

# **Why the UTXO model scales more**

**+ how Utreexo enhances the advantages of the UTXO model**



<https://github.com/kcalvinalvin/btcsingapore-09-2024>

# **Account vs UTXO**

# **Scalability**

**UTXO model is better**

# **What is scalability?**

**My viewpoint**

**Transactions per  
compute resource**





**Software runs on hardware**

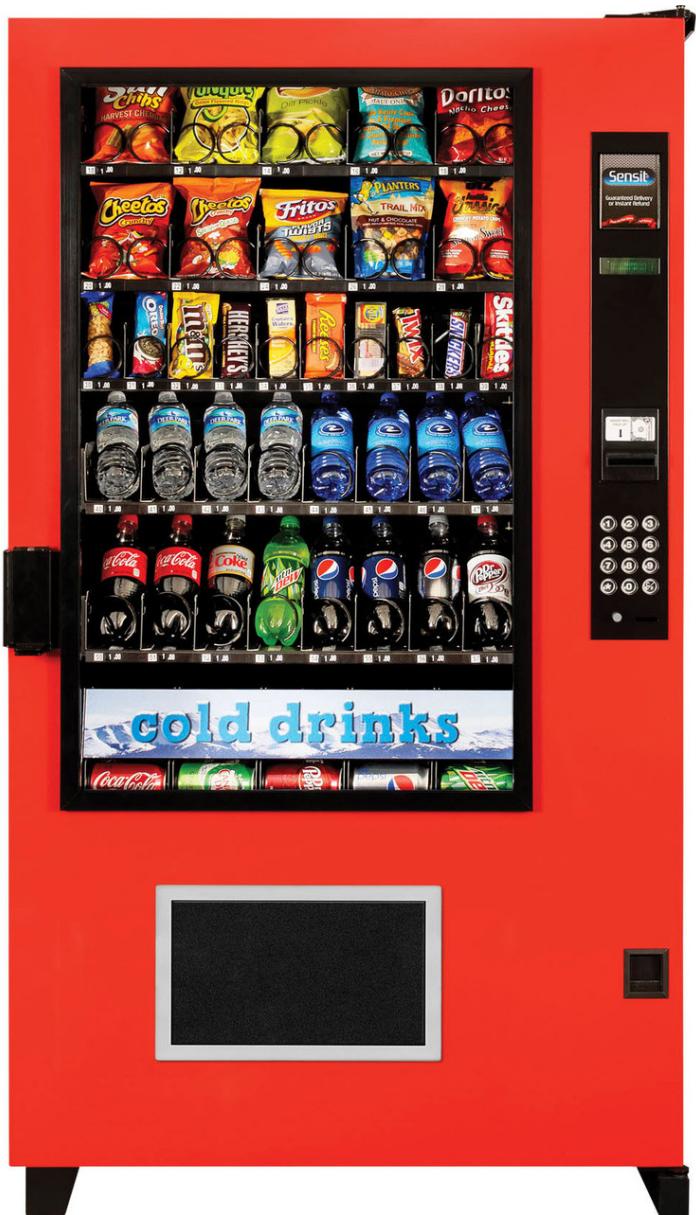
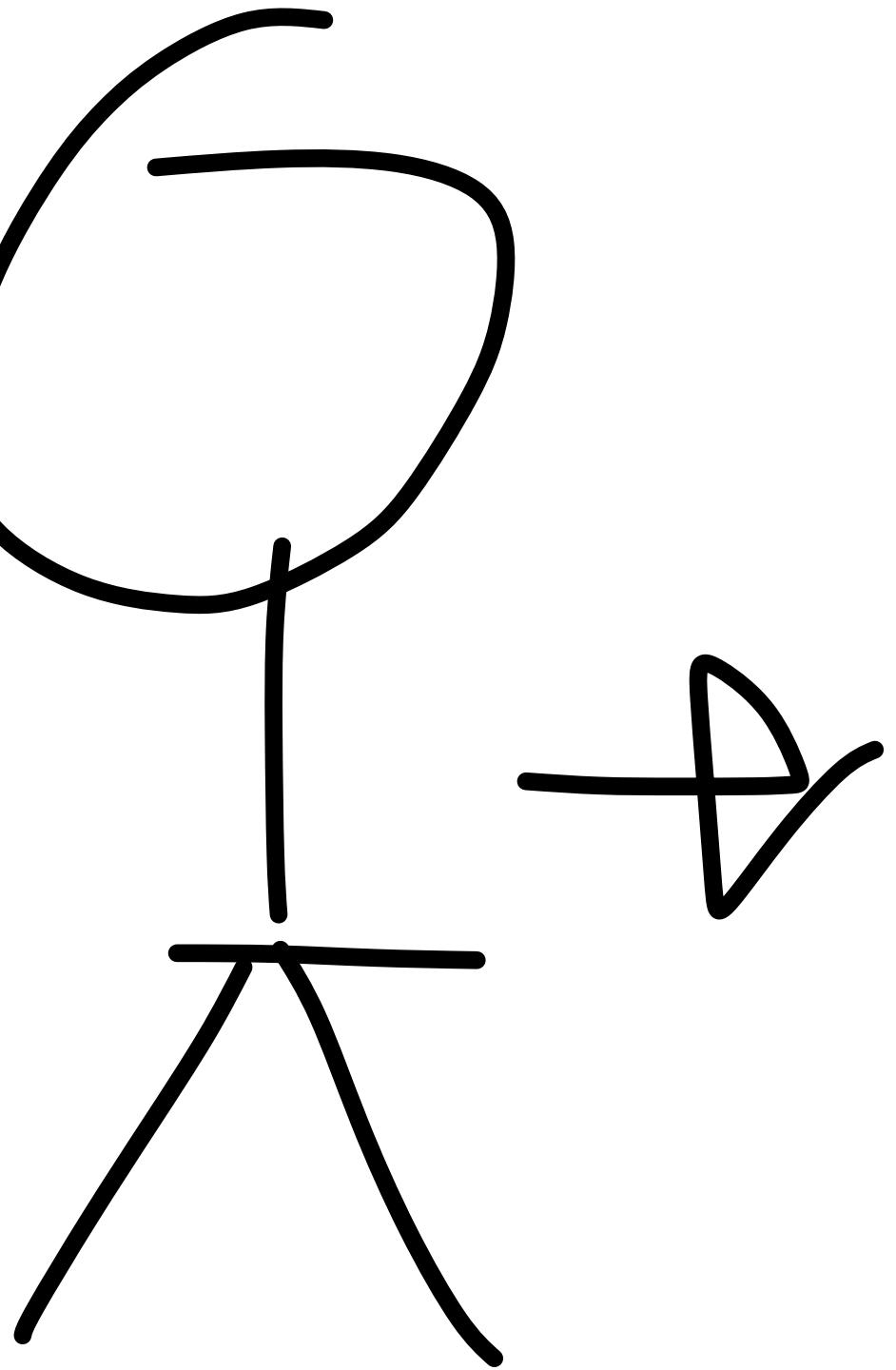
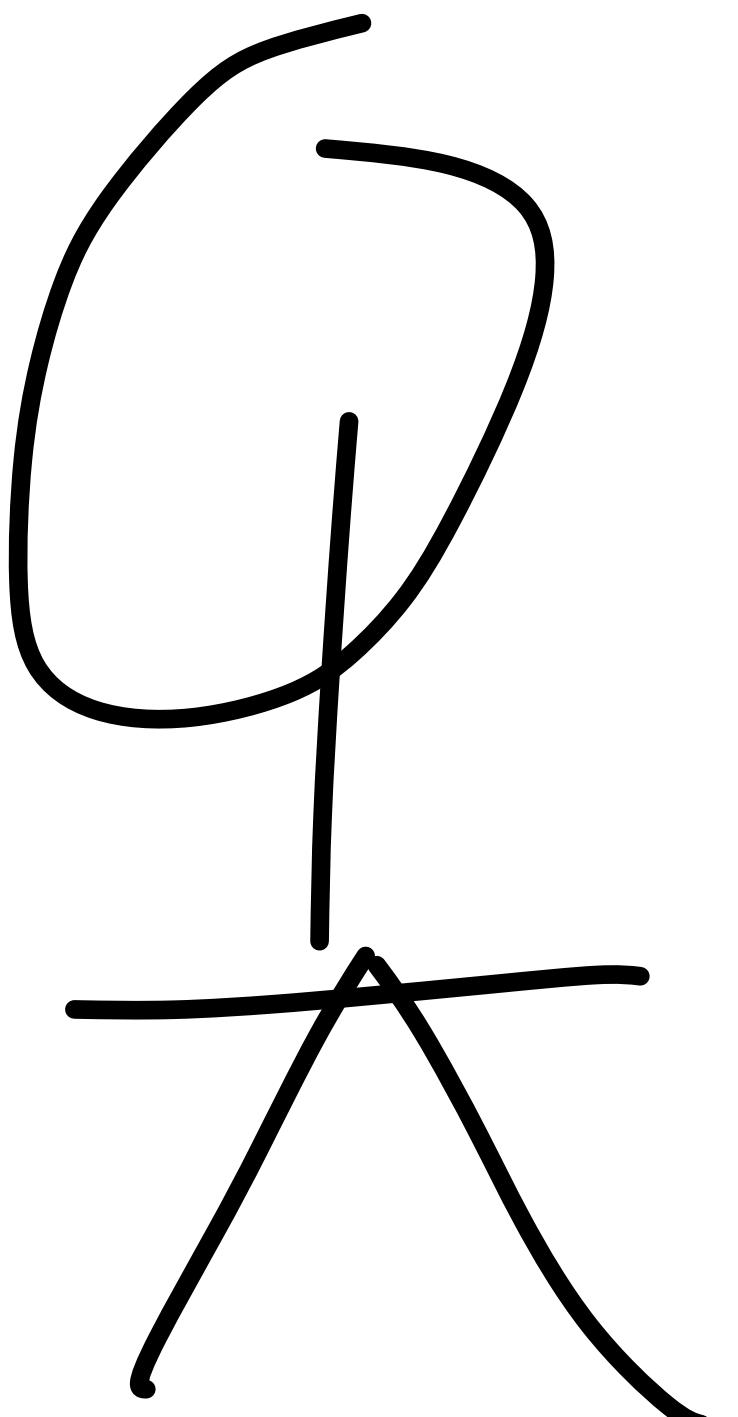
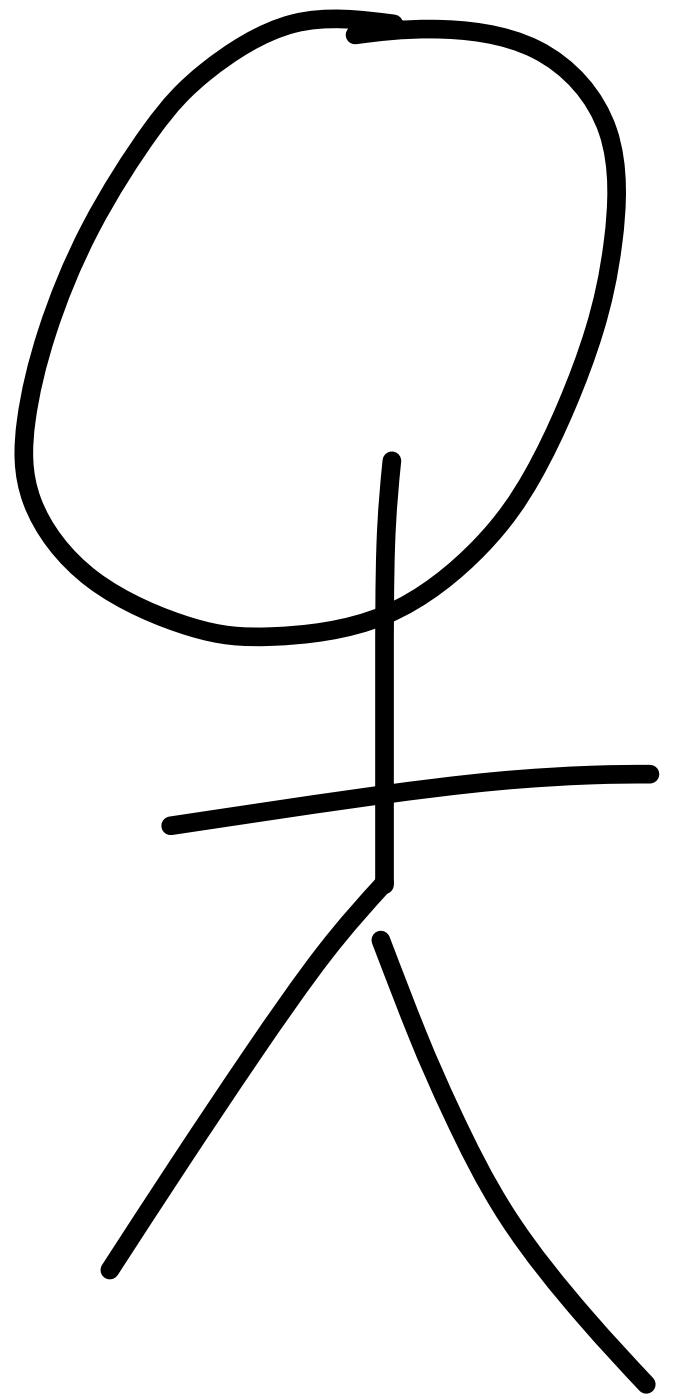
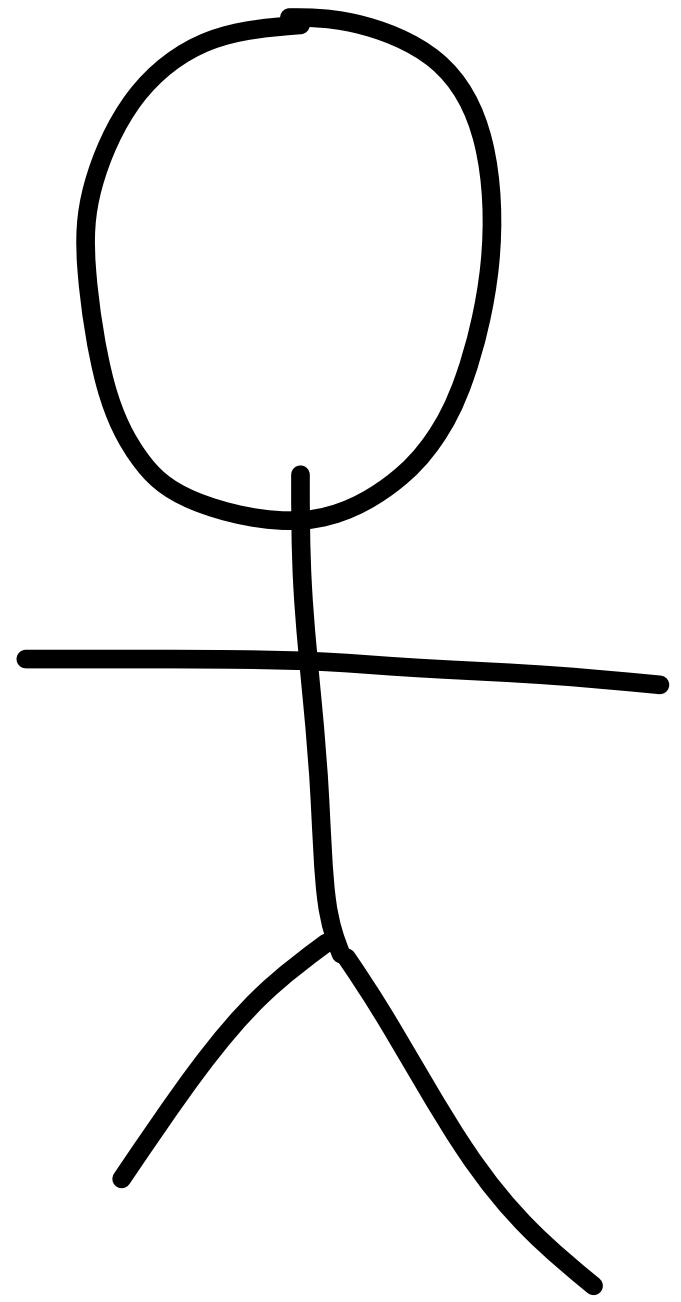
**Software must be  
optimized for hardware**

