BLOCKBENCH: A Framework for Analyzing Private Blockchains

Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi,

Outline

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

Outline

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

Backgrounds

Bitcoin & the Blockchain

"Satoshi Nakamoto"
2009

Cryptocurrency

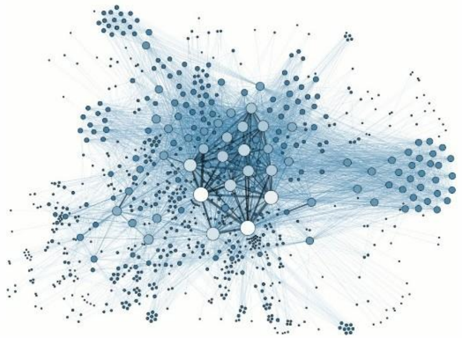
- No central bank
- Transferring coins through trustless P2P network
- ~1200 USD per Bitcoin (coinbase.com 10/03/2017)

Technology

- Blockchain
- Distributed shared ledger
- Cryptograhy (SHA-256, PKI)
- Consensus model
- Smart contracts

4 Key Concepts of Blockchain

Distributed shared ledger



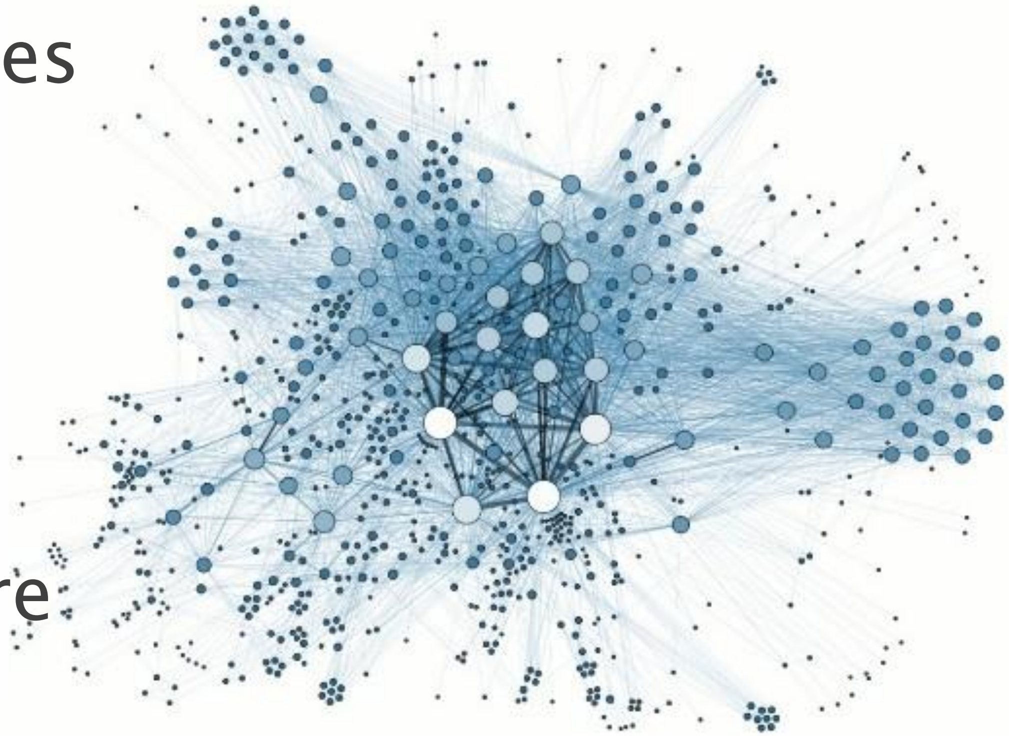
Cryptography

Consensus

Smart contracts

4 Key Concepts of Blockchain: Distributed Shared Ledger

- Group of replicated logs/databases (nodes)
- Transactions packed in blocks
- All nodes hold all transactions
- Parties identified with public key (= anonymised)
- Resilient for failure of one or more nodes



4 Key Concepts of Blockchain:
1.Distributed Shared Ledger

BITNODES

Bitnodes is currently being developed to estimate the size of the Bitcoin network by finding all the reachable nodes in the network.

GLOBAL BITCOIN NODES DISTRIBUTION

Reachable nodes as of Sun Jun 14 2015 14:01:53 GMT+0200.

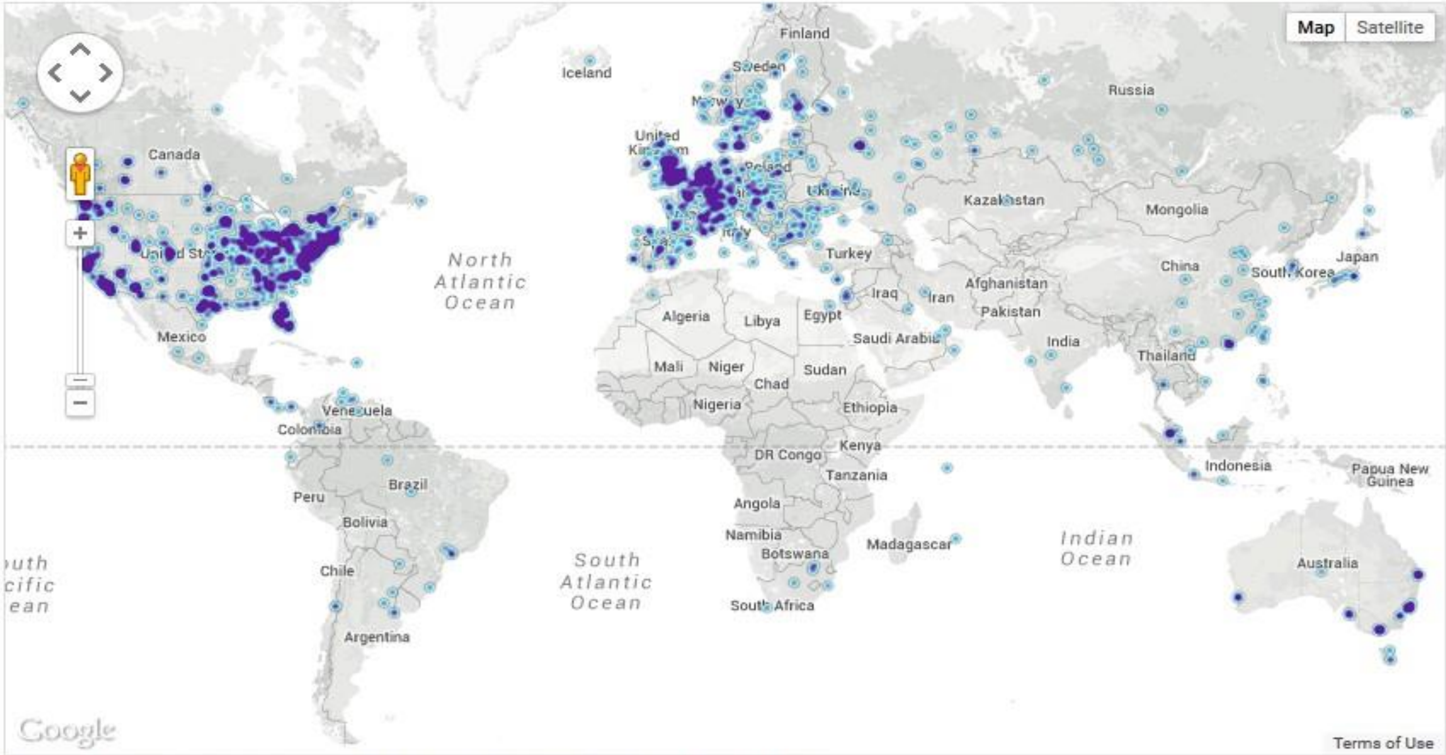
5987 nodes

24-hour charts »

Top 10 countries with their respective number of reachable nodes are as follow.

