Towards Scaling Blockchain Systems via Sharding

区块链扩展性

- Distributed consensus protocols

- cryptocurrency

分片Sharding

- 为什么分片?

- 通信开销减少，吞吐量提高。
- 更多的碎片减轻整个系统的压力。

- 改善拜占庭共识算法表现
- 设计有效的碎片形成协议
- 设计分布式事务协议

现有分片

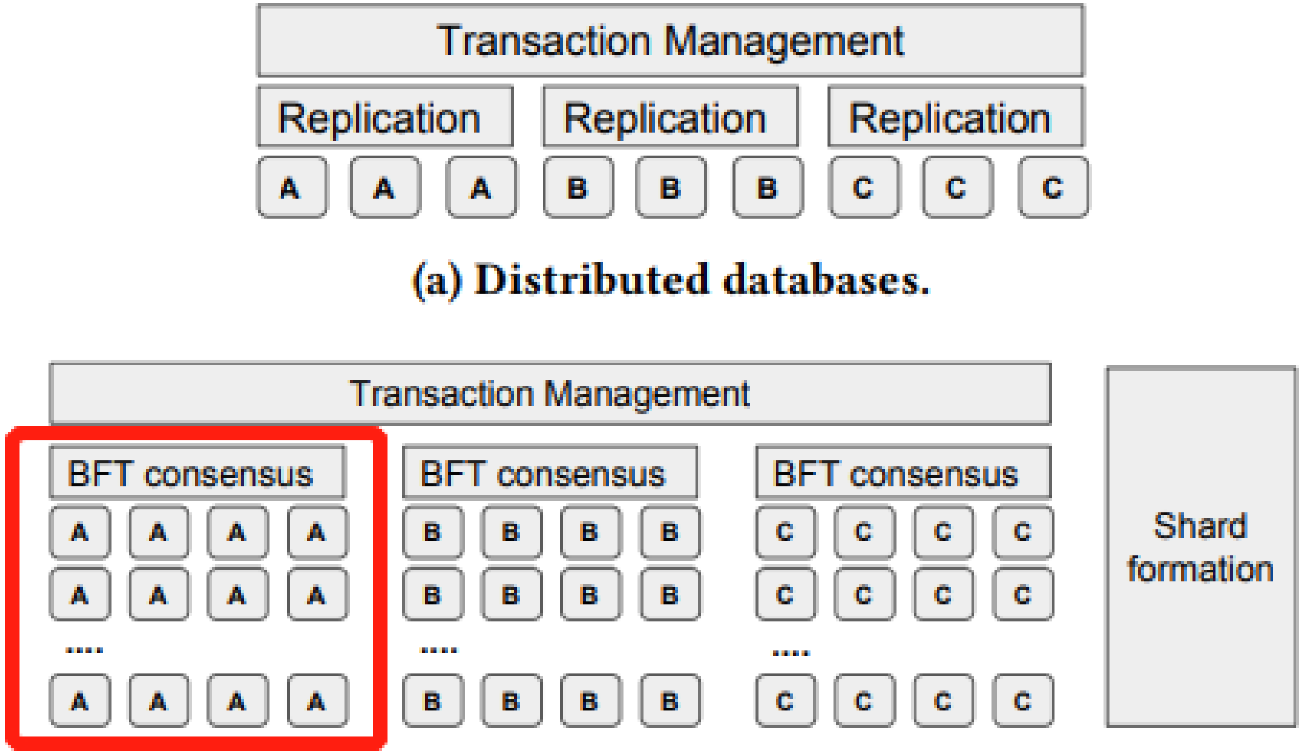
- Elastico
- OmniLedger
- RapidChain

本文分片

- 表现

- 区块链系统
  - 支持大规模网络（Bitcoin & Ethereum规模）
  - 达到高事务吞吐量（如中心化系统visa，2k-4k事务/s）
  - 金融健康

Distributed databases vs. Sharded blockchains



(a) Distributed databases.

(b) Sharded blockchains.

Figure 1: Sharding protocols in traditional databases vs. blockchains.

