Performance Benchmark

- We deployed Hyperledger, Ethereum and Parity
- The experiments run on 48-node commodity cluster.
  - Intel E5-1650 3.5GHz CPU
  - 32GB RAM
  - 2TB hard driver
- We collected comparison results in terms of our five metrics in macro benchmarks.
- We stress tested each individual layer using our micro benchmarks.

Performance Benchmark

Main findings (1/2)

- Hyperledger performs consistently better than Ethereum and Parity across the benchmarks. But it fails to scale up to more than 16 nodes.
- Ethereum and Parity are more resilient to node failures, but they are vulnerable to security attacks that forks the blockchain.
- The main bottlenecks in Hyperledger and Ethereum are the consensus protocols, but for Parity the bottleneck is caused by transaction signing.

Performance Benchmark

Main findings (2/2)

- Ethereum and Parity incur large overhead in terms of memory and disk usage. Their execution engine is also less efficient than that of Hyperledger.

- Hyperledger's data model is low level, but its exibility enables customized optimization for analytical queries of the blockchain data.

Throughput & Latency

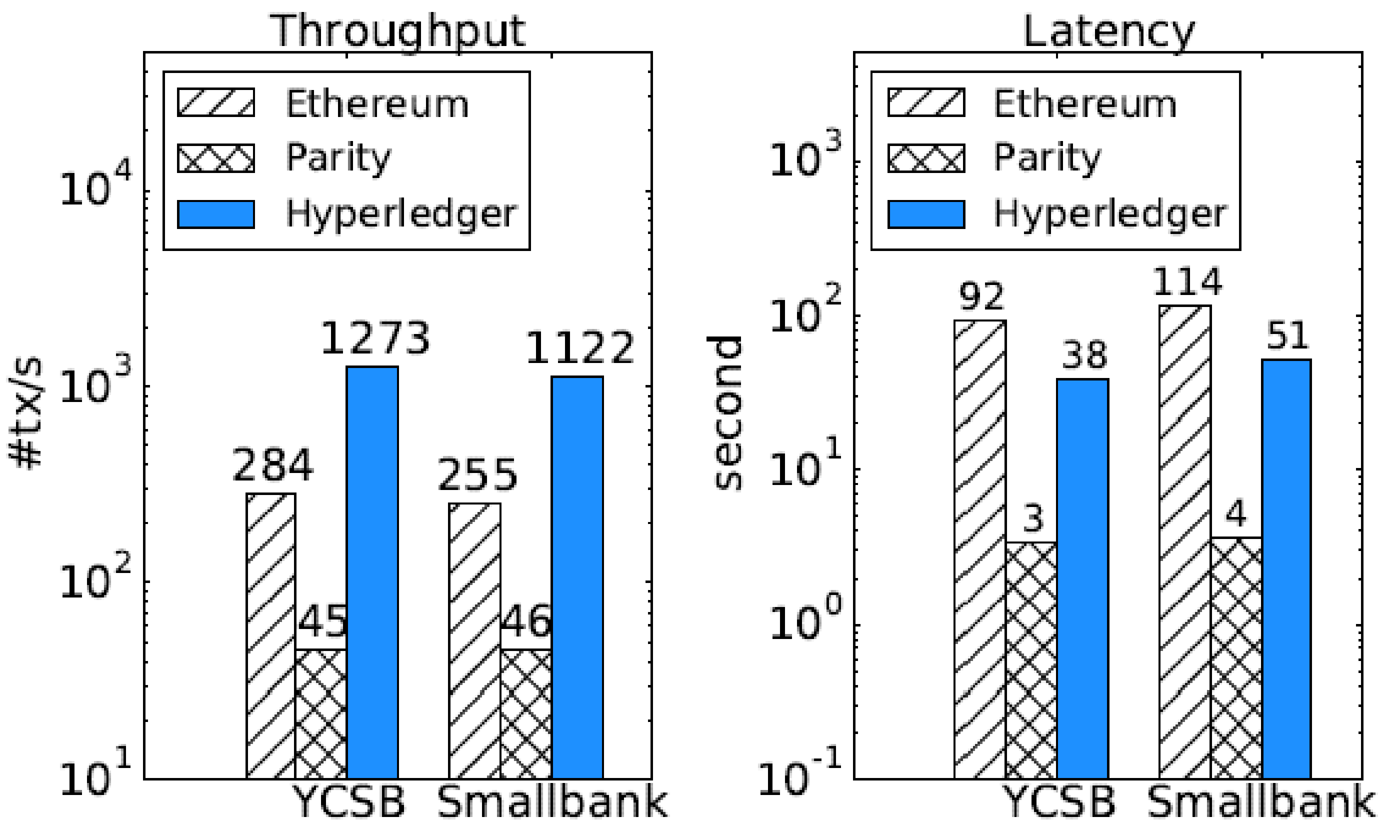

Figure: Throughput and latency of 3 systems over YCSB and SmallBank benchmark

Throughput & Latency

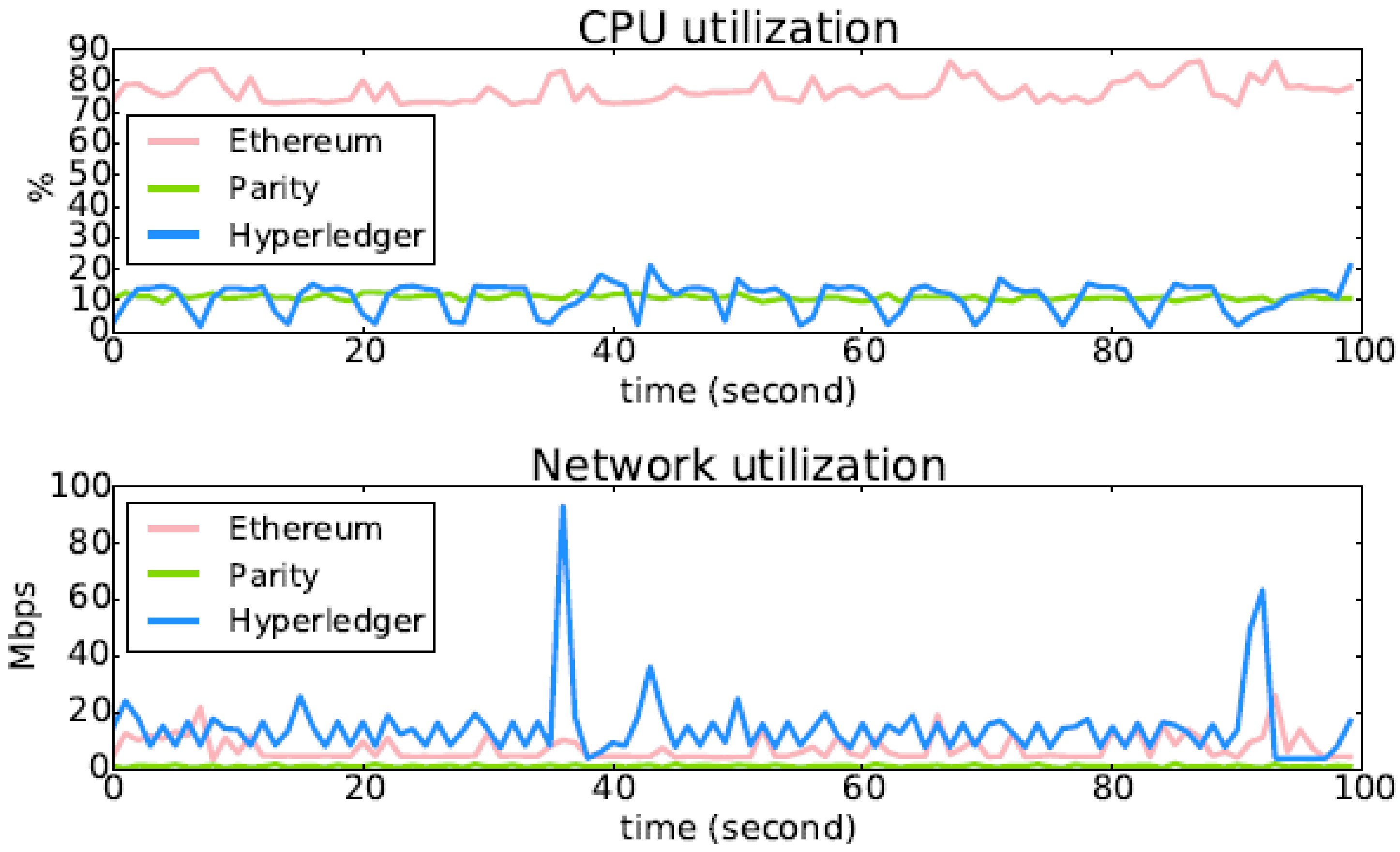

Figure: CPU & network resource utilization of 3 systems over YCSB benchmark

Throughput & Latency

Observations (1/2)

- The gap between Hyperledger and Ethereum is because of the difference in consensus protocol. Hyperledger is communication bound (PBFT) whereas Ethereum is CPU bound (PoW).