RANK	COUNTRY	NODES
1	United States	2161 (36.09%)
2	Germany	626 (10.46%)
3	France	442 (7.38%)
4	United Kingdom	375 (6.26%)
5	Netherlands	307 (5.13%)
6	Canada	302 (5.04%)
7	Russian Federation	187 (3.12%)
8	Australia	136 (2.27%)
9	Sweden	116 (1.94%)
10	China	102 (1.70%)

More (85) »



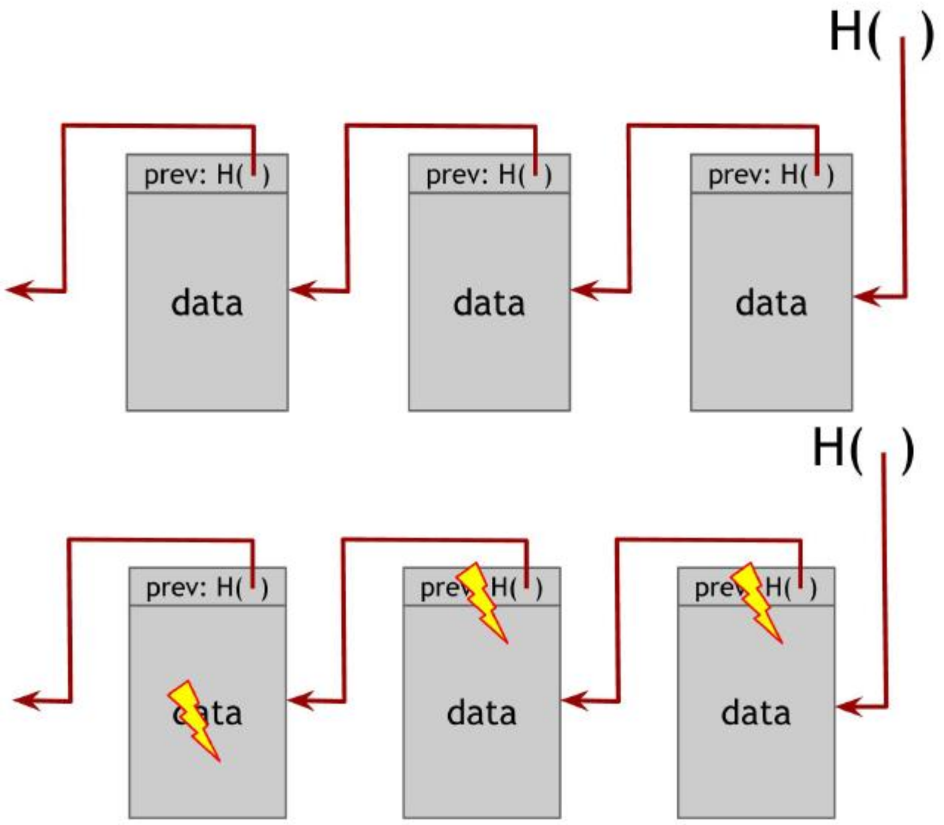
Map shows concentration of reachable Bitcoin nodes found in countries around the world.

JOIN THE NETWORK

Be part of the Bitcoin network by running a full Bitcoin node, e.g. Bitcoin Core.

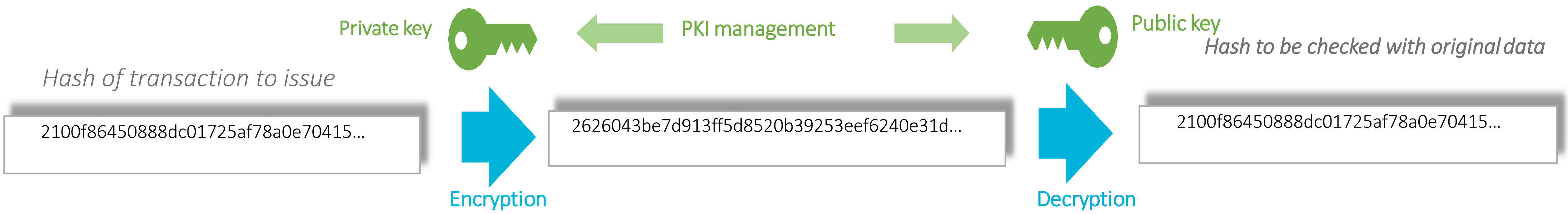
4 Key Concepts of Blockchain: 2.Cryptographic (1/2)