应用数据库分片到区块链的目标与挑战

- Challenges
  - high-performance consensus protocols (x) → BFT
  - Shard formation → TEE
  - Safety(atomicity & isolation), liveness(transaction abort or commit) → 2PC & 2PL

- Goals
  - 大规模网络
  - 高吞吐量
  - 不止是加密货币

- Challenges
  - high-performance consensus protocols(x) → BFT protocols → TEE
  - Shard formation → TEE
  - Safety(atomicity & isolation), liveness(transaction will abort or commit) → 2PC & 2PL

区块链分片系统架构下的三个挑战

- 共识算法的设计
- 节点分配的设计
- 支持分布式事务

挑战1： 共识算法的设计

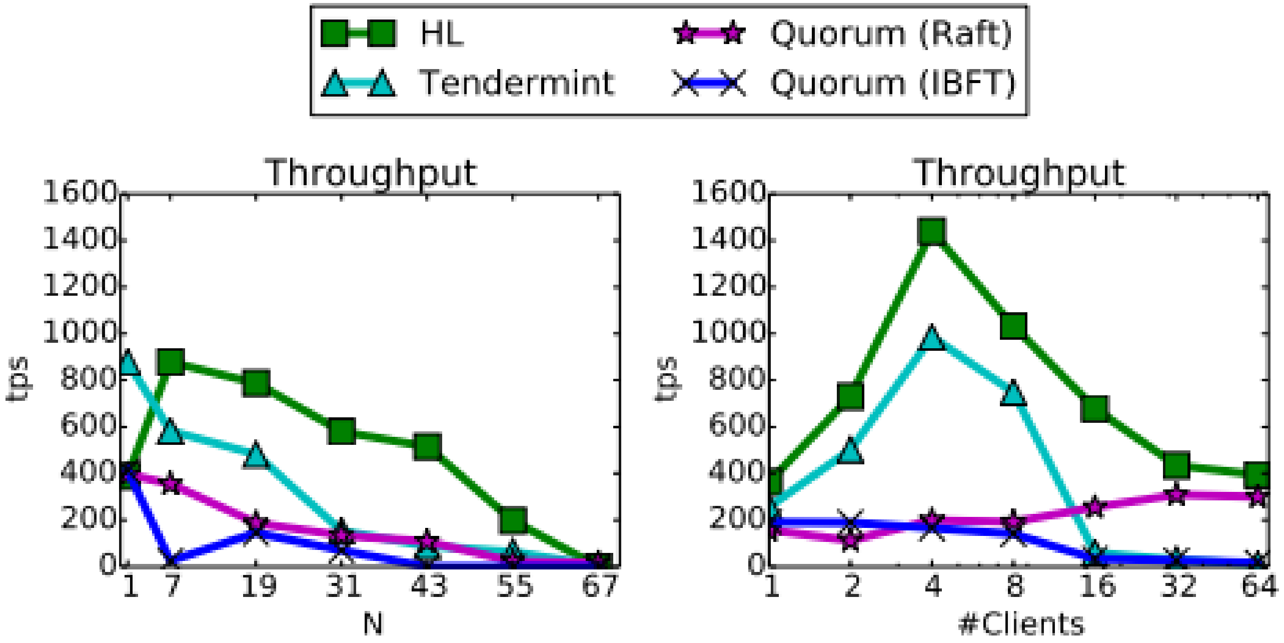


Figure 2: Comparison of BFT protocols with varying number of nodes and clients.

为什么使用PBFT?

- PBFT: pipelined execution
- IBFT & Tendermint: lockstep

PBFT + TEE（可信执行环境）

- PBFT
  - 恶意节点个数不大于f
  - $N = 3f + 1$, 即$f < N / 3$

- PBFT + TEE
  - $N = 2f + 1$, 即$f < N / 2$

PBFT + TEE

- 避免大的TCB
  - 日志（共识信息：pre-prepare, prepare, commit）
  - 消息摘要
  - TEE密钥加密

- AHL(Attested HyperLedger)

PBFT + TEE

- AHL → AHL+
  - Original message queue: consensus & request
  - Remove the request broadcast
- AHL → AHLR
  - Collects & aggregates
  - node ↔ leader
  - Communication overhead: O(N)

PBFT + TEE

- 安全性分析
  - AHL f= (N-1) / 2
    - log operations
    - safety & liveness
  - AHL+
  - AHLR

- PoET → PoET+
- waitTime
- SGX(sgx_read_rand): l-bit value q

PoET+ vs AHL+

- PoET+
  - Byzantine threshold
  - network latency
- AHL+

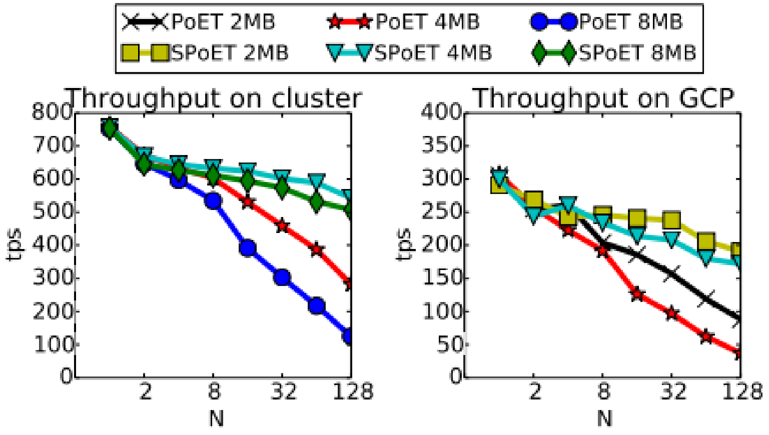


Figure 21: PoET and PoET+ performance.

