MONASH University

MONASH
INFORMATION
TECHNOLOGY

Transaction Allocation in Sharded Blockchain Systems

Yuanzhe Zhang

Supervisor: Dr. Jiangshan Yu and Prof. Shirui Pan

Monash University, Melbourne, Australia

January 2023

GROUP OF EIGHT AUSTRALIA

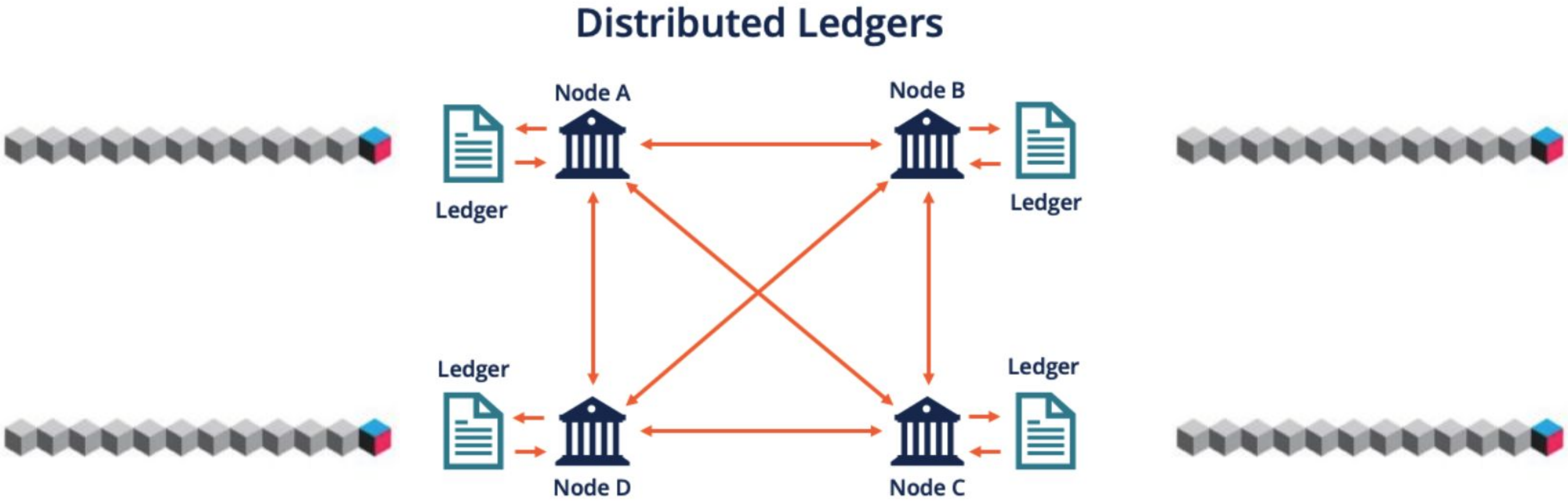
TABLE OF CONTENTS

MONASH University

MONASH
INFORMATION
TECHNOLOGY

Blockchains

Immutable distributed ledgers



- ## Bitcoin [1]

