Rashnu

Data-Dependent Order-Fairness

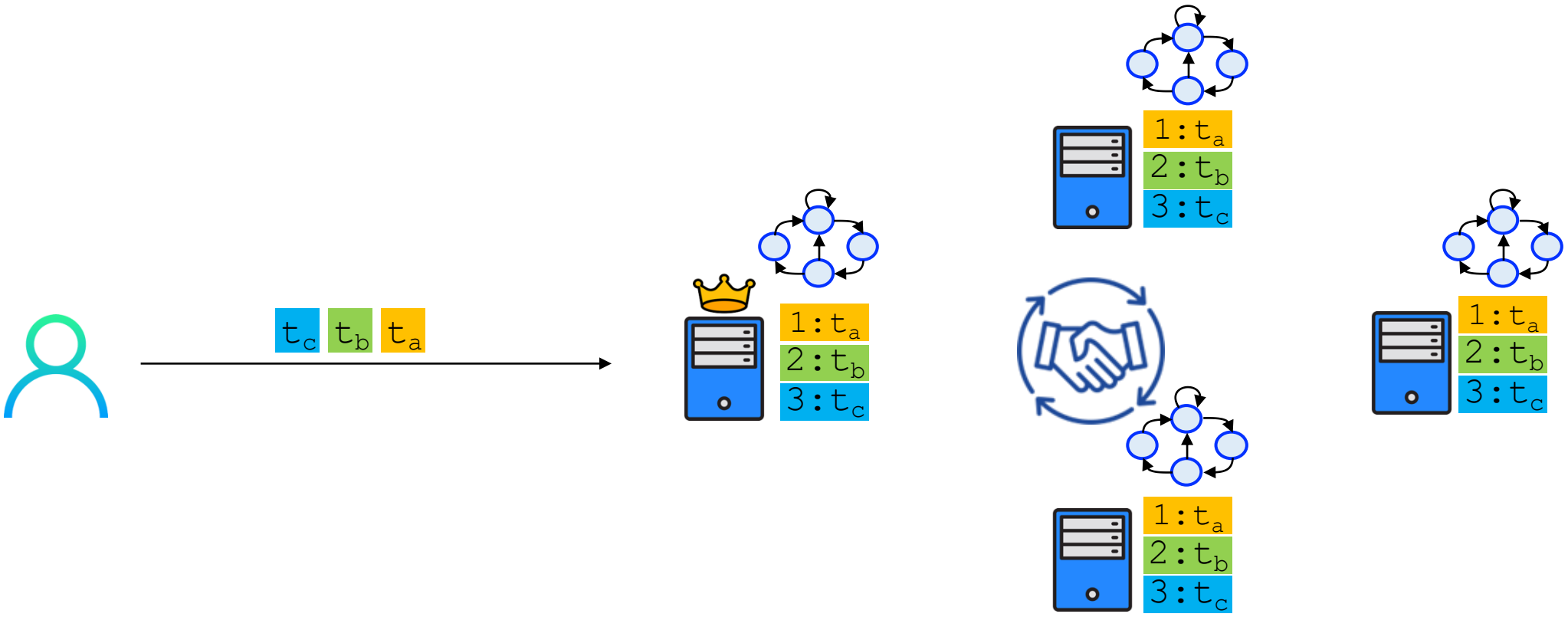
Heena Nagda
Shubhendra Pal Singhal
Mohammad Javad Amiri
Boon Thau Loo