- Parity processes transactions at a constant rate, and that it enforces a maximum client request rate at around 80 tx/s. Parity achieves both lower throughput and latency than other systems.

Throughput & Latency

Observations (2/2)

- In Ethereum and Hyperledger, there is a drop of 10% in throughput and 20% increase in latency from YCSB to Smallbank. This suggest that there are non-negligible costs in the execution layer of blockchains.

Throughput & Latency

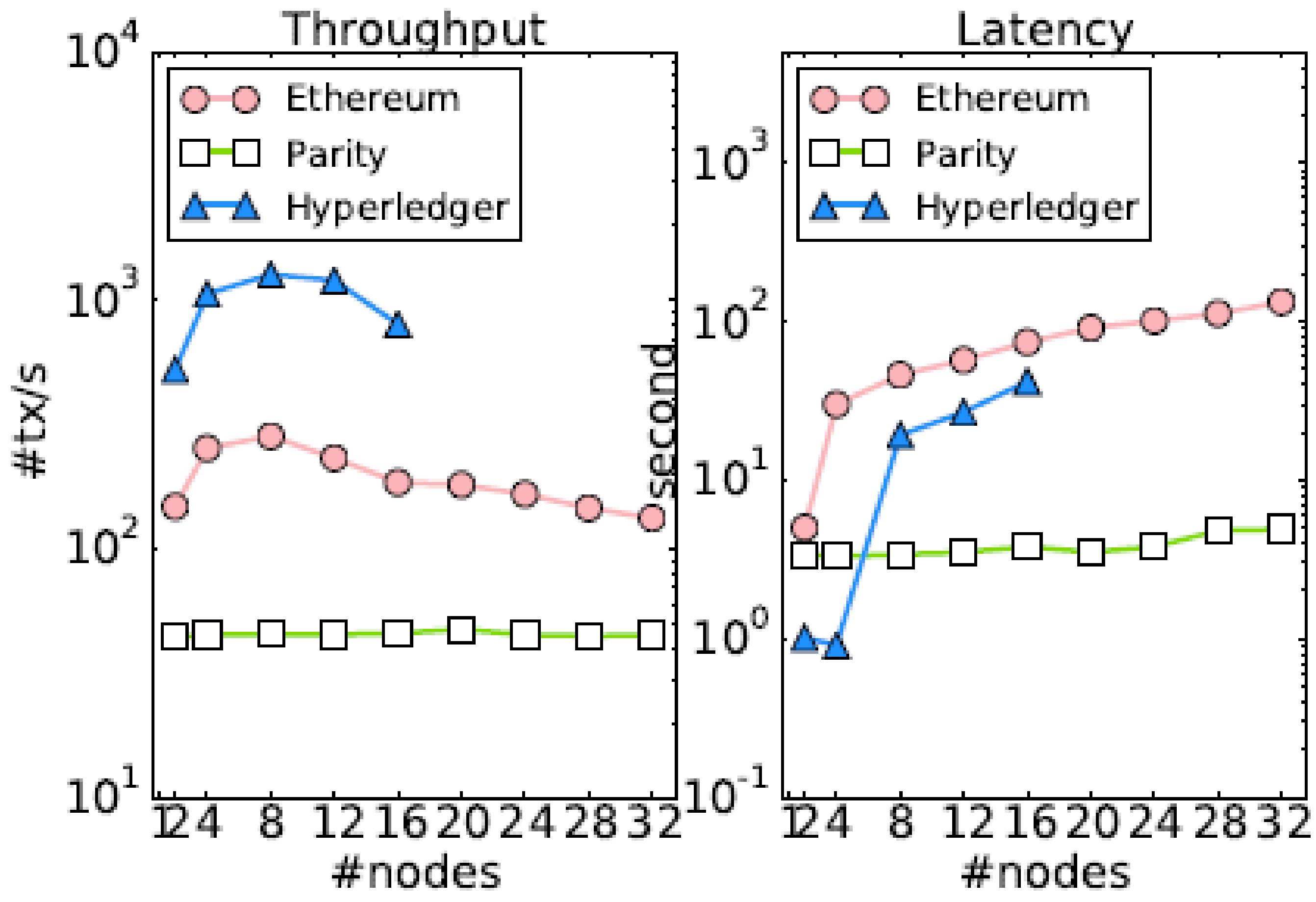

Figure: Performance scalability (with the same number of clients and servers).

