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

Acknowledgement: many diagrams are owned by internet users which we use only for illustration purposes

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)

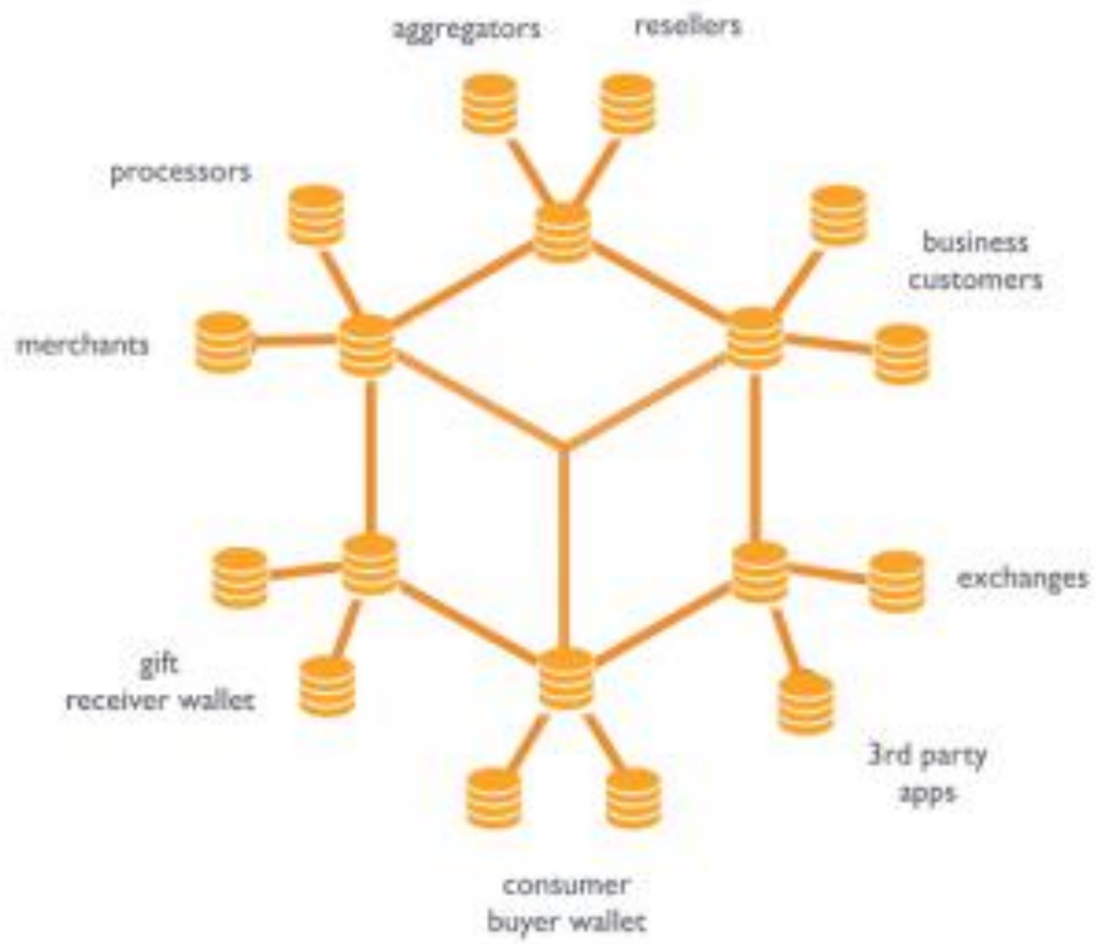
Technology

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

Backgrounds

Blockchains

Blockchains are distributed ledgers – or decentralized databases – that enable parities who do not fully trust each other to form and maintain consensus about the existence, status and evolution of a set of shared facts.



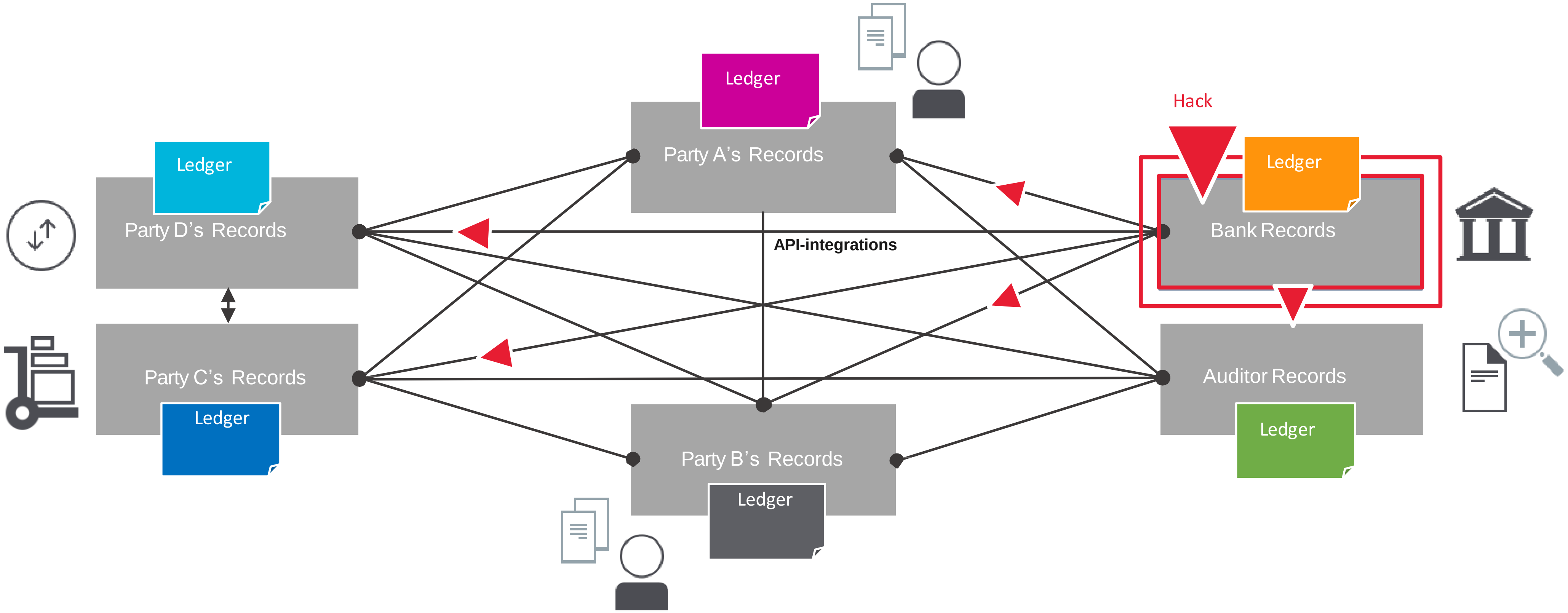
Backgrounds

Smart Contracts

Programs execute real-world contract logic that are encrypted and stored on distributed digital-ledger systems (blockchains), ensuring all parities are working off the same synchronized version, which cannot be unilaterally altered or tampered with.

Need for Blockchain and Smart Contracts

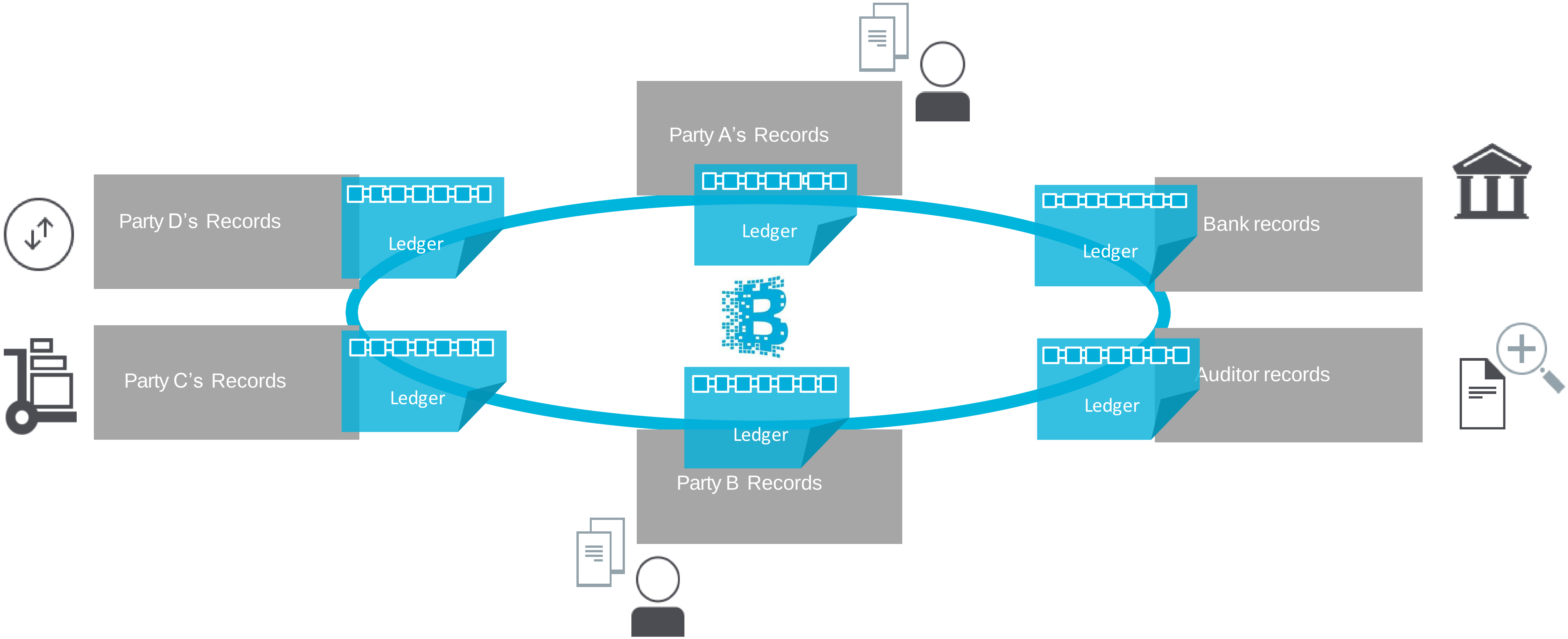
Information & asset exchange in business networks – Separate ledgers



Inefficient, expensive, error sensitive and vulnerable

Need for Blockchain and Smart Contracts

Information & asset exchange in business networks – Shared ledger



Consistency, efficiency, security and resilience

Need for Blockchain and Smart Contracts

Real world example #1. R3CEV financial consortium

- A consortium of more than 70 the world biggest financial institutions.
- Research and develop blockchain system in the financial services.
- Develop and test smart-contract templates that simplify legal documentation.

Need for Blockchain and Smart Contracts

Real world example #2. Linux Foundation Hyperledger Project

- a cross-industry collaborative project started in December 2015 by the Linux Foundation.
- Focus on distributed ledger to support global business transactions, including major technological, financial, and supply chain companies.