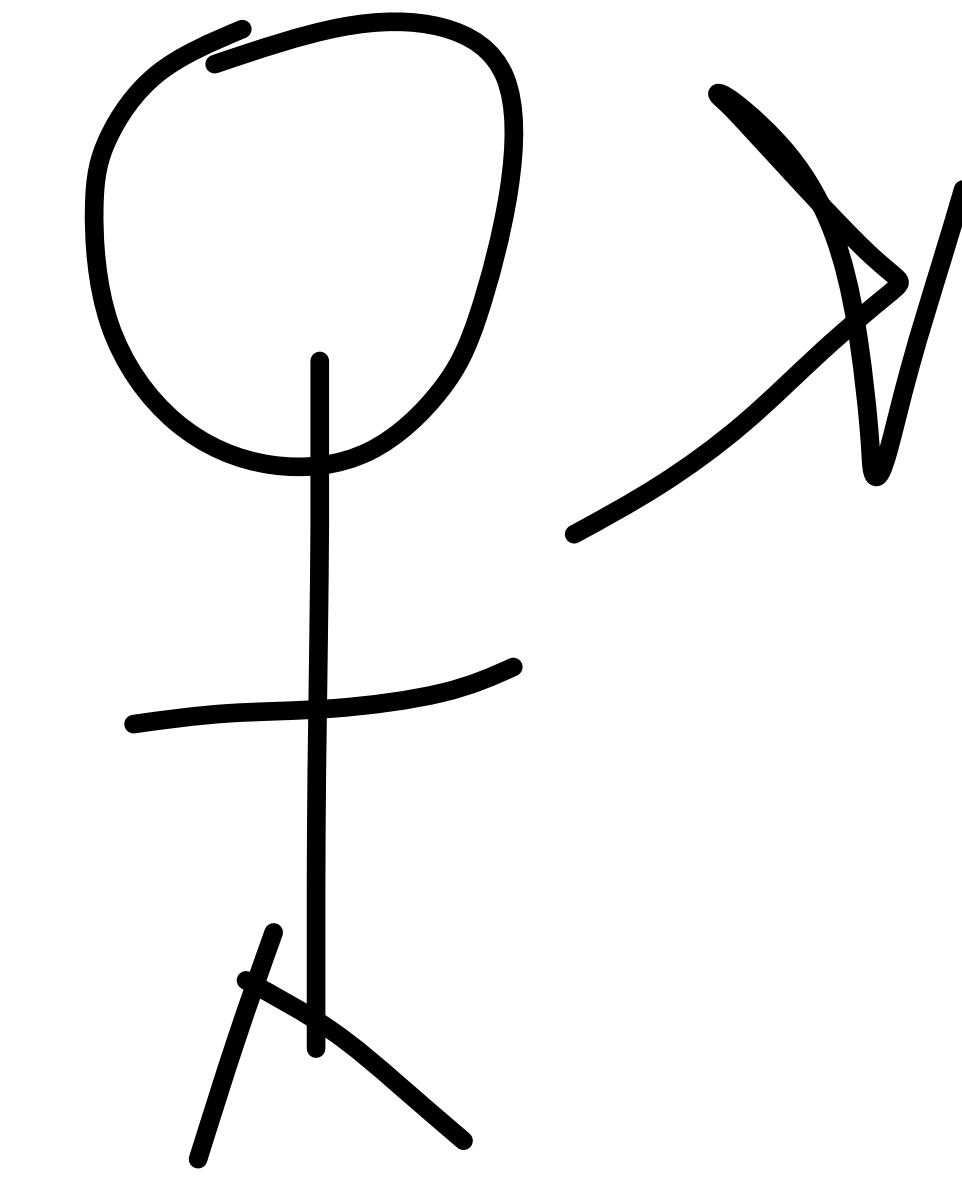
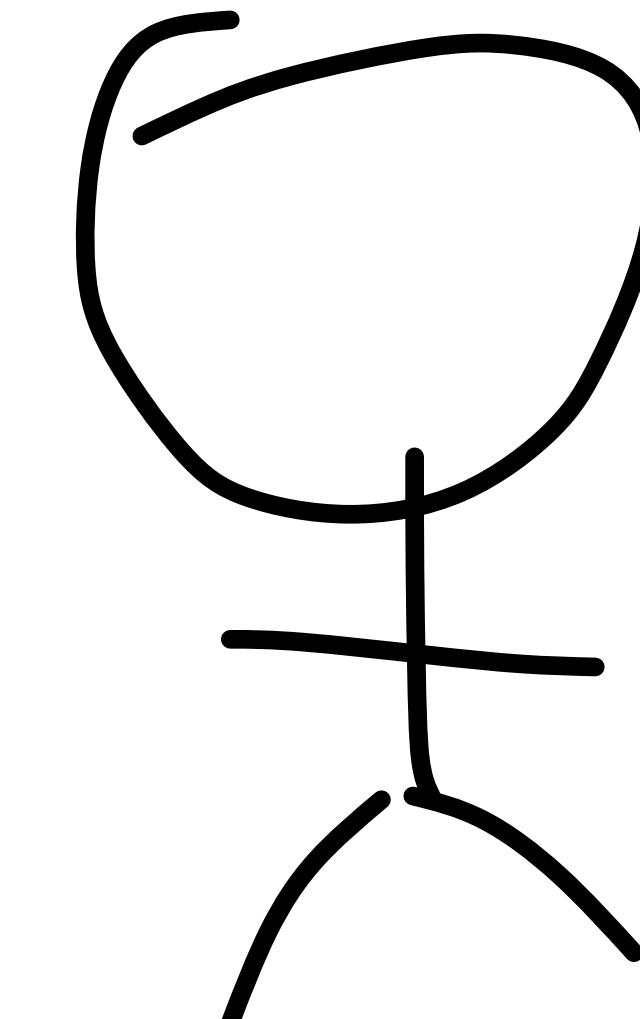
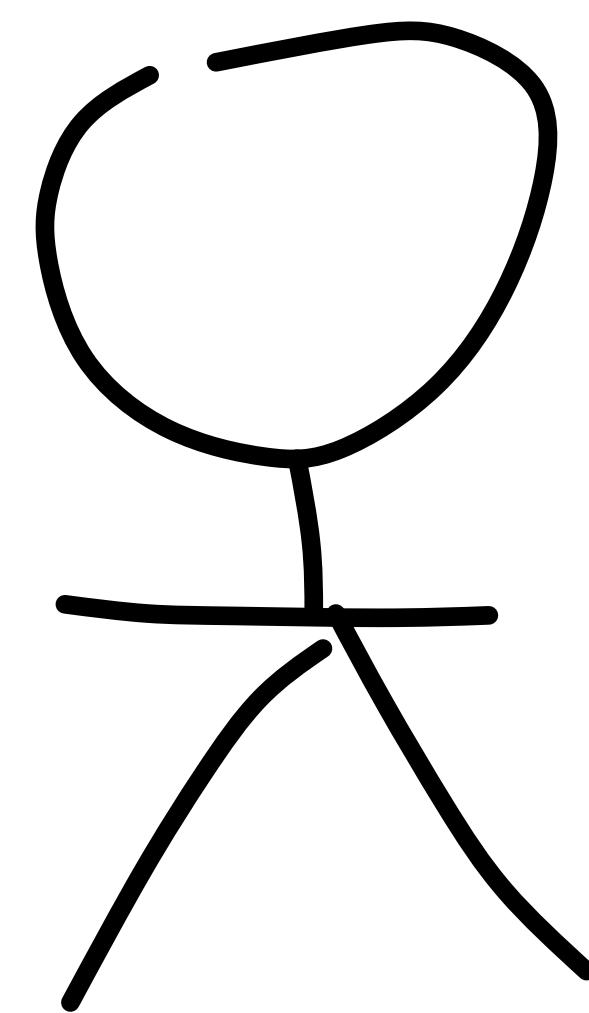
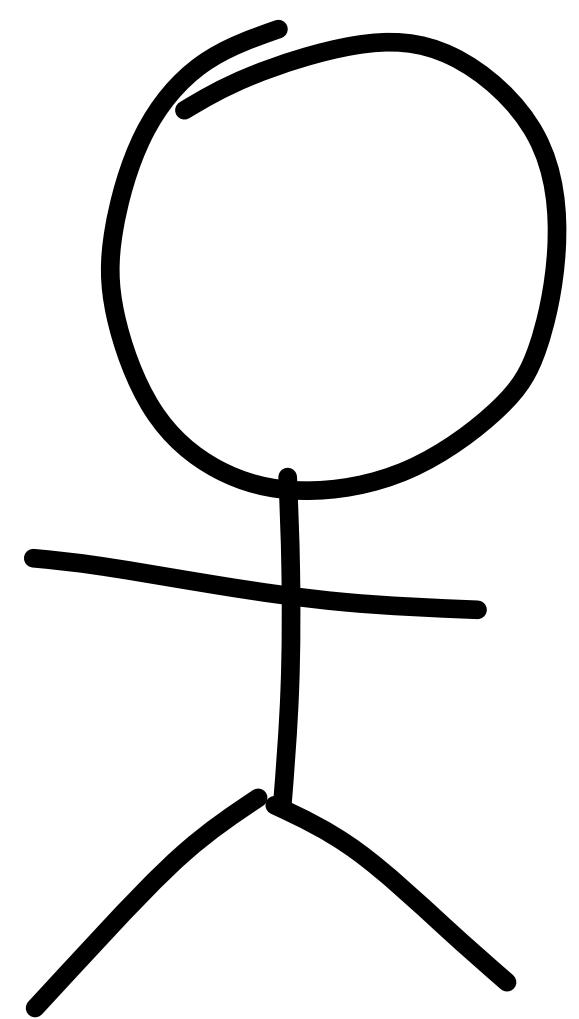
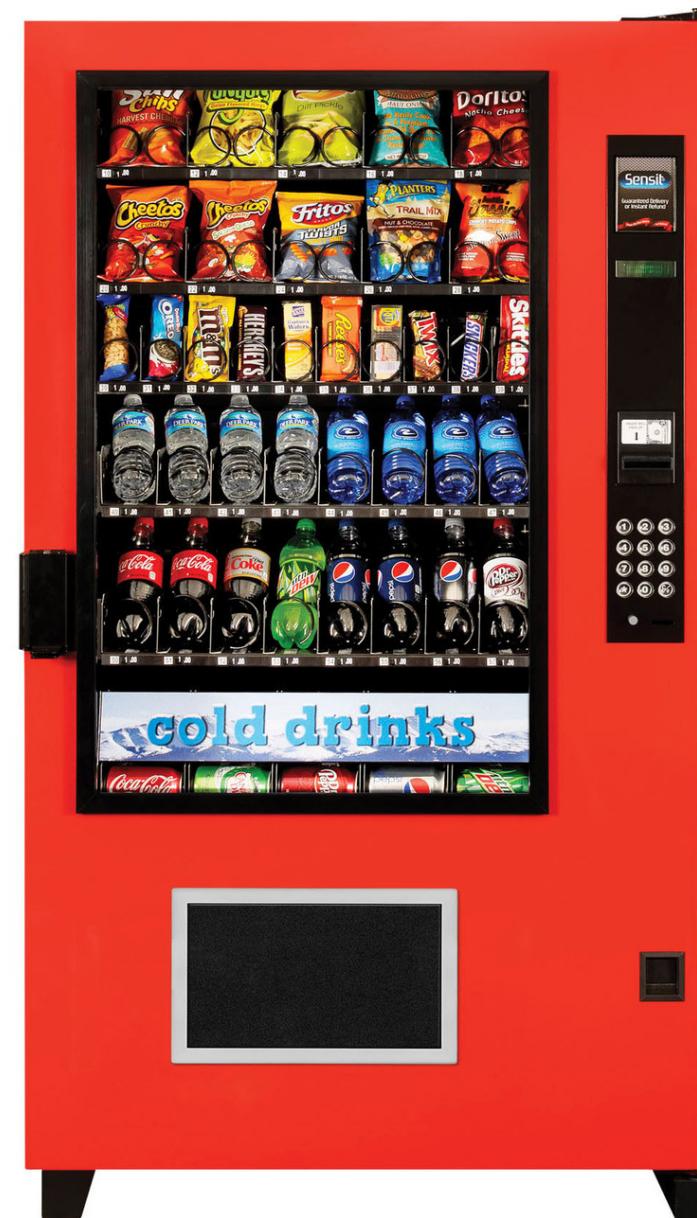
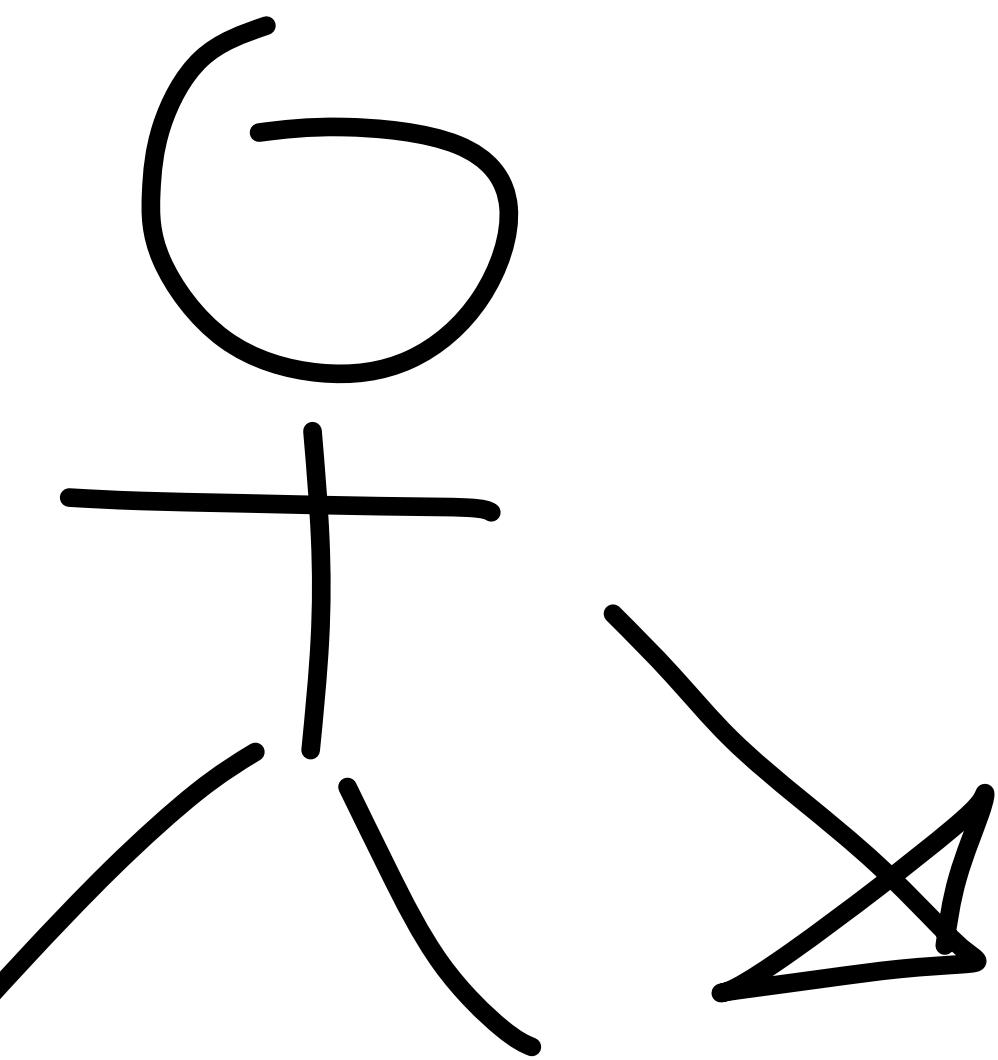
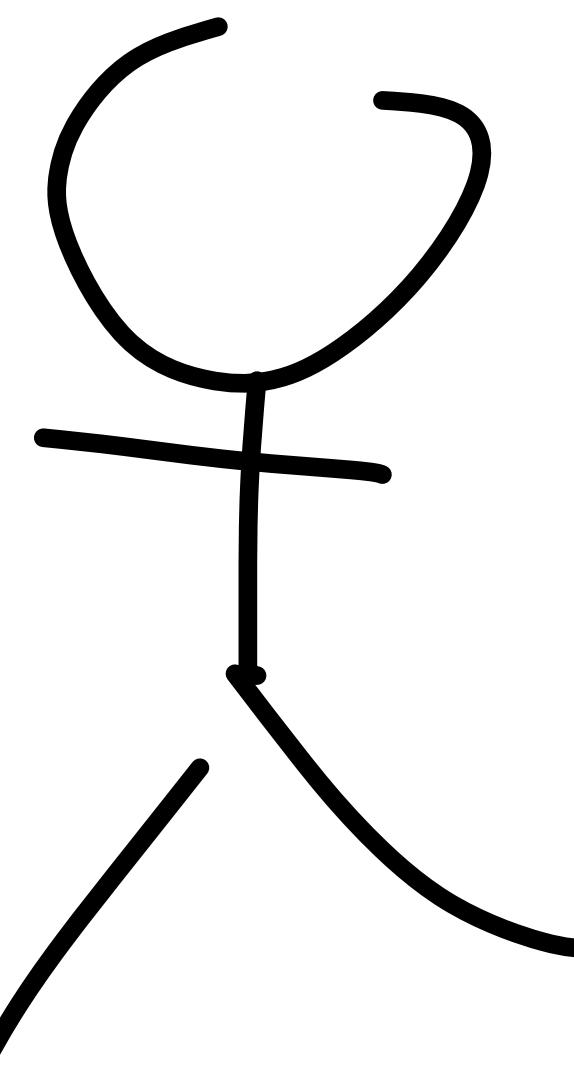
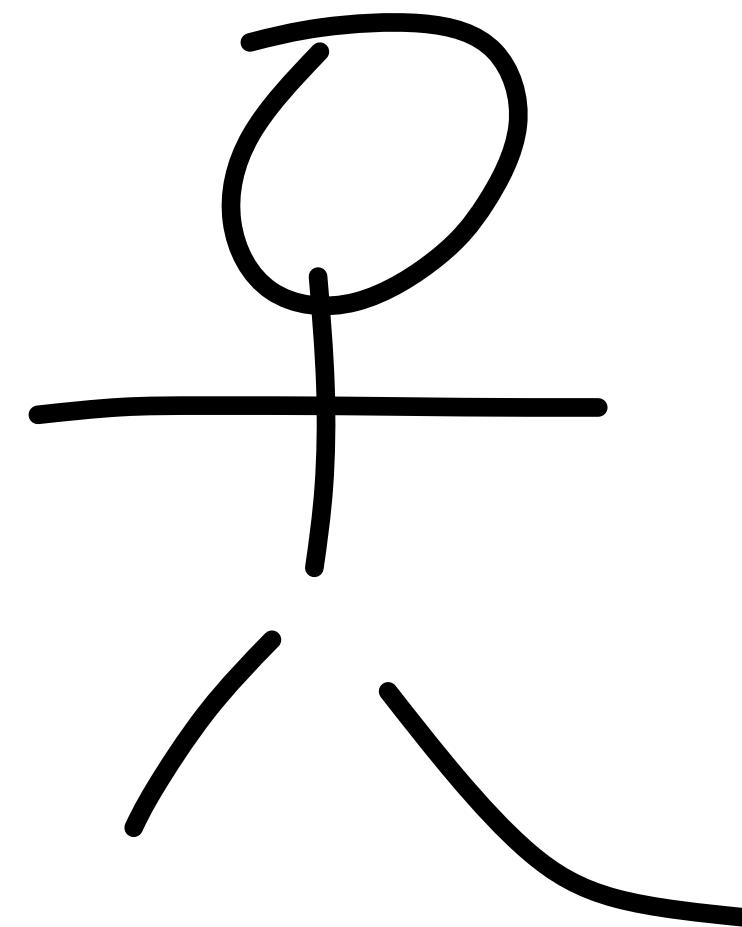
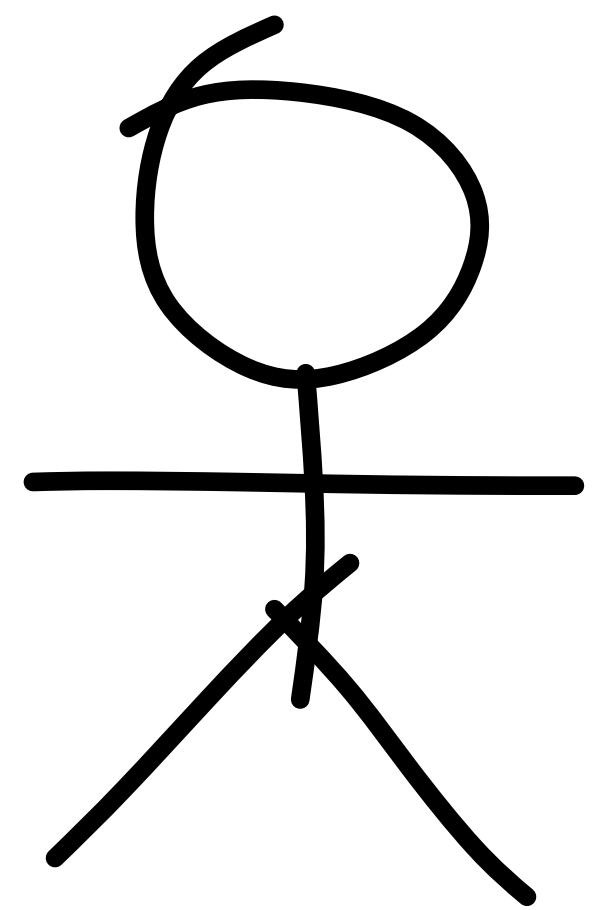
**Why is the UTXO model  
better?**

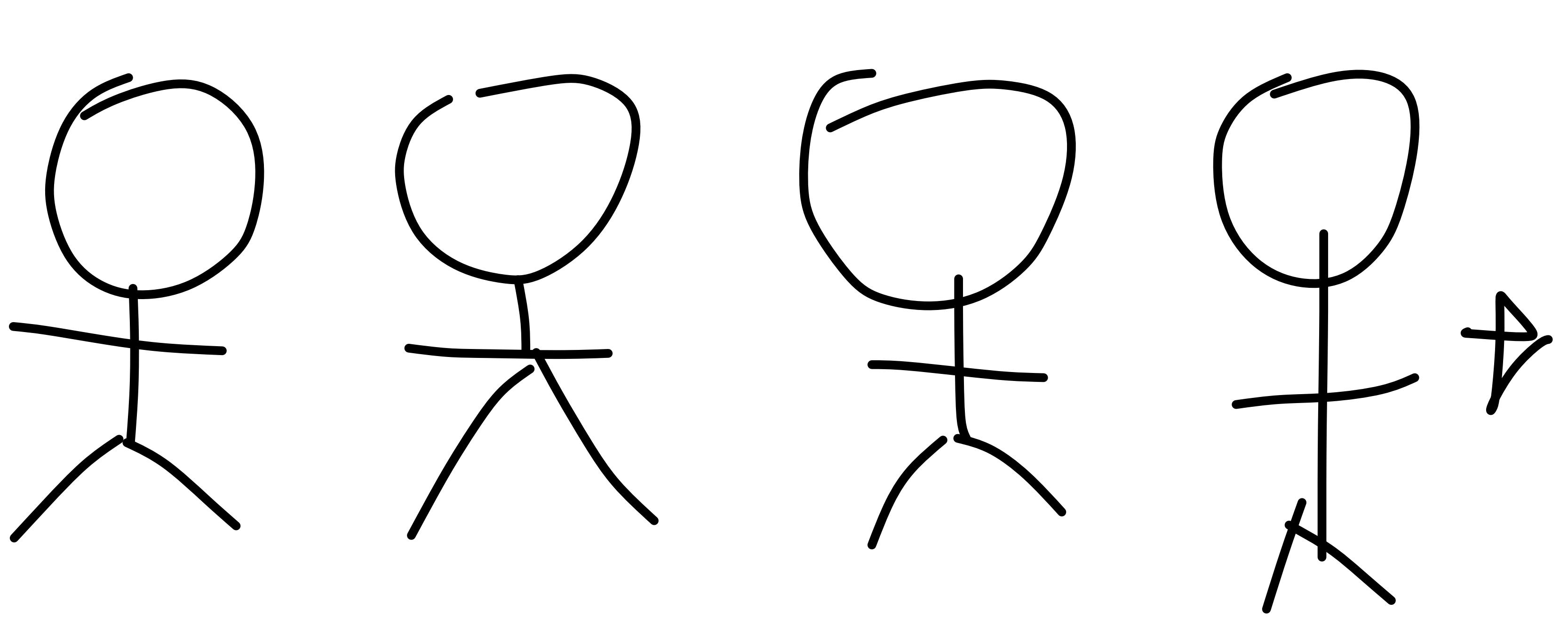
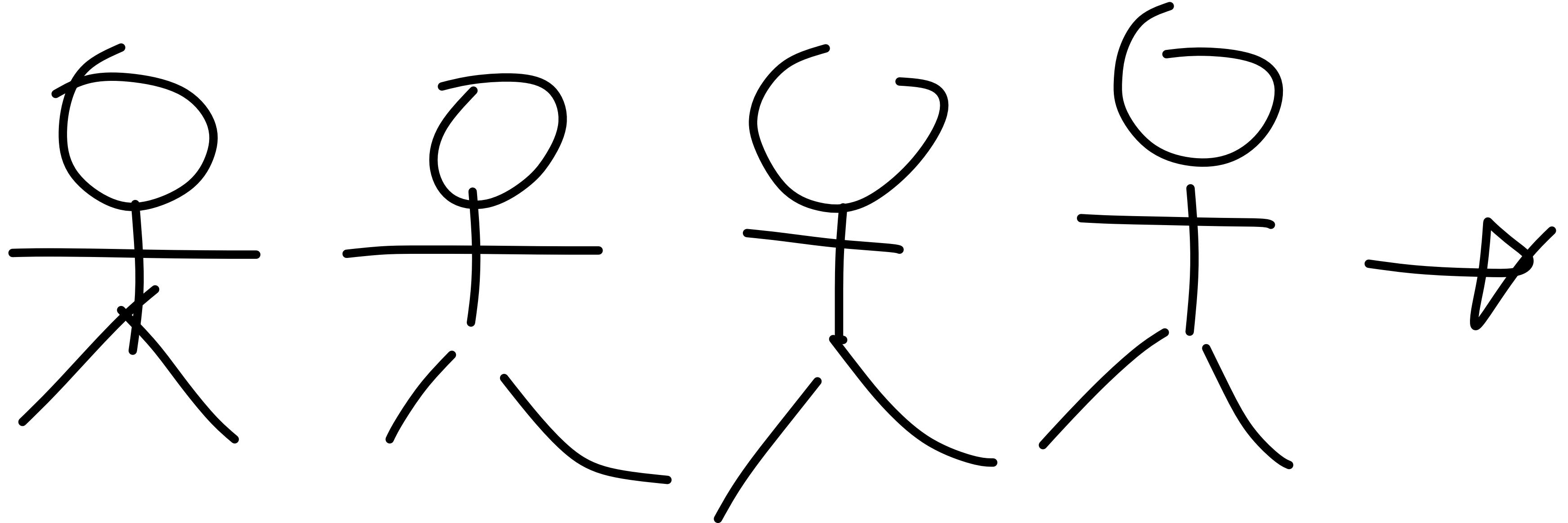
# **Better concurrency vs the account based model**

