About me

My journey

- Joined the Seoul Bitcoin Meetup in 2017 (meetup.com/seoulbitcoin)

About me

My journey

- Joined the Seoul Bitcoin Meetup in 2017 (meetup.com/seoulbitcoin)
- Started contributing to OSS in 2019

About me

My journey

- Joined the Seoul Bitcoin Meetup in 2017 (meetup.com/seoulbitcoin)
- Started contributing to OSS in 2019
- Started getting funded for my work in 2020

"The most important book about technology today, with implications that go far beyond programming."
—Guy Kawasaki

Revised & Expanded

THE CATHEDRAL & THE BAZAAR

MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY

ERIC S. RAYMOND

WITH A FOREWORD BY BOB YOUNG, CHAIRMAN & CEO OF RED HAT, INC.

- There isn't an official roadmap

- There isn't an official roadmap
- Everyone looks at scalability differently

- There isn't an official roadmap
- Everyone looks at scalability differently
- These are my thoughts

- Anthony Towns - www.erisian.com.au/wordpress/2023/06/21/putting-the-b-in-btc
- James O’Berine - jameso.be/2023/08/14/scaling-bitcoin.html

On on-chain scalability

Current bottlenecks and my proposed solutions

https://github.com/kcalvinalvin/2024-05-01-bobspaces

“We very, very much need such a system, but the way I understand your proposal, it does not seem to scale to the required size.”

https://www.mail-archive.com/cryptography@metzdowd.com/msg09963.html

What makes Bitcoin special

Something that cannot be negotiated

Trustlessness

What is it?

Only trusting the software I run

“Of Bitcoin’s many properties, trustlessness, or the ability to use Bitcoin without trusting anything but the open-source software you run, is, by far, king”

https://bluematt.bitcoin.ninja/2017/02/28/bitcoin-trustlessness/

“Of Bitcoin’s many properties, trustlessness, or the ability to use Bitcoin without trusting anything but the open-source software you run, is, by far, king”

https://bluematt.bitcoin.ninja/2017/02/28/bitcoin-trustlessness/

- Decentralization
- P2P
- Self custody

We've rejected anything that hurts trustlessness

How do you use Bitcoin trustlessly?

Running a full node

Achieving Trustlessness

- Full node = fully validating

Running a full node

Achieving Trustlessness

- Full node = fully validating
- Self-validating blocks/ txs

The Validation Process

What all full nodes go through

- Download ~600GB worth of blocks

The Validation Process

What all full nodes go through

- Download ~600GB worth of blocks
- Validate that the transaction is spending Bitcoins that exist

The Validation Process

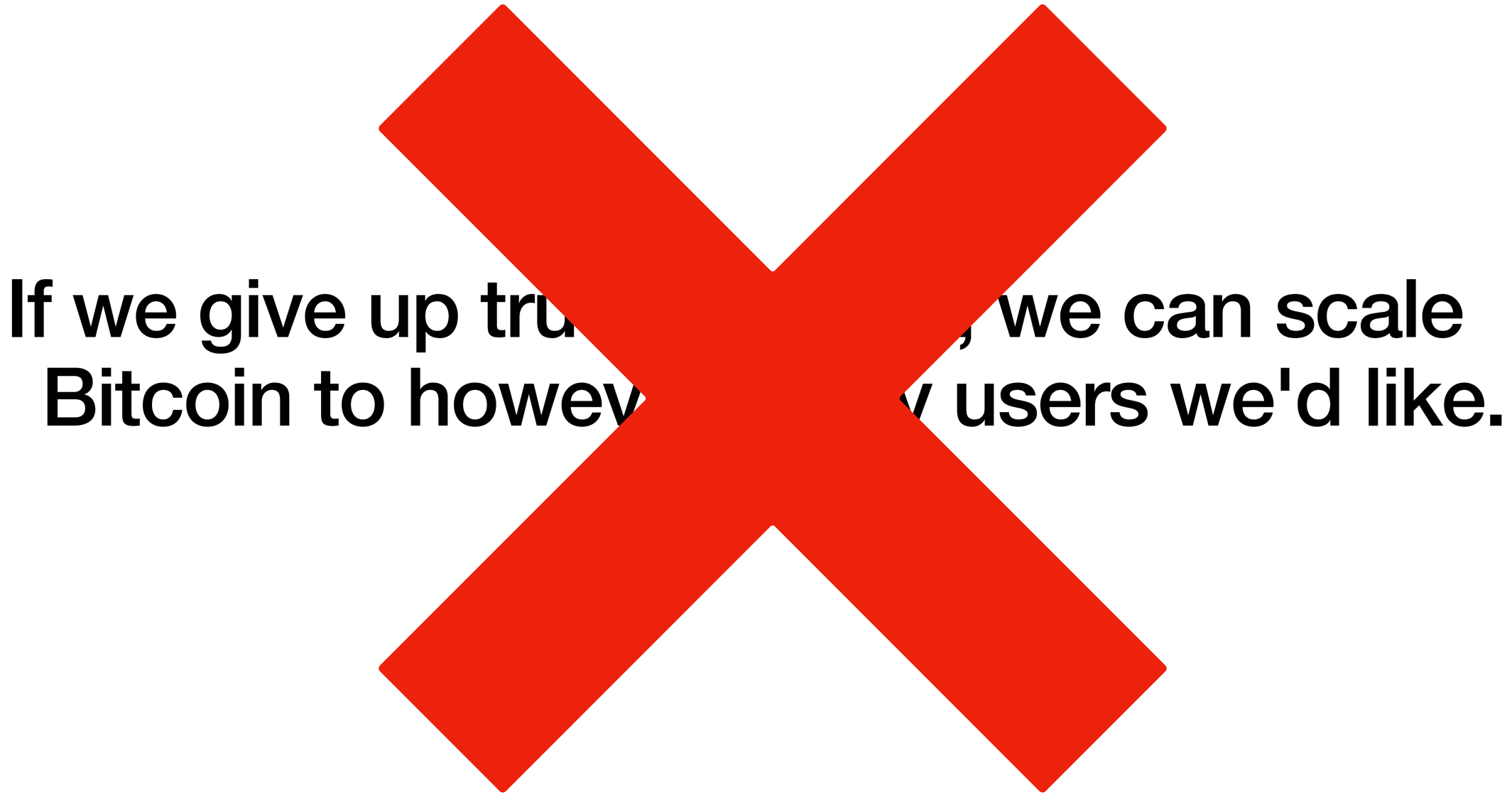
What all full nodes go through

- Download ~600GB worth of blocks
- Validate that the transaction is spending Bitcoins that exist
- Validate that the transaction is able to spend that Bitcoin

Things limiting scalability

Trustlessness!

If we give up trustlessness, we can scale Bitcoin to however many users we'd like.


If we give up tru , we can scale
Bitcoin to howev users we'd like.

Giving up on trustlessness

Letting everyone use Bitcoin but very few being able to validate

- Increase the block size to 1 gigabyte

Giving up on trustlessness

Letting everyone use Bitcoin but very few being able to validate

- Increase the block size to 1 gigabyte
- Everyone uses SPV and blindly trusts miners

Giving up on trustlessness

Letting everyone use Bitcoin but very few being able to validate

- Increase the block size to 1 gigabyte
- Everyone uses SPV and blindly trusts miners
- Miners do whatever they want

If we give up on the amount of users,
everyone can use Bitcoin trustlessly


If we give up on
t of users,
everyone can us
n trustlessly

Giving up on user count

Letting a few amount of people use Bitcoin trustlessly

- Limit the block size to 1KB

Giving up on user count

Letting a few amount of people use Bitcoin trustlessly

- Limit the block size to 1KB
- The entire chain is only 1GB up to block 1 million. Validation takes less than a minute on a normal phone.

Giving up on user count

Letting a few amount of people use Bitcoin trustlessly

- Limit the block size to 1KB
- The entire chain is only 1GB up to block 1 million. Validation takes less than a minute on a normal phone
- Fees are crazy. I can't afford them

There's a balance


SMALLER BLOCK BETTER


BIGGER BLOCK BETTER

What I strive for

The most amount of users while maintaining trustlessness

Methods for getting more bang for the buck

While maintaining trustlessness

- Use offchain protocols

Off chain protocols

What they're really all about

Achieve multiple transfers with a single on-chain transaction

Liquid Network

Offchain protocols

- Trading Bitcoin for IOUs on liquid sidechain

Liquid Network

Offchain protocols

- Trading Bitcoin for IOUs on liquid sidechain
- Trading off trustlessness for scalability

Liquid Network

Offchain protocols

- Trading Bitcoin for IOUs on liquid sidechain
- Trading off trustlessness for scalability
- Opt-in

Liquid Network

Offchain protocols

- Trading Bitcoin for IOUs on liquid sidechain
- Trading off trustlessness for scalability
- Opt-in
- liquid.net

Statechains

Offchain protocols

- Moving a whole UTXO to a statechain ran by an entity

Statechains

Offchain protocols

- Moving a whole UTXO to a statechain ran by an entity
- Need to put trust in the Statechain entity

Statechains

Offchain protocols

- Moving a whole UTXO to a statechain ran by an entity
- Need to put trust in the Statechain entity
- medium.com/@RubenSomsen/statechains-non-custodial-off-chain-bitcoin-transfer-1ae4845a4a39

Lightning Network

Offchain protocols

- Create channels between each other

Lightning Network

Offchain protocols

- Create channels between each other
- Transfer as long as the channel is balanced and is open

Lightning Network

Offchain protocols

- Create channels between each other
- Transfer as long as the channel is balanced and is open
- Security tradeoff. Passive -> active

Ultimately these protocols require an on-chain tx somewhere

Increase the block size

Needed to support more users.
Currently can't due to it hurting trustlesssness

Why can't we increase the block size?

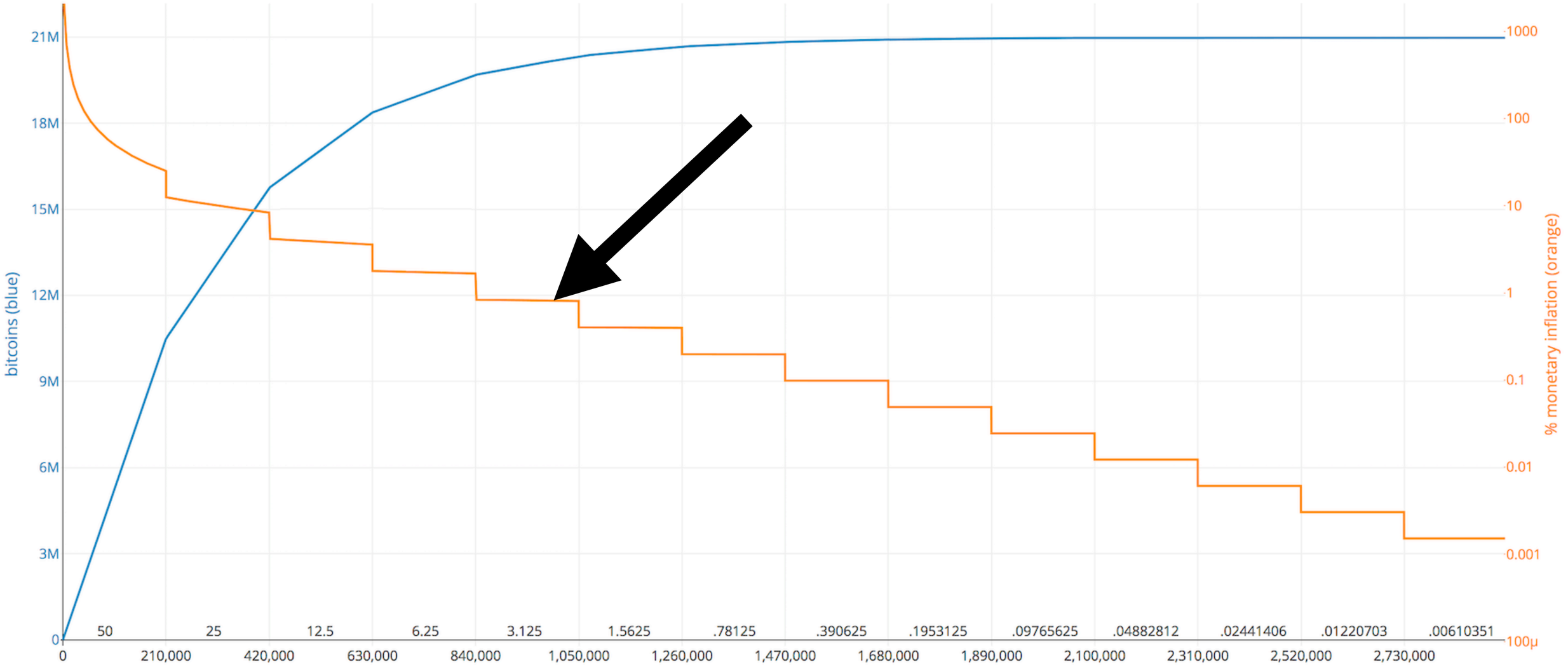
2 reasons

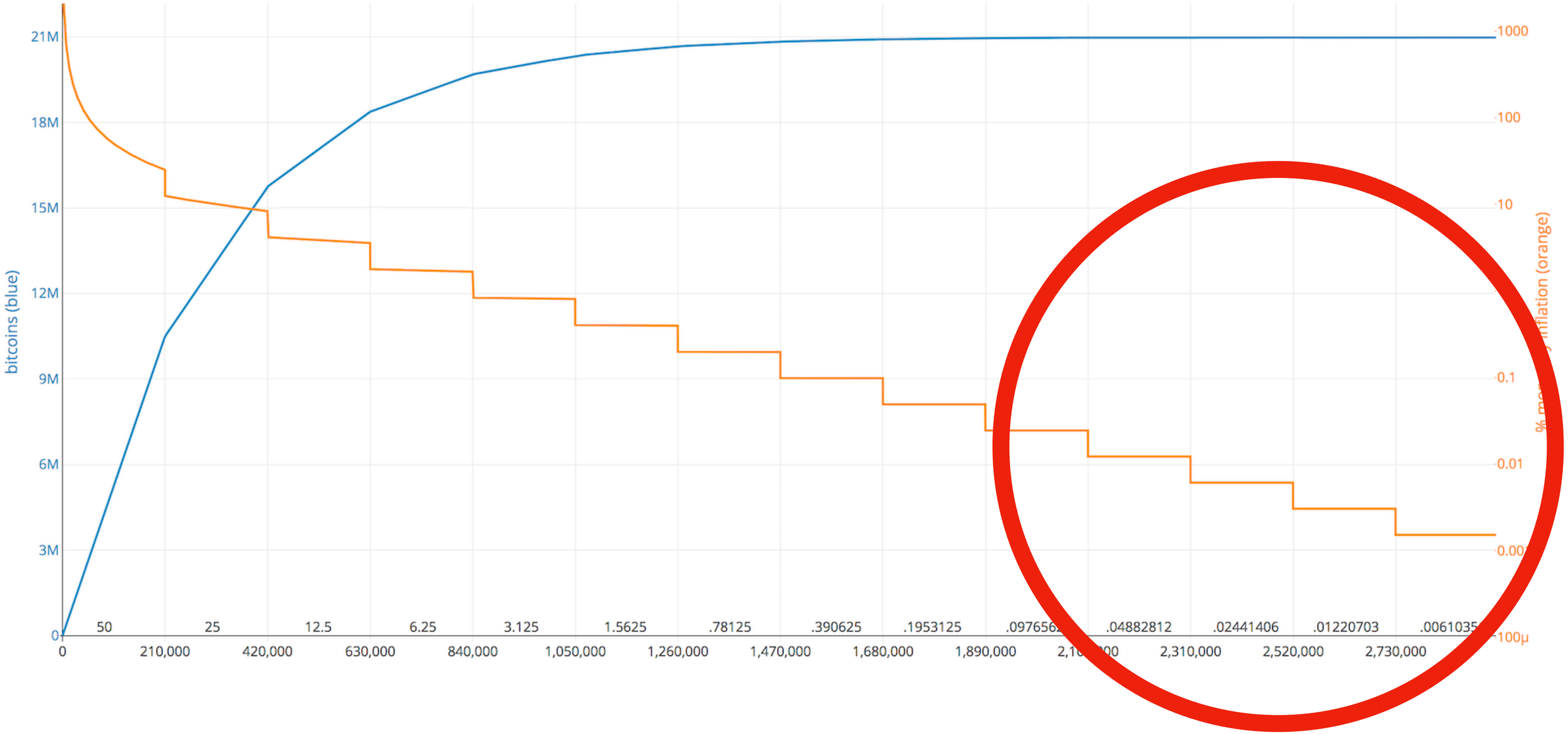
- Economical
- Technical

Economical reason

Supply and demand

- Are there gonna be enough fees when we increase the block size?


