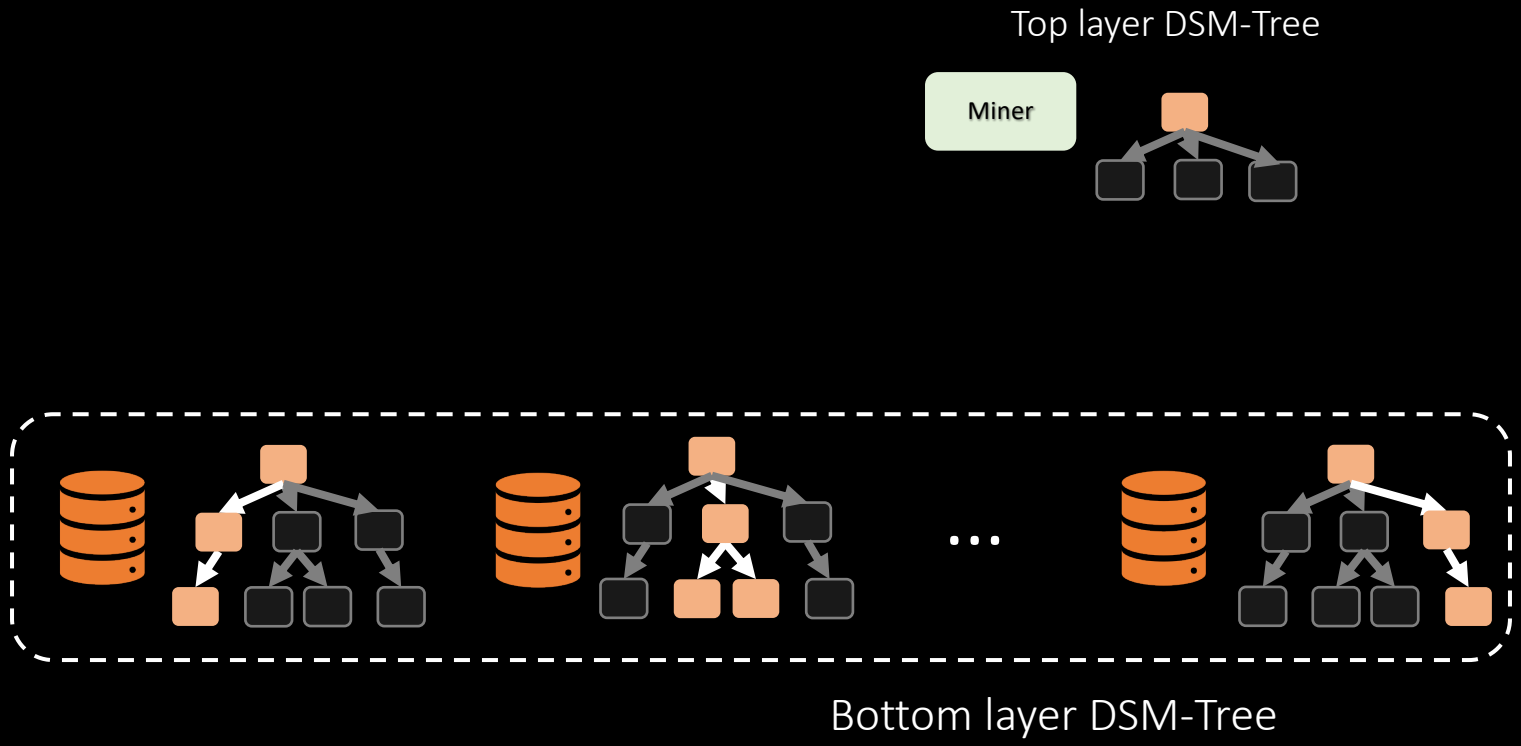
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



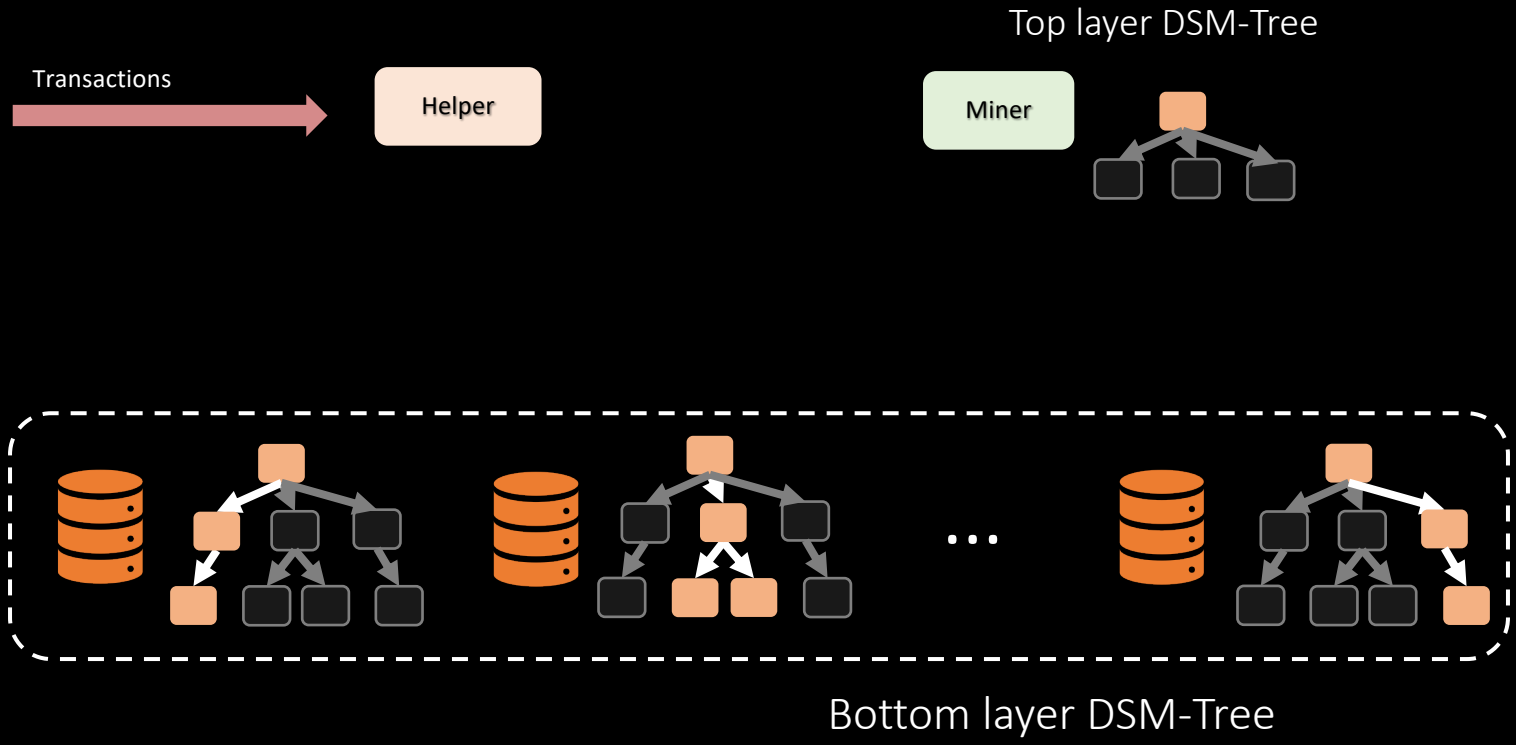
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