U of Pennsylvania
Georgia Tech
Stony Brook University

Rashnu is the Avestan name of the Zoroastrian deity of justice

VLDB2024
GUANGZHOU

Distributed transaction processing

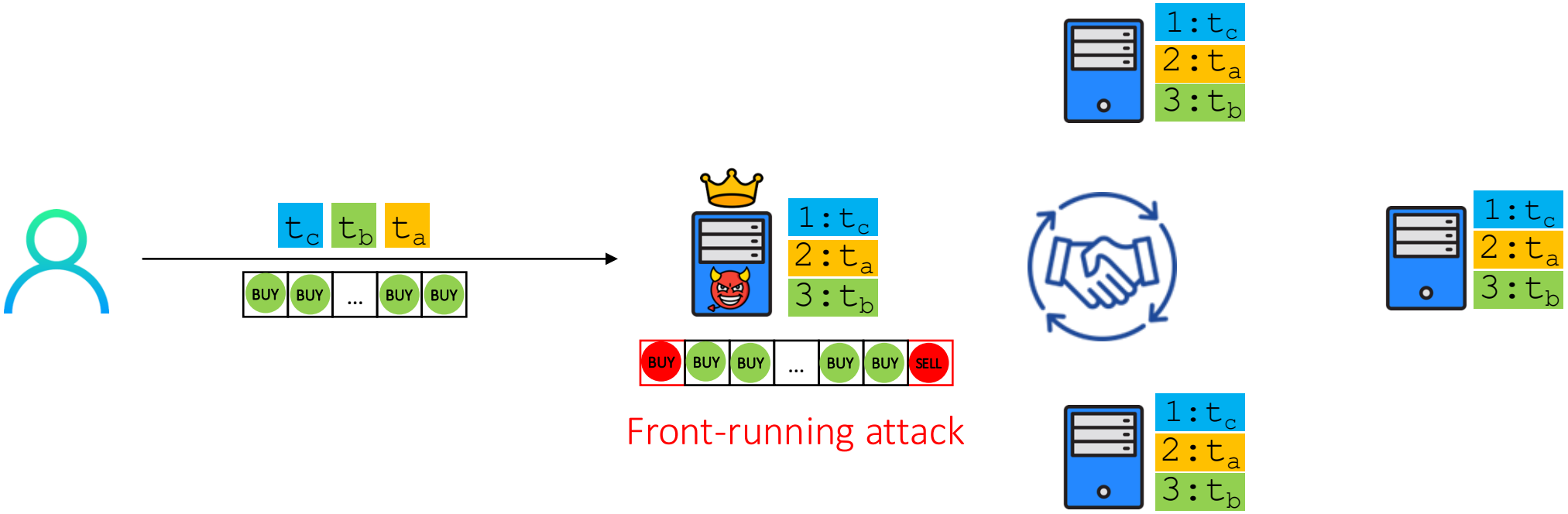


State Machine Replication: a replicated service whose state is mirrored across different deterministic replicas

- Assign each client request an order in the global service history and execute it in that order

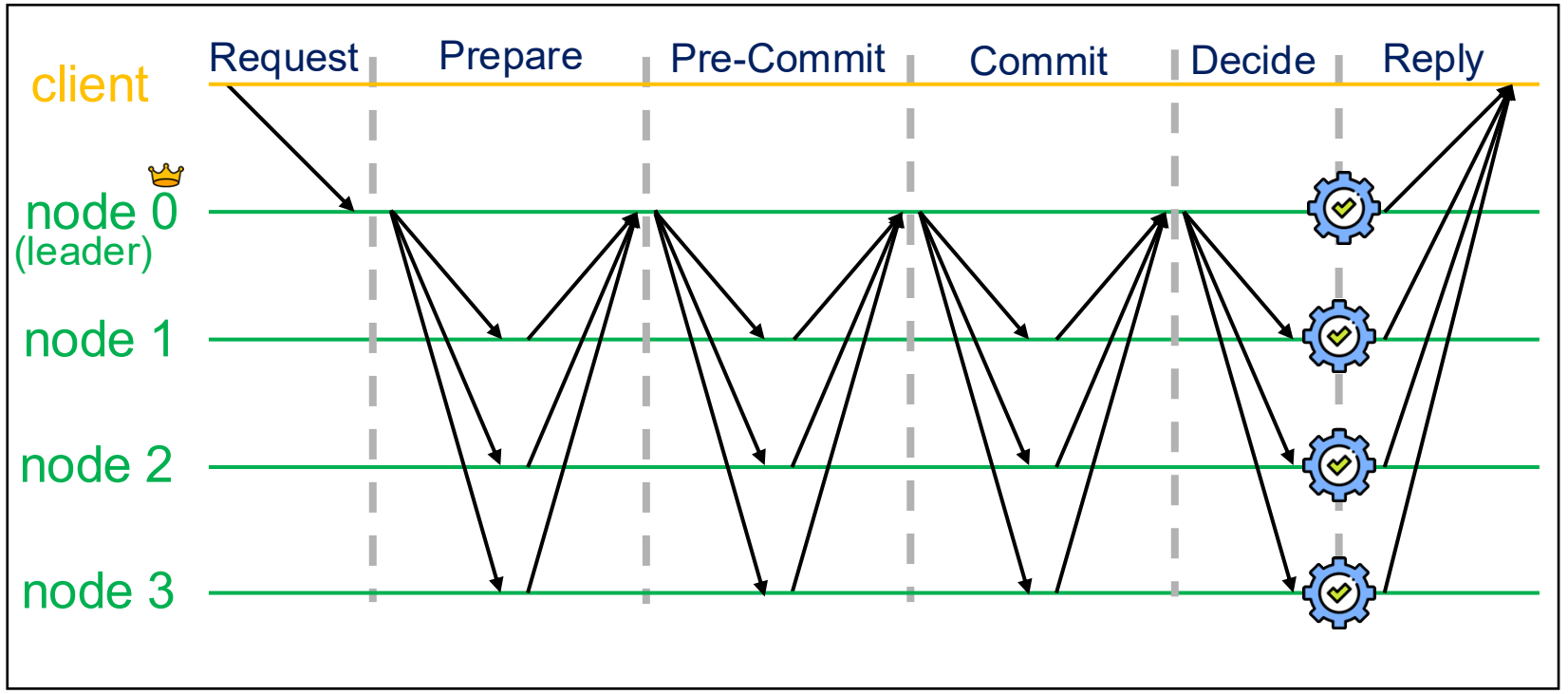
Is consensus on the "order of transactions" enough?