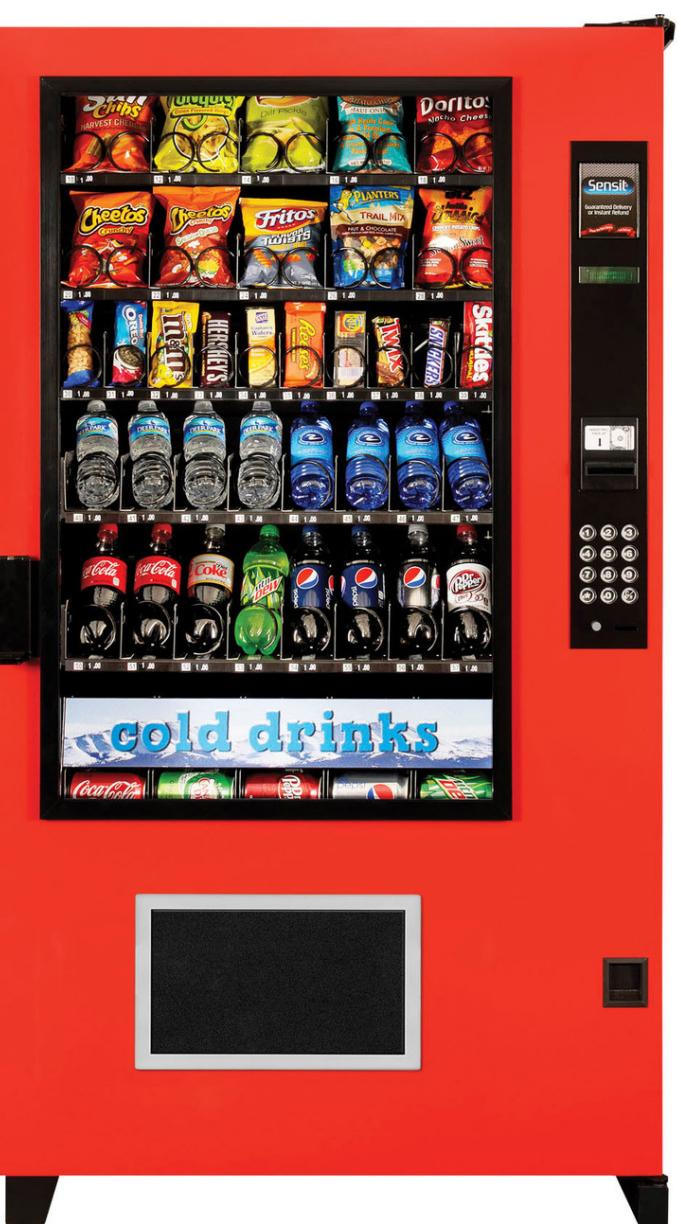
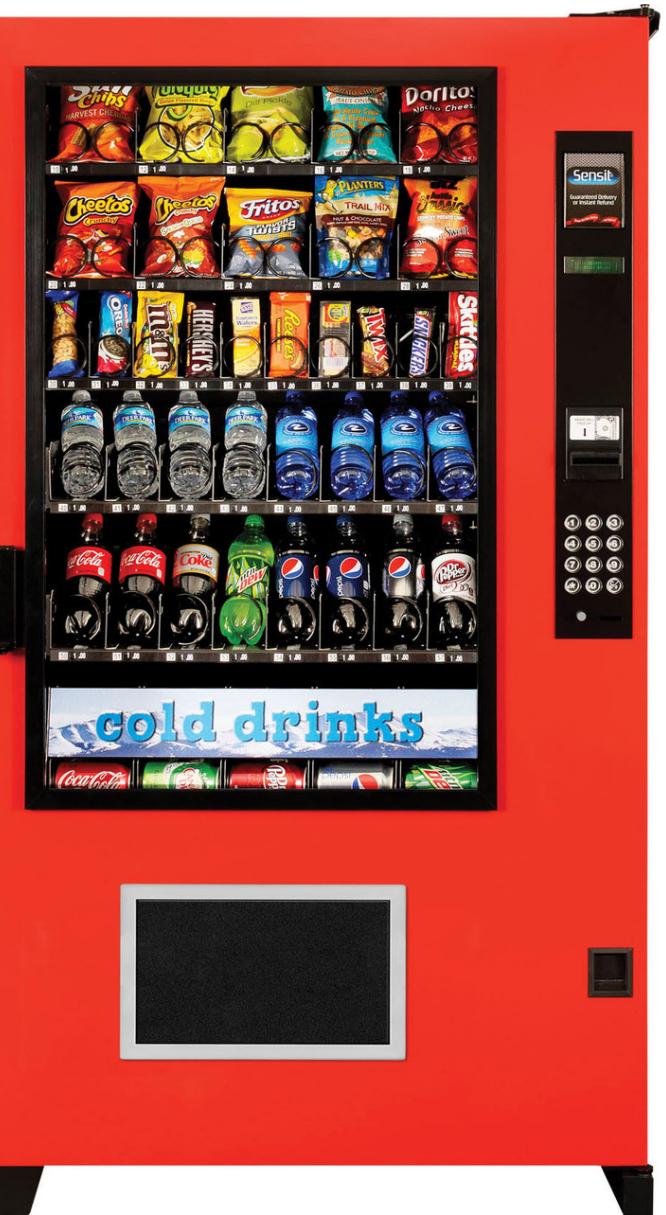
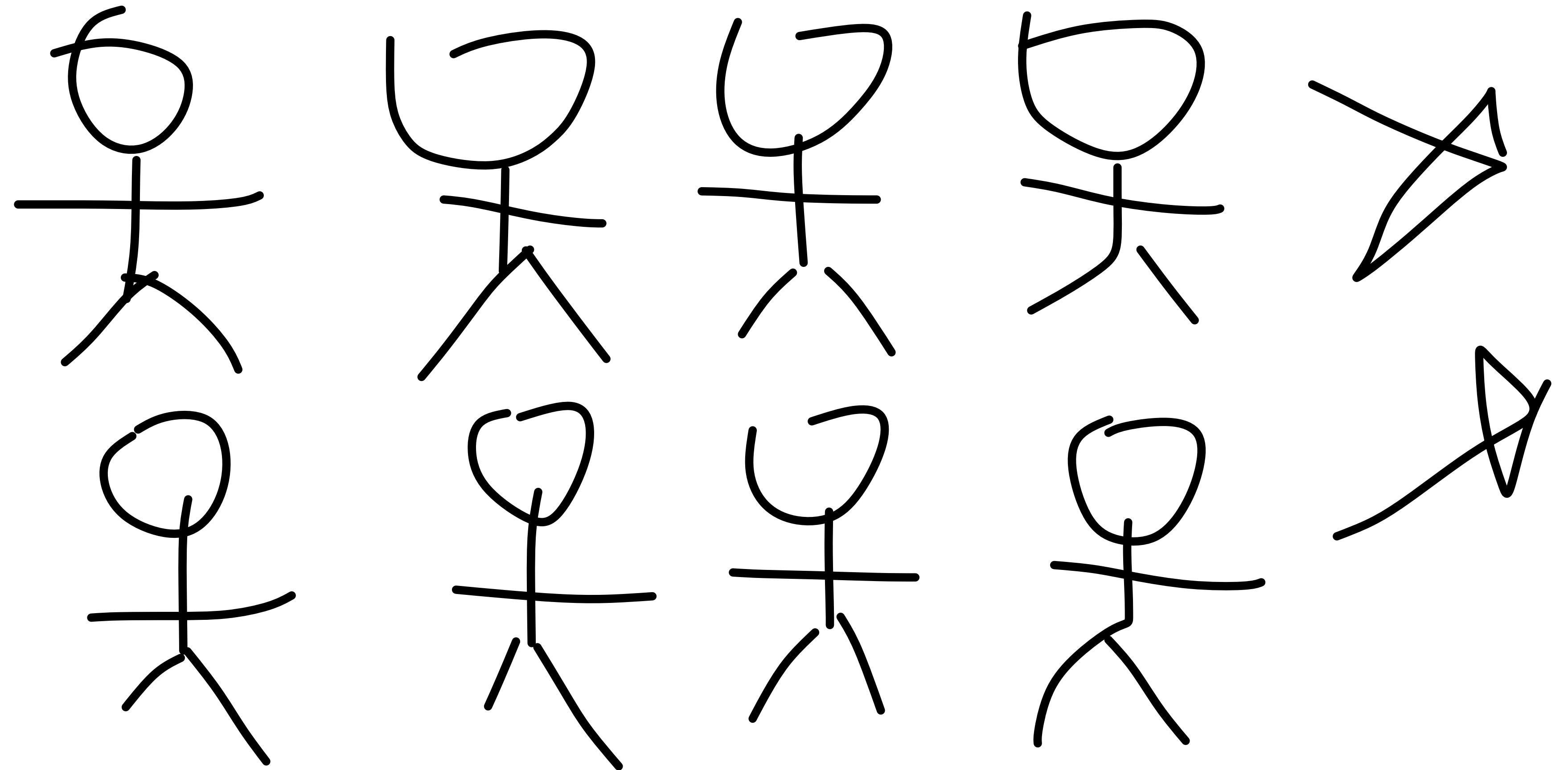
# **Faster block validation**

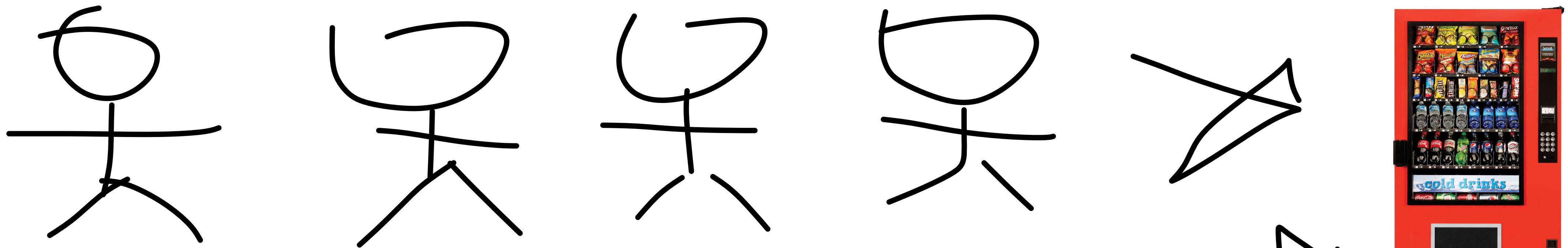
**Allows for more tx throughput  
per compute resource**











# Why the UTXO model scales better

## State change frequency

- UTXO state changes per block

# Why the UTXO model scales better

## State change frequency

- UTXO state changes per block
- It's ok to validate transactions out of order within a block

# Why the UTXO model scales better

## State change frequency

- Account state changes per transaction

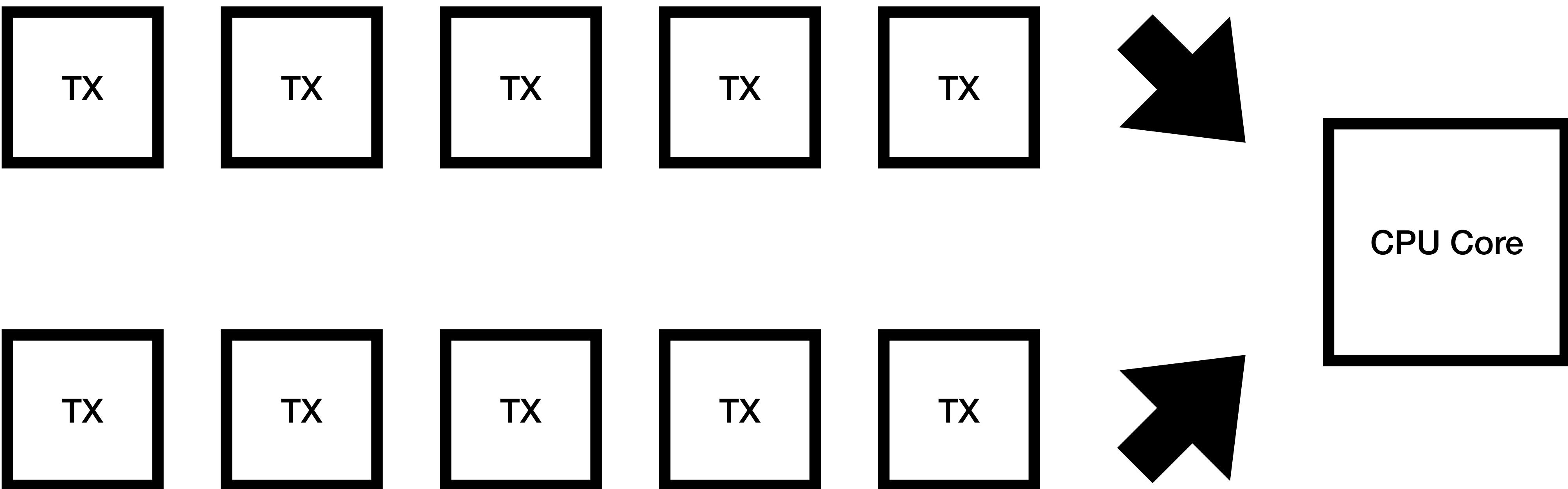
# Why the UTXO model scales better

## State change frequency

- Account state changes per transaction
- It's not safe to validate transactions out of order as the account state will change

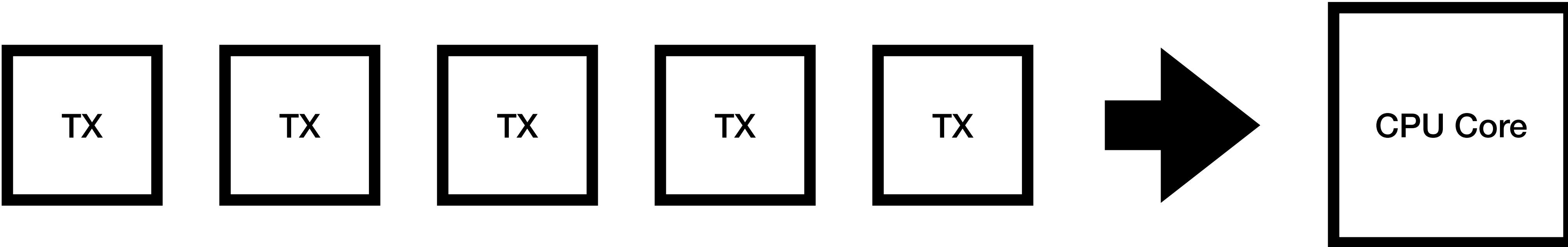
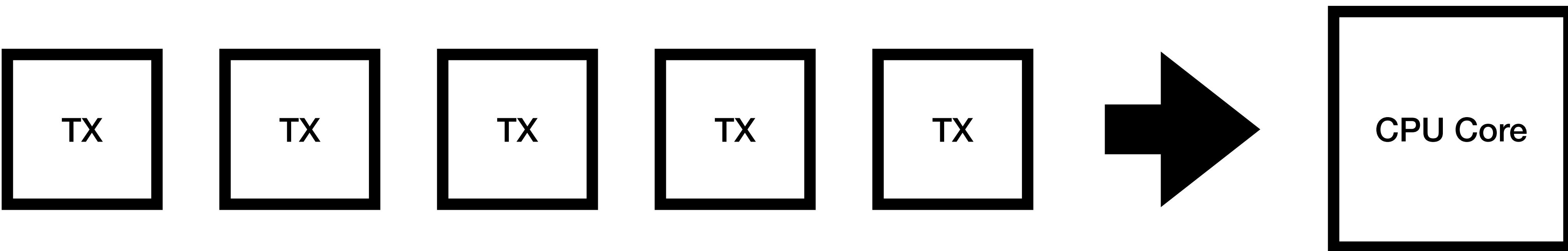
# UTXO model

## Visualization of validation of 10 txs



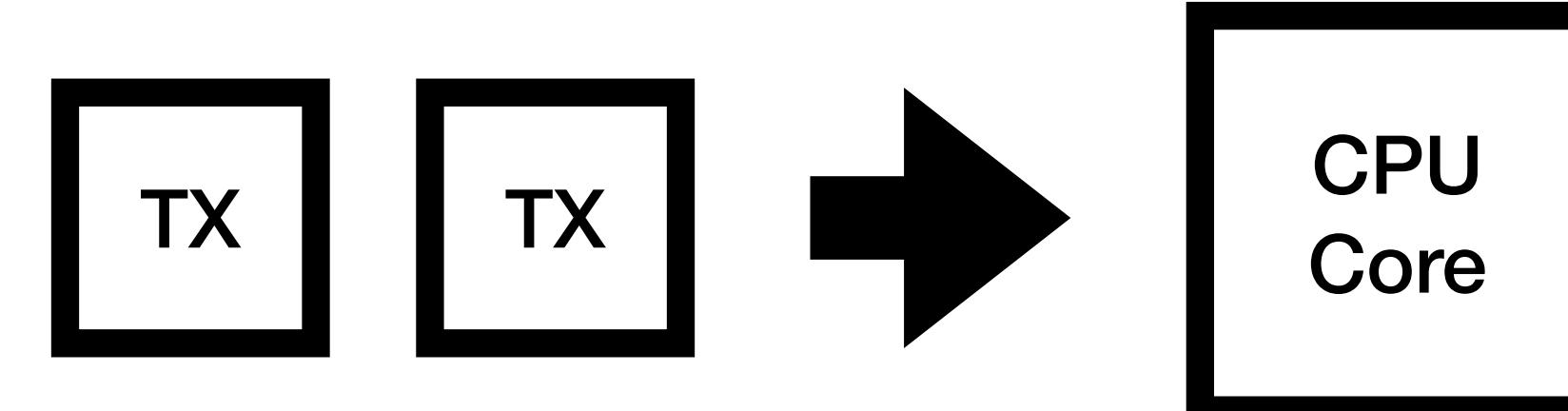
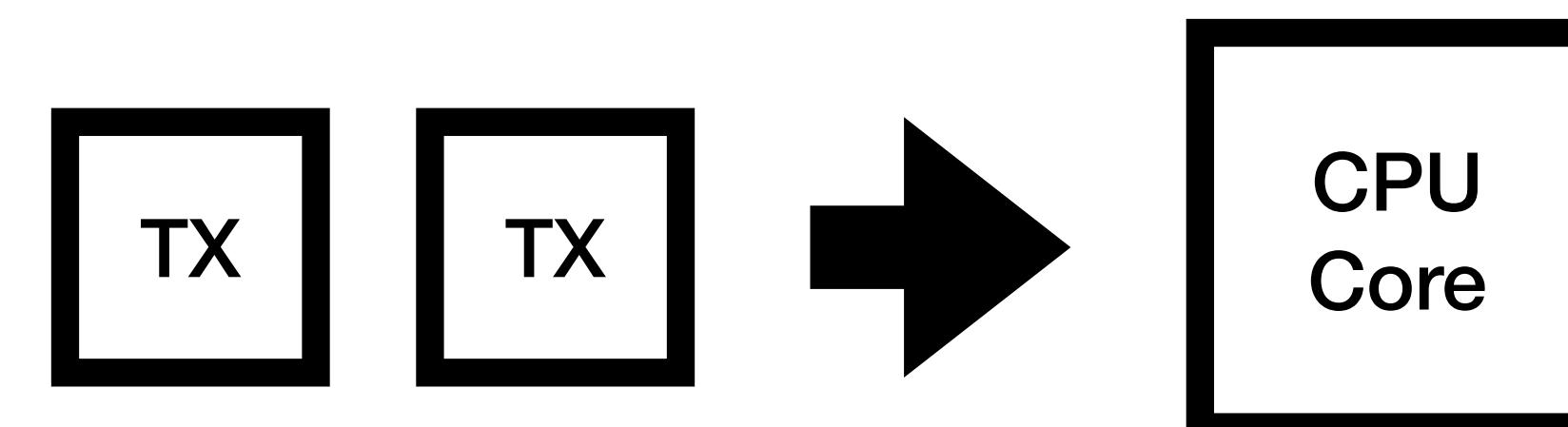
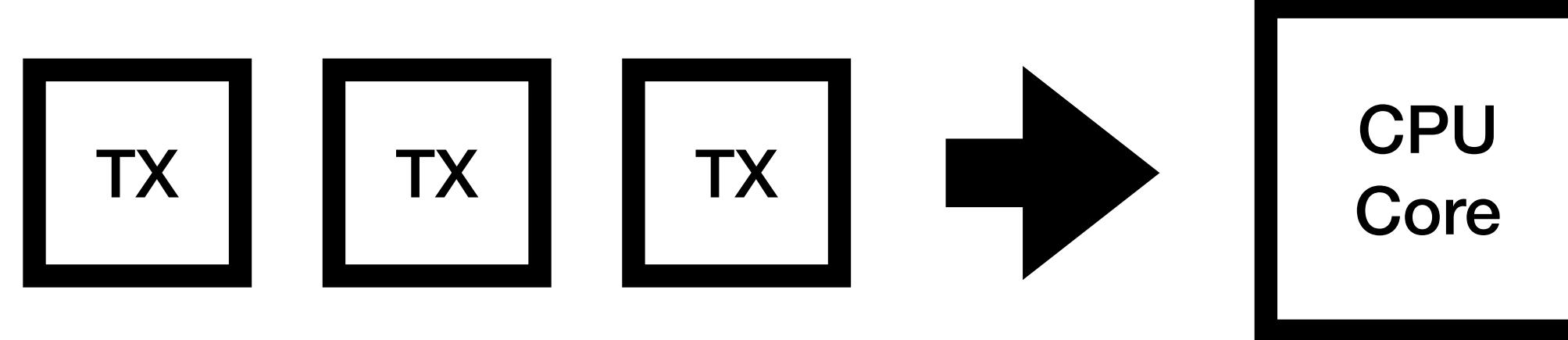
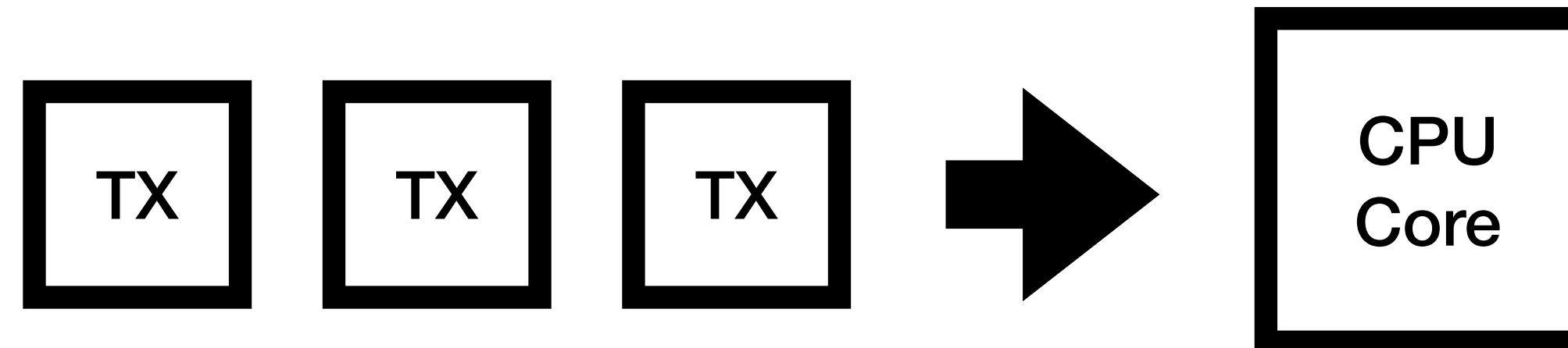
# UTXO model

