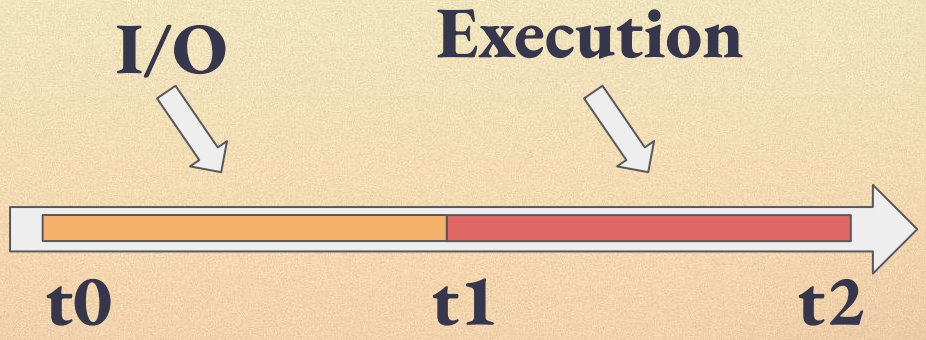
Block-level Access Lists

EIP-7928

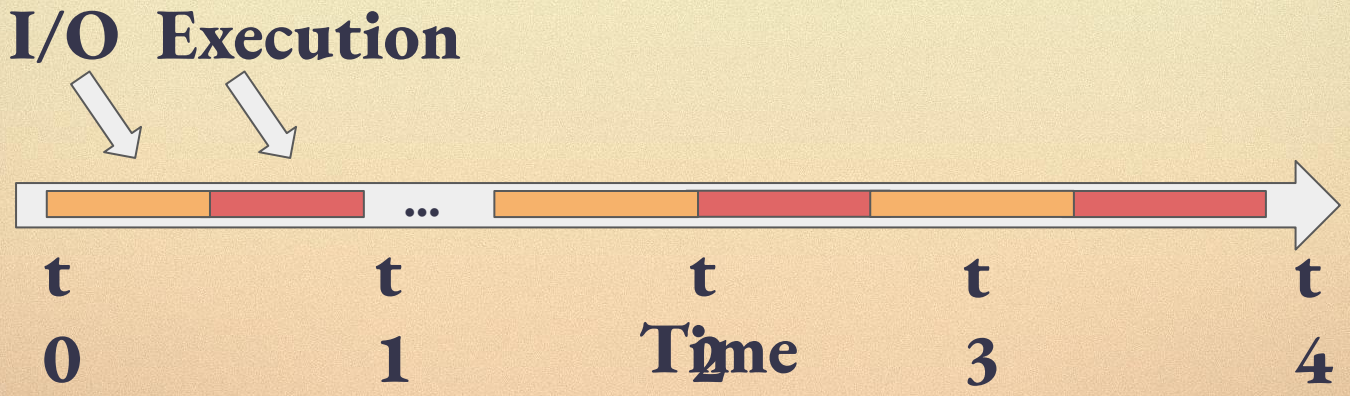
By Toni Wahrstätter

I/O and Execution: *today*

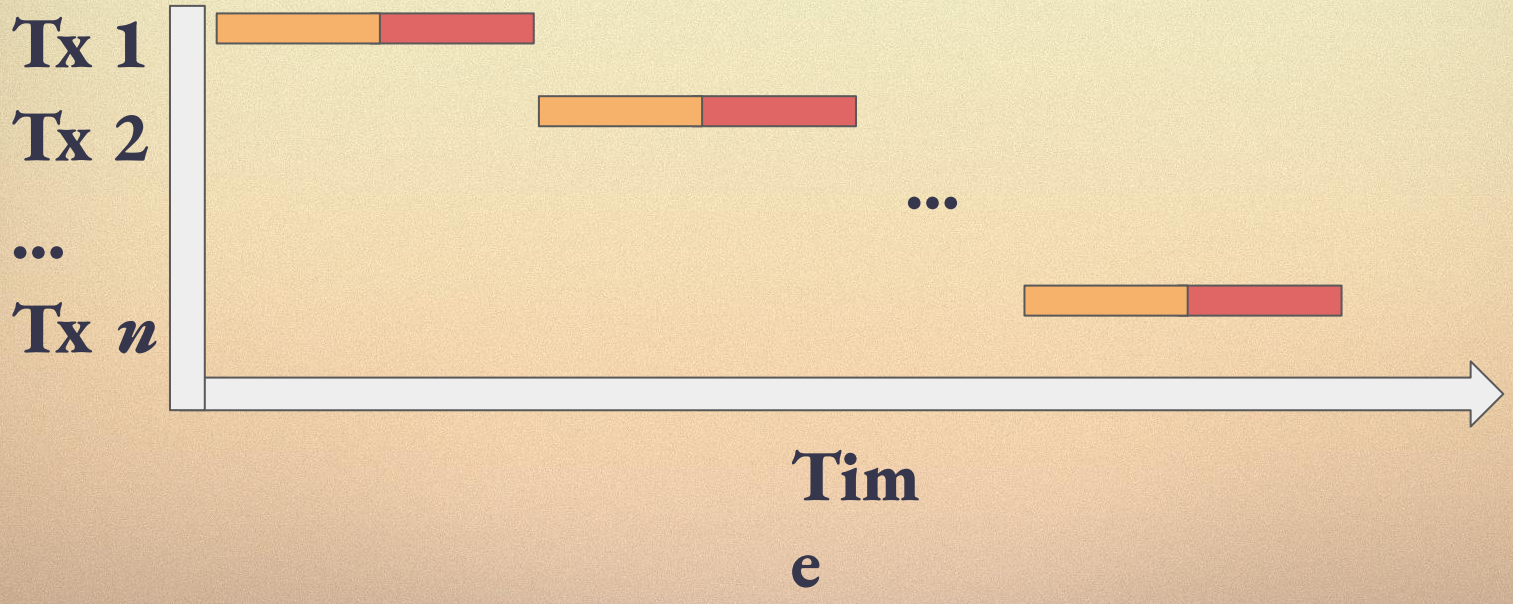
For every transaction/call/access:



