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

Zhihao Chen, Haizhen Zhuo, Quanqing Xu, Xiaodong Qi, Chengyu Zhu, Zhao Zhang, Cheqing Jin, Aoying Zhou, Ying Yan, Hui Zhang

chenzh@stu.ecnu.edu.cn

華東師範大學
EAST CHINA NORMAL UNIVERSITY

蚂蚁集团
ANT GROUP

蚂蚁链
ANTCHAIN

Introduction

- Blockchain provides data integrity, traceability and immutability to tackle trust problems among mutually distrusting parties

- **Consortium blockchain** is being widely applied to support large-scale businesses in enterprise collaborations

Introduction

- As users and applications of blockchain proliferate, the system has to **scale** to provide more transaction processing

  1. exploit the parallelism of network, i.e **sharding**
  2. enhance the capability of every **single participant**

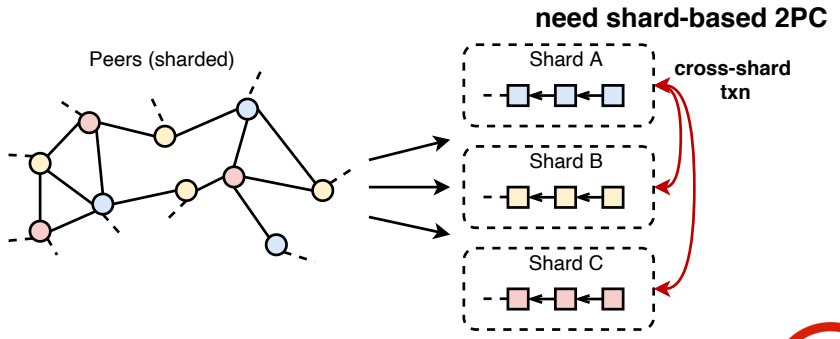


Fig.1: Sharding technique

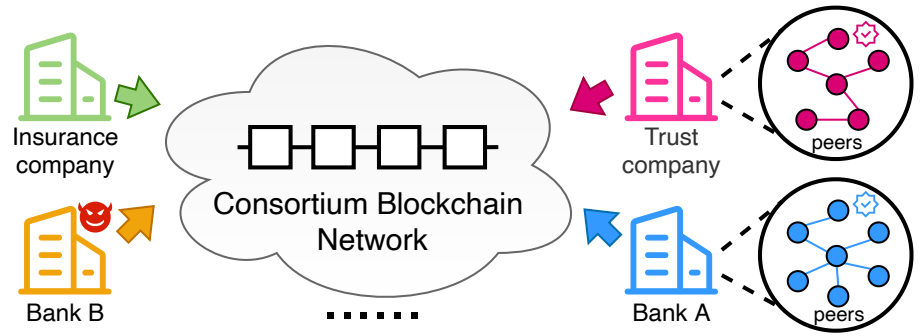


Fig.2: Enhance single participant

Cross-shard txn incurs **a large number of** intra- and cross-shard communications

Scale the consortium blockchain in terms of each participant **based on trust domain**

Background

To empower the individual participant

- Fabric incorporate concurrency
  - **High abort rates** for hotspot workloads
  - Enhanced works still inherits the limitations of serial validation

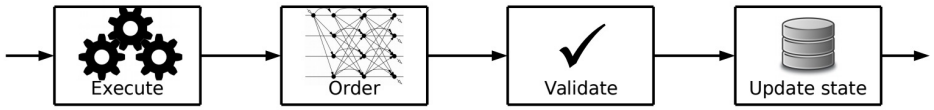


Fig.3: execute-order-validate paradigm

- ParBlockchain and BlockchainDB parallelize the execution
  - Allow non-conflicting transactions to execute in parallel

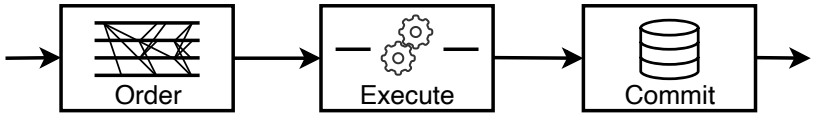


Fig.4: order-execute paradigm

1. **Limited** to single peer
2. **Overlook** transaction parallelism **across** multiple blocks

SChain Overview

- System Architecture
  - **Scalable** order-execute-finalize (SOEF) paradigm
  - Hybrid trust and fault assumptions
  - Exploit **Intra-** and **Inter-Block** concurrency