Tamper-proof log blocks using hash pointer



4 Key Concepts of Blockchain: 2.Cryptographic (2/2)

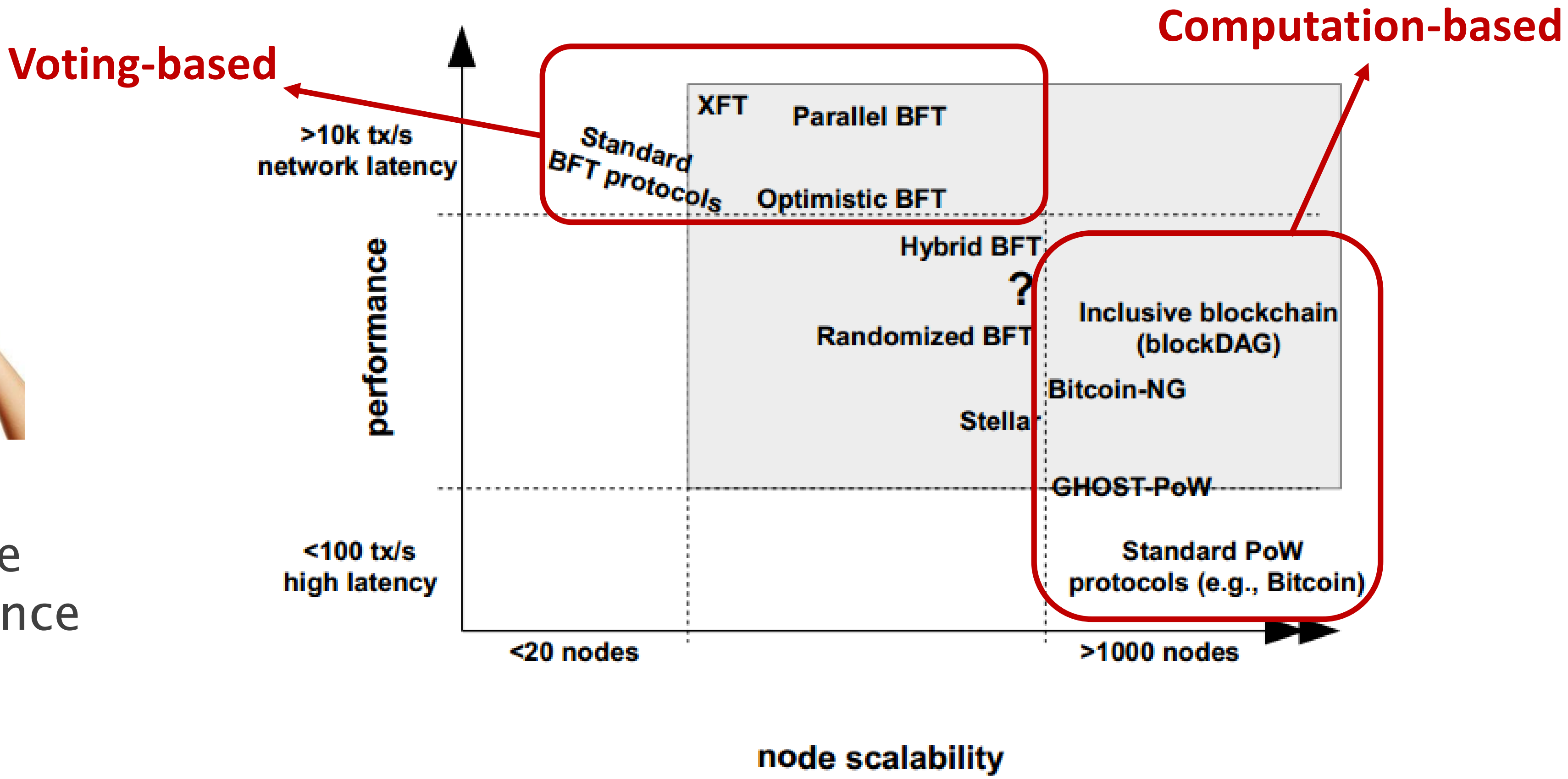
Asymmetric cryptography digital signature system



4 Key Concepts of Blockchain: 3.Consensus

Consensus

- No single point failure
- Byzantine fault tolerance

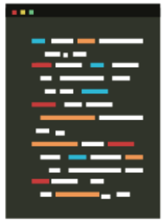


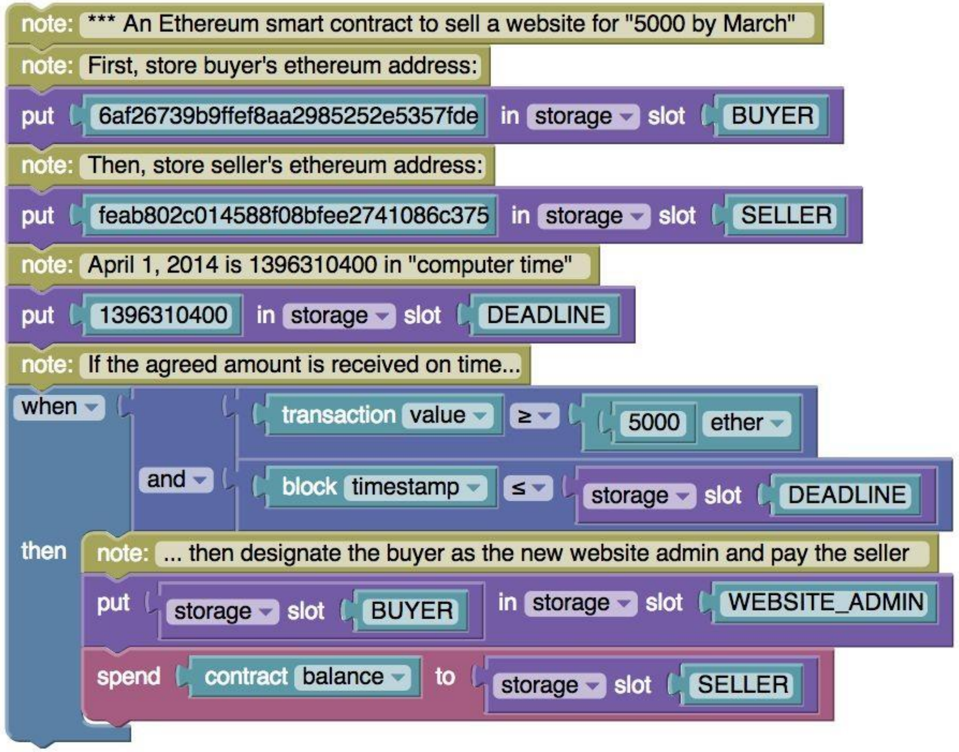
Cite: Vukolić, Marko. "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication."

4 Key Concepts of Blockchain: 4.Smart-Contract

Smart contracts

- Business logic that can be assigned to a transaction on the blockchain
- Acts as a 'notary' of blockchain transactions
- Holds conditions under which specific actions can/must be performed
- Facilitates escrow services
- Can't be modified without predefined permissions





Category of blockchains

Public blockchain V.S. Private blockchain

- The majority of financial services firms exploring the use of blockchain are looking at private or semi-private blockchains, rather than the fully decentralized public blockchains

Public blockchains

- No authoritative permission required in order to participate
- Participants are not vetted
- Mechanisms for maintaining the network against attacks and unwanted parties therefore add cost and complexity to the network
- Usually use computation-based consensus protocols

Private blockchains

- Participants are known and identified.
- Legal contracts can help with system mechanisms.
- Usually use voting-based consensus protocols

Problem Statement

Quest for understanding of private blockchain performance

- Design a general benchmark framework to find out to what extent can blockchain handle data processing workload.

Problem Statement

Quest for understanding of private blockchain performance

- Design a general benchmark framework to find out to what extent can blockchain handle data processing workload.

Our framework will:

- Help blockchain application developers to assess blockchain's potentials in meeting the application needs.
- Help blockchain platform developers to identify and improve on the performance bottlenecks.

Related Works

- TPC benchmark series
  - End-to-end macro-benchmarks
  - Focus on relational data model
- Yahoo! Cloud Serving Benchmark (YCSB)
  - For NoSQL data storage
  - To evaluate performance and scalability
- GridMix, PigMix, TeraSort/GraySort, etc.
  - Benchmark for MapReduce-like systems
- BigBench
  - Industry standard end-to-end benchmark
  - For big data processing systems

