

# Items to be described as performance measurement conditions and results



Couger Inc.

Lead Project Manager: Keita Shimizu

# Since June this year, EHTerakoya has held discussions about blockchain performance evaluation indicators over seven expert workshops.

## Schedule

- Details of the previous discussions are as follows:

| # | Date       | Details   |
|---|------------|---|
| 1 | 6/3/2021   | • Working Group Goals, Progress, and Results                  |
| 2 | 6/21/2021  | • Important indicators for blockchain performance evaluation  |
| 3 | 7/7/2021   | • Blockchain performance indicator deep-dive                  |
| 4 | 8/5/2021   | • Test conditions and implementation method                   |
| 5 | 8/25/2021  | • Test implementation/results sharing and detected problems 1 |
| 6 | 9/16/2021  | • Test implementation/results sharing and detected problems 1 |
| 7 | 10/1/2021  | • Test implementation/results sharing and detected problems 2 |
| 8 | 10/29/2021 | • Public Workshop   |

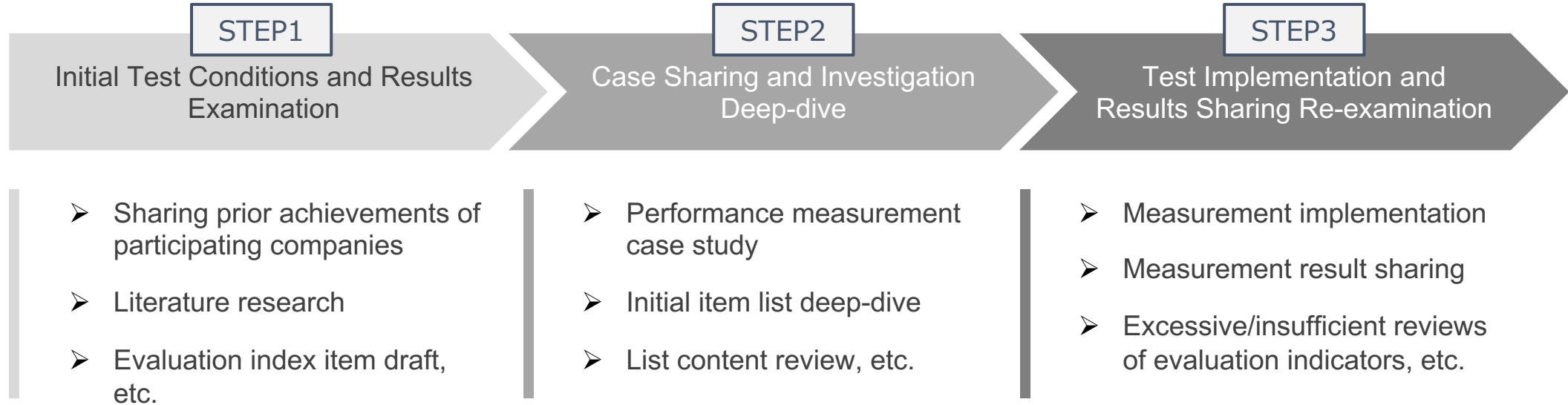
## Participating Companies

- We created working groups of companies that have track records in blockchain development and companies that are considering/have already started using blockchain in their business, and proceeded with discussions.



**As for test conditions and results requiring close scrutiny during performance verification, finalization for the initial listed items was accomplished by fleshing them out based on past cases, surveys and verifications of each participating company.**

### Performance Evaluation Index Scrutiny Process



**Each test condition / result item is categorized into three viewpoints and then further broken down into subcategories. Since there is no valid hardware information available from the results, we will elaborate on the remaining 5 classifications.**

### Breakdown of Test Conditions and Results

|              |                 | Outlook                |                      |                        |
|--------------|-----------------|------------------------|----------------------|------------------------|
|              |                 | Blockchain information | Hardware information | System Load Conditions |
| Input/Output | Test Conditions | ●<br>Valid             | ●<br>Valid           | ●<br>Valid             |
|              | Test Results    | ●<br>Valid             | —<br>Invalid         | ●<br>Valid             |

