A
Review
of

LedgerGuard: Improving Blockchain Ledger Dependability

Sajan Maharjan

20182095

CSED601 Dependable Computing

Purpose

- Partial Requirement of CSED601 Course
- Term Paper
  - Study on dependability aspects of blockchain technology
  - One of the components (case study)
  - Additional case studies to be presented later

Contents

- Blockchain & Dependability
- LedgerGuard
- Hyperledger Fabric
- LedgerGuard Mechanism- Corruption detection & recovery
- Evaluation
- Conclusion

Blockchain & Dependability

- Blockchain
  - Portmanteau of *block* (storing information) and *chain* (links)
  - Distributed, decentralized, immutable, append-only ledger of records
  - Underlying technology of Bitcoin, started in 2008
  - Maintaining ledger integrity and security is one of the crucial design aspects of any blockchain platform
- Dependability
  - Availability
  - Reliability
  - Safety
  - Integrity
  - Maintainability

But Isn’t Blockchain already secure?

- Yes! It is relatively secure
  - Distributed, decentralized copy of records (Node Failure Tolerant)
  - Makes use of cryptographic hashes and consenus protocols (algorithms) to validate and verify blocks
- No! There have been compromises
  - Bitcoin Hacks (Mt. Gox)
  - Sybil Attacks, DDoS
  - Application Level Vulnerabilities

But Isn’t Blockchain already secure?

- Also
  - Blocks running on a blockchain node over an extended period of time can be corrupted due to software or hardware failures. (Magnetic disk or SSD failure)
  - Antivirus software or firewalls may delete or corrupt files relating to blockchain.
  - Low probabilistic success at tampering data

    “In private Blockchains such as Hyperledger Fabric or R3 Corda, it is critical to maintain the nodes hosting peers highly secure. However, when a peer is hosted in a less secure environment, an external attacker or malicious user can hack into the peer node and modify the content of the ledger files.” *- Qi Zhang et al*

LedgerGuard

- A tool to maintain ledger integrity by detecting corrupted blocks and recovering these blocks by synchronizing with the rest of the network
- Based on Hyperledger Fabric (permissioned, open source blockchain platform)
- Integrity of records in blockchain is essential before executing any analytical or auditing applications on the records
- Not used to check the integrity of incoming node data, but used to check the integrity of existing node data

  "However, the peer lacks the capability of detecting and recovering the corrupted blocks existing in the ledger during its runtime." - Qi Zhang et al

LedgerGuard

- A runtime self correction mechanism for ledger
- Enforces integrity using two techniques-
  - First, it validates the contents of each block and the hash links between each blocks
  - Second, if corrupted block is identified, LedgerGuard recovers the block and corrects the affected part of ledger without the need for rebuilding the whole ledger.
- Can be used as a tool (by an operator) or started as a service

Hyperledger Fabric

- Permissioned blockchain platform suitable for enterprises
- Byzantine Fault Tolerance Consensus Algorithm
- Orderer signs the transaction using certificate issued by CA

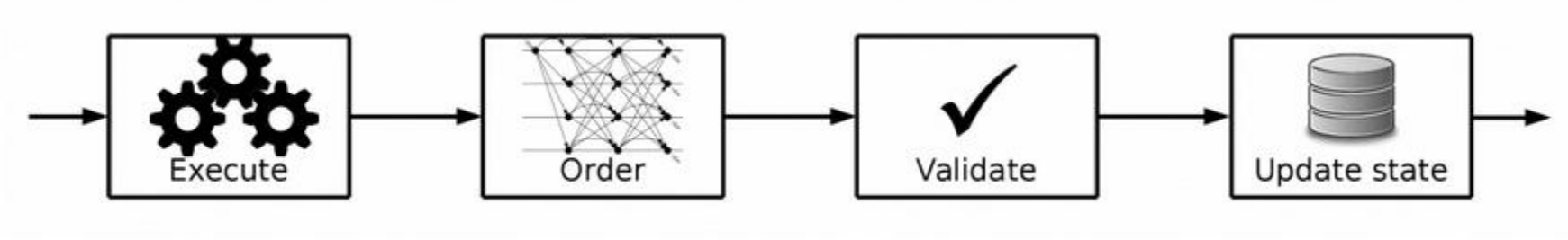

Fig. Transaction flow in Hyperledger Fabric

Hyperledger Fabric (contd…)