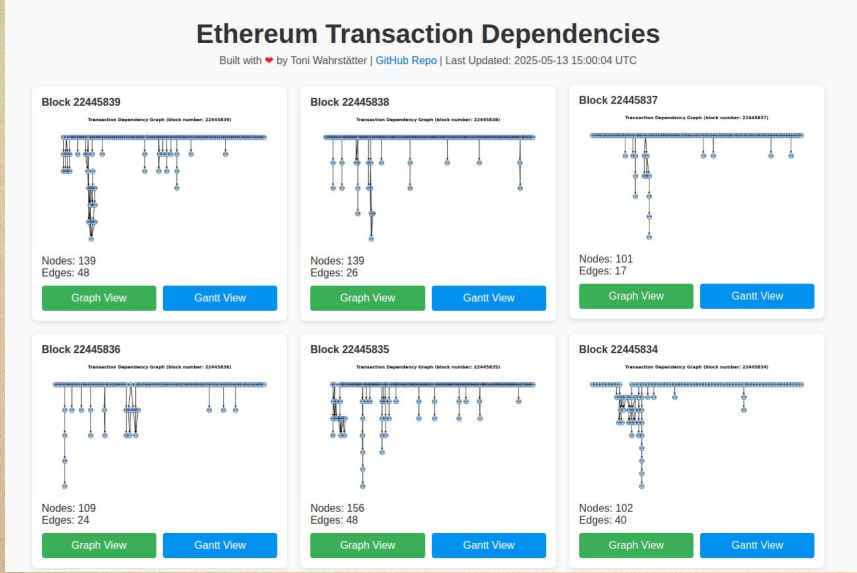
I/O and Execution: *today*

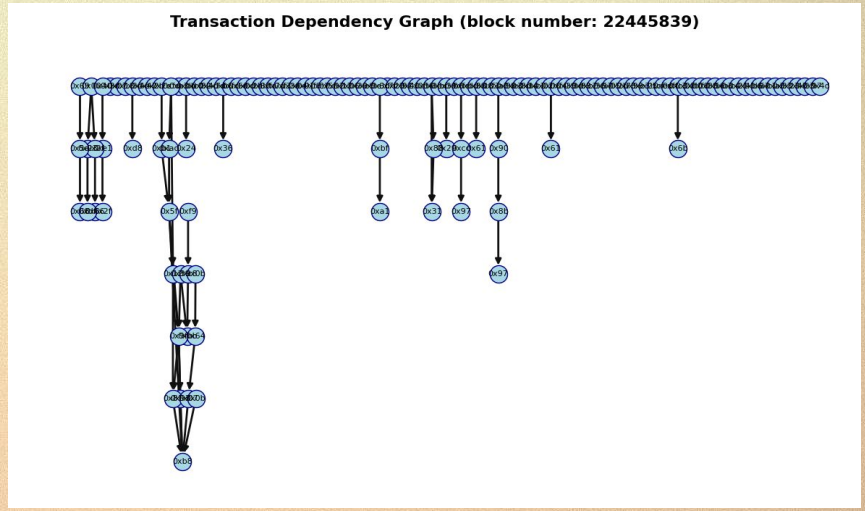


I/O and Execution: *today*

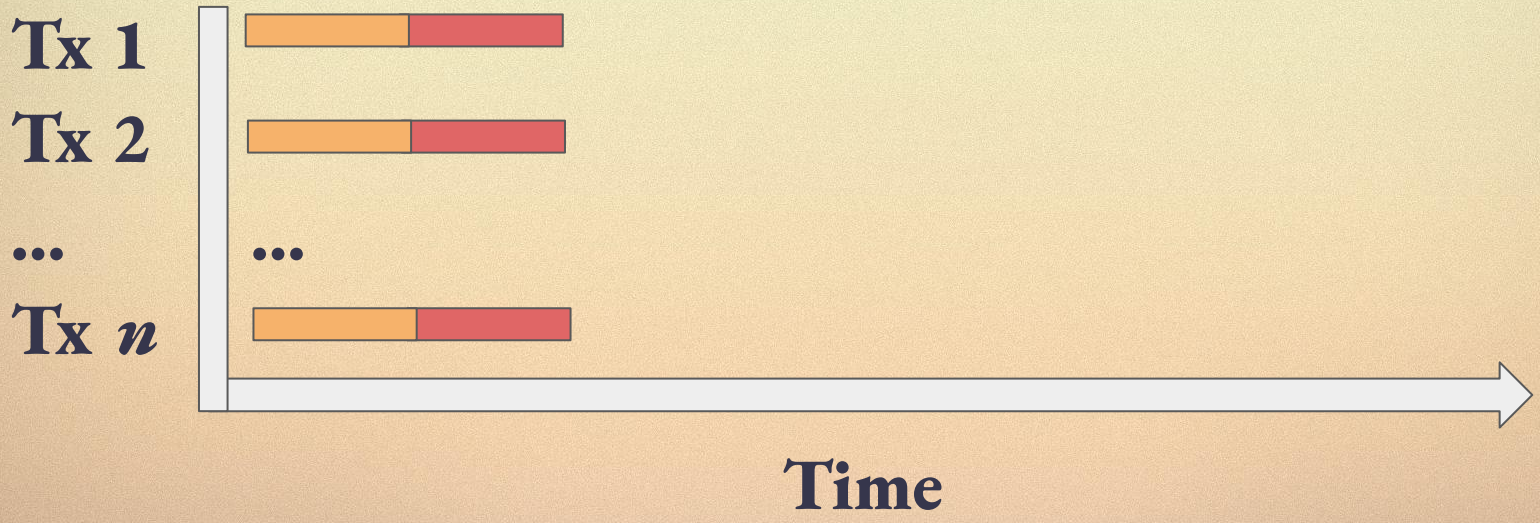


Dependency.pics

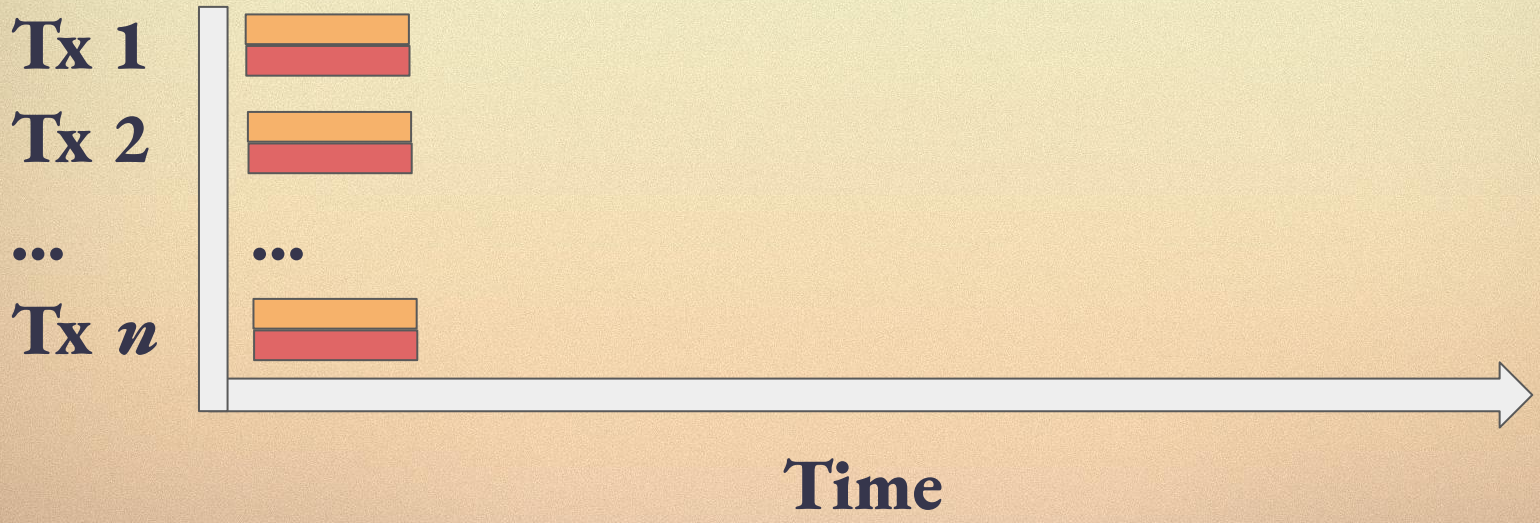




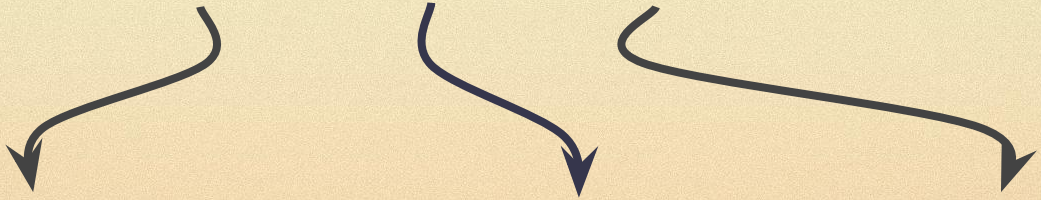
I/O and Execution: *tomorrow*



I/O and Execution: *tomorrow*



BALs



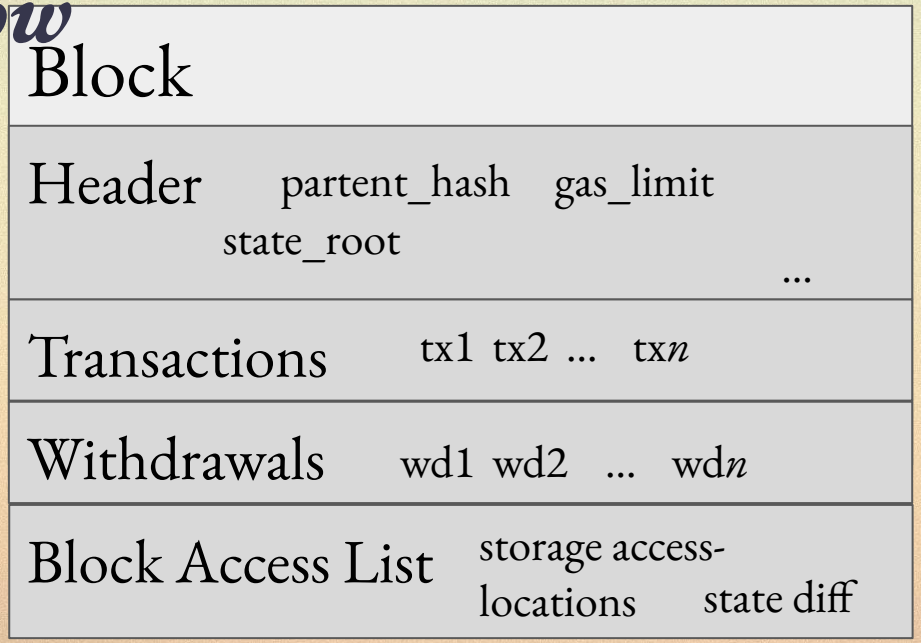
Block-leve l

Access Lists

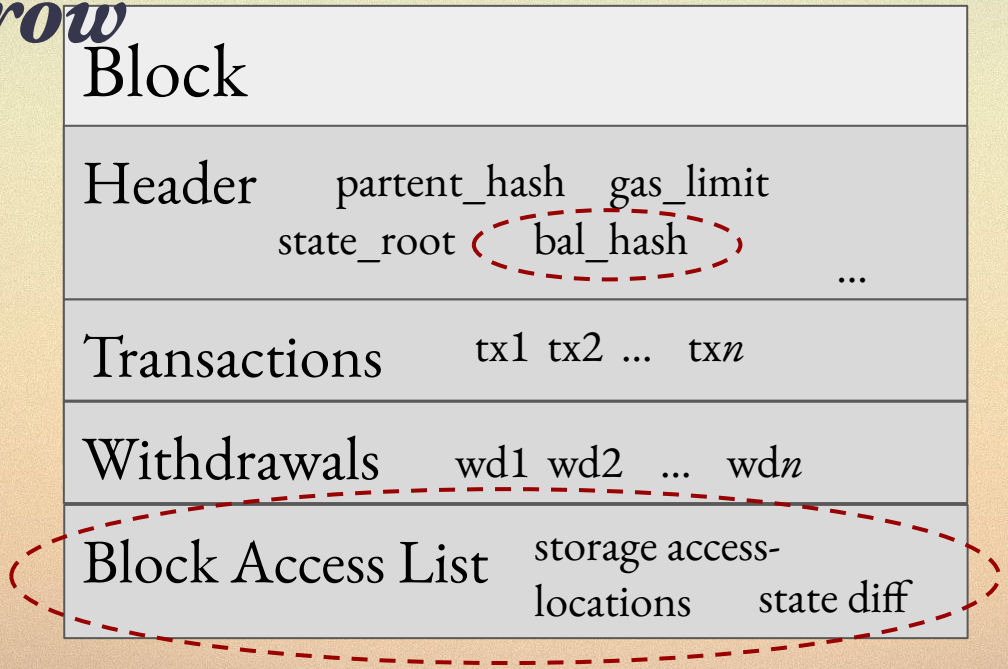
Block Structure:
today

Block	
Header	partent_hash gas_limit state_root ...
Transactions	tx1 tx2 ... tx*n*
Withdrawals	wd1 wd2 ... wd*n*

Block Structure:
tomorrow



Block Structure:
tomorrow