Why is order-fairness needed?

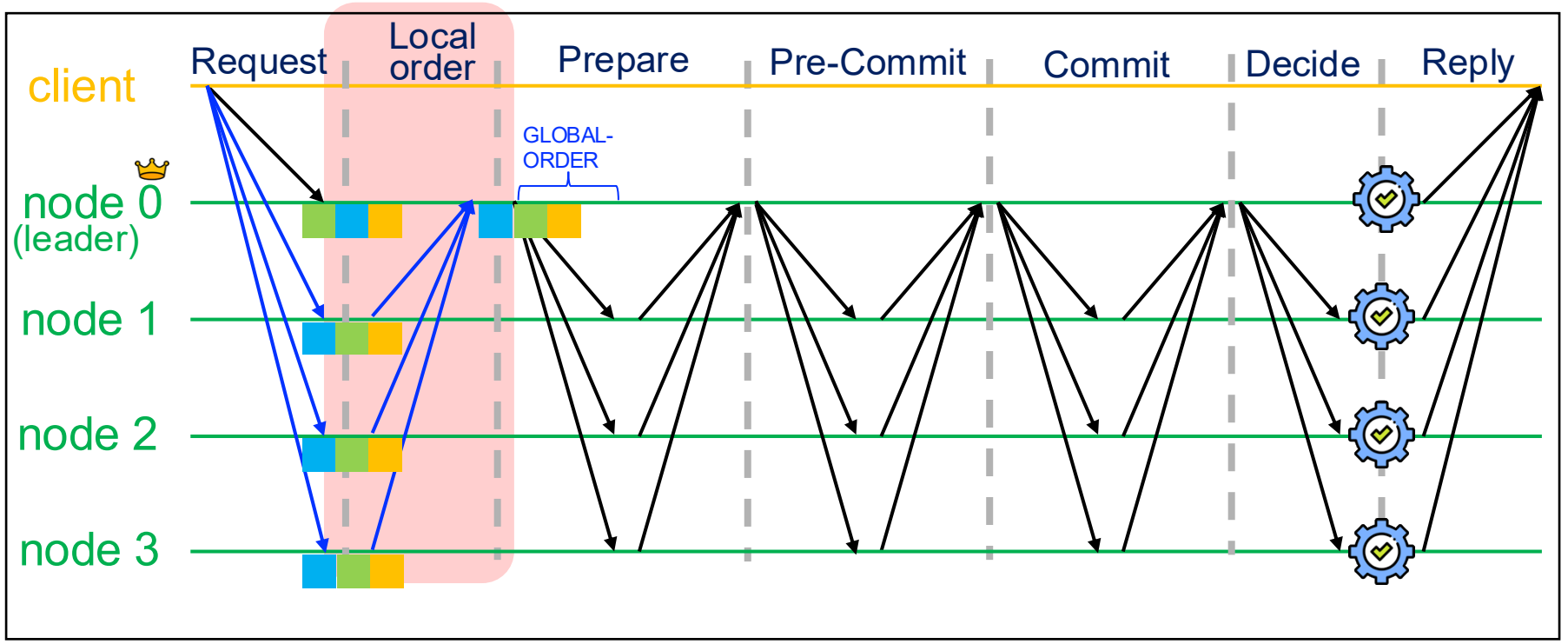


- Maximal Extractable Value (MEV): proposers make profit by including, excluding, or re-ordering transactions within blocks
- Extracting more than $686m in Ethereum from unsophisticated users
- $1.38B across all EVM powered networks