No benchmark for private blockchains at the moment

Outline

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

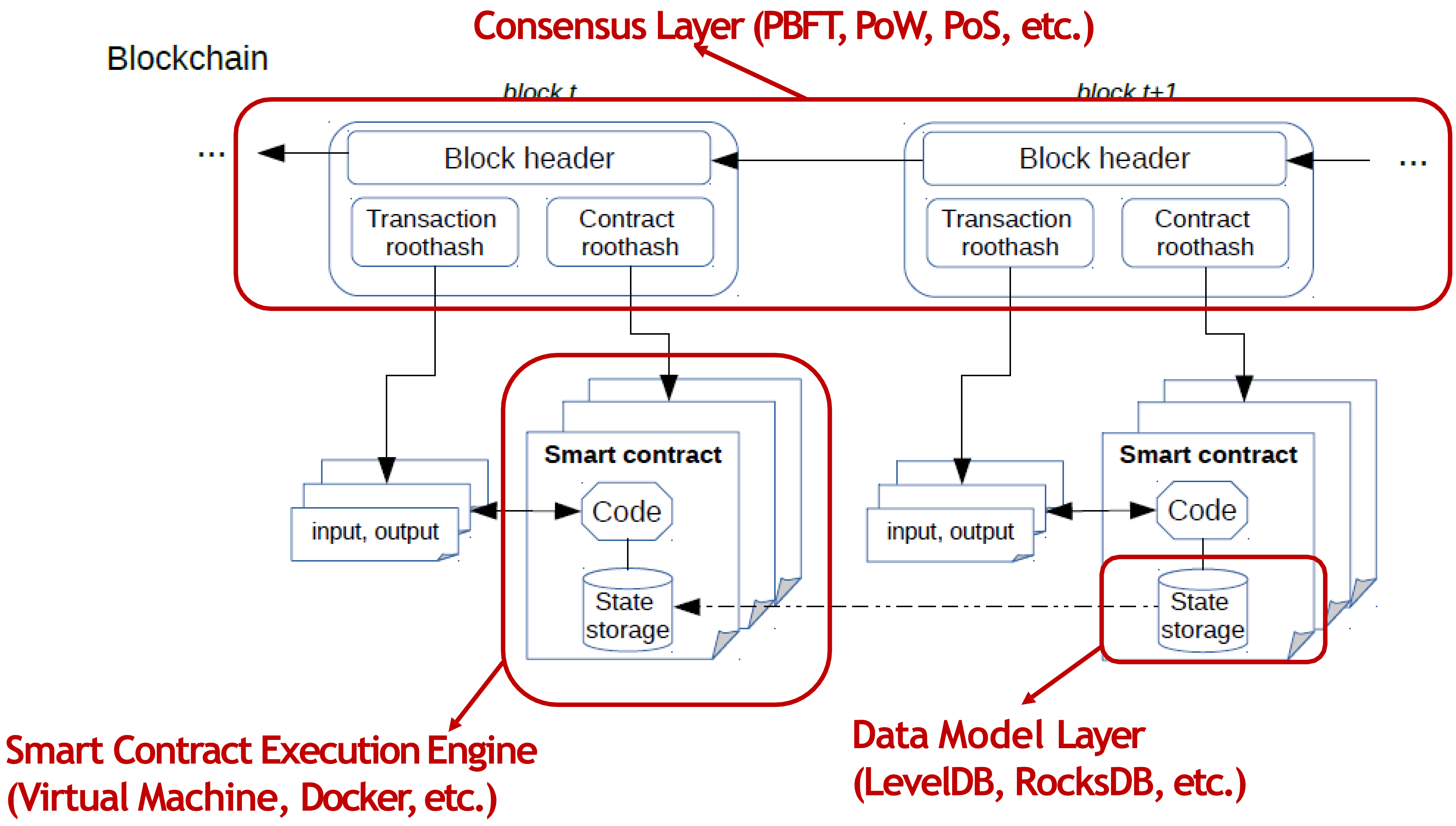
Challenges

- Three main challenges

Challenge 1: a blockchain system comprises many parts, we observe that a wide variety of design choices are made among different platforms at almost every single detail.

Approach: We extract the common modules of blockchain platform, and divide the blockchain architecture into three modular layers and focus our study on them: the consensus layer, the data model layer and smart-contract execution layer.

Challenges


Consensus Layer (PBFT, PoW, PoS, etc.)
Blockchain
block t
block t+1
...
Block header
Transaction roothash
Contract roothash
Block header
Transaction roothash
Contract roothash
...
Smart contract
Code
State storage
input, output
Smart contract
Code
State storage
input, output
Smart Contract Execution Engine
(Virtual Machine, Docker, etc.)
Data Model Layer
(LevelDB, RocksDB, etc.)

Challenges

- Three main challenges

Challenge 2: there are many different choices of platforms, but not all of them have reached a mature design, implementation and an established user base.

Approach: We start designing BlockBench based on three most mature platforms which support smart-contract funcionality, namely Hyperledger Fabric, Ethereum and Parity, and the framework is general to support future platforms.

Challenges

- Three main challenges

Challenge 3: There is lack of a database-oriented workloads for blockchain.

Approach: We treat blockchain as a key-value storage coupled with an engine which can realize both transactional and analytical functionality via smart contracts.
We design and run both transaction and analytics workloads in our benchmark framework.

Workloads

Smart contracts	Description
YCSB	Key-value store
Smallbank	OLTP workload
EtherId	Name registrar contract
Doubler	Ponzi scheme
WavesPresale	Crowd sale
VersionKVStore	Keep state's versions (Hyperledger only)
IOHeavy	Read and write a lot of data
CPUHeavy	Sort a large array
DoNothing	Simple contract, do nothing