**Chain information is essential as a test condition. It is possible to make a fair chain comparison by setting the client's name, the consensus building algorithm, the target method, and the number of nodes as the test conditions.**

**Test Condition item 1- Chain Information**

| Category               | Item                    | Definition   | Example   |
|------------------------|-------------------------|--|---|
| Blockchain information | Blockchain Client Name  | Blockchain Client Name   | Ethereum / Geth, Hyperledger Fabric etc.                  |
|                        | Consensus Algorithm     | A method for making network-wide agreements in the blockchain  | Proof of Work (Ethash), Proof of Authority (Clique) etc.  |
|                        | Transaction Method      | A state transition method for changing blockchain data values. Processing subject performance measurement contents | Smart contract code used for performance evaluation, etc. |
|                        | Network Size (Node No.) | Number of validating nodes participating in consensus building   | Number of nodes   |

**Since hardware performance is also a factor that affects the performance of the chain, it is desirable to provide information on the operating environment of the chain and the load generation server.**

**Test Condition item 2- Hardware information**

| Category             | Environment           | Item                                    | Definition                           | Example                   |
|----------------------|-----------------------|---|--------------------------------------|---------------------------|
| Hardware information | In the cloud          | Utilized cloud services                 | Cloud services name                  | AWS, GCP, Azure etc.      |
|                      |                       | Instance type                           | Cloud services instance name         | 2.xlarge, t3.medium etc.  |
|                      |                       | RAM capacity                            | Cloud services instance RAM capacity | 16GB                      |
|                      | In the actual machine | CPU type                                | Machine CPU type                     | Intel Core i9 3.5GHz etc. |
|                      |                       | GPU type<br>※ When using GPU in the PoW | Machine GPU type                     | RTX 3090 etc.             |
|                      |                       | RAM capacity                            | Machine RAM capacity                 | 16GB                      |

Since using tools in the chain affects the test results, it is necessary to describe the system load conditions as well.

### Test Condition item 3- System Load Conditions

| Category               | Item   | Definition  | Example    |
|------------------------|--|---|------------|
| System Load Conditions | Load Generation Client Number                | Number of load bearing clients in the system                        | 200        |
|                        | Total Load Requests                          | Total number of requests sent from the client                       | 73,555 tps |
|                        | Load Duration                                | Total amount of time for the client load                            | 60s        |
|                        | Request limit number from a single load tool | Maximum number of requests that can be made from a single load tool | 13,555 tps |

**It is thought that chain performance can be seen by looking at the latency throughput for processing and CPU/disk loads as a test result.**

