Crossing makes the Future.
まじわる力で みらいを創る

ETHTerakoya

Performance Assessment of Besu with Hyperledger Caliper (Try&Error)

NTT TechnoCross Corporation
Digital transformation department
First business unit
Kazuhiro Kanematsu

NTTテクノクロス

1. Company Profile

A branch of the NTT group.

The company became involved with blockchain in 2016 based on a research topic by FINTECH. Following this, each NTT research lab and group has participated in a variety of BC-related development projects and demonstration tests.

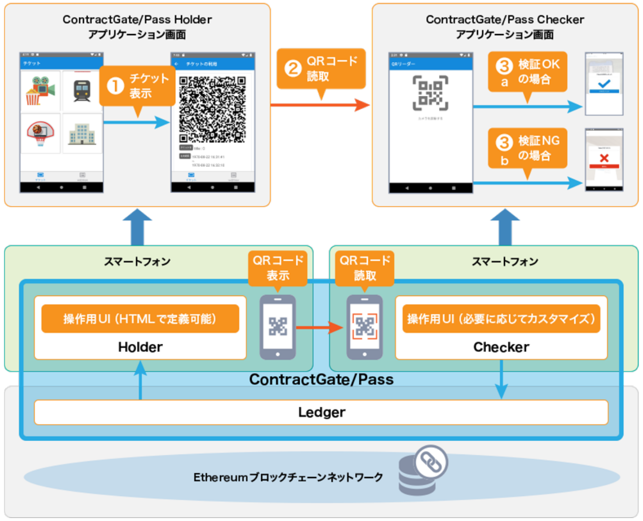
Additionally, the company provides in-house smartphone apps as well as the backend system services, and a monitoring tool that makes viewing BC data simpler.

Holder
An app for using blockchain passes

A smartphone app for passholders to gain access to their blockchain ticket Information such as their prices by displaying a QR code.

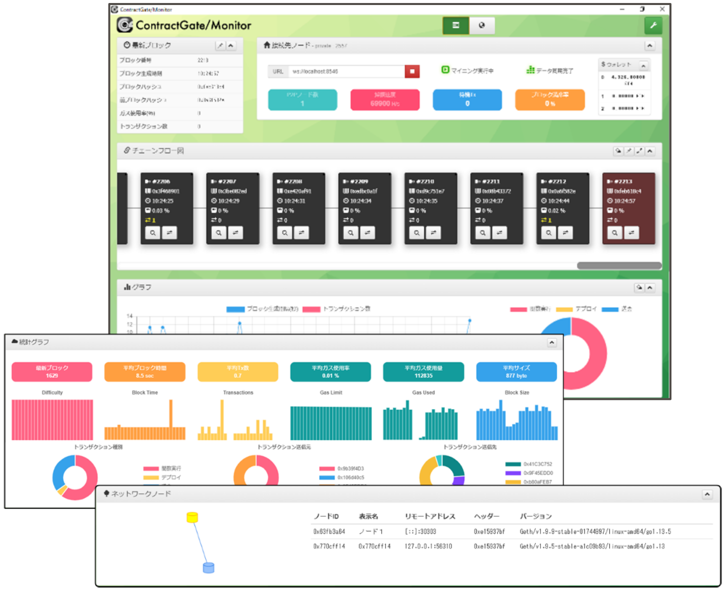
Checker
An app for displaying blockchain pass

With the QR code displayed on Holder, the app can be used to access and display blockchain data