Macro-Benchmarks

Micro-Benchmarks

Storage-oriented

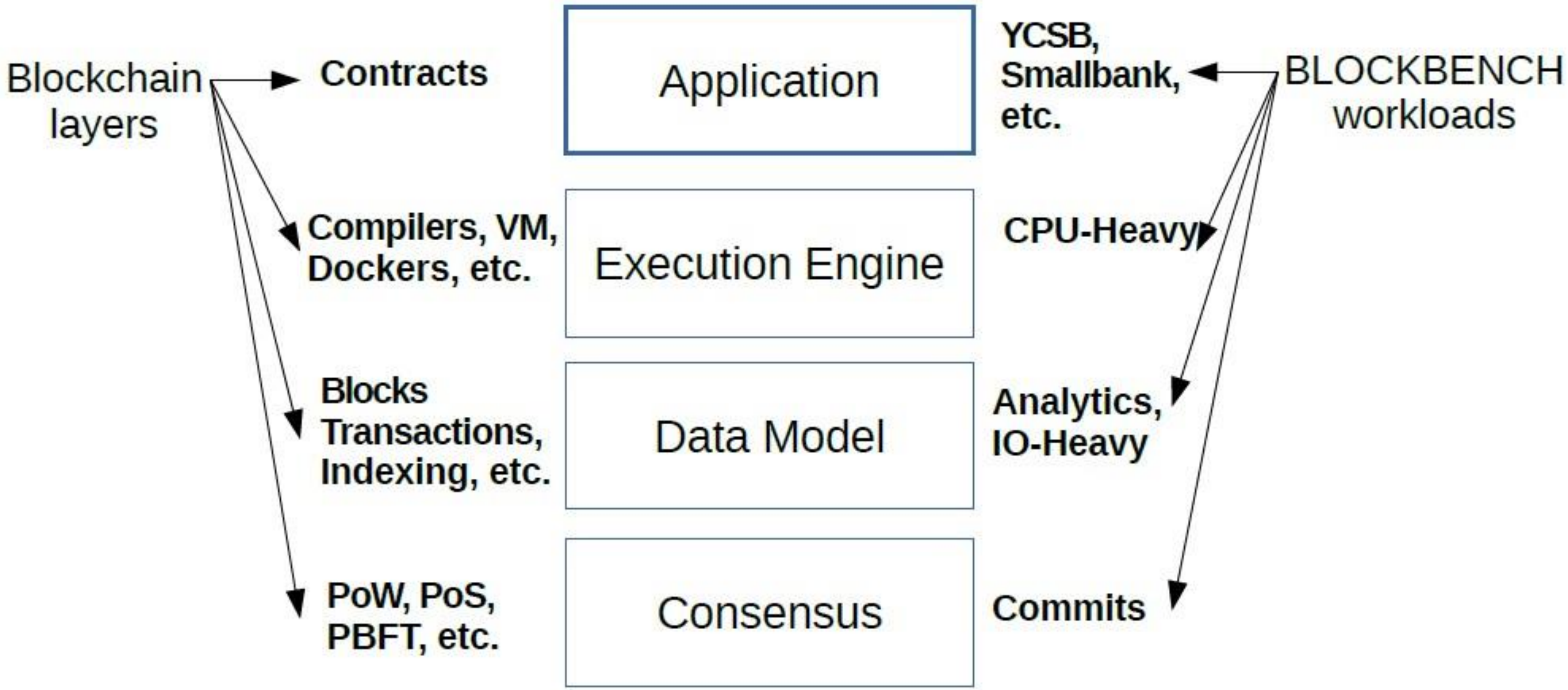
Application-oriented

Data model

→ Execution engine

→ Consensus layer

Framework Design



Workloads

Smart contracts	Description
YCSB	Key-value store
Smallbank	OLTP workload
EtherId	Name registrar contract
Doubler	Ponzi scheme
WavesPresale	Crowd sale
VersionKVStore	Keep state's versions (Hyperledger only)
IOHeavy	Read and write a lot of data
CPUHeavy	Sort a large array
DoNothing	Simple contract, do nothing

Macro-Benchmarks

Storage-oriented

Application-oriented

Micro-Benchmarks

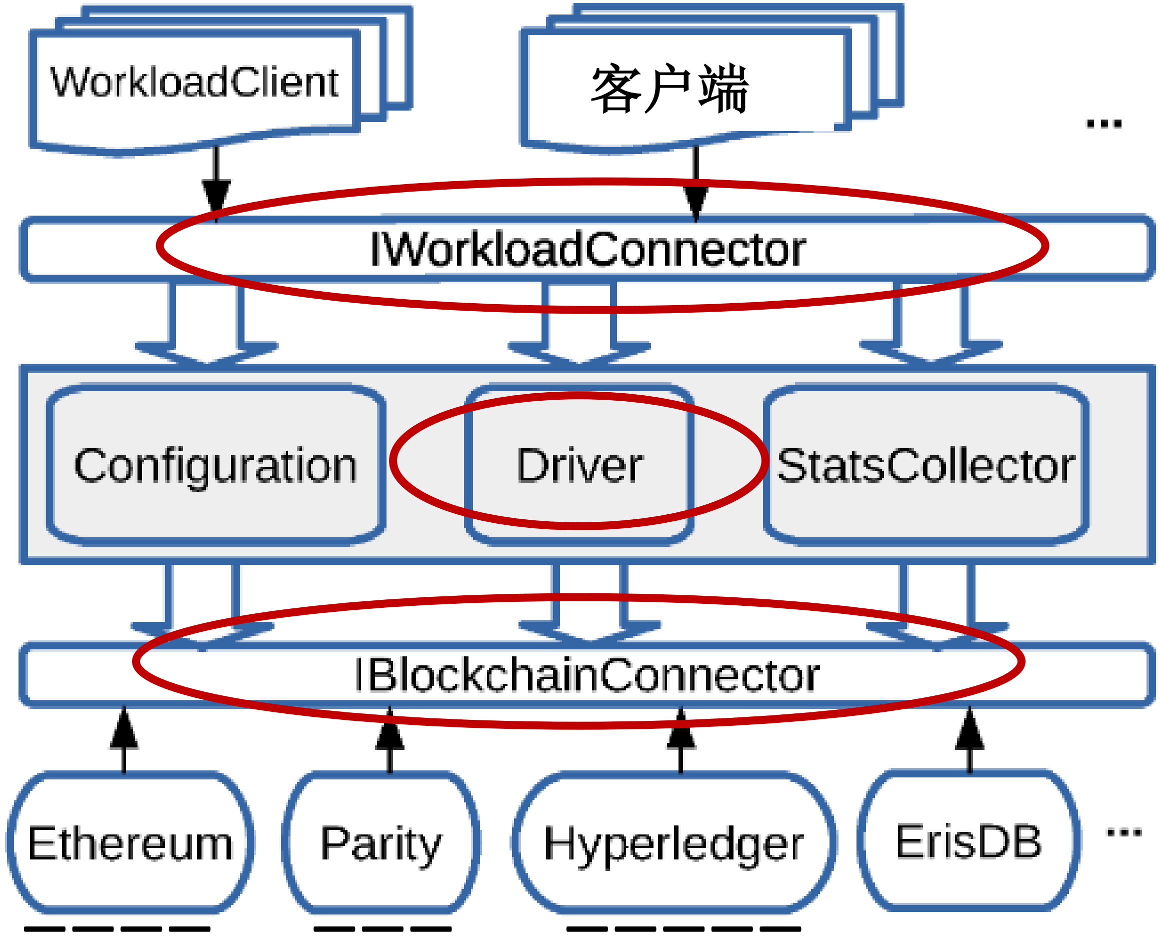
Data model

→ Execution engine

→ Consensus layer

BlockBench 的软件栈

将工作负载部署于客户端



通过异步驱动，执行工作负载（workloads）得到分析数据。

将三个平台链接到后端

评测指标

01 交易吞吐量（tps）

02 交易延迟

03 可扩展性

增加节点数量、并发工作负载时，交易量和延迟的变化来衡量。

04 容错性

节点失效时，交易量和延迟的变化来衡量。

05 安全性指标

利用BGP 攻击，在一段时间内分割节点群。
然后导致区块链分叉（fork）。
主链上产生的块和用户确认块数之比，可以用来衡量安全性。
主链上产生的块越多，“双花”攻击、自私挖矿发生的概率就越低。