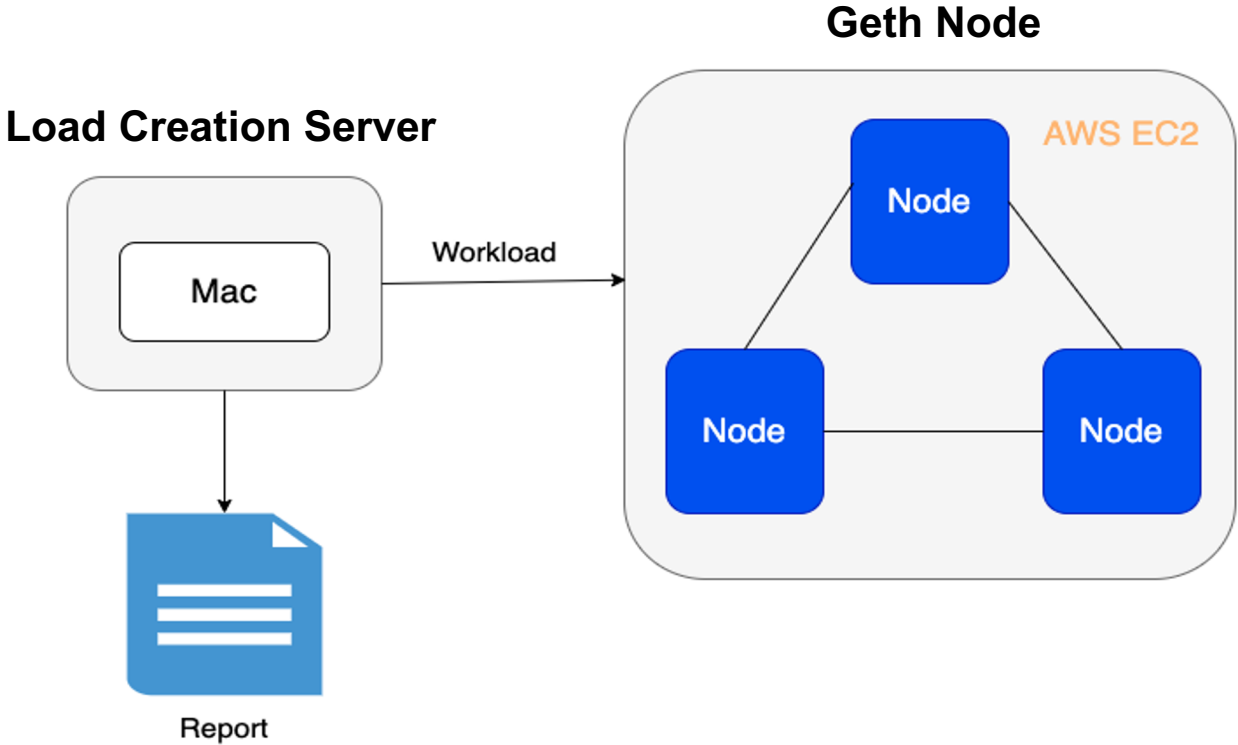
### Test Result Items

| Category                | Item                   | Definition  | Value (Example)                                    |
|-------------------------|------------------------|---|--|
| Blockchain Information  | Read Latency           | Total amount of time to send and receive read requests                                    | 0.18 s   |
|                         | Read Throughput        | Amount processed per second   | 813.1  |
|                         | Transaction Latency    | The time it take for the entire network to validate a transaction.                        | 11.18 s  |
|                         | Transaction Throughput | Percentage of valid transactions executed by the blockchain over a defined period of time | 27.4   |
| System Load Information | CPU Load               | Load on CPU   | Max: 56.7%<br>Avg: 21.83% etc.                     |
|                         | Disk Capacity Load     | Load on disk capacity   | Min:0.80 KB/s<br>Max:146.80 KB/s<br>Avg:49.47 KB/s |



We also built an actual 3-node Geth network, and verified performance by clarifying each test condition and result item.

System Configuration Overview Diagram



# The condition and result item information from the working groups is as follows.

## Test implementation condition-Blockchain information

- The client used Geth this time.
- The method targets Transfer for transaction measurement and balanceOf for read measurement.

| Item                    | Definition   | Value                       |
|-------------------------|--|-----------------------------|
| Blockchain Client Name  | Blockchain Client Name   | Geth                        |
| Consensus Algorithm     | A method for making network-wide agreements in the blockchain.   | Proof of Authority (Clique) |
| Transaction Method      | A state transition method for changing blockchain values. The content of the target performance measurement process. | Transfer balanceOf          |
| Network Size (Node No.) | The number of validator nodes participating in consensus building.   | 3 nodes                     |

## Test implementation condition-Hardware information

- Load Creation used a MacPC, and the chain nodes used AWS to create a network.