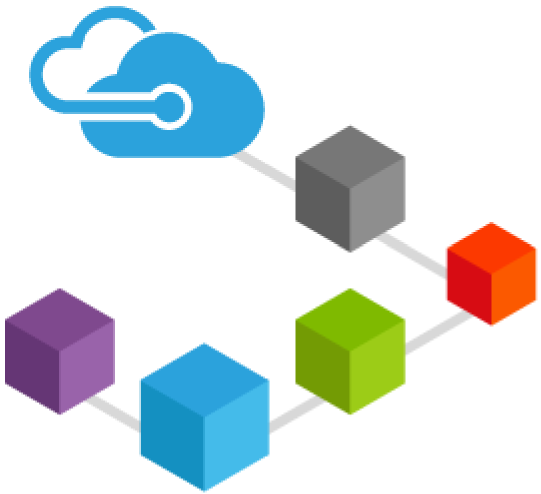
PREMIER MEMBERS

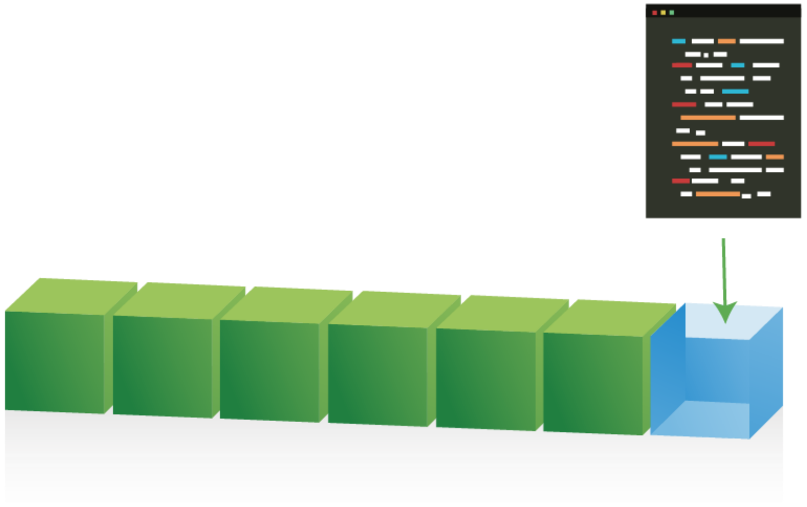
GENERAL MEMBERS

Need for Blockchain and Smart Contracts

Real world example #3. Microsoft and IBM Blockchain-As-A-Service

- Microsoft Azure cloud platform support many open–source blockchain platforms, e.g., Etheruem and ErisDB, as well as their own blockchain named Bletchley.
- IBM Bluemix provide Hyperledger Fabric platform as a service.





Need for Blockchain and Smart Contracts

More real world examples…

Financial institutions show huge interest in Blockchain by publishing many research reports


EQUITY RESEARCH | May 24, 2016
Is the hype around blockchain justified? Since Bitcoin introduced the world to the concept of secure distributed ledgers, much has been written about their potential to address other business problems. But the discussion often remains abstract, focusing on the opportunity to decentralize markets and disrupt middlemen. In the latest in our Profiles in Innovation series, we shift the focus from theory to practice, examining seven real-world applications of blockchain, such as enhancing trust in the Sharing Economy, building a distributed smart grid, lowering the cost of title insurance, and changing the face of finance across capital markets, trading and control. We identify, itemize, and quantify the players, dollars and risks for blockchain to reach its full potential.
SECTOR BRIEFING
WORLD ECONOMIC FORUM
COMMITTED TO IMPROVING THE STATE OF THE WORLD
The future of financial infrastructure
An ambitious look at how blockchain can reshape financial services
An Industry Project of the Financial Services Community | Prepared in collaboration with Deloitte
Part of the Future of Financial Services Series • August 2016

Need for Blockchain and Smart Contracts

More real world examples…

Use cases

Need fo

More real world examples…

- Global trade finance

Commonwealth Bank, Wells Fargo Test Blockchain for Cotton Trade

Stan Higgins (@mpmcsweeney) | Published on October 24, 2016 at 16:06 BST

NEWS

281 | 23

Commonwealth Bank and Wells Fargo have announced they are testing blockchain for use in trade finance, focusing on the global cotton market.

Working alongside blockchain startup Skuchain and Australian cotton trading firm Brighann Cotton, the two banks facilitated a transaction between a cotton buyer and seller. In statements, Commonwealth said that the test enabled all parties involved "to track a shipment in real time" using a distributed ledger.

Michael Eidel, executive general manager for Commonwealth Cash-flow and Transaction Services office, said in a statement:

> *"The interplay between blockchain, smart contracts and the Internet of Things is a significant development towards revolutionising trade transactions that could*

Need for

More real world examples…

- **Global trade finance**
- **Supply chains**

Walmart Wants to Apply Blockchain to Other Products Beyond Pork

Michael del Castillo (@DelRayMan) | Published on October 25, 2016 at 14:23 BST

NEWS

300 | 92

Trying to make pork products in China safer was just the first step of Walmart's global plans for blockchain.

The pilot unveiled last week uses technology from the Hyperledger project to track pork shipping information, including farm origination details, batch numbers and storage temperatures on a secure blockchain.

Over the months ahead, the retail giant wants to expand on that work. Walmart vice president of global food safety Frank Yiannas told CoinDesk that, in anticipation of a successful pilot launch, the company is already looking to the future for other applications.

Yiannas told CoinDesk:

> *"We will immediately work to identify additional food products where we might*

Need for

More real world examples…

- Global trade finance
- Supply chains
- Post-trading process

Russian, Chinese Central Securities Depositories Partner on Blockchain

Stan Higgins (@mpmcsweeney) | Published on October 25, 2016 at 15:07 BST

NEWS

270 | 90

The central securities depositories (CSDs) in Russia and China have signed a memorandum of understanding that sets the stage for the two institutions to begin partnering on post-trade blockchain applications.

Announced today, the deal will see Russia's National Settlement Depository (NSD) and China's Securities Depository and Clearing Corporation Limited (CSDC) "exchange experience and information" on a range of issues, according to an announcement from NSD. The two institutions will also collaborate on experimenting with fintech, which will include trials involving blockchain.

According to NSD executive board chairman Eddie Astanin, the cooperation on fintech and blockchain is one of the primary aspects of the deal.

Astanin said in a statement:

Need for

More real world examples...

- Global trade finance
- Supply chains
- Post-trading process
- Fintech

Singapore Central Bank Inks Blockchain Deals With India, South Korea

Stan Higgins (@mpmcsweeney) | Published on October 26, 2016 at 14:43 BST

NEWS

It's been a busy week on the blockchain front for Singapore's central bank.

On 22nd October, the Monetary Authority of Singapore (MAS) signed an agreement with the government of Andhra Pradesh, a coastal state in India, to collaborate on blockchain development projects.

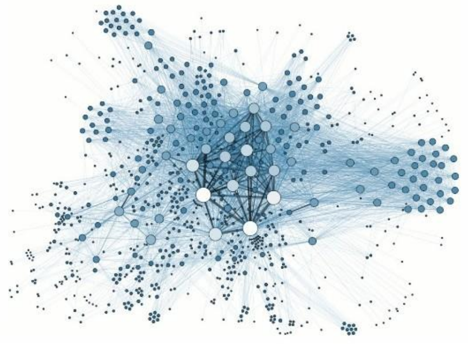
According to statements, the partnership will include a specific focus on digital payments, as well as the creation of educational resources related to the tech. MAS and the Andhra Pradesh government committed to broader discussions over regulation focused on "innovations in financial services".

The goal, the two institutions said, is to spur the development of a new fintech startup hub in the Indian state.

J. A. Chowdary, a technology advisor to the Andhra Pradesh government, said in a statement:

4 Key Concepts of Blockchain

Distributed shared ledger



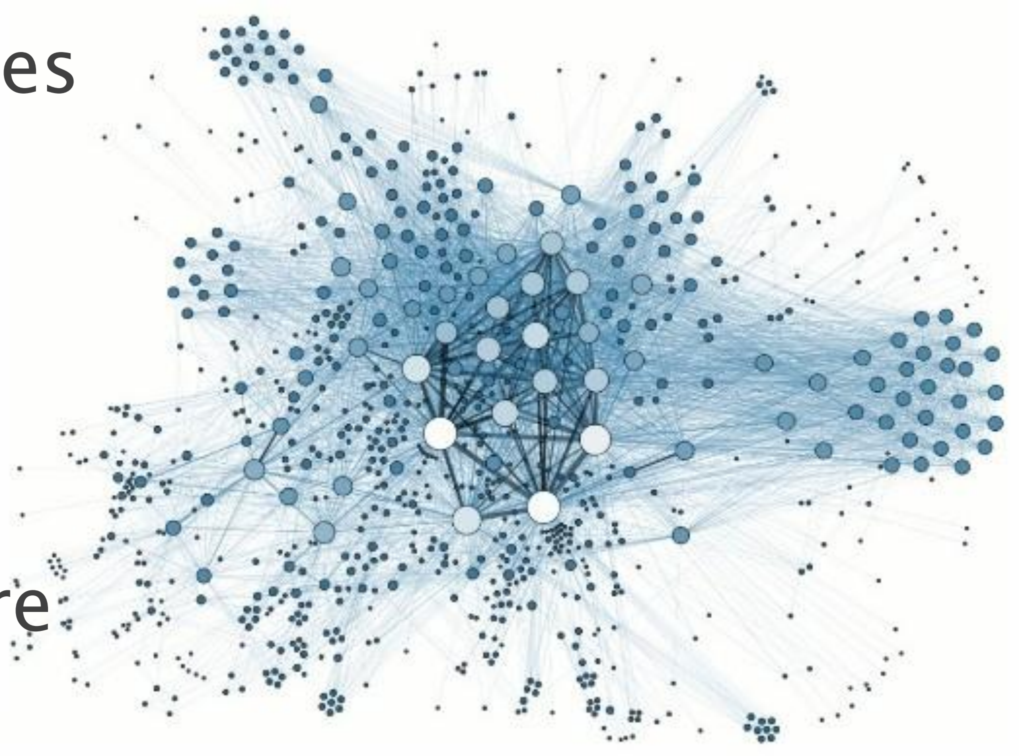
Cryptography

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.

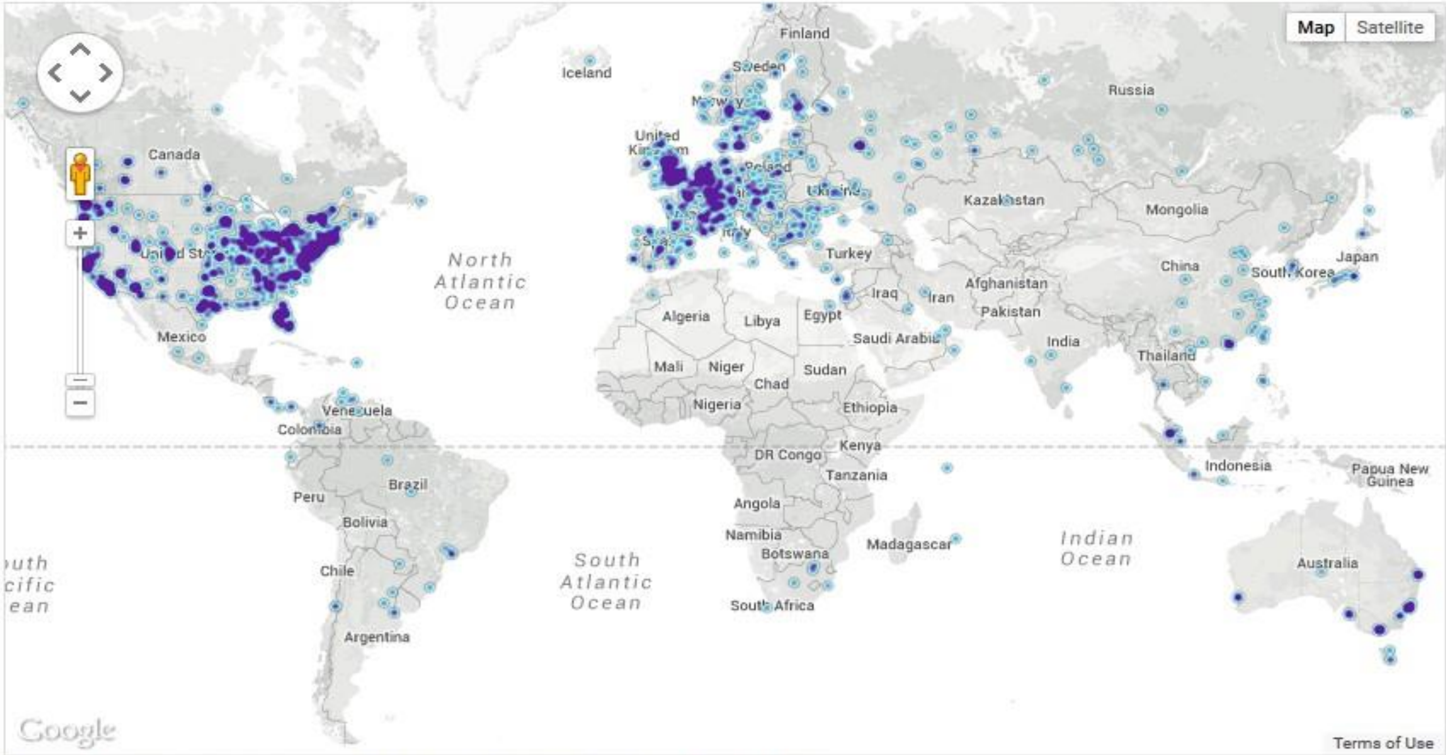
5987 nodes

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) »