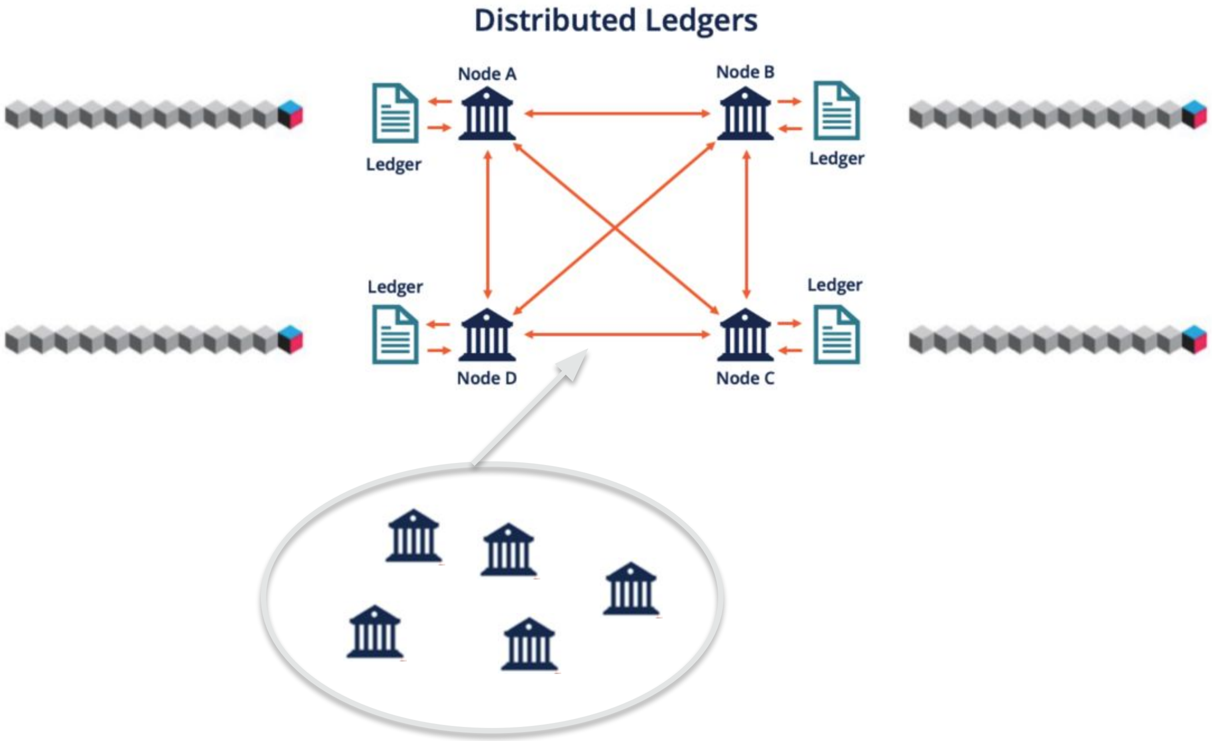
First blockchain proposed in 2008
Market Cap: about 419 billion USD [3]
UTXO model (Unspent Transaction Outputs)
Throughput: about 7 transactions/second (TPS)

- ## Ethereum [2]

Smart contracts enabled blockchain
Market Cap: about 140 billion USD [3]
Account/Balance model
Throughput: about 15 transactions/second (TPS)

[1] S Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[2] https://ethereum.org/en/
[3] https://coinmarketcap.com/, Data fetched on 14/June/2022

Loosely speaking, throughput should increase linearly with the number of miners. However, Bitcoin and Ethereum remain about 7 and 15 TPS, no matter how many miners involved.



