UC Santa Barbara

Computer Science Department

On Sharding Permissioned Blockchains

Mohammad Javad Amiri, Divyakant Agrawal, Amr El Abbadi

The second IEEE International Conference on Blockchain

China Aims to Use Blockchain

Nike, Telegram, Facebook, and Everyone Else Suddenly Love Blockchain

by P. H. Madore — 25/04/2019 in Bitcoin & Blockchain Investments, Blockchain News, Cryptocurrency Ne
3 min read

How important will blockchain be to the world's economy?

By Tim Harford
Presenter, 50 Things That Made the Modern Economy

3 July 2019 Share

BBC

3 key things to know about Facebook's Libra cryptocurrency project

By Clare Duffy, CNN Business
Updated 3:38 PM ET, Tue June 18, 2019

Anyone can participate without a specific identity

Participants are known and Identified

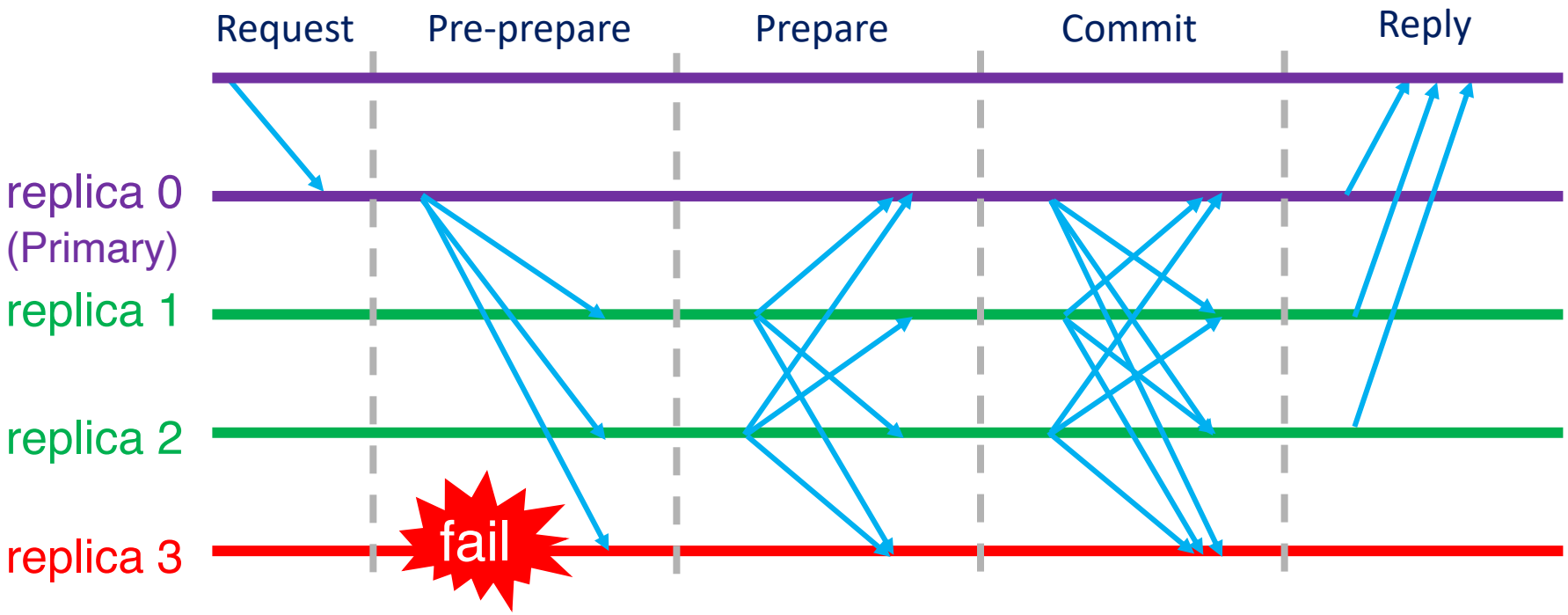
Permissionless Blockchain

Permissioned Blockchain

A Permissioned Blockchain system consists of a set of known, identified nodes that might not fully trust each other.

Practical Byzantine Fault Tolerance (PBFT)

- Provides **safety** in asynchronous system (with 3f+1 replicas)
- The algorithm has three main phases: (1) *pre-prepare* picks order of requests (2) *prepare* ensures order within views, (3) *commit* ensures order across views



What if $n \gg 3f+1$?!

