BLOCKBENCH: A Framework for Analyzing Private Blockchains

Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi,
What is The Blockchain?

[A hash pointer linked list of blocks]

• An append-only sequential data structure
• New blocks can only be appended at the end of the chain
• To change a block in the middle of the chain, all subsequent blocks need to be changed
Blockchain:
A Distributed Transaction Ledger

• Every block contains multiple transactions
• Massively duplicated across network nodes
• Shared with a P2P file transfer protocol
• Updated by nodes, known as miners, appending new blocks of transactions
HOW A BITCOIN IS MINTED
The birth of a new Bitcoin begins when a number of existing Bitcoins are used in transactions.

1. Alice sends Bob a unique Bitcoin from her secure digital wallet.

2. Software bundles this transaction, along with others, into a block, and transmits this block to a network of computers for verification.

3. Computers on this network race to verify the block of transactions.

4. Once it is proven valid, the block is added to the entire shared ledger of all Bitcoin transactions. This shared ledger is called the blockchain.

5. The updated blockchain is then distributed to the entire network and used for future verification.

6. The “miners” whose computers are first to verify transactions and maintain the blockchain win a chunk of brand new Bitcoins as their reward.

Block 51
- Proof of work: 0000009857vvv
- Previous block: 0000004327grza1
- Transactions: lK54f1xv, 0934w1d, vc4232v32

Block 52
- Proof of work: 0000009857vvv
- Previous block: 0000009857vvv
- Transactions: d55g31bm, 001hi009

Block 53
- Proof of work: 0000009857vvv
- Previous block: 0000009857vvv
- Transactions: 94xcv14, abb7bxq, 34aui86a
malicious node(miner)
A smart contract, also known as a crypto contract, is a computer program that directly controls the transfer of digital currencies or assets between parties under certain conditions.
Public vs Private blockchains

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
blockchain-based distributed computing platform featuring smart contract (scripting) functionality.

**Ethereum in one slide**

- Account: controlled by key (like in Bitcoin) or by code (smart contract)
- Nodes store state (balances, code, data), execute code, extend blockchain (PoW)
- Developers write contracts in Solidity, compile to bytecode, deploy to blockchain
- Users interact with contracts via transactions (e.g., send ether, perform computation)
**Parity** Technologies' primary **blockchain** solution which makes it possible to run a variety of Ethereum-based chains and interact with them. Using Rust language and PoA consensus. Can be public and private.

**Hyperledger Fabric** is a business blockchain framework hosted by the Linux Foundation intended as a foundation for developing blockchain applications or solutions with a modular architecture. **Hyperledger Fabric** allows components such as consensus and membership services to be plug-and-play. Using node.js, Java languages and PBFT consensus. Only private blockchain
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
There are many different choices of platforms, but not all of them have reached a mature design, implementation and an established user base.

They designed BlockBench based on the three most mature platforms which support smart-contract functionality, namely Hyperledger Fabric, Ethereum and Parity, and the framework is general to support future platforms.
Abstraction layers in blockchain, and the corresponding workloads in Blockbench.
Framework Implementation

- New workloads are added by implementing `IWorkloadConnector` interface.
- New blockchain backends are added by implementing `IBlockchainConnector`
## Five Key Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput</td>
<td>measured as the number of successful transaction per second</td>
</tr>
<tr>
<td>Latency</td>
<td>measured as the response time per transaction</td>
</tr>
<tr>
<td>Scalability</td>
<td>measured as how the throughput and latency change when increasing number of</td>
</tr>
<tr>
<td></td>
<td>nodes and number of concurrent workloads.</td>
</tr>
<tr>
<td>Fault tolerance</td>
<td>measured as how the throughput and latency change during node failure, such</td>
</tr>
<tr>
<td></td>
<td>as fail-stop, network delay and arbitrary message errors.</td>
</tr>
<tr>
<td>Security</td>
<td>simulate network partition attacks, measure as stale block rates</td>
</tr>
</tbody>
</table>
## 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
- Storage-oriented
  - VersionKVStore
  - IOHeavy

### Micro-Benchmarks
- Application-oriented
  - YCSB
  - Smallbank
  - EtherId
  - Doubler
  - WavesPresale

### 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 drive
- We collected comparison results in terms of our five metrics in macro benchmarks.
- We stress tested each individual layer using our micro benchmarks.
Figure: Throughput and latency of 3 systems over YCSB and SmallBank benchmarks
Throughput & Latency

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

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

Figure: Block generation rate with varying block size
Execution Layer - CPUHeavy

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.
Observations

- **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.
Consensus Layer - DoNothing

Transaction throughput

- SmallBank
- YCSB
- DoNothing

Figure: DoNothing workloads.
Discussion

Bringing database designs into blockchain
Huge performance gap between blockchains and transactional databases

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