Scalability

Observations

- Parity's performance remains constant as the network size and oered load increase, due to the constant transaction processing rate at the servers.
- Ethereum's throughput and latency degrade almost linearly beyond 8 servers.
- Hyperledger stops working beyond 16 servers due to flaws in the implementation of the consensus protocol.

Fault-tolerance & Security

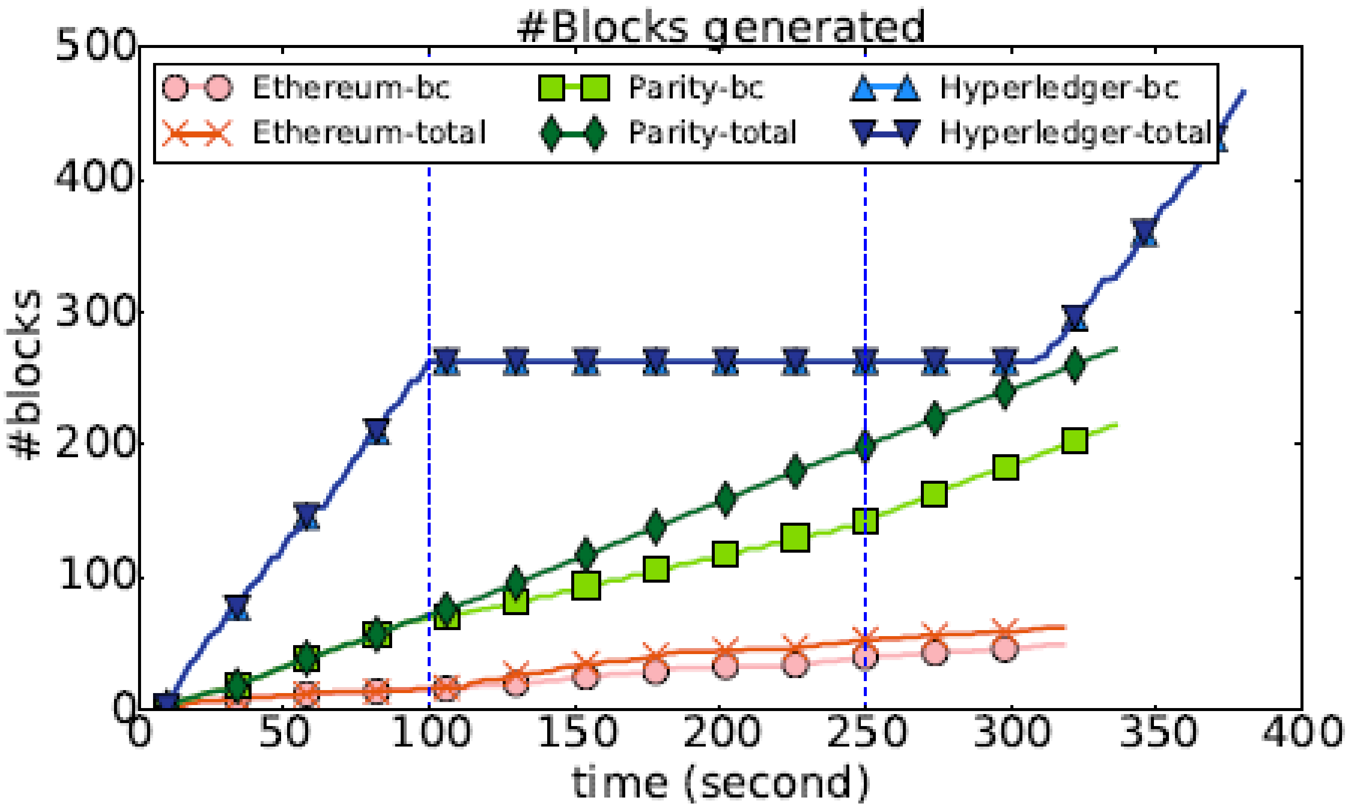

Figure: Blockchain forks caused by attacks that partitions the network in half at 100th second and lasts for 150th seconds. X-total means the total number of blocks generated in blockchain X, X-bc means the total number of blocks that reach consensus in blockchain X.