“Builders provide mandatory access lists that let validators verify blocks faster.”

BAL Design Space

BAL Design Space

- Storage Locations
- Storage Values
- State Diffs

BAL Design Space: Storage Locations

Storage location tuples: (address, storage key)

BAL Design Space: Storage Locations

Storage location tuples: (address, storage key)

- Parallel I/O
- Parallelization for average cases
- No parallelization in the worst-cases

BAL Design Space: Storage Values

- Pre-Block
  - The value of the storage slot at the pre-state
- Post-Block
  - The value of the storage slot at the post-state
- Pre-Tx
  - The value of the storage slot at the state before the tx
- Post-Tx
  - The value of the storage slot at the state after the tx

BAL Design Space: State Diff

- Nonces
  - For CREATE and CREATE2
- Balances
  - ETH balance deltas of all accounts with changes

All other state diffs can be derived by looking at the transactions.

Which design is best?

Some facts

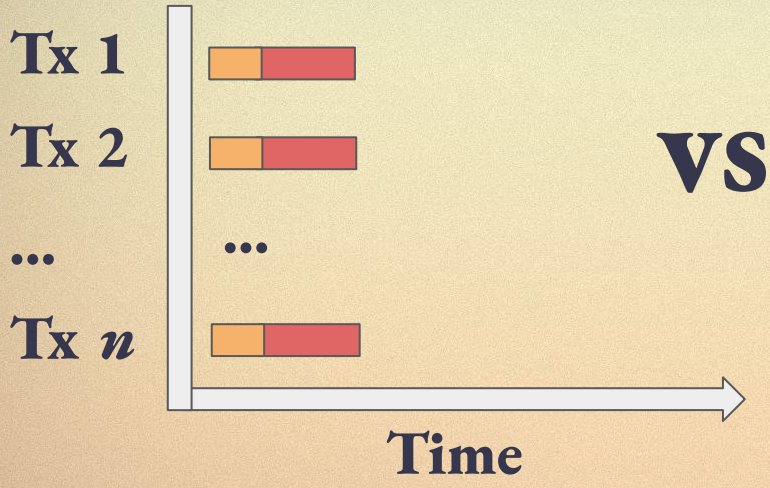
Execution is the bottleneck, not necessarily I/O

- For geth:
  - 50% EVM Execution
  - 13% I/O

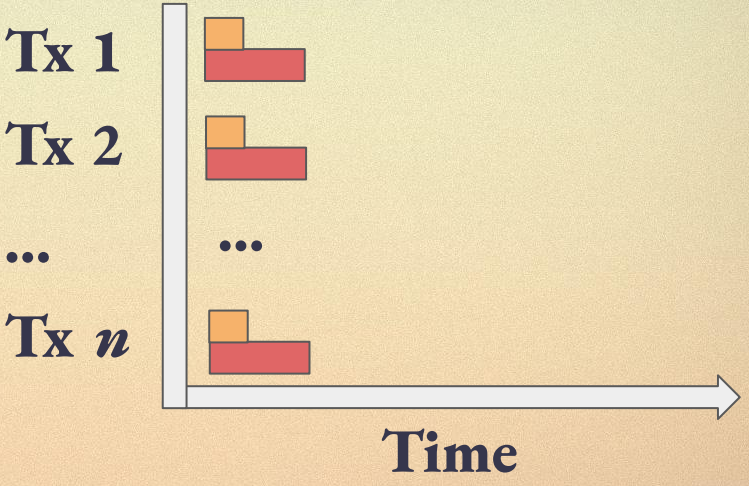
Prefetchers and similar approaches do an amazing job for avg. blocks but struggle with worst-cases

60-80% of all transaction are independent

Design Tradeoff: Post-tx vs. Pre-tx values



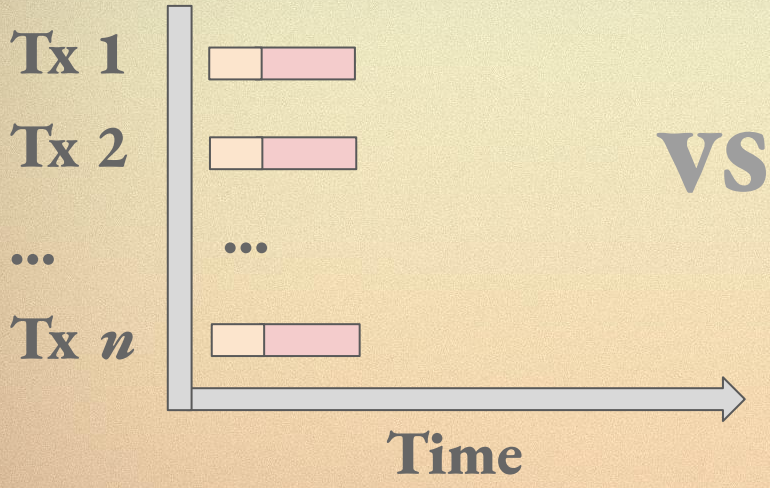
vs.