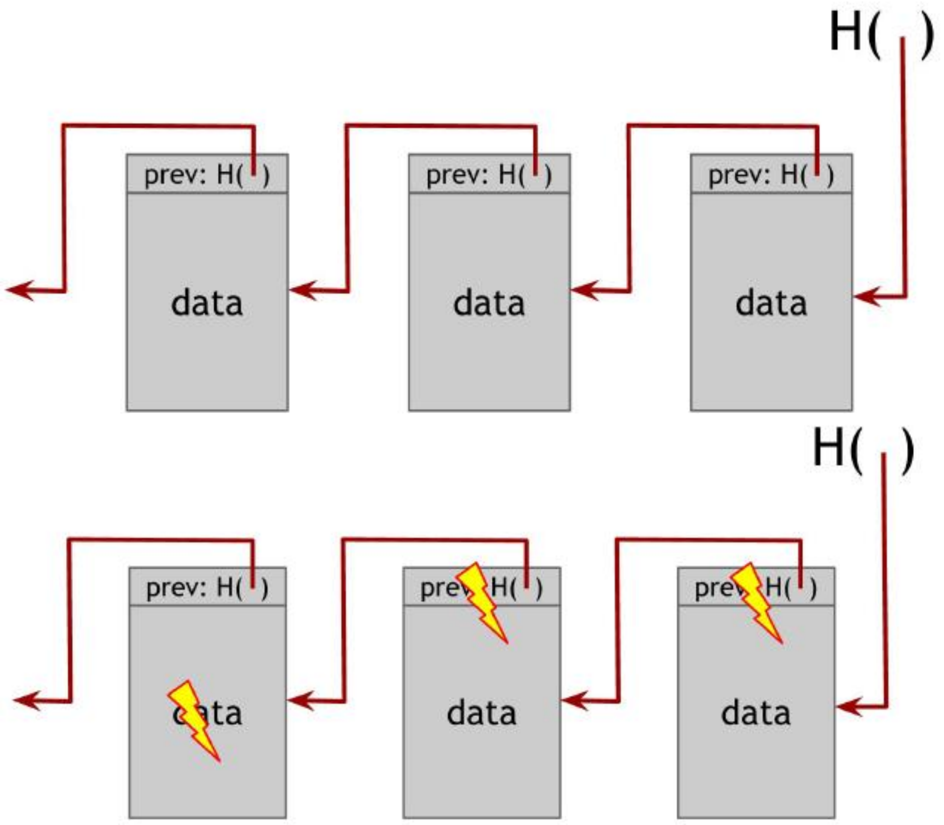
Map shows concentration of reachable Bitcoin nodes found in countries around the world.

JOIN THE NETWORK

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

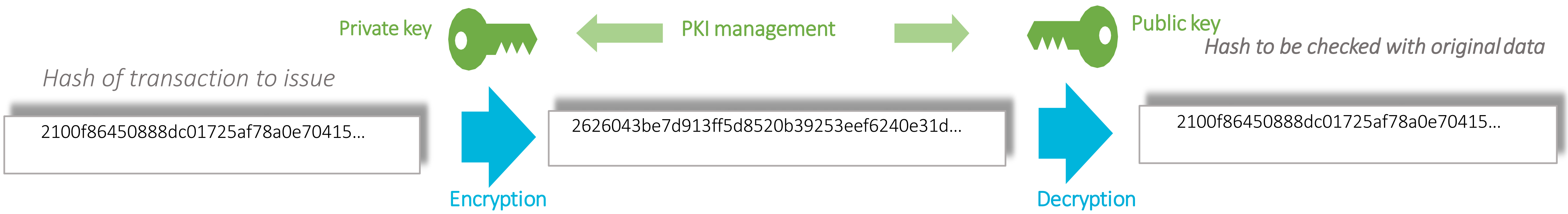
4 Key Concepts of Blockchain: 2.Cryptographic (1/2)

Tamper-proof log blocks using hash pointer



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

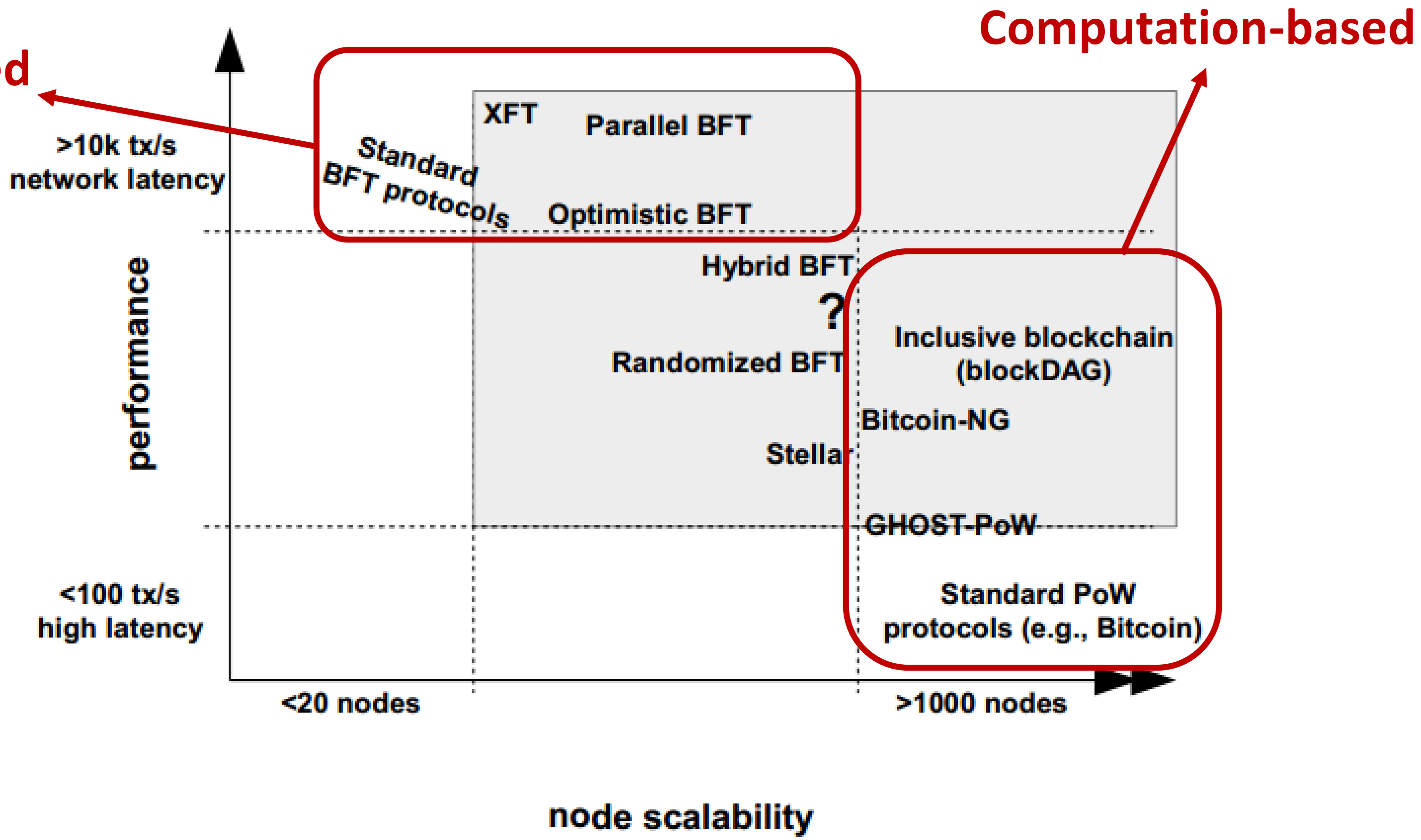
Asymmetric cryptography digital signature system



4 Key Concepts of Blockchain: 3.Consensus

Consensus

- No single point failure
- Byzantine fault tolerance

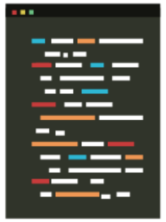


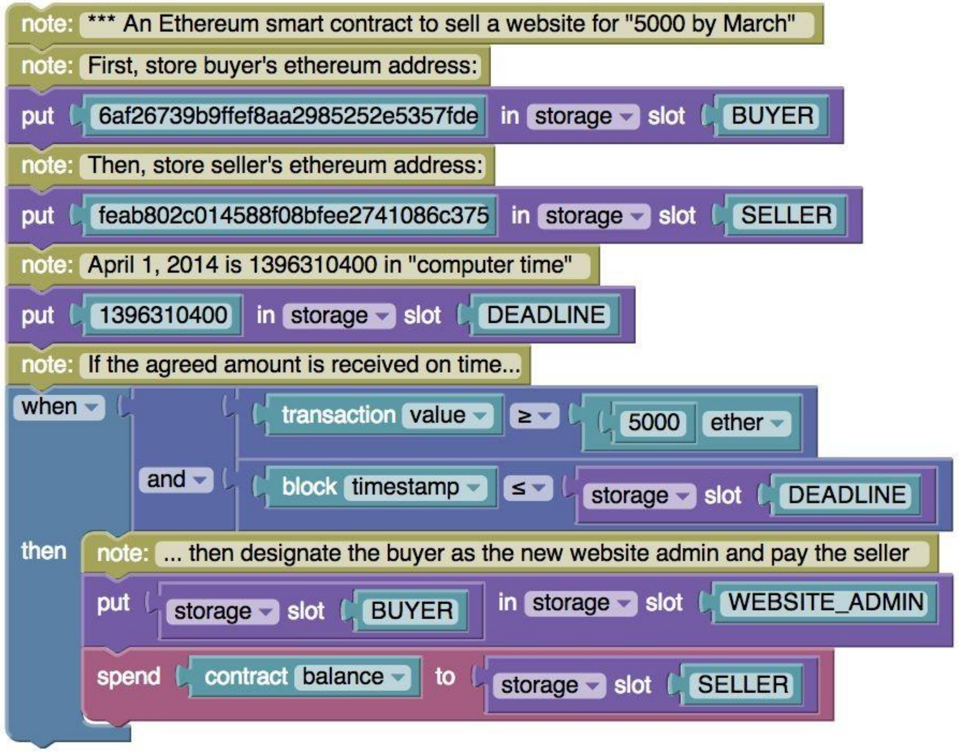
Cite: Vukolić, Marko. "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication."

