Monoxide: Scale out Blockchains with Asynchronous Consensus Zones, NSDI, 2019

Jiaping Wang, ICT/CAS & Sinovation AI Institute; Hao Wang, Ohio State University

Youngjune Park (yjpark@kdb.snu.ac.kr)
Minbok Wi (minbok.wi@scale.snu.ac.kr)

Distributed Systems 2020 Fall
Contents

● Introduction
● Motivation
● Asynchronous Consensus Zones
● System Design
● Cross-Zone Atomicity (Eventual Atomicity)
● Defense Per-Zone Security (Chu-ko-nu Mining)
● Discussion
● Experiments and Results
● Conclusion
Introduction

- What is blockchain?
  - A type of Distributed Ledger Technology
  - Consists of a connection of blocks
  - Distributed across multiple nodes

- What is block?
  - Contain multiple transactions.
  - Has a previous block hash and its block hash

Blockchain is a technology that stores “transactions” in the form of “chain” and in units of “blocks”.

[Diagram of blockchain structure]
What is the author’s focus?

● A fixed low TPS(transaction-per-second)
  ○ Regardless of how many full nodes and miners participate in the network
● Limited scalability of a consensus system
  ○ Communication
  ○ Storage
  ○ State representation of the entire network

All nodes have same blocks(a blockchain).
Contents

- Introduction
- **Motivation**
  - Asynchronous Consensus Zones
  - System Design
  - Cross-Zone Atomicity (Eventual Atomicity)
  - Defense Per-Zone Security (Chu-ko-nu Mining)
- Discussion
- Experiments and Results
- Conclusion
Limitation 1 - A fixed low TPS

● Why is it problem?
   ○ Other Services
     ■ VisaNet payment and clearance, 4k TPS
     ■ Alipay mobile payment, 256k TPS
   ○ Blockchain Systems
     ■ Bitcoin, 7 TPS
     ■ Ethereum, 15 TPS

● Why is it “fixed”?
● Why is this process necessary?
Limitation 1 - A fixed low TPS

- Why is it problem?
- Why is it “fixed”?
  - When is a block created?

1) Transactions are created and propagated to the blockchain network
2) Enough transactions are accumulated
3) Create a hash value for each transaction
4) Collect these hash values to create a Merkle Root Hash
5) Some **Miner** find a **Nonce** that meet a difficulty condition
6) Block is created
7) Bitcoin, the average block creation time is set to about 10 minutes, and the difficulty is adjusted to maintain this.
8) Block creation time interval is determined → TPS is determined

Block Structure
Limitation 1 - A fixed low TPS

Block

Block Hash

Prev Block Hash

Merkel Root Hash

Nonce

Timestamp

Version

Difficulty

Body (Transactions)

TX0 TX1 TX2 TX3 TX4 TX5

TX6 TX7 TX8 TX9 TX N

Only can changeable

Difficulty

Conditions that Block Hash values must comply with

1
2
...
8

0xxx,xxxx,xxxx,....
00xx,xxxx,xxxx,....
...
0000,0000,xxxx,....

Difficulty example

Other Header + Nonce 0

fa8c8a....

Other Header + Nonce 1

932d16....

...

Other Header + Nonce 51234

000a83....

Nonce example
Limitation 1 - A fixed low TPS

- Why is it problem?
- Why is it “fixed”?
- **Why is this process necessary?**
  - Proof of Work
    - To create a hash of a specific block, the hash value of that block’s previous block is used. → Forgery of one block requires forgery of all subsequent blocks.
    - Without a puzzle, making a block is too easy, and this is dangerous for blockchain systems that trust long blocks.
  - Easier consensus
Limitation 2 - Limited Scalability

● Every single node is required to
  ○ Replicate ← (network I/O bound)
  ○ Store ← (Disk I/O bound)
  ○ Validate ← (CPU bound)

all transactions

For the ledger management

● Every single node is required to:
  ○ maintain states ← (Memory bound)

of all users in entire network

For the block validation

Full node: A full node is a complete copy of the blockchain and is able to verify all transactions since the beginning. This requires about 140GB of drive space (currently).