| Item                                 | Definition                           | Value                     |
|--------------------------------------|--------------------------------------|---------------------------|
| Load Creation Server: Actual Machine |                                      |                           |
| CPU Type                             | Machine's CPU Type                   | Apple M1                  |
| GPU Type<br>※When using GPU in PoW   | Machine's GPU Type                   | Apple Standalone Octacore |
| RAM Capacity                         | Machine's RAM Capacity               | 16 GB                     |
| Blockchain Name Node: Cloud          |                                      |                           |
| Utilized Cloud Services              | Cloud Services Name                  | AWS                       |
| Instance Type                        | Cloud Services Instance Name         | t2.xlarge                 |
| RAM Capacity                         | Cloud Services Instance Ram Capacity | 16 GB                     |

**Verification was run with the total number of load requests set to 1000. For individual conditions, the target TPS was 50 TPS, the block generation time was 10 seconds, and the number of approvals considered successful was 2 blocks.**

### Test Implementation Conditions- System Load Conditions

■ The total number of requests is set to 1,000.

| Item                           | Definition  | Value               |
|--------------------------------|---|---------------------|
| Load Creation Client Number    | Number of load bearing clients on the system                        | 1                   |
| Total Load Request Number      | Total number of requests sent from the client                       | 1,000 transactions  |
| Load Duration                  | Length of load time from the client                                 | 35 seconds          |
| Single Load Tool Request Limit | Maximum number of requests that can be made with a single load tool | 17,541 transactions |



### Test Implementation Results

■ CPU/Disk Load was measured every 5 seconds with iostat and vmstat

| Item                   | Definition   | Value  |
|------------------------|--|--|
| Read Latency           | Total time taken to send and receive a read request  | Min.: 2.37s<br>Max.: 4.59s<br>Avg.: 3.41s  |
| Read Throughput        | Number of readings per second  | 41.8 TPS   |
| Transaction Latency    | Amount of time it takes to validate all transactions in the entire network                 | Min.: 2.52s<br>Max.: 9.36s<br>Avg.: 5.88s  |
| Transaction Throughput | The rate at which a valid transaction is committed by the blockchain over a period of time | 35.6 TPS   |
| CPU Load               | Load on the CPU  | CPU Usage<br>Min.: 1%<br>Max.: 5%<br>Avg.: 3%  |
| Disk Capacity Load     | Load on the Disk Capacity  | HDD write average per 1 second<br>Min.: 0.80 KB/s<br>Max.: 146.80 KB/s<br>Avg.: 49.47 KB/s |

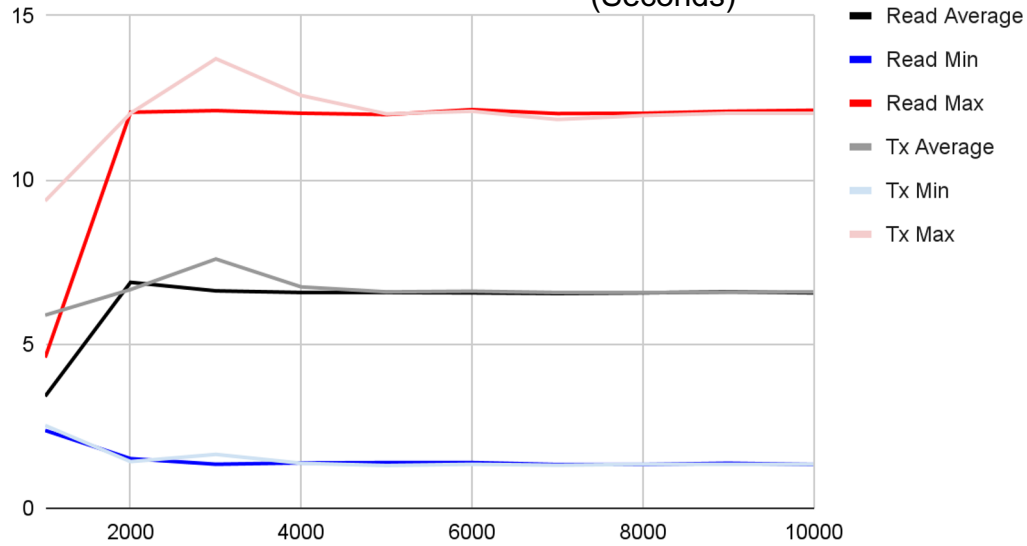
# **[Additional Conditions 1] Set Target TPS of 50 TPS : The following graph shows the increase in the total number of transactions**