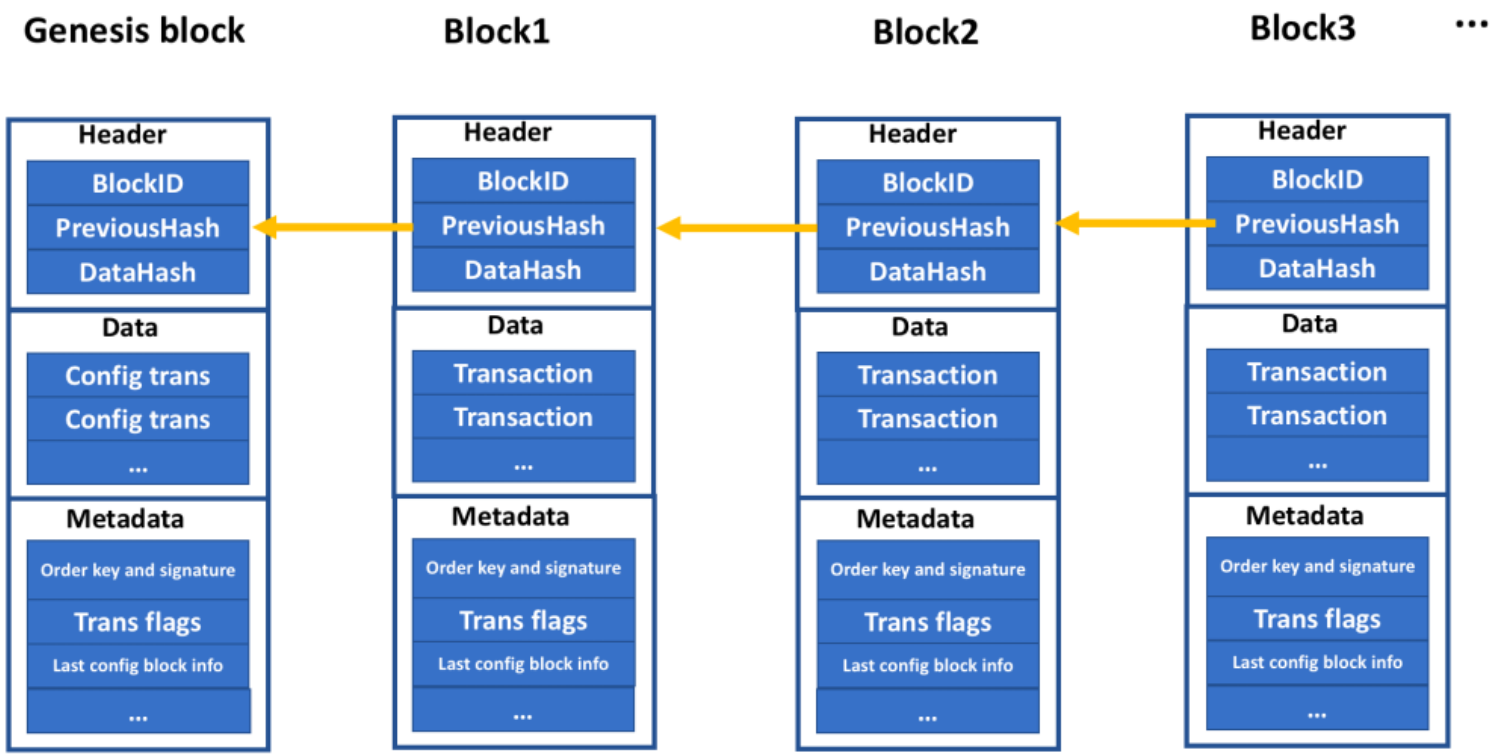

Fig. Block structure in hyperledger fabric

Ledger Corruption Detection

- Like integrity maintained in blockchain, LedgerGuard validates integrity from two aspects-
  - Each single block in the ledger is not corrupted
  - The hash pointers between the blocks are valid
- To validate a single block, LedgerGuard uses the certificate of the ordering service to validate the correctness of each block header.
- To validate the correctness of the hash pointer, LedgerGuard calculates the hash value of the current block and compares it with the value of “PreviousHash” of the next block