Active/passive replication technique

3f+1 active replicas, others are passive and be informed about decisions

Fast Byzantine Agreement technique

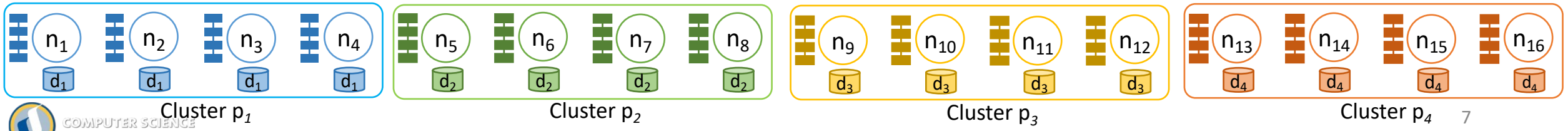
Use $5f + 1$ replicas to establish consensus and reduce one phase of communication in comparison to PBFT

Sharding Techniques

Partitioning the data into multiple shards that are maintained by different subsets (clusters) of nodes

Sharding Blockchains

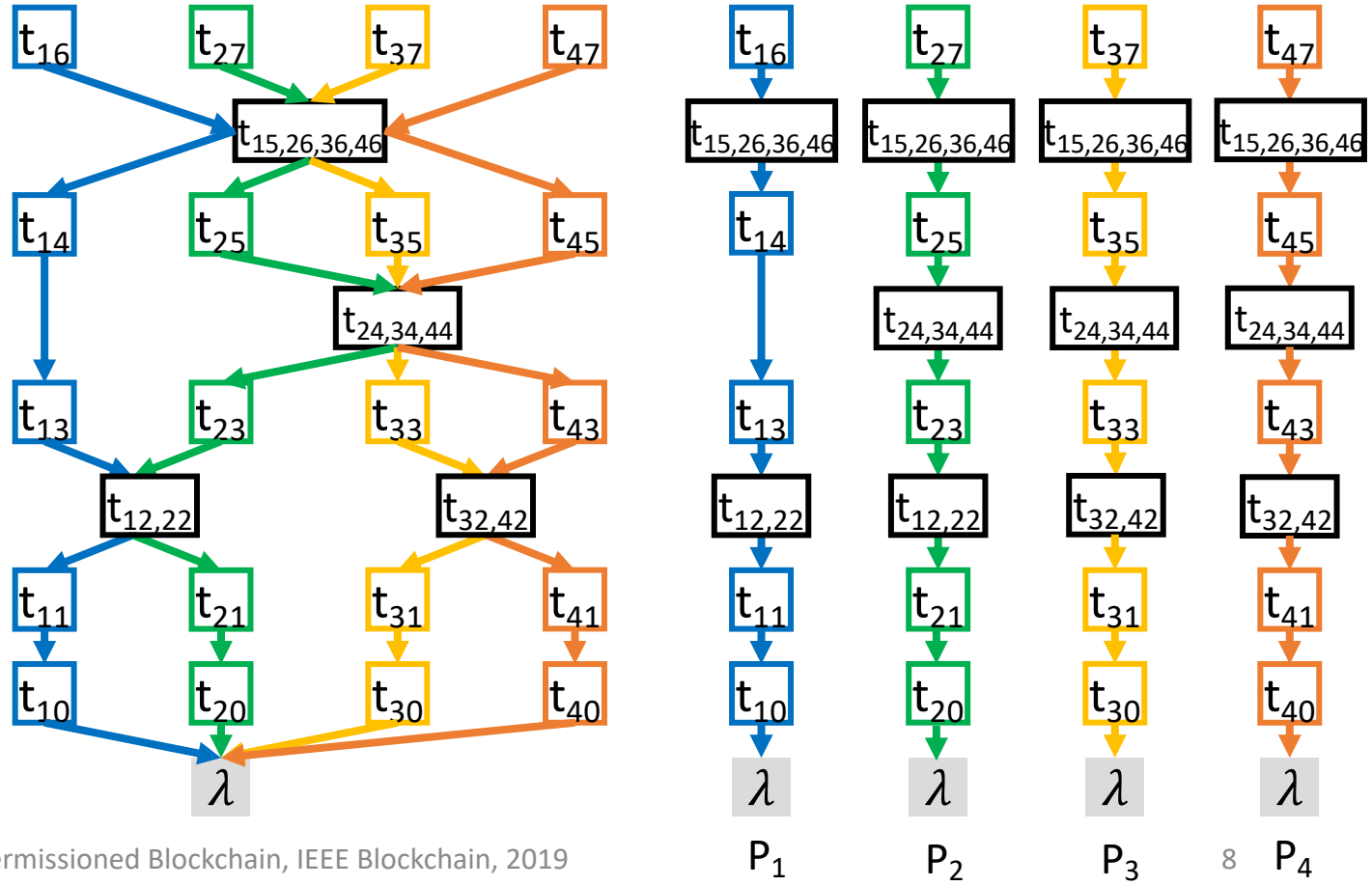
- Partition the network into clusters of *3f+1* nodes (to guarantee safety in each cluster in the presence of f malicious nodes)
- How to form clusters such that each cluster includes at most f faulty nodes?
  - Assign nodes to clusters in a random manner (uniform distribution): works if f is very large
  - Assume that N is much larger than 3f+1 (reasonable assumption in blockchain environment)
- Shard the data
  - Shard the application data and assign shards to clusters
  - Each data shard is replicated across nodes of a cluster
  - Different clusters process the transactions of their corresponding shard in parallel
  - The Blockchain ledger is also sharded
- Cross-Shard transactions
  - Needs participation of all (and only) involved clusters