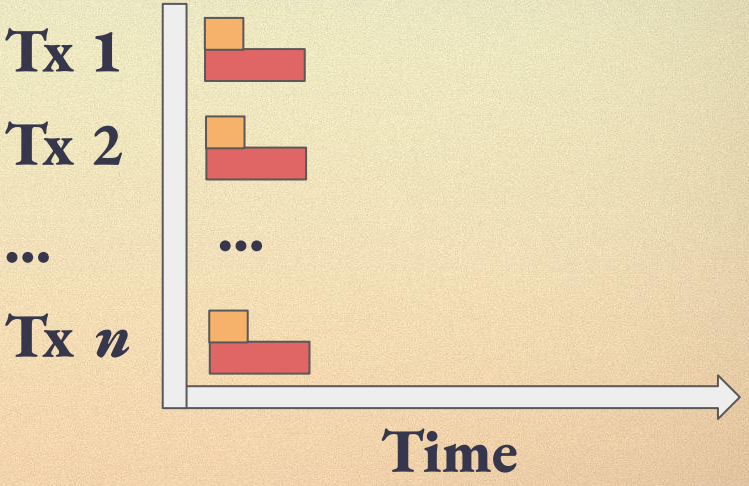
Execution Time:
parallel I/O + parallel EVM

Execution Time:
max(parallel I/O, parallel EVM)

Design Tradeoff: Post-tx vs. Pre-tx values



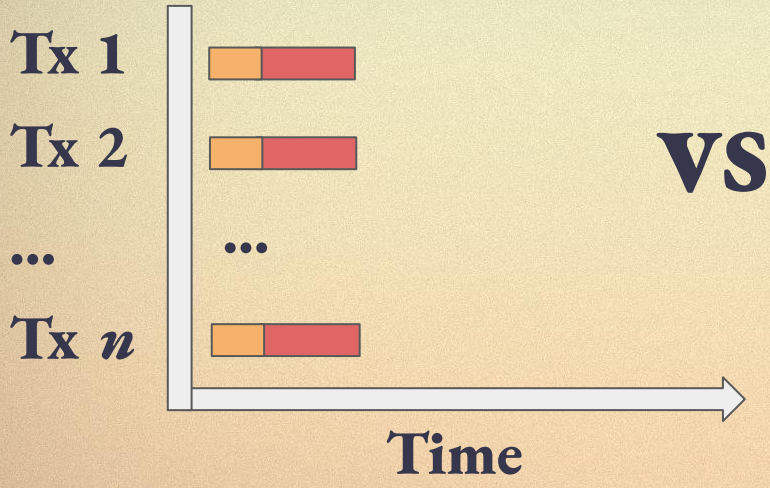
vs.



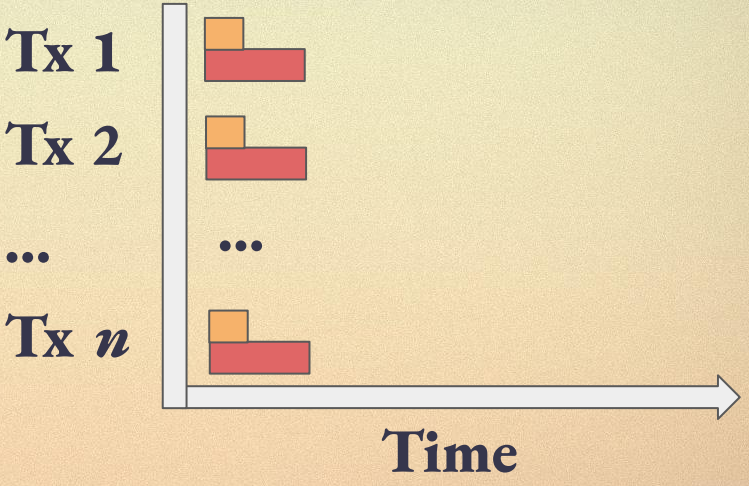
Execution Time:
parallel I/O + parallel EVM

Execution Time:
max(parallel I/O, parallel EVM)

Design Tradeoff: Post-tx vs. Pre-tx values



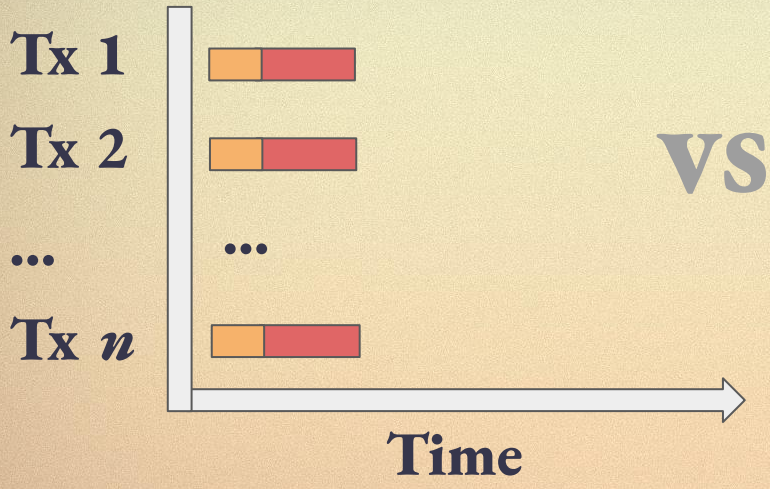
vs.



