HITACHI
Inspire the Next

Blockchain Benchmark Tools

2021/10/29 Hitachi, Ltd.

HITACHI
Inspire the Next

Contents

1. Blockchain Performance Challenge

HITACHI
Inspire the Next

- Different foundations for different decentralized blockchain systems.
- Public/Private, different Consensus Algorithm, etc.
  - Bitcoin, Ethereum / Hyperledger Fabric
  - PoW, PoS, PoC, PBFT
- Extremely difficult to measure and compare performances of different blockchains.
- Examples of transaction performances of different blockchains:
  - Bitcoin can theoretically reach a maximum of 7 tps (actual performance of 3~4 tps).
    https://www.comp.nus.edu.sg/~prateeks/papers/Bitcoin-scaling.pdf
  - Ethereum processes between 10~15 tps.
    https://ethereum.org/ja/developers/docs/dapps/#implications-of-dapp-development
  - Hyperledger Fabric can theoretically reach a maximum of 2200 tps but generally processes between 500~1200 tps in typical environment.
    https://ieeexplore.ieee.org/document/9169454
    https://www.ibm.com/blogs/blockchain/2019/01/answering-your-questions-on-hyperledger-fabric-performance-and-scale/

2-1. Blockchain Benchmark Tools

HITACHI
Inspire the Next

- Blockchain Benchmark Tools:
  - Ethereum/test-tools

    Accessible in Ethereum repositories, but has not been updated in over 5 years.

    https://github.com/ethereum/test-tools
  - Chainhammer

    Benchmark tool for Ethereum.

    https://github.com/drandreaskrueger/chainhammer
  - BCTMark

    Generic benchmark tool. Works with Ethereum and Hyperledger Fabric.

    https://gitlab.inria.fr/dsaingre/bctmark
  - Hyperledger Caliper

    Benchmark tool created by the Hyperledger community. Works with Ethereum and Hyperledger Fabric.

    https://github.com/hyperledger/caliper
- Keep Software Audience in Mind
  - Developers : Metrics should be carefully measured to help with system improvement.
  - Users : Transactions should be fast and secure.

2-2. Important Things to Keep in Mind in Performance Measurement

HITACHI
Inspire the Next

- Parameters:
  - Consensus Protocol
  - Geographical distribution of nodes
  - Hardware environment
  - Network model (using FW, etc)
  - Number of active nodes in test transactions
  - Reliable software components
  - Test tools
  - Types of data stores being used (CouchDB, H2, Postgres, etc)
  - Workload
- To easily compare performance tests between different platforms, parameters listed above must be clearly detailed as part of test results.

2-3. Hyperledger Performance and Scale Working Group

HITACHI
Inspire the Next

- For proper performance measurement we must first define some key terms.
- Key Metrics:
  - **Read Latency** : response time – submit time
  - **Read Throughput** : total read operations / total time

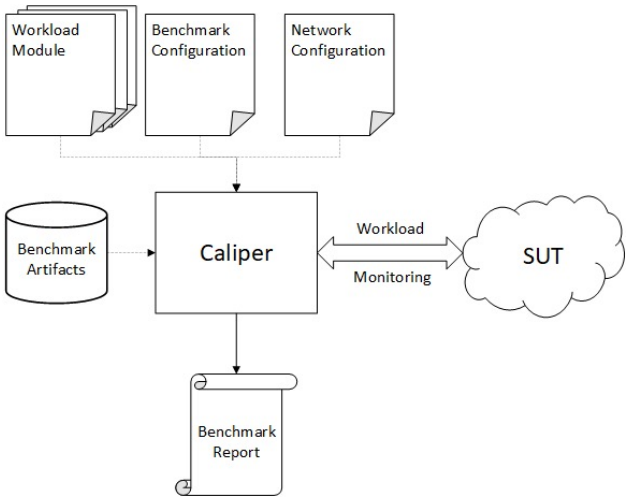
    Measures how many read operations are carried out in a defined time period.
  - **Transaction Latency** : (confirmation time @ threshold) – submit time