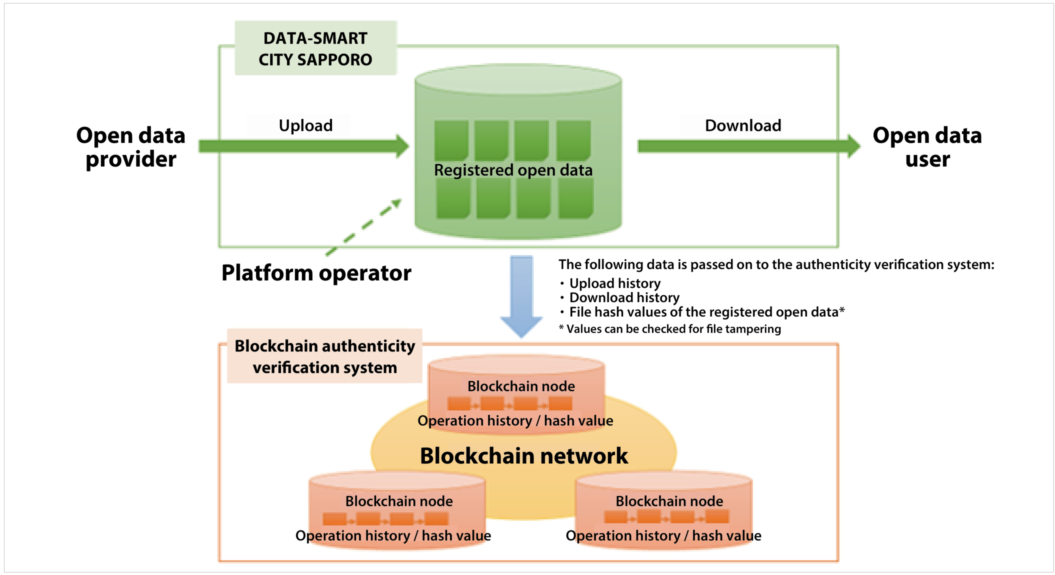
https://www.ntt-tx.co.jp/products/contractgate/pass.html

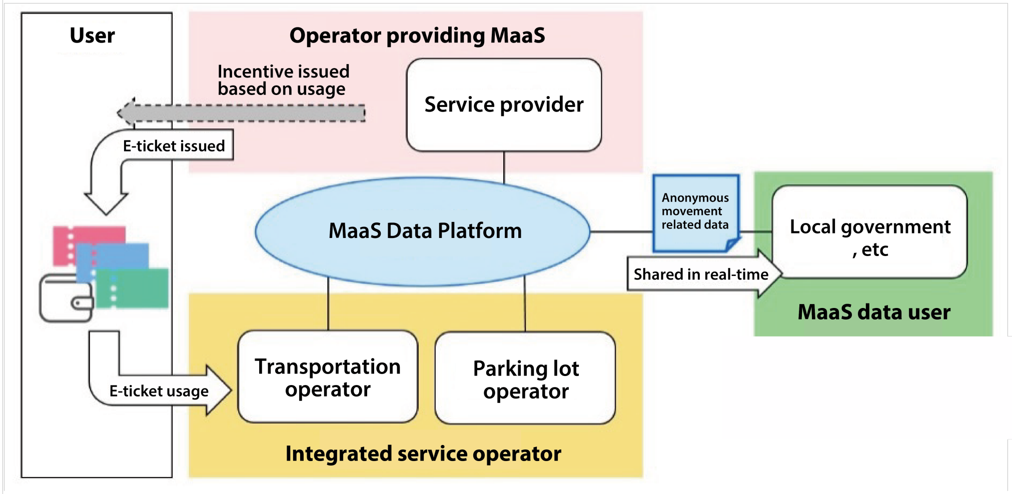
Monitor
A tool for viewing blockchain data



https://www.ntt-tx.co.jp/products/contractgate/monitor.html

2. Previous Proof-of-Concept


DATA-SMART
CITY SAPPORO
Open data provider
Upload
Registered open data
Download
Open data user
Platform operator
The following data is passed on to the authenticity verification system:
・Upload history
・Download history
・File hash values of the registered open data*
* Values can be checked for file tampering
Blockchain authenticity verification system
Blockchain node
Operation history / hash value
Blockchain network
Blockchain node
Operation history / hash value
Blockchain node
Operation history / hash value



From January 2018 to March 2018, a study was conducted with SAPPORO ELECTRONICS AND INDUSTRIES CULTIVATION FOUNDATION that introduced blockchain technology for open data on a platform operated by Sapporo City.

https://www.ntt-tx.co.jp/whatsnew/2018/181018.html

The "MaaS data platform" was developed to store, share, and utilize mobile data using blockchain technology. Presently, the Okinawa version of MaaS, that utilizes Okinawa's transportation IC card and covers all public transportation services, is being developed.