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





Values of blockchain

Reduction of costs and complexity

Shared trusted transactions

Reduction of errors

Resilience

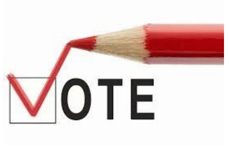
Secure

Auditability

Potential of blockchain

Financial Services
- Payments
- Securities registration & processing
- Lending

Property
- Real estate
- Intellectual property
- Cars



Governmental services
- Voting
- Registrations (passports, driving license)
- Permits

Identification & Security
- Party/device registration
- Authentication
- Access control

Trade
- Document exchange
- Asset exchange
- Escrow services
- Trade agreements

Internet of Things (IoT)
- Autonomous devices, such as
    - Cars
    - Drones
    - Robots

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

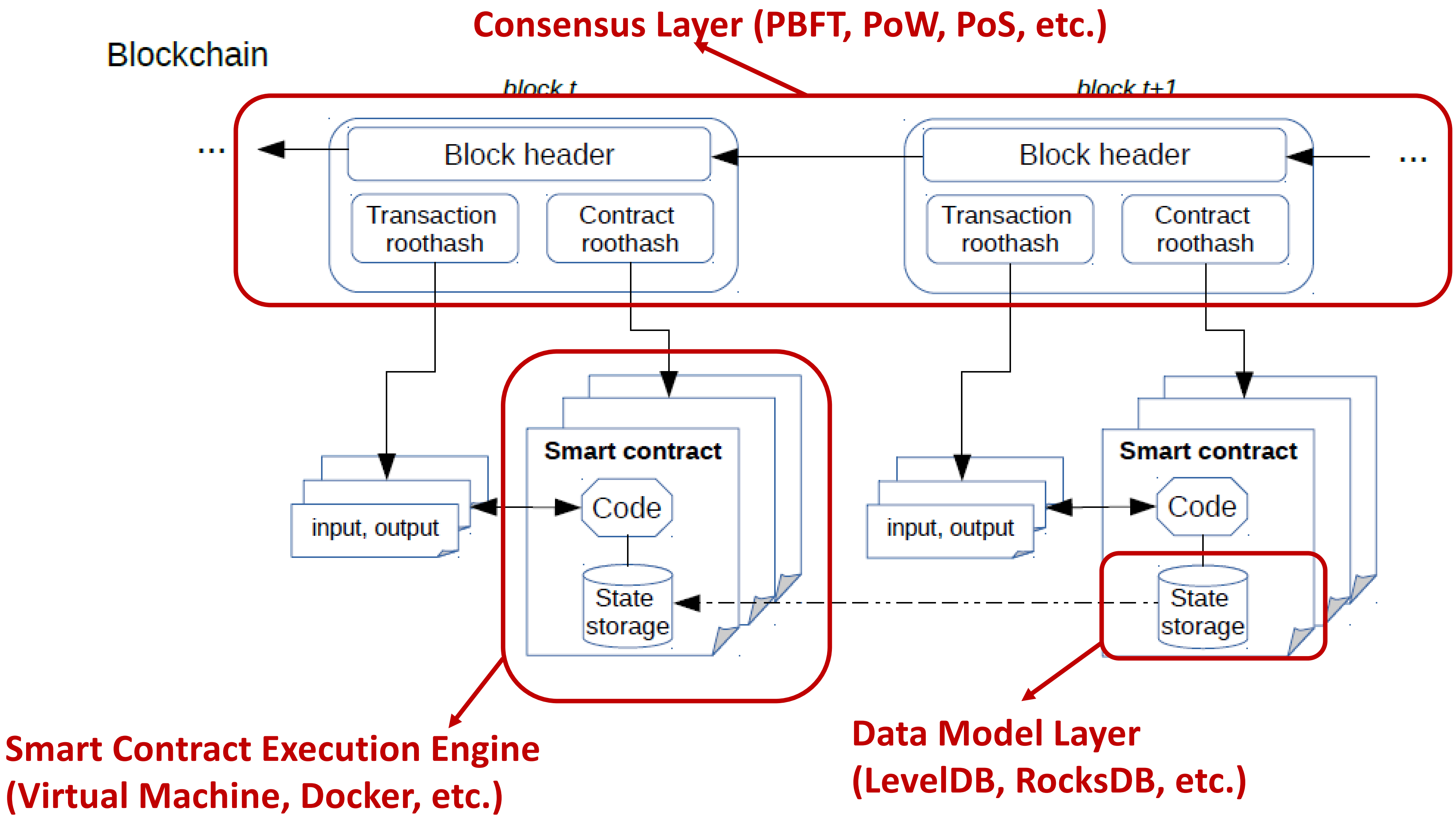
Challenges

- 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.

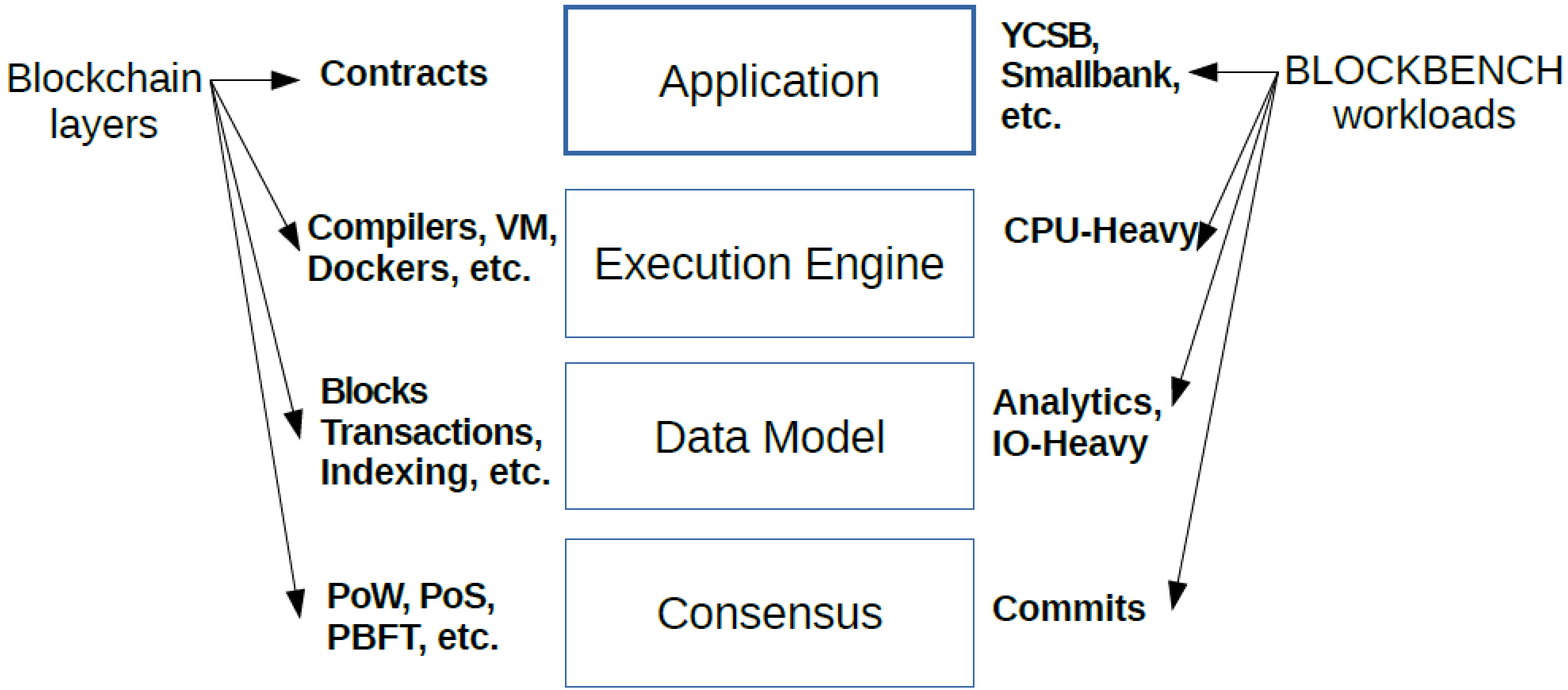
Challenges

- Three main challenges

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

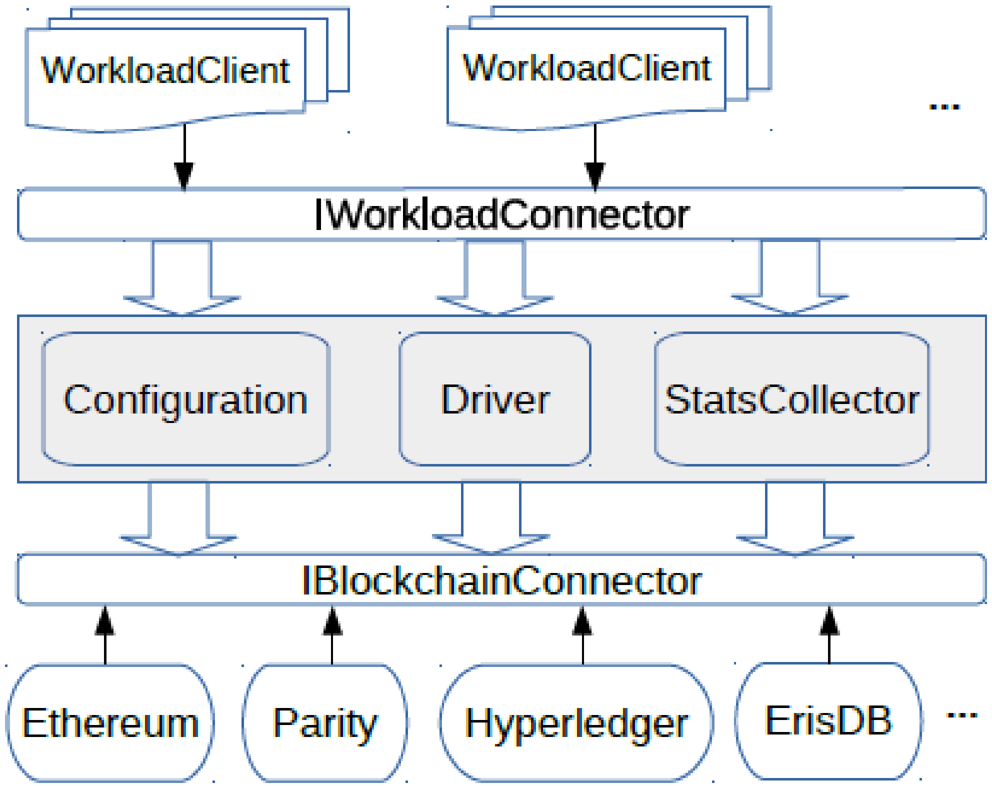
Approach: We treat blockchain as a key-value storage coupled with an engine which can realize both transactional and analytical functionality via smart contracts.
We design and run both transaction and analytics workloads in our benchmark framework.

Framework Design



Framework Implementation

- New workloads are added by implementing `IWorkloadConnector` interface.
- New blockchain backends are added by implementing `IBlockchainConnector`



Five Key Metrics

- Throughput
  - measured as the number of successful transaction per second

- Latency
  - measured as the response time per transaction

- Scalability
  - measured as how the throughput and latency change when increasing number of nodes and number of concurrent workloads.

- Fault tolerance
  - measured as how the throughput and latency change during node failure, such as fail-stop, network delay and arbitrary message errors.