Fault-tolerance & Security

Observations

- Hyperledger is more vulnerable to fail-stop fault.
- Ethereum and Parity fork under network partition, they are vulnerable to fork attacks.
- Hyperledger has safety property for consensus because of PBFT protocol.
- Hyperledger uses more time to recovery from network partition.

Execution Layer – CPUHeavy

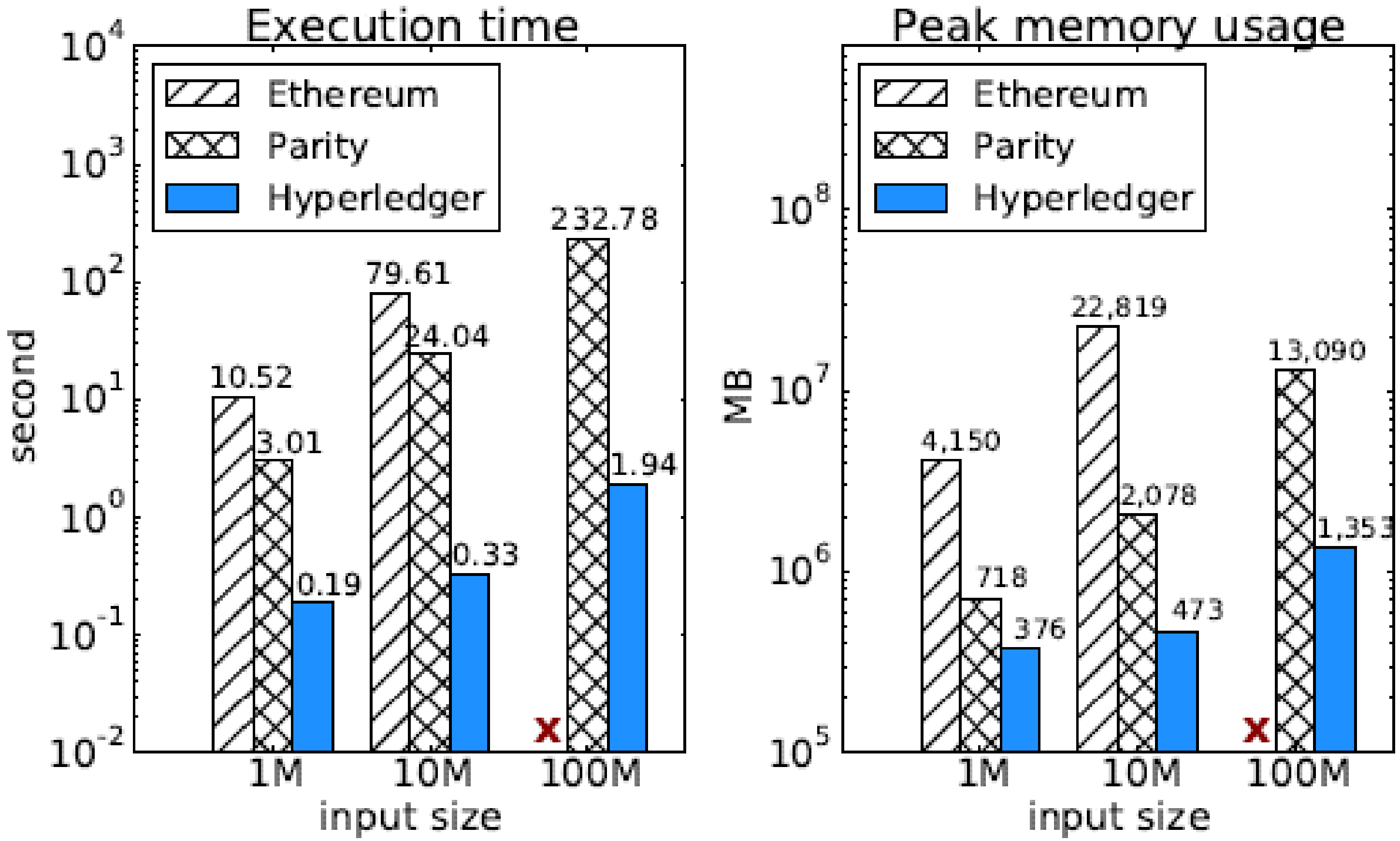

Figure: CPUHeavy workload, ‘X’ indicates Out-of-Memory error.

Execution Layer – CPUHeavy

Observations

- Ethereum and Parity use the same execution model (i.e., EVM), but Parity has more optimized implementation.