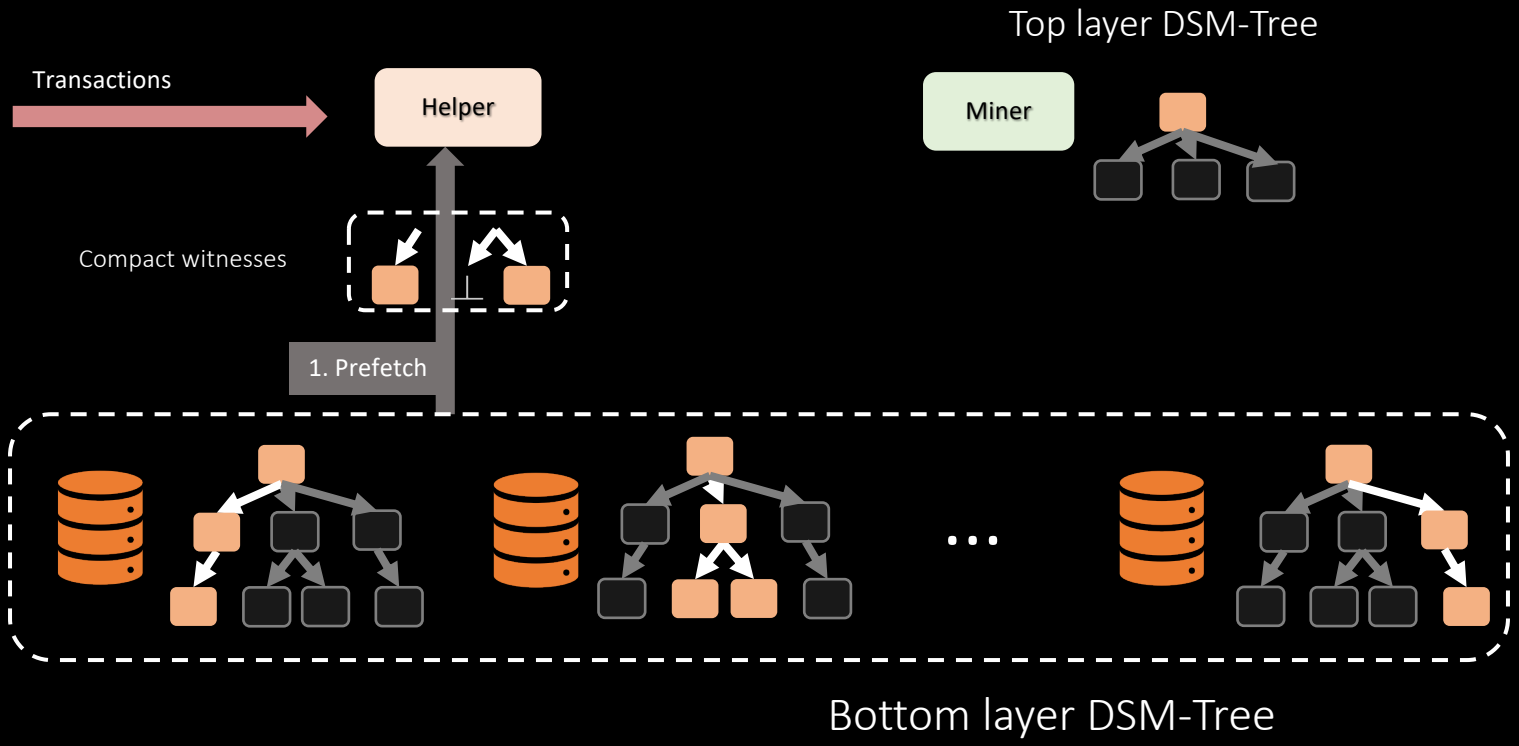
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



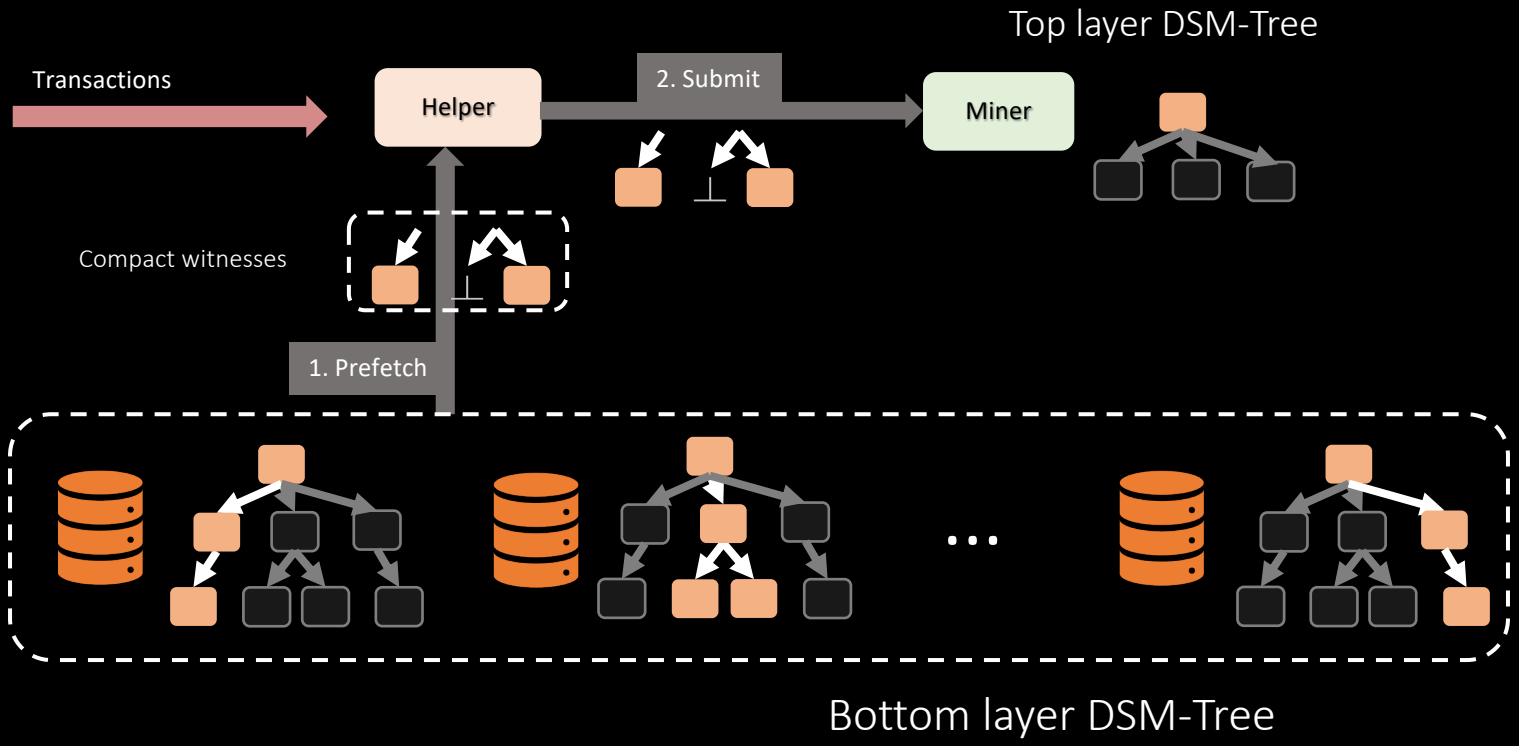
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