- Security
  - simulate network partition attacks, measure as stale block rates

Outline

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

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 (YCSB, Smallbank, EtherId, Doubler, WavesPresale)

Micro-Benchmarks (VersionKVStore, IOHeavy, CPUHeavy, DoNothing)

Storage-oriented (YCSB, Smallbank, EtherId)

Application-oriented (Doubler, WavesPresale)

Data model (VersionKVStore, IOHeavy)

→ Execution engine (CPUHeavy)

→ Consensus layer (DoNothing)

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

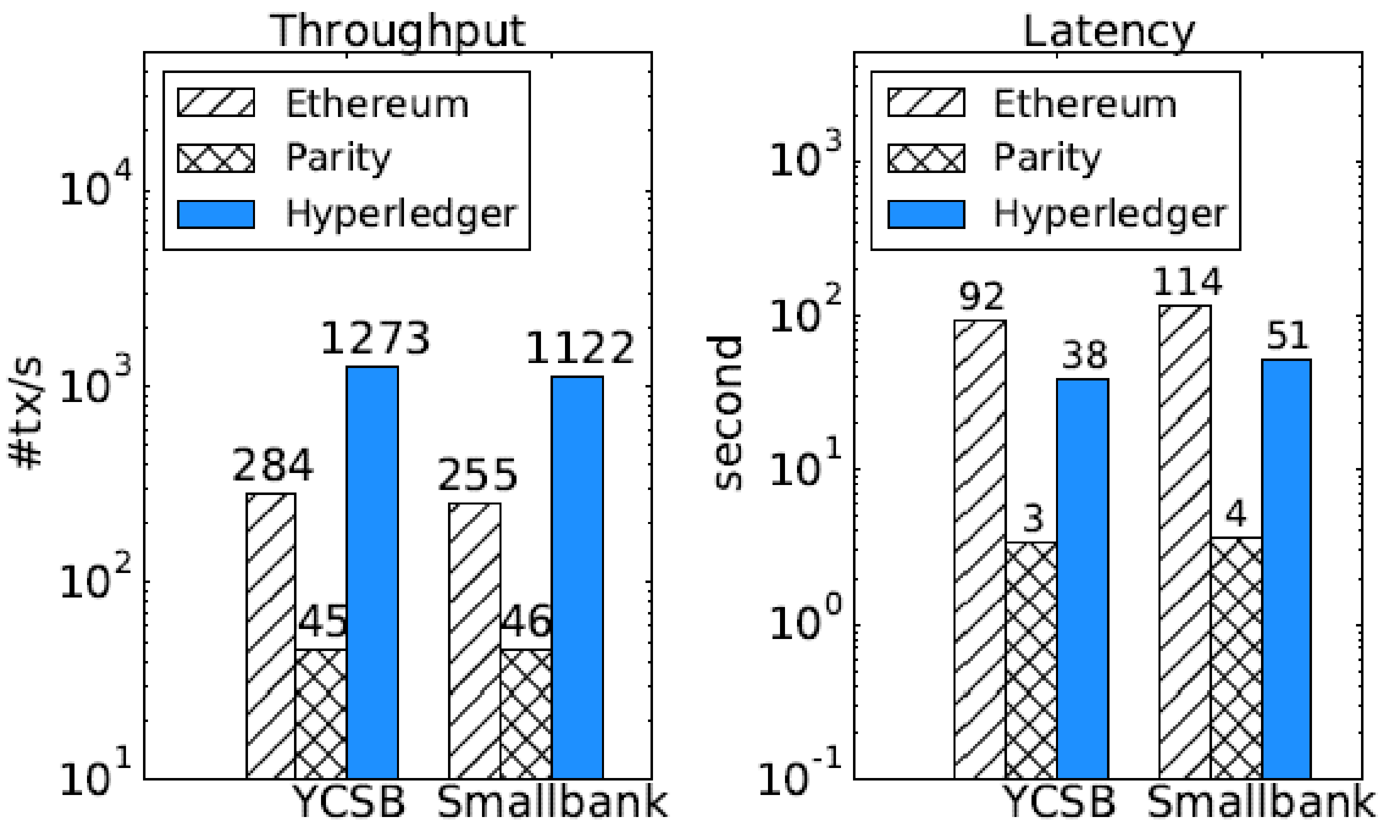


Figure: Throughput and latency of 3 systems over YCSB and SmallBank benchmark

Throughput & Latency

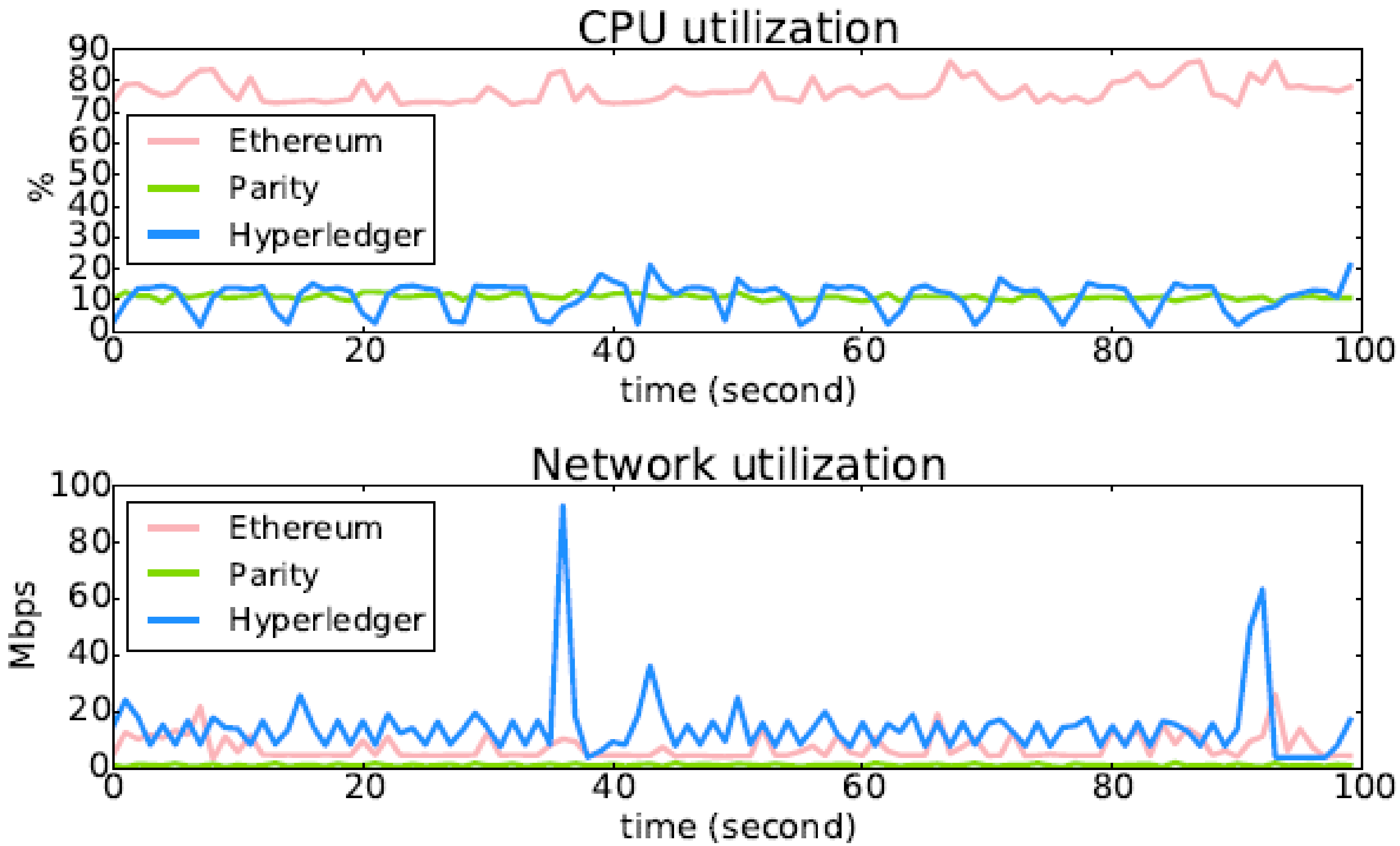


Figure: CPU & network resource utilization of 3 systems over YCSB benchmark

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

Simply increasing block size does not help:
larger block size means lower block generation rate