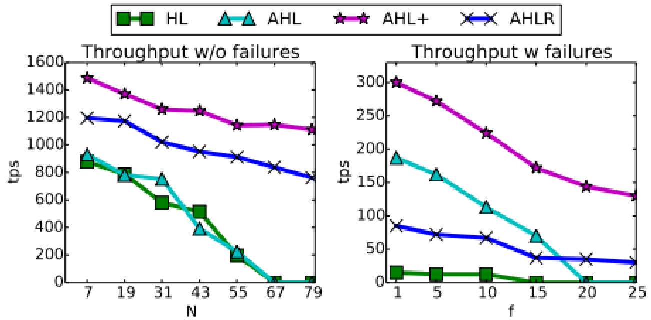


Figure 8: AHL+ performance on local cluster.

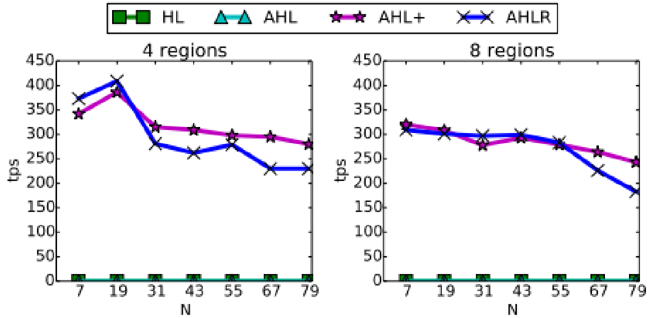


Figure 9: AHL+ performance on GCP (4 and 8 regions).

挑战2： 节点分配的设计

- 节点无偏随机分配到委员会
- 委员会大小：balance performance & security
- 自适应攻击者

节点分配

- Intel SGX（一种TEE）
  - *sgx_read_rand*
  - *sgx_get_trusted_time*

- 思路：
  - rnd → random seed → [1:N]

- 每个节点相同rnd的获取
  - 分时期工作
  - **R**ANDOMNESS**B**EACON ← an epoch number **e**

- STEP 1:生成两个随机数q和rnd

- STEP 2:如果q=0，那么该节点就生成一个包含<e,rnd>的签名证书，广播该证书到其它所有节点

- STEP 3:所有节点等待Δ时间之后，锁定所获收集到的最小的rnd

- STEP 4:使用最小rnd来作为当前时期的委员会分配的随机种子seed。

- 重复执行概率：$P_{\text{repeat}} = (1 - 2^{-l})^N$ (l: the bit length of q)

委员会大小

- probability of a faulty committee(超过f个拜占庭节点)

$$Pr[X \geq f] = \sum_{x=f}^{n} \frac{\binom{F}{x}\binom{N-F}{n-x}}{\binom{N}{n}}$$

- PBFT: 600+ nodes
- AHL+: 80

自适应攻击

- 切片的重新配置
  - B nodes to new committees
  - $\frac{n(k-1)}{k\cdot \mathrm{B}}$ 个过渡委员会
  - $Pr(\mathtt{faulty}) \leq \sum_{i=1}^{\frac{n(k-1)}{k\cdot \mathrm{B}}} \sum_{x=f}^{n} \frac{\binom{F}{x}\binom{N-F}{n-x}}{\binom{N}{n}}$