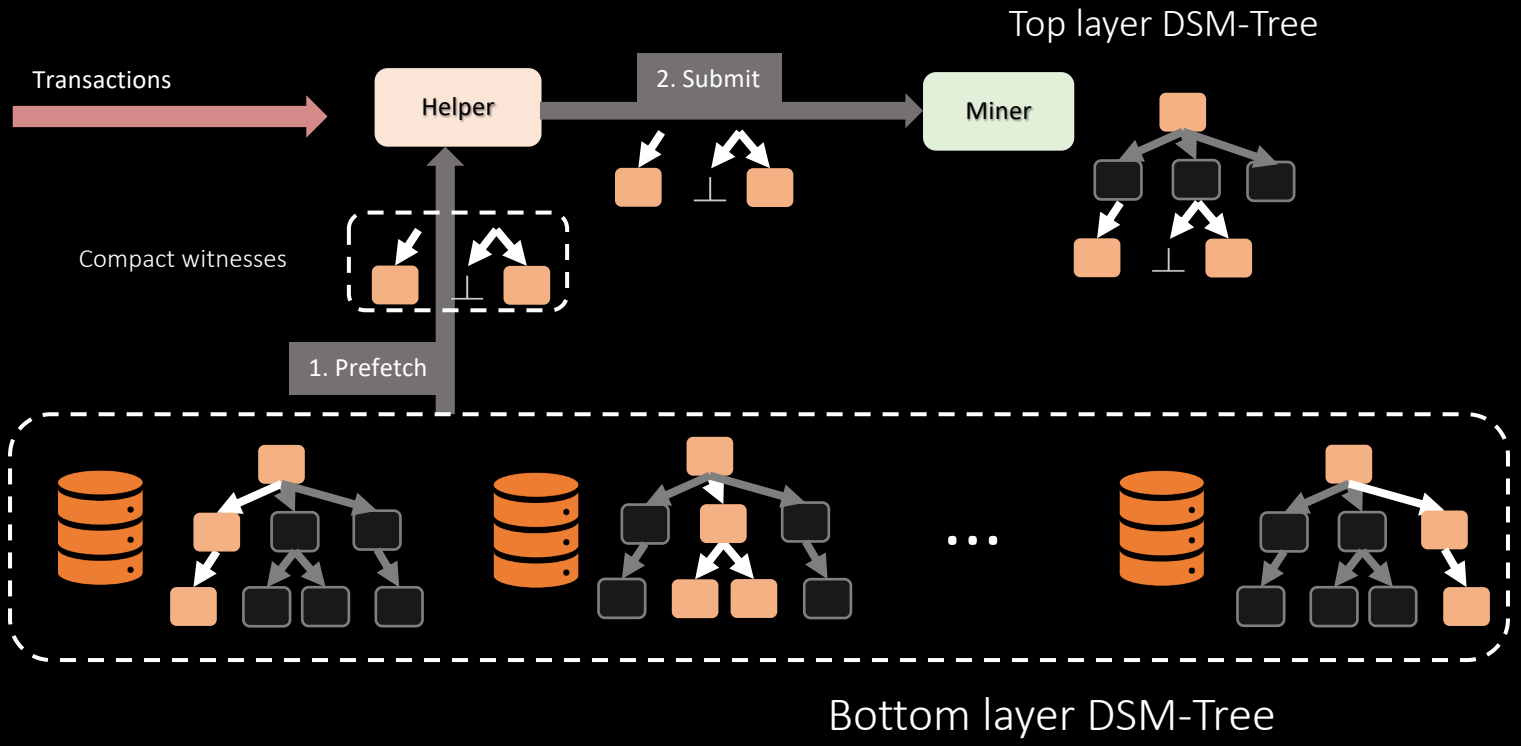
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



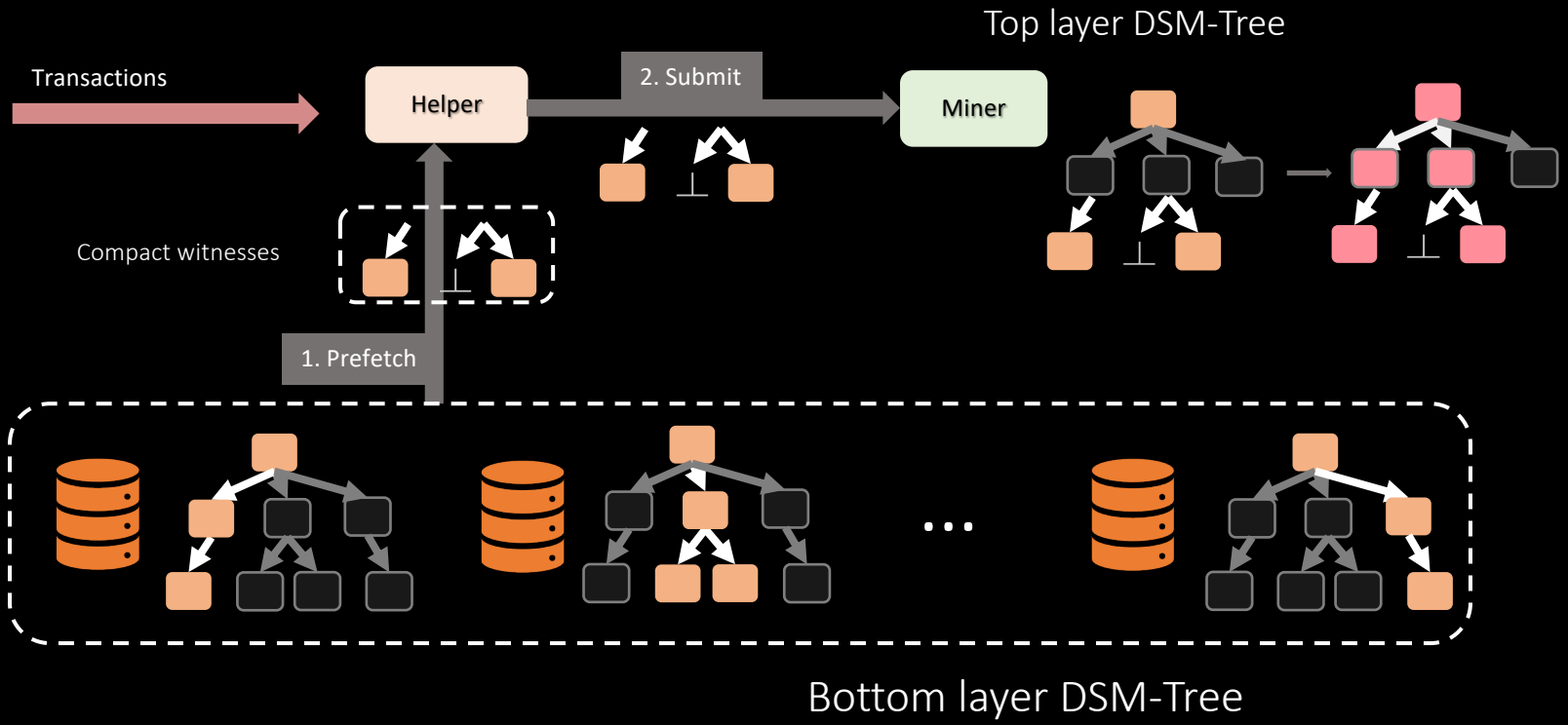
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