Blockchain Ledger

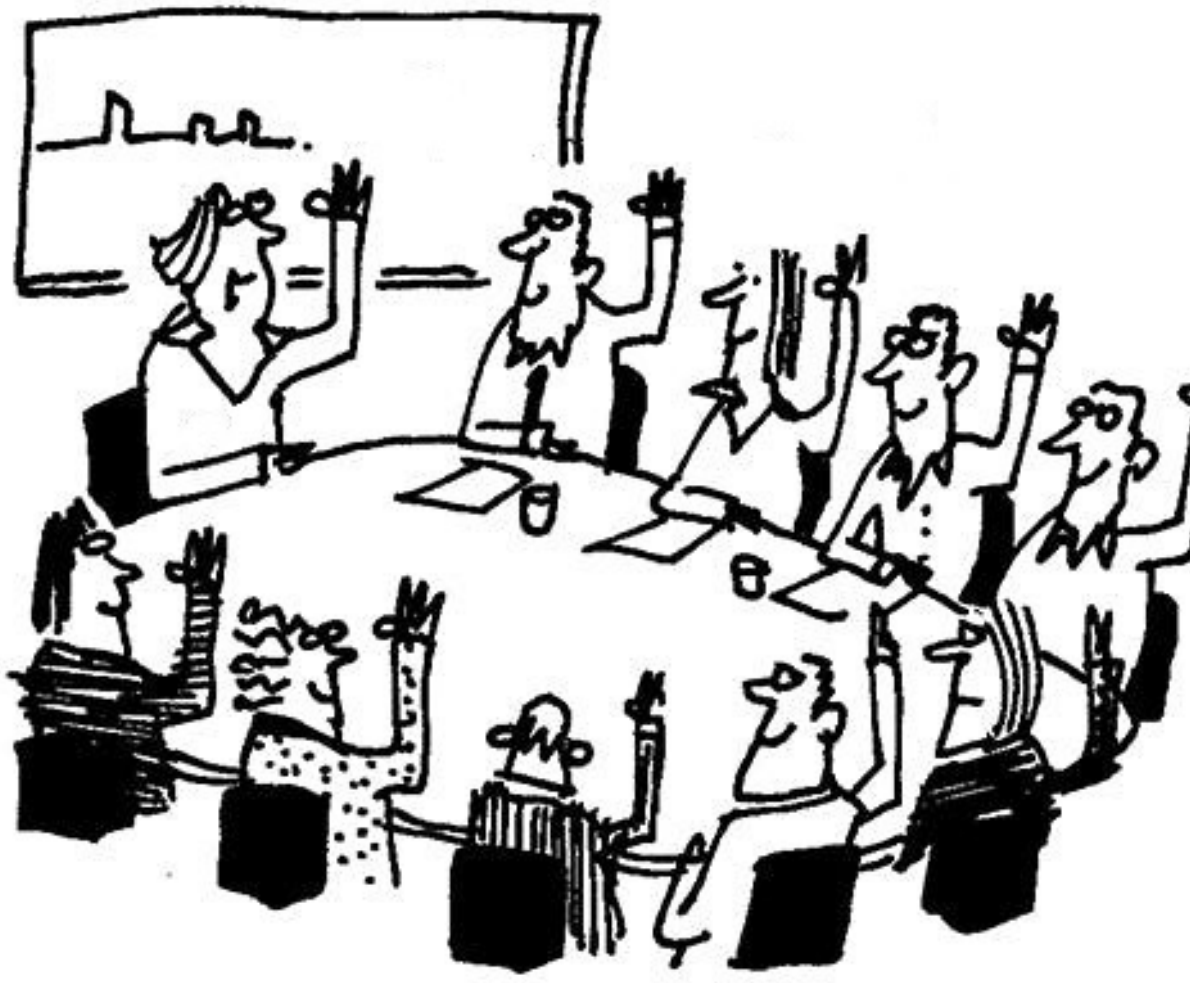
The blockchain ledger is generalized from a linear chain to a directed acyclic graph (DAG)