21M
18M
15M
12M
9M
6M
3M
0
bitcoins (blue)
1000
100
10
1
0.1
0.01
0.001
100μ
% monetary inflation (orange)
50
25
12.5
6.25
3.125
1.5625
.78125
.390625
.1953125
.09765625
.04882812
.02441406
.01220703
.00610351
0
210,000
420,000
630,000
840,000
1,050,000
1,260,000
1,470,000
1,680,000
1,890,000
2,100,000
2,310,000
2,520,000
2,730,000


21M
18M
15M
12M
9M
6M
3M
0
bitcoins (blue)
1000
100
10
0.1
0.01
100µ
% inflation (orange)
50
25
12.5
6.25
3.125
1.5625
.78125
.390625
.1953125
.04882812
.02441406
.01220703
0
210,000
420,000
630,000
840,000
1,050,000
1,260,000
1,470,000
1,680,000
1,890,000
2,310,000
2,520,000
2,730,000

Economical reason

Supply and demand

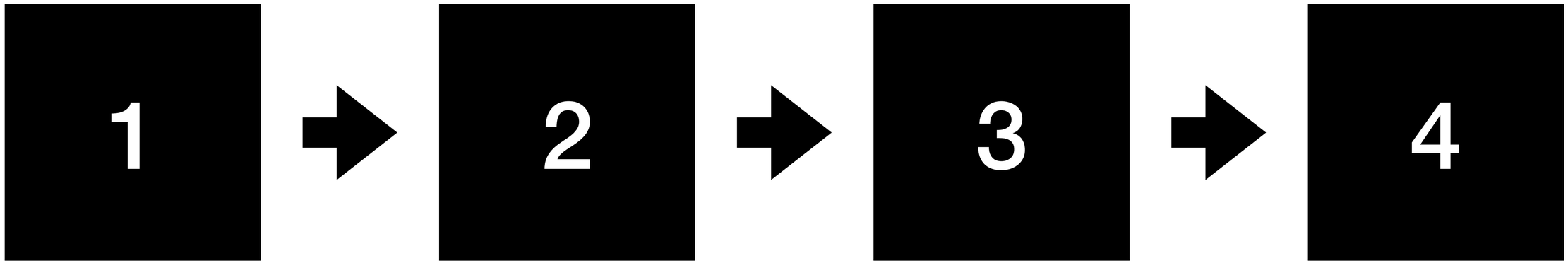
- Current *double spends preventions* are provided by the block subsidy

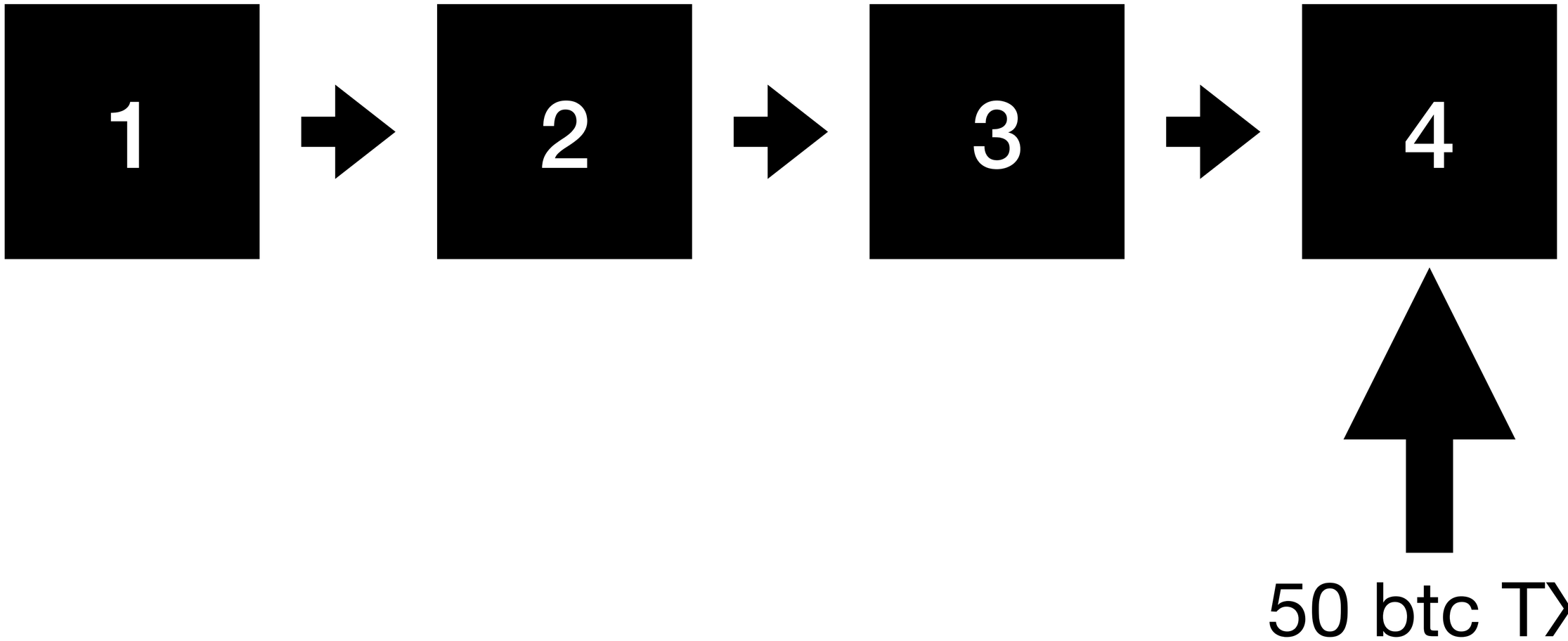
Economical reason

Supply and demand

- Current *double spends preventions* are provided by the block subsidy
- As the subsidy drops, the fees need to increase to keep the same amount of security

Example of a double spend due to less fees

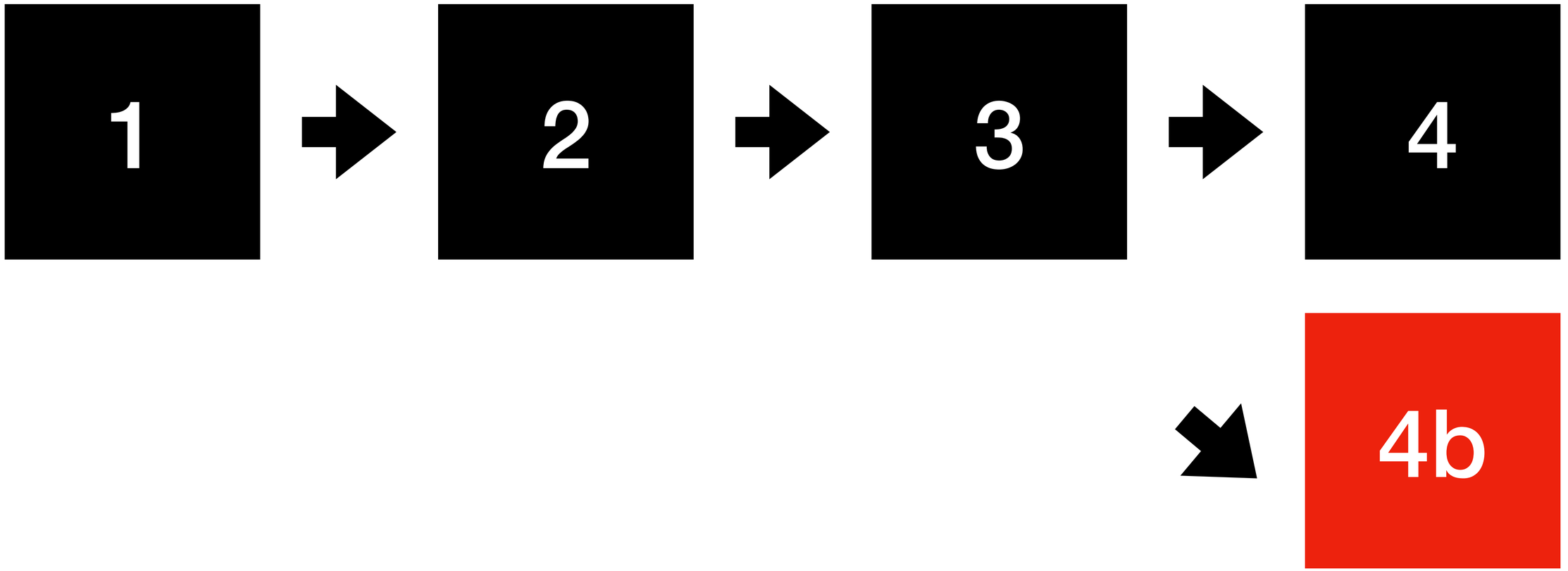


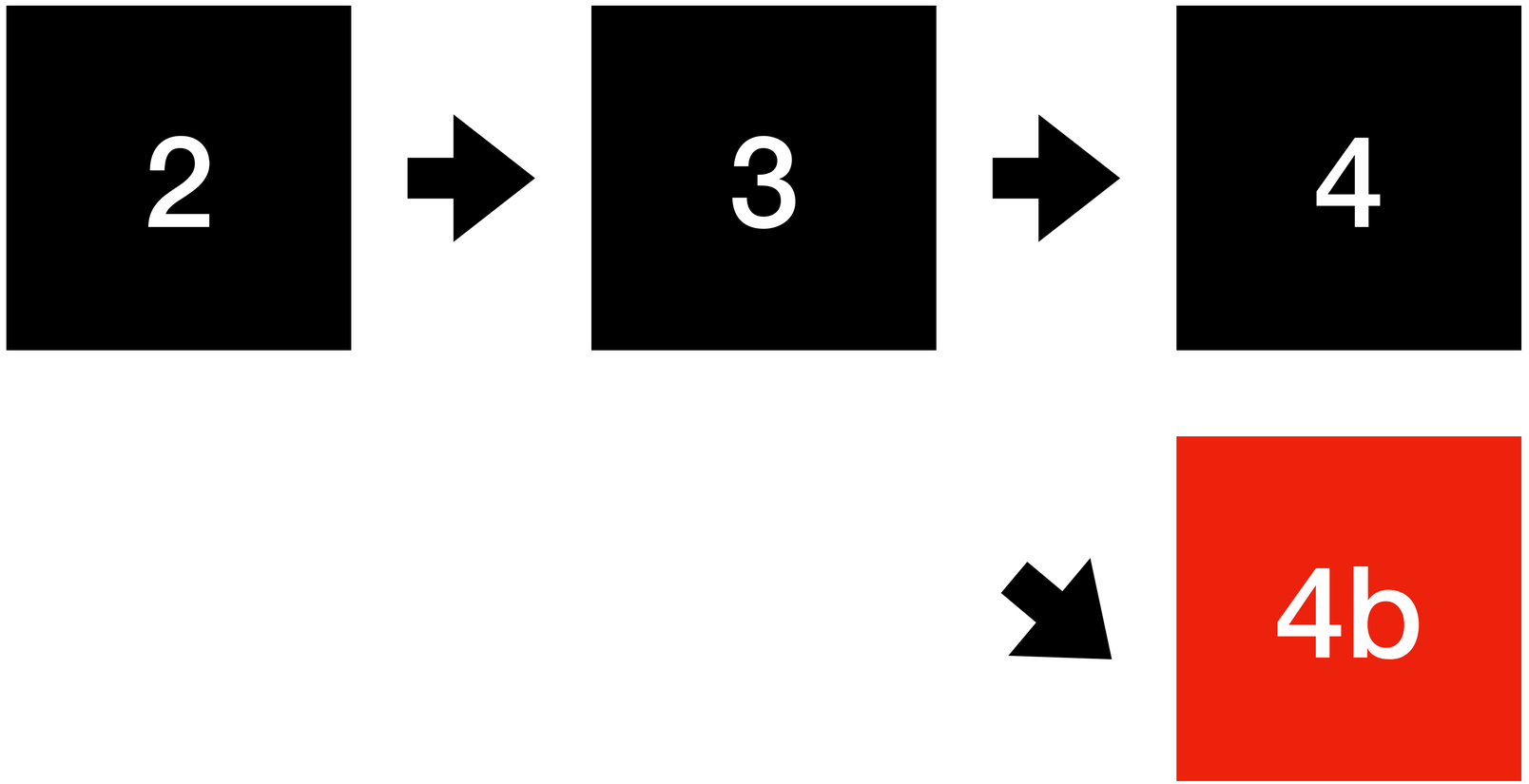

1
2
3
4
50 btc TX

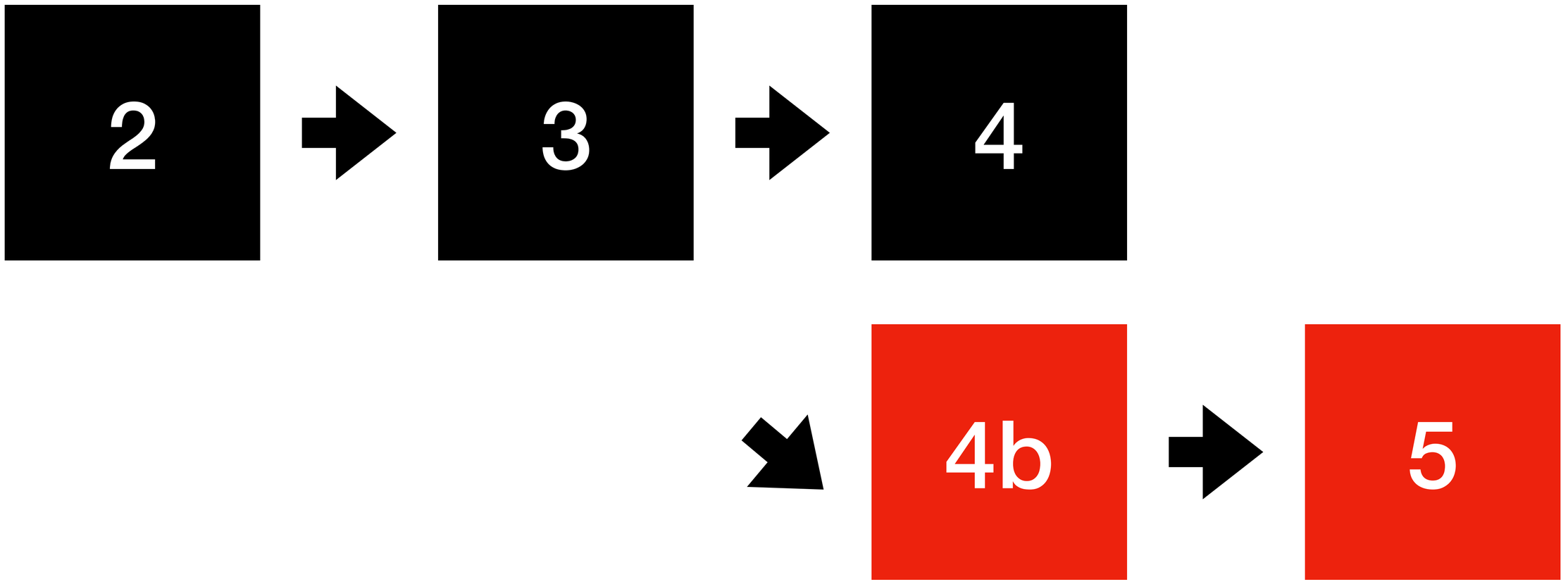
- Alice buys gold from Bob with 50btc

- Alice buys gold from Bob with 50btc
- Alice bribes miner Charlie 10btc to double spend Bob