The usefulness was confirmed in a study carried out in Naha City / Tomigusuku City.

https://www.ntt-tx.co.jp/whatsnew/2021/210412.html

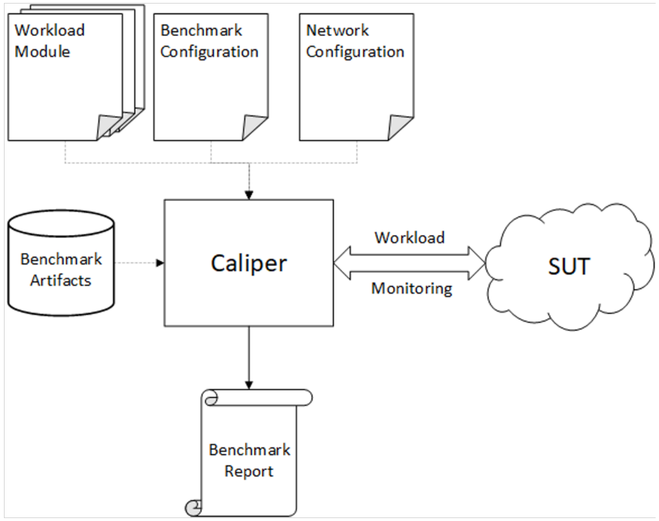
3. What is Hyperledger Caliper?

HYPERLEDGER CALIPER

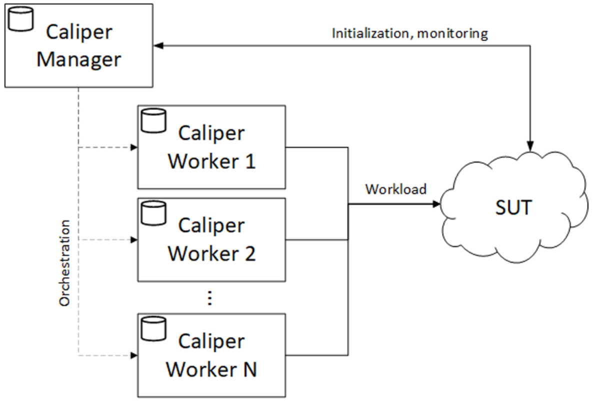
- A blockchain performance measurement tool provided by the Hyperledger community.
- It employs the Apache License Version 2.0.
- Different blockchain solutions can be tested and results obtained utilizing pre-defined use-cases.
- At the present time, it supports the following products.
  - Ethereum
  - Hyperledger Besu
  - Hyperledger Fabric (v1.x, v2.x)
  - FISCO BCOS
- The following values are able to be output as performance indicators.
  - SUCCESS RATE
  - Transaction throughput
  - Transaction wait time (minimum, maximum, average)

3. What is Hyperledger Caliper?

Caliper overview



Multiple worker load



https://hyperledger.github.io/caliper/v0.4.2/architecture/

- It generates a workload for the System Under Test (SUT) and monitor its responses continuously.
- It generates a report based on the observed SUT responses.
- It is even able to support load execution via multiple worker processes.

- Caliper includes the following components:
  - Benchmark configuration files: Benchmark execution method, SUT monitoring settings.
  - Network configuration files: Access settings for the SUT.
  - Workload modules: Loads the scripts to run (Node.js module).
  - Benchmark artifacts: The parts needed to run a benchmark. Smart contracts, etc.

4. Assessment with Hyperledger Caliper

- It is basically a tool that measures "the infrastructure segments of blockchain products".

- In the requirement specification phase of the study, BC products (Ethereum, HLF, Corda, etc.) may be selected. However, the selection is often made in terms of functionality rather than performance.

- Since the users do not require such specialized performance measurements, performance measurements between products is often not emphasized.