Miner: Creates blocks
Contents

● Introduction
● Motivation
● **Asynchronous Consensus Zones**
● System Design
● Cross-Zone Atomicity (Eventual Atomicity)
● Defense Per-Zone Security (Chu-ko-nu Mining)
● Discussion
● Experiments and Results
● Conclusion
Asynchronous Consensus Zones

- Consensus Zones: Multi-instantiation of independent blockchain systems
- Partitioning workloads of the entire network, distribute to zones
- Parallel block creation and transaction handling
- Linear scalable as the entire network is divided into more zones
Asynchronous Consensus Zones

- **Scalable blockchain system**
  - By partitioning and handling workloads in multiple independent and parallel instances, **Consensus Zones**
  - Blocks, transactions → **Zone specific**
  - Mining competition, chain growth, transaction confirmation → **Zone specific**

- **Linear scalability**

- **Challenges**
  - Cross-zone transaction’s high throughput
  - Individual zone’s security
Contents

- Introduction
- Motivation
- Asynchronous Consensus Zones

**System Design**
- Cross-Zone Atomicity (Eventual Atomicity)
- Defense Per-Zone Security (Chu-ko-nu Mining)
- Discussion
- Experiments and Results
- Conclusion
Partitioning and Naming

The Number of Zone: \( n=2^k \)

Sharding scale \( k \), Zone index \( s \), Block height \( h \)

User Address:
\[ \text{c64493a658f6ffca1fc8884120c7f7b5c0940946} \]
First \( k \)-bits maps to zone index

Transaction:
\[ \text{From: 5e032243d507c743b061ef021e2ec7fcc6d3ab8} \]
\[ \text{To: cbe290ef248eb14442a071fbcb58a9ce5dced28e} \]
First \( k \)-bits of From maps to zone index
(this example, \( (k=8, n=256) \), c6->Z 198, 5e->Z 94)
Isolated Intra-Zone Workload

- Each zone is a miniature version of the existing blockchain system (bitcoin, ethereum, etc.).
  - Mining competition
  - Chain (transaction blocks)
  - State
  - Unconfirmed transaction
Detailed Block Structure

- 100 bytes
- is propagated to all zones
- hundreds of kilo-bytes
- only in its zone
Design Effect

- Many zones \((n=2^k)\) -> many blockchains -> limitation 1 (A fixed low TPS)
- Each zone manages each zone’s transaction
  - Limitation 2 - Limited Scalability
  - A consensus zone
    - Replicate (**all -> its**) transactions (network I/O ↓)
    - Store (**all -> its**) transactions (Disk I/O ↓)
    - Validate (**all -> its**) transactions (CPU ↓)
  - Other zones also work for their transaction.

Scalability, Linear scaled capacity
But… it is only **intra zone workload**! How to handle **Cross-Zone** transaction?
Contents

- Introduction
- Motivation
- Asynchronous Consensus Zones
- System Design
- Cross-Zone Atomicity (Eventual Atomicity)
- Defense Per-Zone Security (Chu-ko-nu Mining)
- Discussion
- Experiments and Results
- Conclusion
Cross-Zone Transaction

- Transfer = Withdraw + Deposit

Figure 2: Data structure of the chaining-blocks and transaction-blocks. Outbound relay transactions are derived from confirmed transactions and forwarded with verification data ($h_q$ and etc.)
Transaction Verification

- Unconfirmed transfer TX, \(<\rho, a, \Phi, b>\)
  - Withdraw operation \(\rho\)
  - Deposit operation \(\Phi\)
  - \(a \rightarrow b\)
  - Its originate chaining block \(\Theta_a\)
- The Merkle tree root will be recalculated
- Is it matched with \(\Theta_a\)'s Merkle tree root?
Block Verification

- Block’s all TX will be checked (is contained the block)
- With User states
- Inbound TX
  - Merkle tree path
- Outbound TX
  - Merkle tree root of the list of all outbound relay transactions
Eventual Atomicity

- Payment Transaction = Initiate TX + Relay TX
  - Transfer x tokens from user A to user B in different zone
  - Will be executed eventually
Fork Resolution

- Monoxide builds on the Proof-of-Work scheme
- Monoxide uses GHOST protocol to resolve its forks for better reliability
  - GHOST protocol (Greedy Heaviest Object subTree)

Fork Resolution (cont’d)

- In each consensus zone, fork resolution is performed independently.
- What is the effect of the fork resolution in the presence of relay transactions?
- With the eventual atomicity, the consequence of fork resolution in one zone may affect the validity of relay transactions forwarded and confirmed in another zone (cross-zone effects)
  - Two cases: Pre-confirmed and Post-confirmed
- Pre-confirmed case:
  - Relay transactions will stay in the unconfirmed transaction set regardless of the validity of their originate block
  - And wait for originate blocks become available state.
Fork Resolution (cont’d)