HotStuff Protocol



Fair HotStuff Protocol



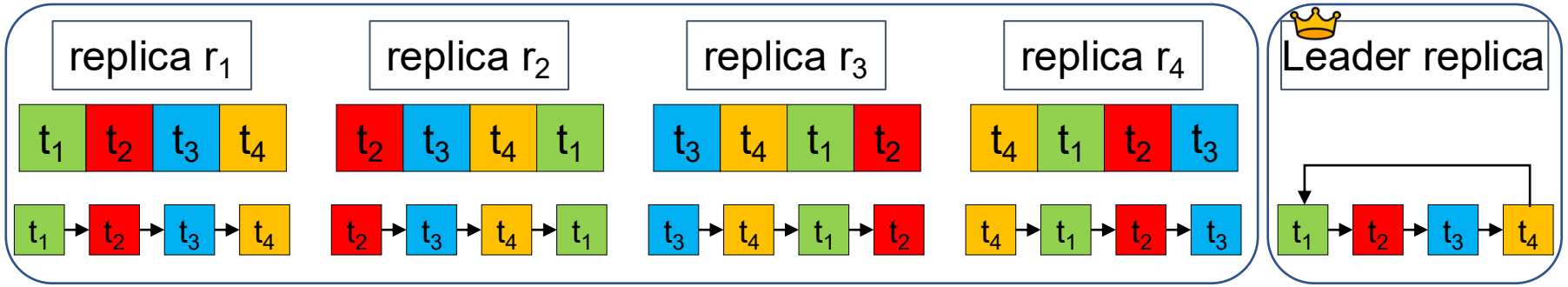
Receive order-fairness

- Order-fairness parameter γ: the fraction of replicas that receive transactions in a particular order (between 0.5 and 1)

- Receive order-fairness: If γ-fraction of replicas receive t before t', all honest nodes will deliver t before t'

Condorcet paradox

- In an asynchronous network replicas might receive transactions in different orders.



- Batch order-fairness: If γ-fraction of replicas receive t before t', no honest replica outputs t' before t.

Is order fairness practical?

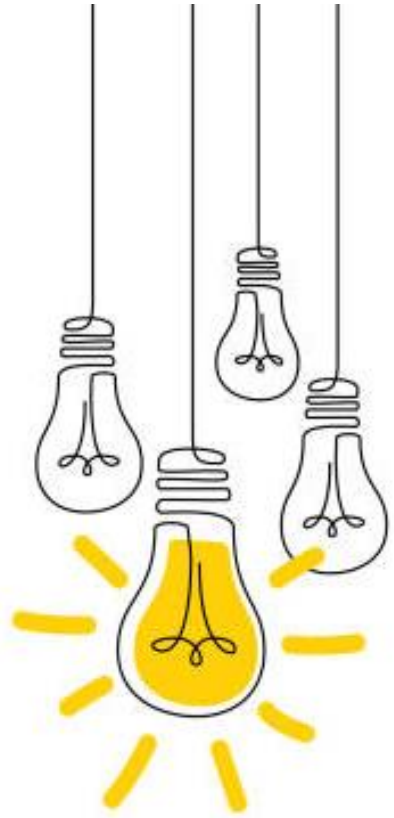
Poor performance caused by the time required to generate a fair order

- Different replicas might receive transactions in different orders
- Transactions might be arbitrarily delayed
- Byzantine replicas might send maliciously manipulated local ordering to the leader.
- Collecting the local ordering of different replicas might lead to cycles