挑战3： 支持分布式事务

- safety & liveness
- multiple shards
- concurrency

- 2PC & 2PL

2PC & 2PL

- 2PC: two-phase commit
- 2PL: two-phase locking

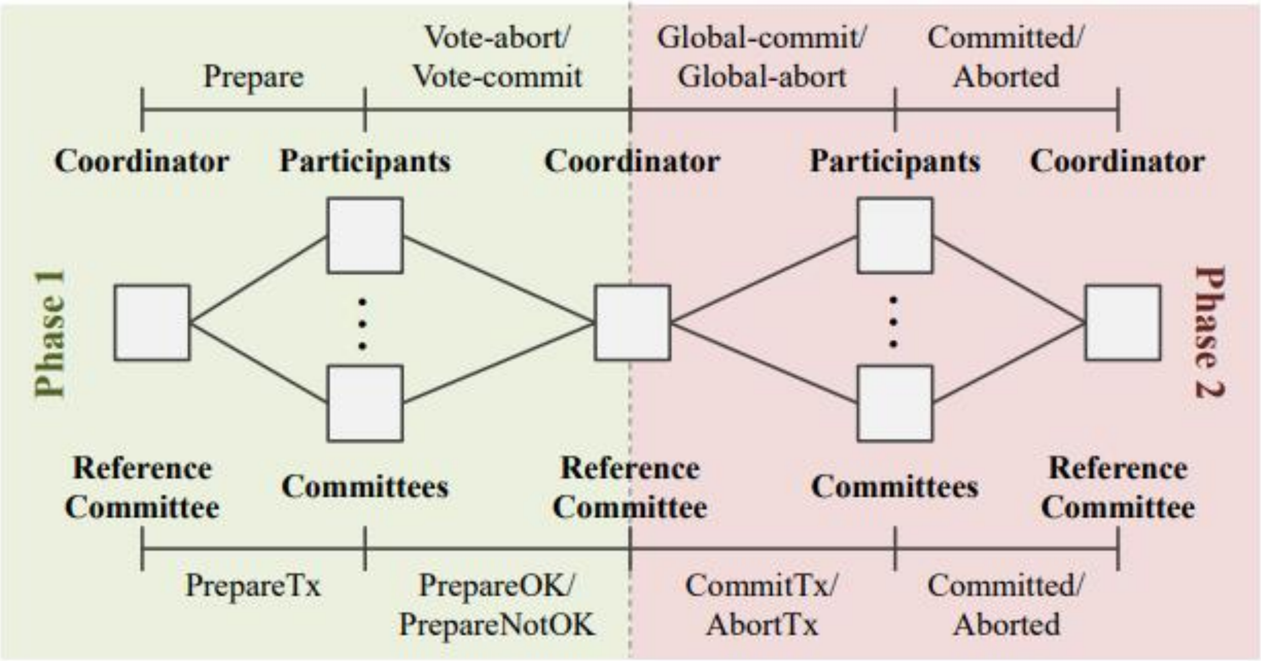