- Alice buys gold from Bob with 50btc
- Bob bribes miner Charlie to double spend Alice
- Alice's profits - 40 btc + gold
- Miner Charlies profits - 10 btc + block subsidy


1
2
3
4
4b


2
3
4
4b


2
3
4
4b
5

Economical reason

Supply and demand

- Need enough fees to prevent these attacks

Economical reason

Supply and demand

- Need enough fees to prevent these attacks
- Block size ↑, Fees ↓

Currently no research ongoing for the economical reason

Technical reason

Trustlessness

- Can anyone that wants to use Bitcoin trustlessly do it?

Technical reason

Trustlessness

- Can anyone that wants to use Bitcoin trustlessly do it?
- Block size ↑, Cost of full nodes ↑

Technical reason

Trustlessness

- Download ~600GB worth of blocks
- Validate that the transaction is spending Bitcoins that exist
- Validate that the transaction is able to spend that Bitcoin

Technical reason

The bottlenecks

- Internet
- Disk i/o
- CPU

Internet

Downloading blocks/transactions

- Download ~600GB on first startup

Internet

Downloading blocks/transactions

- Download ~600GB on first startup
- 100mbps = ~800 min

600/(0.0125*60). 600GB to download. 0.0125GB a second. 60 seconds in a minute

Internet

Downloading blocks/transactions

- Download ~600GB on first startup
- 100mbps = ~800 min
- 1gbps = ~80 min

600/(0.125*60). 600GB to download. 0.125GB a second. 60 seconds in a minute

Internet

Downloading blocks/transactions

- Download ~600GB on first startup
- 100mbps = ~800 min
- 1gbps = ~80 min
- 10gbps = ~8 min

600/(1.25*60). 600GB to download. 1.25GB a second. 60 seconds in a minute

Internet bandwidth is ok

Disk performance

Read and write to the UTXO set

- UTXO set - Collection of all the Bitcoins available to be spent

What's a UTXO set?

- TXO - Transaction Output

Typical transaction

https://blockstream.info/tx/8ad63210d059908229f00ad177469d9f16a5d37a7ea64cede9c92684ff7cf06e

The outputs

https://blockstream.info/tx/8ad63210d059908229f00ad177469d9f16a5d37a7ea64cede9c92684ff7cf06e

What's a UTXO set?

- TXO - Transaction Output
- UTXO - Unspent TXO

What's a UTXO set?

- TXO - Transaction Output
- UTXO - Unspent TXO
- UTXO set - All the UTXOs

Disk performance

Read and write to the UTXO set

- UTXO set uses a key-value database (leveldb)
- leveldb does random read and writes

Disk performance

Read and write to the UTXO set

- At least 59 bytes per UTXO
- Average tx is 1 input, 2 output
- For 2000txs: ~2000 read, ~4000 write. (2000-3000 txs per block)

Outpoint: 33B, p2pkh script: 21B, height+iscoinbase: 4, amount: 1

기술의 경이로움이 시작되다

Platinum P41 SSD

SK하이닉스 Platinum P41 SSD는 업계 최고 수준의 성능으로
차세대 컴퓨팅 시스템을 완성하는 제품입니다. PCIe 4세대 NVMe 인터페이스와
176단 NAND의 한층 더 강화된 기술력을 토대로 최강의 데이터 전송 속도와
성능을 제공합니다. 이제 차원이 다른 성능을 경험해 보세요.

성능

	2TB	1TB	500GB
순차성능 (최대)	읽기 : 7,000 MB/s 쓰기 : 6,500 MB/s	읽기 : 7,000 MB/s 쓰기 : 6,500 MB/s	읽기 : 7,000 MB/s 쓰기 : 4,700 MB/s
랜덤성능 (최대)	읽기 : 1,400K IOPS 쓰기 : 1,300K IOPS	읽기 : 1,400K IOPS 쓰기 : 1,300K IOPS	읽기 : 960K IOPS 쓰기 : 1,000K IOPS
TBW	최대 1,200TBW	최대 750TBW	최대 500TBW

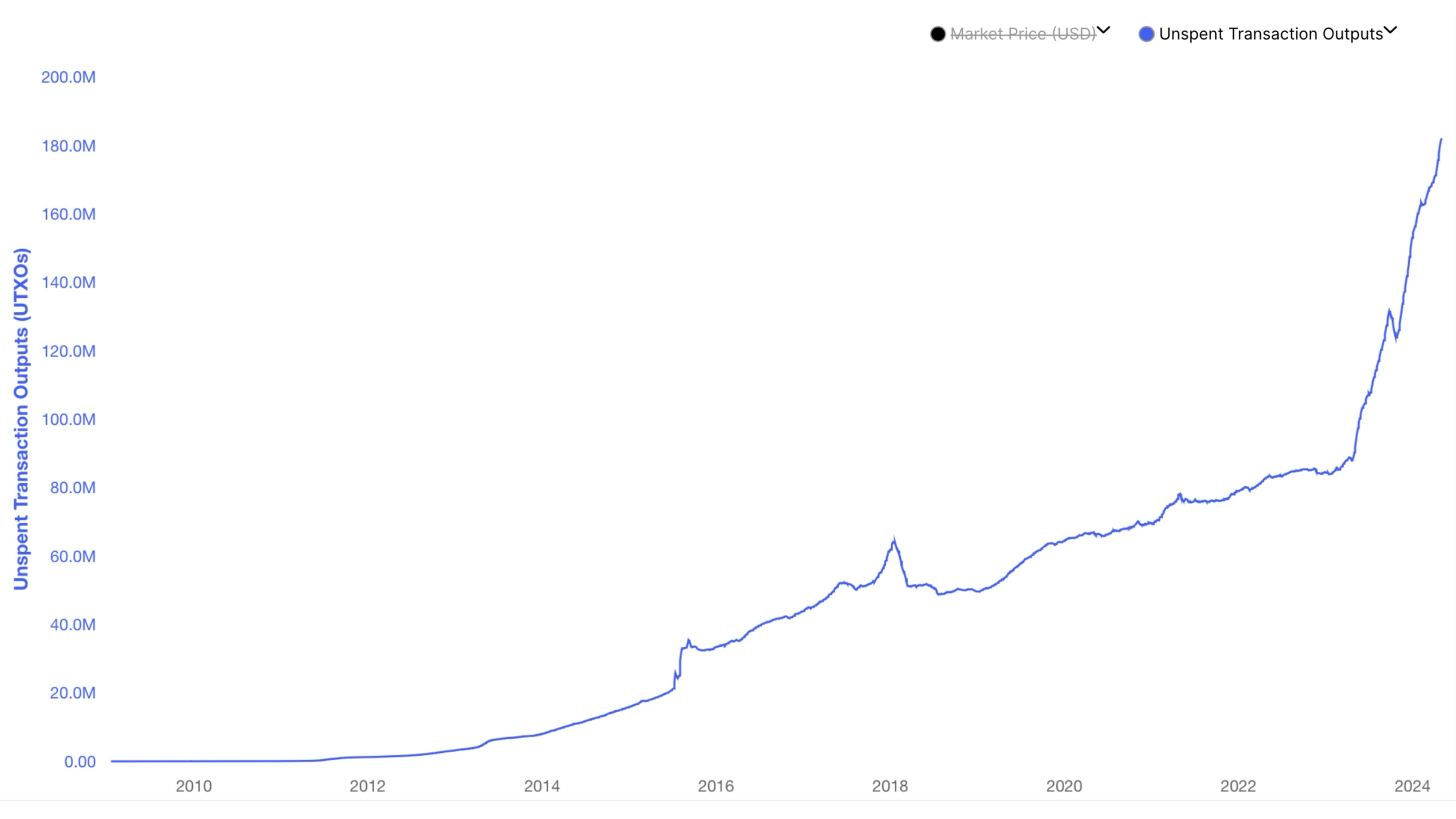
- 1,400 IOPS= ~5.6MB/s read
- 1,300 IOPS= ~5.2MB/s write
- Per block 0.12MB read, 0.24MB write

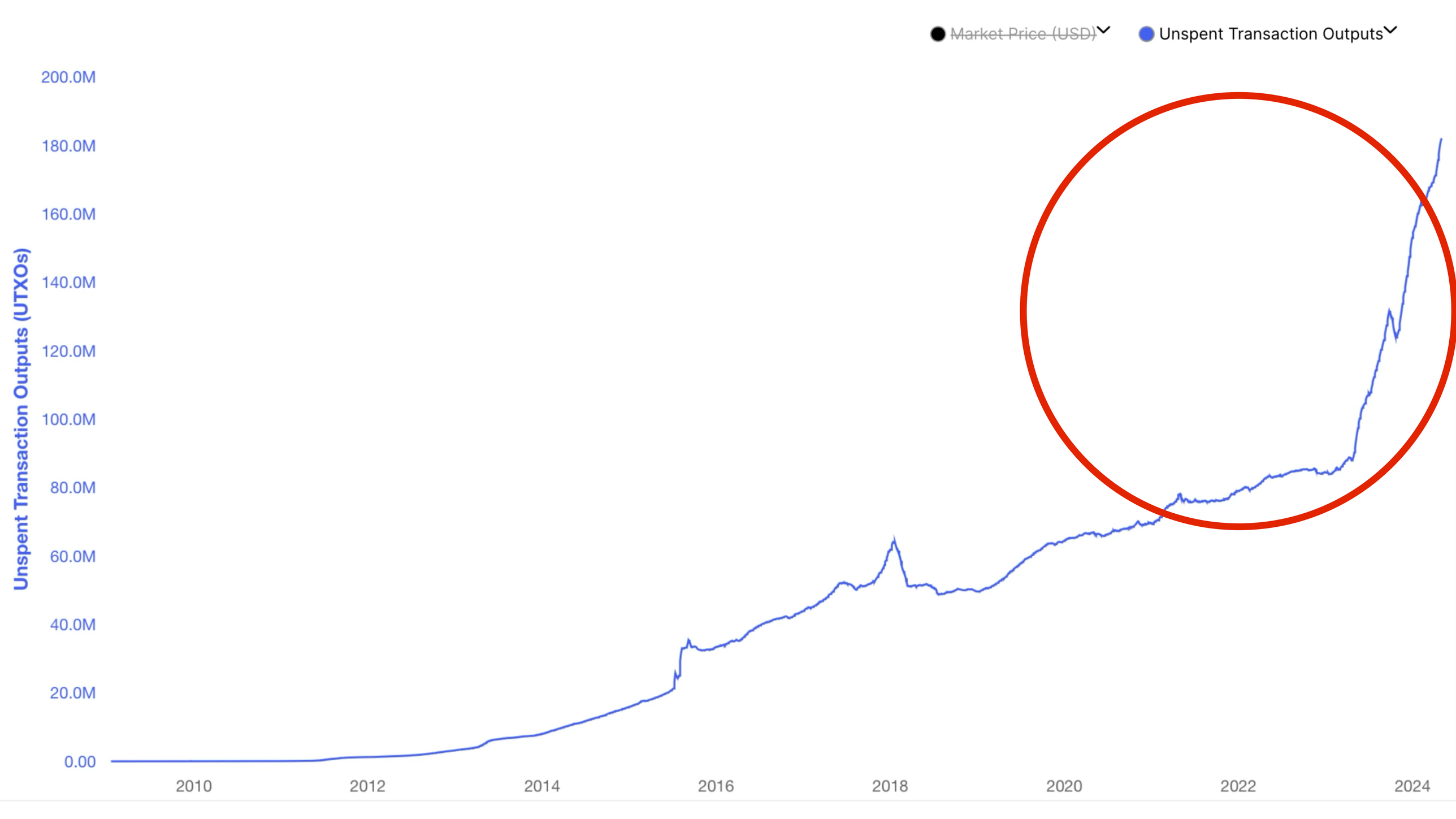
(1,400 * 4) / 1000. OS page size of 4(mb). Divide by 1000 to get MB/s instead of byte/s


WD_BLACK™
GAMING HARD DRIVE
www.wdblack.com
8TB
S/N:ACO8JNAL
MDL:WD8002FZWX-00BKUA0
WD8002FZWX
SATA 6Gb/s PC HA710
Made in Thailand
5VDC: 0.42A 12VDC: 0.60A
14 MAR 2023
R/N: 810032
WD, PO Box 13379, Swords, Co Dublin, Ireland
WD UK, PO Box 471, Leatherhead, KT22 2LU, UK
WWN:50014EE2T6A60092
CAN ICES-3 (B)/NMB-3(B)
R-R-WDT-810032
Western Digital®
DO NOT COVER
ANY DRIVE HOLES

- 73 IOPS= ~0.29MB/s read&write
- Per block: 0.12MB read, 0.24MB write

https://en.wikipedia.org/wiki/IOPS


Market Price (USD)
Unspent Transaction Outputs
Unspent Transaction Outputs (UTXOs)
200.0M
180.0M
160.0M
140.0M
120.0M
100.0M
80.0M
60.0M
40.0M
20.0M
0.00
2010
2012
2014
2016
2018
2020
2022
2024