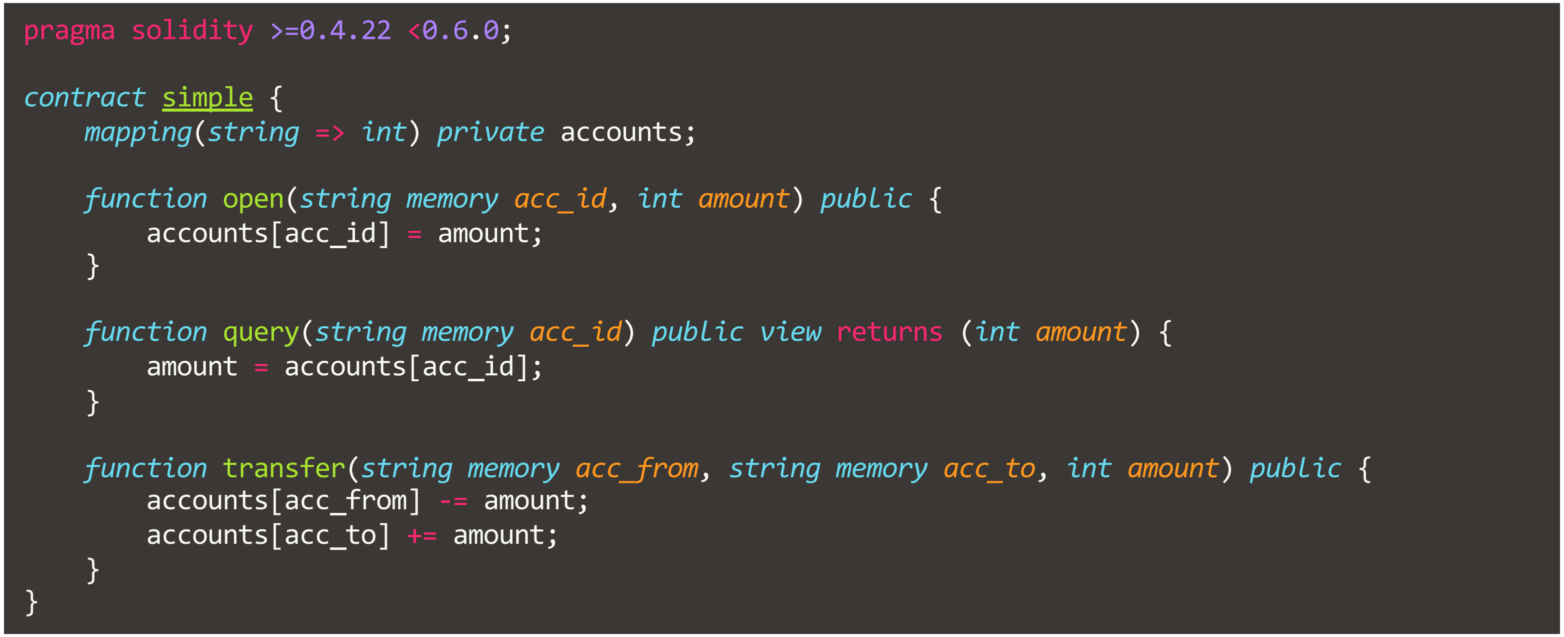
- As we decided that a proprietary functional survey of Hyperledger Besu would be conducted, we also decided to measure the performance characteristics of a simple Besu simultaneously.

- **Now, we would like to discuss the flow of Try & Error step by step.**

4. Assessment with Hyperledger Caliper

(1) Differential measurements by product

- Since both Go-Ethereum and Besu are smart contract implementations in the Solidity language, tests were carried out using common sources.



```
pragma solidity >=0.4.22 <0.6.0;

contract simple {
    mapping(string => int) private accounts;

    function open(string memory acc_id, int amount) public {
        accounts[acc_id] = amount;
    }

    function query(string memory acc_id) public view returns (int amount) {
        amount = accounts[acc_id];
    }

    function transfer(string memory acc_from, string memory acc_to, int amount) public {
        accounts[acc_from] -= amount;
        accounts[acc_to] += amount;
    }
}
```

- Simply record your account number and balance with a source that makes ERC20 even simpler.
    - Open : Update a single value for an associative array type.
    - Query : a single point reference to a value from an associative array type.
    - Transfer: Update two associative array values.

4. Assessment with Hyperledger Caliper

(1) Differential measurements by product

Go-Ethereum
Voyager Cluster (v1.10.4)

VS

HYPERLEDGER BESU
21.1.7

- First, once a small-scale environment has been created, verify that there is no performance degradation.