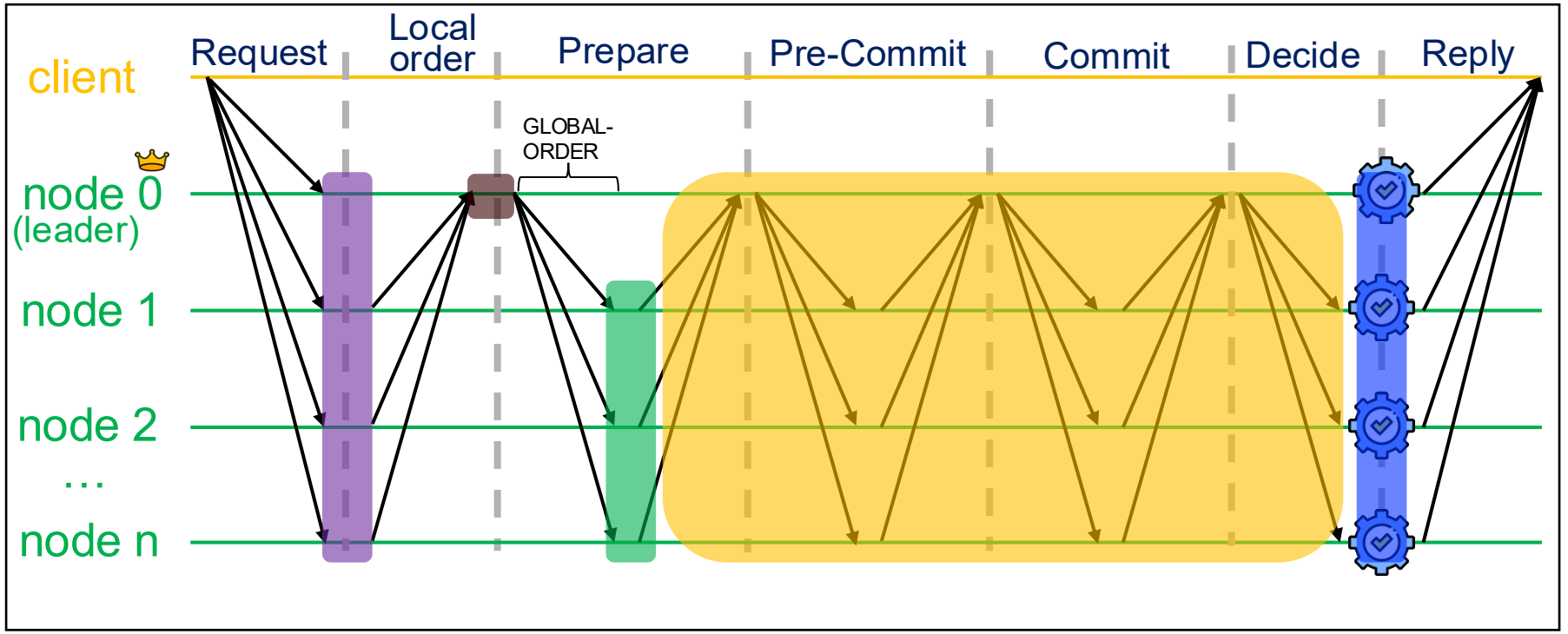
Data-dependent order-fairness

- Fair transaction ordering is essential when transactions access a shared resource and manipulating the order gives an unfair advantage to some transactions