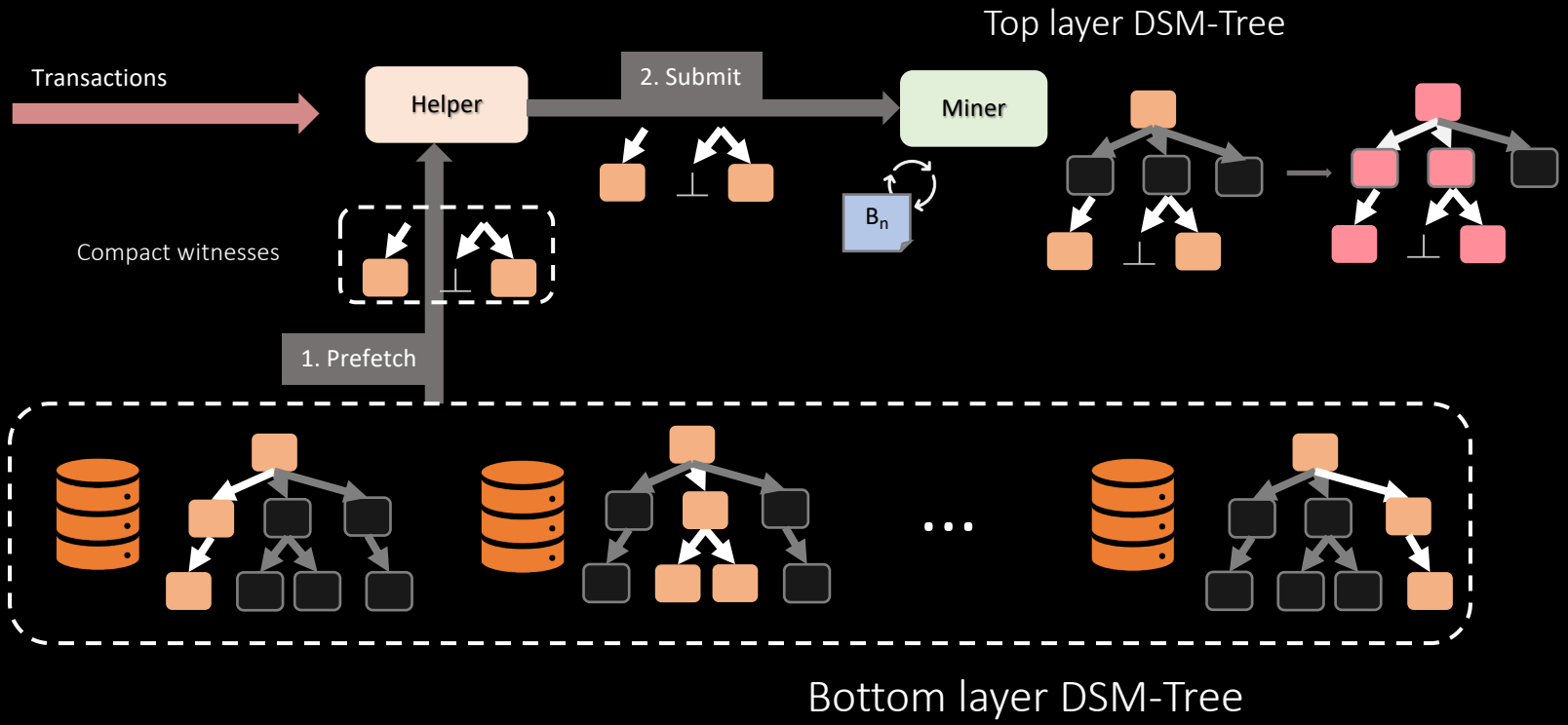
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



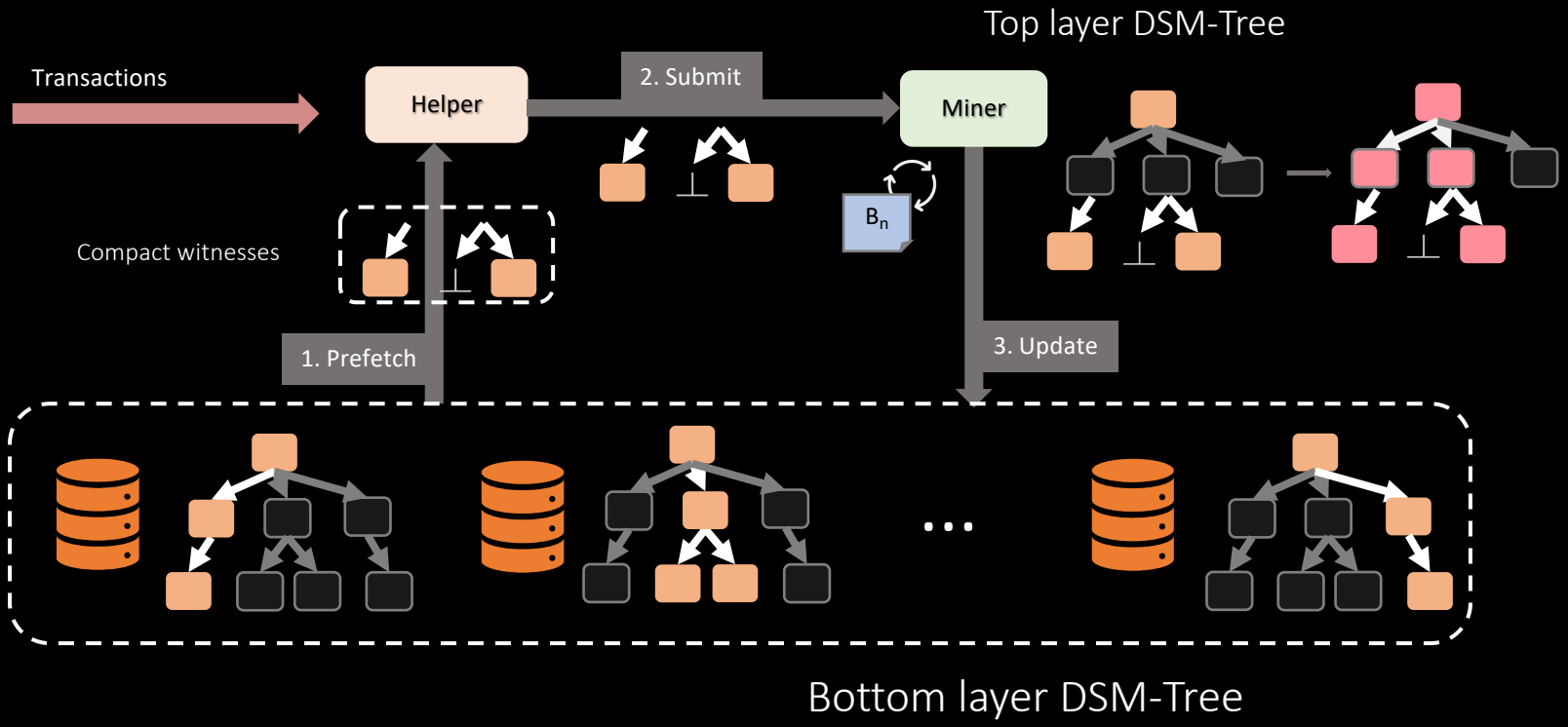
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