Machine specifications

CPU: Core i7 7700U(3.6GHz)
vCPU: 2
Memory: 8GB
Storage: 40GB

Blockchain settings

Consensus-forming algorithm: PoA (clique)
Block generation interval: 5 seconds
Block gas limit: 21,733,540
Number of nodes: 1

- Attempt to issue workload from Caliper

Workload tool settings

Issue pattern: fixed rate
Target TPS :
open : 50TPS
query: 100TPS
transfer: 50TPS

```
$ npx caliper launch manager ¥
--caliper-bind-sut besu:latest ¥
--caliper-benchconfig benchmarks/scenario/simple/config.yaml ¥
--caliper-networkconfig networks/besu/1node-clique/networkconfig.json ¥
--caliper-workspace .
```

4. Assessment with Hyperledger Caliper

(1) Differential measurements by product

Hyperledger Besu

```
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name     | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| open     | 1000 | 0    | 63.3            | 29.41           | 2.44            | 14.75           | 22.3             |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| query    | 1000 | 0    | 100.2           | 0.03            | 0.00            | 0.00            | 100.2            |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| transfer | 1000 | 0    | 58.1            | 19.06           | 0.39            | 9.41            | 28.6             |
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

Go-Ethereum

```
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name     | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| open     | 1000 | 0    | 63.0            | 26.32           | 2.15            | 14.36           | 25.0             |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| query    | 1000 | 0    | 100.1           | 0.02            | 0.00            | 0.00            | 100.1            |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| transfer | 1000 | 0    | 57.7            | 20.20           | 2.10            | 11.97           | 28.5             |
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

- ◆ The combination of the measurement conditions does not create a large difference between Go-Ethereum and Besu with these workload conditions.

- ◆ Of course, the difference between references and updates that do not have transactions issued is significant. (Particularly Latency, Response Time).

- ◆ Due to the timing of block generation, etc., the results of the measurements fluctuate (compared to conventional systems).
- ◆ In order to do it properly, it needs to be attempted several times before getting the total.

- ◆ One interesting point is that “open”, which updates a single value, produces a "slightly slower" result than Transfer, which runs two updates.
- ◆ In my opinion, it is assumed that modifying the two mapping values that already have storage slots available would be a lighter workload than the process of creating new storage slots.
- ◆ Please be aware that these characteristics may differ from the intuitive "processing size".

4. Assessment with Hyperledger Caliper

(2) Differential measurement of consensus-building

- Only the consensus-building settings were changed to get the performance difference.

HYPERLEDGER BESU

VS

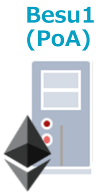
HYPERLEDGER BESU

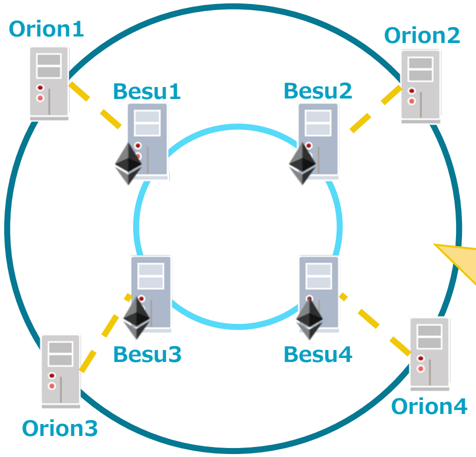
Blockchain settings

Consensus-forming algorithm: PoA (clique)
Block generation interval: 5 seconds
Block gas limit: 21,733,540
Number of nodes: 1

Blockchain settings

Consensus forming algorithm: IBFT
Block generation interval: 5 seconds
Block gas limit: 0x1fffffffffffff
Number of nodes: 4 (besu4+orion4)



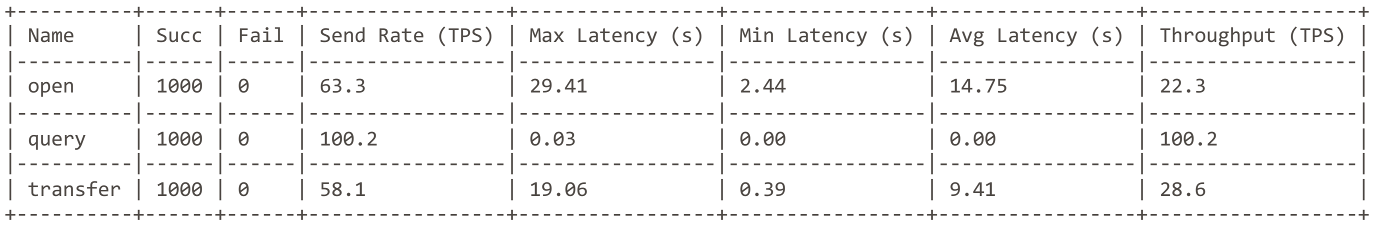


At first glance, due to the number of machines (processes) being different, it seems to be advantageous in IBFT, but the increase in the number of machines in IBFT does not actually give a performance advantage. The minimum number of nodes for each consensus formation was four
(set).