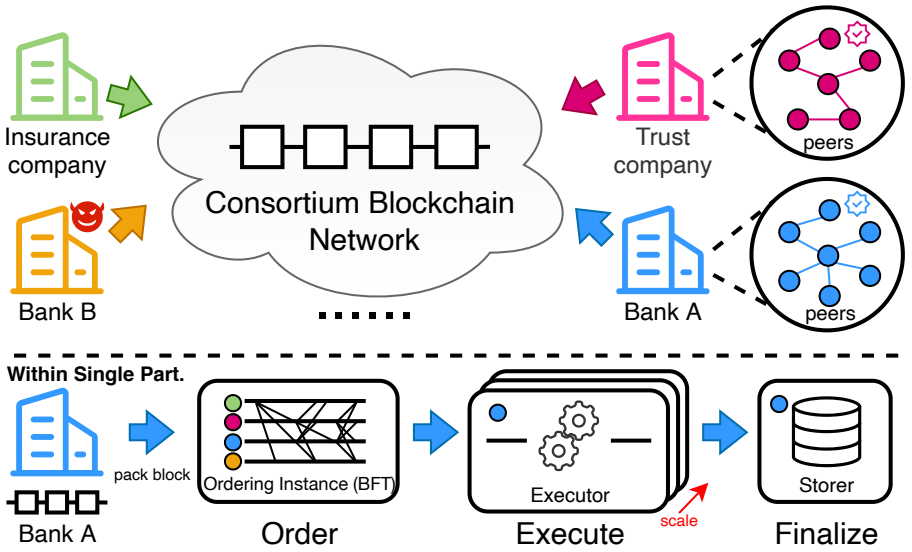


Fig.5: Scalable order-execute-finalize paradigm

SChain's Intra-Block Concurrency

- Multiple executors
  - **Deterministic** concurrency control
  - Early read/write **keys** acquisition for Turing-complete smart contract
  - Guarantee the **merge** of execution result is **equivalent** to the predetermined serial order

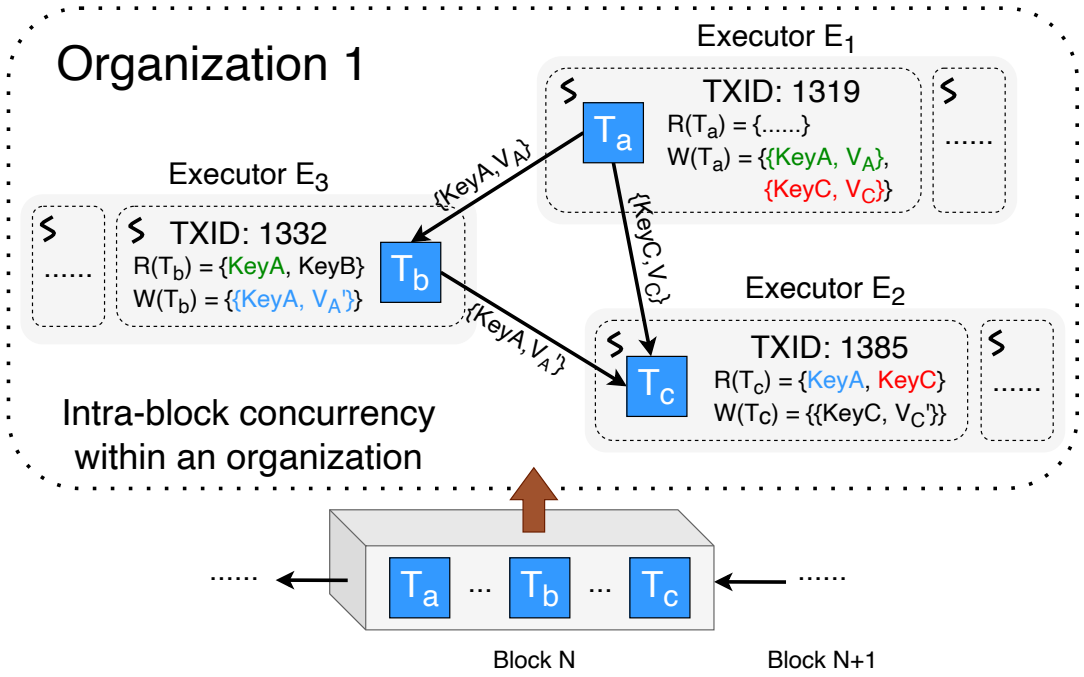


Fig.6: Intra-Block Concurrency

defined by ordering phase

Transactions are executed
- **Execute in parallel** among all executors ✓
- **Concurrently** within a single executor ✓