- Two transactions t and t' are data-dependent if they access the same data object and one performs a write operation on the data object.

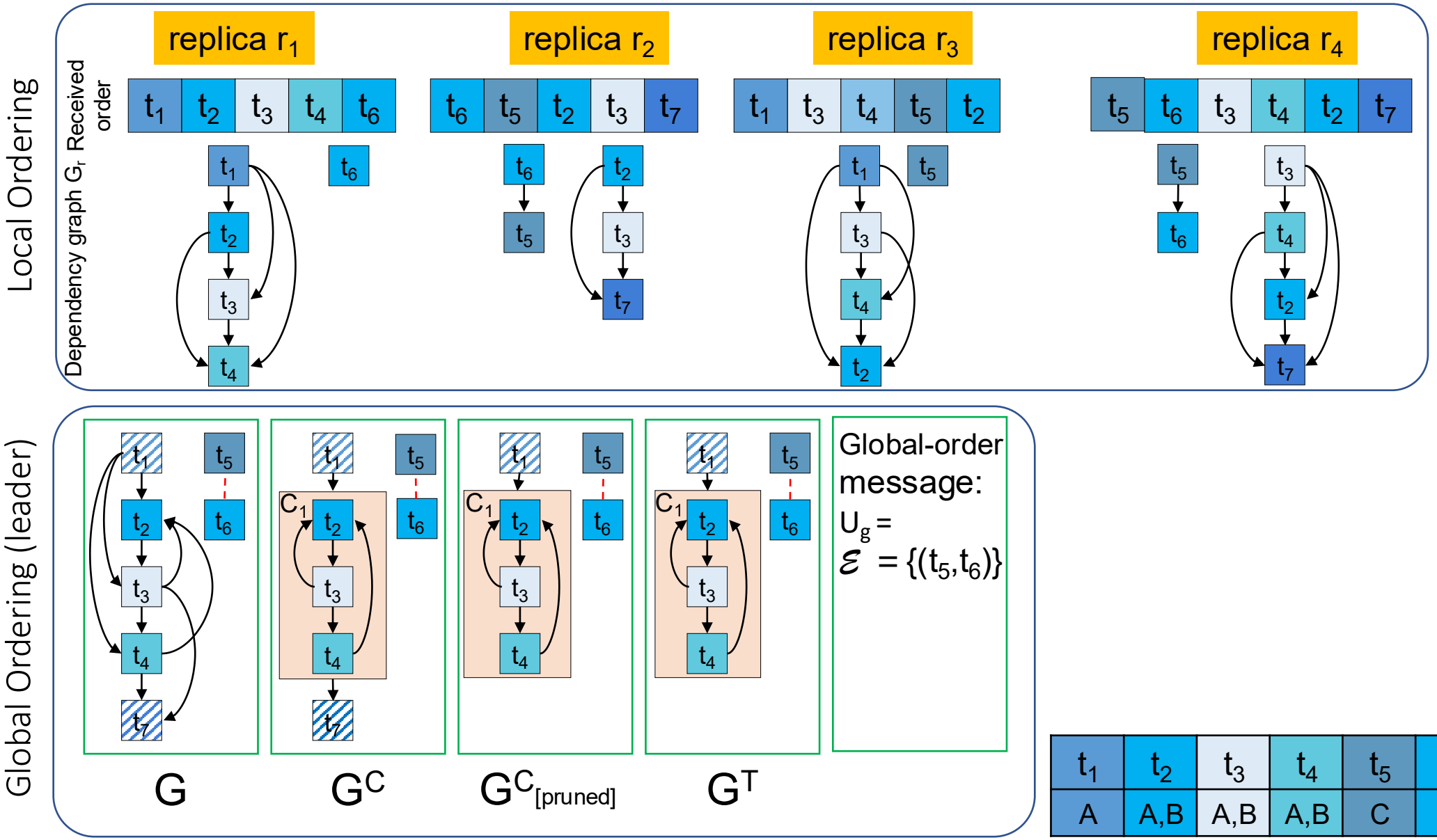
If t and t' are data-dependent and γ-fraction of replicas receive t before t', no honest replica outputs t' before t.