4. Assessment with Hyperledger Caliper

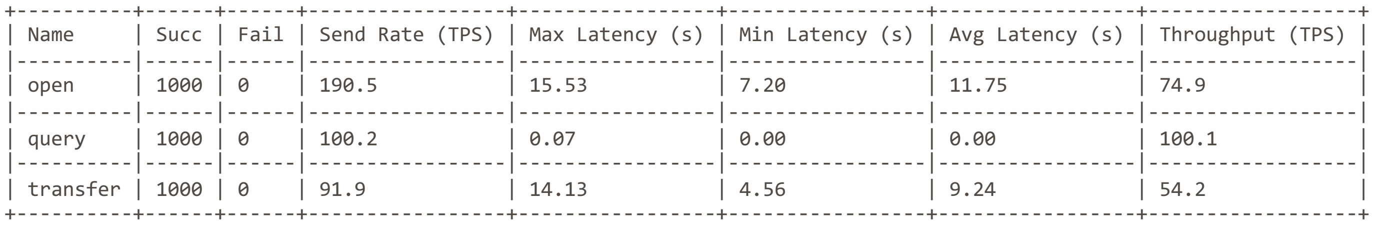
(2) Differential measurement of consensus building

Hyperledger Besu (PoA)



```
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name     | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| open     | 1000 | 0    | 63.3            | 29.41           | 2.44            | 14.75           | 22.3             |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| query    | 1000 | 0    | 100.2           | 0.03            | 0.00            | 0.00            | 100.2            |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| transfer | 1000 | 0    | 58.1            | 19.06           | 0.39            | 9.41            | 28.6             |
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

Hyperledger Besu (IBFT)



```
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name     | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| open     | 1000 | 0    | 190.5           | 15.53           | 7.20            | 11.75           | 74.9             |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| query    | 1000 | 0    | 100.2           | 0.07            | 0.00            | 0.00            | 100.1            |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| transfer | 1000 | 0    | 91.9            | 14.13           | 4.56            | 9.24            | 54.2             |
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

- **We expected that IBFT would be "heavier" by the consensus building overhead, but the results turned out to be different.**

- Overall, IBFT has a higher TPS, but "Min Latency" has PoA <IBFT, which is to be expected.
- In contrast, PoA has Tx with high "Max Latency".

- This result is unexpected as the gas upper limit was set differently.
    - PoA has a block gas upper limit of 21,733,540, while IBFT has a block gas limit of 377,777,777,777,777,777.
  - In PoA, the workload has “reached the upper limit of gas (so Max Latency is high)", whereas in IBFT, the difference in Avg Latency is small.

- In the assessment of blockchain products and systems, the upper limit of gas (block size) affects the processing performance of the whole system, so it is necessary to adjust the values appropriately when measuring.

4. Assessment with Hyperledger Caliper

(3) Differential measurement of block generation interval

- In this test, only the block generation interval has been changed to get the performance difference.