## **Test Results**

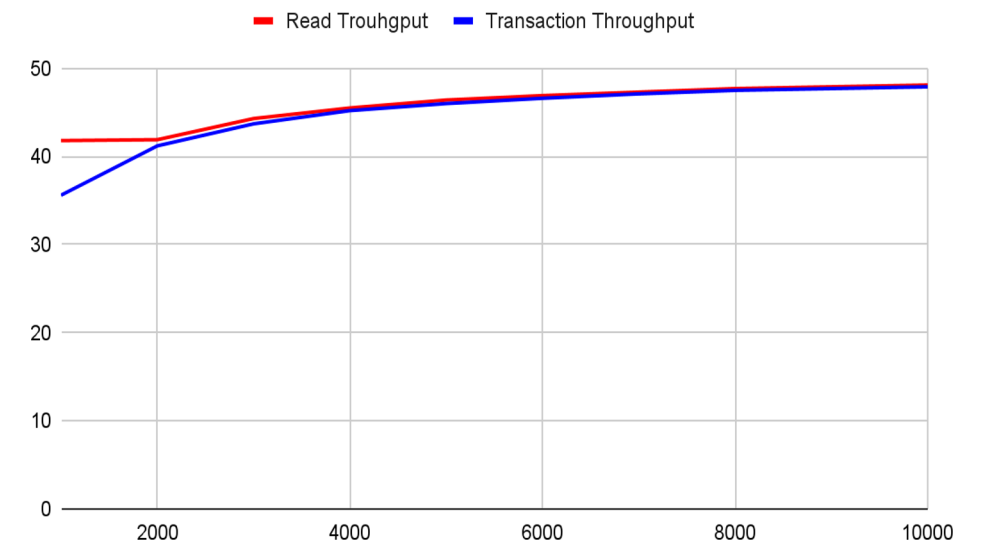
| Total Load Request No. | Read Latency                               | Read Throughput | Transaction Latency                        | Transaction Throughput | CPU Usage                            | Disk Space Load   |
|------------------------|--|-----------------|--|------------------------|--------------------------------------|---|
| 1000                   | Min.: 2.37s<br>Max.: 4.59s<br>Avg.: 3.41s  | 41.8 TPS        | Min.: 2.52s<br>Max.: 9.36s<br>Avg.: 5.88s  | 35.6 TPS               | Min.: 1%<br>Max.: 5%<br>Avg.: 3%     | Min.: 0.80 KB/s<br>Max.: 146.80 KB/s<br>Avg.: 49.47 KB/s  |
| 2000                   | Min.: 1.51s<br>Max.: 12.06s<br>Avg.: 6.88s | 41.9 TPS        | Min.: 1.42s<br>Max.: 12.03s<br>Avg.: 6.66s | 41.2 TPS               | Min.: 2%<br>Max.: 11%<br>Avg.: 5.29% | Min.: 0.80 KB/s<br>Max.: 408.80 KB/s<br>Avg.: 73.55 KB/s  |
| 3000                   | Min.: 1.34s<br>Max.: 12.11s<br>Avg.: 6.62s | 44.3 TPS        | Min.: 1.64s<br>Max.: 13.69s<br>Avg.: 7.59s | 43.7 TPS               | Min.: 1%<br>Max.: 12%<br>Avg.: 4.78% | Min.: 0.30 KB/s<br>Max.: 496.00 KB/s<br>Avg.: 69.37 KB/s  |
| ...                    | ...  | ...             | ...  | ...                    | ...                                  | ...   |
| 9000                   | Min.: 1.37s<br>Max.: 12.09s<br>Avg.: 6.59s | 47.9 TPS        | Min.: 1.34s<br>Max.: 12.03s<br>Avg.: 6.58s | 47.7 TPS               | Min.: 1%<br>Max.: 15%<br>Avg.: 5.96% | Min.: 0.80 KB/s<br>Max.: 571.60 KB/s<br>Avg.: 87.43 KB/s  |
| 10000                  | Min.: 1.33s<br>Max.: 12.12s<br>Avg.: 6.56s | 48.1 TPS        | Min.: 1.34s<br>Max.: 12.03s<br>Avg.: 6.59s | 47.9 TPS               | Min.: 1%<br>Max.: 16%<br>Avg.: 6.41% | Min.: 0.80 KB/s<br>Max.: 605.60 KB/s<br>Avg.: 109.76 KB/s |