## Visualization of validation of 10 txs



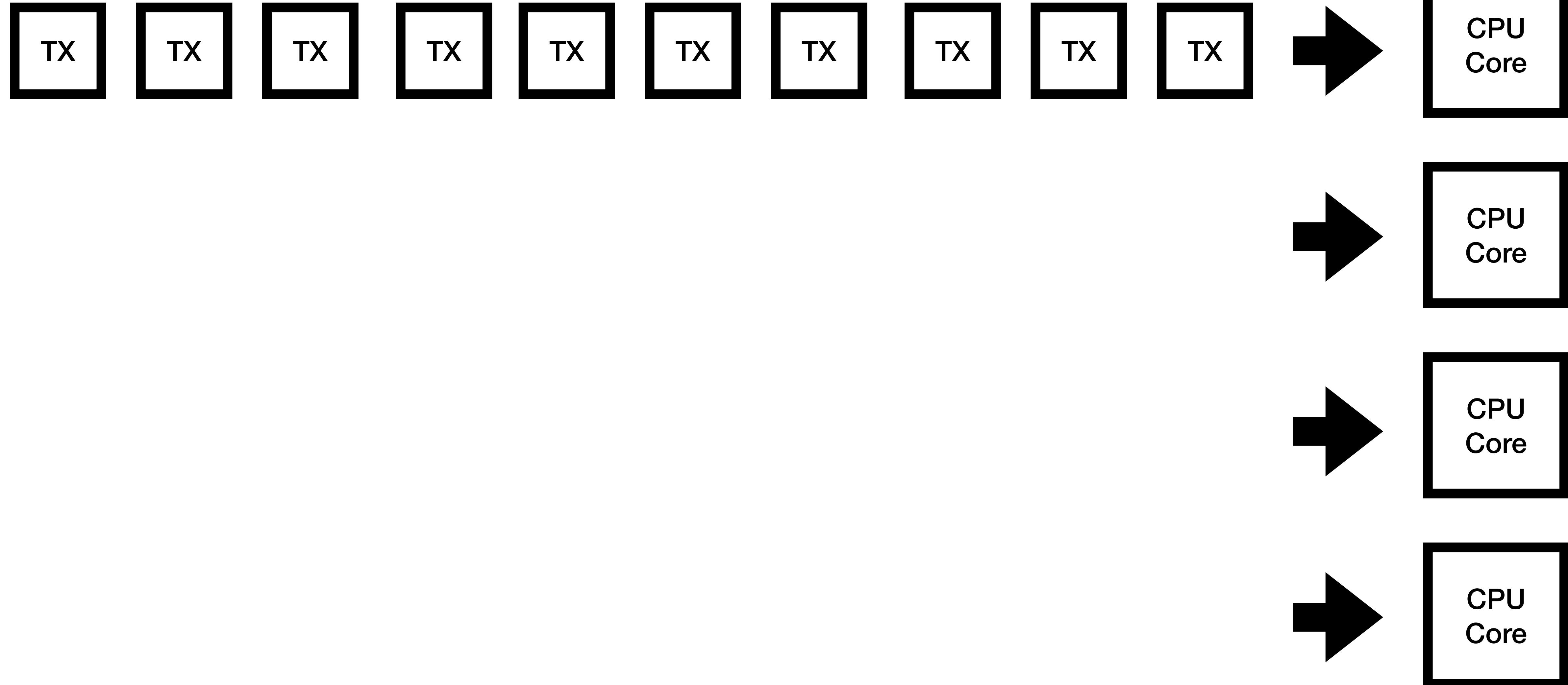
# UTXO model

## Visualization of validation of 10 txs



# Account based model

## Visualization of validation of 10 txs

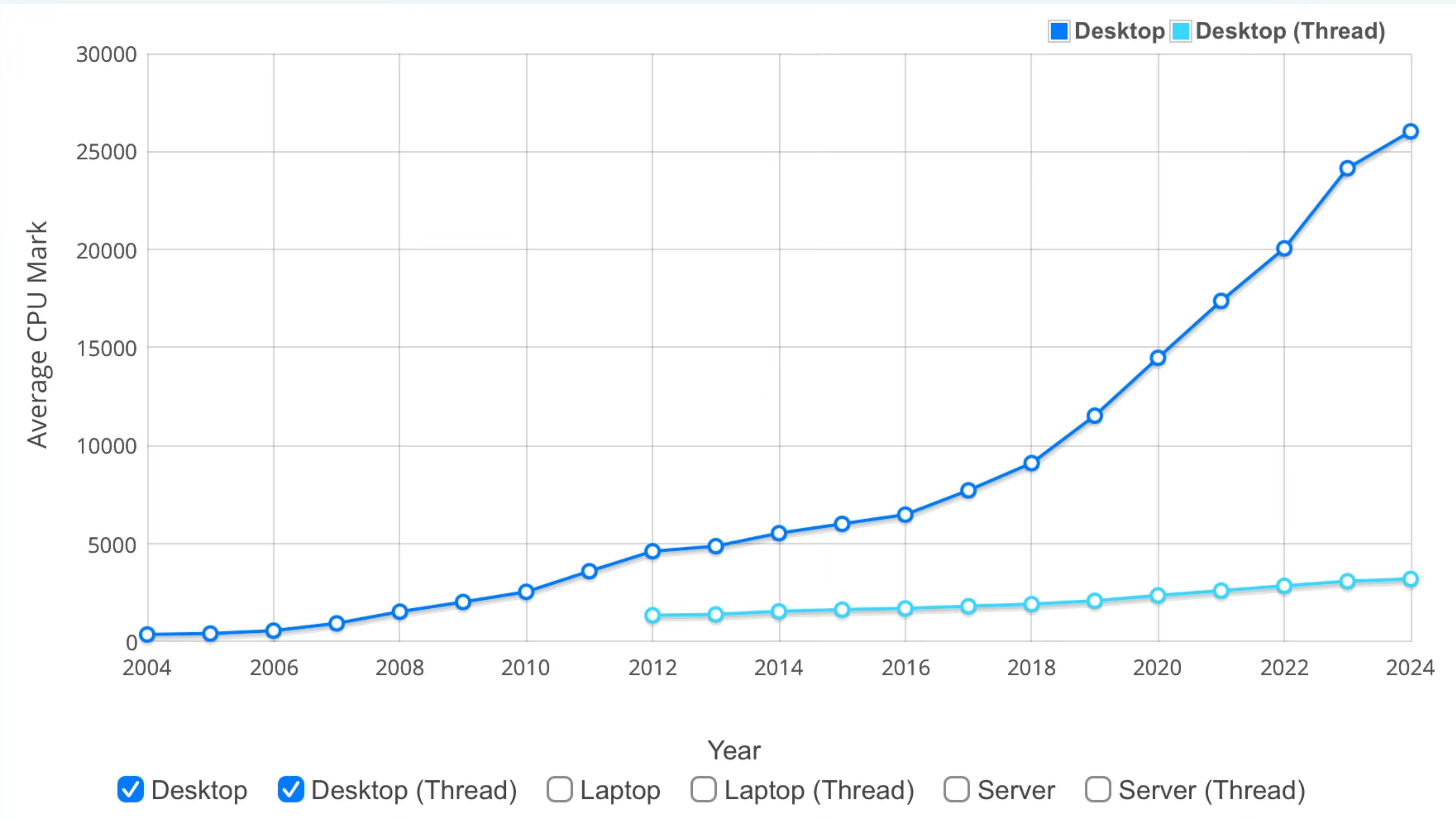




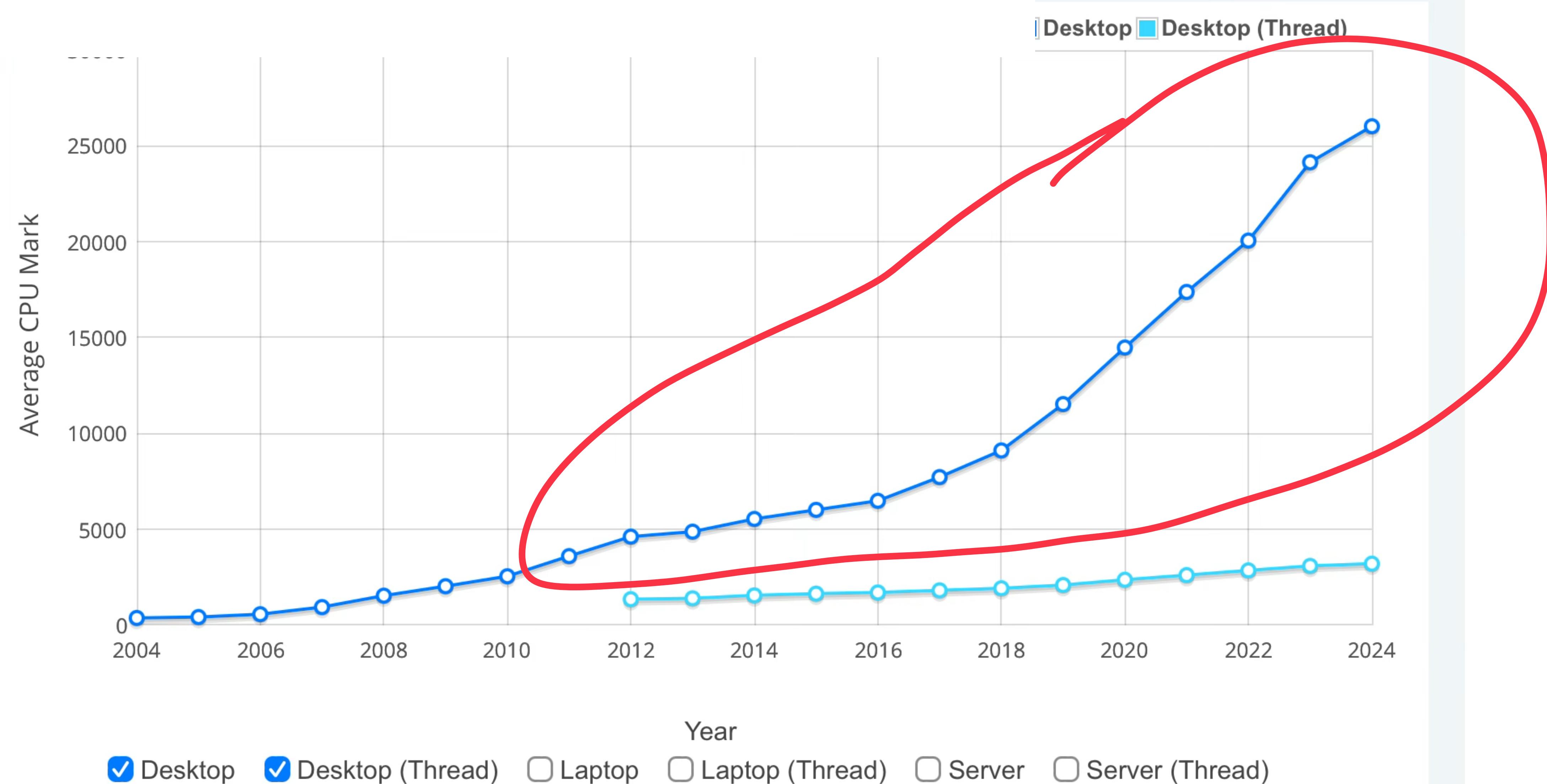


# Year on Year Performance

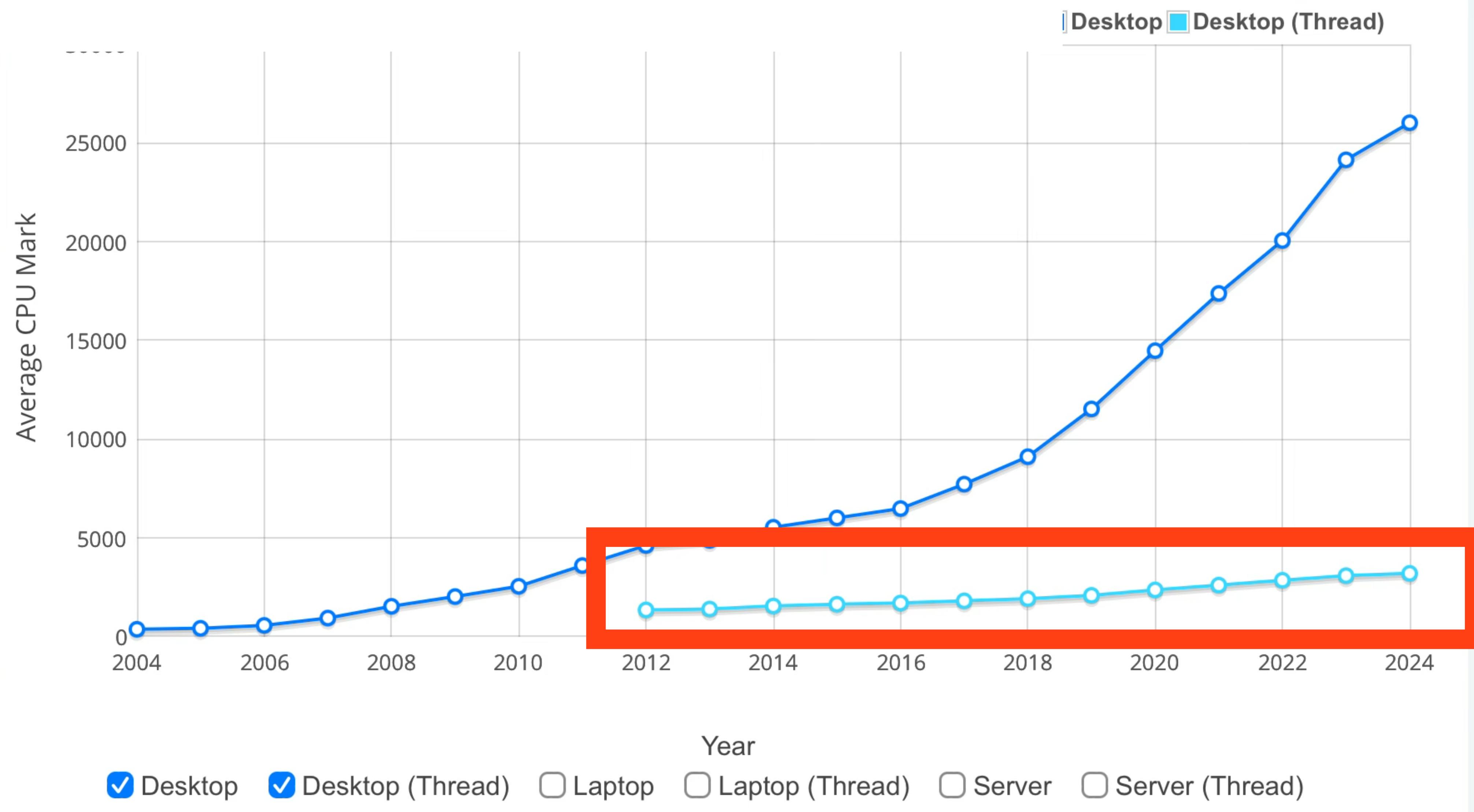
Updated 14th of September 2024



# Validation speed of a UTXO based model



# Validation speed of an account based model



# **Bottlenecks for block validation**

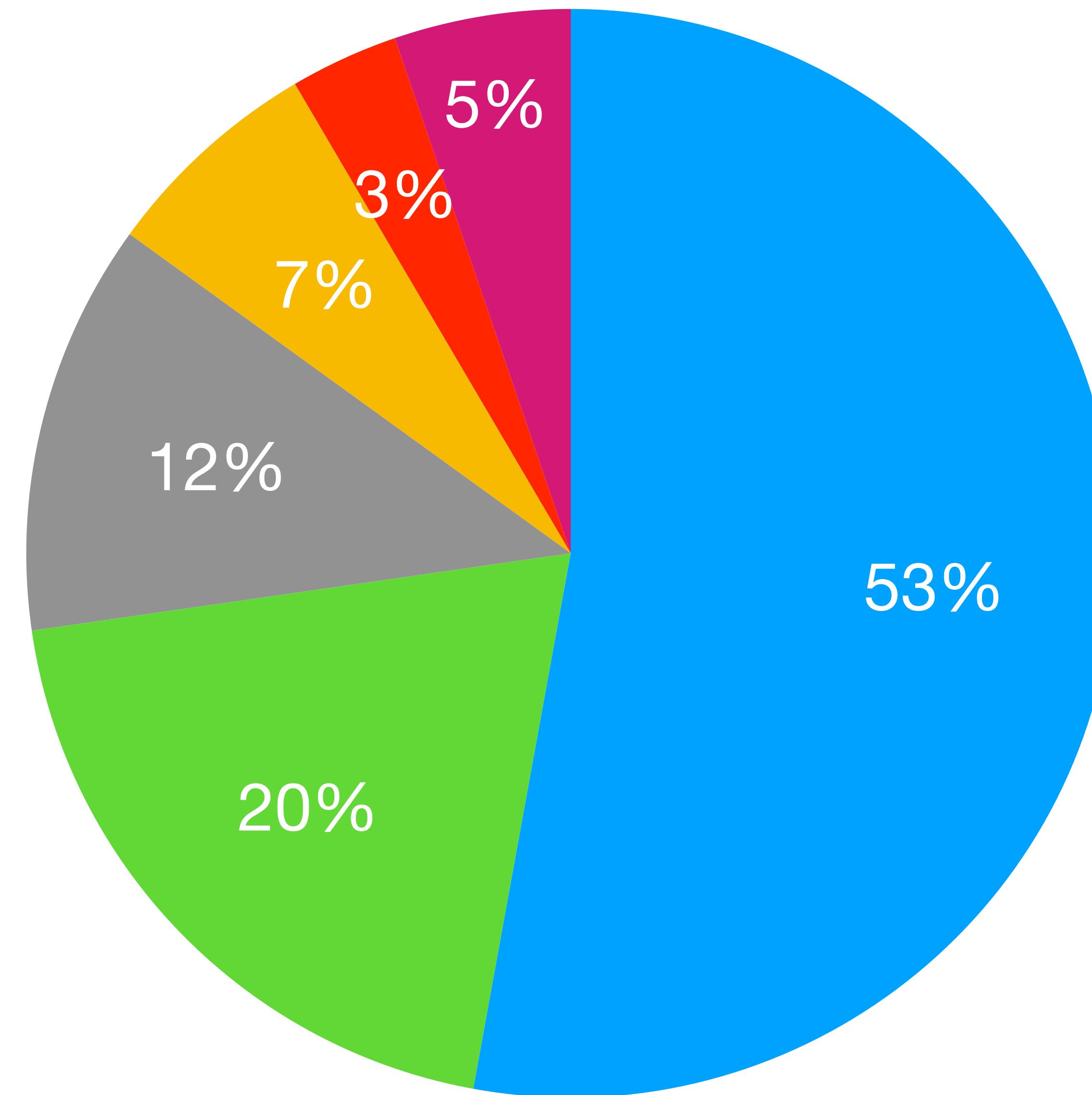
## **For problems for account based models**