- **Post-Confirmed**
  - There exists a relay transaction that originates from orphan or unsolved block.
  - Should rebuild the block with related to such relay transactions.
  - Optimization: **State Checkpoints**

---

**Figure 3**: Relay transaction invalidated due to orphaned initiative block after fork resolution.
Fork Resolution (cont’d)

- Even worse case:
  - Implicate subsequent transactions in invalidating
  - A miner will validate candidate transactions against a special state by **delaying execution** of inbound relay transaction for $\lambda$ blocks.
  - Maintains two states $\{S, S_\lambda\}$

Figure 3: Relay transaction invalidated due to orphaned initiative block after fork resolution.
Contents

- Introduction
- Motivation
- Asynchronous Consensus Zones
- System Design
- Cross-Zone Atomicity (Eventual Atomicity)
- **Defense Per-Zone Security (Chu-ko-nu Mining)**
- Discussion
- Experiments and Results
- Conclusion
Diluted Mining Power due to Multiple Zones

- Security issue:
  - Single-zone focused attack
  - Propose: *Chu-ko-nu mining*
Chu-ko-nu Mining

- A single PoW solution to create multiple blocks at different zones simultaneously, but no more than one block per zone.
- And enforce the mining power to be evenly distributed across zones.
  - The difficulty of attacking a single zone is as difficult as attacking the entire zone.

Source: https://www.pinterest.co.kr/pin/426786502166824878/
Chu-ko-nu Mining (cont’d)

- One-block per-zone leads to a different data structure: Batch-Chaining-Block
- A miner will perform the transaction validation for all involved zones and need to collect all chaining-headers.
  - A: chaining-header
  - B: Merkle tree path
  - C: Batch configuration
- Finding nonce:

\[
\text{hash}(\langle A_i, \eta_i \rangle) < \tau, \quad \text{hash}(\langle h_0, C, \eta_b \rangle) < \tau,
\]
Chu-ko-nu Mining (cont’d)

Existing blockchain system

Total Hashrate: \( \text{t hash/sec} \)
Total Effective Mining Power: \( \text{t hash/sec} \)

*effective mining power: the speed of chain grows or block creation
Chu-ko-nu Mining (cont’d)

Monoxide

Consensus Zone 1

Consensus Zone 2

... ...

Consensus Zone n

Total Hashrate: \( t \text{ hash/sec} \)
Total Effective Mining Power: \( t \text{ hash/sec} \)

*effective mining power: the speed of chain grows or block creation

for(nonce=0; ; nonce++) {
  ...
}
Chu-ko-nu Mining (cont’d)

Consensus Zone 1

Consensus Zone 2

... 

Consensus Zone n

Block

Block

Block

0000000000000000000ba3295985f23 7068b89b3738200d77ba5bbc79280a76

Total Hashrate: \( t \) hash/sec
Total Effective Mining Power: \( t \times n \) hash/sec

*effective mining power: the speed of chain grows or block creation
Independent Validation in Zones

- Not only improvements in mining but also needs to efficiently handle the validation process
- Handle validation independently in zones.
  - Per-zone batch-chaining-blocks will be validated and accepted independently in each zones
  - Also allows efficient handling of mixed PoW targets of zones in one batch
- Proof for security of Chu-ko-nu mining?
  - Key: an attacker must have more than half of the effective hash rate.
    - which converges to 50% of the total physical hash rate in the network as the number of zones and mining power increases (considered impossible)
  - Please refer to the paper for more details
Scalable Mining System

- A mining system is desired to monitor multiple zones.
- (Central) **Mining Coordinator** handles the update of block header and control of mining units.
Contents

- Introduction
- Motivation
- Asynchronous Consensus Zones
- System Design
- Cross-Zone Atomicity (Eventual Atomicity)
- Defense Per-Zone Security (Chu-ko-nu Mining)

- **Discussion**

- Experiments and Results
- Conclusion
Discussions

● Monoxide should efficiently handle the below issues
  ○ Single address hotspot
  ○ Incentives and fees (how to design the reward model?)
  ○ Generalization beyond payment (more complex transactions)
Discussions (cont’d)