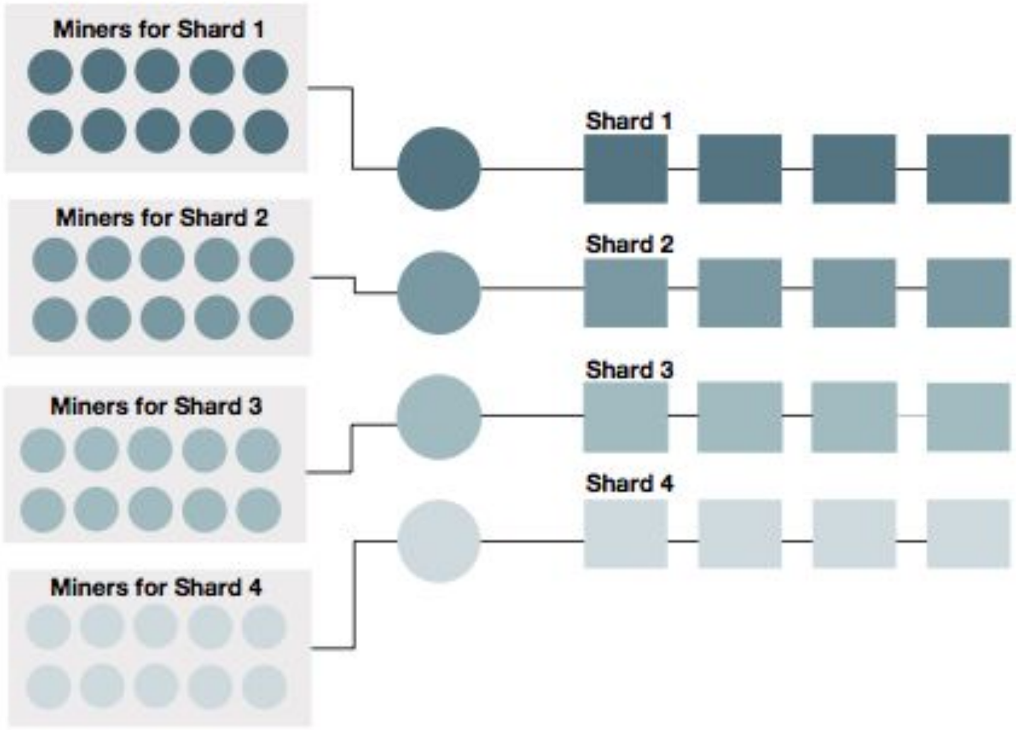
MONASH University

MONASH
INFORMATION
TECHNOLOGY

Sharding Techniques

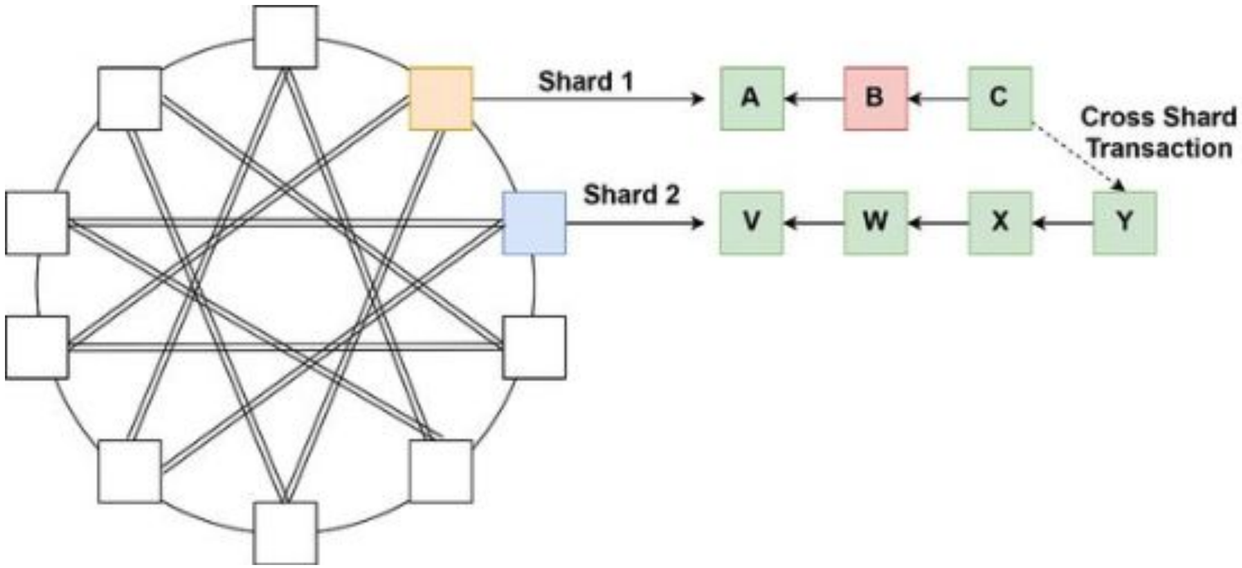
Sharding protocols allocate Txs and Miners into multiple shards for parallel processing.
Miners → # Shards → # TPS

Linear throughput improvement!

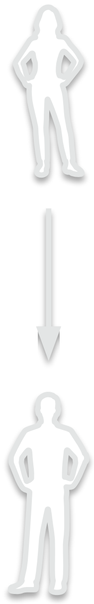


Cross-shard Transactions

- This occurs when affected accounts of a Tx are in different shards.
- The transaction modifies state in different shards.
- They have to communicate and achieve consensus.
- Expensive to process!