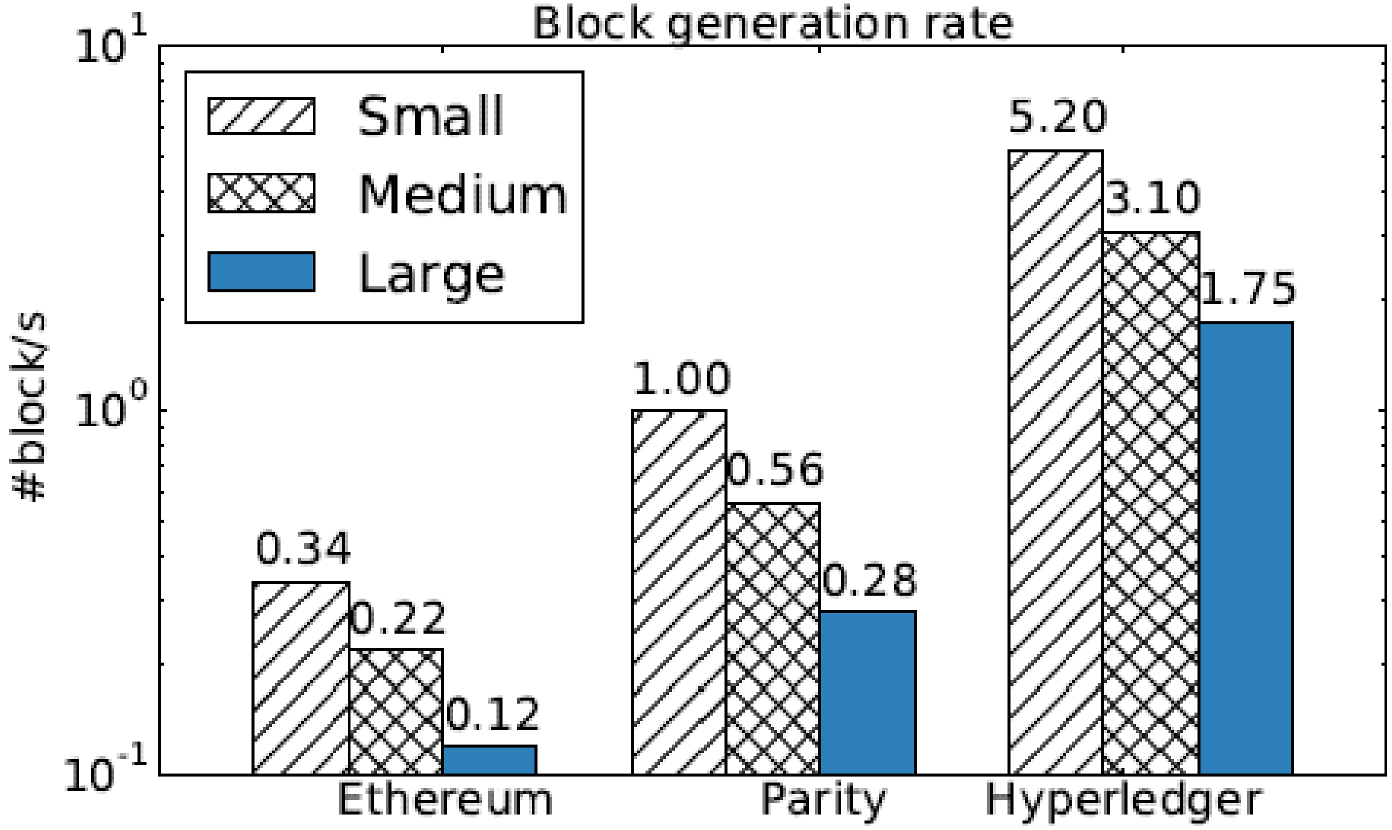


Figure: Block generation rate with varying block size

Throughput & Latency

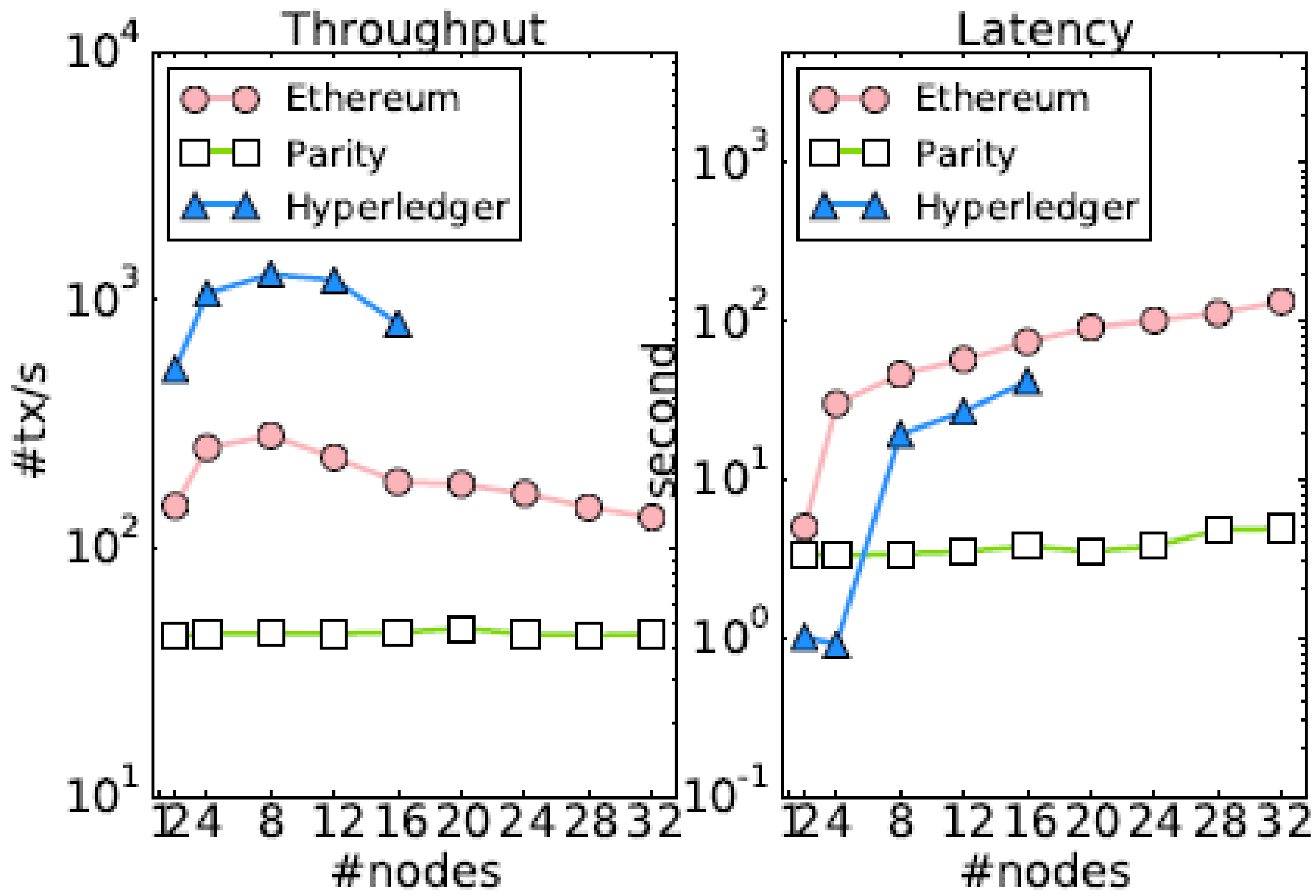


Figure: Performance scalability (with the same number of clients and servers).

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.

Throughput & Latency

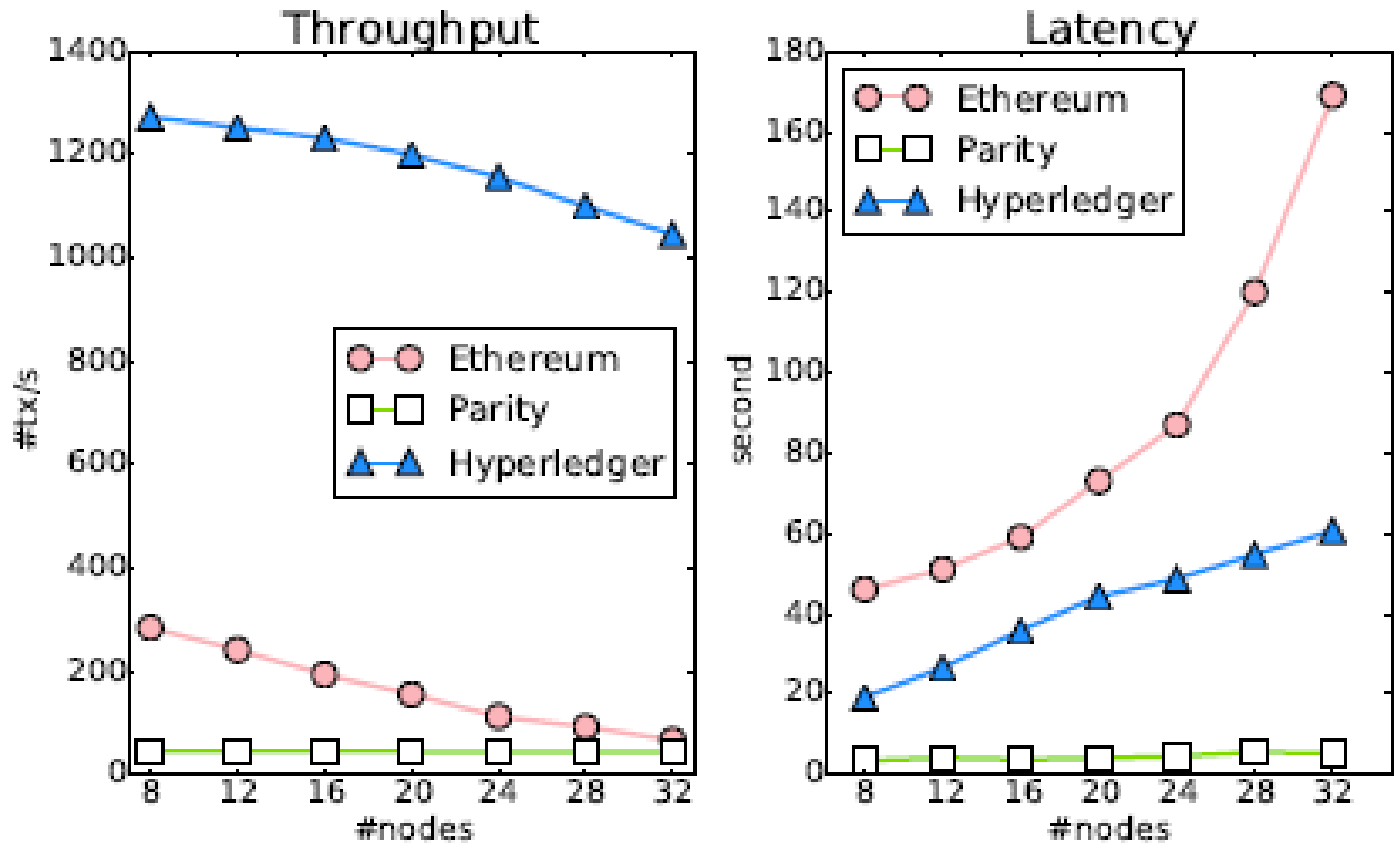


Figure: Performance scalability (with 8 clients).

Scalability

Observations

- The performance becomes worse as there are more servers, meaning that the systems incur some network overheads.

- Hyperledger is communication bound, having more servers means more messages being exchanged and higher overheads.

- Ethereum consumes a modest amount of network resources for propagating transactions and blocks to other nodes.

Fault-tolerance & Security

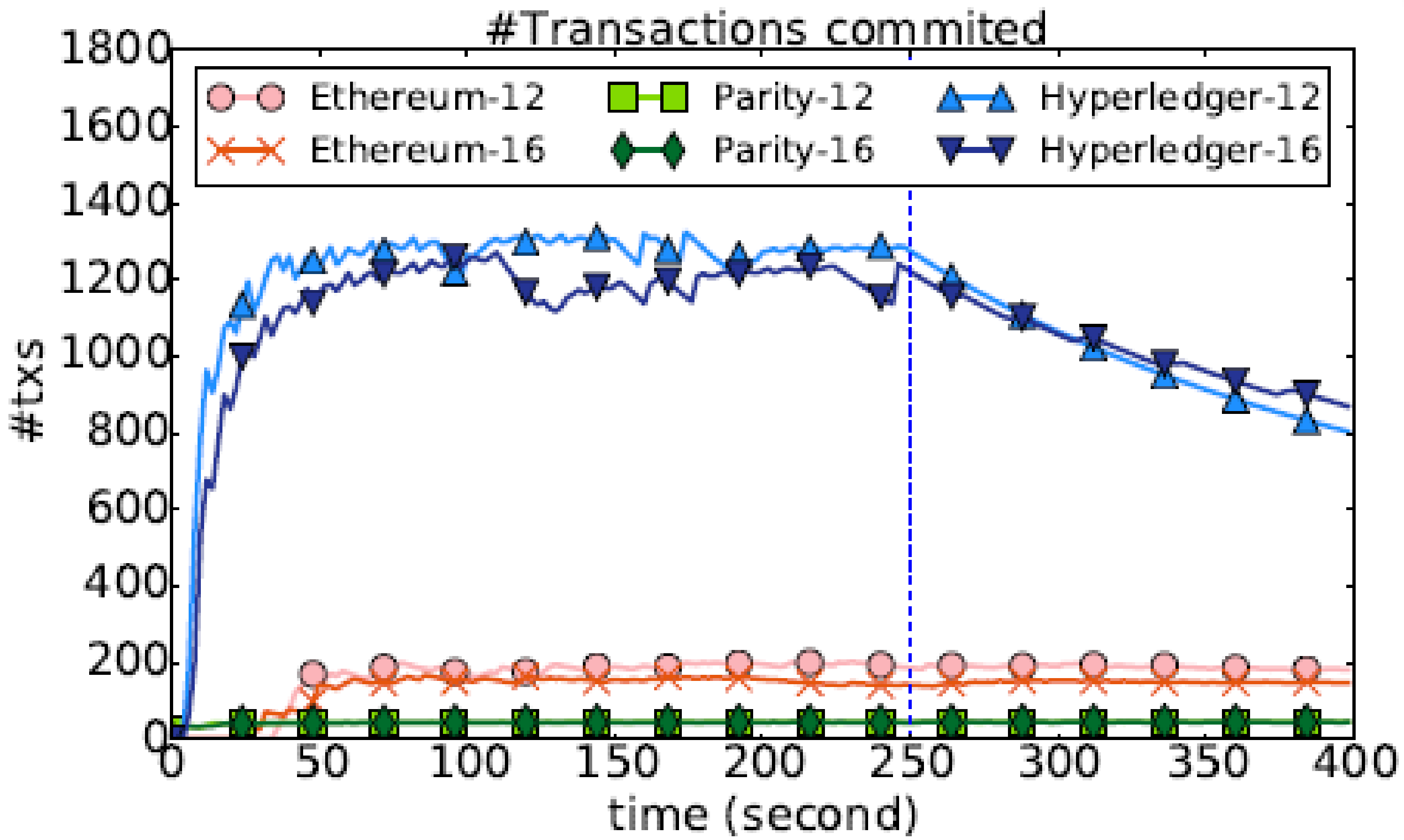


Figure: Failing 4 nodes at 250th second (fixed 8 clients) for 12 and 16 servers. X-12 and X-16 mean running 12 and 16 servers using blockchain X respectively.

