BLOCKBENCH: A Framework for Analyzing Private Blockchains

Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, Kian-lee Tan
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
- Cryptography (SHA-256, PKI)
- Consensus model
- Smart contracts
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.
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.
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., Ethereum 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
Need for Blockchain and Smart Contracts

More real world examples…

The future of financial infrastructure
An ambitious look at how blockchain can reshape financial services

Goldman Sachs does and seeks to do business with all of the firms whose stocks are covered in this research report. As a result, investors should be aware that the firm may have a financial interest in the companies being covered. For Reg A2 certification and Appendix, go to www.gs.com/research/hedge.html registered for qualified as research analysts with FINRA

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
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:
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

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 Blockchain and Smart Contracts

More real world examples...

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

Russian, Chinese Central Securities Depositories Partner on Blockchain

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

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 Blockchain and Smart Contracts

More real world examples...

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

News Article:

Singapore Central Bank Inks Blockchain Deals With India, South Korea

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

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
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

Hash of transaction to issue

2100f8645088dc01725af78a0e70415...

Encryption

PKI management

2626043be7d913ff5d8520b39253eef6240e31d...

Decryption

Public key

Hash to be checked with original data

Hash of transaction to issue

2100f8645088dc01725af78a0e70415...
4 Key Concepts of Blockchain: 3. Consensus

- No single point failure
- Byzantine fault tolerance

4 Key Concepts of Blockchain: 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

<table>
<thead>
<tr>
<th>Reduction of costs and complexity</th>
<th>Shared trusted transactions</th>
<th>Reduction of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="COST" /></td>
<td><img src="image2.png" alt="TRUST" /></td>
<td><img src="image3.png" alt="errors" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resilience</th>
<th>Secure</th>
<th>Auditability</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4.png" alt="Resilience" /></td>
<td><img src="image5.png" alt="Secure" /></td>
<td><img src="image6.png" alt="Auditability" /></td>
</tr>
</tbody>
</table>
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.
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 header

Transaction root hash

Contract root hash

Block header

Transaction root hash

Contract root hash

Smart contract

Code

State storage

Smart contract

Code

State storage

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 functionality, namely Hyperledger Fabric, Ethereum and Parity, and the framework is general to support future platforms.
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 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 transactions per second

- **Latency**
  - measured as the response time per transaction

- **Scalability**
  - measured as how the throughput and latency change when increasing the 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
# Workloads

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

**Macro-Benchmarks**

**Micro-Benchmarks**

**Storage-oriented**

**Application-oriented**

**Data model**

**Execution engine**

**Consensus layer**
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.
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**.
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.
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
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 suggests that there are non-negligible costs in the execution layer of blockchains.
Simply increasing block size does not help: larger block size means lower block generation rate.
Figure: Performance scalability (with the same number of clients and servers).
Scalability

Observations

• Parity's performance remains constant as the network size and offered 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.
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.
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.
Figure: CPUHeavy workload, ‘X’ indicates Out-of-Memory error.
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

Figure: IOHeavy workload, `X' indicates Out-of-Memory error.
Ethereum and Parity use the same data model but make different design trade-offs. Parity caches 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.
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.
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:
Thanks!