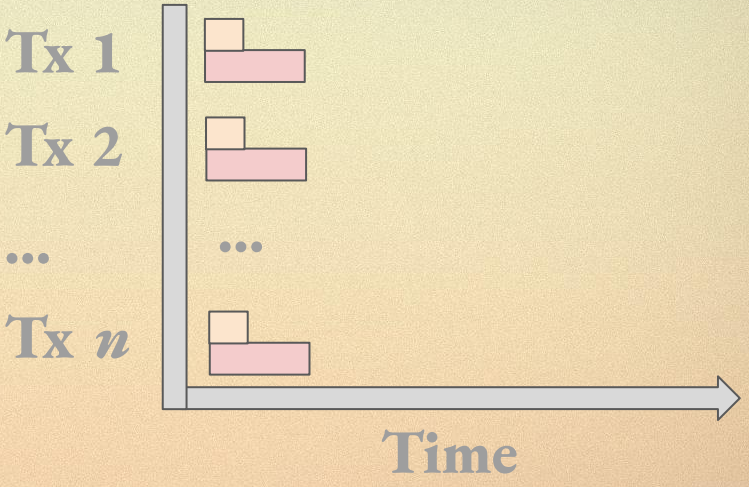
Execution Time:
parallel I/O + parallel EVM
Worst-case size: 0.91 MiB

Execution Time:
max(parallel I/O, parallel EVM)
Worst-case size: 1.51 MiB

Design Tradeoff: Post-tx vs. Pre-tx values



VS.



Execution Time:
parallel I/O + parallel EVM
Worst-case size: 0.91 MiB

Execution Time:
max(parallel I/O, parallel EVM)
Worst-case size: 1.51 MiB

BAL Design Space

Design	Execution Times	Max Size	State diff
Storage Locations	Parallel I/O + sequential EVM	0.93 MiB	❌
Pre-tx values[1]	max(parallel I/O, parallel EVM)	1.51 MiB	❌
Post-tx values[1] [2]	Parallel I/O + parallel EVM	0.93 MiB	✅
Pre-block + Post-tx values[1] [2]	max(parallel I/O, parallel EVM)	1.51 MiB	✅

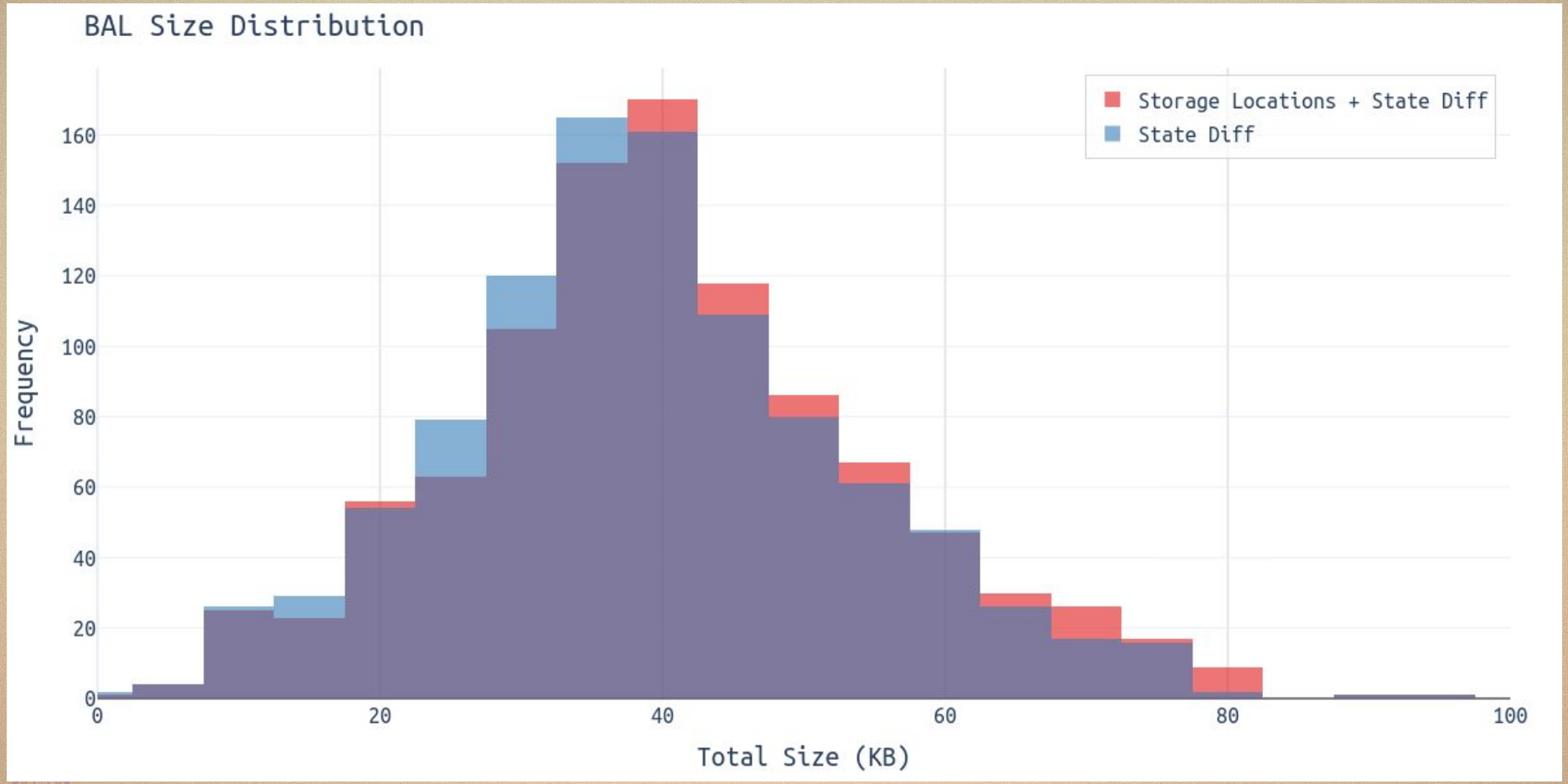
[1] with storage locations
[2] with state diffs