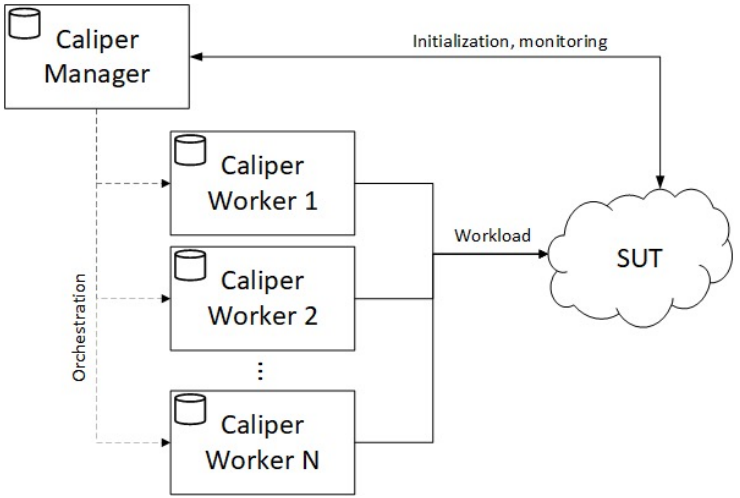
    Amount of time required for a transaction's effect to become usable throughout the network. Includes propagation time and consensus time, measured using all nodes. Network thresholds (ex. 50% or 90%) of network nodes can be set using transaction latency.
  - **Transaction Throughput** : total committed transactions / total time in seconds @ number of committed notes

    Rate of transactions committed across all network nodes, expressed in transactions per second (TPS). Does not include invalid transactions.

3-1. Hyperledger Caliper

- Blockchain Benchmark Tools developed in collaboration with the PSWG.
- Supports Hyperledger Fabric, Besu, Ethereum, and FISCO BCOS
- Architecture:
  - Monitoring the workload response generated in the SUT (system under test) as defined by Network Configuration (measured environment) and Workload Module (Tx output type).
  - Workload can be generated from multiple machines.





3-2. Hyperledger Caliper

HITACHI
Inspire the Next

- Sample configuration of Network and Workload

Network

```
    "caliper": {
        "blockchain": "ethereum"
    },
    "ethereum": {
        "url": "ws://localhost:8546",
        "contractDeployerAddress": "0xc0A8e4D21…",
        "contractDeployerAddressPassword": "password",
        "fromAddress": "0xc0A8e4D217eB85b812aeb…",
        "fromAddressPassword": "password",
        "transactionConfirmationBlocks": 2,
        "contracts": {
            "simple": {
                "path": "./src/ethereum/simple/simple.json",
                "estimateGas": true,
                "gas": {
                    "query": 100000,
                    "transfer": 70000
                }
            }
        }
    }
```

Workload

```
workers:
  type: local
  number: 1
rounds:
  - label: open
    txNumber: *number-of-accounts
    rateControl:
      type: fixed-rate
      opts:
        tps: 50
    workload:
      module: benchmarks/scenario/simple/open.js
      arguments: *simple-args
  - label: query
    txNumber: *number-of-accounts
    rateControl:
      type: fixed-rate
      opts:
        tps: 100
    workload:
      module: benchmarks/scenario/simple/query.js
      arguments: *simple-args
  - label: transfer
    txNumber: 50
    rateControl:
      type: fixed-rate
      opts:
        tps: 5
    workload:
      module: benchmarks/scenario/simple/transfer.js
      arguments:
        << : *simple-args
        money: 100
```

3-2. Results

HITACHI
Inspire the Next

- Performance measurement report of a predefined sample case
  - Ethereum/client-go
  - Executed with 1 node (m5.2xlarge)

Workload from previous page

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
open	1000	0	50.1	27.80	2.07	14.52	20.9
query	1000	0	100.1	0.00	0.00	0.00	100.1
transfer	50	0	5.1	6.93	2.12	4.52	3.4

Results with each parameter multiplied by a factor of 10

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
open	1000	0	501.5	25.85	2.49	12.91	36.1
query	1000	0	761.0	0.00	0.00	0.00	761.0
transfer	500	0	50.1	7.44	2.03	4.77	37.7

4. Conclusion

HITACHI
Inspire the Next

- It is very difficult to measure and compare performances of different blockchains.
- Key metrics must be defined in order to properly measure blockchain performance.
- Details of test results should be publicly shared.

- ex. Hyperledger Caliper
  - Network configuration
  - Workload configuration
  - System information
  - etc.

HITACHI

Inspire the Next