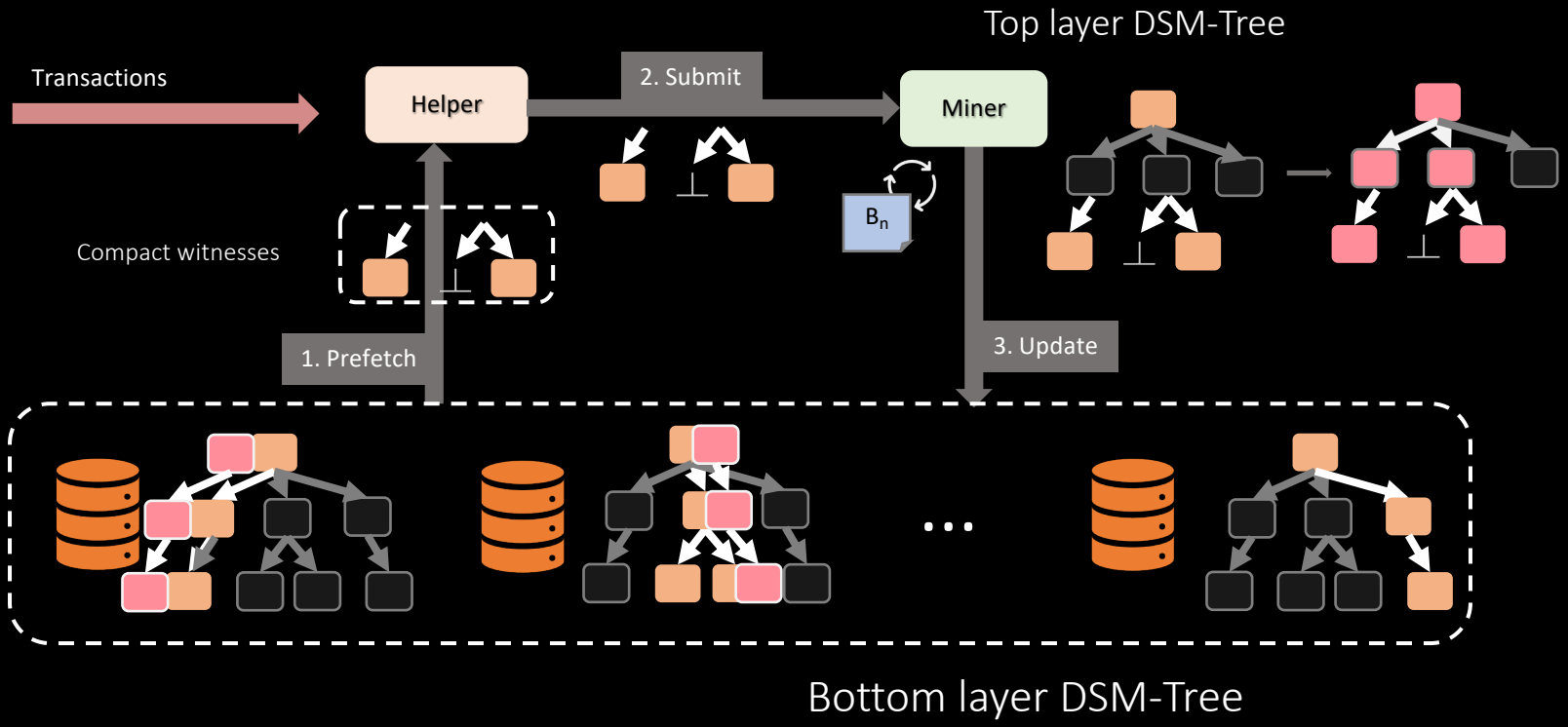
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



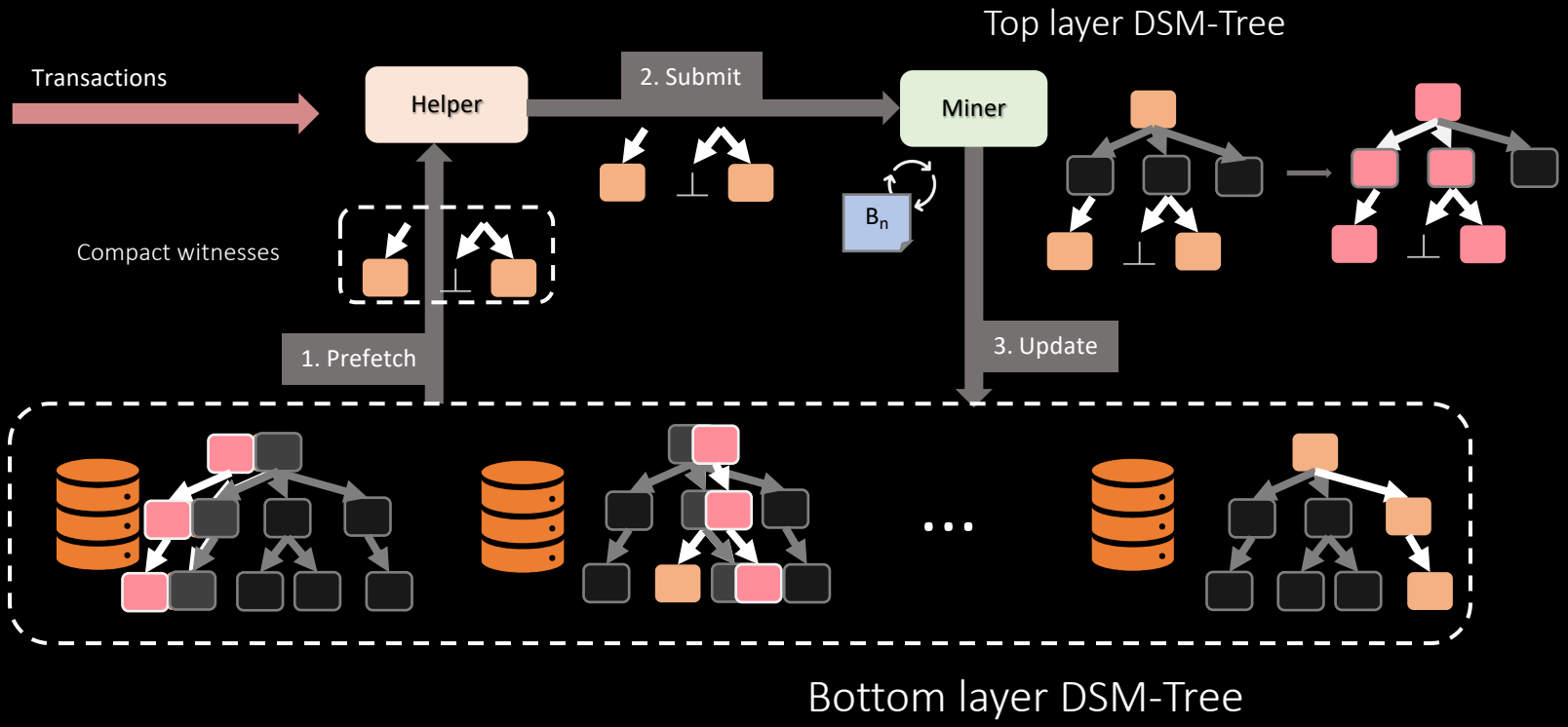
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