Figure 7: Correspondence between our distributed transaction management protocol (i.e., bottom half) and the original 2PC protocol (i.e., top half).

委员会间通信

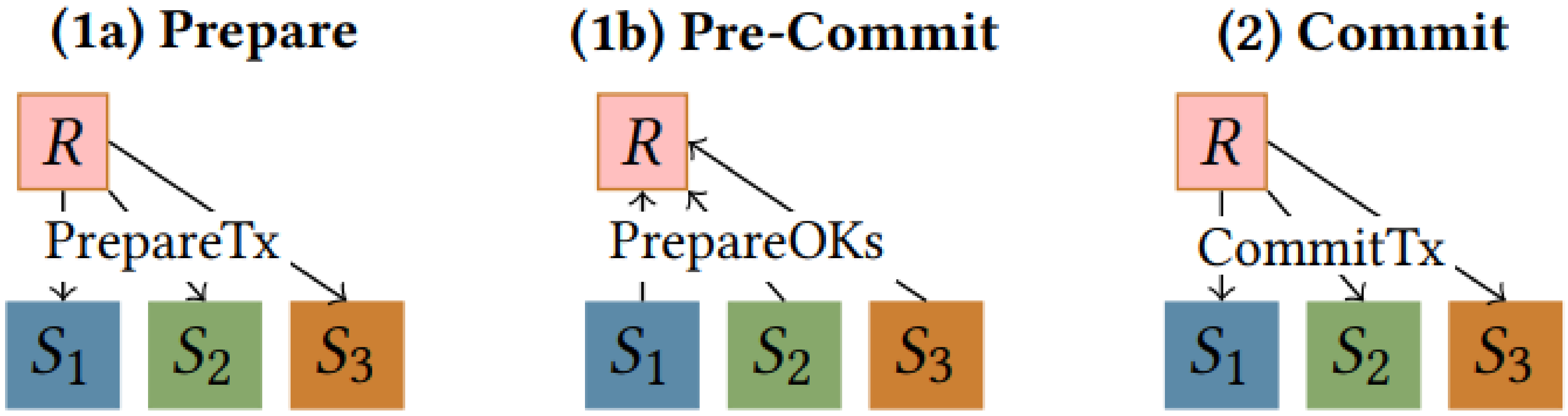
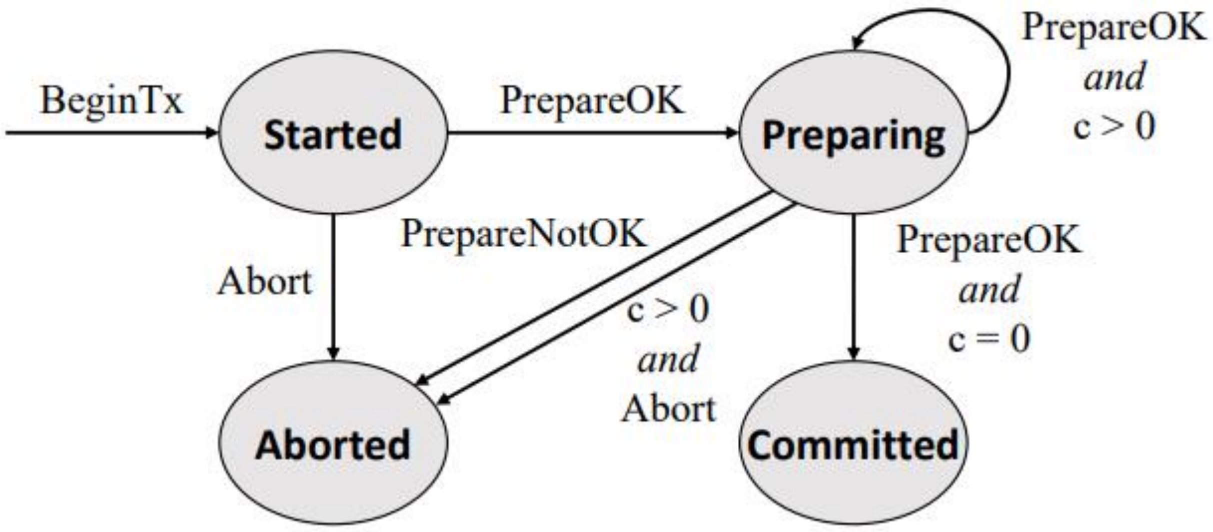


