

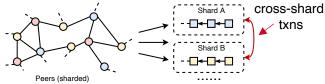
A Scalable Consortium Blockchain Exploiting Intra- and Inter-Block Concurrency



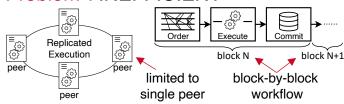
Zhihao Chen, Haizhen Zhuo, Quanging Xu, Xiaodong Qi, Chengyu Zhu, ANTCHAIN Zhao Zhang, Cheging Jin, Aoying Zhou, Ying Yan, Hui Zhang

Background and Motivation

- Consortium blockchain needs more transaction processing to support large-scale businesses in enterprise collaborations
- Existing solutions such as sharding¹ and incorporating concurrency² introduce new problems



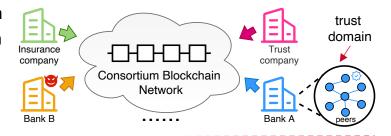
Problem¹: INEFFICIENT



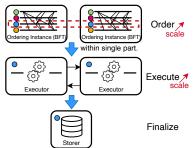
Problem²: LIMITED & QUIESCENT

SChain Solution

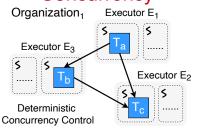
- Scale the consortium blockchain in terms of each participant based on TRUST DOMAIN
- Exploit Intra- and Inter-Block concurrency



Design 1: Scalable **Paradigm**

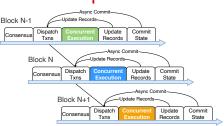


Design 2: Intra-block Concurrency



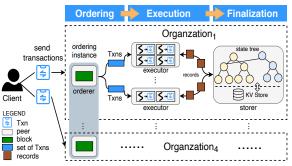
- Execute in parallel among all executors
- Concurrently within a single executor

Design 3: Inter-block **Pipeline**



- Non-quiescent workflow
- Resource-efficient pipeline

Demonstration Details



Demo Implementation

1) Gain Insight into

Intra-Block Concurrency Inter-Block Concurrency

 distributed concurrent execution sendPayment::ReadSet sendPayment::WriteSet - KeyA Local Collected - KevA » Tx 54767 (Ex. 2) - KeyB ≪ <u>Tx 546</u>50 **⊘** - KevB » Tx 54735 (Ex. 1) ✓ Executor 2 Contract - Test Contract updateBalance::ReadSet updateBalance::WriteSet - KeyA ≪ Tx 54728 ♥ - KeyA ⊝ Pending Execution

KeyC Lock Not Acquired

2) Investigate into

- transaction streaming pipeline