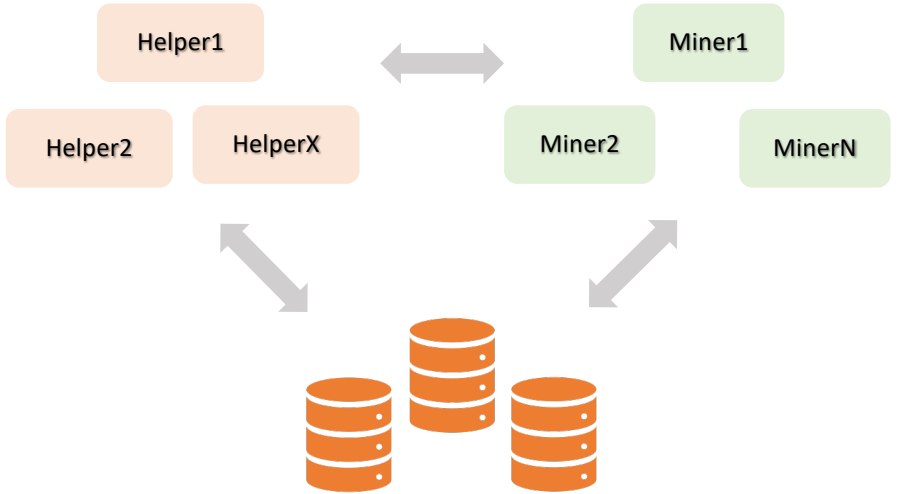
Life of a Transaction in RainBlock

Miners do not perform I/O in the critical path



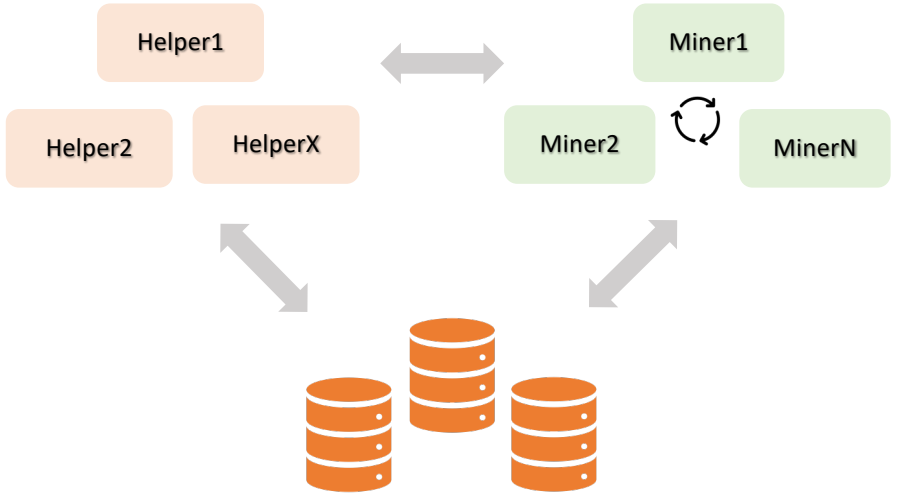
Trust, Safety, and Security

- Trust assumptions
  - All components work **without** trust



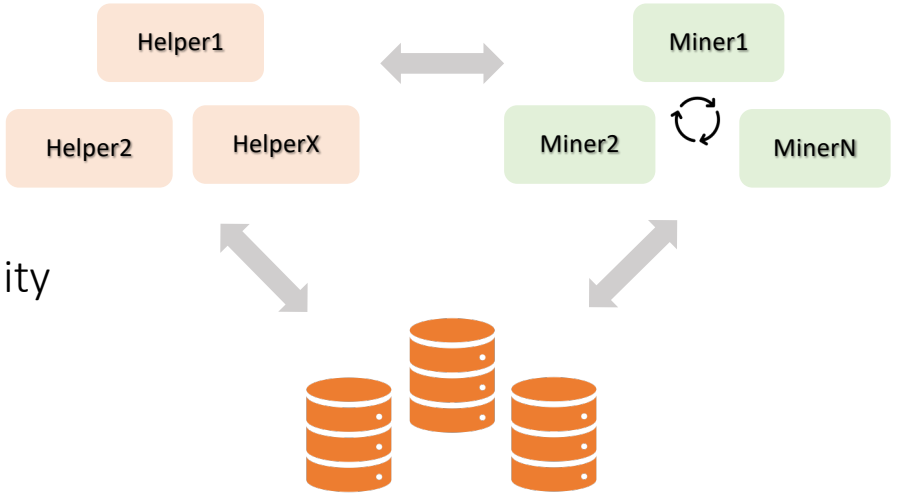
Trust, Safety, and Security

- Trust assumptions
  - All components work **without** trust
- Safety and Security
  - PoW remains unchanged



Trust, Safety, and Security

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



Evaluation

Experimental Setup

Evaluation

Experimental Setup

- Amazon EC2 m4.2xlarge instances
  - 32GB RAM
  - 48 threads per machine

Evaluation

Experimental Setup

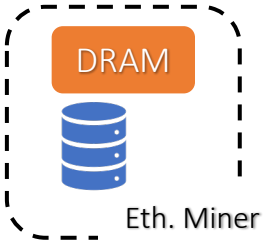
- Amazon EC2 m4.2xlarge instances
  - 32GB RAM
  - 48 threads per machine
- Storage nodes, miners, and I/O-Helpers are deployed on their own instance

Evaluation

Experimental Setup

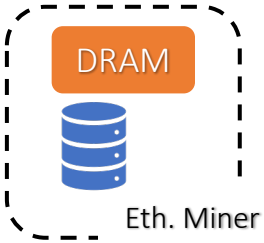
- Amazon EC2 m4.2xlarge instances
  - 32GB RAM
  - 48 threads per machine
- Storage nodes, miners, and I/O-Helpers are deployed on their own instance
- Workloads reflecting transactions in the public Ethereum network

Evaluation



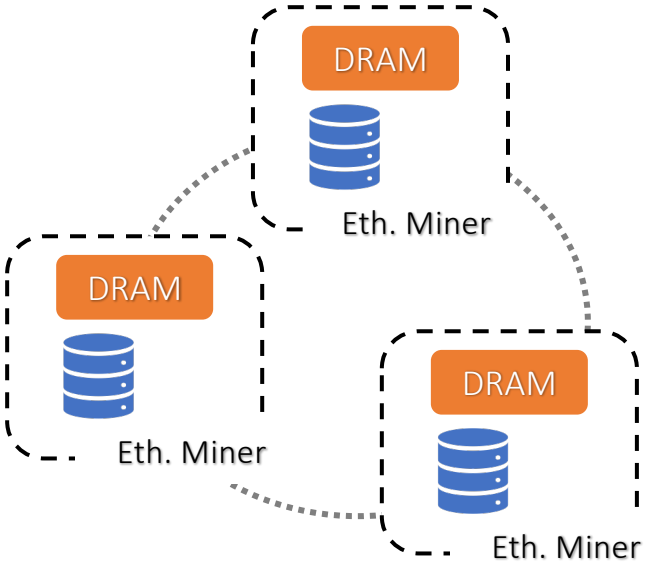
Ethereum 1-miner | Performance Breakdown

Evaluation



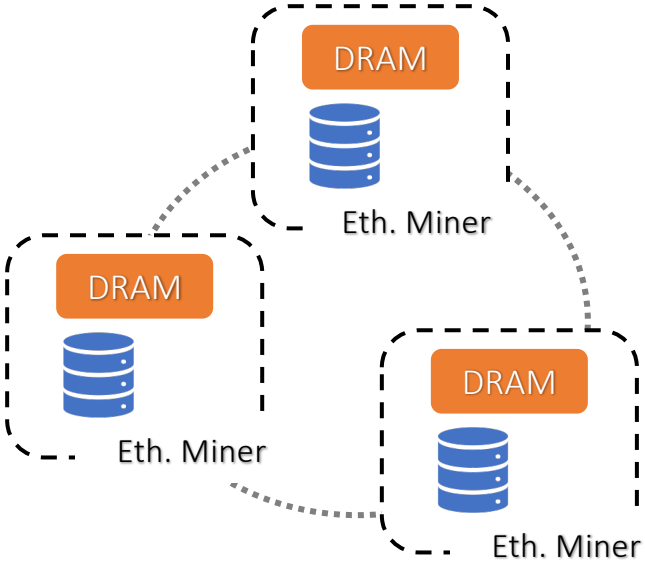
Ethereum 1-miner | Performance Breakdown