Figure 5: Our coordination protocol.



安全性 & 活性

- assume R & tx-committees ensure safety
- Byzantine nodes < 50% the size of R

实现

- Hyperledger Fabric
  - sendPayment → preparePayment, commitPayment, abortPayment
  - “L_”acc. 标识区块链状态的bool值
  - preparePayment: 检查元组⟨L_acc, true⟩是否存在
  - commitPayment

- 2PL
  - batching

- 扩展
  - a library containing functions for sharded applications

  - add programming language features
  - introduce a client library

表现评估

- scalable consensus protocol
- shard formation protocol
- the scalability of sharding approach

- KVStore & Smallbank

本地

- Intel Xeon E5-1650 3.5GHz CPUs
- 32GB RAM
- 2TB hard drive

Google Cloud Platform

- Client
  - 16 vCPUs
  - 32GB RAM
- Node
  - 2 vCPUs
  - 12GB RAM

- 1400
- 8 regions

共识算法对比

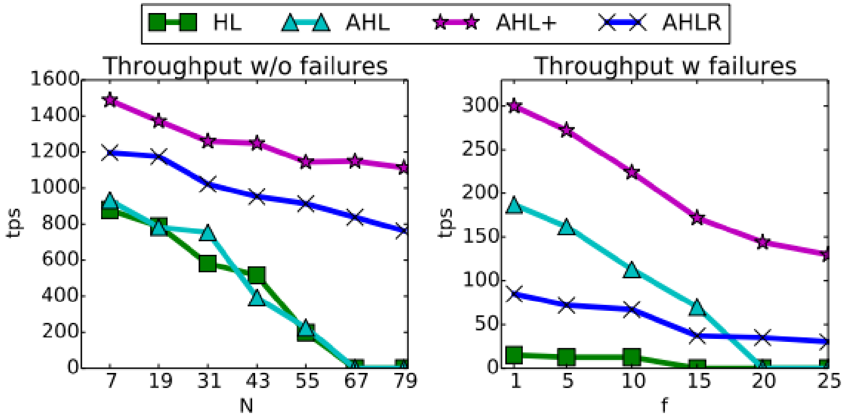


Figure 8: AHL+ performance on local cluster.

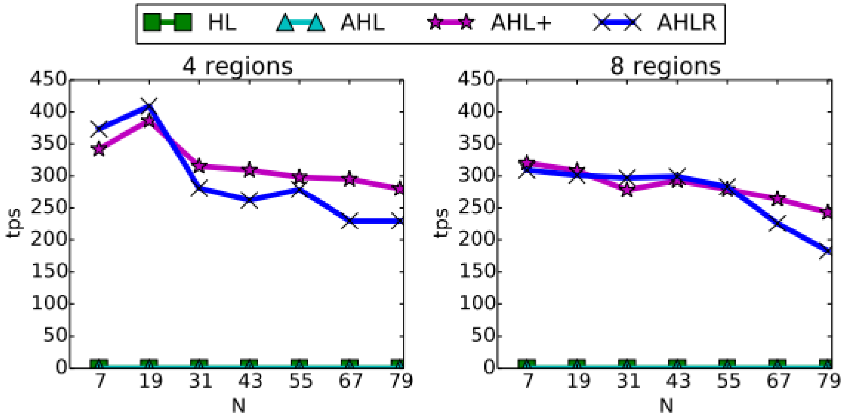


Figure 9: AHL+ performance on GCP (4 and 8 regions).