# What Ethereum Classic spends time on

● database ● headers verification ● go runtime ● eip155 ● hashing ● etc

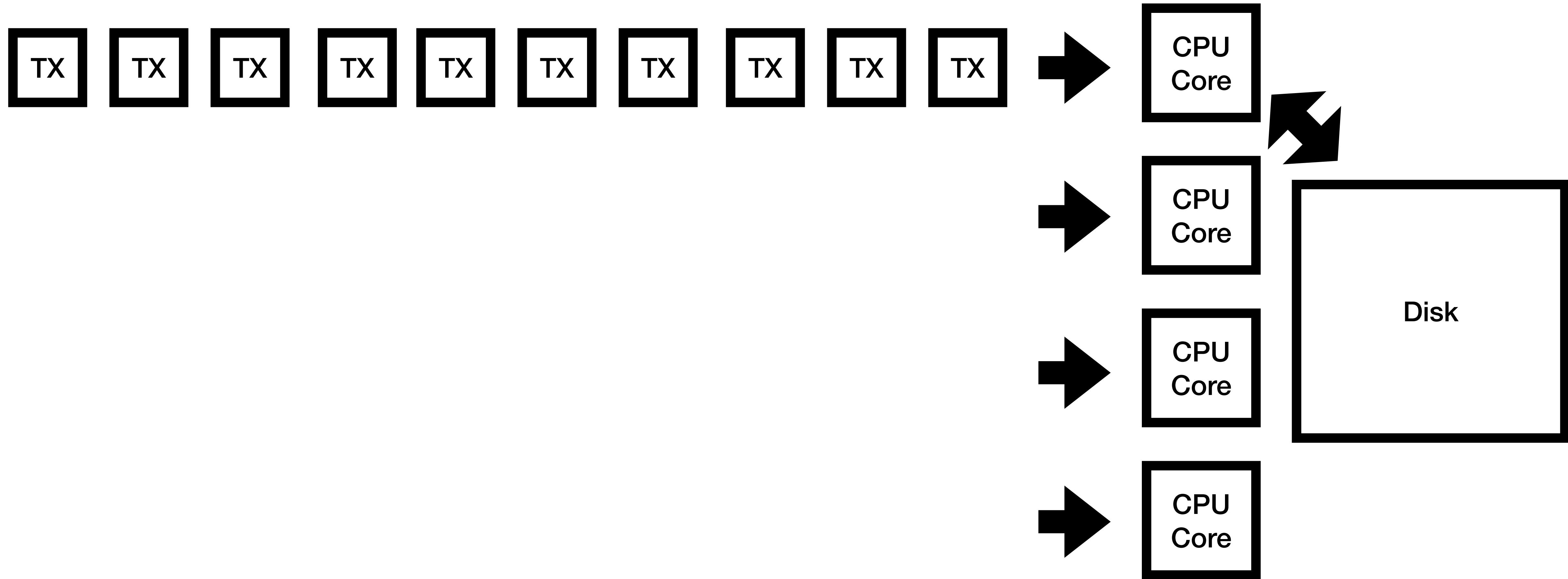


Core-geth v1.12.20

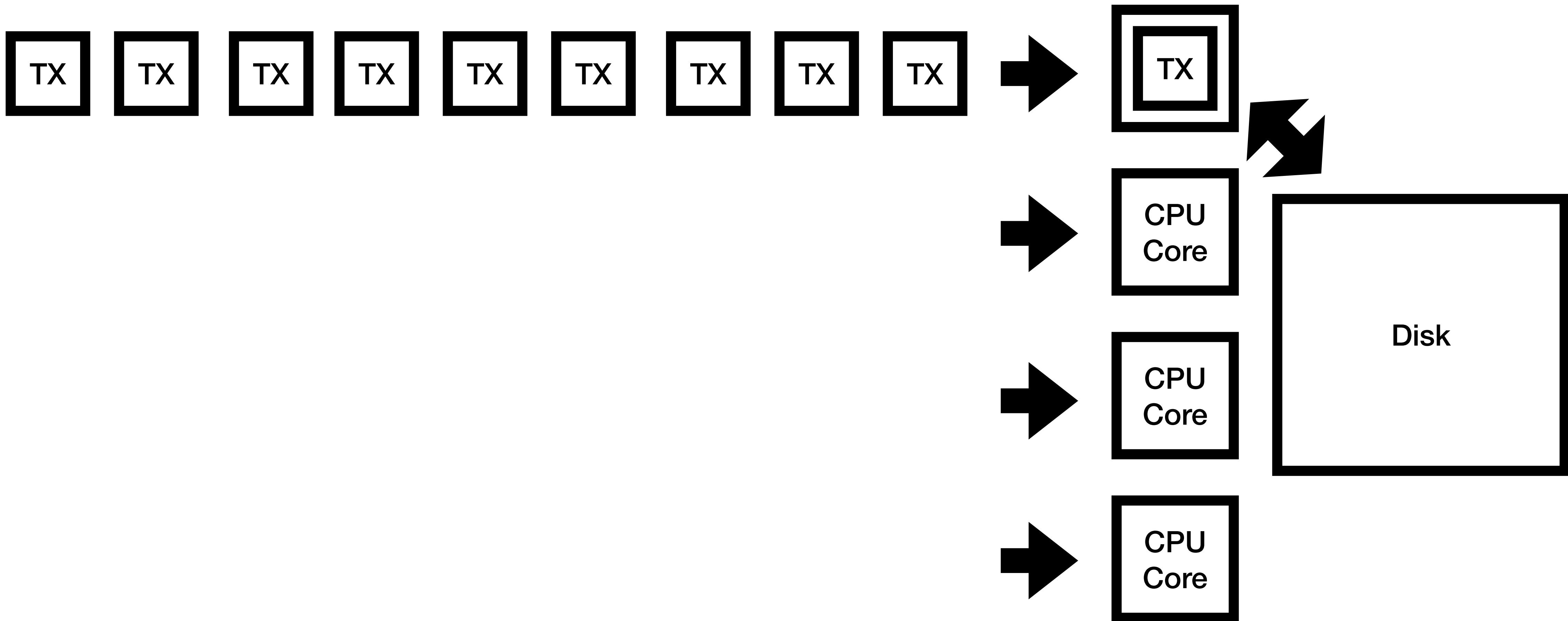
**53%**

**How much Ethereum Classic spends on disk operations**

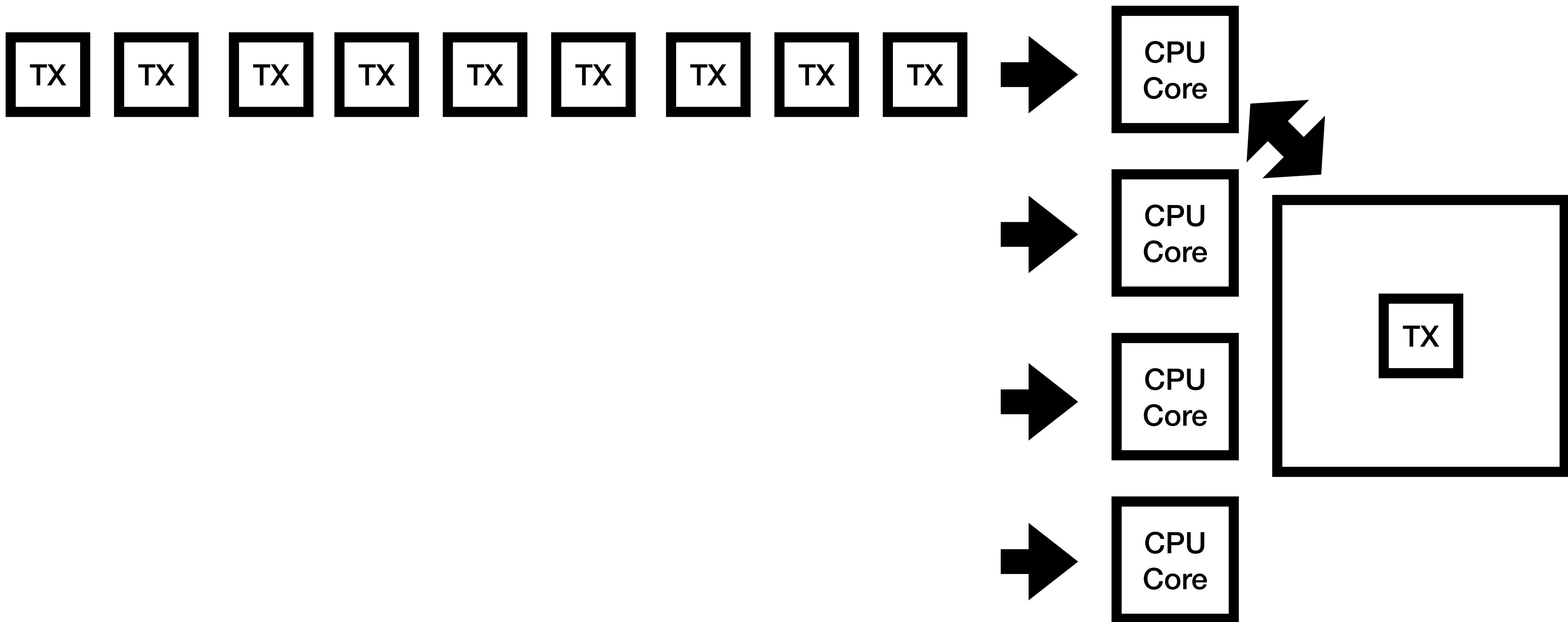
# Extremely simple visualization



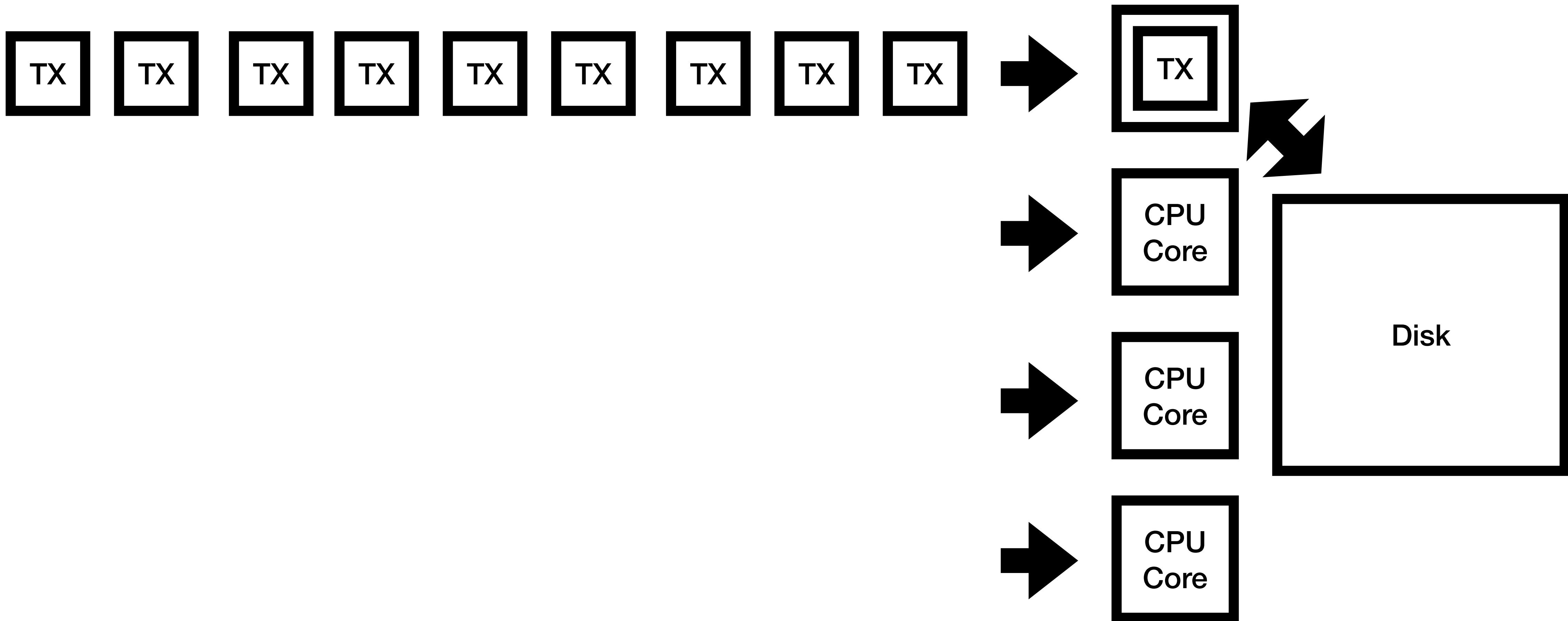
# Extremely simple visualization



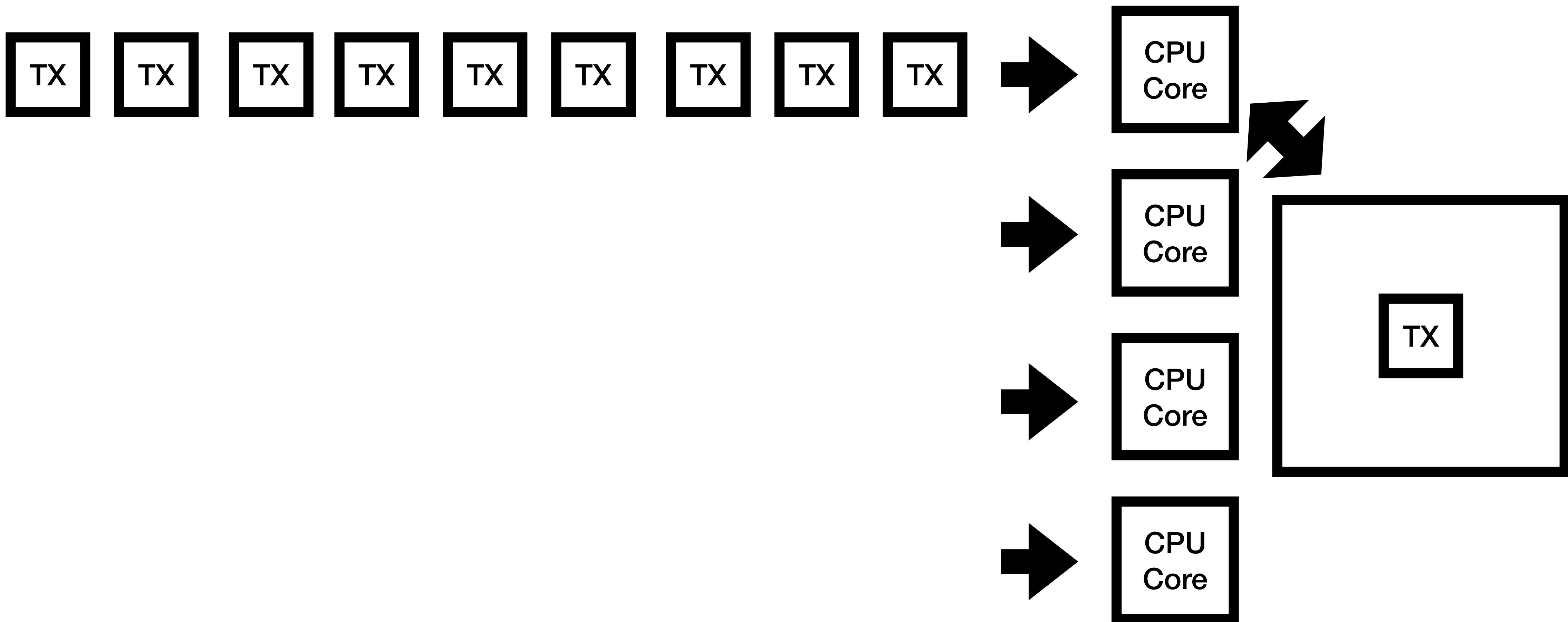
# Extremely simple visualization



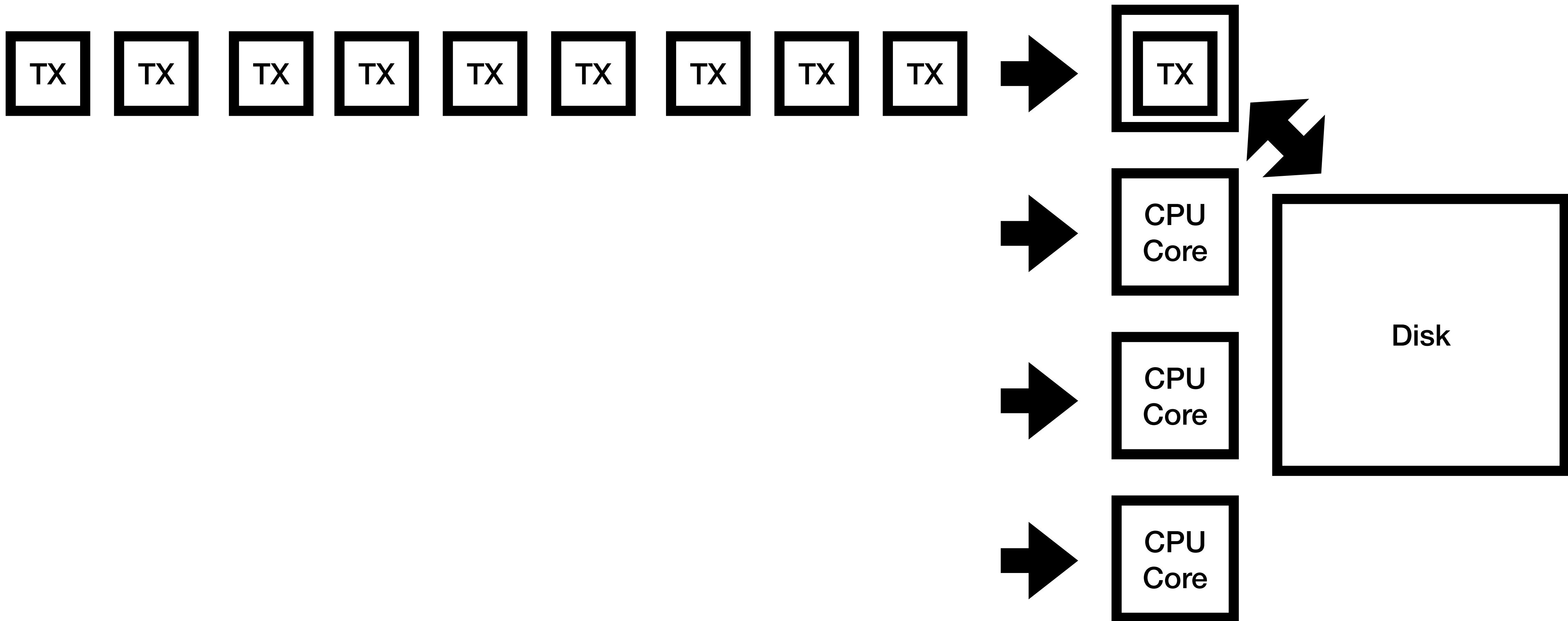
# Extremely simple visualization



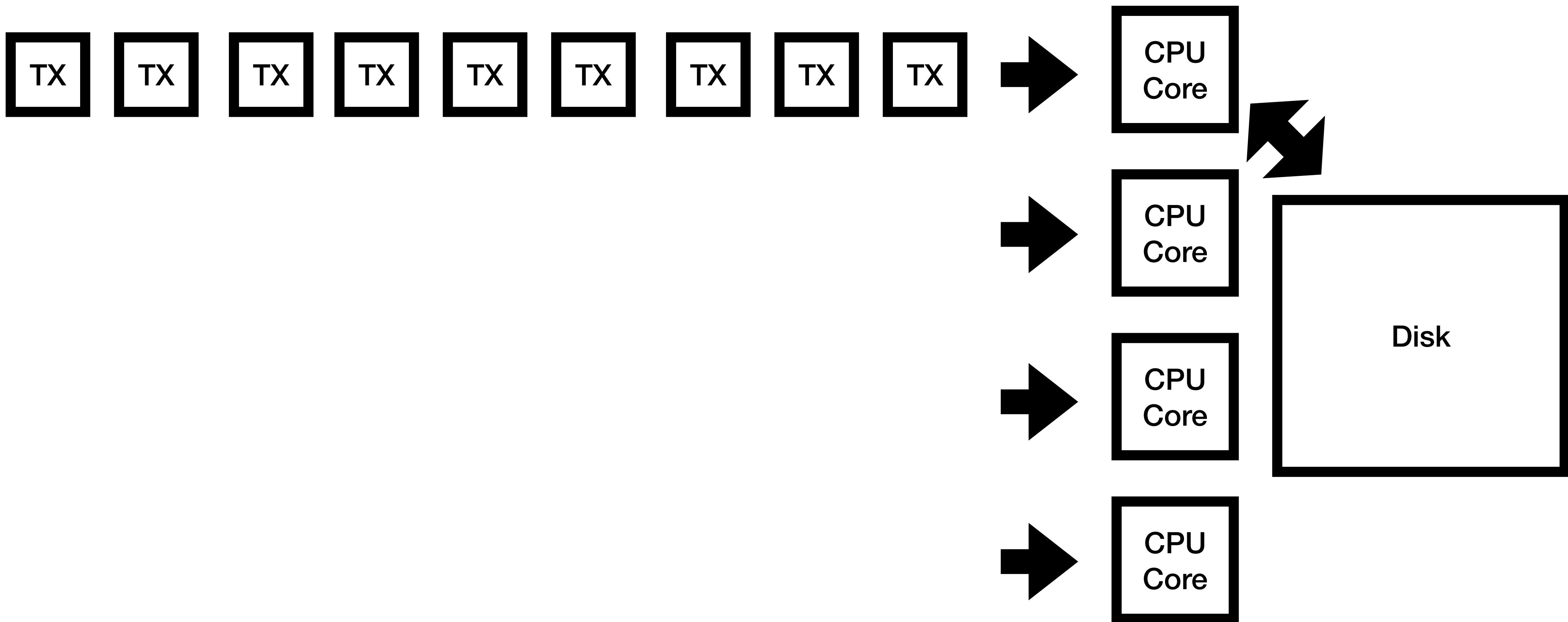
# Extremely simple visualization



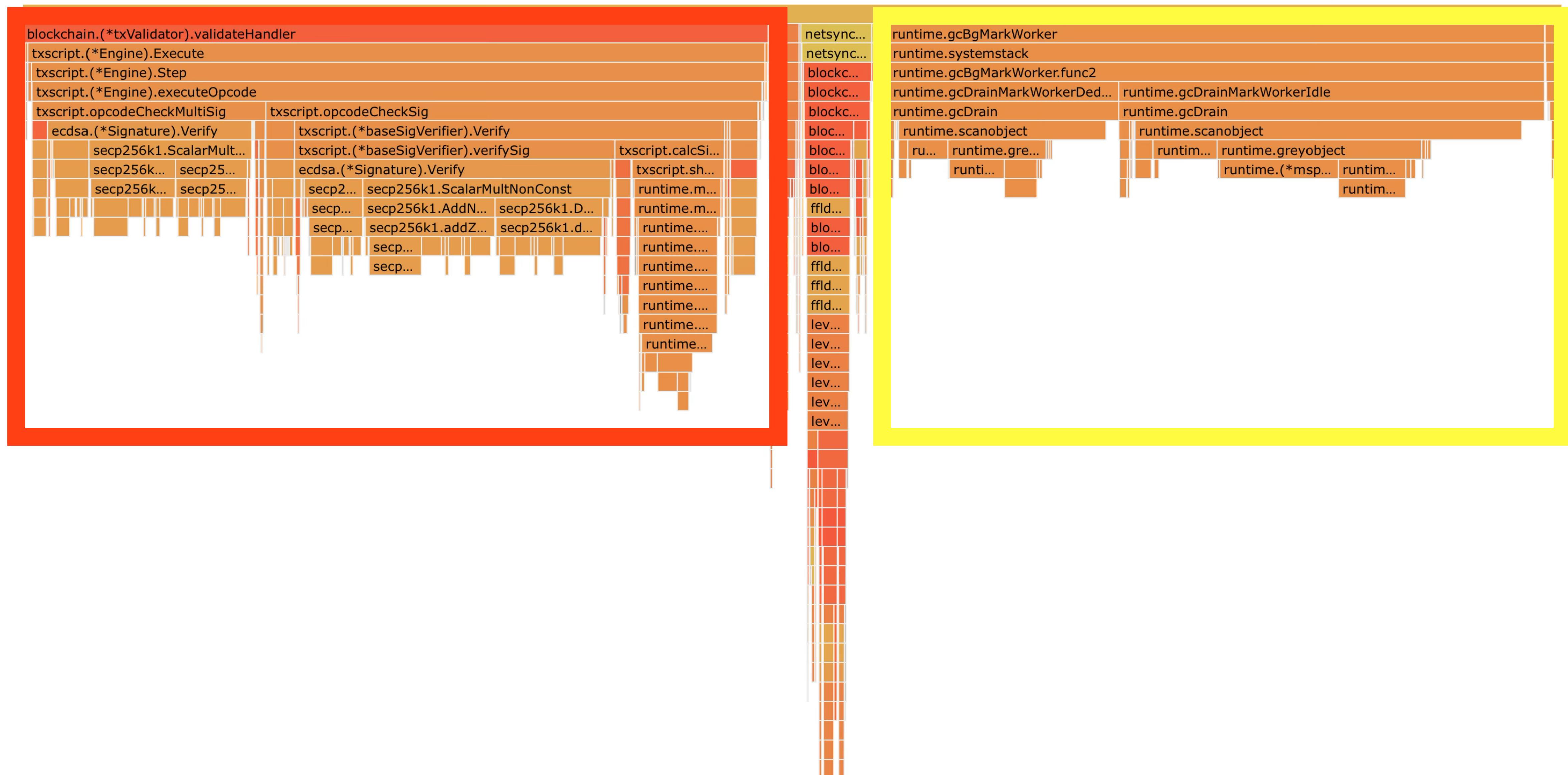