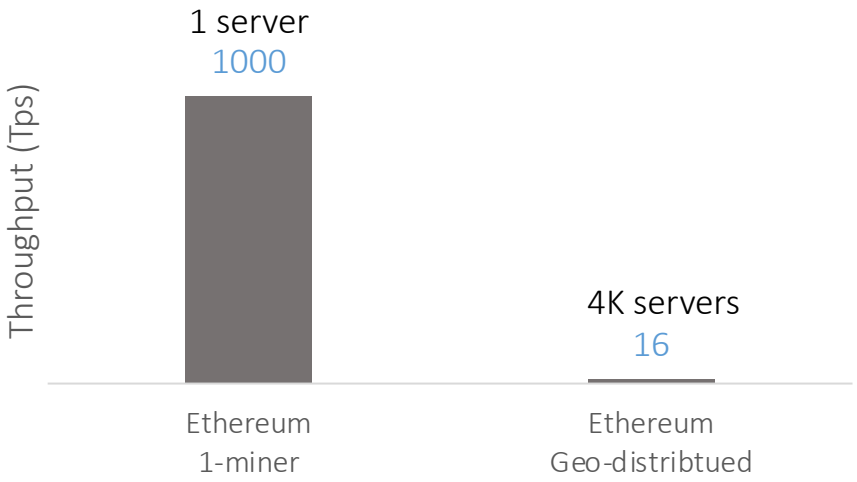
Evaluation



Geo-distributed Ethereum | Performance Breakdown

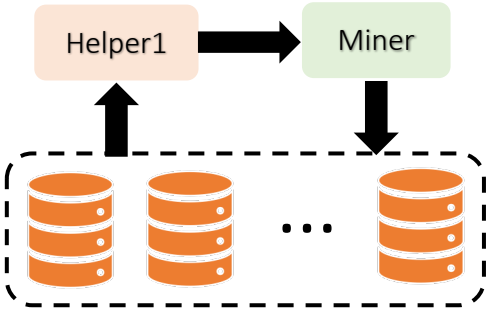
Evaluation





Geo-distributed Ethereum | Performance Breakdown

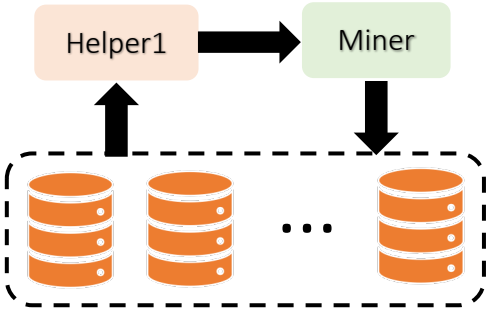
Evaluation

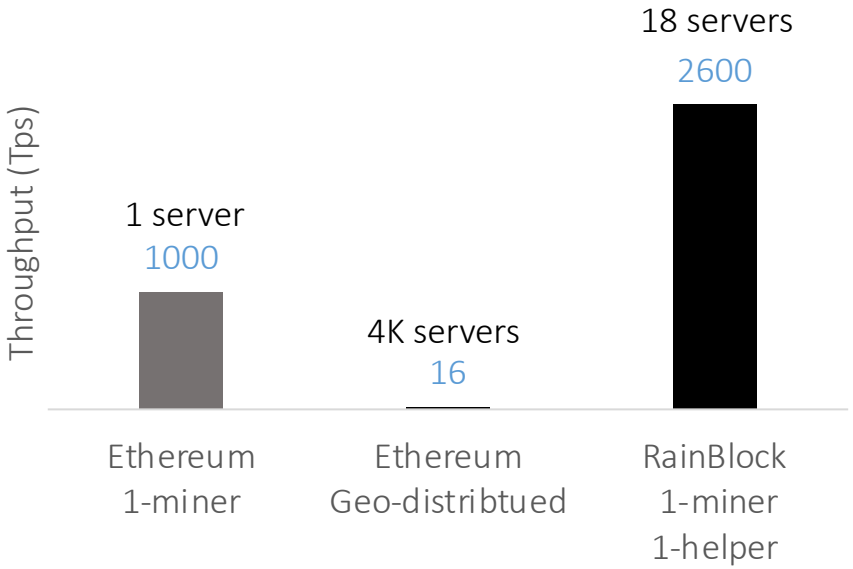


RainBlock 1-miner, 1-helper, 16-storage nodes

Performance Breakdown

Evaluation

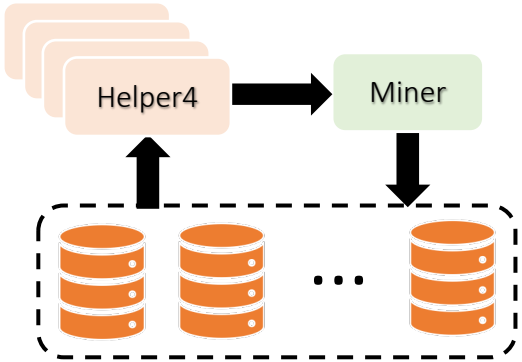




RainBlock 1-miner, 1-helper, 16-storage nodes

Performance Breakdown

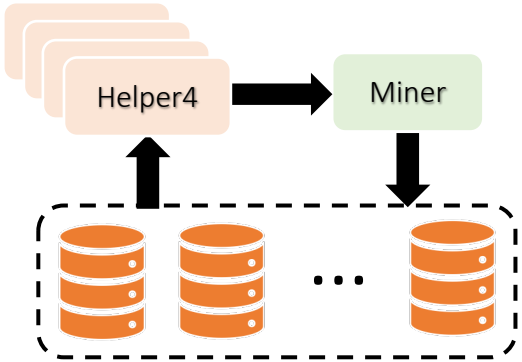
Evaluation

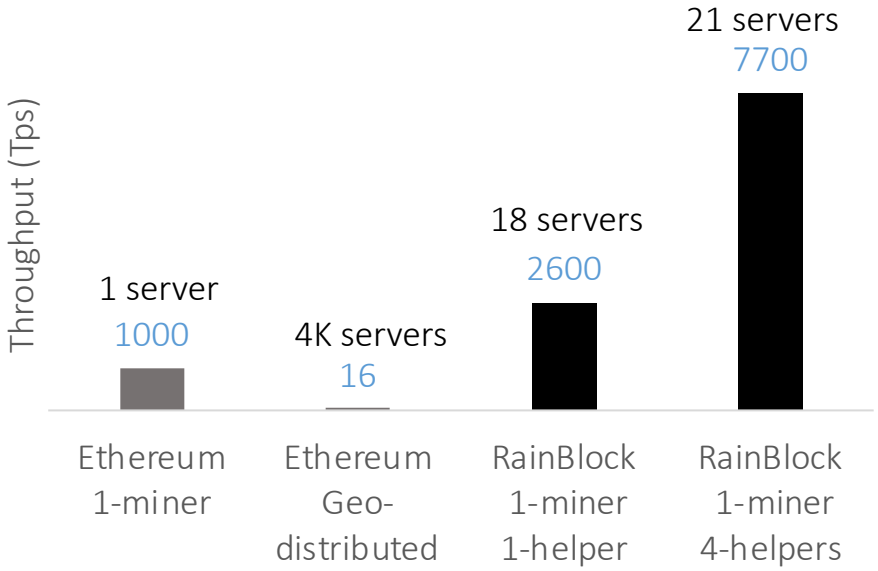


RainBlock 1-miner, 4-helpers, 16-storage nodes

Performance Breakdown

Evaluation

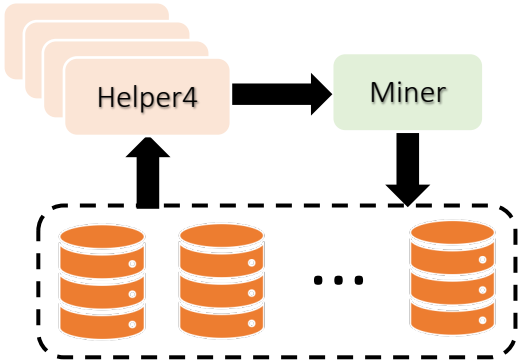




RainBlock 1-miner, 4-helpers, 16-storage nodes

Performance Breakdown

Evaluation

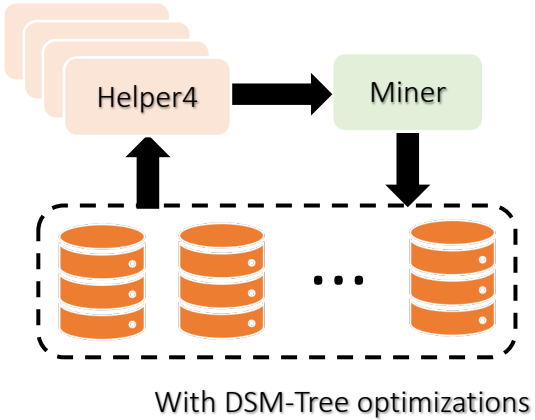


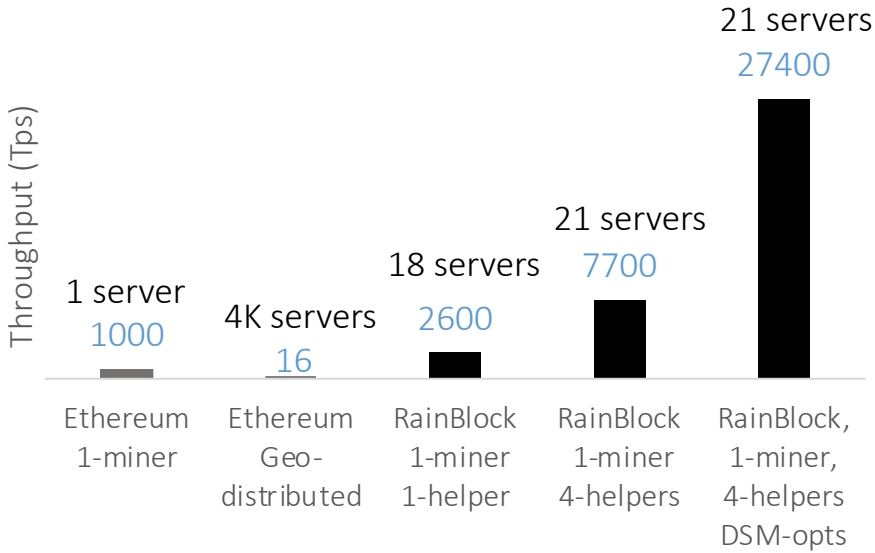
With DSM-Tree optimizations

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