Amritraj Singh, PUBLIC BLOCKCHAIN SCALABILITY: ADVANCEMENTS, CHALLENGES AND THE FUTURE



Alice in Shard C

Money Transfer

Bob in Shard Y

MONASH University

MONASH
INFORMATION
TECHNOLOGY

Research Question

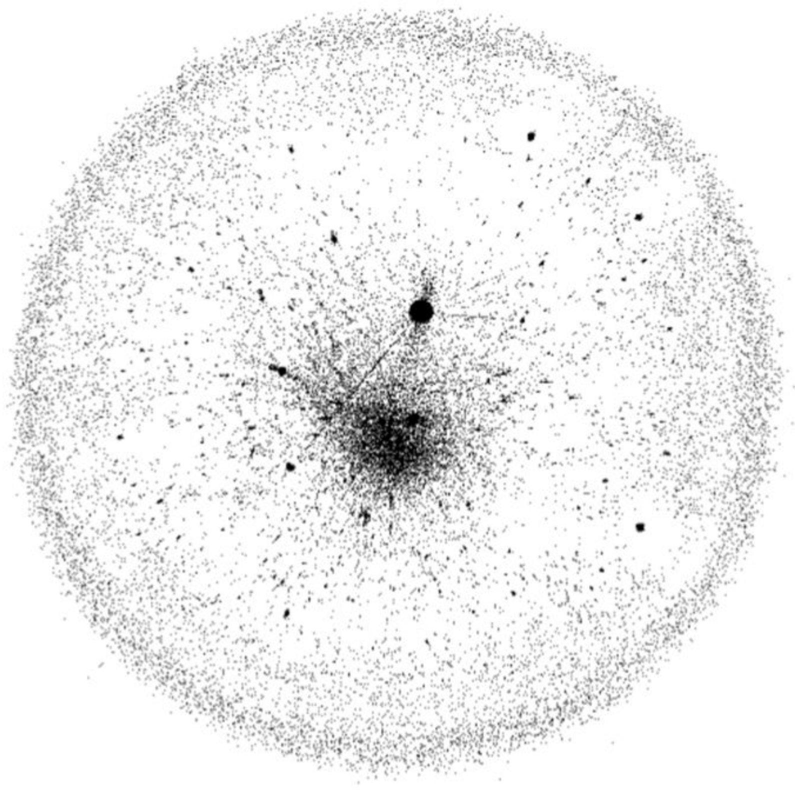
Transaction Allocation to reduce the occurrence of Cross-shard Transactions

Two directions to tackle with cross-shard transactions:

1. Efficient cross-shard consensus,
2. Reducing the occurrence of cross-shard transactions.

Basic Idea: Account Allocation determines cross-shard transactions!

→ Put intensively-interacted accounts into one shard!



Related Works

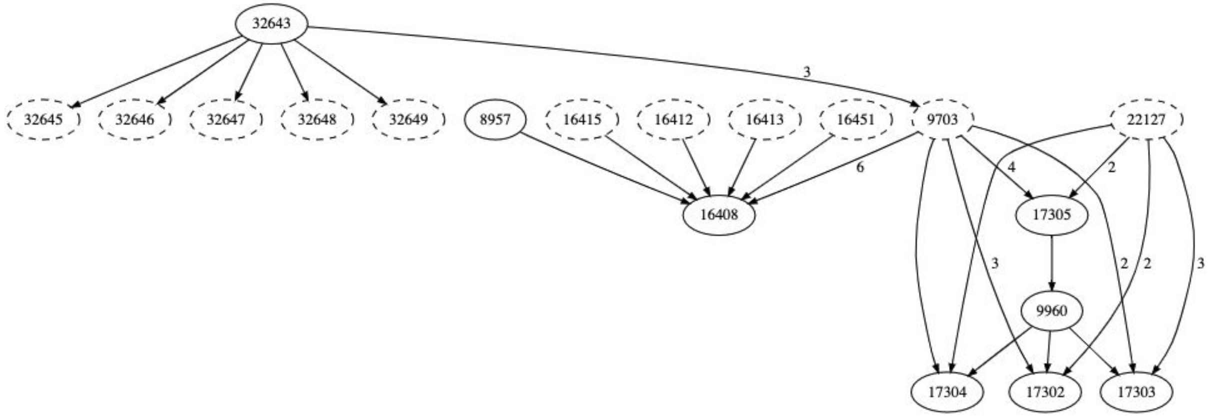
[4-6] focus on the security design and efficiency of cross-shard consensus

Hash-bash allocation — e.g. SHA256 (address) mod k, where k is the number of shards. [4]

- ➢ Random to historical transaction patterns.
- ➢ Huge amount of cross-shard transactions, more than 90% [7]

[4] M Al-Bassam, A Sonnino, S Bano, D Hrycyszyn, and G Danezis, "Chainspace: A sharded smart contracts platform." NDSS, 2018
[5] J Wang and H Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones," in *16th USENIX Symposium on Networked Systems Design and Implementation*, 2019
[6] E Kokoris-Kogias, P Jovanovic, L Gasser, N Gailly, E Syta, and B Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in IEEE Symposium on Security and Privacy (S&P), IEEE, 2018
[7] G Wang, ZJ Shi, M Nixon, and S Han, "Sok: Sharding on blockchain," in Proceedings of the 1st ACM Conference on Advances in Financial Technologies, 2019

TRANSACTION ALLOCATION



- UTXO-based (ICDCS’19 [9])
- Account-based
  - Transaction-level approach (AFT’21 [11])
  - Graph-based approach
    - First identified this problem, METIS graph partition for solution. (DSN-W’18 [8])
    - Targeting on storage problem, also using METIS for allocation. (TNSM’21 [10])
    - Targeting on hot-shard problem, also using METIS for allocation. (INFOCOM’22 [12])

METIS [13] considers the number of inter-group links and degree balance for partition.

