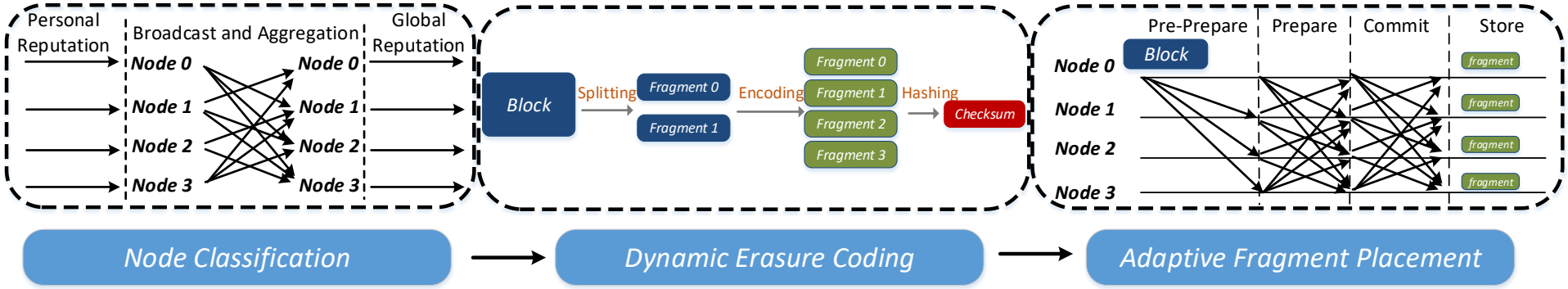
Dynamic-EC: An Efficient Dynamic Erasure Coding Method for Permissioned Blockchain Systems

Mizhipeng ZHANG, Chentao WU, Jie LI, Minyi GUO

Frontiers of Computer Science, DOI: 10.1007/s11704-023-3209-3

Problems & Ideas

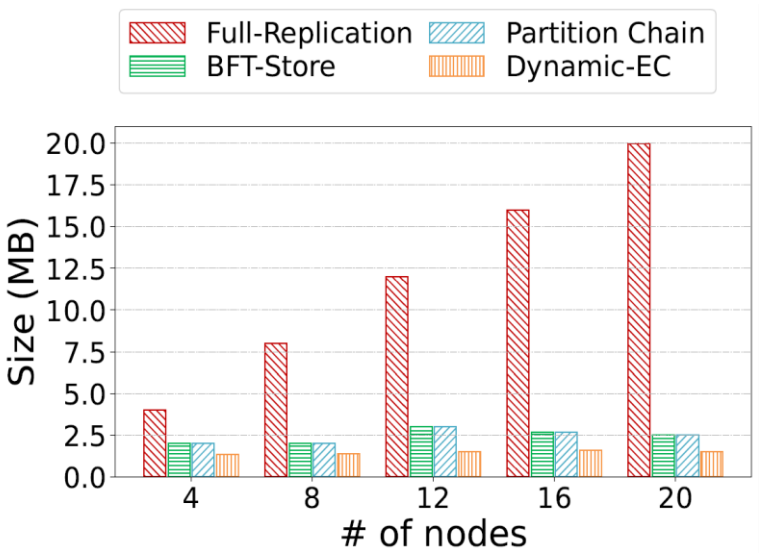
- Problems of existing storage methods in blockchain system:
  - the full replication method has extraordinarily high storage cost.
  - The static erasure coding method utilizes a fixed pattern code type to store blocks. When the actual number of malicious nodes is relatively low, excessive checks can lead to storage wastage.
- Ideas : We employ dynamic erasure coding to store blocks, predict the number of risk nodes, and dynamically adjust the number of parities, while ensuring block security and reducing total storage overhead.

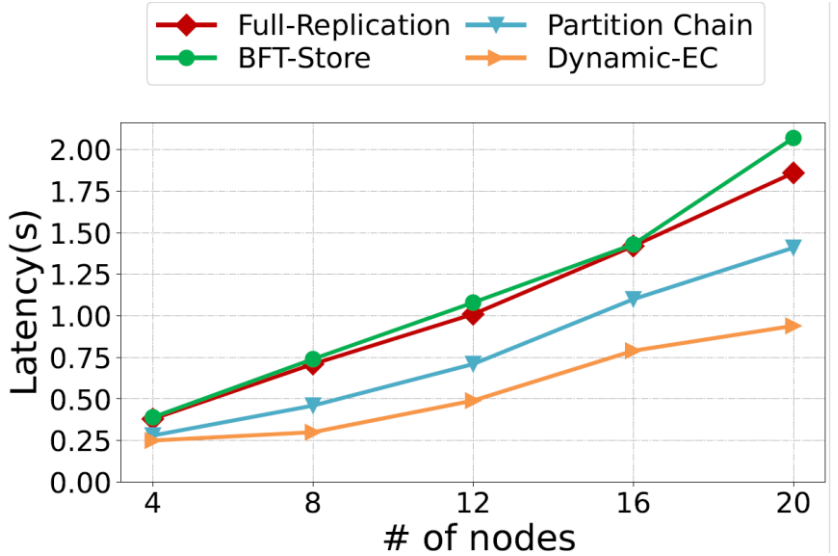


Node Classification is used to evaluate the global reputation value of each node. According to the reputation value, the nodes are classified into three categories: honest nodes, risk nodes and malicious nodes. ***Dynamic Erasure Coding*** is responsible for encoding the blocks into fragments and calculating the corresponding checksums in the leader node. The encoding schema is adaptively adjusted according to the current risk level of the system which is determined by the current number of risk nodes. ***Adaptive Fragment*** Placement distributes the encoded fragments from the leader node to the follower nodes.

Main Contributions

- Contributions:
  - We propose a method of node reputation evaluation, which enables the leader node to perceive the network status while incentivizing the nodes to correctly comply with the protocol;
  - We propose a novel storage redundancy method called Dynamic-EC for permissioned blockchain systems, which provides low storage costs and high I/O performances;
  - We conduct several experiments and demonstrate Dynamic-EC reduces the storage overhead over the state-of-the-art methods.





Dynamic-EC decreases the storage overhead by up to 42% and shorten the latency for writing a block by up to 25% . Left: the block storage overhead; Right: the average writing latency.