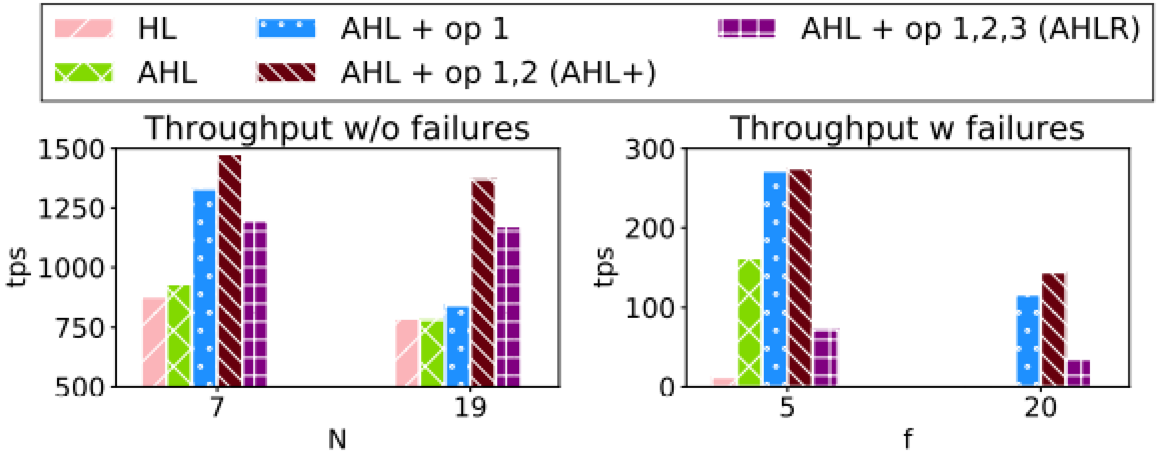


Figure 10: Effect of optimizations on throughput.

分片形成算法对比

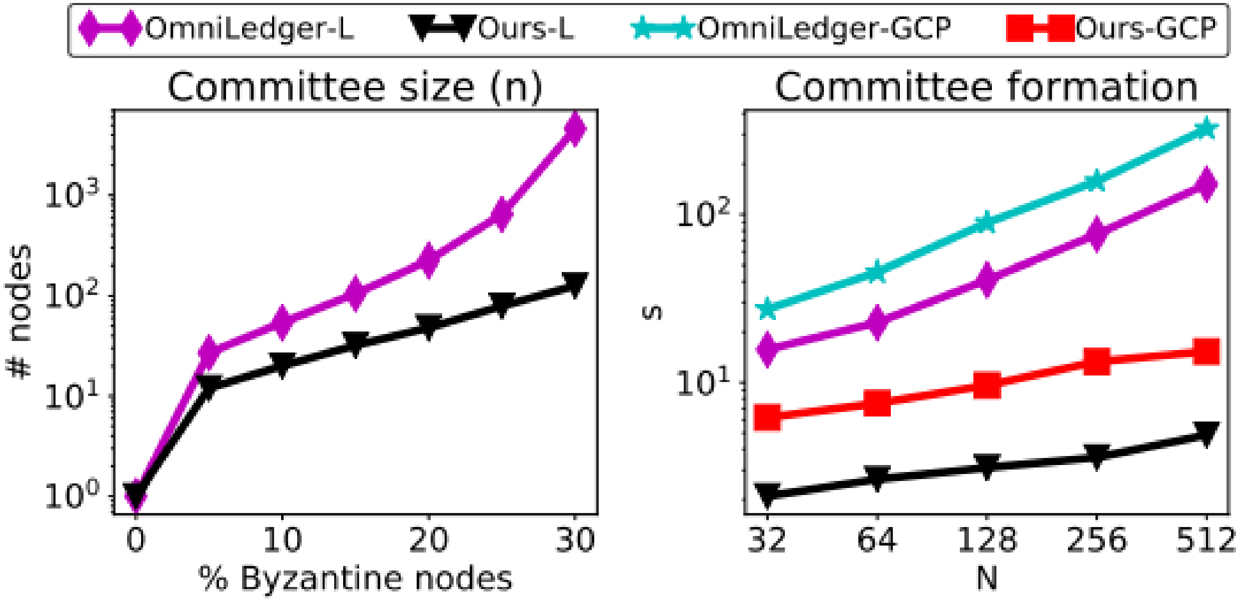
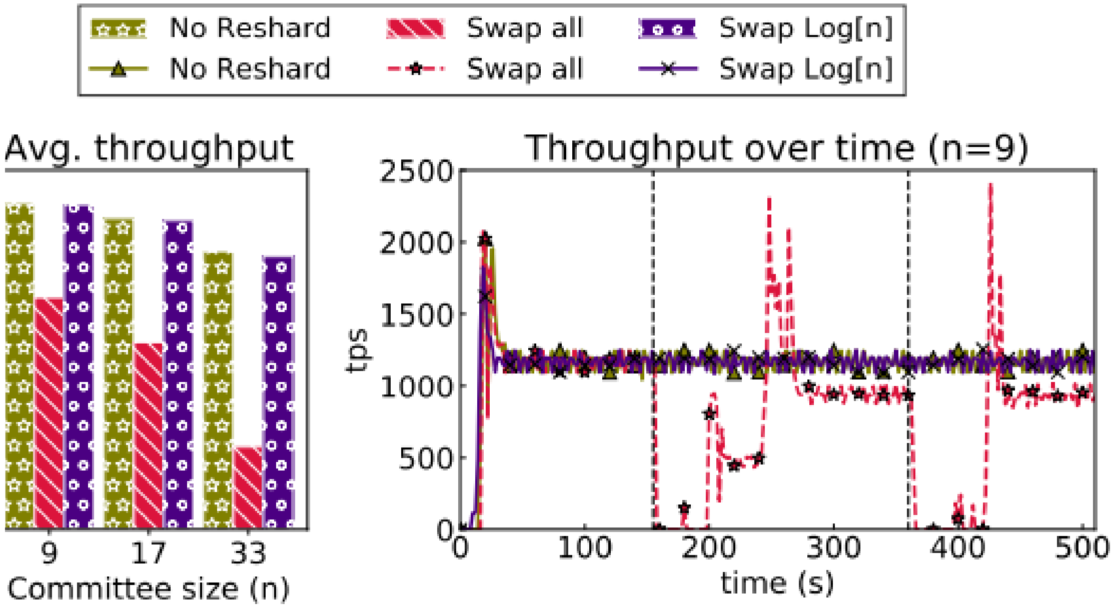