# Extremely simple visualization



# Extremely simple visualization

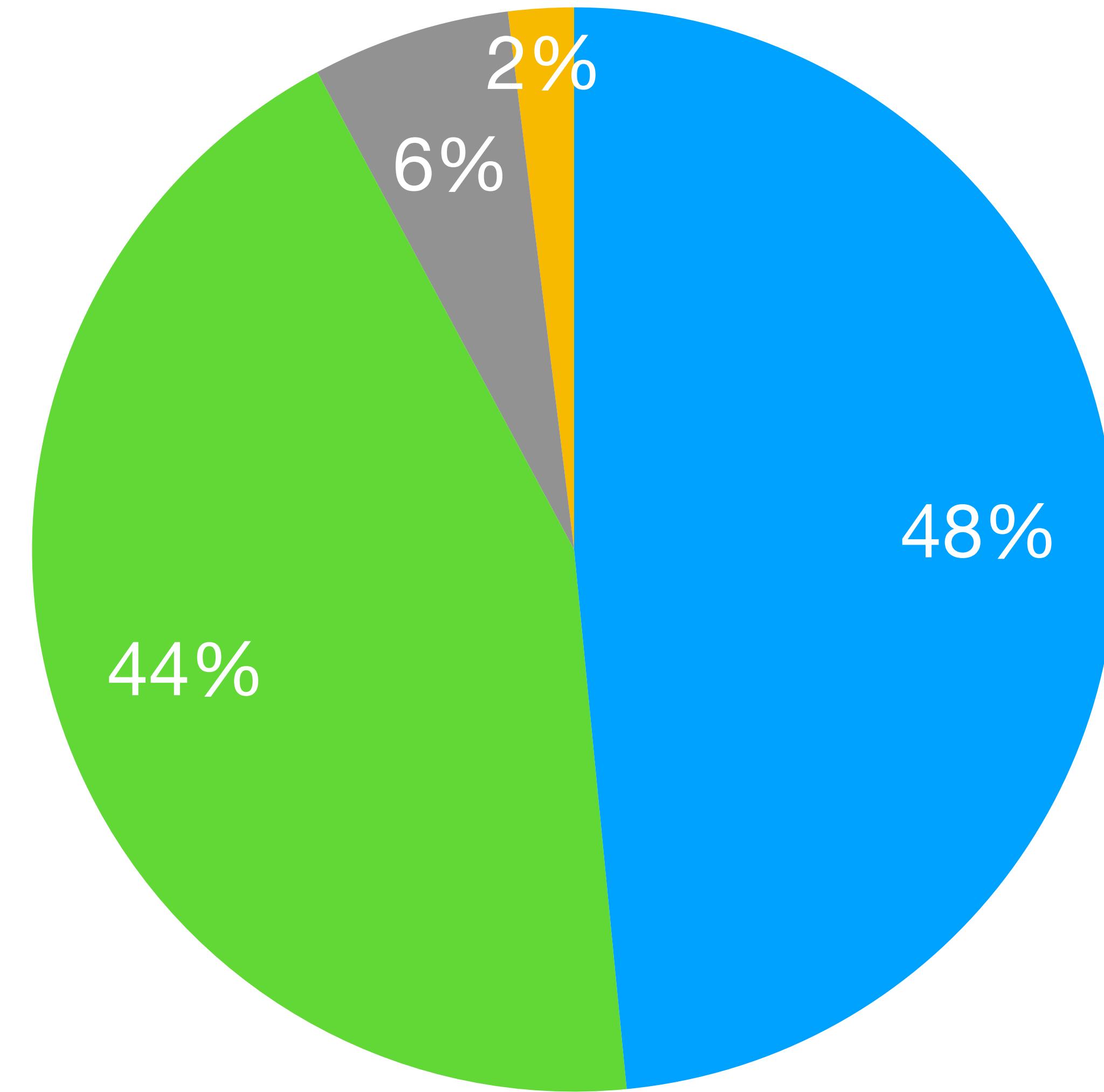






# What Bitcoin spends time on

- Signature verification
- Go runtime
- Database
- etc

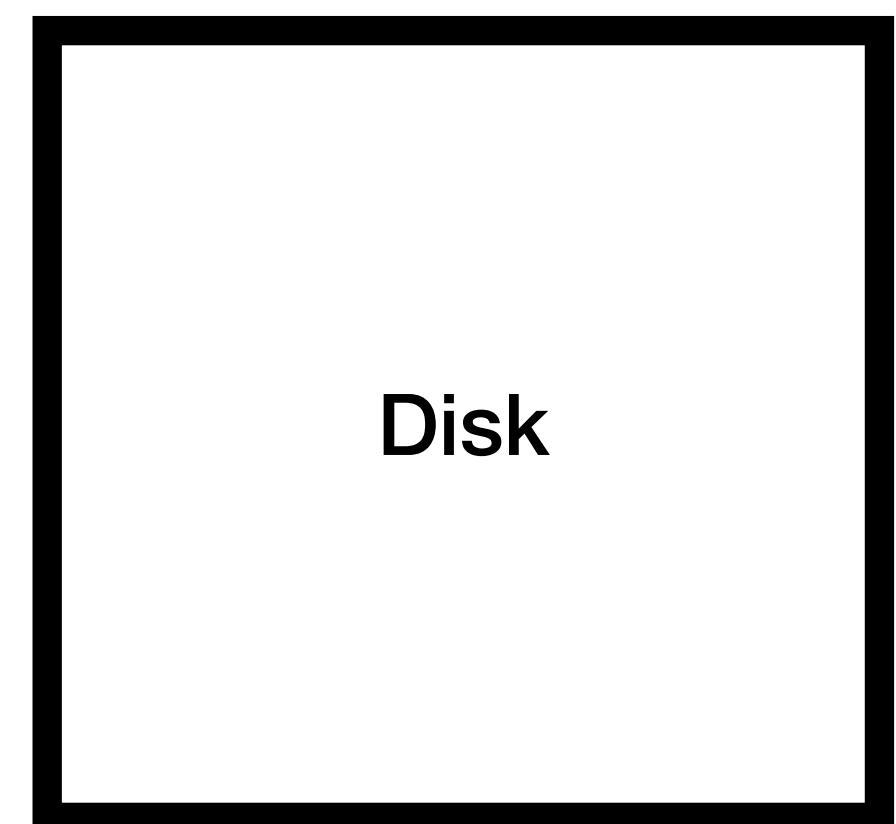
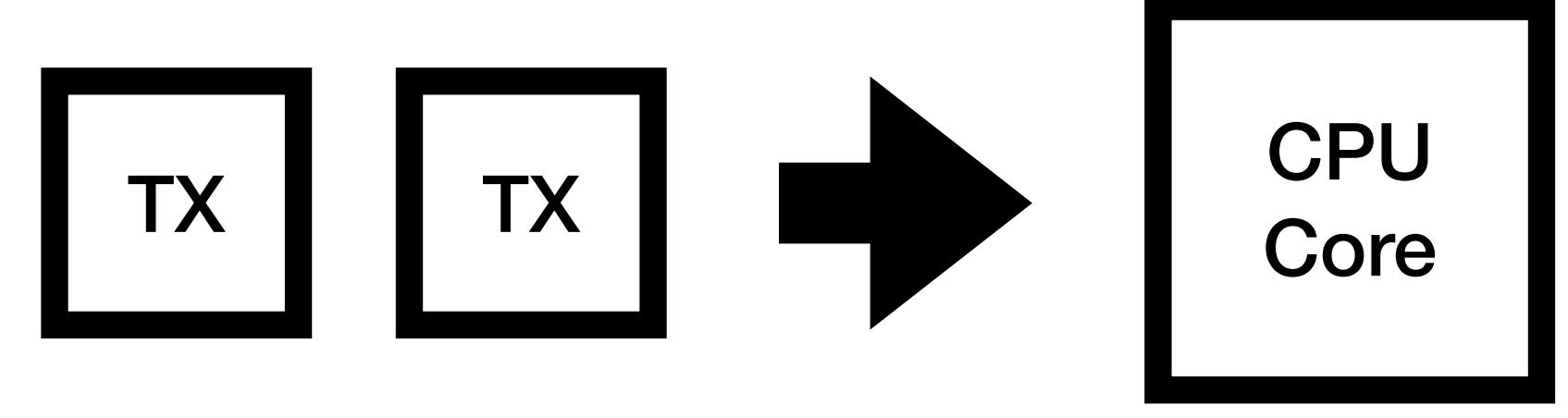
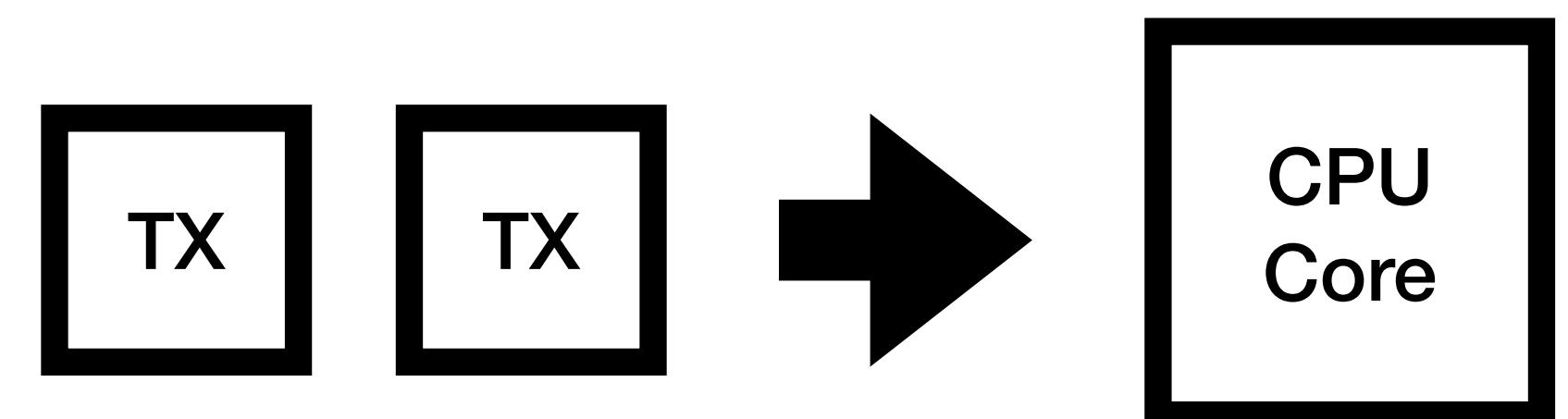
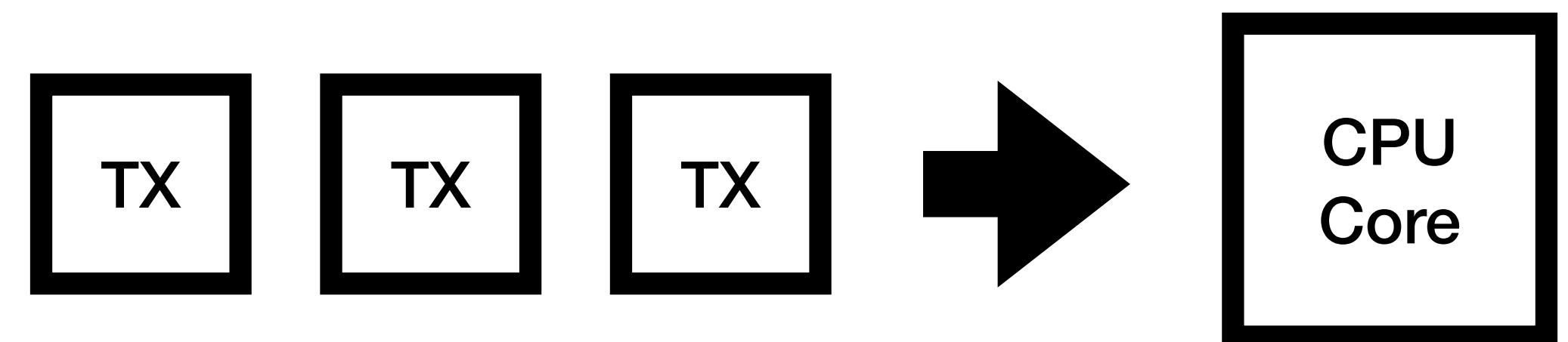
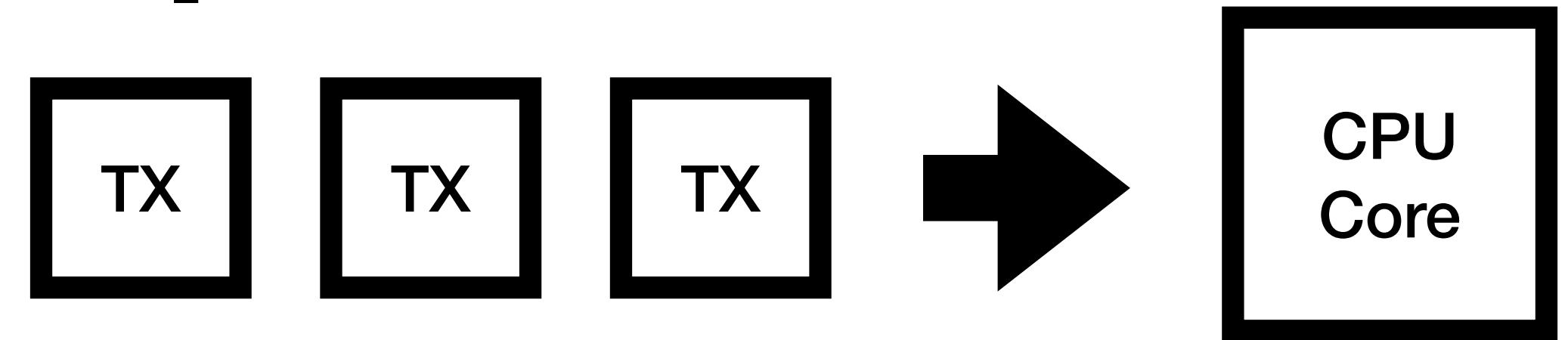


btcd v0.24.2-beta

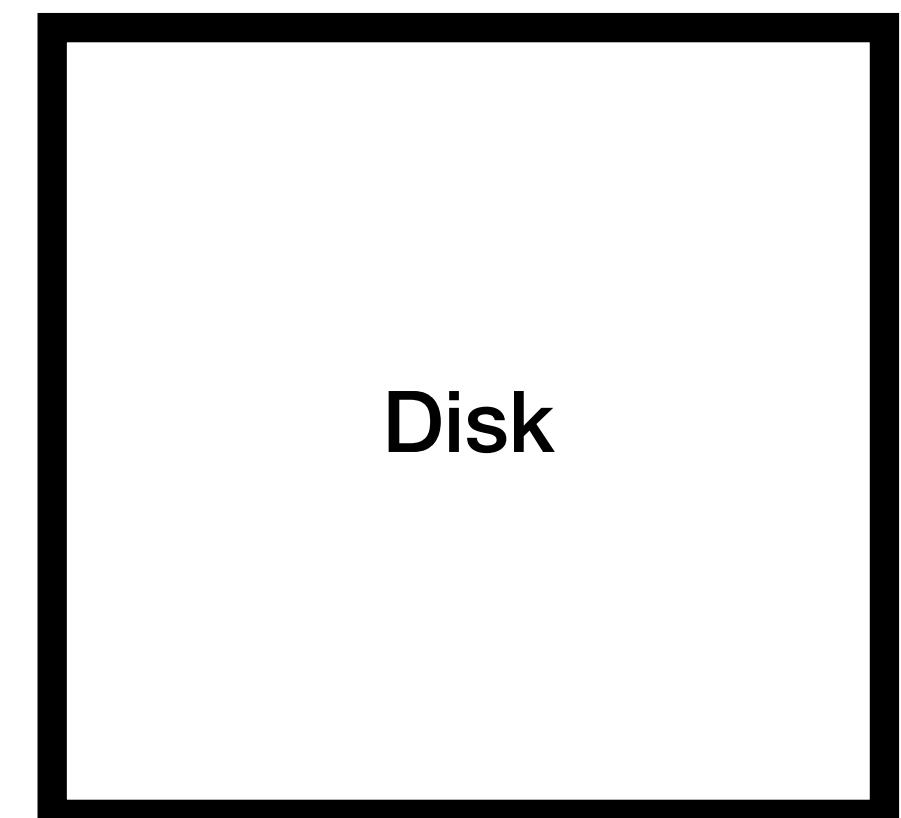
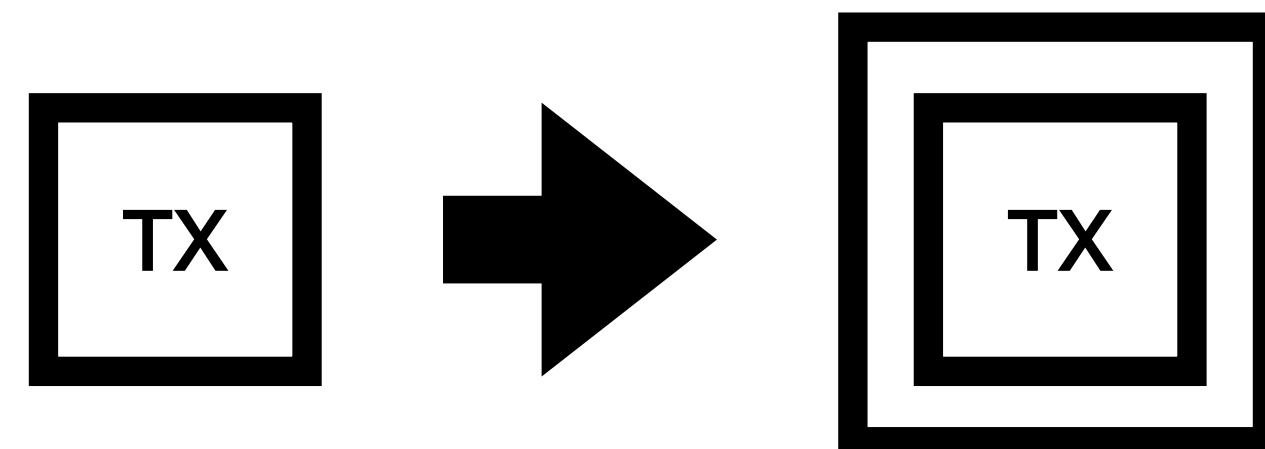
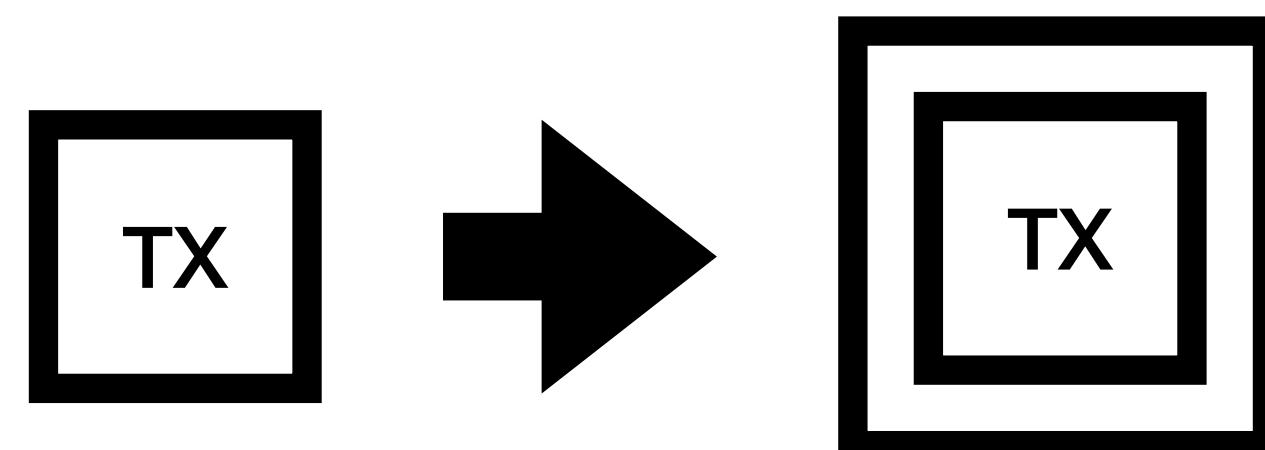
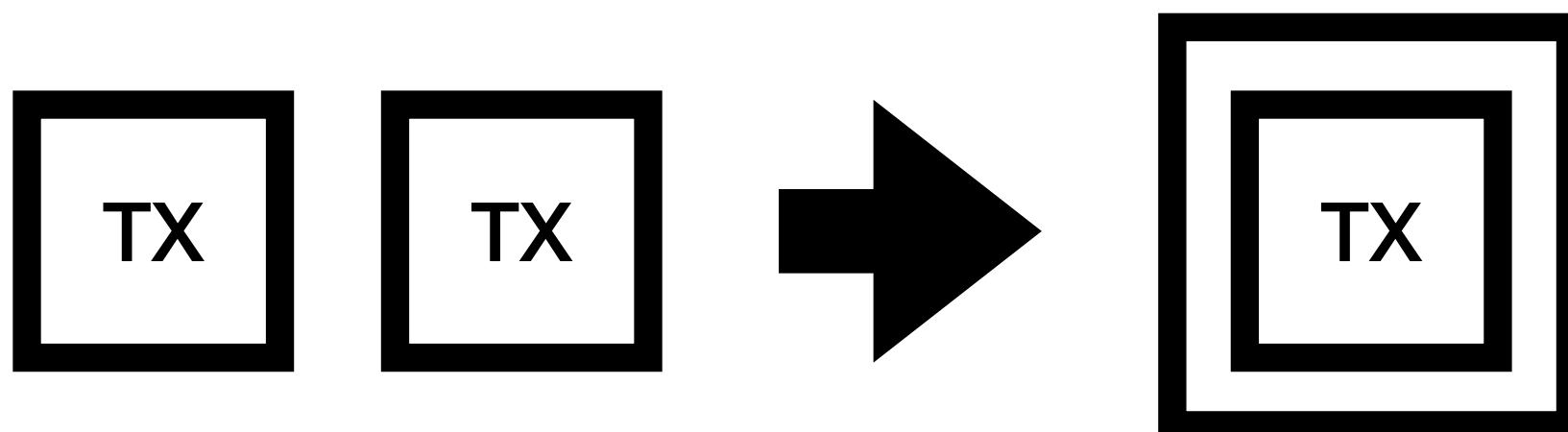
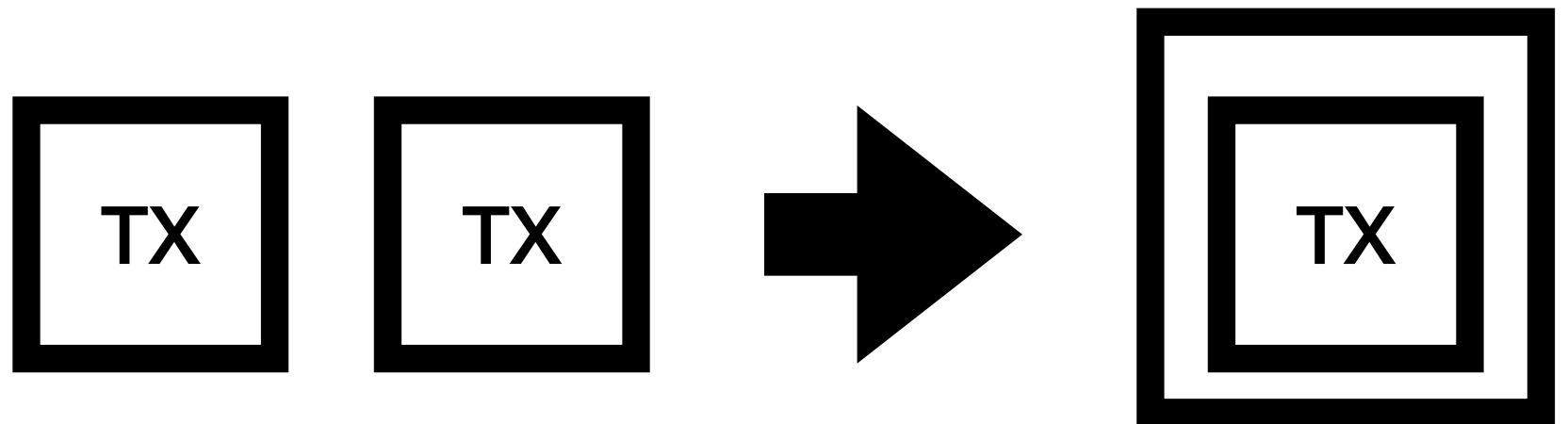
**92%**