Rashnu algorithm



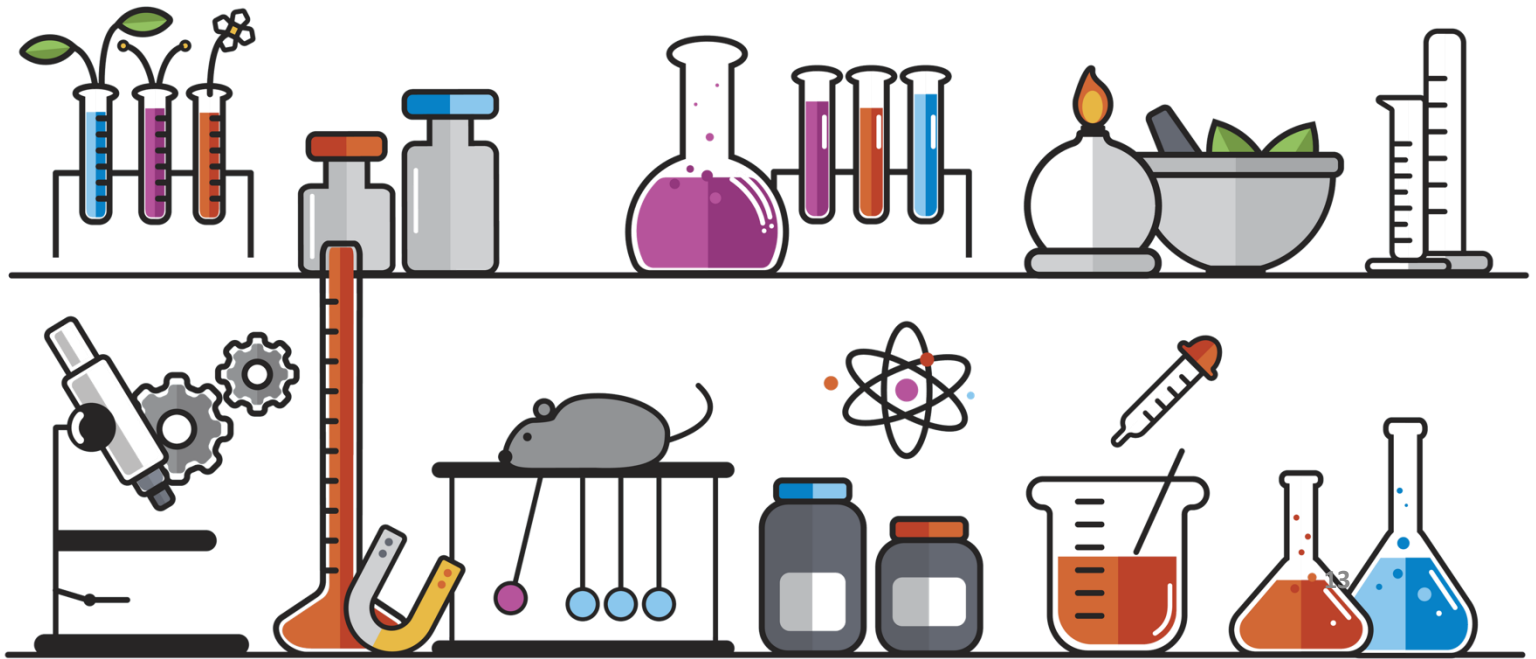
Rashnu algorithm



t_1	t_2	t_3	t_4	t_5	t_6	t_7
A	A,B	A,B	A,B	C	C	A,B

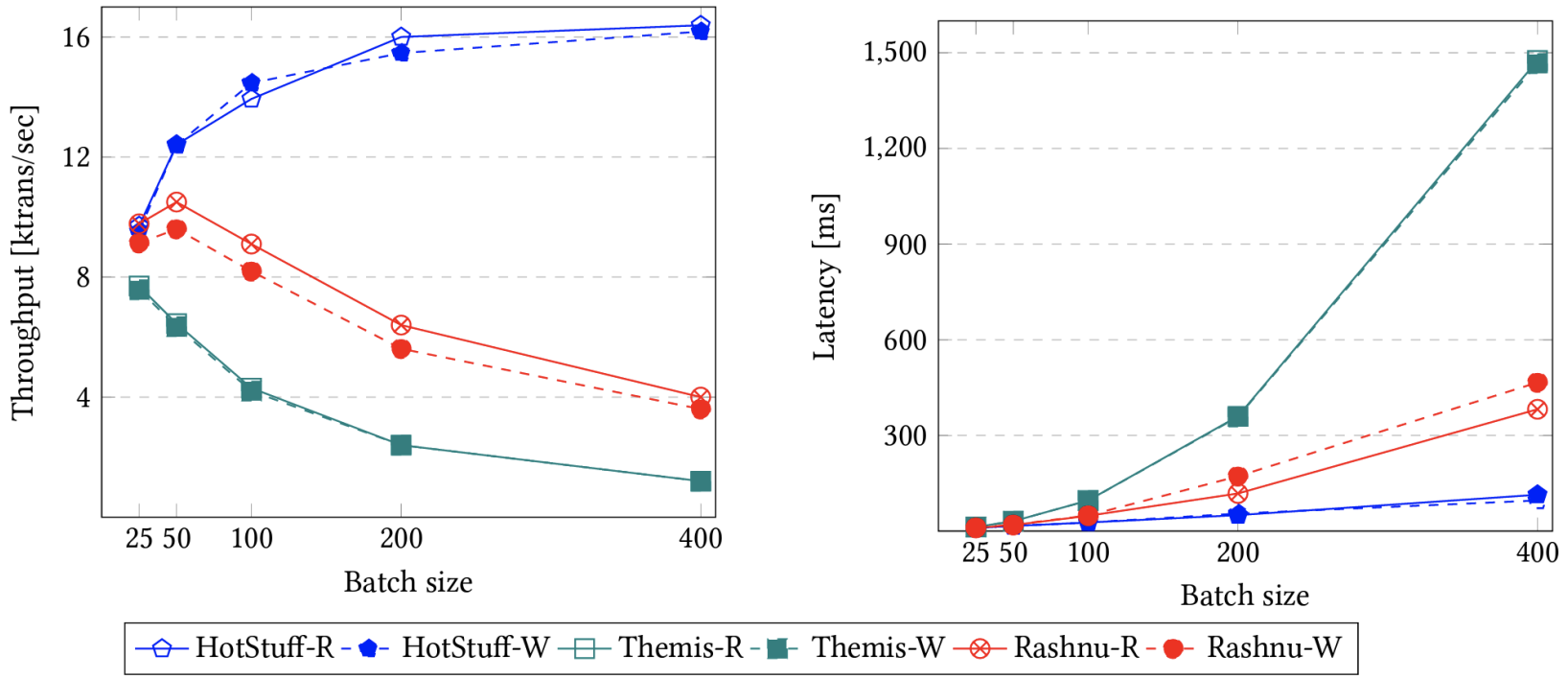
Experimental settings

- Platform: CloudLab
- HotStuff, Themis and Rashnu [read-heavy (Pw = 0.05) and write-heavy (Pw = 0.95)]
- Workload: SmallBank and YCSB
- Measuring performance
  - Throughput
  - Latency