Fault-tolerance & Security

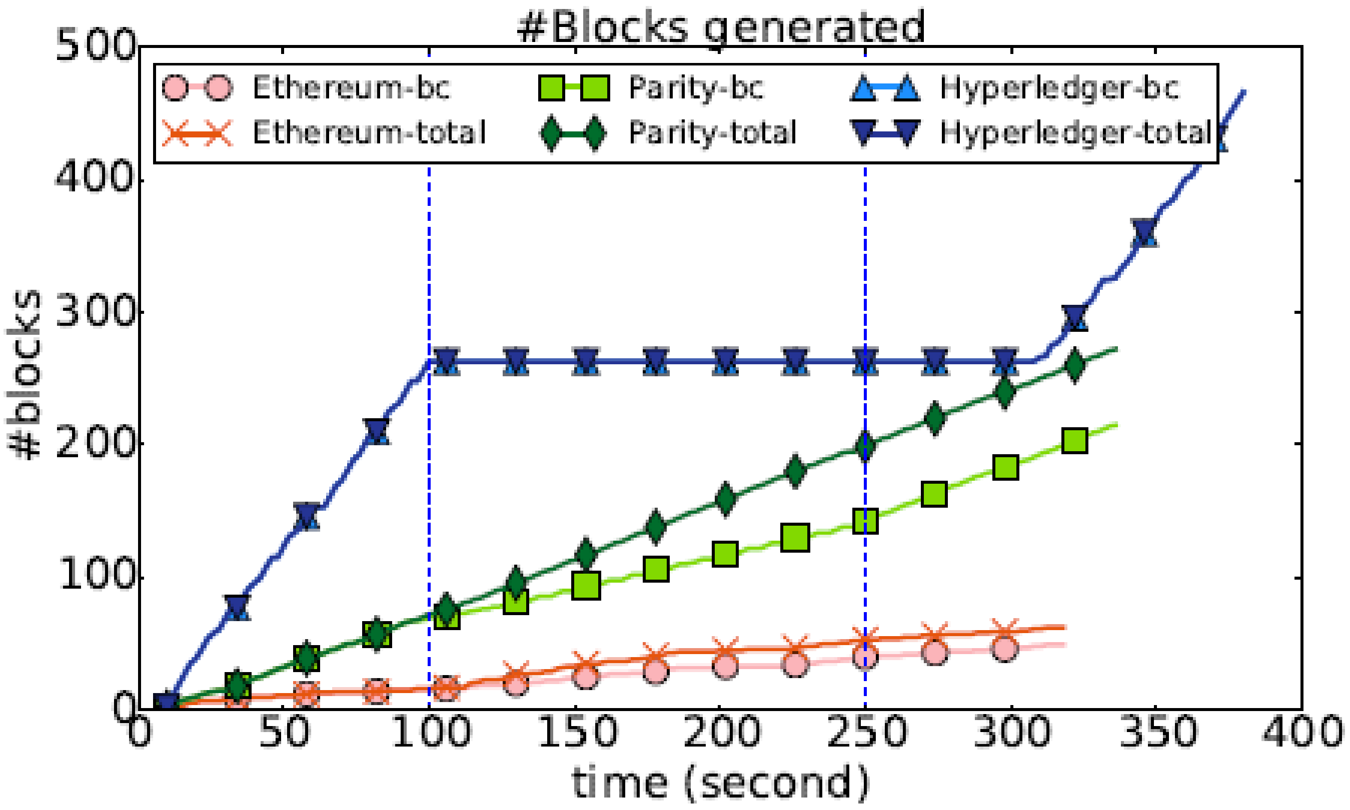


Figure: Blockchain forks caused by attacks that partitions the network in half at 100th second and lasts for 150th seconds. X-total means the total number of blocks generated in blockchain X, X-bc means the total number of blocks that reach consensus in blockchain X.

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

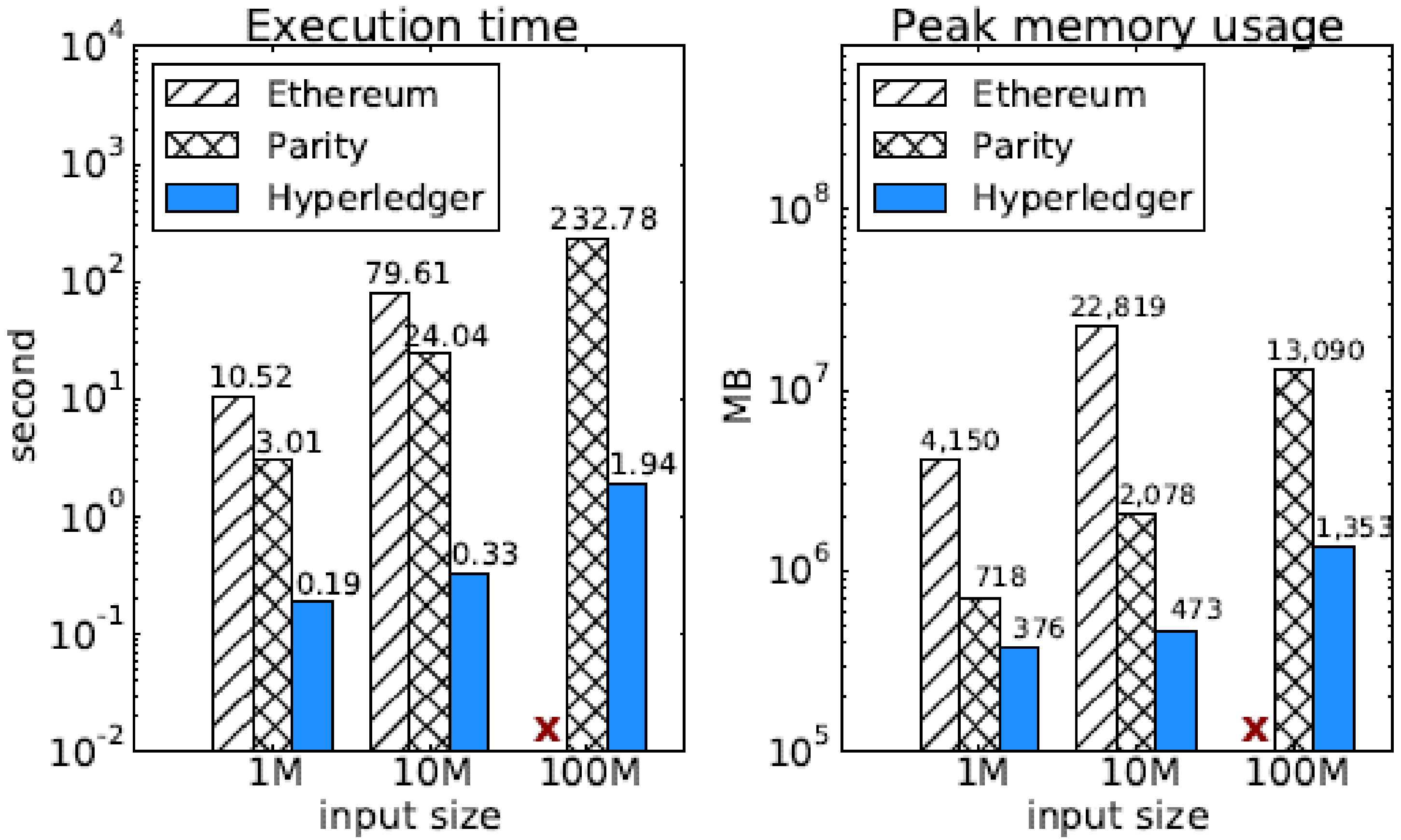


Figure: CPUHeavy workload, ‘X’ indicates Out-of-Memory error.

Execution Layer – CPUHeavy

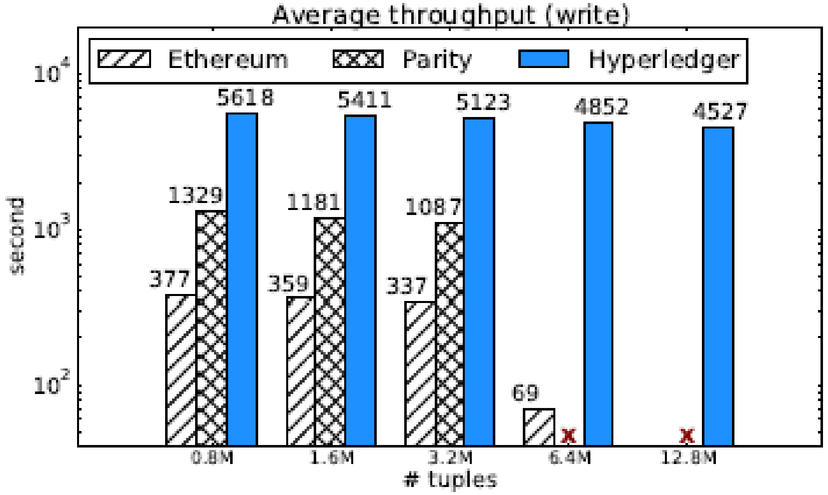
Observations

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

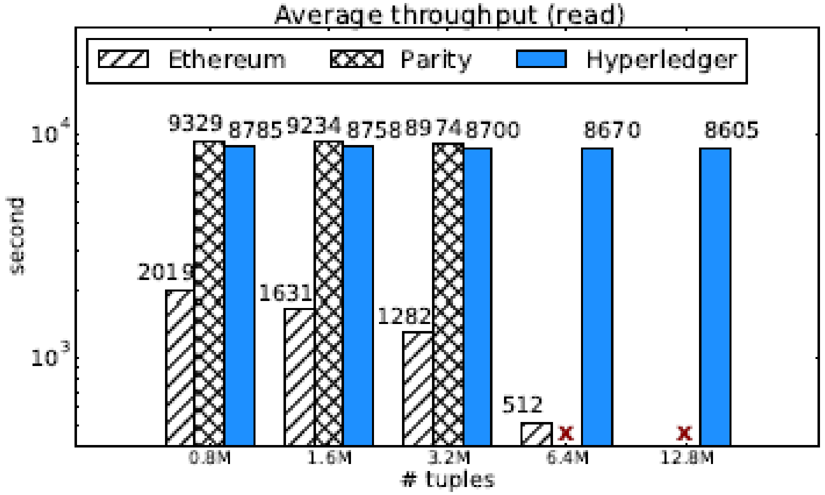
- Hyperledger's execution engine is more computation and memory efficient than EVM.