Market Price (USD)
Unspent Transaction Outputs
Unspent Transaction Outputs (UTXOs)
200.0M
180.0M
160.0M
140.0M
120.0M
100.0M
80.0M
60.0M
40.0M
20.0M
0.00
2010
2012
2014
2016
2018
2020
2022
2024

As the UTXO set grows, leveldb performance drops

UTXO set size is unlimited

8billion UTXO, 59 bytes each

472GB

Outpoint: 33B, p2pkh script: 21B, height+iscoinbase: 4, amount: 1

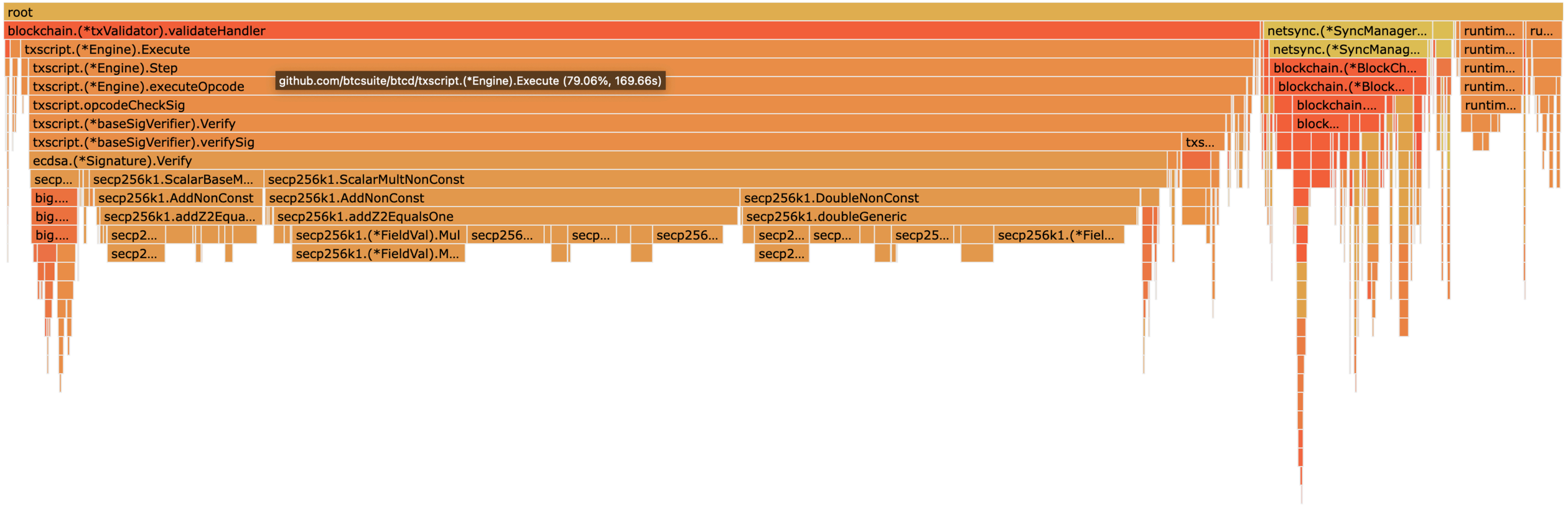
HDDs are already bottlenecked

CPU Performance

Transaction signature validation

- Currently most of the block validation is spent on signature validation

github.com/btcsuite/btcd/txscript.(*Engine).Execute (79.06%, 169.66s)



80% of the time spent on tx validation

Based on btcd. Bitcoin Core does much better.

CPU Performance

Transaction signature validation

- CPU cores are still increasing
- Taproot txs can be validated together

README.md

libsecp256k1

Optimized C library for ECDSA signatures and secret/public key operations on curve secp256k1.

This library is intended to be the highest quality publicly available library for cryptography on the secp256k1 curve. However, the primary focus of its development has been for usage in the Bitcoin system and usage unlike Bitcoin's may be less well tested, verified, or suffer from a less well thought out interface. Correct usage requires some care and consideration that the library is fit for your application's purpose.

Features:

- secp256k1 ECDSA signing/verification and key generation.
- Additive and multiplicative tweaking of secret/public keys.
- Serialization/parsing of secret keys, public keys, signatures.
- Constant time, constant memory access signing and public key generation.
- Derandomized ECDSA (via RFC6979 or with a caller provided function.)
- Very efficient implementation.
- Suitable for embedded systems.
- No runtime dependencies.
- Optional module for public key recovery.
- Optional module for ECDH key exchange.
- Optional module for Schnorr signatures according to BIP-340.

CPU already bottlenecked
Improvements possible

What do we need to do to increase the block size?

Signature validation speed ↑
UTXO read/write speed ↑

How do we improve the UTXO read/write speed?

Utreexo

What I work on

Let's turn the UTXO set into a merkle tree

Same performance on HDD and SSD

Disk i/o X

Utreexo eliminates random reads/writes

Benchmarking this

Testing methodology

- Compare the performance drop on a pruned node vs a non-pruned node

36% slower on testnet

Bitcoin Core

https://blog.bitmex.com/progress-towards-utreexo-goals/

3% slower on testnet

Utreexo node

https://blog.bitmex.com/progress-towards-utreexo-goals/

What we write with Utreexo

No reads

```
	{443000, newHashFromStr("0000000000000000047efc3c126c3398cfdef1609682d1492cd2909e3c0a17f"),
		43243051, []*chainhash.Hash{
			newLeafHashFromStr("df31c496a54a8021c38c77809da15a8bd3968970315f5a56c07f0fd9763262ff"),
			newLeafHashFromStr("755f4cfbf5727c3e14aaf6dcc6e5ab150c61101cead06fade5a886ee8d1dd377"),
			newLeafHashFromStr("c572a92c7cd9e14ce5d7c94473ef90bee21856654eed5535314b3c117006bc83"),
			newLeafHashFromStr("9e85567347ef0ea38c1797d8ed8e399ed0d99a07025388f6e06791ff6b88643c"),
			newLeafHashFromStr("ab4efc738bd7b8281689e899979d44ed0ca8fe1489efa0f5a09f5b1c276fe792"),
			newLeafHashFromStr("1c763d8f490baef0b756597dfc2da5a31d0aee324da6314b52f21707351cf51a"),
			newLeafHashFromStr("29fcd9989939a17e5d113fb1f247f3bd05606488bcb48c9cfad55a57006f2248"),
			newLeafHashFromStr("e79f840fe1bcdbb6b638a53b637885dda7ea3da2b2b5f1f5447ebf4d54fc0a2e"),
			newLeafHashFromStr("02f33c3ce0d684e644410fa35fee4c7572454facc710389ea0f69a00e27c1780"),
			newLeafHashFromStr("4e29ad940088ab42991bd99d7943a68ea38d498f8435c4bea21b8252788b5a7d"),
			newLeafHashFromStr("feabfc6565e918827df8ade12d0885e7da28bb81410c71a0fcbd92da28e5538e"),
			newLeafHashFromStr("42275481edcd150c0891f8f7486941e7ecc9dbec0d1554d1ae3027d8523571b4"),
			newLeafHashFromStr("84dd84bbb564ab9e878ea774d66a2cb818c2b74680974e01a07dde68bd064bfd"),
			newLeafHashFromStr("19a64f2f77403d4a1e3482b3a1801aaf360120f6429f5ed47261fc8676920d9d"),
		},
	},
```

https://github.com/mit-dci/utcd/blob/7310b99094acbeb11ecda230475d127af064c82e/chaincfg/mainnetroots.go#L6665-L6682

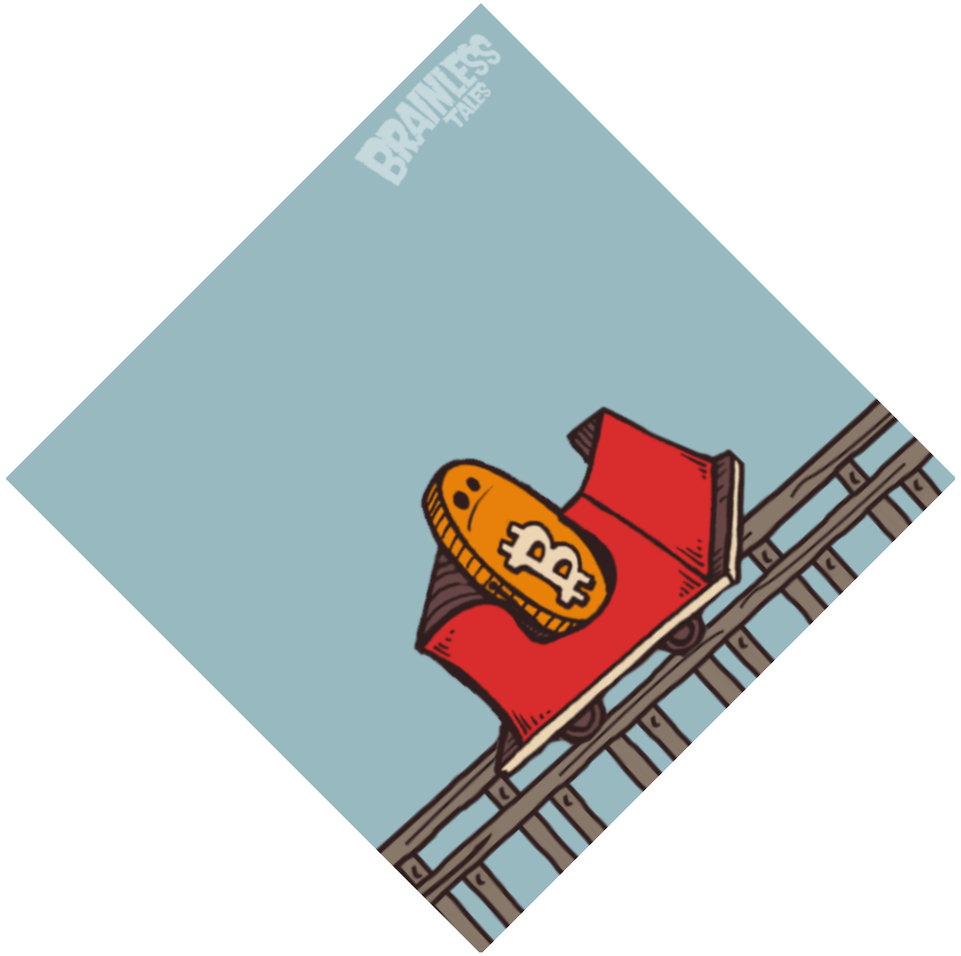
Tradeoffs?

The internet usage increases

Utreexo nodes need to download extra data

- 500GB of proof data needs to be downloaded (as of height 841,642). Block data is 613GB
- Currently working on caching to reduce this