[8] E. Fynn and F. Pedone, “Challenges and pitfalls of partitioning blockchains,” in 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2018
[9] LN Nguyen, TD Nguyen, TN Dinh, and MT Thai, “Optchain: Optimal transactions placement for scalable blockchain sharding,” in *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2019
[10] A Mizrahi and O Rottenstreich, “State Sharding with Space-aware Representations,” IEEE Transactions on Network and Service Management,, *2021*
[11] M Kro1, O Ascigil, S Rene, A Sonnino, M Al-Bassam, and E Riviere, “Shard scheduler: Object placement and migration in sharded account- based blockchains,” in *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, 2021
[12] H Huang, X Peng, J Zhan, S Zhang, Y Lin, Z Zheng, and S Guo, Brokerchain: A cross-shard blockchain protocol for account/balance-based state sharding,” in *IEEE INFOCOM*, 2022.
[13] Karypis, G., & Kumar, V. (1997). METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices.

MONASH University

MONASH
INFORMATION
TECHNOLOGY

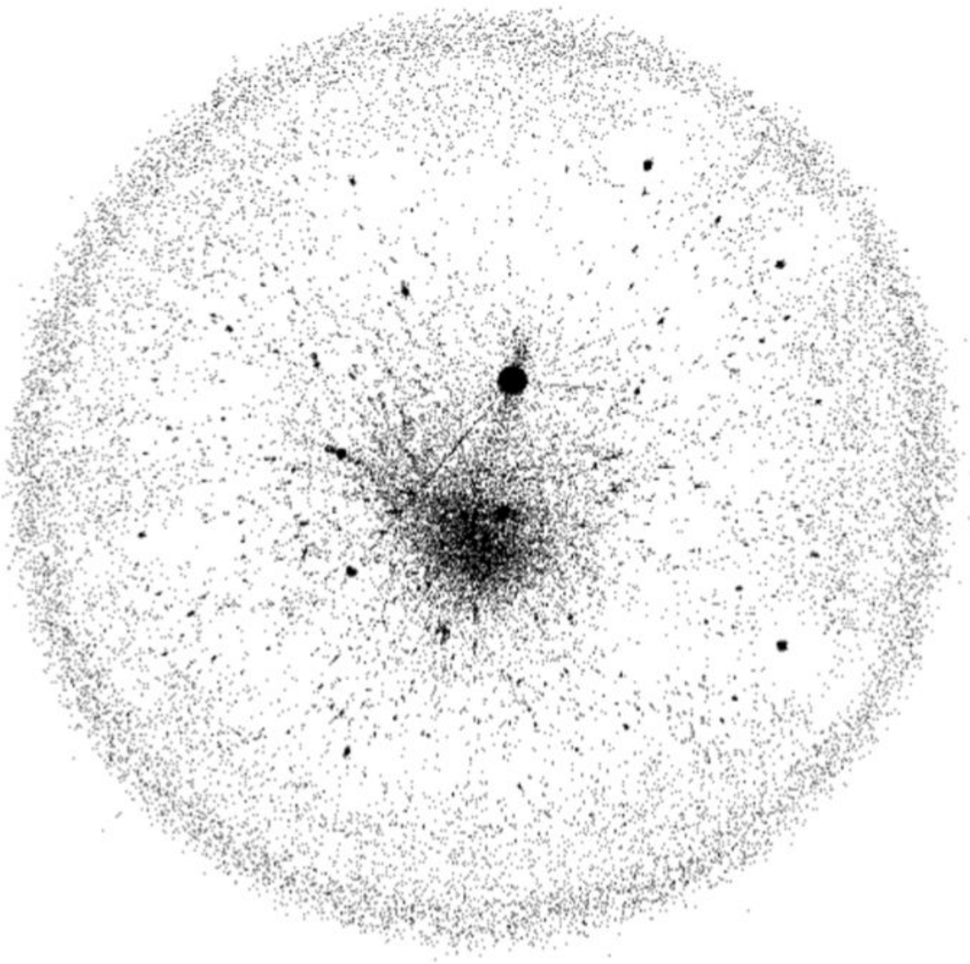
Our Proposed Method

Our Paper has been accepted and will be presented at IEEE ICDE 2023.