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

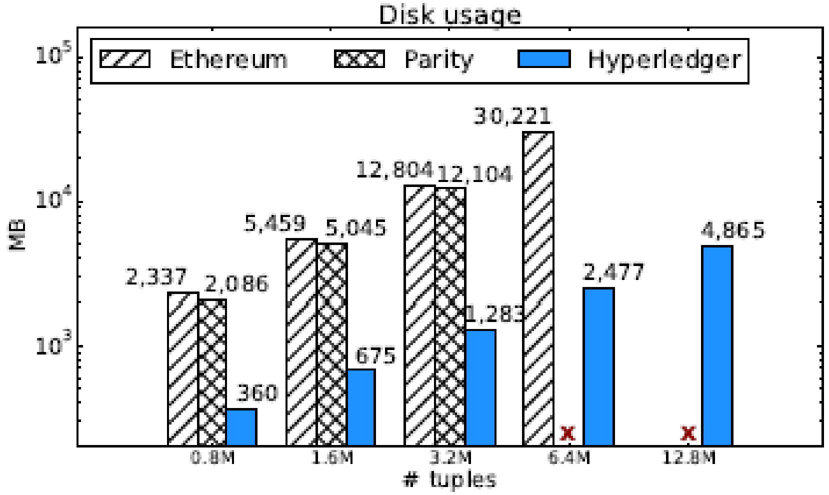
Data Model Layer - IOHeavy



(a) Write



(b) Read



(c) Disk usage

Figure: IOHeavy workload, `X' indicates Out-of-Memory error.

Data Model Layer - IOHeavy

Observations

- Ethereum and Parity use the same data model but make different design trade-offs. Parity cache the whole states in-memory so capped by memory size. Ethereum uses LRU eviction policy so can handle more states data but has less efficiency.

- Hyperledger provides lower-level data model which has less overhead.

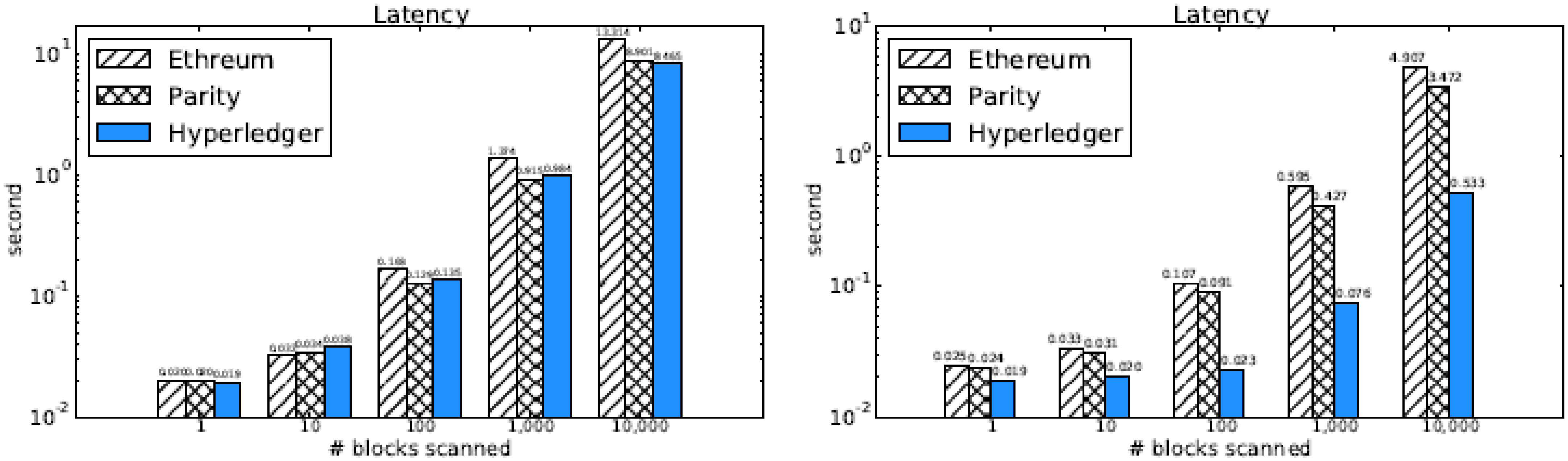
Data Model Layer – Analytics

This workload considers the performance of blockchain system in answering analytical queries about the historical data.

Q1: Compute the total transaction values committed between block i and block j.

Q2: Compute the largest transaction value involving a given state (account) between block i and block j.

Data Model Layer – Analytics



(a) Analytics workload (Q1)

(b) Analytics workload (Q2)

Figure: Analytics workloads.

Data Model Layer - Analytics

Observations

- Main bottleneck for query is RPC round-trip latency.

- It is important to provide customizable query API to push the computation to the server-side.

Consensus Layer – DoNothing

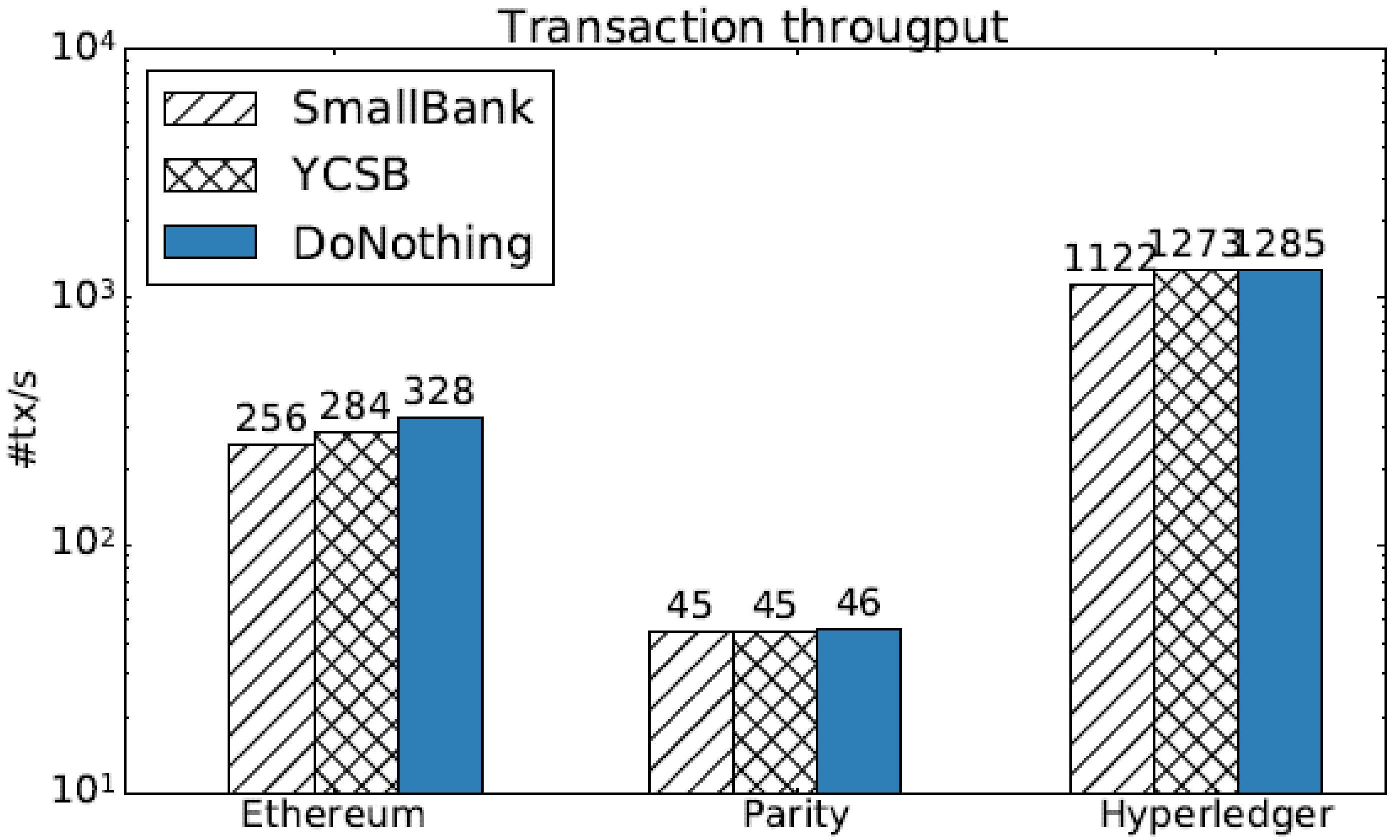


Figure: DoNothing workloads.

Consensus Layer - DoNothing

Observations

- Consensus layer contributes the most overhead in Ethereum and Hyperledger.

- For Ethereum 10% increases in throughput as compared to YCSB, which means that execution of the YCSB transaction accounts for the 10% overhead.

- No difference in YCSB, SmallBank and DoNothing for Parity. Performance bottleneck of Parity is the transaction signing.

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

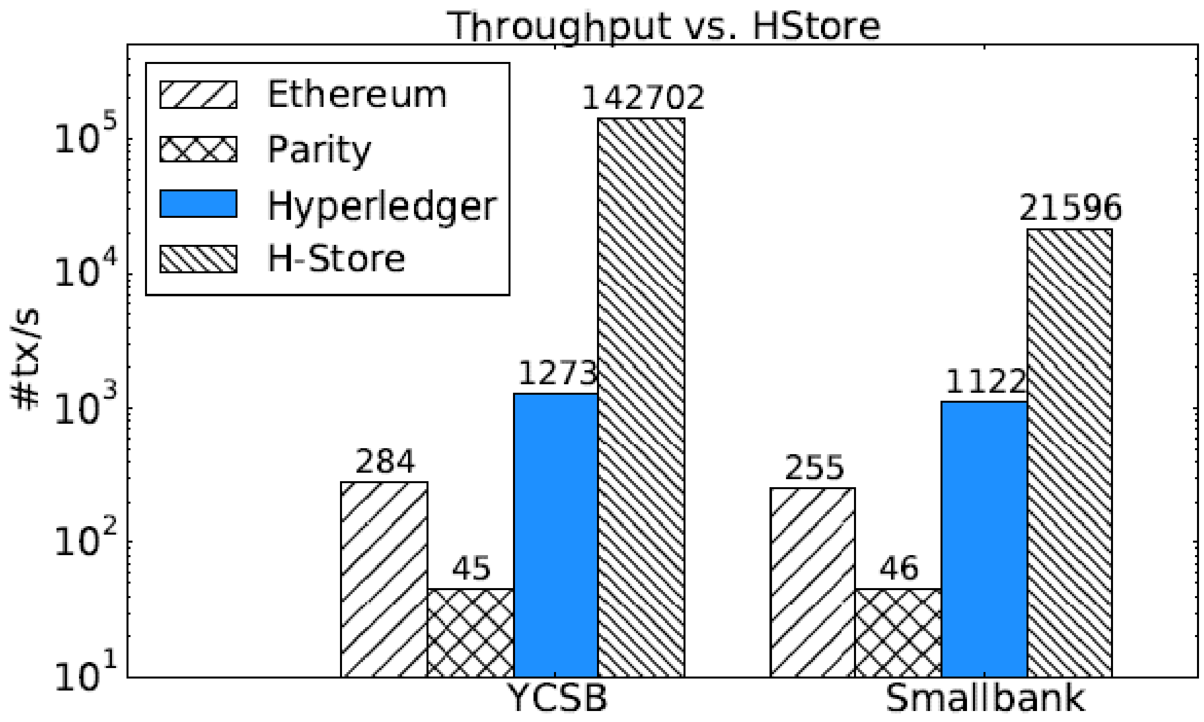


Figure: Performance of the three blockchain systems versus H-Store.

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.

Discussion

Bringing database designs into blockchain

- Sharding.

* Existing consistency protocols used in database systems do not work under Byzantine failure.

* Nevertheless, designs of sharding database systems can offer insights into realizing a more scalable sharding protocol for blockchain.

* The main challenge with sharding is to ensure consistency among multiple shards.

Discussion

Bringing database designs into blockchain

- Support declarative language.

* Having a set of high-level operations that can be composed in a declarative manner makes it easy to define complex smart contracts.

* Declarative language also opens up opportunities for low-level optimizations that speed up contract execution.

Outline

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

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!