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

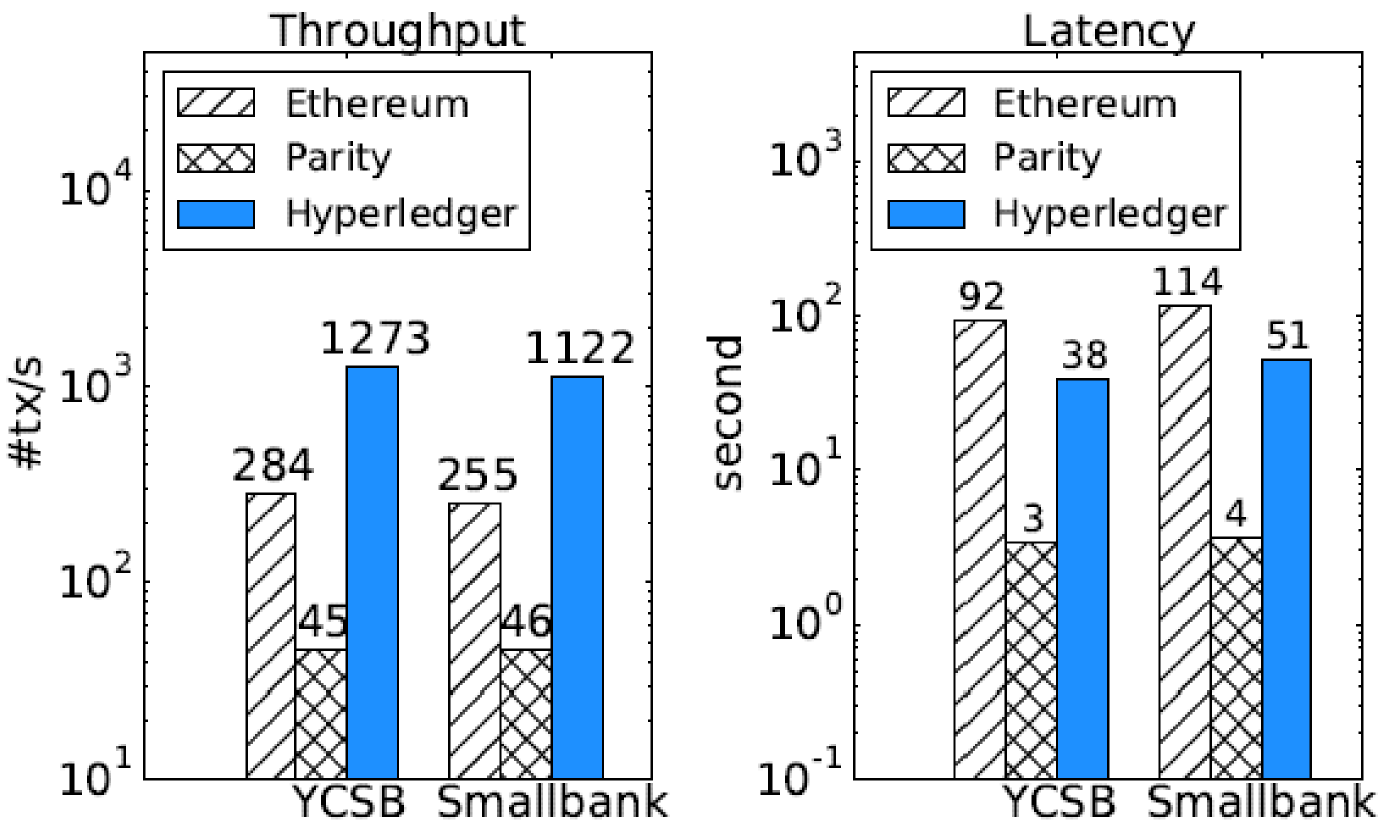


Figure: Throughput and latency of 3 systems over YCSB and SmallBank benchmark

Throughput & Latency

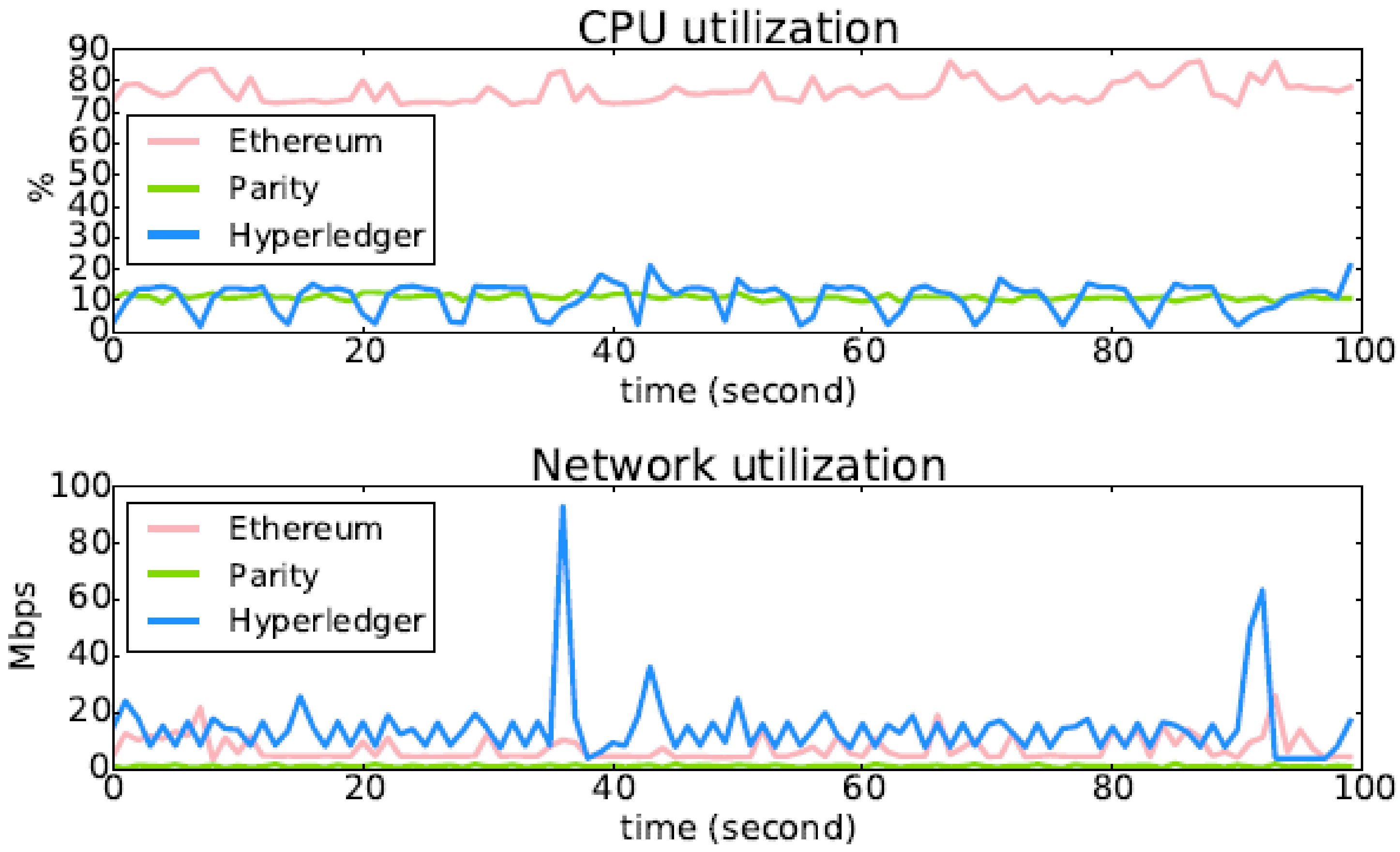


Figure: CPU & network resource utilization of 3 systems over YCSB benchmark

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

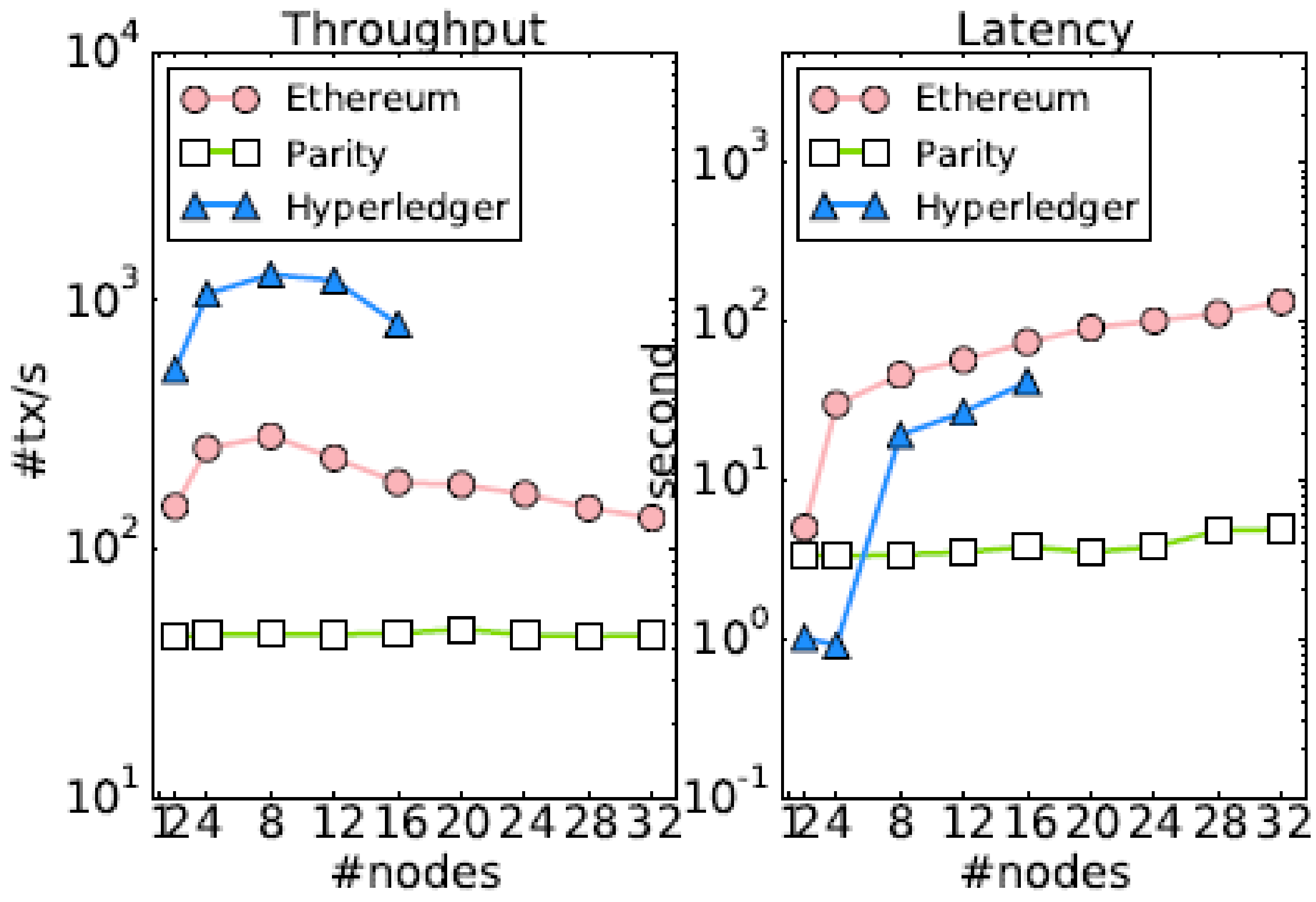