※The total load request range from 4000-8000 is omitted due to space limitations.

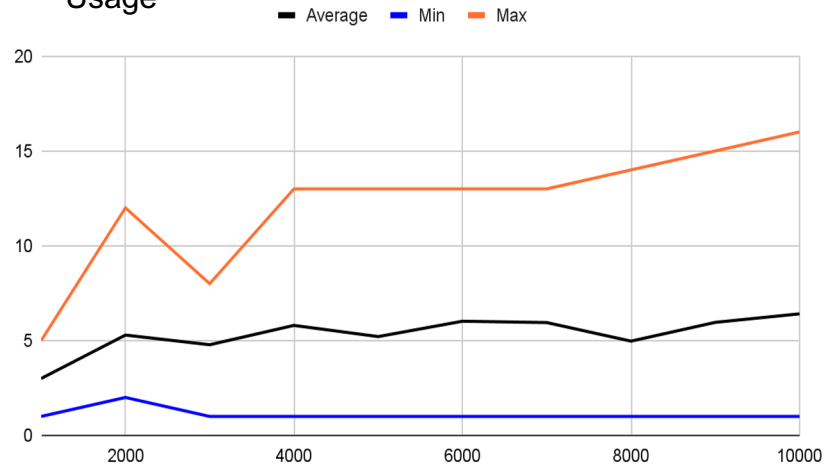
### Read Latency / Transaction Latency (秒) (Seconds)



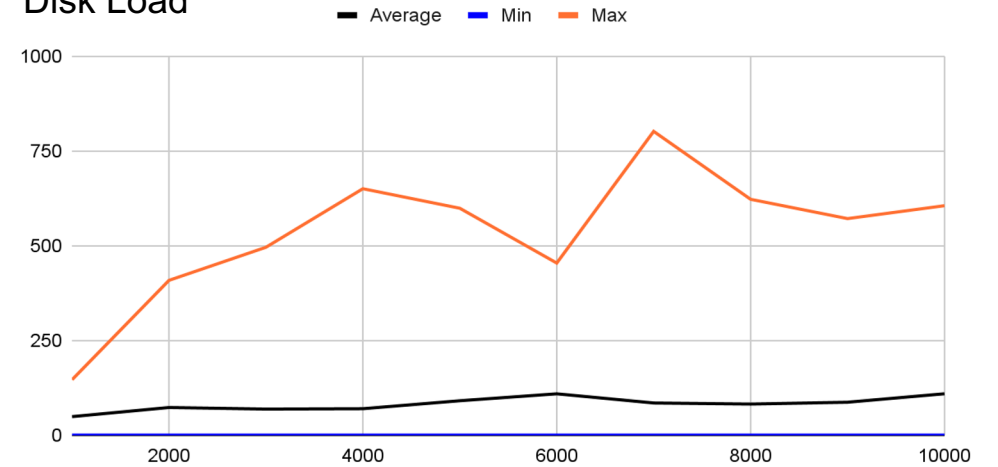
### Read Throughput / Transaction Throughput (TPS)