Corrupted Ledger Recovery

- A corrupted block is recovered by sending a request to its peers asking for the block with the same ID.
- Sometimes multiple blocks need to be retrieved to fix the ledger even though only one block is corrupted.
  - In hyperledger fabric, a ledger consists of one or multiple fixed size files, and each file contains continuous series of blocks. A corrupted block can be larger or smaller than the size of the correct block
  - Replacing block A with the correct block A’, but of different size will overwrite part of the successive blocks or leave a gap in between

Corrupted Ledger Recovery

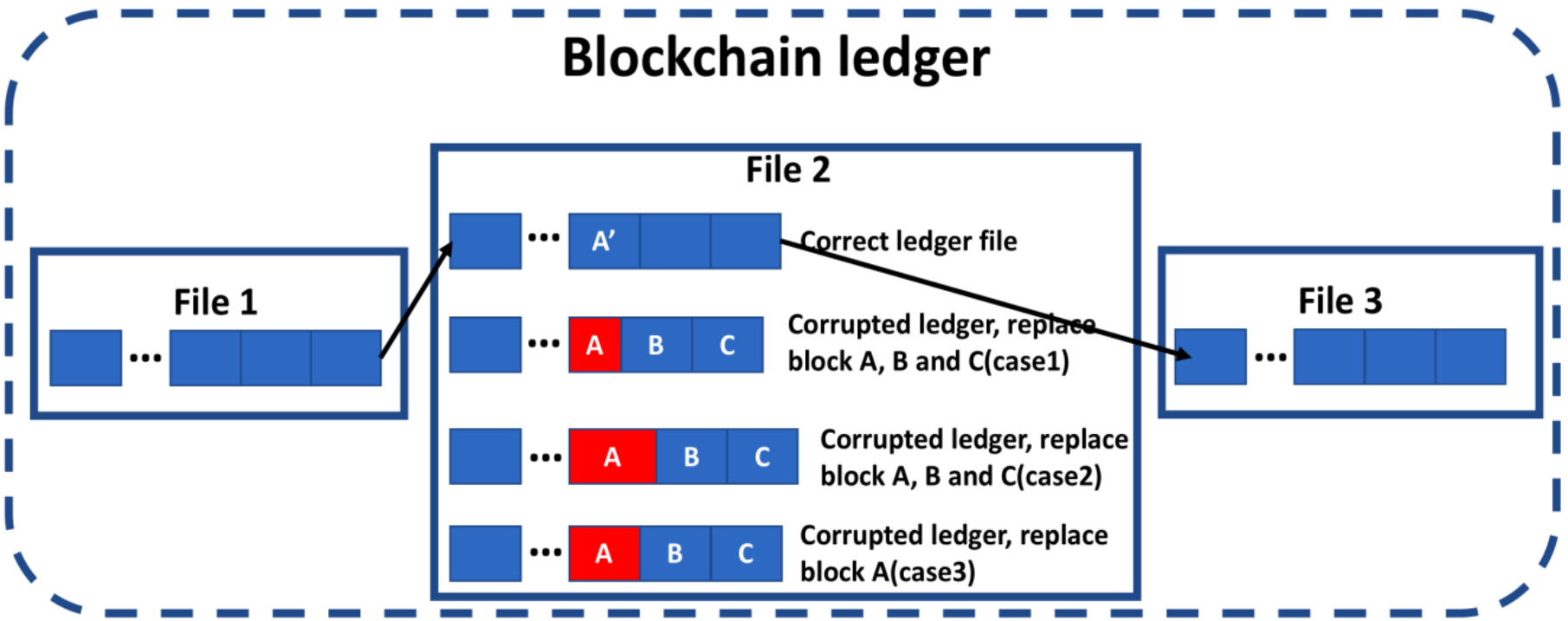

Fig. Blockchain ledger stored in files

Corrupted Ledger Recovery

- LedgerGuard first checks the size of the corrupted block against the correct block in the peers.
- If the size of the corrupted block when replaced with the correct block doesn’t overlap or gap the successive block, only the corrupted block is replaced.
- Else, all the blocks in that file is replaced.