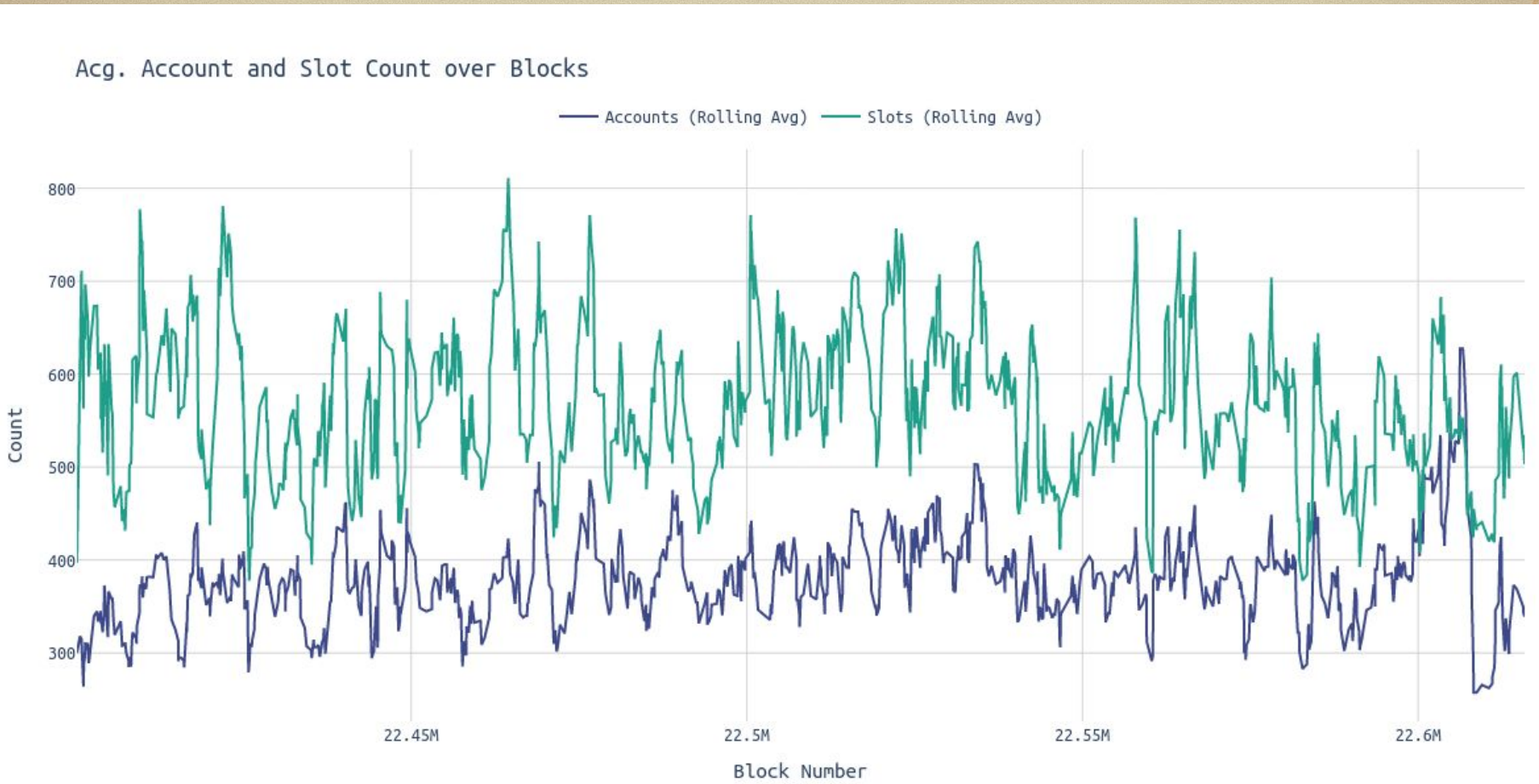
Further readings

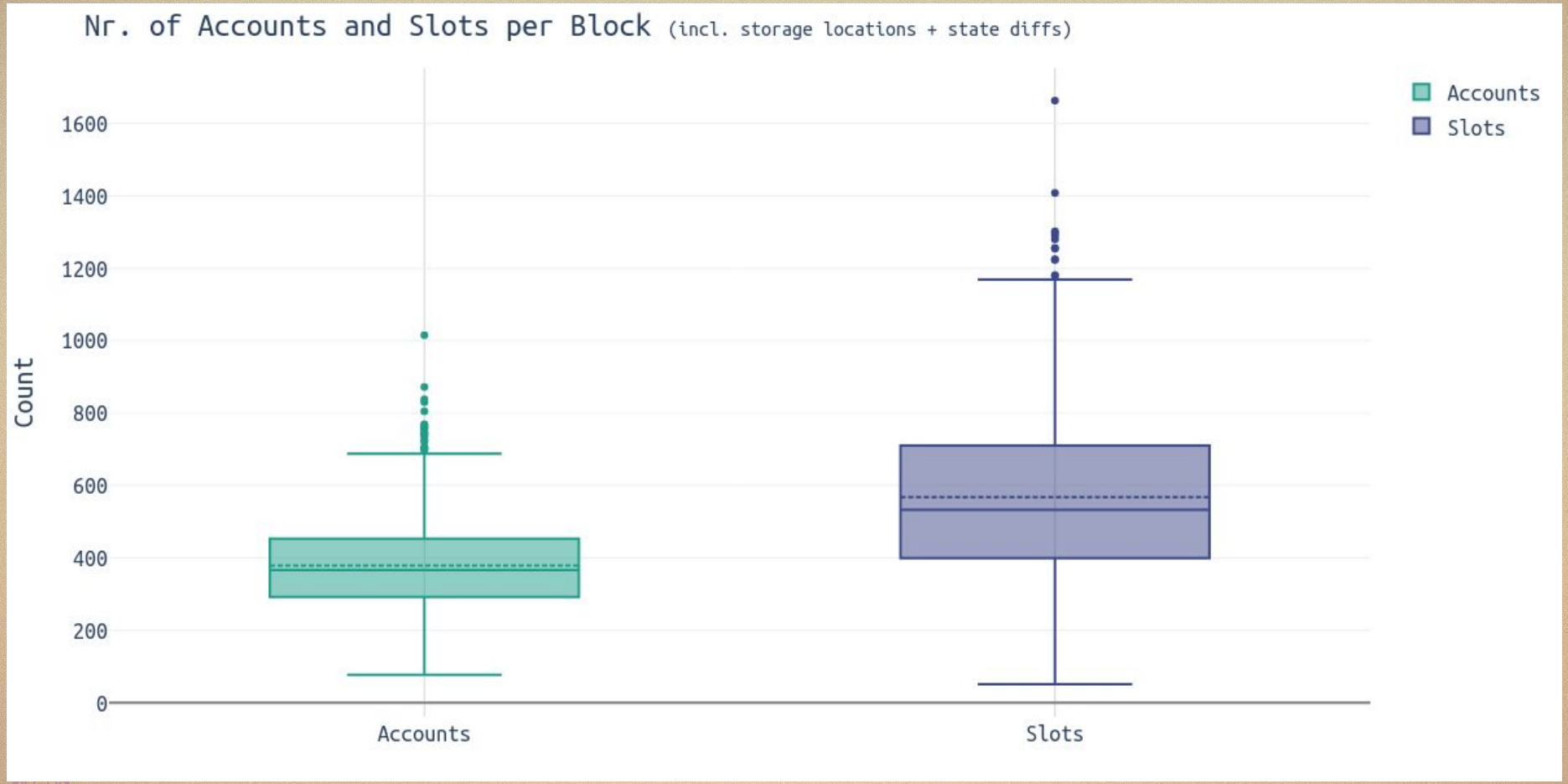
EIP-7928

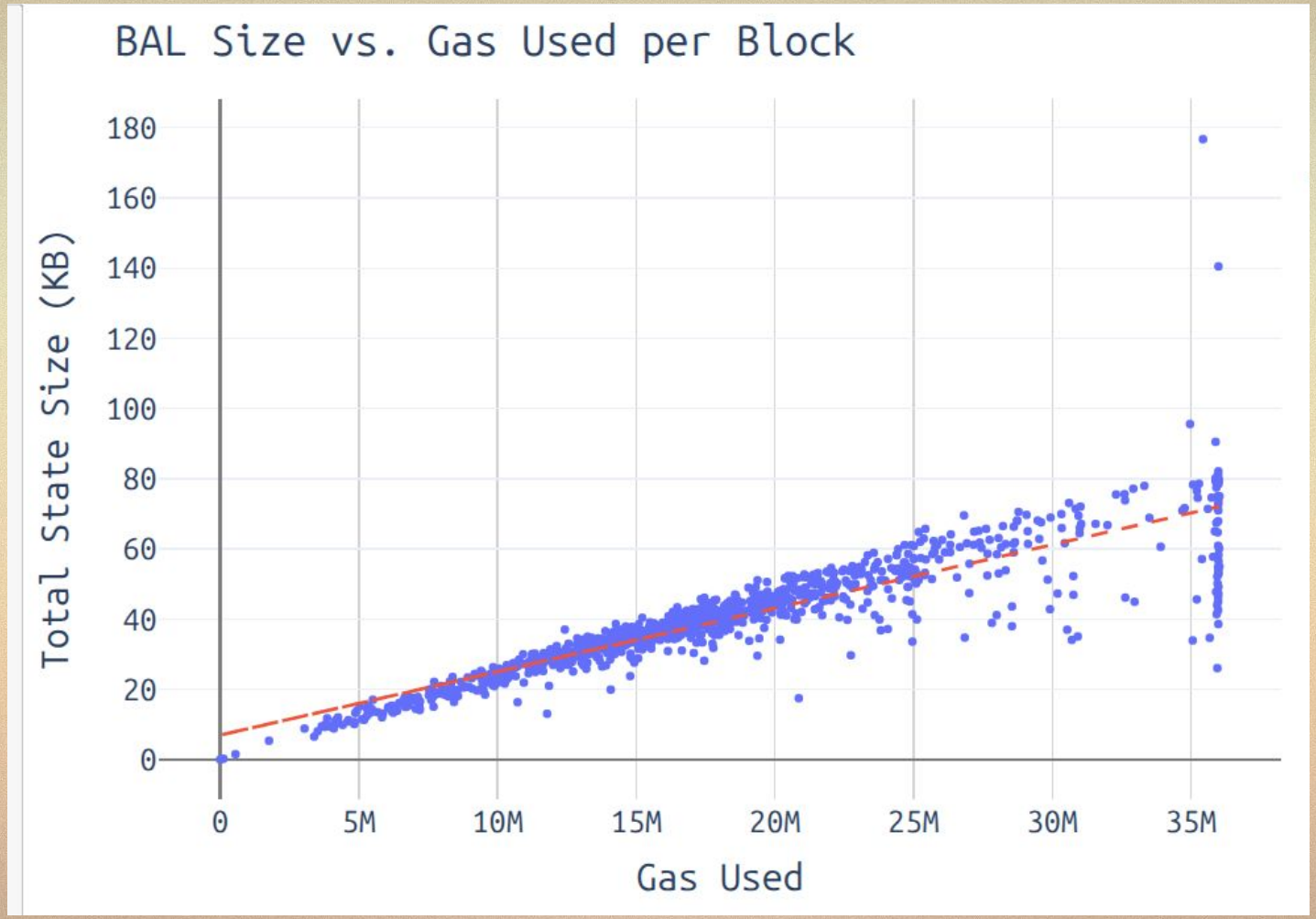
ethresear.ch post

dependency.pics


BAL Size Distribution
Storage Locations + State Diff
State Diff
Frequency
0
20
40
60
80
100
120
140
160
0
20
40
60
80
100
Total Size (KB)


Acg. Account and Slot Count over Blocks
Accounts (Rolling Avg)
Slots (Rolling Avg)
800
700
600
500
400
300
Count
22.45M
22.5M
22.55M
22.6M
Block Number


Nr. of Accounts and Slots per Block (incl. storage locations + state diffs)
Accounts
Slots
Count
1600
1400
1200
1000
800
600
400
200
0
Accounts
Slots


BAL Size vs. Gas Used per Block
Total State Size (KB)
180
160
140
120
100
80
60
40
20
0
0
5M
10M
15M
20M
25M
30M
35M
Gas Used