Figure: Performance scalability (with the same number of clients and servers).

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

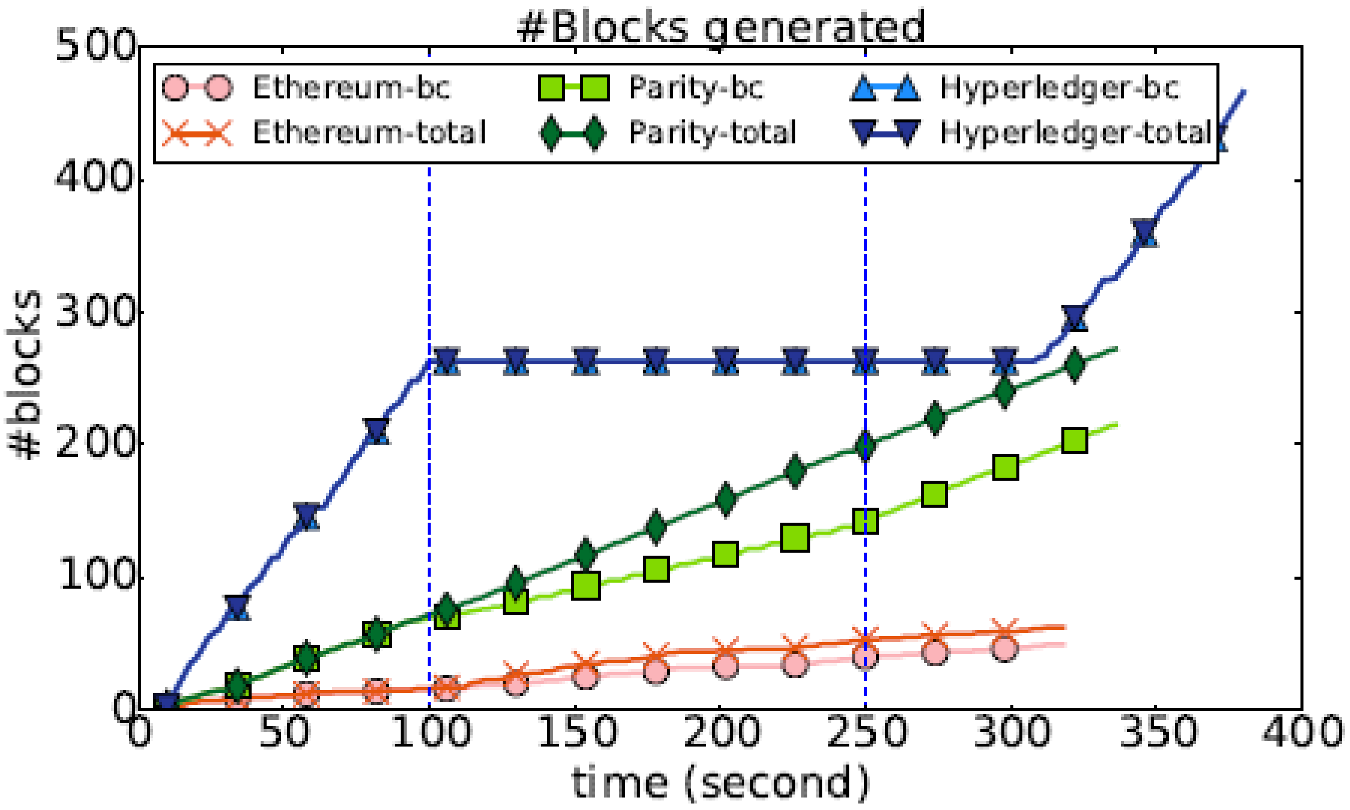


Figure: Blockchain forks caused by attacks that partitions the network in half at 100th second and lasts for 150th seconds. X-total means the total number of blocks generated in blockchain X, X-bc means the total number of blocks that reach consensus in blockchain X.