“*TxAllo: Dynamic Transaction Allocation in Sharded Blockchain Systems. IEEE International Conference on Data Engineering, ICDE-23, Anaheim, California, United States, April 3 - 7, 2023 (CORE A*)*”

Challenges

- ❖ Workload balance among shards
  - ➢ long-tail distribution of accounts activeness
- ❖ Fast execution
  - ➢ large-scale data and keeping growing
- ❖ Deterministic algorithm for easy verification



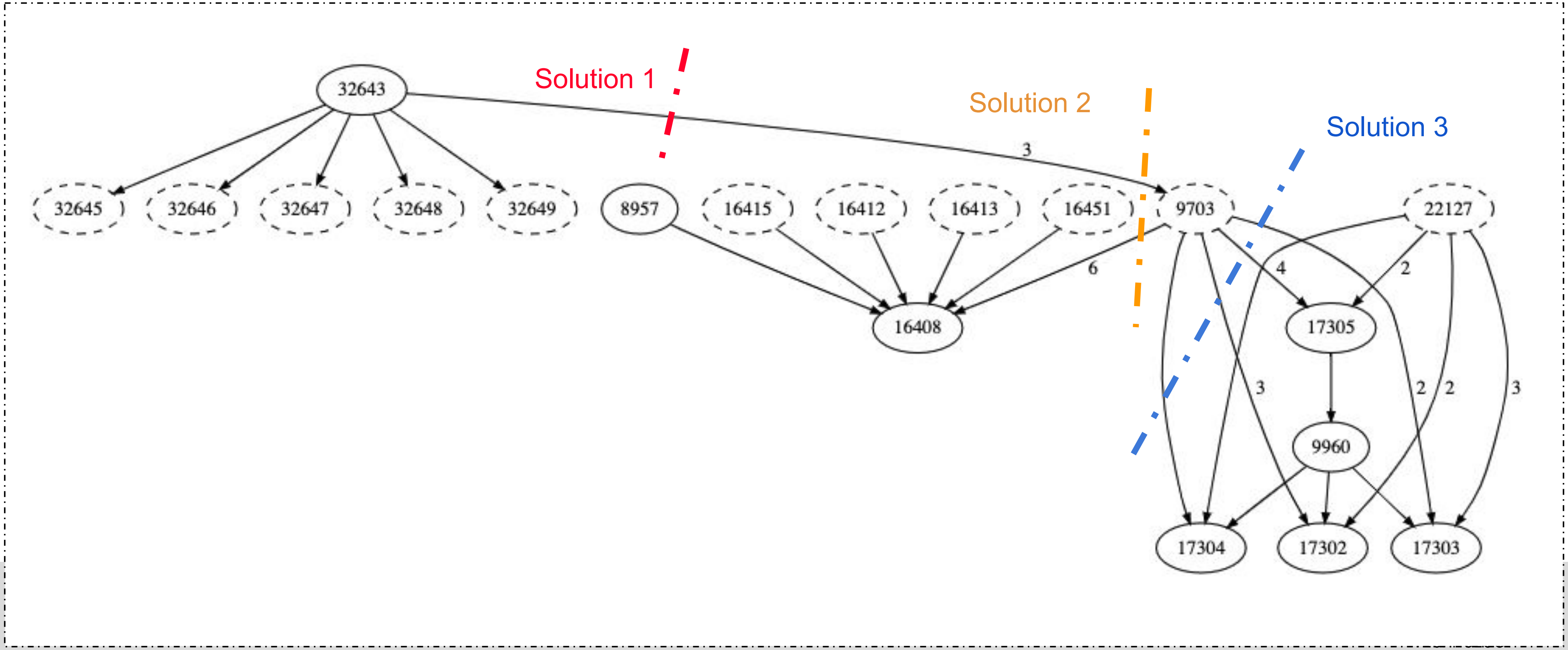
Our Contributions

- We convert this problem to community detection problem on a graph.
  - We define the key concepts on the graph including **cross-shard Tx, processing workload in each shard, throughput, Tx confirmation latency.**
  - We **unify** the optimization target to **one** function, i.e. throughput, considering both **cross-shard Tx ratio** and **workload balance.**
- We propose a dynamic allocation mechanism TxAllo.
  - Deterministic
  - Adaptive updating – using previous allocation and new transaction patterns
  - Fast execution
- We implement TxAllo on Ethereum data with over 91m Txs and 10m accounts.
  - Significant improvement in terms of performance and running time

Hard to tell which solution is the best.