- The entire blockchain ledger is not maintained by any node
- Each node only maintains its own view of the blockchain ledger
- Each block includes a single transaction
- Intra-shard transactions of different clusters are processed in parallel
- Cross-shard transactions with non-overlapping clusters are processed in parallel
- A cross-shard transaction includes multiple hash pointers
- All clusters might be involved in a cross-shard transaction



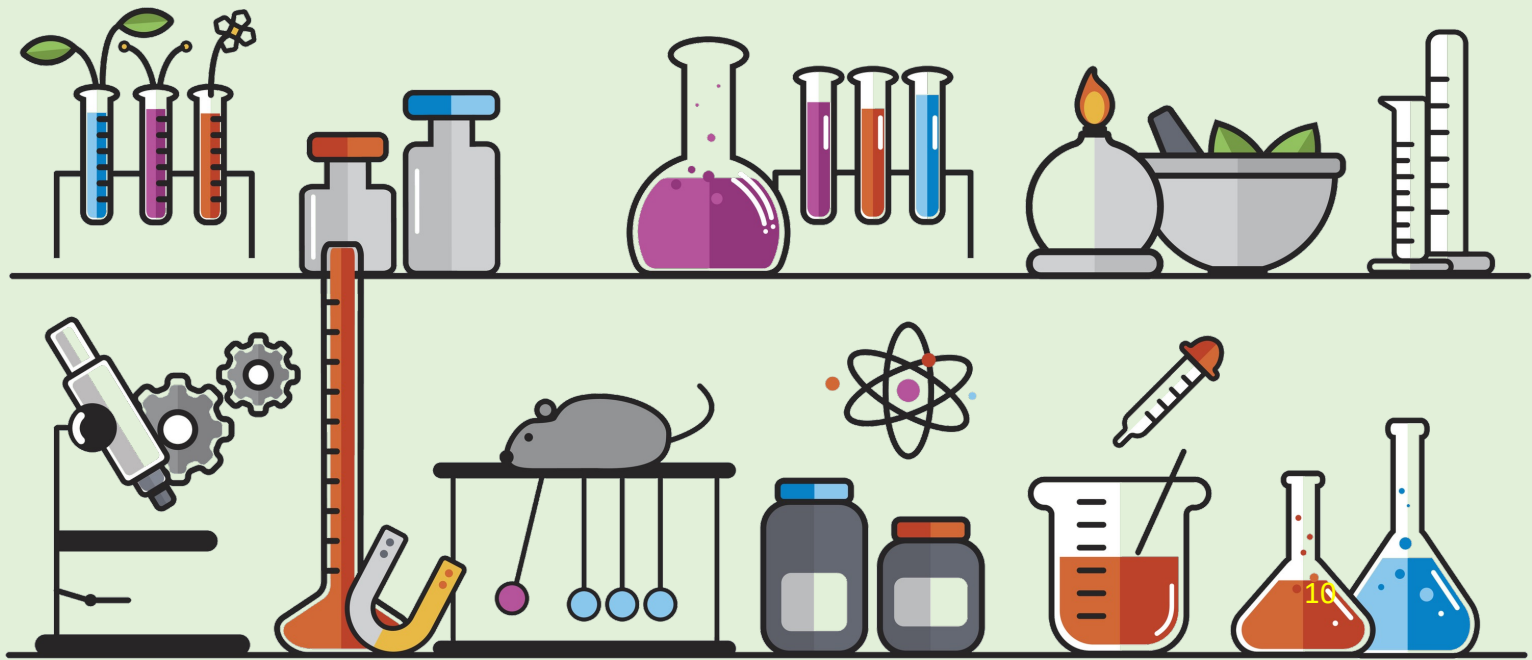
Intra- and Cross-Shard Consensus

- Intra-Shard Consensus: using any Byzantine fault-tolerant protocols
- Cross-Shard Consensus: needs the participation of all involved clusters
  - In each step 2f+1 nodes of every involved cluster must participate