Bandwidth
CPU



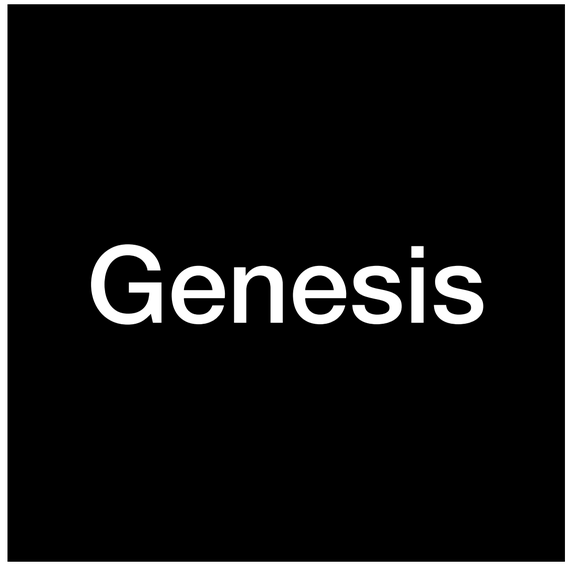
disk i/o
RAM usage

Utilizing Utreexo for faster block verification

Parallel block validation

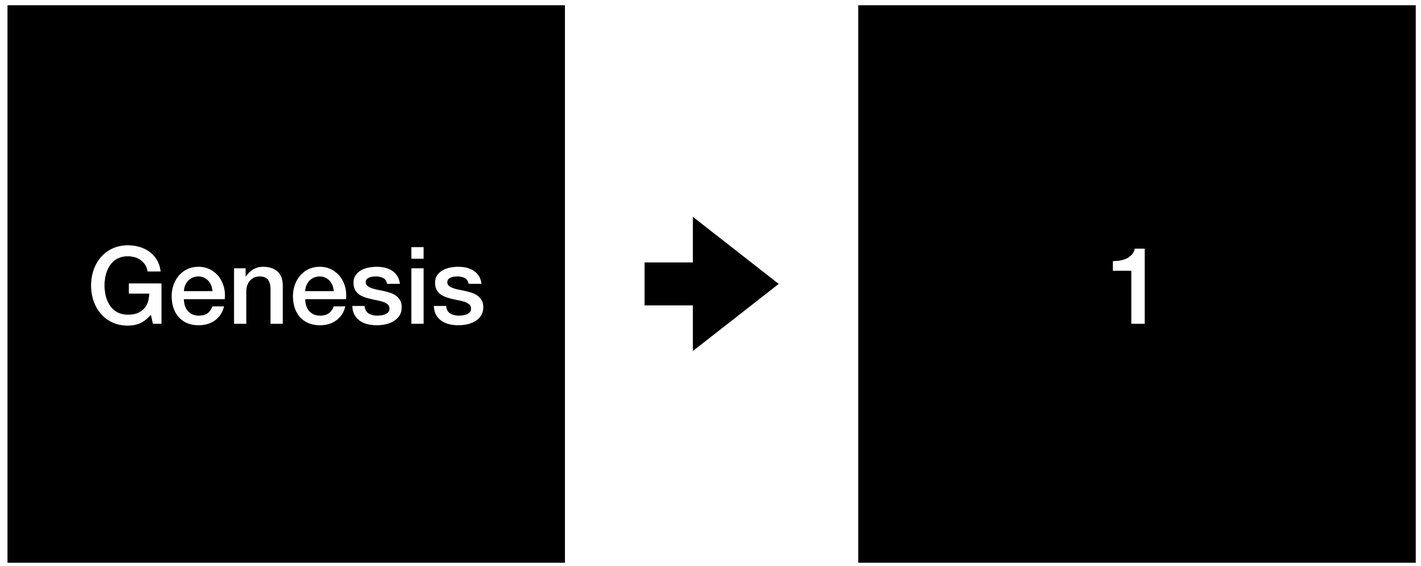
Current block validation

Sequential validation



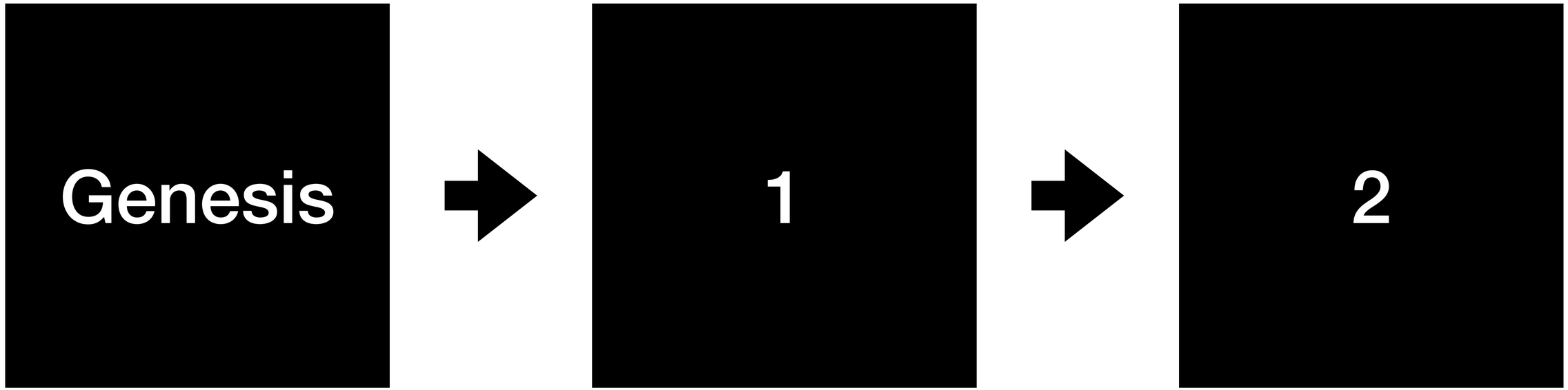
Current block validation

Sequential validation



Current block validation

Sequential validation



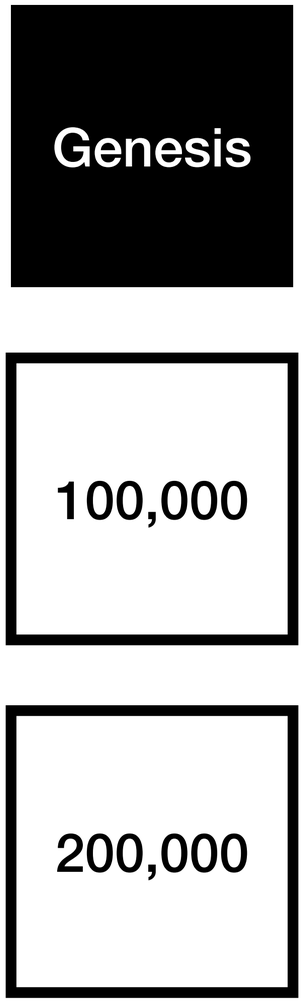
Parallel block validation

Replay blocks in any order

- Utreexo roots committed into the binary
- Process blocks starting from any of the roots that are committed
- https://github.com/mit-dci/utcd/blob/master/chaincfg/mainnetroots.go