- Solution 1 – imbalanced
- Solution 2 and 3 – more inter-shard cuts

An unified allocation target is required.



Minimize the number of cross-shard transactions, with workload balance bounded by the processing capacity.

Workload $\sigma_i = \sum_{v \in V_i, u \in V_i} w_{\{v,u\}} + \eta * \sum_{v \in V_i, u \notin V_i} w_{\{v,u\}}.$

Difficulty parameter η

Processing capacity in each shard λ

Throughput when capacity is enough $\hat{\Lambda}_i = \sum_{v \in V_i, u \in V_i} w_{\{v,u\}} + \frac{\sum_{v \in V_i, u \notin V_i} w_{\{v,u\}}}{2}$

Throughput $\Lambda_i = \begin{cases} \hat{\Lambda}_i, & \sigma_i \le \lambda \\ \frac{\lambda}{\sigma_i} * \hat{\Lambda}_i, & \sigma_i > \lambda \end{cases}.$

Global Method

Basic idea: loop for accounts

Each of them joins the shard with optimal throughput

Adaptive Method

Basic idea: Only change the allocation for accounts which appear in newly-included blocks.

MONASH University

MONASH
INFORMATION
TECHNOLOGY

Experimental Results

Dataset & Evaluation Metrics

– Over 91m Txs and 12m accounts Ethereum data.

– Implementation with Python 3.8 on Intel Xeon Gold 6150 CPU and 250 GB memory

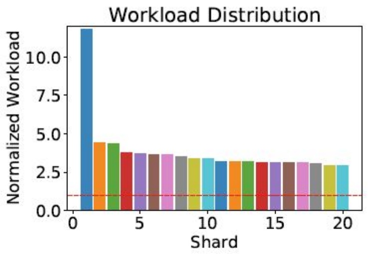
– Evaluations:

1. Cross-shard Transaction Ratio
2. Workload Balance, i.e. standard deviation of workload
3. Throughput
4. Transaction confirmation latency
5. Algorithm running time

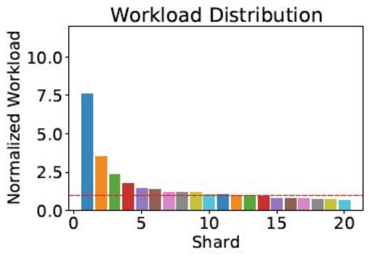
- ~200 seconds running time/ ~400s METIS-based algorithm,

Within 60 shards:

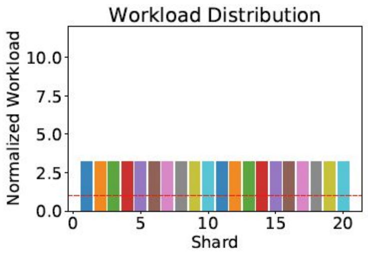
- ~12% Cross-shard transactions/~27% METIS
- Better Throughput and workload balance