- Hyperledger's execution engine is more computation and memory efficient than EVM.

- All three systems fail to make use of the multi-core architecture.

Outline

- Introduction
    - Backgrounds
    - Problem Statement
    - Related Works
- BlockBench Framework
    - System Design
    - Implementation
- Performance Benchmark
    - Macro Benchmarks
    - Micro Benchmarks
- Discussion
- Conclusion

Discussion

Bringing database designs into blockchain

Huge performance gap between blockchains and transactional databases

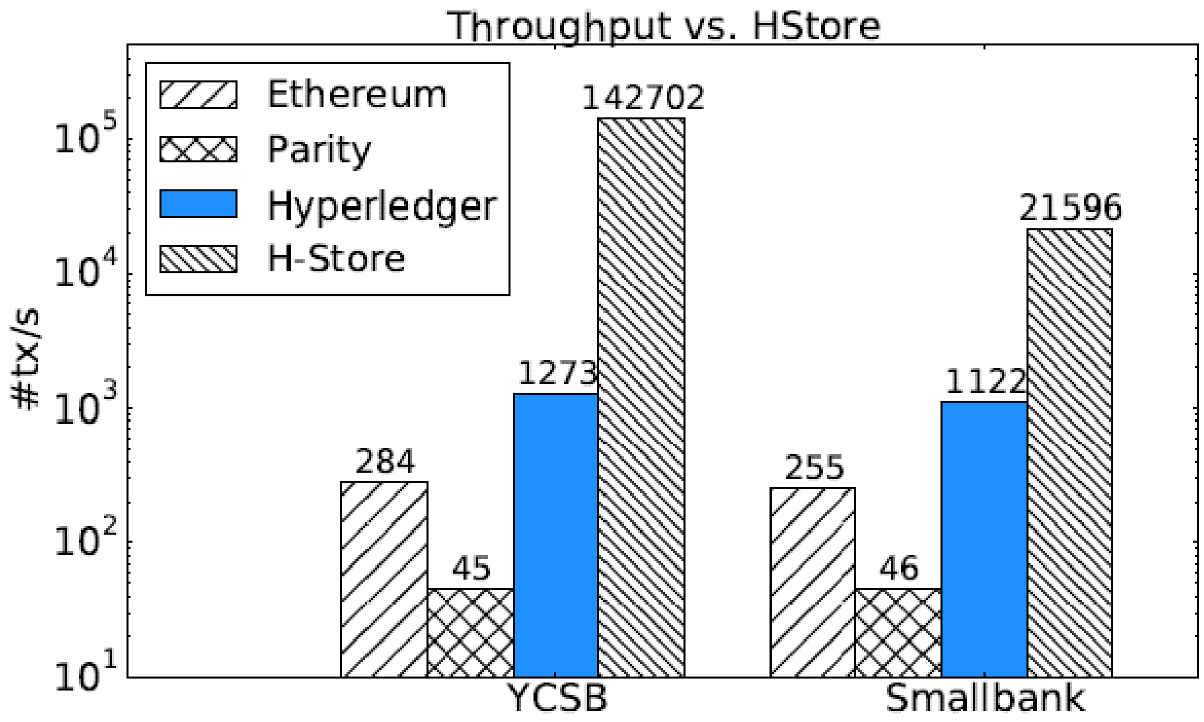

Figure: Performance of the three blockchain systems versus H-Store.

Discussion

Bringing database designs into blockchain

- Decouple storage, execution engine and consensus layer from each other, then optimize and scale them independently.

* Our system UStore demonstrates that a storage designed around the blockchain data structure is able to achieve better performance than existing implementations.

Discussion

Bringing database designs into blockchain

- Embrace new hardware primitives.

* For blockchain, using trusted hardware, the underlying Byzantine fault tolerance protocols can be modified to incur fewer network messages.

* Systems like Parity and Ethereum can take advantage of multi-core CPUs and large memory to improve contract execution and I/O performance.

Conclusion

- **BlockBench** , to our knowledge, is the first comprehensive benchmark framework for private blockchain systems.
- We hope our results will serve as a baseline for further development of blockchain technologies.
- Further Information:
  - Paper: https://arxiv.org/abs/1703.04057 (to appear in ACM SIGMOD 2017)
  - Code+Workloads at project web site: http://www.comp.nus.edu.sg/~dbsystem/blockbench/

Thanks!