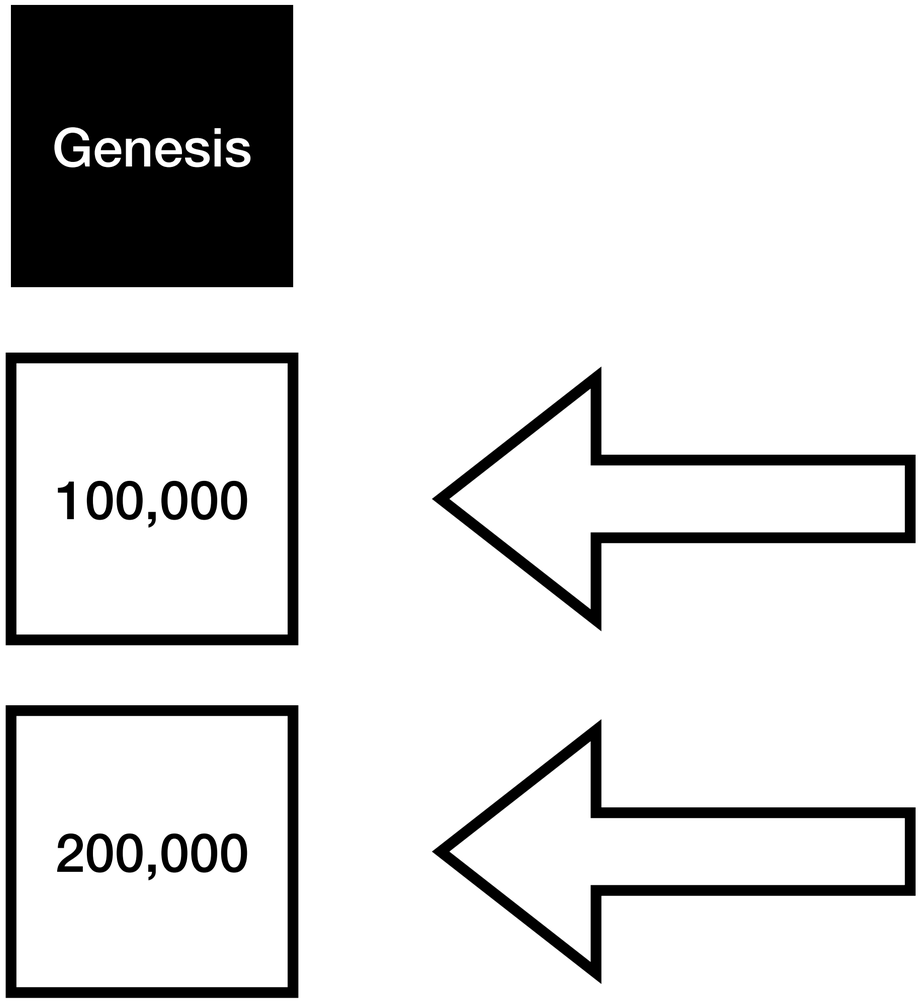
With Utreexo

Efficient parallel validation



With Utreexo

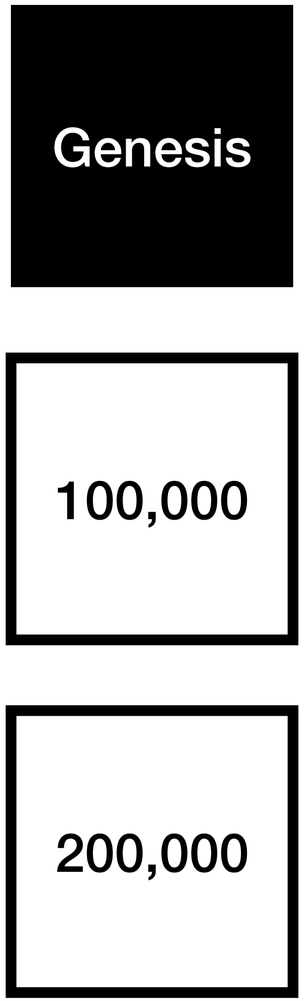
Efficient parallel validation



Untrusted roots at the start

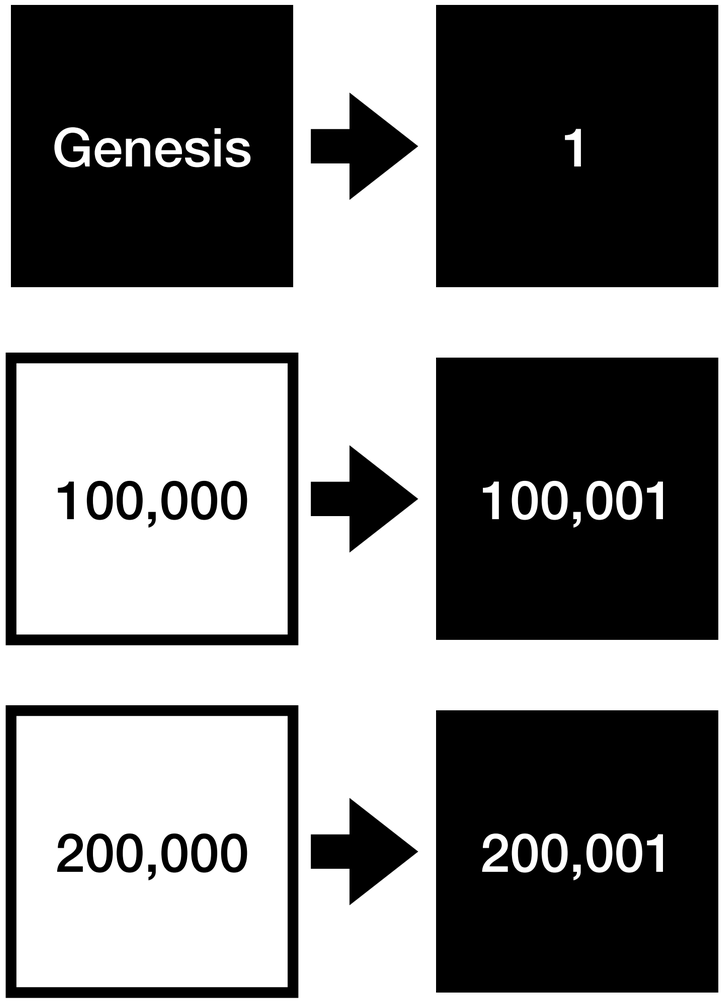
With Utreexo

Efficient parallel validation



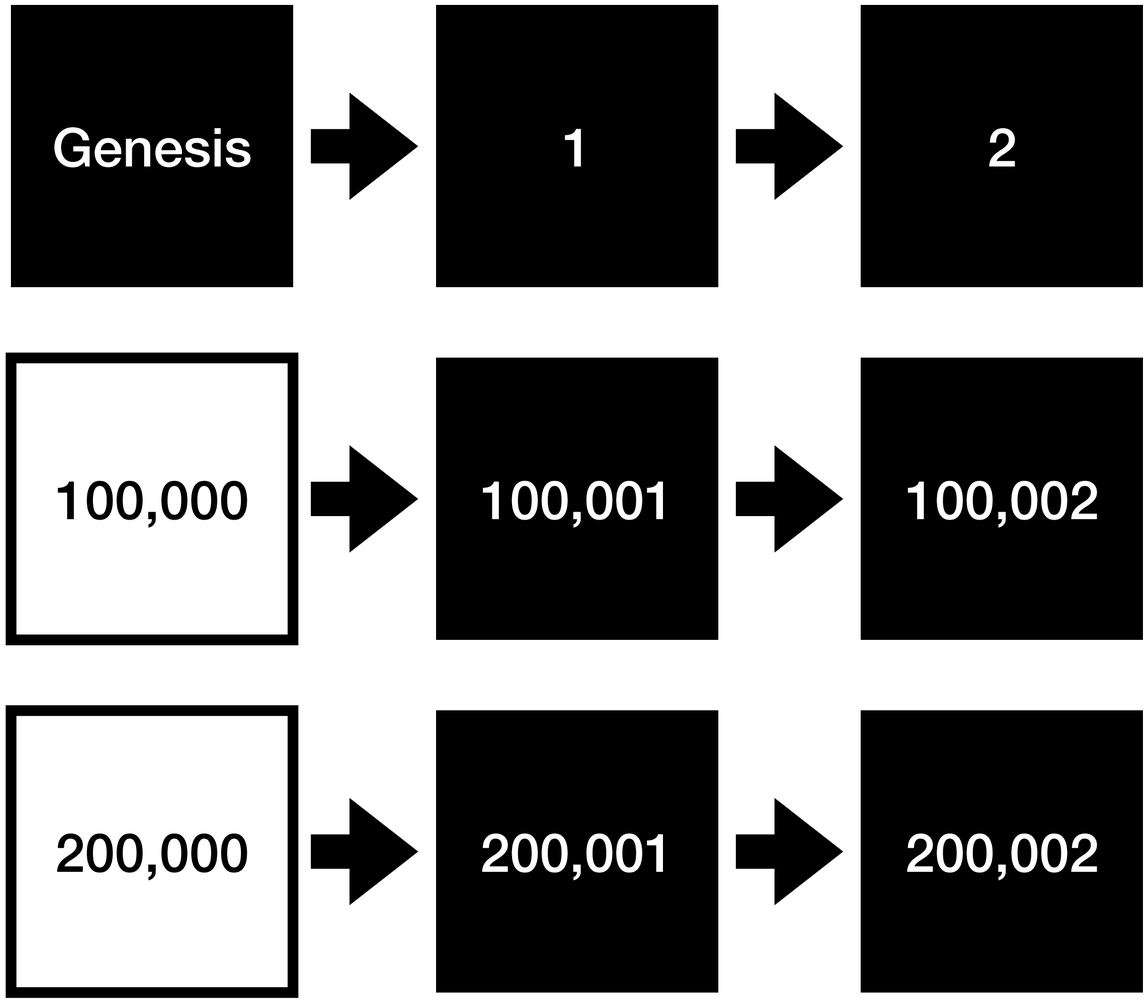
With Utreexo

Efficient parallel validation



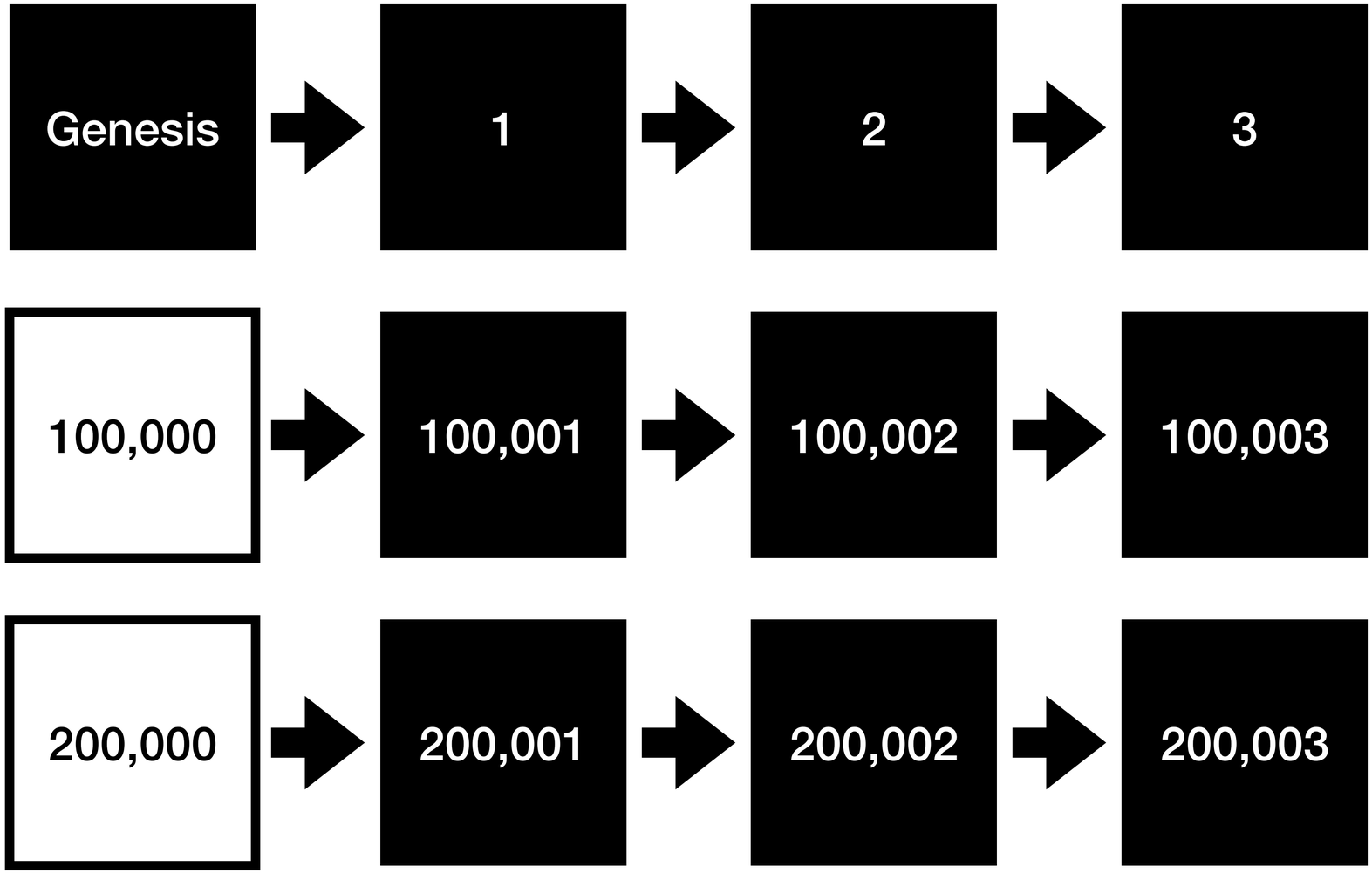
With Utreexo

Efficient parallel validation



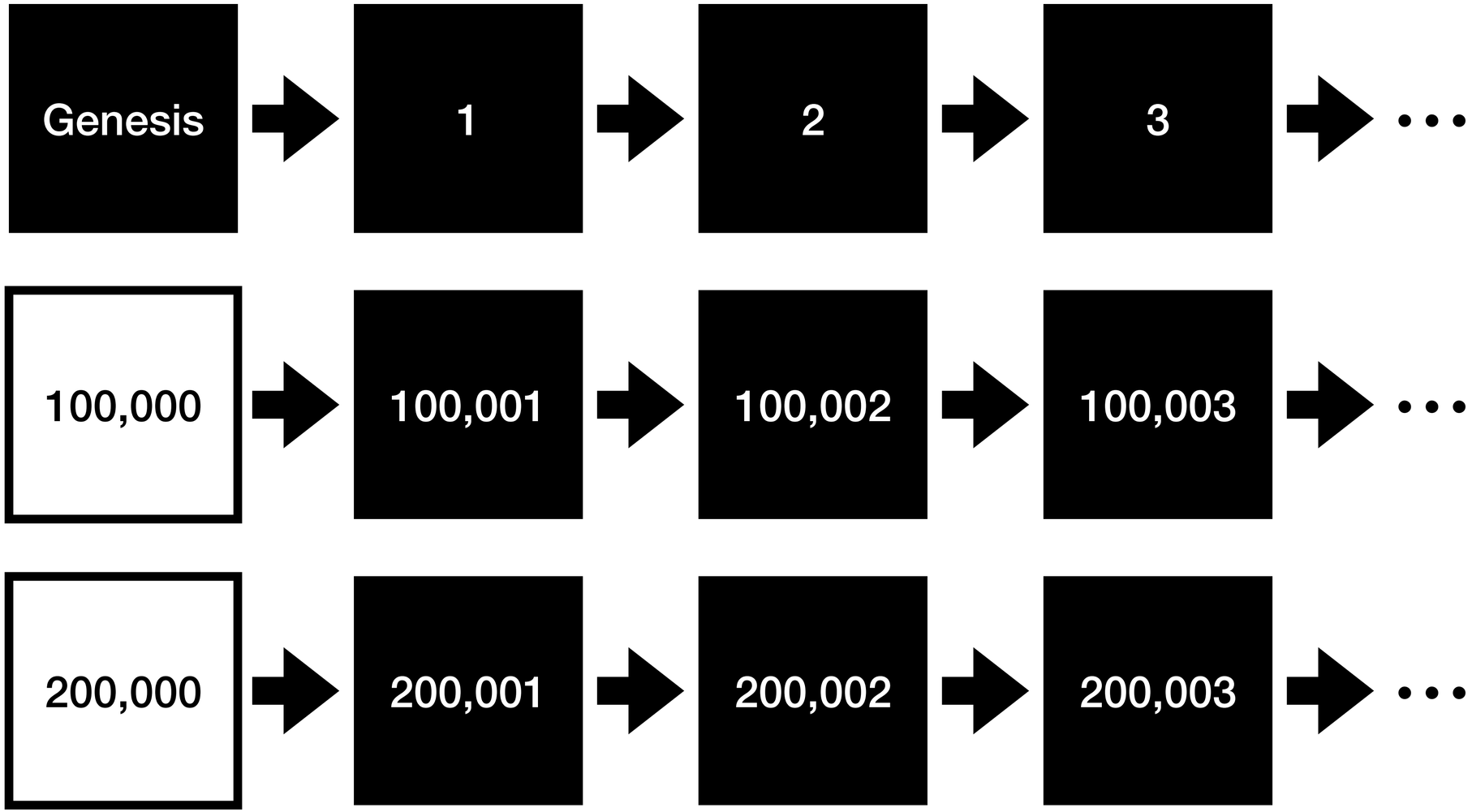
With Utreexo

Efficient parallel validation



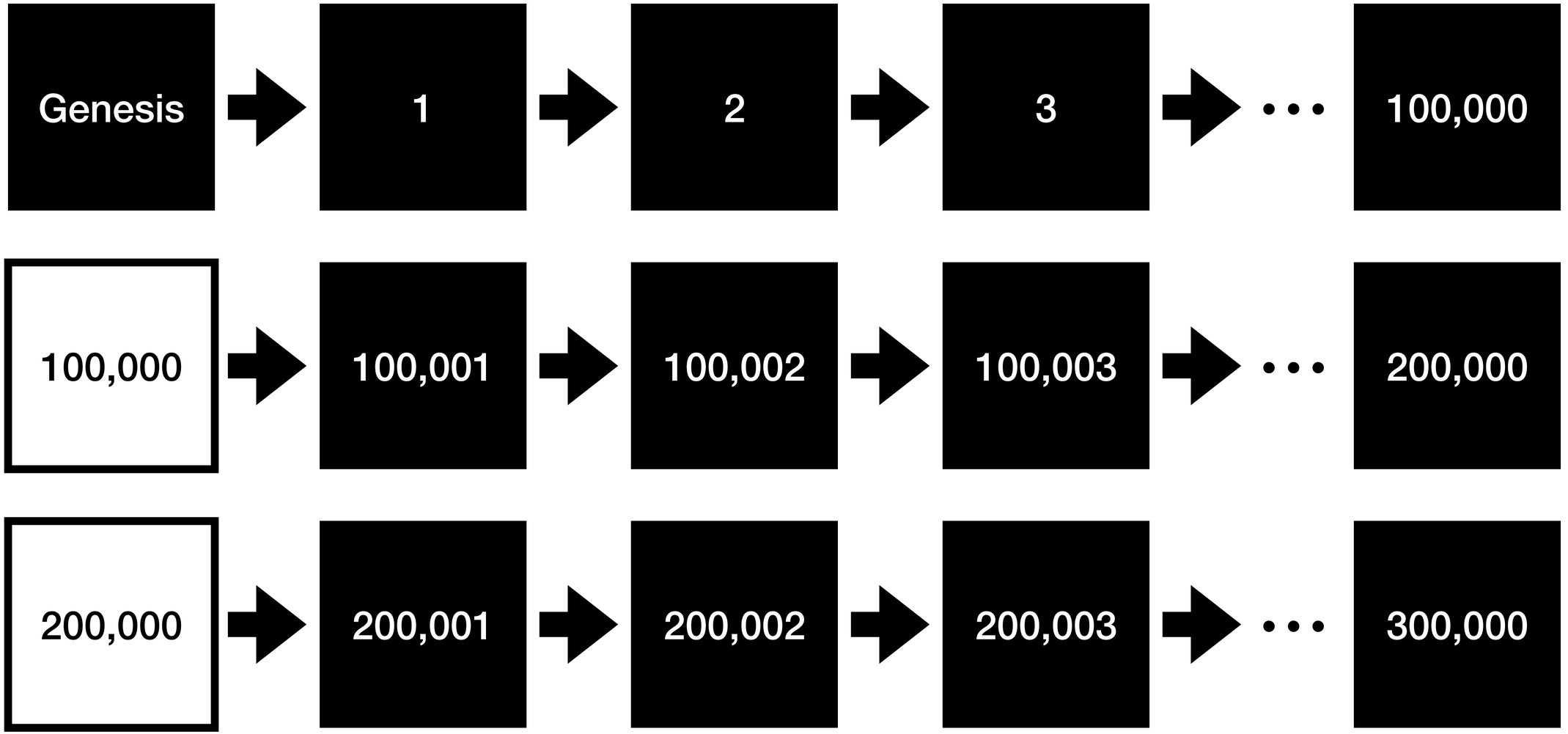
With Utreexo

Efficient parallel validation



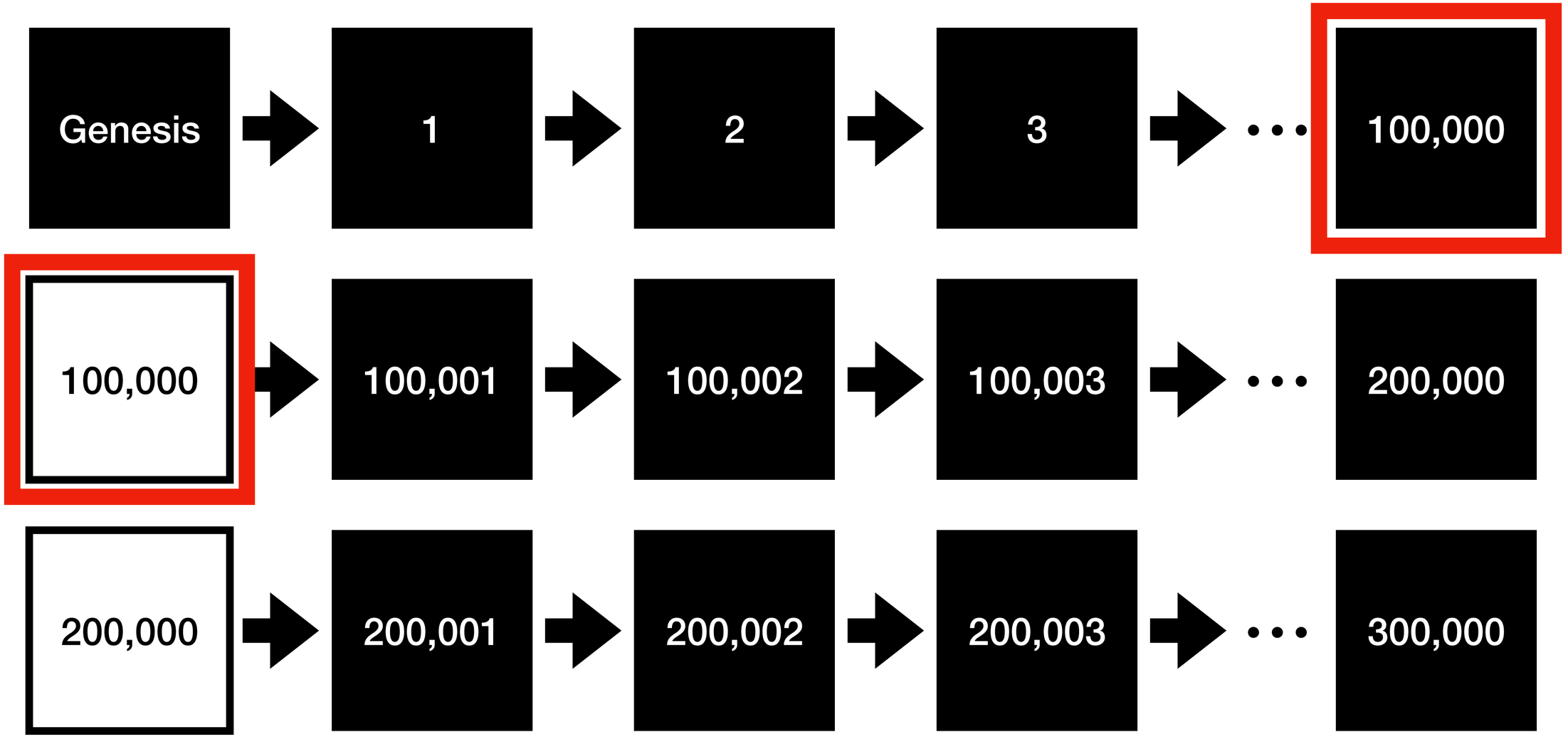
With Utreexo

Efficient parallel validation



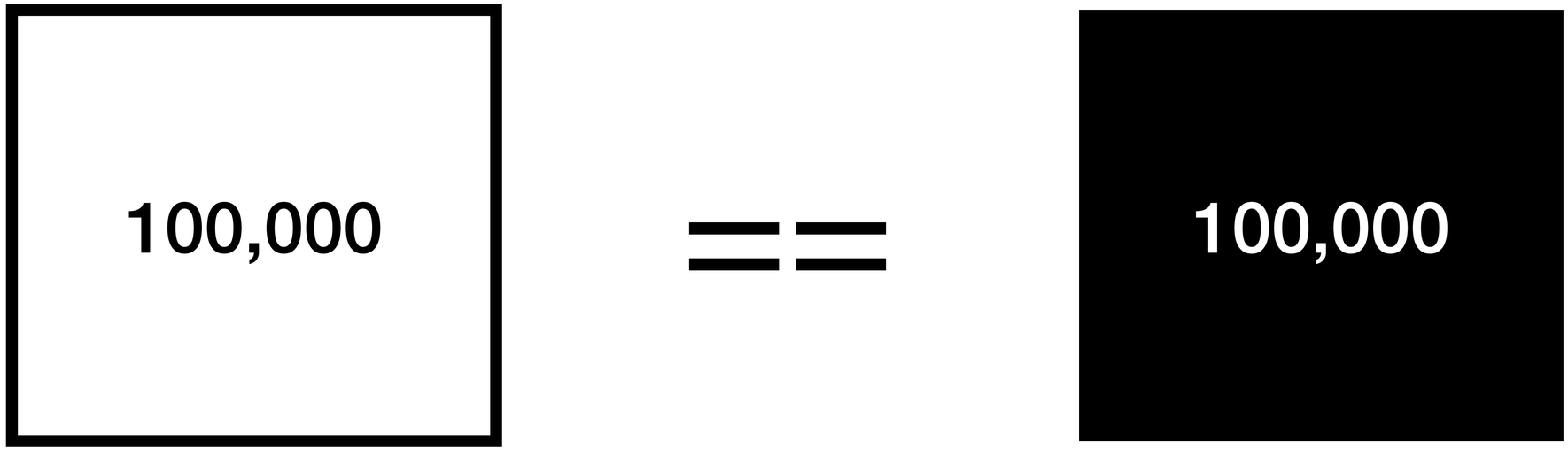
With Utreexo

Efficient parallel validation



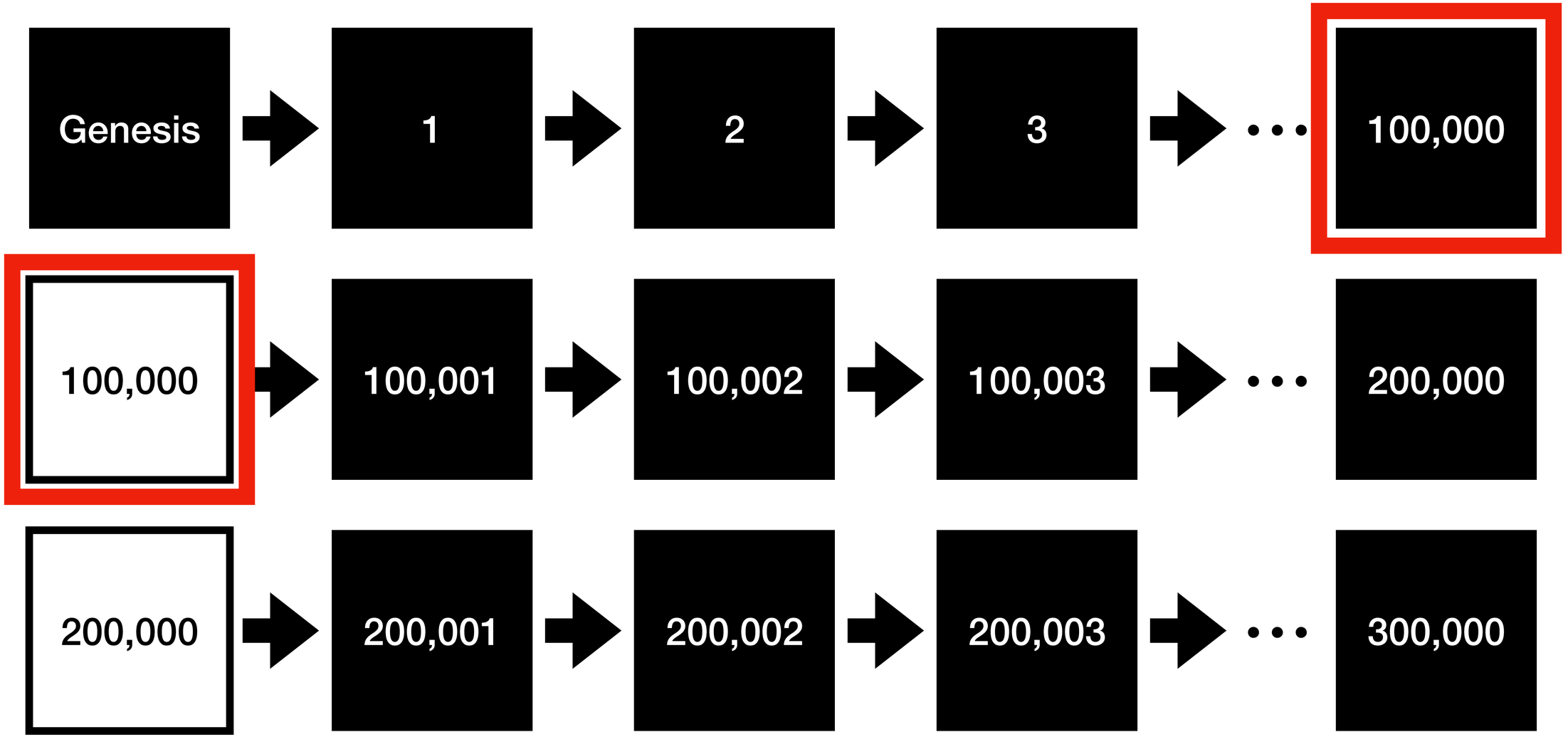
Compare roots