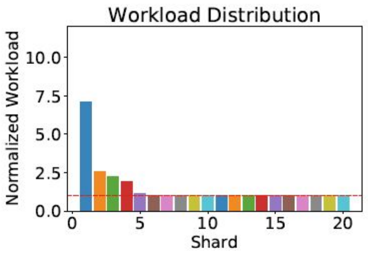
(a) Random



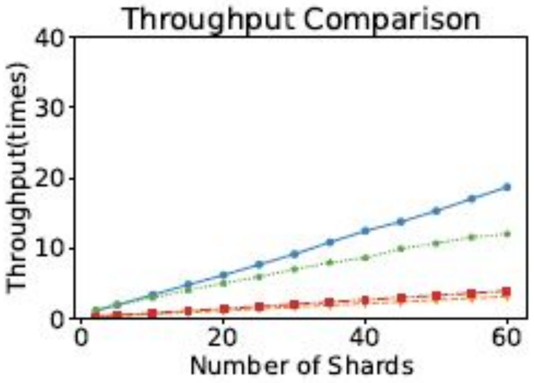
(b) METIS

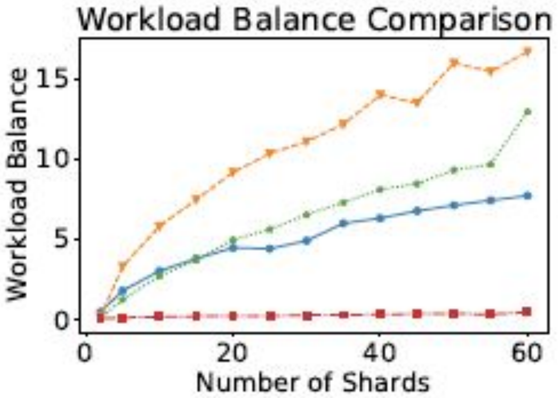


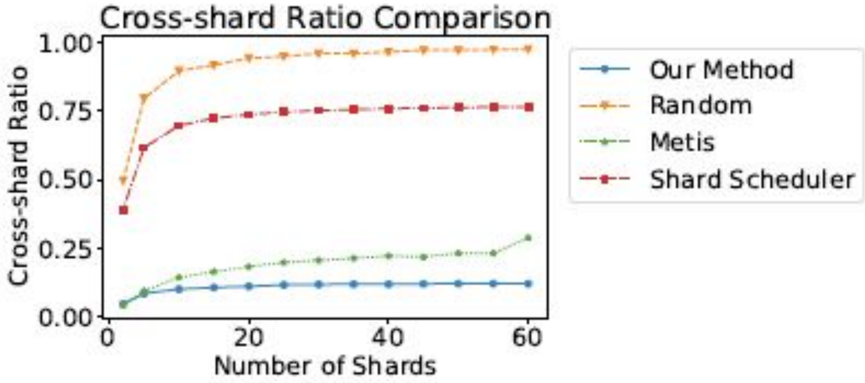
(c) Shard Scheduler



(d) Our method

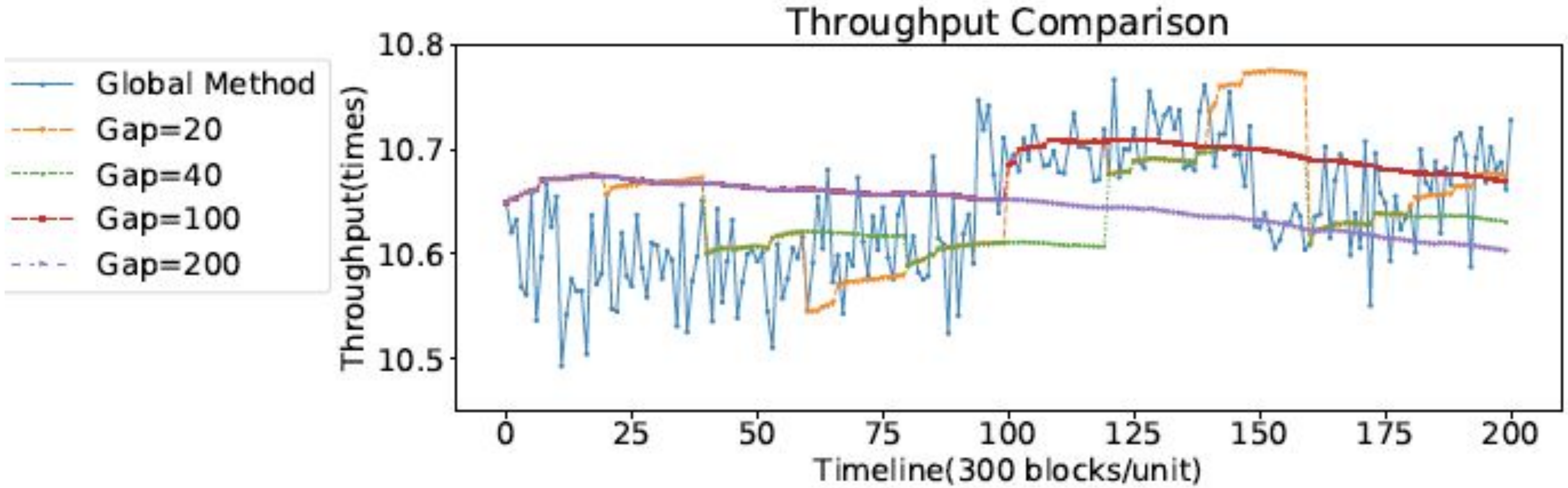






Key findings – Adaptive Algorithm

- ~0.5 seconds running time/ ~200s global algorithm,
- less than 1% performance loss than global algorithm.



Conclusions

Conclusions

- Convert to Community Detection on graph
- A fast and deterministic algorithm
- Directly optimize throughput on graph
- Significant performance improvement on ETH data

THANK YOU!