HYPERLEDGER BESU

HYPERLEDGER BESU

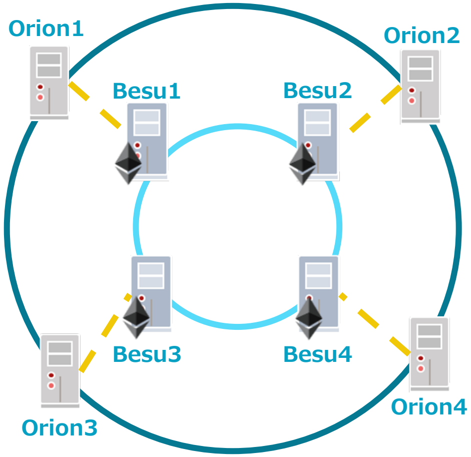
VS

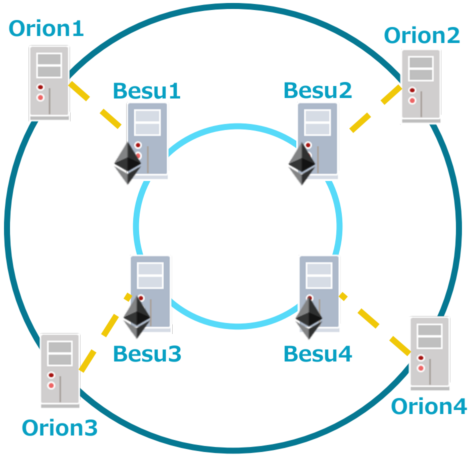
Blockchain settings

Consensus forming algorithm: IBFT
Block generation interval: 2 seconds
Block gas limit: 0x1fffffffffffff
Number of nodes: 4 (besu4+orion4)

Blockchain settings

Consensus forming algorithm: IBFT
Block generation interval: 5 seconds
Block gas limit: 0x1fffffffffffff
Number of nodes: 4 (besu4+orion4)





4. Assessment with Hyperledger Caliper

(3) Differential measurement of block generation interval

Hyperledger Besu (IBFT 5秒)

```
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name     | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| open     | 1000 | 0    | 190.5           | 15.53           | 7.20            | 11.75           | 74.9             |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| query    | 1000 | 0    | 100.2           | 0.07            | 0.00            | 0.00            | 100.1            |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| transfer | 1000 | 0    | 91.9            | 14.13           | 4.56            | 9.24            | 54.2             |
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

Hyperledger Besu (IBFT 2秒)

```
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name     | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| open     | 1000 | 0    | 309.2           | 17.27           | 6.45            | 12.18           | 78.4             |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| query    | 1000 | 0    | 100.2           | 0.15            | 0.00            | 0.00            | 100.2            |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| transfer | 1000 | 0    | 104.5           | 13.53           | 4.84            | 9.19            | 54.6             |
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

- **We expected 2 seconds to "be much faster" than 5 seconds, but the results were different from our expectations.**

- Strictly speaking, the block generation interval is different, and there is the setting of the block generation waiting time, so there is little impact on the performance even in load situations where "transactions are clogged to the limit".

- In an environment where the block sizes are sufficient, the number of nodes participating (voting) in IBFT and the response time of each node up until the specified number of votes is reached are critical.
- Since the test is performed in a network-wise "very close" situation, it is difficult to judge the effects of the 4-node configuration and block waiting time.

- In the evaluation of blockchain products and systems, the whole system network distribution status affects the processing performance, so it is necessary to examine the assumed environment and perform the test with the configurations set as close to the actual production (estimated) as possible.

4. Assessment with Hyperledger Caliper

(4) Differential measurement of node placement

- In this test, nodes were distributed across AZ on AWS to obtain performance differences.

HYPERLEDGER BESU

Blockchain settings

Consensus formation algorithm: IBFT
Block generation interval: 5 seconds
Block gas limit 0x1 fffffffffffff
Number of nodes: 4 (Besu4 / Orion4)