### CPU使用率 (%) Usage



### ディスク負荷 (KB/s) Disk Load

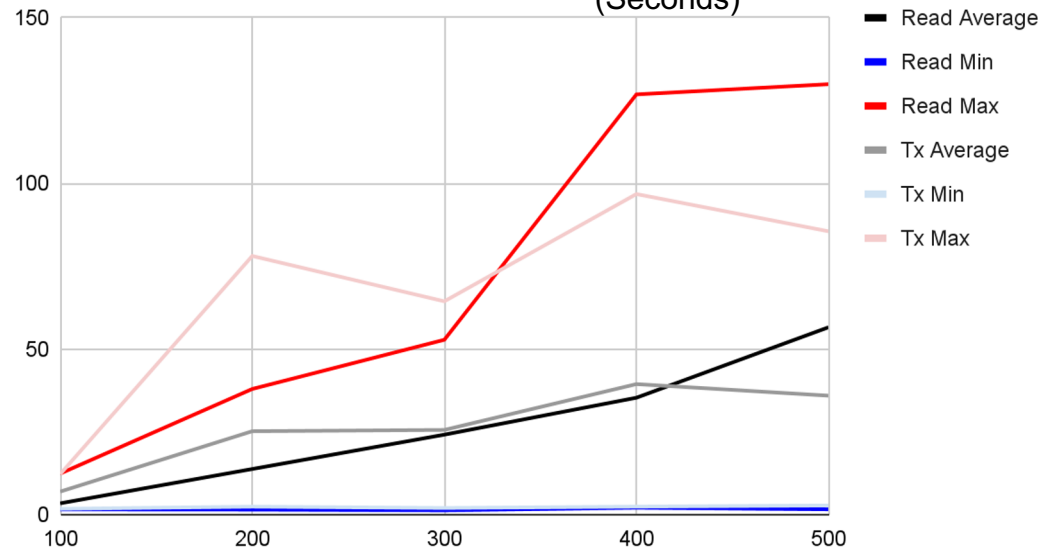


**[Additional Conditions 2] Total number of transactions set to 10,000: the following graph shows the increase in the target TPS.**