SChain's Inter-Block Concurrency

- Pipelined workflow
  - **Interleave** workflows for different blocks

    -> no longer block-by-block quiescently

  - Explore the **inter-block concurrency**

    -> allow txns in later blocks to be executed **earlier**

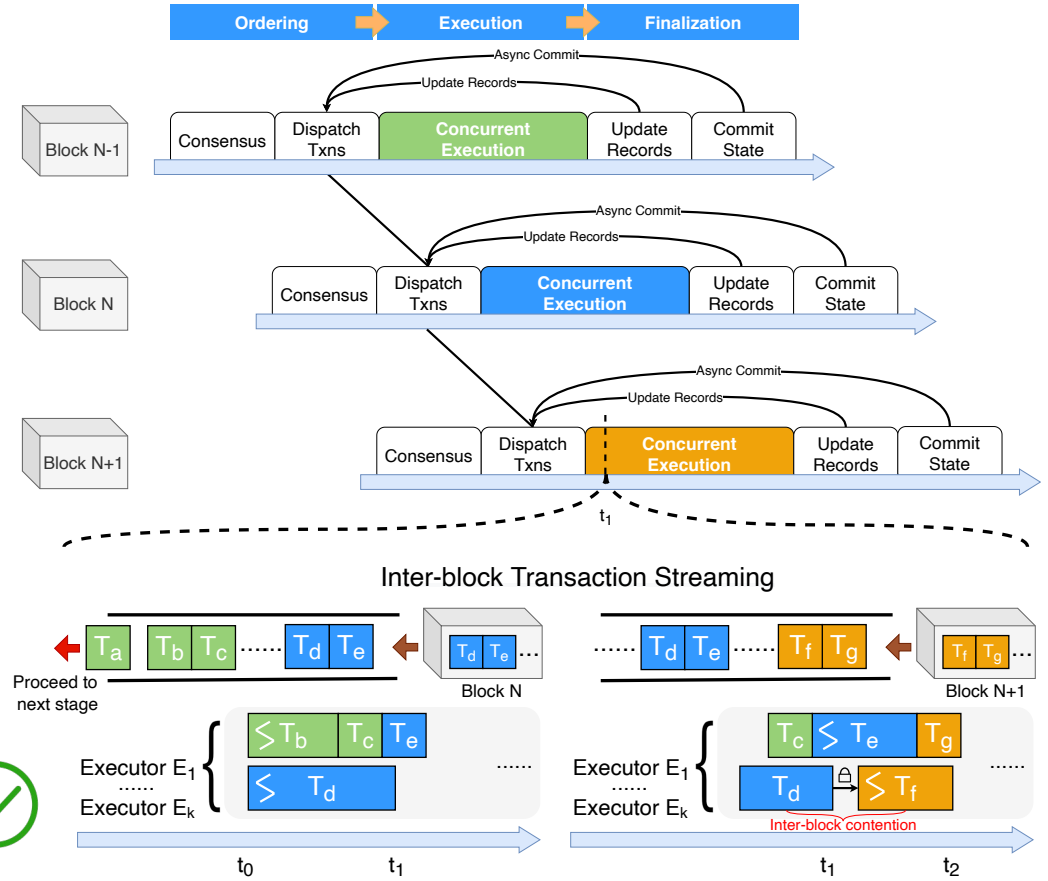


Fig.7: Inter-Block Concurrency

Non-quiescent workflow ✓ **Inter-Block** concurrency ✓

Fully-utilized resources ✓

SChain's Scalability

- Ordering ✓
  - **Merely order** the transactions
  - Concurrent instances (easily get a **global order** due to trust domain)

follow rules

- Execution ✓
  - Devote more executors **on demand**

- Finalization ⊖
  - Complexity of state partition
  - Expect to design a scalable storage