Root at 100,000 becomes trusted if equal



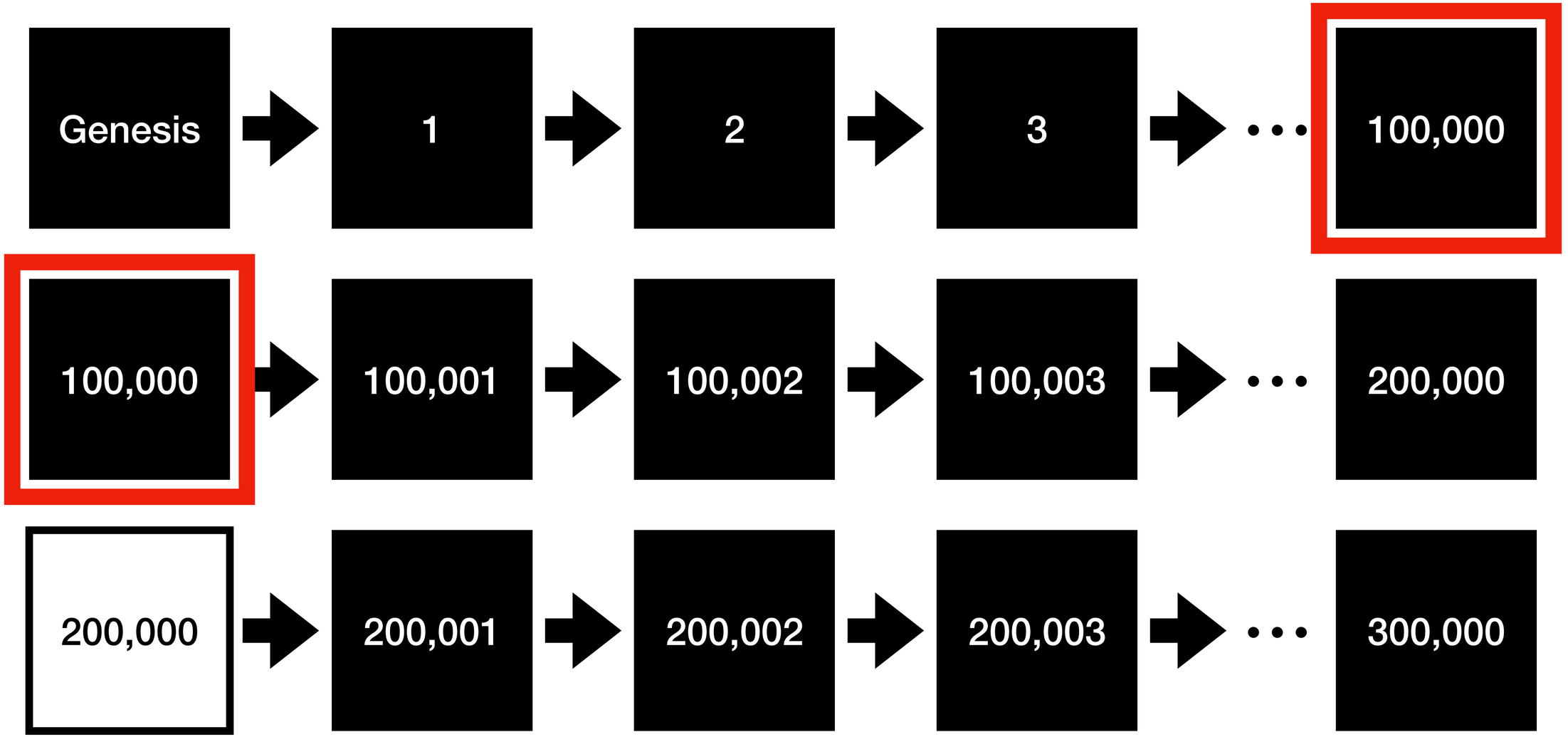
With Utreexo

Efficient parallel validation



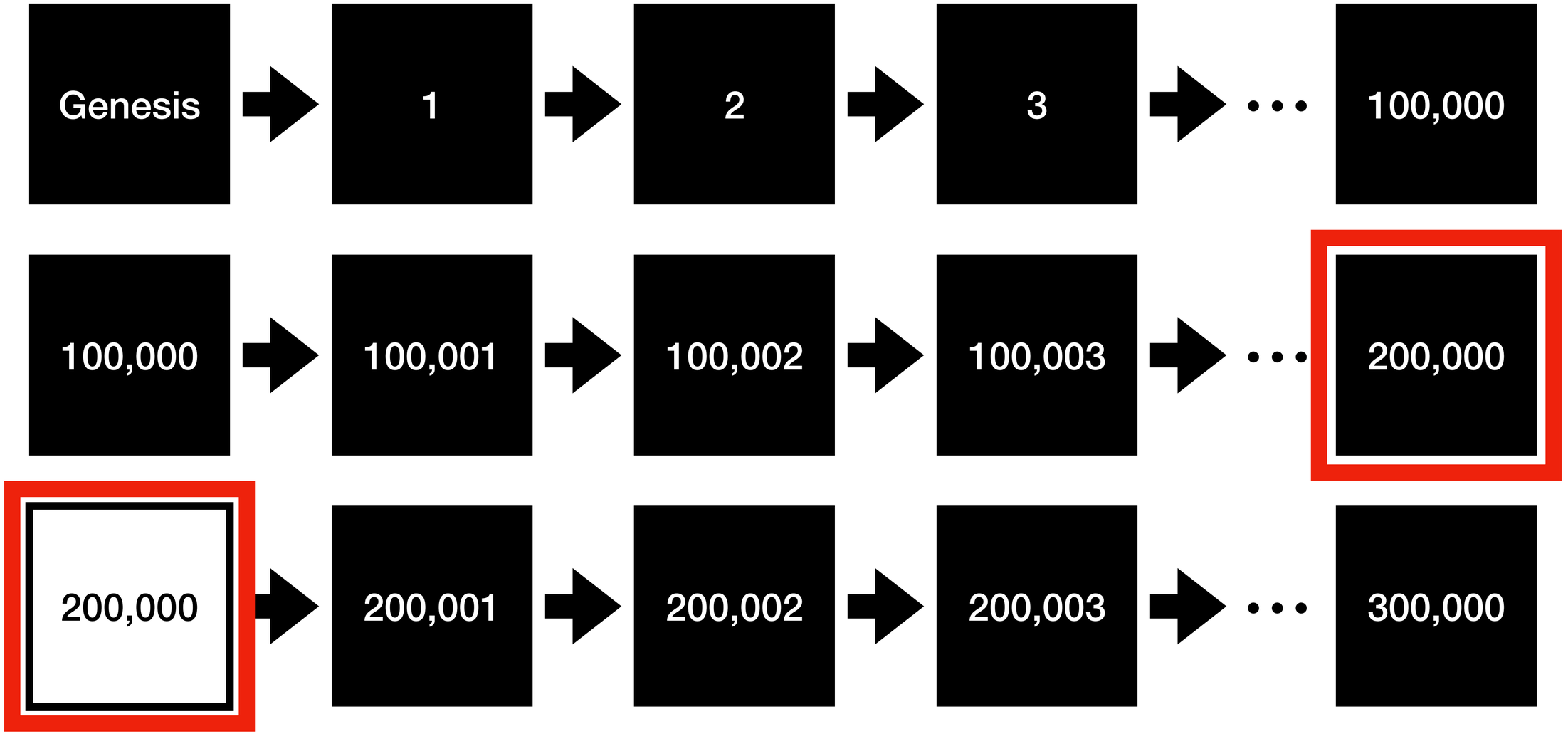
With Utreexo

Efficient parallel validation



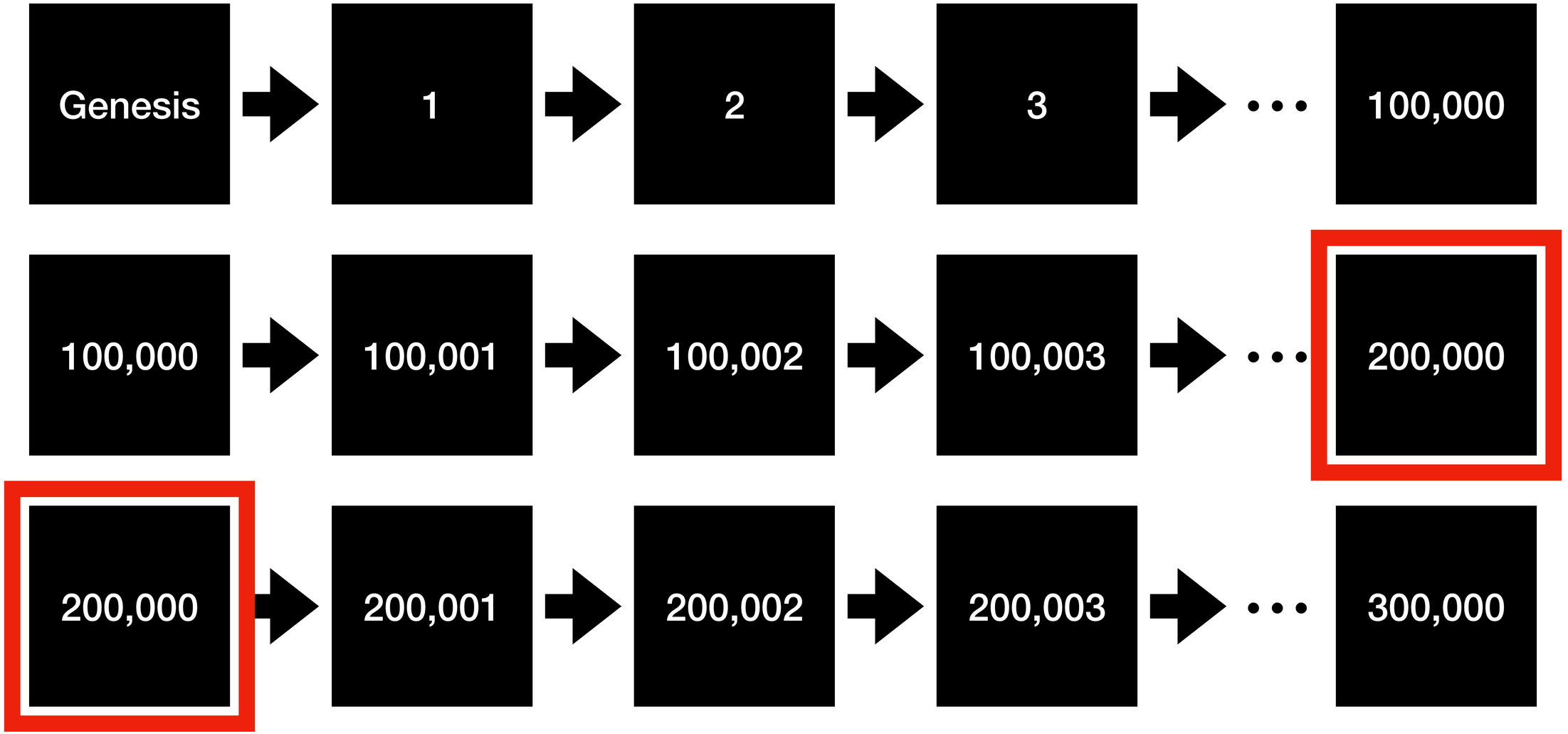
With Utreexo

Efficient parallel validation



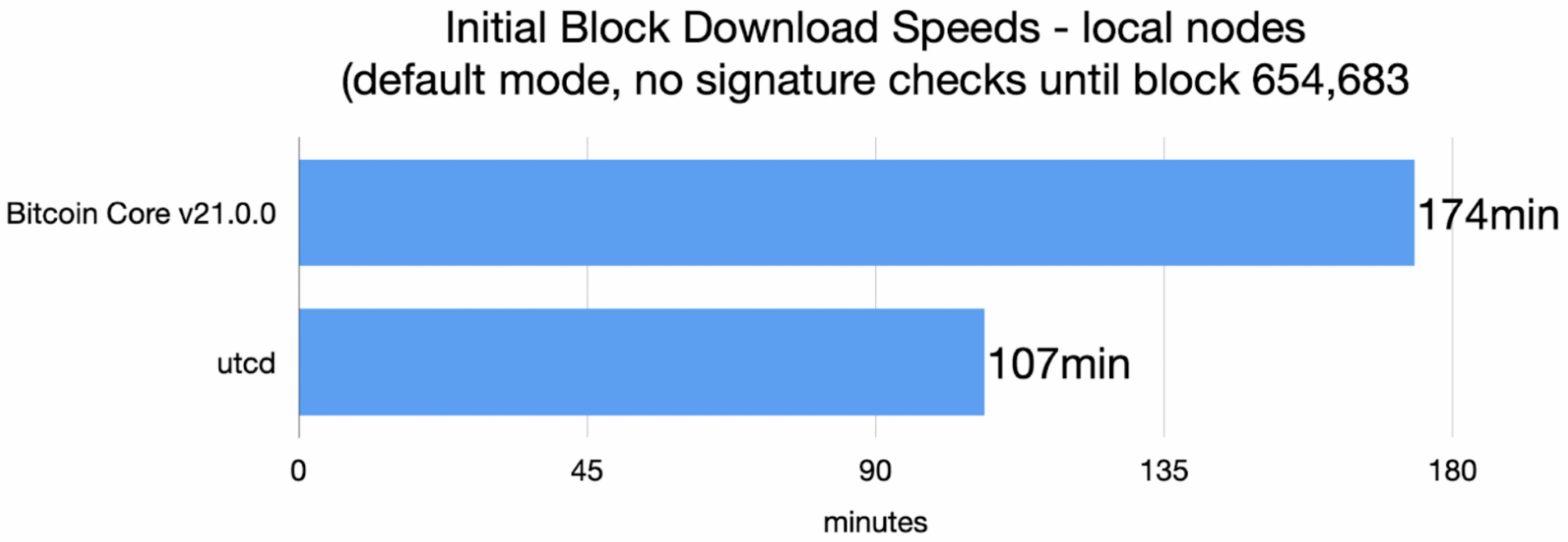
With Utreexo

Efficient parallel validation



Efficient Parallel Validation

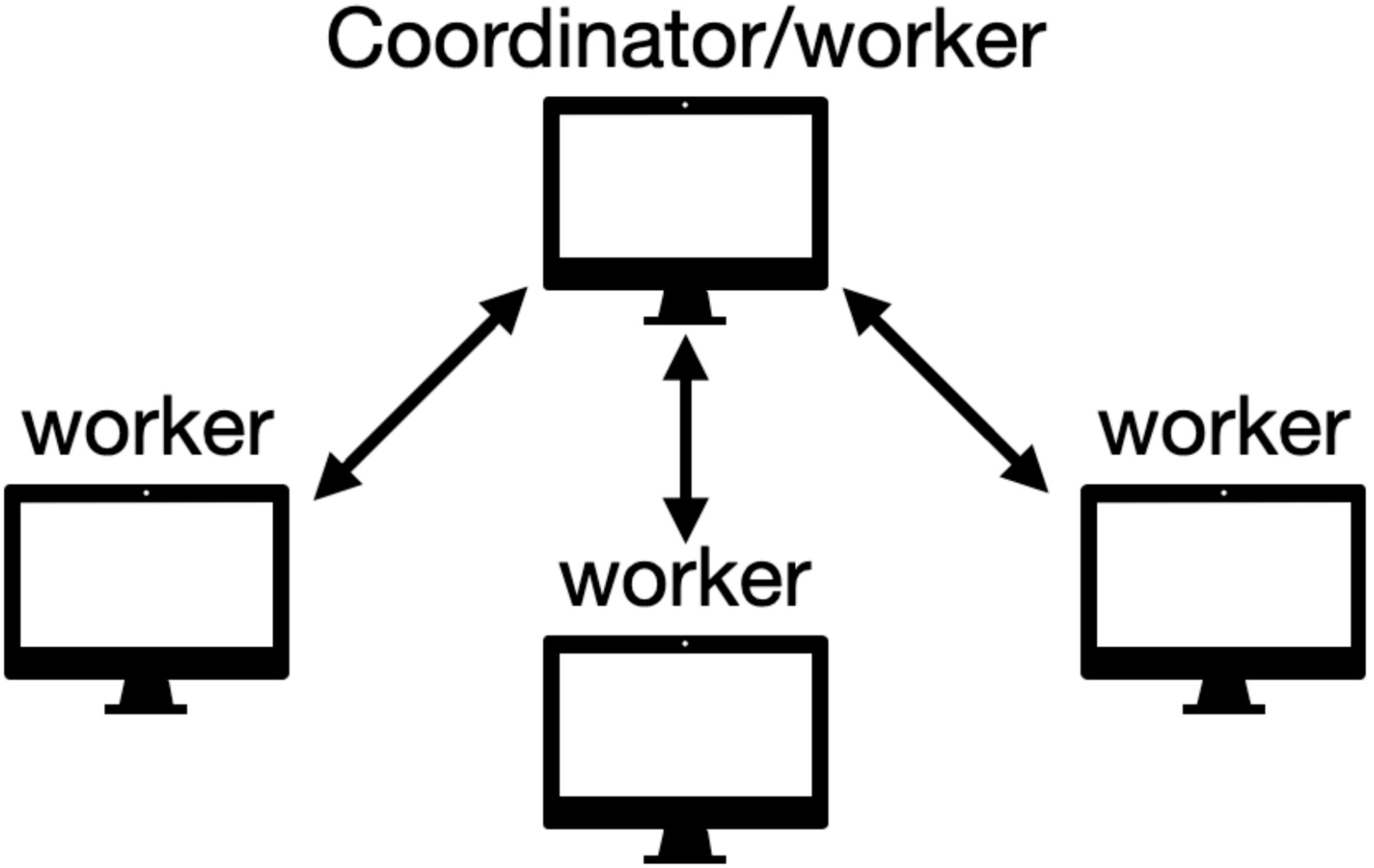
SHA256 🐇 > Map access 🐢



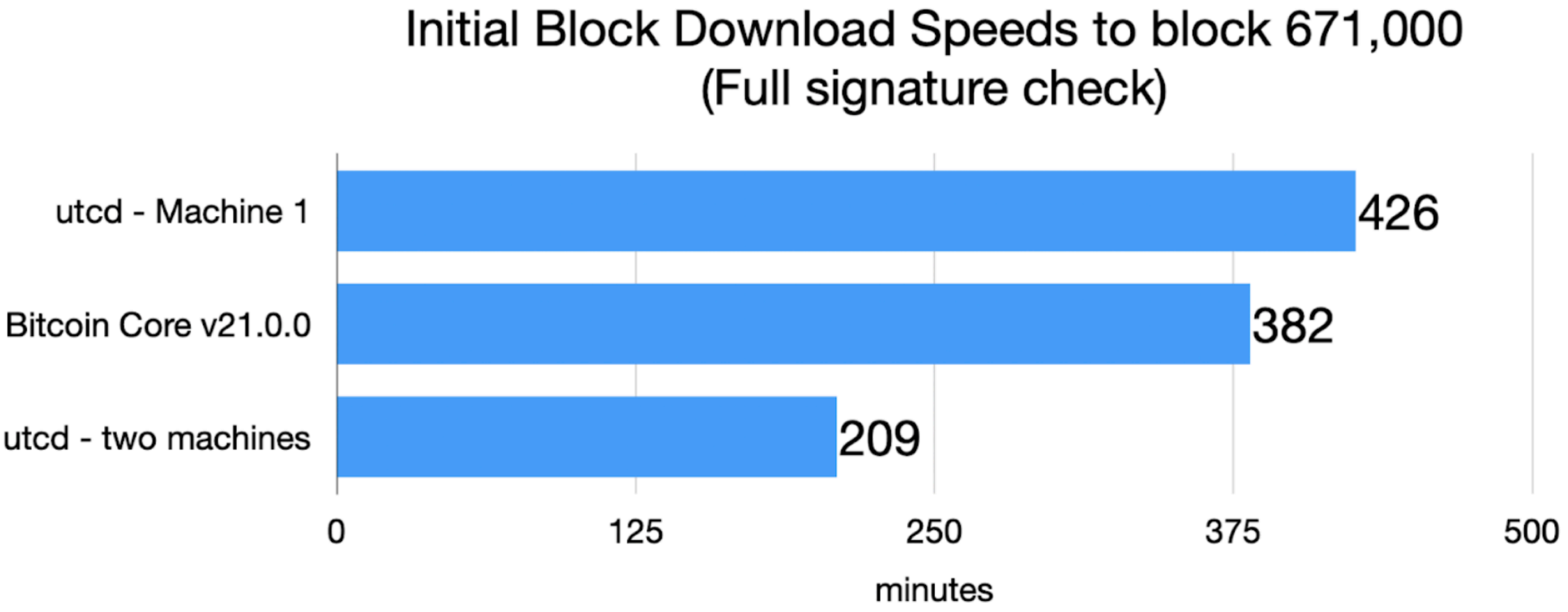
https://blog.bitmex.com/faster-blockchain-validation-with-utreexo-accumulators/

~62% faster blockchain validation

https://blog.bitmex.com/faster-blockchain-validation-with-utreexo-accumulators/



https://blog.bitmex.com/faster-blockchain-validation-with-utreexo-accumulators/

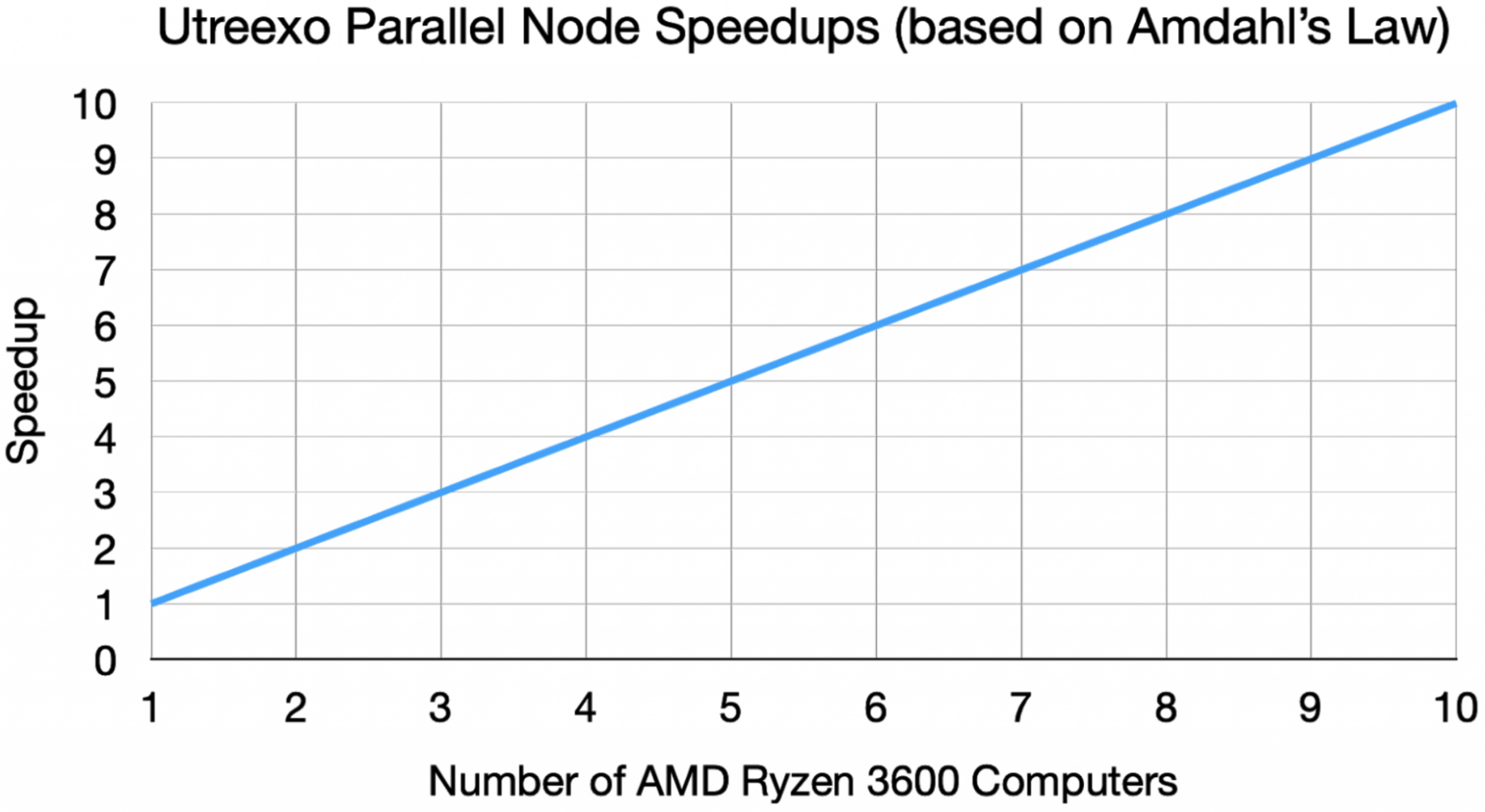


https://blog.bitmex.com/faster-blockchain-validation-with-utreexo-accumulators/

2x faster blockchain validation

When using 2 computers

https://blog.bitmex.com/faster-blockchain-validation-with-utreexo-accumulators/


Utreexo Parallel Node Speedups (based on Amdahl's Law)
Speedup
0
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
6
7
8
9
10
Number of AMD Ryzen 3600 Computers

Summary

What I want you to take away

- We want to preserve trustlessness while having the most amount of Bitcoin users

Summary

What I want you to take away

- We want to preserve trustlessness while having the most amount of Bitcoin users
- Current bottleneck is with block validation

Summary

What I want you to take away

- We want to preserve trustlessness while having the most amount of Bitcoin users
- Current bottleneck is with block validation
- Utreexo helps eliminate this bottleneck

Questions?