**How much btcd (Bitcoin impl.) spends in CPU operations**

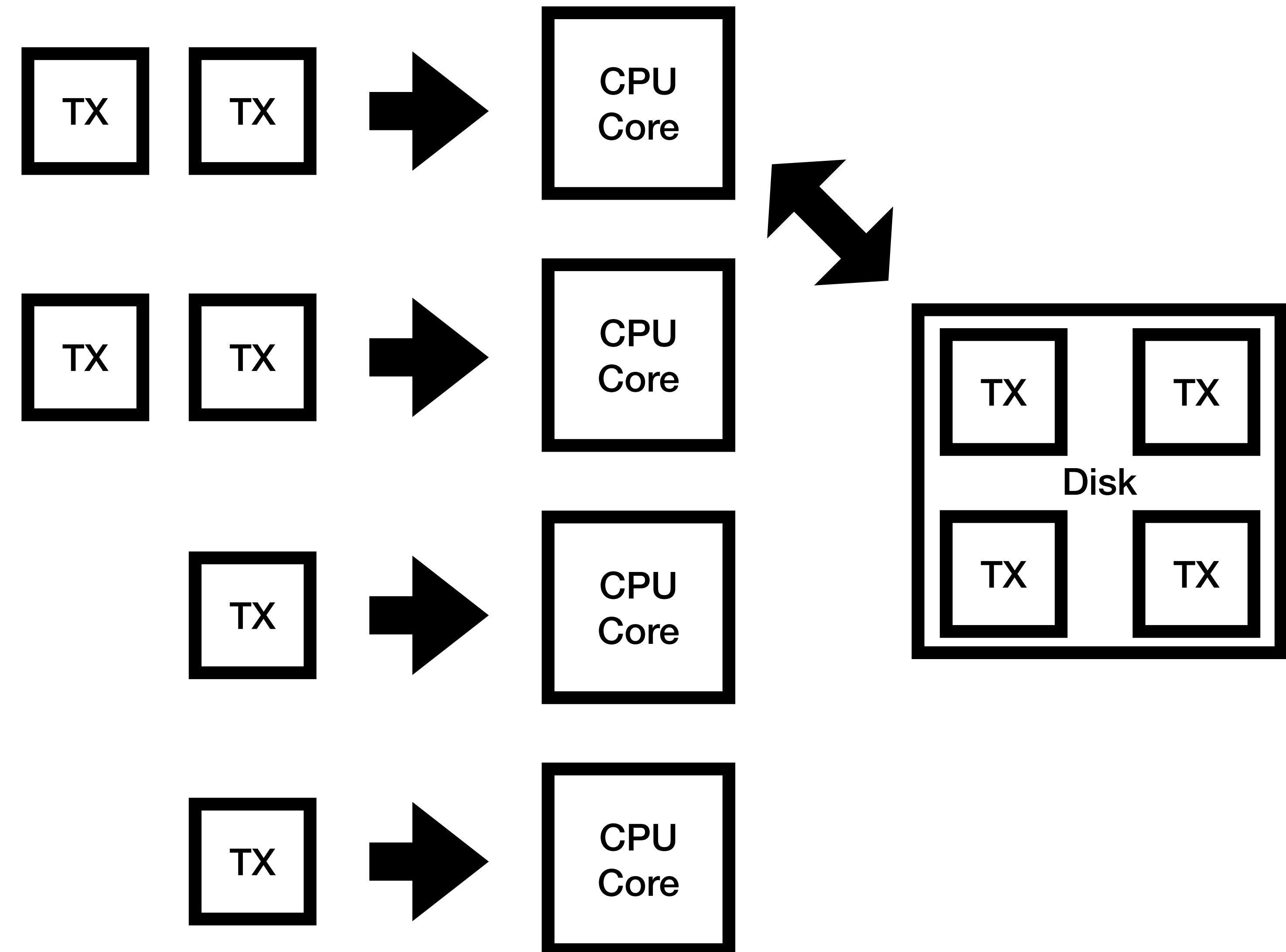
# Extremely simple visualization



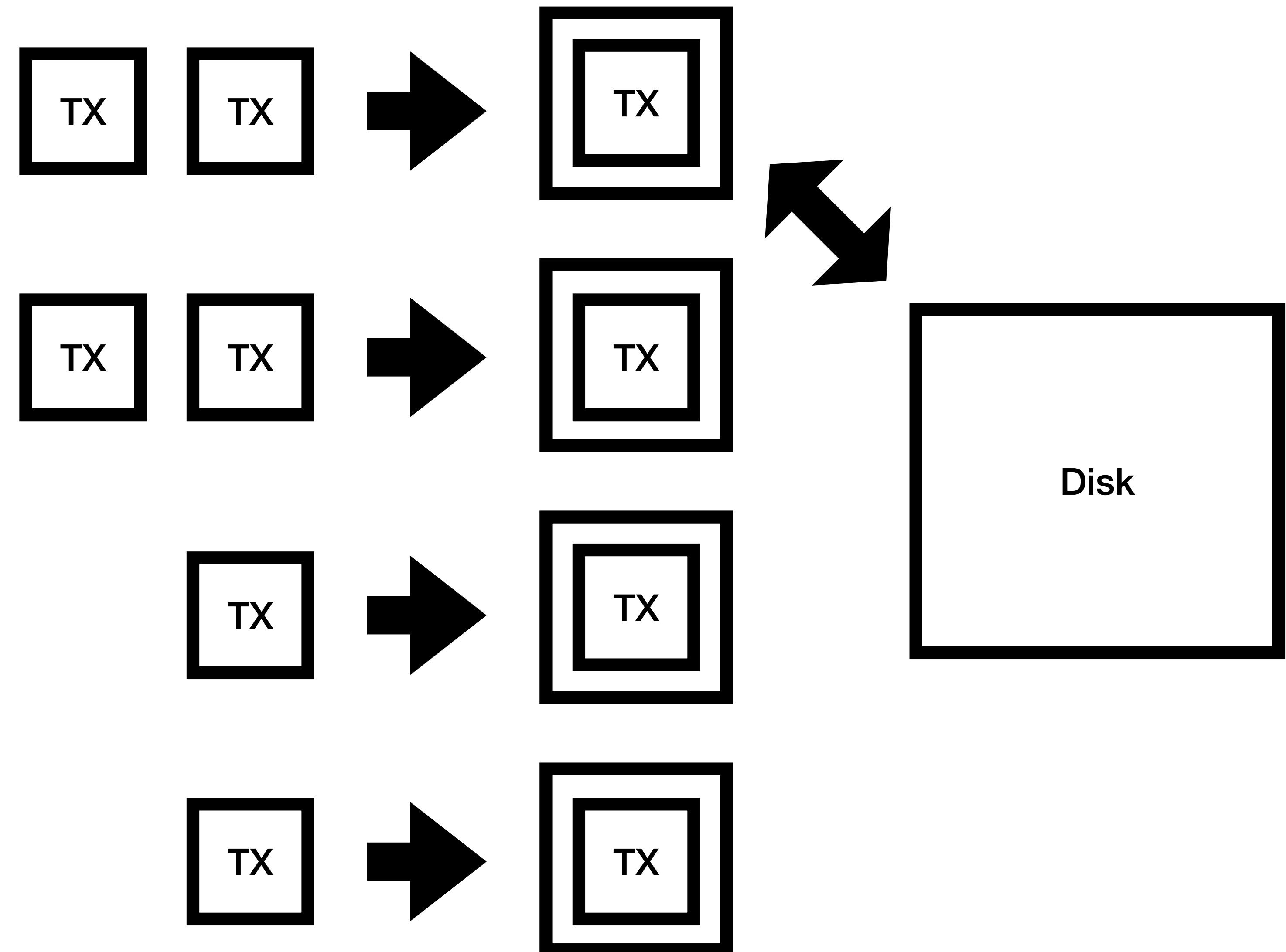
# Extremely simple visualization



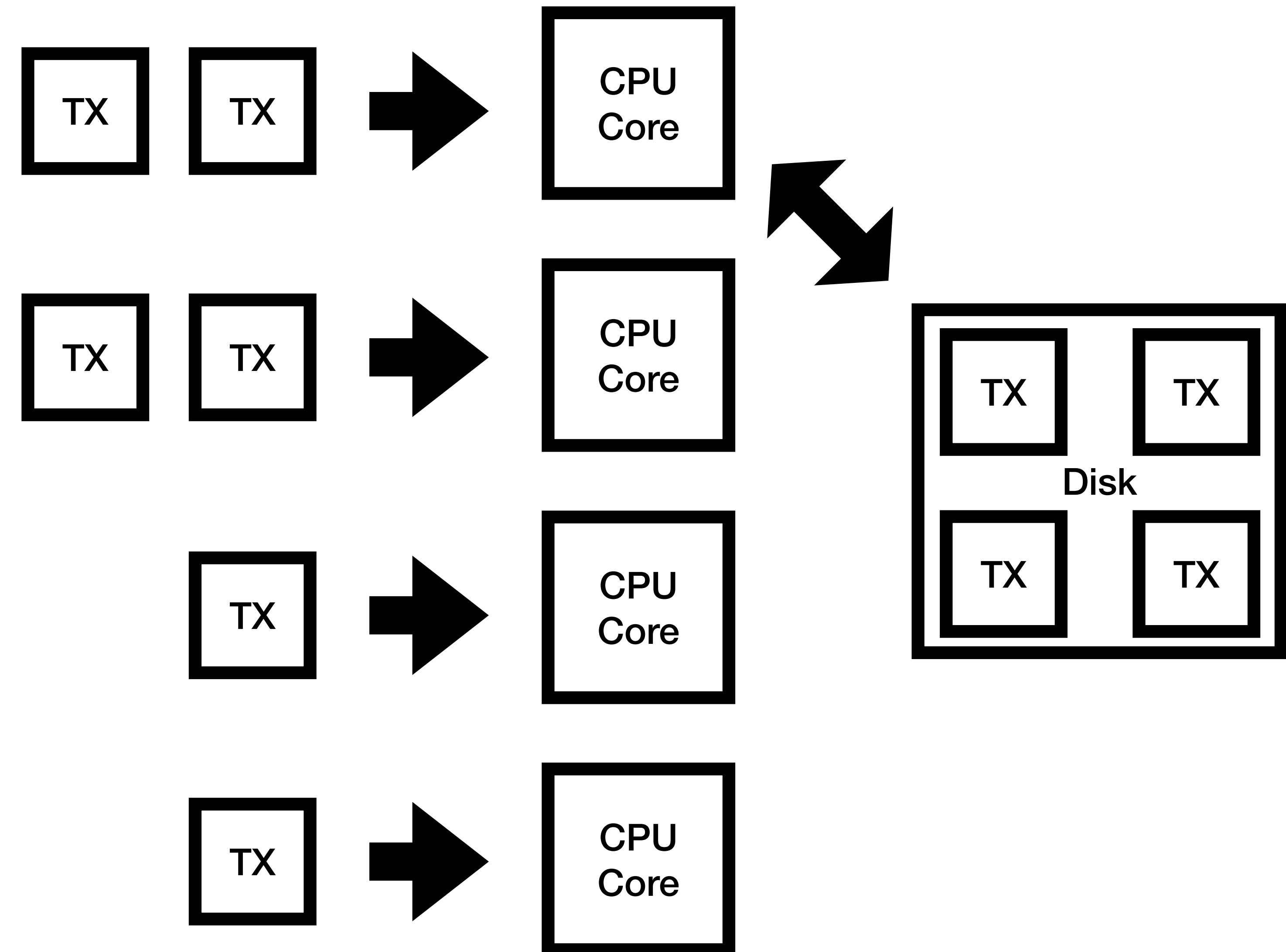
# Extremely simple visualization



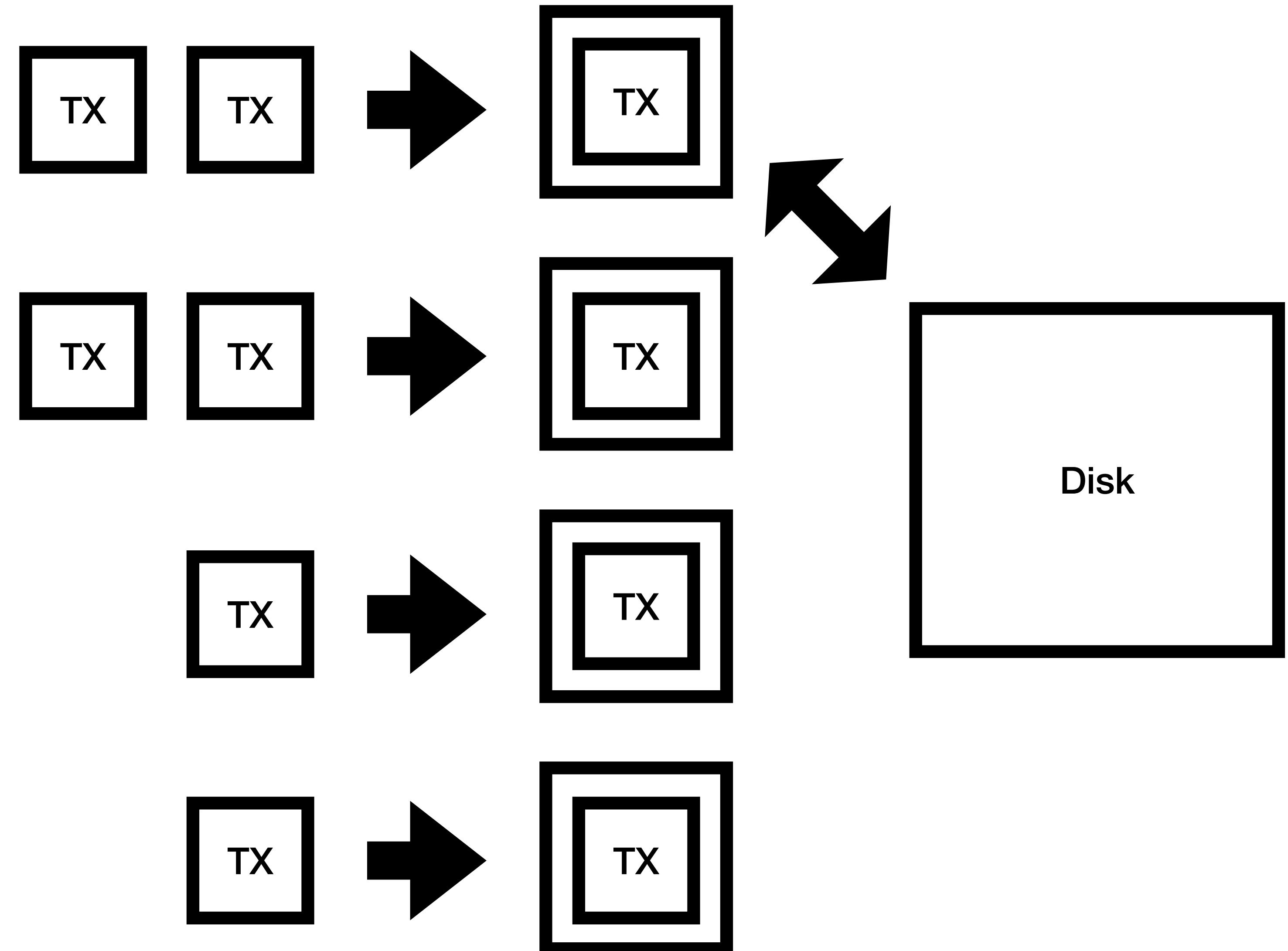
# Extremely simple visualization



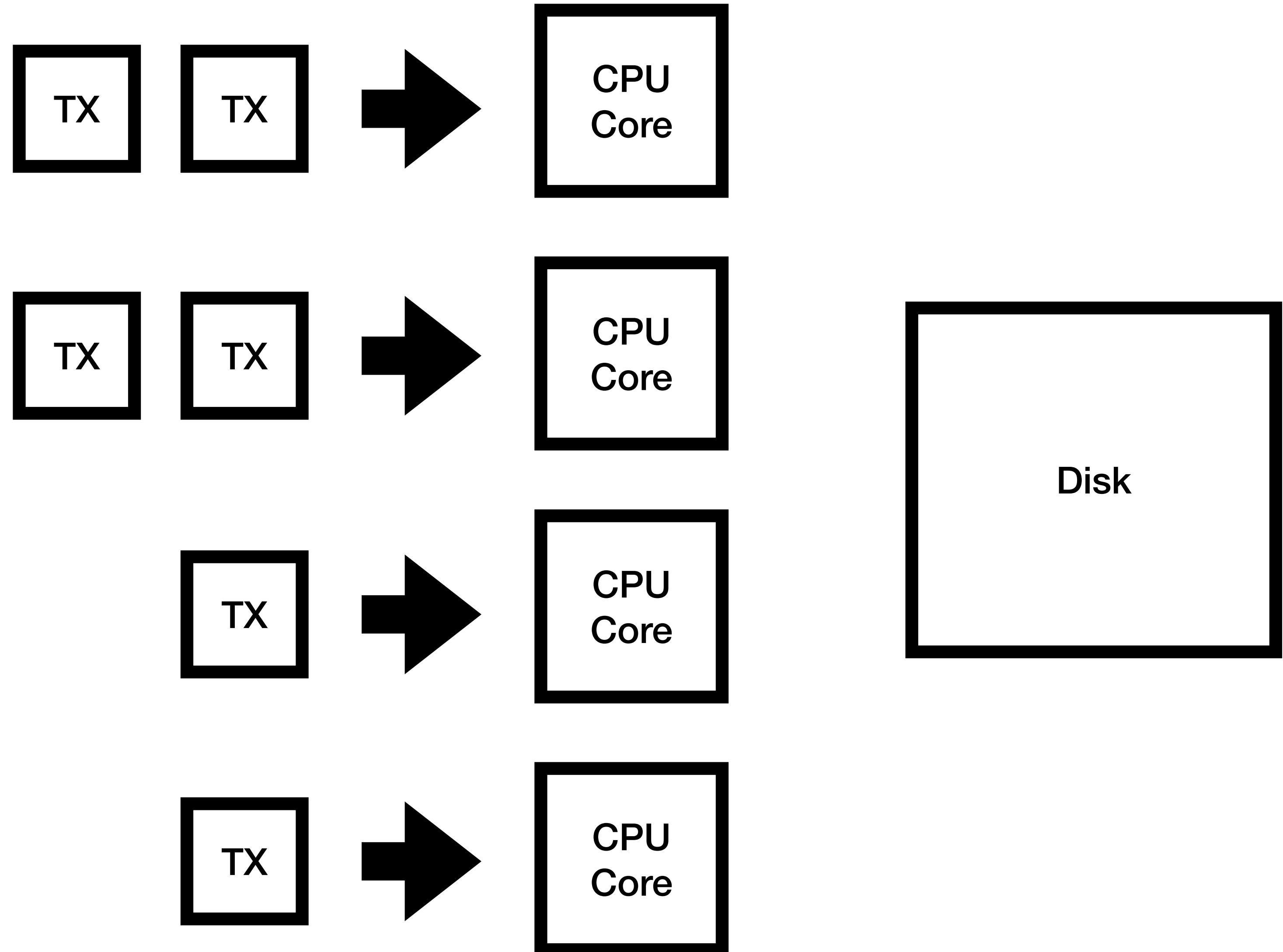
# Extremely simple visualization



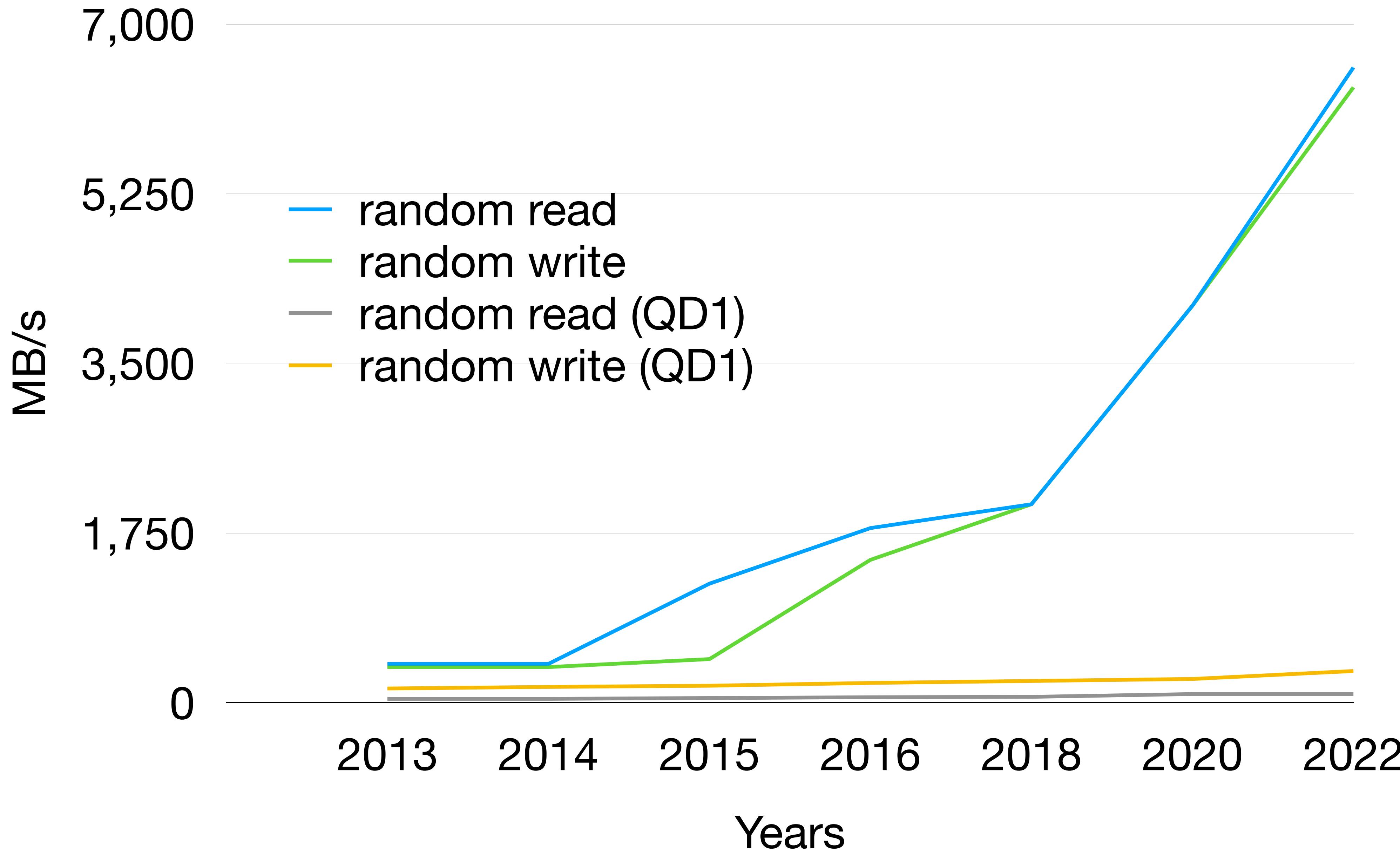
# Extremely simple visualization



# Extremely simple visualization

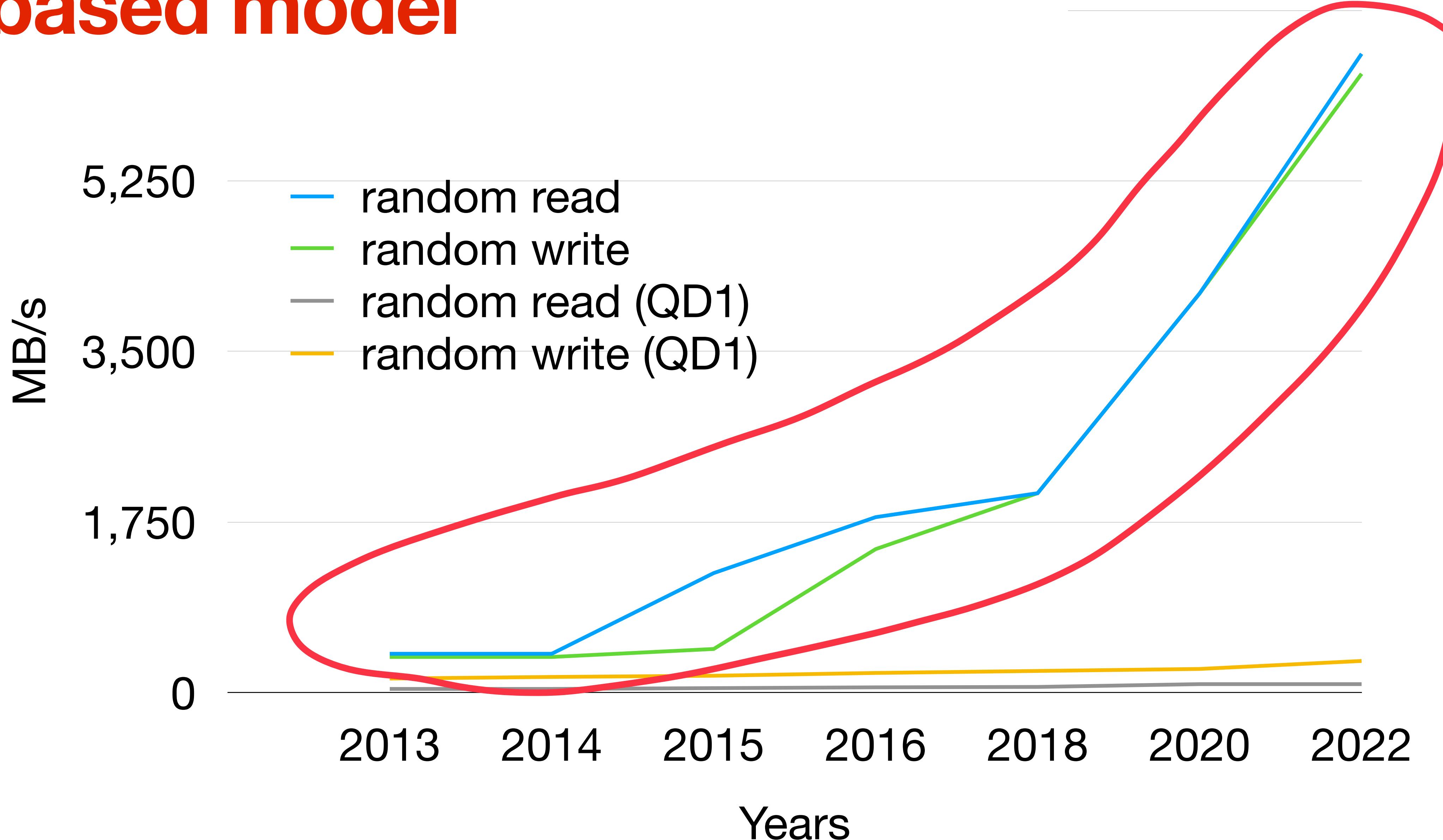


# Samsung SSD performance from 2013-today



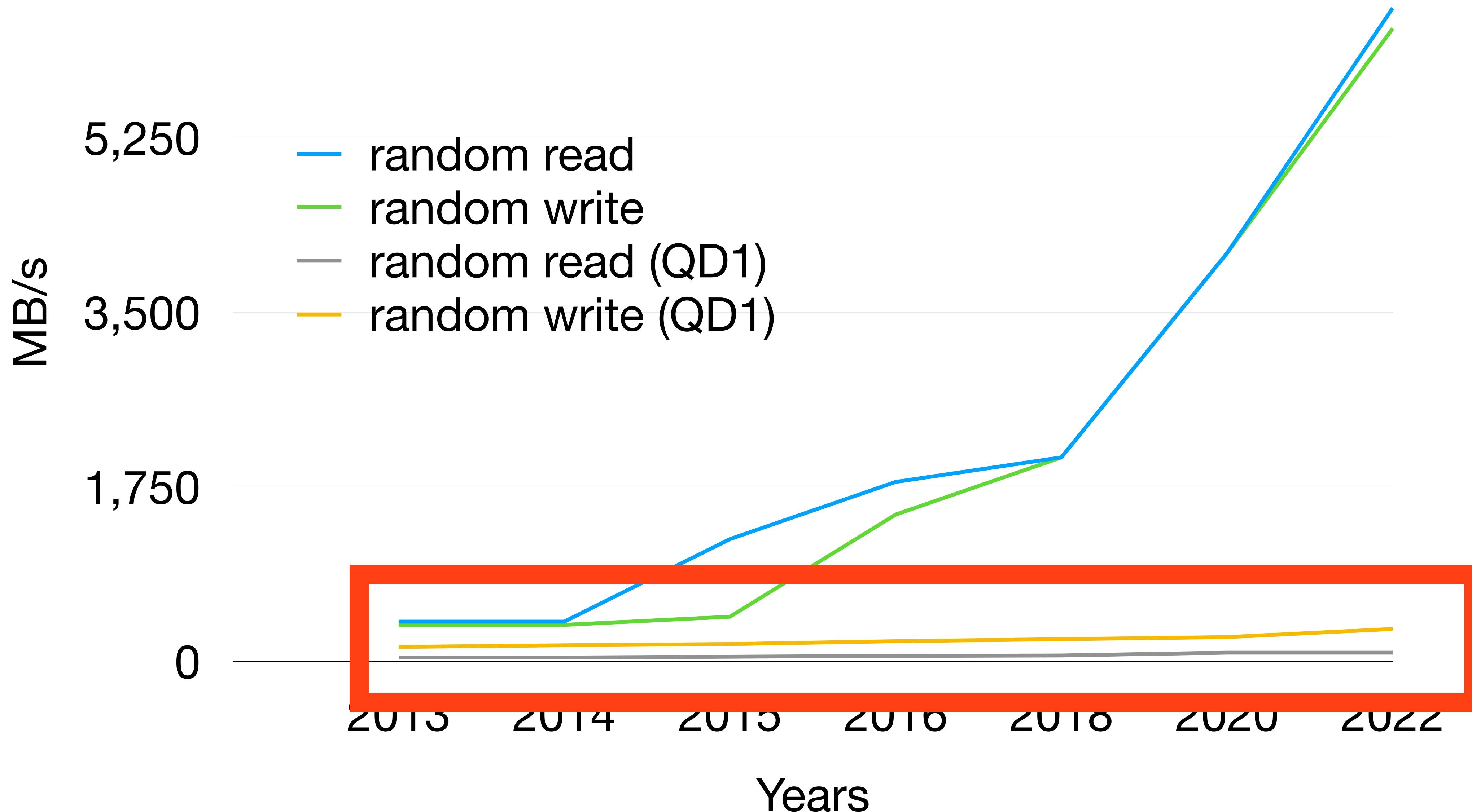
# Validation speed of a UTXO based model

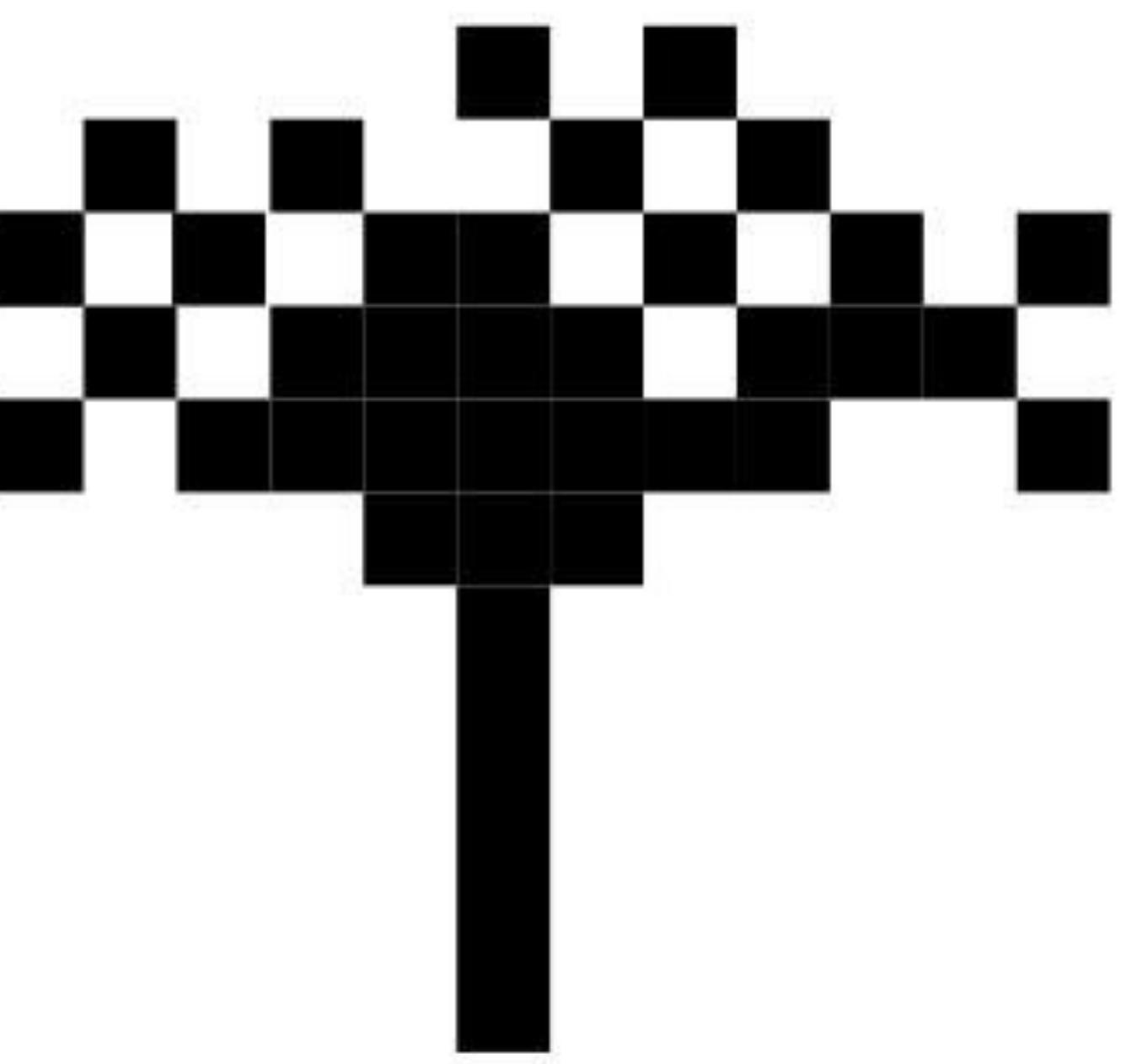
from 2013-today



# Validation speed of an account based model

from 2013-today





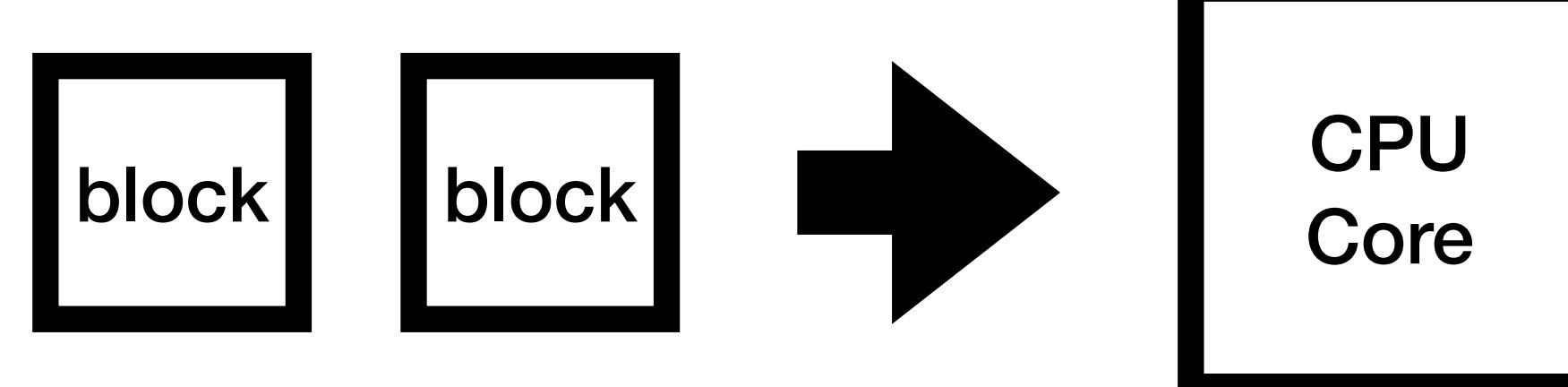
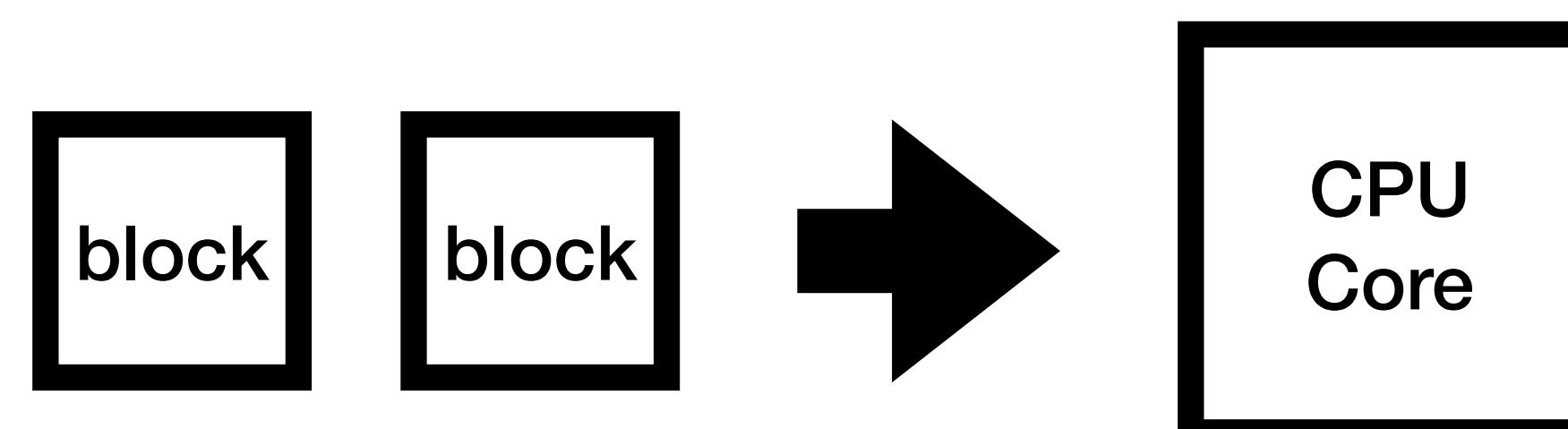
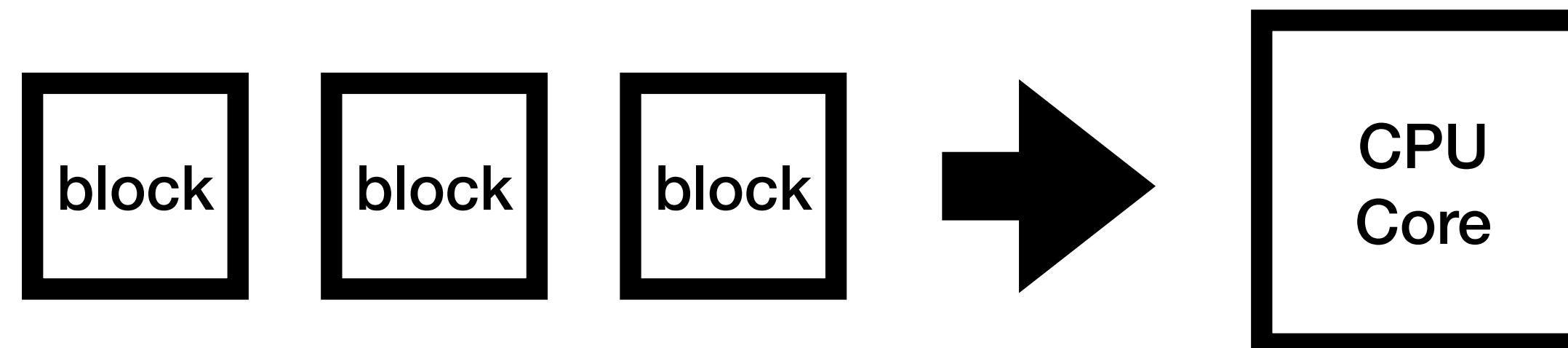