● Single Address Hotspot
  ○ A single address can be involved in a great number of transactions.
    ■ e.g., a deposit address of a large cryptocurrency exchange
  ○ Since a Monoxide’s finest unit of partitioning is an address, system cannot further partition such workload into multiple zones.
  ○ But it can be solved through co-design of applications at the upper layer
    ■ e.g., User allocate multiple deposit addresses in different zones for load balancing
Discussions (cont’d)

- Incentives and Fees
  - Follows **Bitcoin’s incentive model** by rewarding the miners with phase down coinbase in every zone
  - And the transaction fee is determined by someone who issue the transaction.
  - Introduce the **Fee splitting** for cross-zone transaction to equally prioritized the relay transaction
    - Incentivize both miners working on initial step and relayed step of transaction handling
  - Block creation with cross-zone transaction is costly than intra-zone’s.
    - It can be handled by differing the amount of transaction fees. (and it will follow a market economy.)
  - Encourage Chu-ko-nu mining by providing equal coinbase reward
Discussions (cont’d)

- Generalization beyond Payment
  - Extend the model from the withdraw-deposit paradigm to **complex transaction logic**
  - Main challenge: How to correctly handle the generalized **relay transactions**?
  - See appendix for more details
Contents

- Introduction
- Motivation
- Asynchronous Consensus Zones
- System Design
- Cross-Zone Atomicity (Eventual Atomicity)
- Defense Per-Zone Security (Chu-ko-nu Mining)
- Discussion

**Experiments and Results**
- Conclusion
Experiment Setup

● Playback historical ERC20 payments log in Ethereum from the beginning up to the block height: 5,867,279
  ○ 16.5 million unique addresses
  ○ 75.8 million transactions

● Deploy Monoxide on a distributed environment that includes 1,200 virtual machines (8 cores and 32 GB memory each)
  ○ Machines are uniformly distributed in 15 availability zones
  ○ 15 second block creation interval (15.6 TPS)
Experimental Results

- Measured TPS scales out as the number of zones increase.
- Transactions handled in each zone are balanced, with different sharding scale.

![Figure 6: Linear scaling out with multiple zones.](image1)

![Figure 7: Transaction distribution across zones.](image2)
Experimental Results (cont’d)

- The percentage of cross-zone transaction grows when increasing the number of zones.
  - Caused by relay transactions
  - However it scales out well

Figure 8: Percentage of cross-zone transactions, which approaches to 100%. Almost every original transaction produced a relay transaction.
Experimental Results (cont’d)

- Default transaction proportion still the same as the number of zones increase (red bar)
- Increase in relay transactions double the size of transaction blocks (blue bar)
- Duplication of chaining block to all zones lead to increase in total size on storage, but less significant (black bar)

Figure 9: Sizes of the blockchain data in the entire network.
Experimental Results (cont’d)

- The number of connections per node affects the propagation speed.

- The confirmation latency is low enough compared to time for securing the original transaction (when $n = 6, 90$ sec)

Figure 10: Transaction (<1KB) propagation speed with different average number of connections from each node to other peers. Fastest propagation shown here is 128-connected, slowest one is 8-connected. Ones in between are 64/32/16-connected.

Figure 11: Average first confirming time of transactions.
Experimental Results (cont’d)

- Enlarging block size or lessening block creation interval yields almost linear TPS. (Blue)
- A larger block size or a smaller block creation interval leads to a higher orphan rate. (Red)
- Reasonable choice: 32 KB and 15 seconds for each
Contents

● Introduction
● Motivation
● Asynchronous Consensus Zones
● System Design
● Cross-Zone Atomicity (Eventual Atomicity)
● Defense Per-Zone Security (Chu-ko-nu Mining)
● Discussion
● Experiments and Results
● Conclusion
Conclusion

● Monoxide achieves scalability, security and decentralization
● Monoxide partitions the workload of all key components (sharding)
  ○ Communication, transaction processing, state representation, history archiving, network bandwidth, computing power, memory size, disk I/O

● Key contributions:
  ○ **Eventual atomicity**: Efficient cross-zone transaction handling
  ○ **Chu-ko-nu mining**: Security guarantee for individual zones

● Experiment shows:
  ○ **1,000x throughput** and **2,000x capacity** over Bitcoin and Ethereum on a testbed (Monoxide: 10K TPS, Bitcoin: 7 TPS, Ethereum: 15 TPS)
Thank you