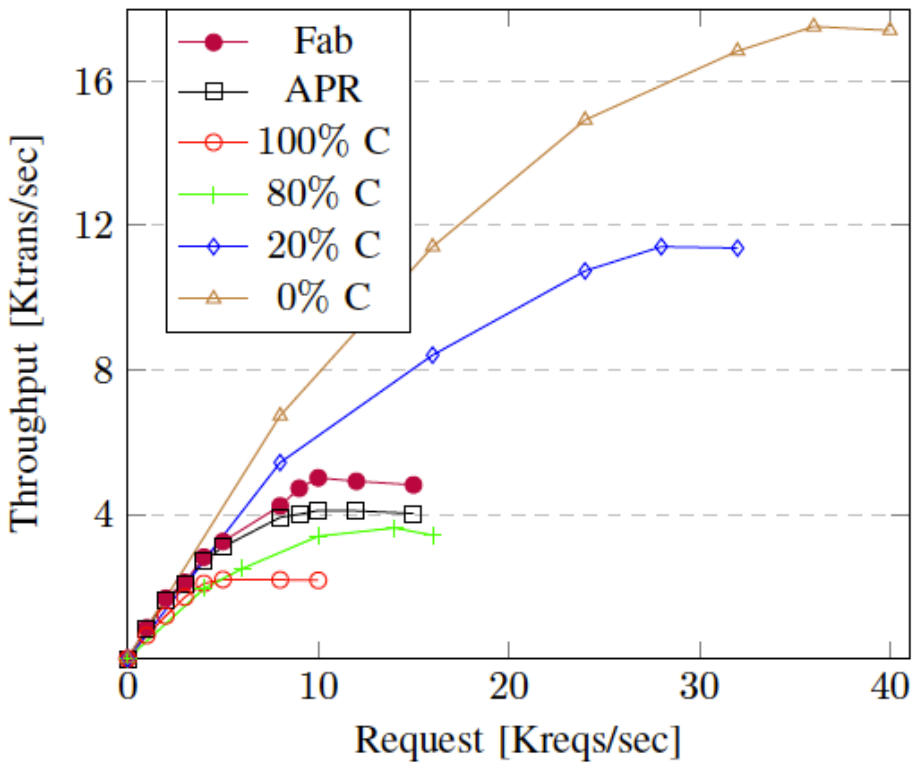


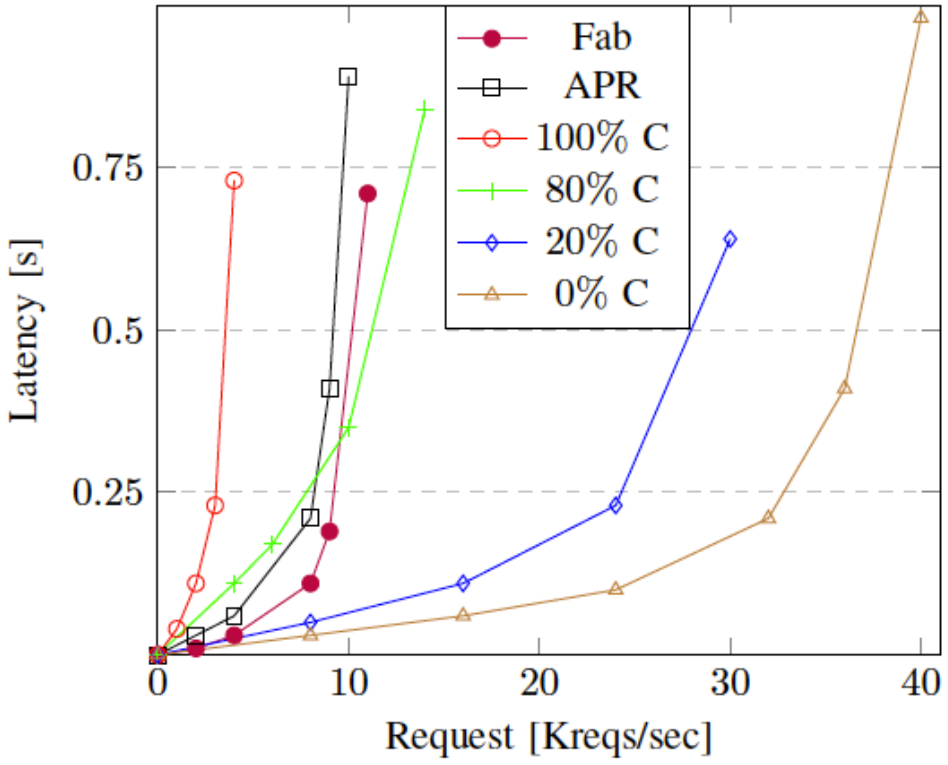
Experimental Settings

- Systems:
  - FaB: Fast Byzantine Protocol
  - APR: Active/passive replication
  - Sharding: Sharding technique