Figure 11: Evaluation of shard formation.



:e 12: Performance during shard reconfiguration.

分片表现

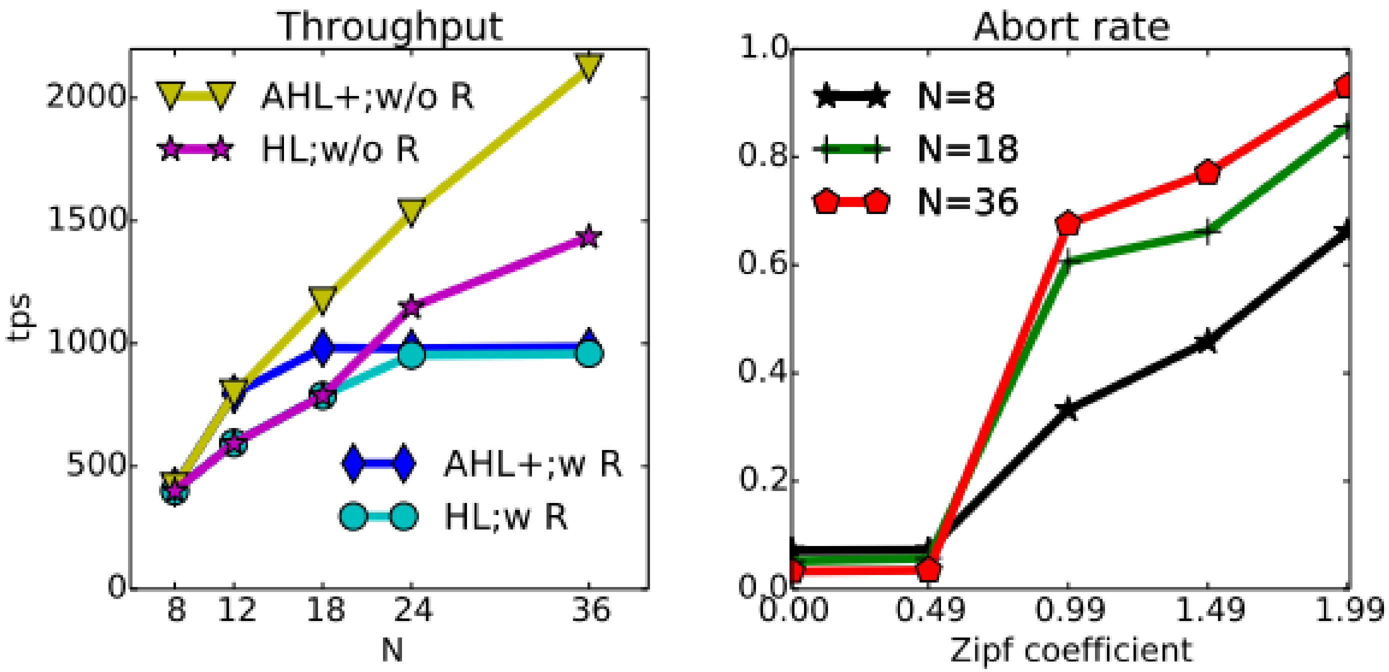


Figure 13: Sharding performance on local cluster with and without reference committee.

分片表现

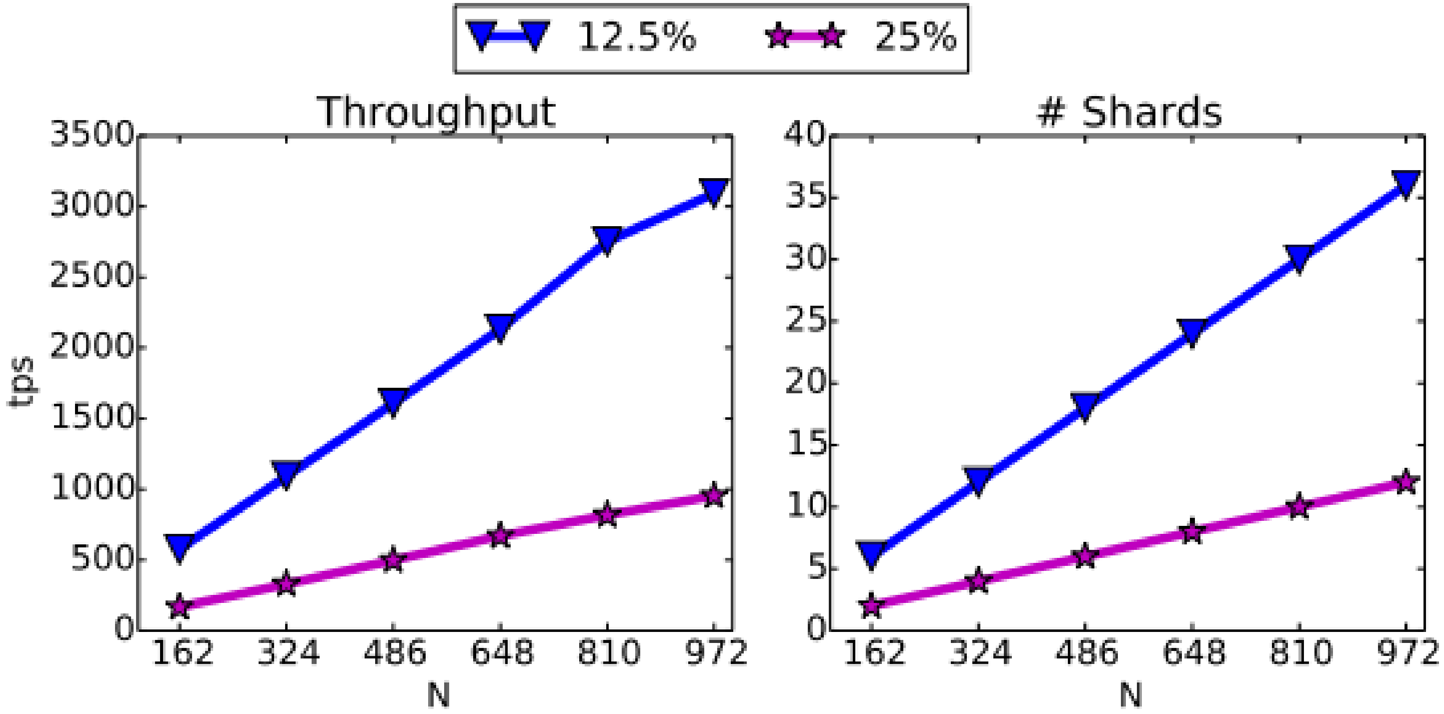


Figure 14: Sharding performance on GCP.

相关工作

- 分片区块链
  - **Elastico**
  - **OmniLedger**
  - **RapidChain**
  - Chainspace
- 扩展区块链数据库技术
  - 区块链存储
  - 执行引擎
- 链下扩展
  - 事务移出区块链

总结

- 指出挑战并提出解决方法
  - fault-scalable consensus protocols
  - shard formation protocol
  - coordination protocol

  - evaluation: 3000/s