Performance Breakdown

Evaluation

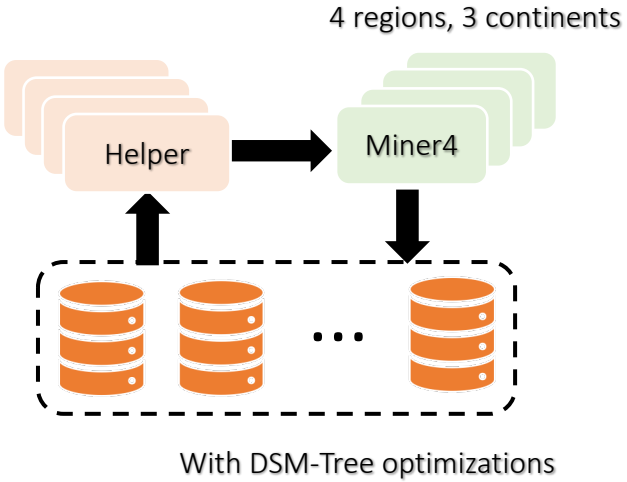


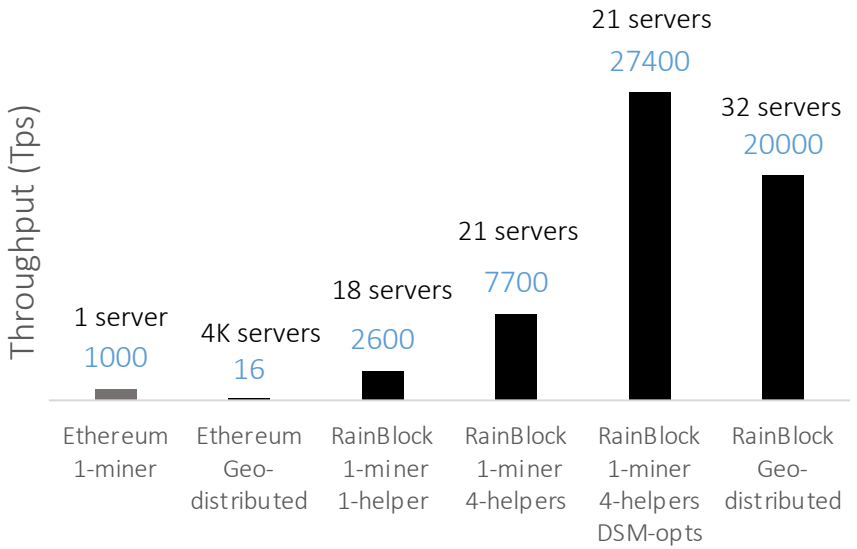


RainBlock 1-miner, 4-helpers, 16-storage nodes

Performance Breakdown

Evaluation





Geo-distributed RainBlock

Performance Breakdown

Summary

Summary

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

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

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

Summary

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

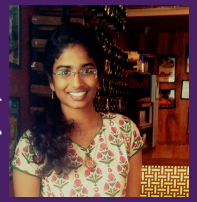
RainBlock avoids I/O in the critical path with I/O-Helpers and storage nodes

RainBlock uses DSM-Trees to reduce network traffic

RainBlock processes 20K tps in a geo-distributed setting

RainBlock: Faster Tx Processing in Public Blockchains

https://github.com/RainBlock


Soujanya


Aashaka

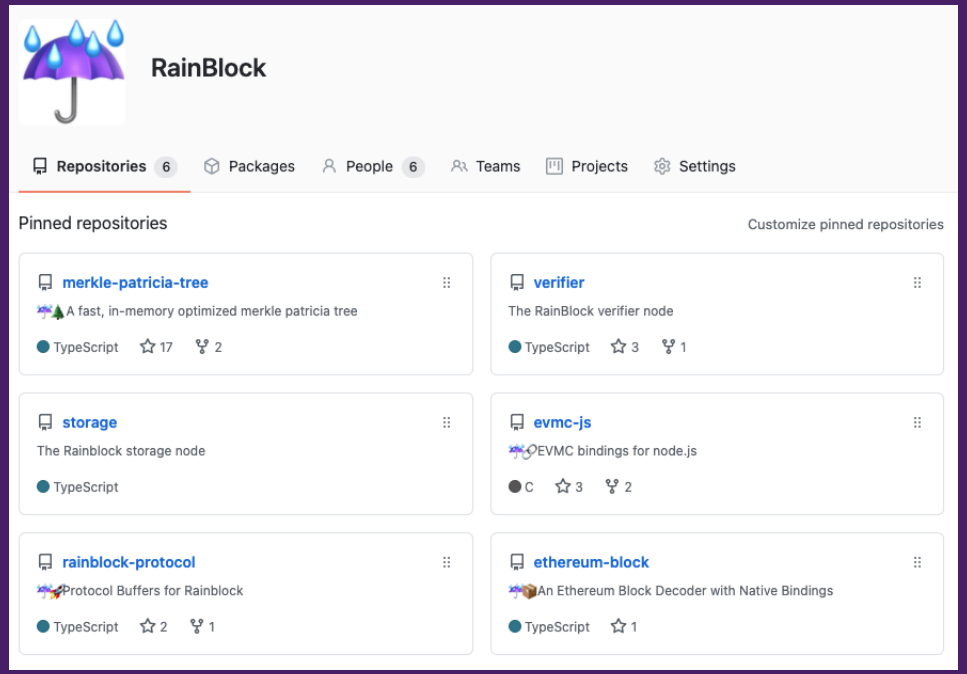

Souvik


Amy


Dahlia


Vijay


Michael



soujanya@cs.utexas.edu