- Applications: Accounting
- Platform: Amazon EC2
- Measuring performance
  - Throughput
  - Latency



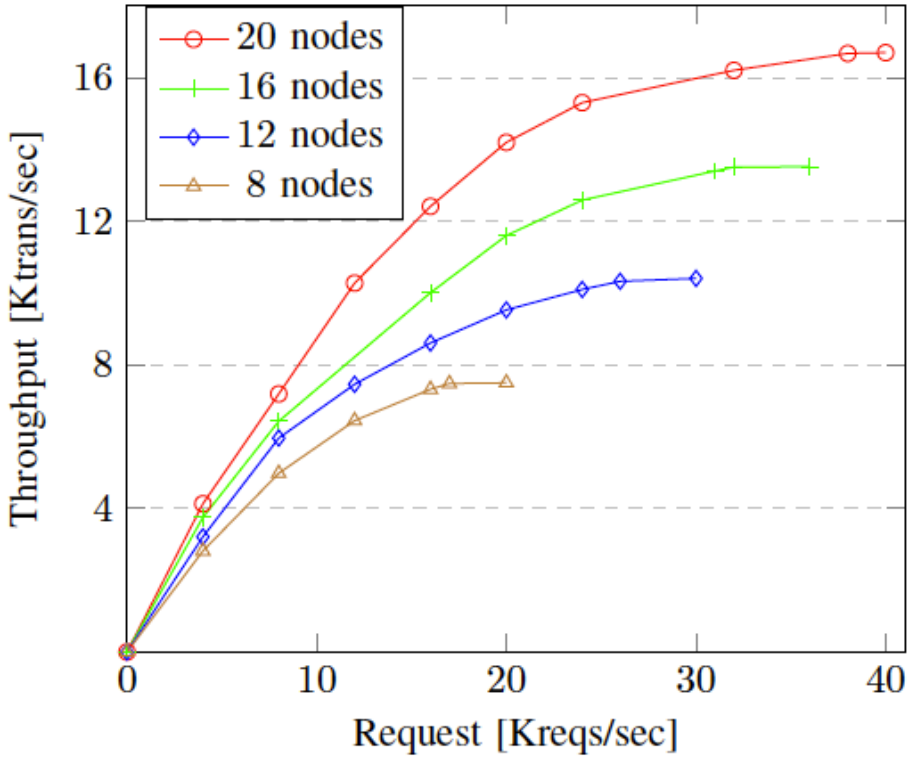
Performance with cross-shard transactions