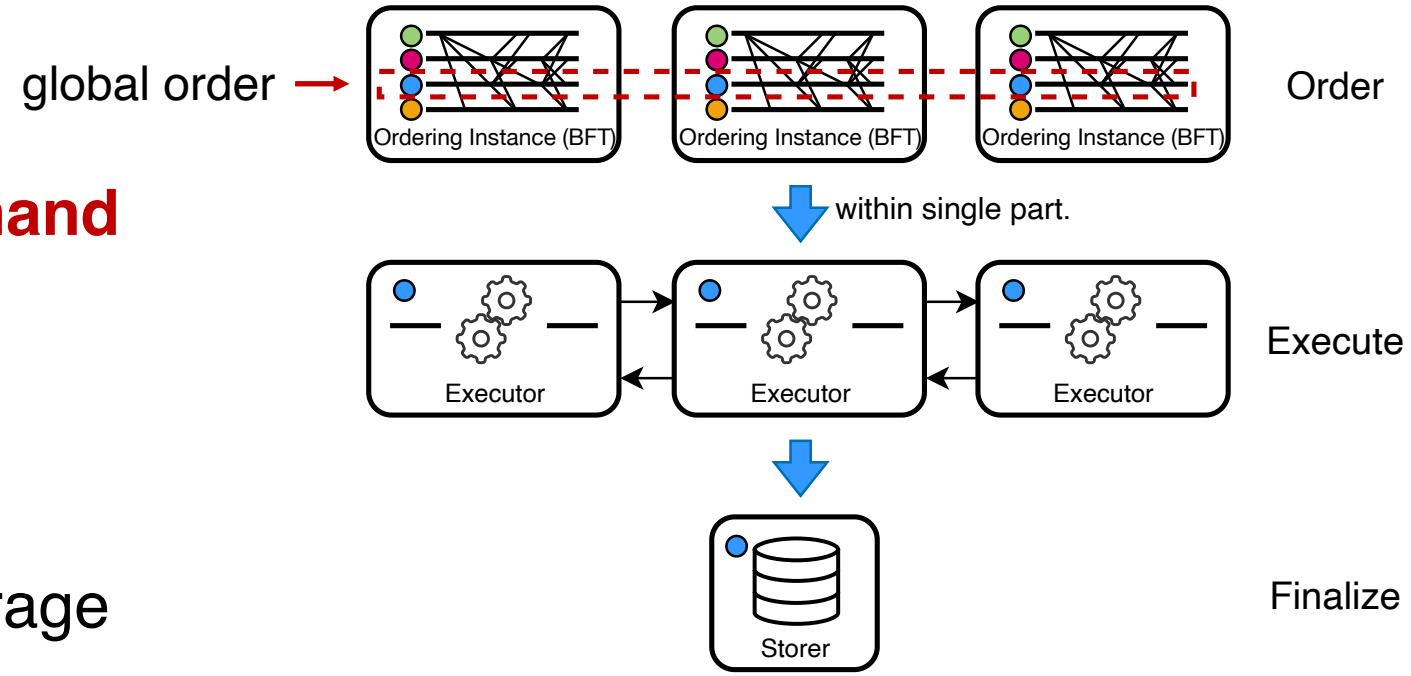
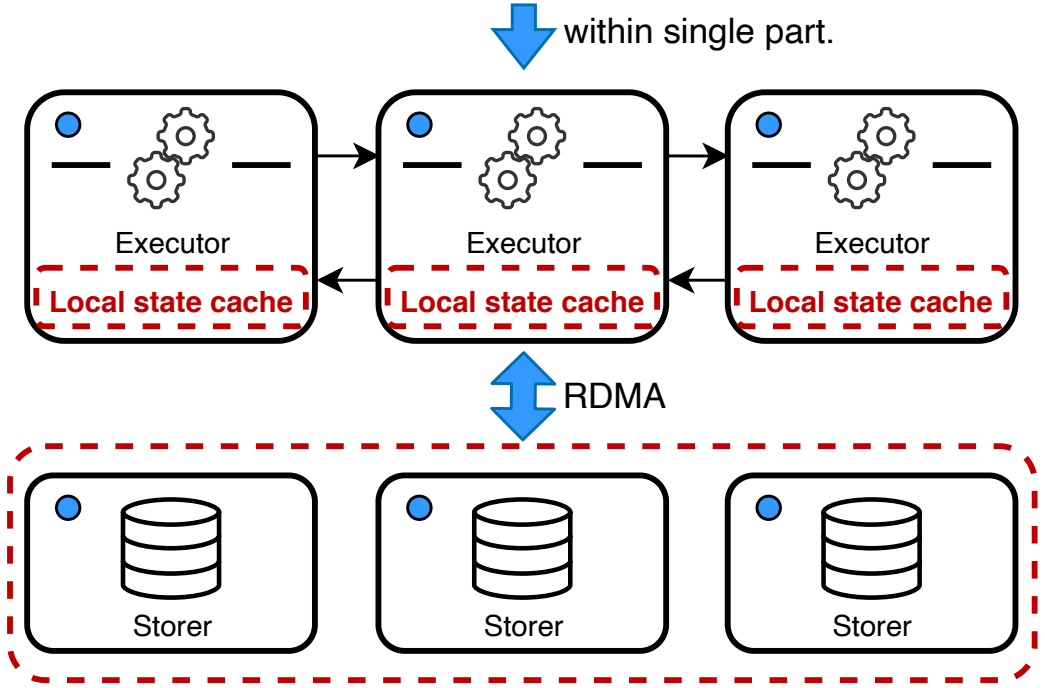


Fig.8: SOEF paradigm

Conclusion and discussion

- We introduce **SChain**, a scalable consortium blockchain that scales transaction processing by exploiting intra- and inter-block concurrency

- Future works
  - Design efficient cache maintenance to leverage data locality
  - Explore the scalable storage



THANKS !