Execution & Evaluation

- Tested on 4-core Vmware virtual machine, with Intel® Xeon® CPU E5-2698 2.2GHz with 4GB of RAM
- Ledgers were generated using a tool that simulates the block generation on a real Hyperledger Fabric blockchain.
- 4 peers hyperledger fabric blockchain network

Execution & Evaluation

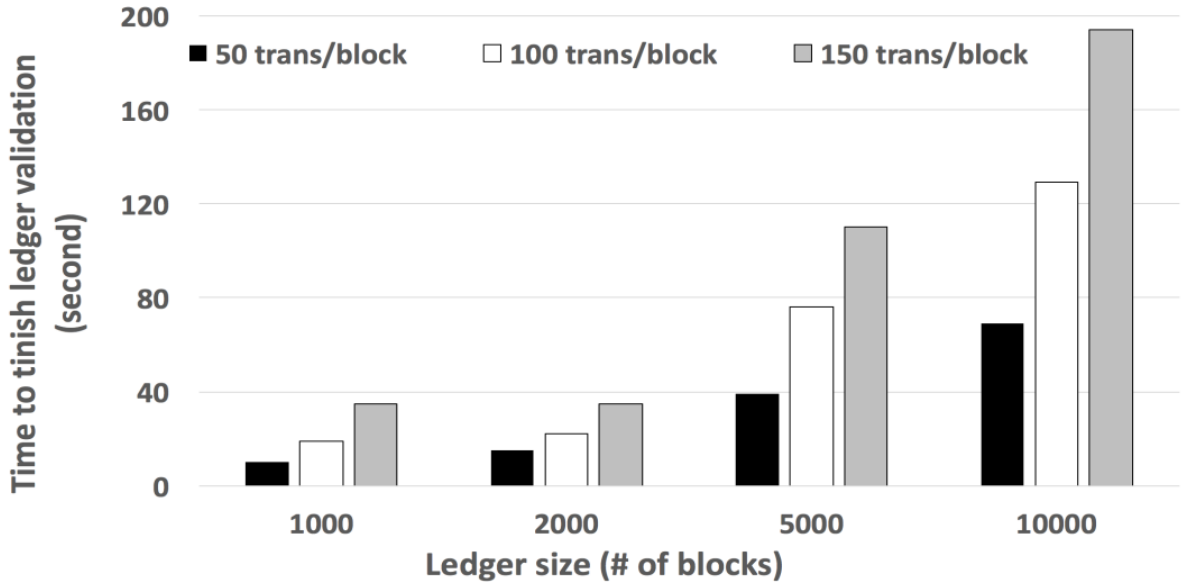

Fig. LedgerGuard ledger validation time

LedgerGuard sequentially scans through each block in the ledger- O(n) complexity

Execution & Evaluation

- Uses 60MB memory
- CPU Utilization starts with 110% and then drops to 20%- the initial spike is due to the need to do some initialization work such as opening the ledger, reading configuration of the blockchain network, etc
- Given a block size of 150 transactions per block, LedgerGuard can fetch blocks from another peer, validate and commit at the speed of 8.5 blocks per second

Conclusion

- An existing blockchain maybe corrupted due to several reasons- hardware or software failure, blocked or deleted by antivirus software or malicious node inside private networks
- LedgerGuard- a runtime mechanism for maintaining the ledger integrity for HyperLedger fabric
- Detects and corrects block errors by synchronizing with the rest of the network.
- A prototype was implemented and its effectiveness was evaluated

References

- Behind the Architecture of Hyperledger Fabric, https://www.ibm.com/blogs/research/2018/02/architecture-hyperledger-fabric/
- Zhang Q. et al, LedgerGuard: Improving Blockchain Ledger Dependability, (2018), https://arxiv.org/pdf/1805.01081.pdf