Impact of batch size

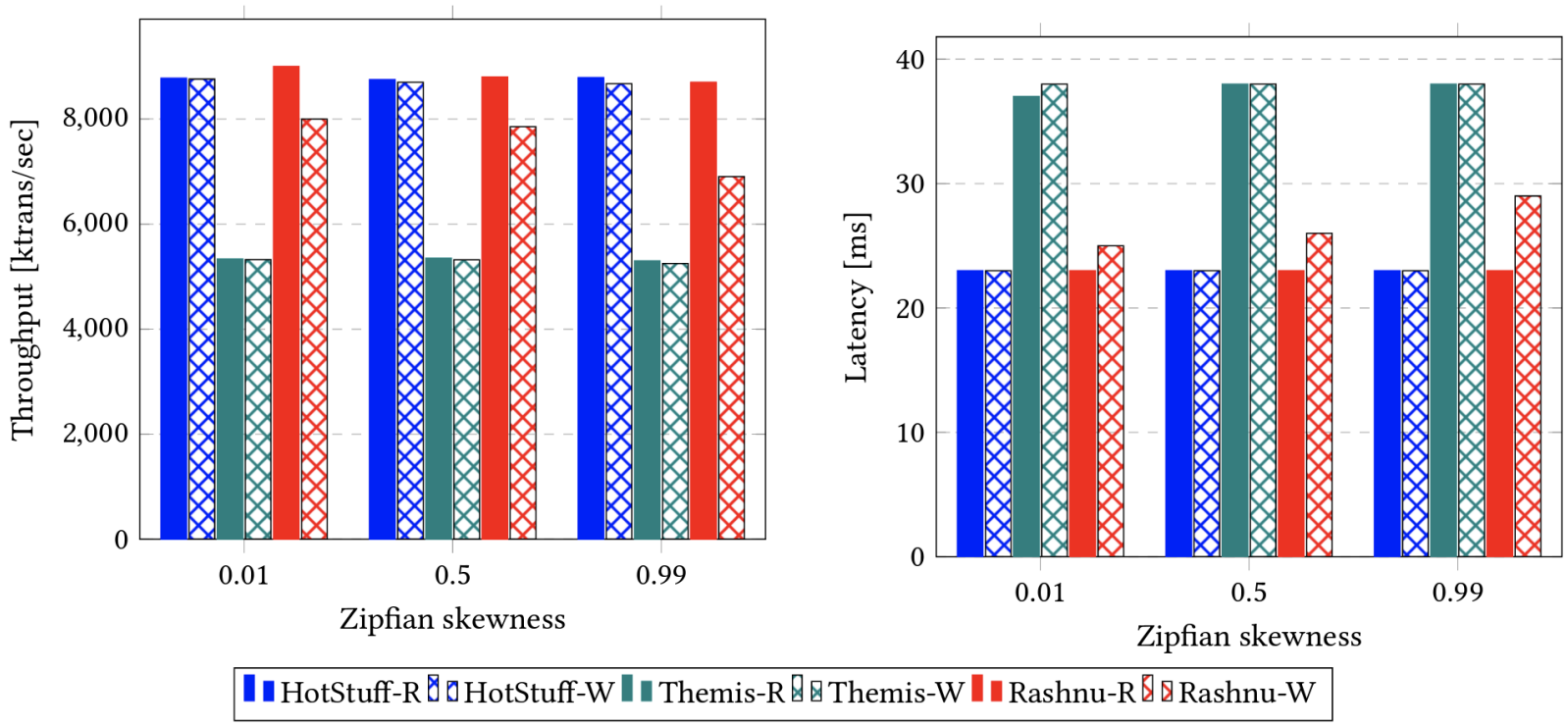
n = 5
γ = 1
Uniform distribution



Small blocks (block size = 25): Rashnu provides fairness with zero overhead.
Large blocks (400) : 233% higher throughput and 74% lower latency compared to Themis.

Performance with compute-intensive workloads

n = 5
γ = 1
Zipfian distribution
Execution time: 10 μs
Eight threads



Rashnu is able to execute transactions in parallel
Throughput gain of parallel execution outnumbers the overhead caused by dependency graph generation

Summary

- The notion of data-dependent order-fairness has been defined.
- Rashnu ,a high-performance fair-ordering protocol is presented.
- Requires minimal modification to any existing leader-based BFT protocol.
- Adds a single communication step to the existing protocol: Replicas submit the local ordering to to the leader.
- Rashnu shows significant performance improvement compared to Themis in different settings, especially in small networks.


Questions?