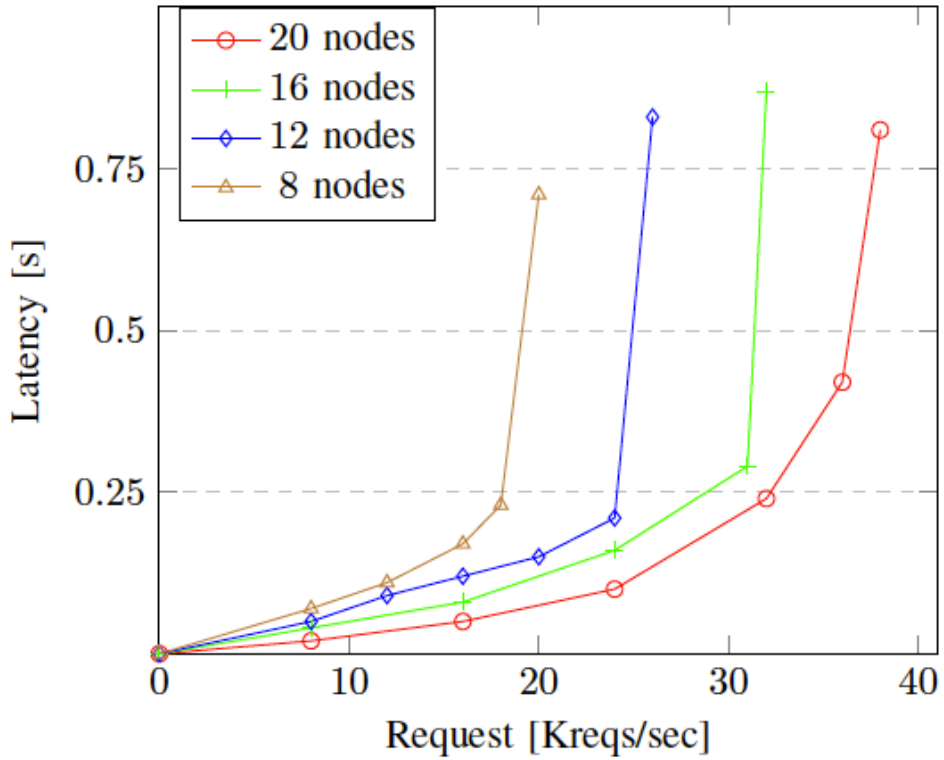




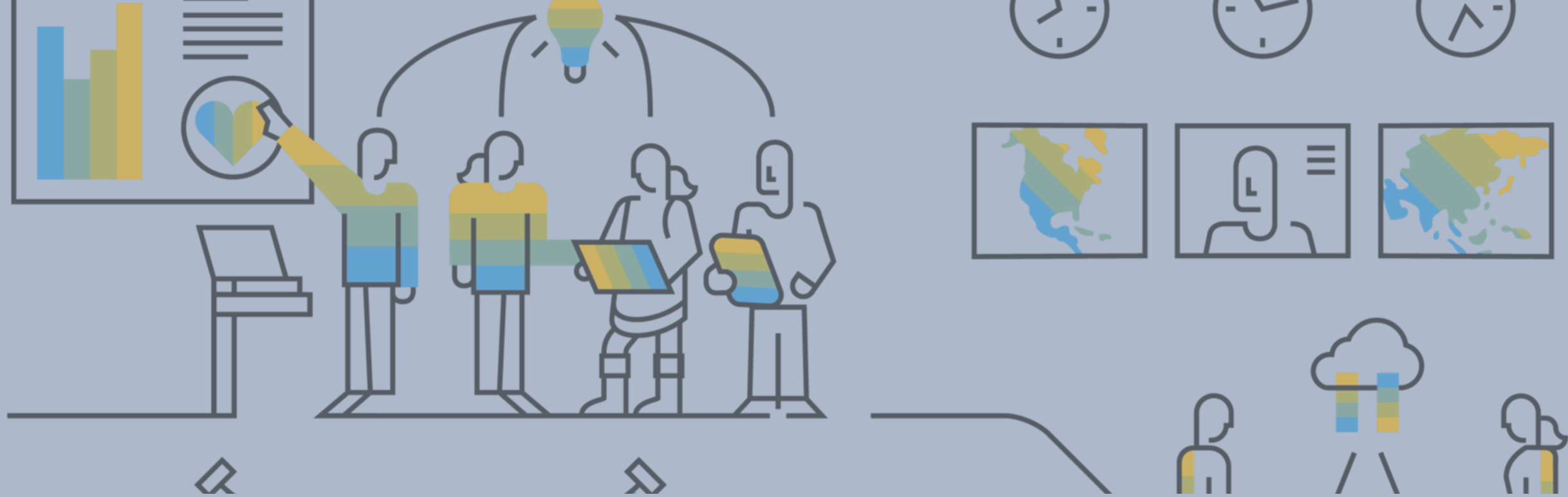
16 nodes (4 Clusters)
All clusters are involved in every cross-shard transactions