Machine environment

Instance: m5.large
vCPU: 2
Memory: 8GB
Storage: gp1

AWS

VS

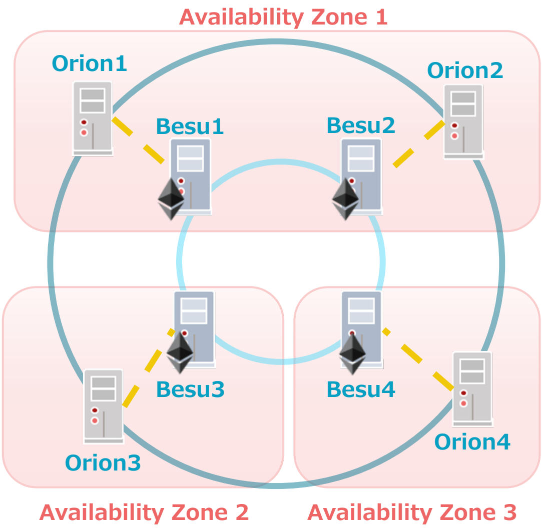
HYPERLEDGER BESU

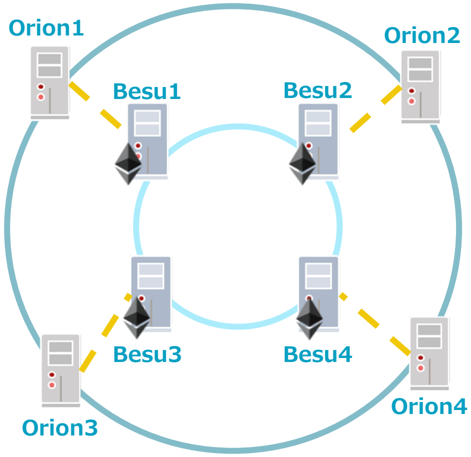
Blockchain settings

Consensus formation algorithm: IBFT
Block generation interval: 10 seconds
Block gas limit 0-1 fffffffffffff
Number of nodes:: 4 (Besu4 / Orion4)

Machine environment

CPU: Core i7 7700U(3.6GHz)
vCPU: 2
Memory: 8GB
Storage: 40GB





4. Assessment with Hyperledger Caliper

(4) Differential measurement of node placement

Hyperledger Besu (local)

```
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name     | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| open     | 1000 | 0    | 309.2           | 17.27           | 6.45            | 12.18           | 78.4             |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| query    | 1000 | 0    | 100.2           | 0.15            | 0.00            | 0.00            | 100.2            |
|----------|------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| transfer | 1000 | 0    | 104.5           | 13.53           | 4.84            | 9.19            | 54.6             |
+----------+------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

Hyperledger Besu (AWS)

```
+----------+-------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name     | Succ  | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|----------|-------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| open     | 10000 | 0    | 111.9           | 14.99           | 3.69            | 9.00            | 101.5            |
|----------|-------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| query    | 10000 | 0    | 100.0           | 0.01            | 0.00            | 0.00            | 100.0            |
|----------|-------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| transfer | 10000 | 0    | 110.1           | 14.82           | 2.97            | 8.99            | 101.5            |
+----------+-------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

- **With AWS, it was estimated that "the consensus building overhead time would be slower", but the results did not match our expectations.**

- The difference in the infrastructure segments was too great due to the differences between the local and cloud environments.

  - The latency of AZ-to-AZ communication, which was estimated to be a bottleneck, was originally not so slow (depending on DC or AZ, as well as time, but it is about 2.5 to 5.0 ms)

  - When the AZ was separated, naturally the EC2 instance also separates, which coincidentally resulted in improved performance.

- As each platform focuses on improving the cloud environment daily, it is a challenge to (fairly) compare local environment with past measurement data (And vice versa. It is **vital to comprehend it as the value "As Is" and measure it in an environment (cloud / on-premise) that matches the actual environment.**

5. Summary

◆ Hyperledger Caliper

- ◆ It makes it easier to measure loads repeatedly, so it’s easy to perform Try & Error measurements while changing the environment.
- ◆ In the future, it will be able to handle multiple executions and various transaction processes, and as the number of functions are expanded, it will be able to handle more complex load conditions, so we hope to further enhance its functions.

◆ Hyperledger Besu

- ◆ When it comes to "private network Ethereum clients," it's not inferior in performance compared to Go-Ethereum.
- ◆ There are many points to "judge" from the perspective of the developer, such as fine setting values. (Fixed difficulty, contract code size settings etc.)
- ◆ Additional functions that were not available in Go-Ethereum, such as permission networks or privacy groups, have been developed, so it easy to use as a product for EEA.
- ◆ In essence, the stand-out function of performance is the division of public/private Tx, so it is unfortunate that performance measurement could not be performed based on this.

- The sample script used for this verification has been published on Github by the participating members.

  https://github.com/hkiridera/caliper-benchmarks

- Additionally, please refer to the official website to read blog articles on the same topic..

  NTT TechnoCross homepage column Information field: "Performance measurement of BESU with Hyperledger Caliper"
  https://www.ntt-tx.co.jp/column/hyperledger_caliperbesu/210910/

Thank you for listening.