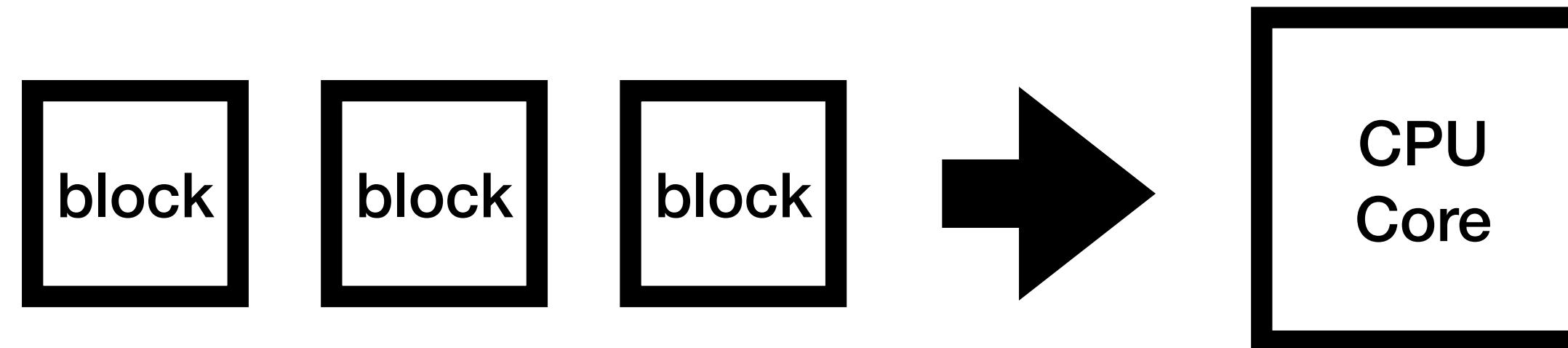
UTRECKO

# Why node with Utreexo scales better

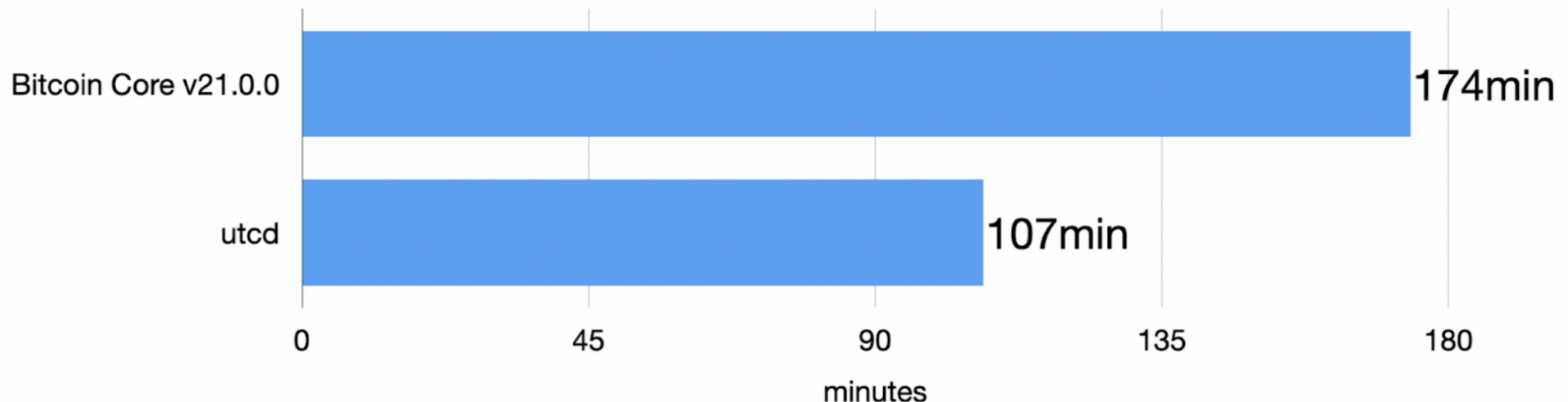
- Can now verify blocks in parallel as well as verifying transactions in parallel

# UTXO model with Utreexo

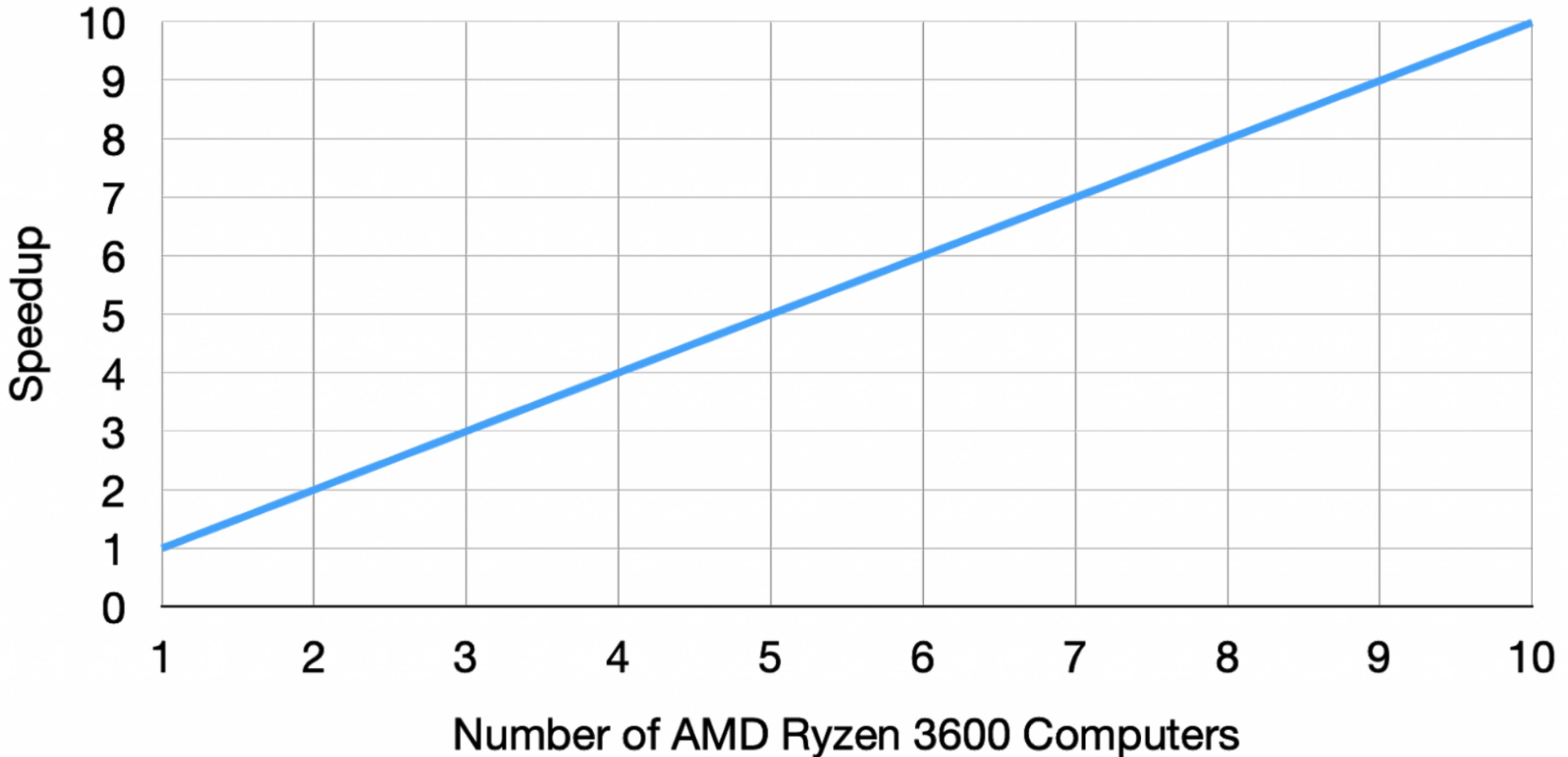
## Visualization of validation of 10 blocks



Initial Block Download Speeds - local nodes  
(default mode, no signature checks until block 654,683)



# Utreexo Parallel Node Speedups (based on Amdahl's Law)



# **The Free Lunch Is Over**

## **A Fundamental Turn Toward Concurrency in Software**

**By Herb Sutter**

The biggest sea change in software development since the OO revolution.

*This article appeared in Dr. Dobb's Journal, 30(3), March 2005. A much*

# Why the UTXO model scales better

## Conclusion

- Parallelism during tx validation in a single block
- Less disk i/o bottlenecked

# Slides



<https://github.com/kcalvinalvin/btcsingapore-09-2024>