Performance with different number of nodes



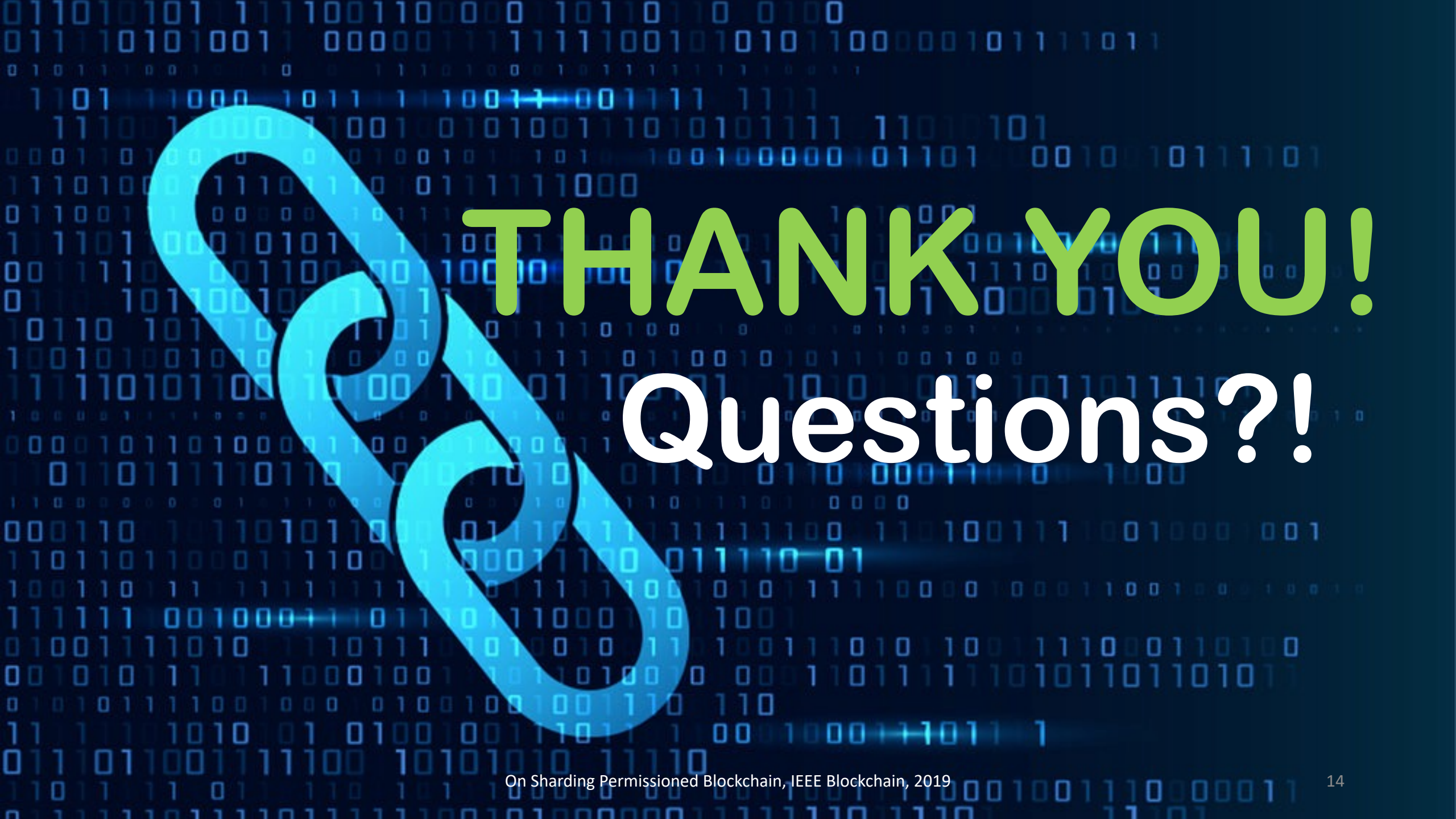


The workload includes 10 % cross-shard transactions
All clusters are involved in every cross-shard transactions



Future Work

- Designing consensus protocols to support cross-shard transactions
- Preserving the confidentiality of the blockchain ledger
- Leveraging Transaction Parallelism to enhance performance


THANK YOU!
Questions?!