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

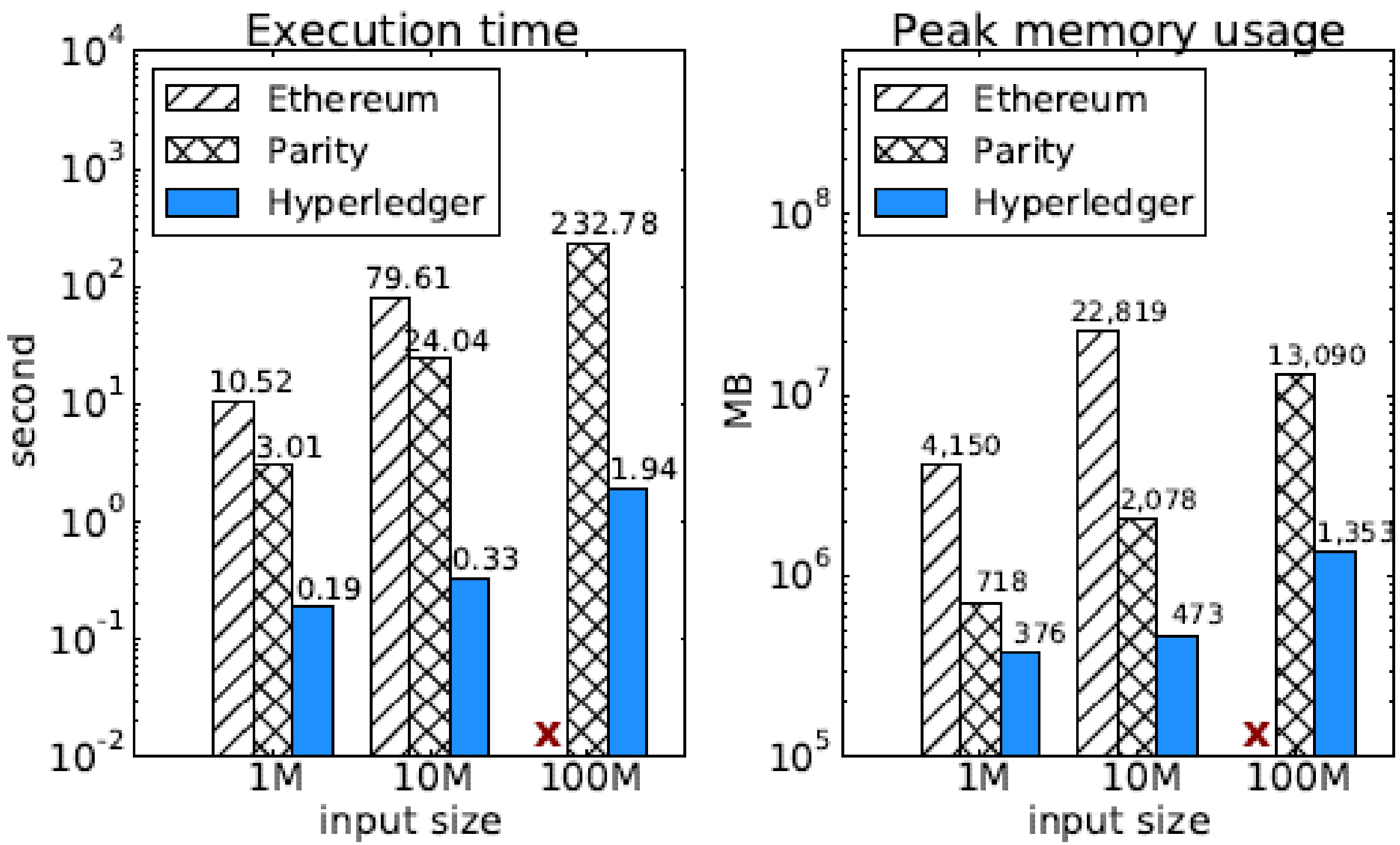


Figure: CPUHeavy workload, ‘X’ indicates Out-of-Memory error.

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

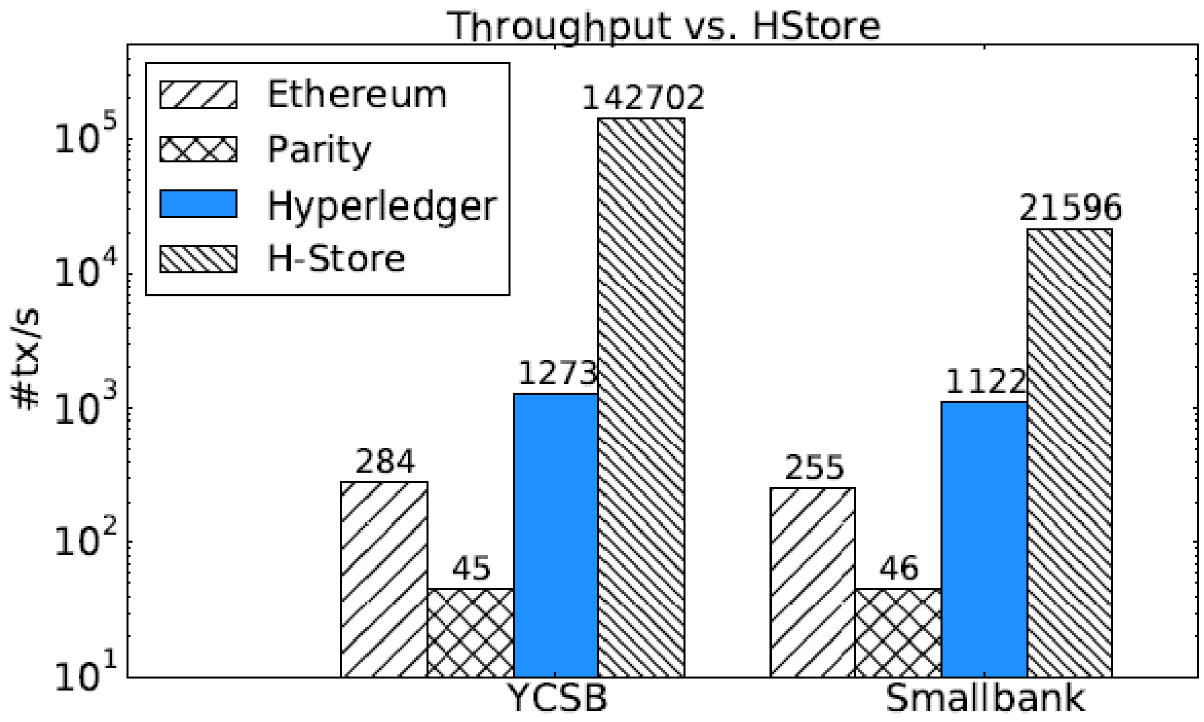


Figure: Performance of the three blockchain systems versus H-Store.

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!