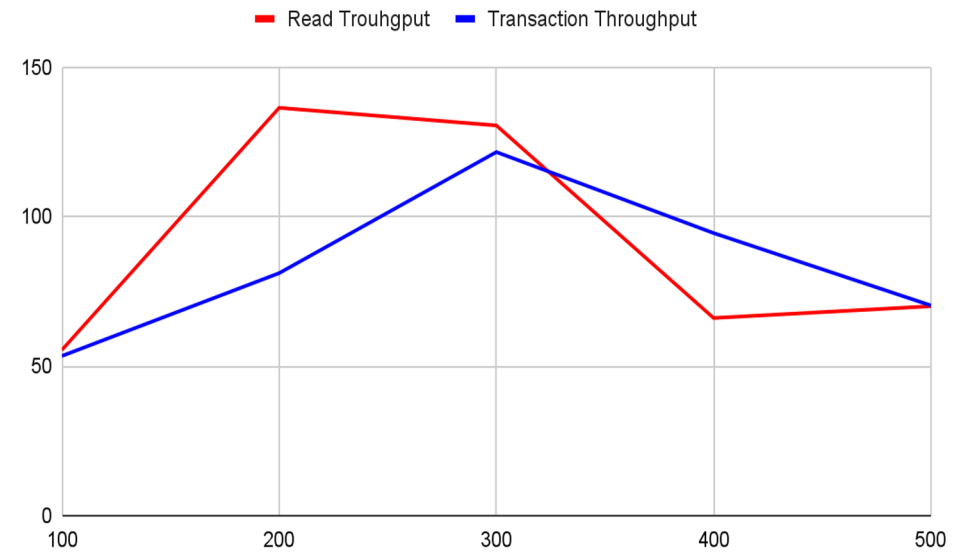
## Test Results

| Target TPS | Read Latency                                 | Read Throughput | Transaction Latency                         | Transaction Throughput | CPU Usage                              | Disk Space Load  |
|------------|--|-----------------|---|------------------------|--|--|
| 100        | Min.: 1.57s<br>Max.: 12.39s<br>Avg.: 3.41s   | 55.4 TPS        | Min.: 1.75s<br>Max.: 12.23s<br>Avg.: 6.95s  | 53.4 TPS               | Min.: 1%<br>Max.: 39%<br>Avg.: 17%     | Min.: 0.8 KB/s<br>Max.: 1093 KB/s<br>Avg.: 147.58 KB/s     |
| 200        | Min.: 1.51s<br>Max.: 37.93s<br>Avg.: 13.77s  | 136.4 TPS       | Min.: 2.42s<br>Max.: 78.08s<br>Avg.: 25.18s | 81.1 TPS               | Min.: 1%<br>Max.: 83%<br>Avg.: 53.65%  | Min.: 0.8 KB/s<br>Max.: 1072 KB/s<br>Avg.: 125.6 KB/s      |
| 300        | Min.: 1.29s<br>Max.: 52.78s<br>Avg.: 24.14s  | 130.5 TPS       | Min.: 2.01s<br>Max.: 64.39s<br>Avg.: 25.57s | 121.6 TPS              | Min.: 1%<br>Max.: 82%<br>Avg.: 56.78%  | Min.: 0.7 KB/s<br>Max.: 2708 KB/s<br>Avg.: 192.75 KB/s     |
| 400        | Min.: 2.09s<br>Max.: 126.85s<br>Avg.: 35.29s | 66.1 TPS        | Min.: 2.43s<br>Max.: 96.77s<br>Avg.: 39.40s | 94.5 TPS               | Min.: 1%<br>Max.: 82%<br>Avg.: 75.42%  | Min.: 0.7 KB/s<br>Max.: 1894 KB/s<br>Avg.: 108.02 KB/s     |
| 500        | Min.: 1.61s<br>Max.: 129.93s<br>Avg.: 56.55s | 70.3 TPS        | Min.: 2.76s<br>Max.: 85.53s<br>Avg.: 35.91s | 70.3 TPS               | Min.: 1%<br>Max.: 82%<br>Avg.: 63.717% | Min.: 0.60 KB/s<br>Max.: 2947.2 KB/s<br>Avg.: 129.093 KB/s |

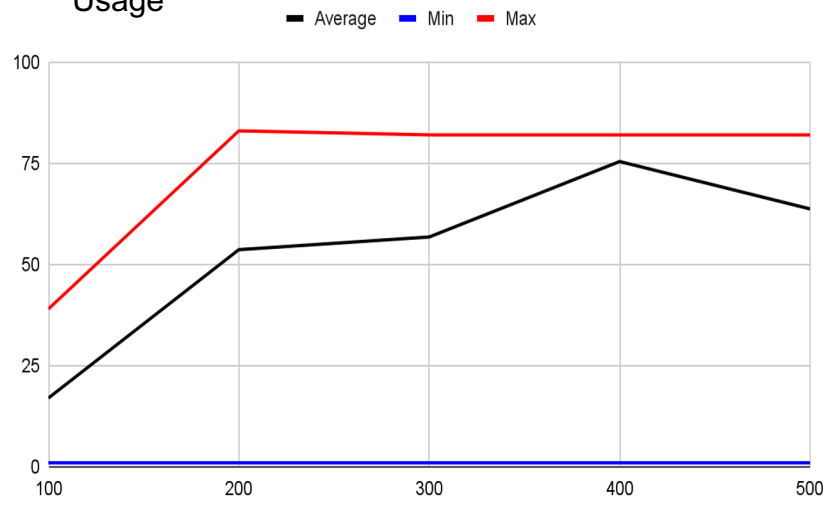
### Read Latency / Transaction Latency (秒) (Seconds)



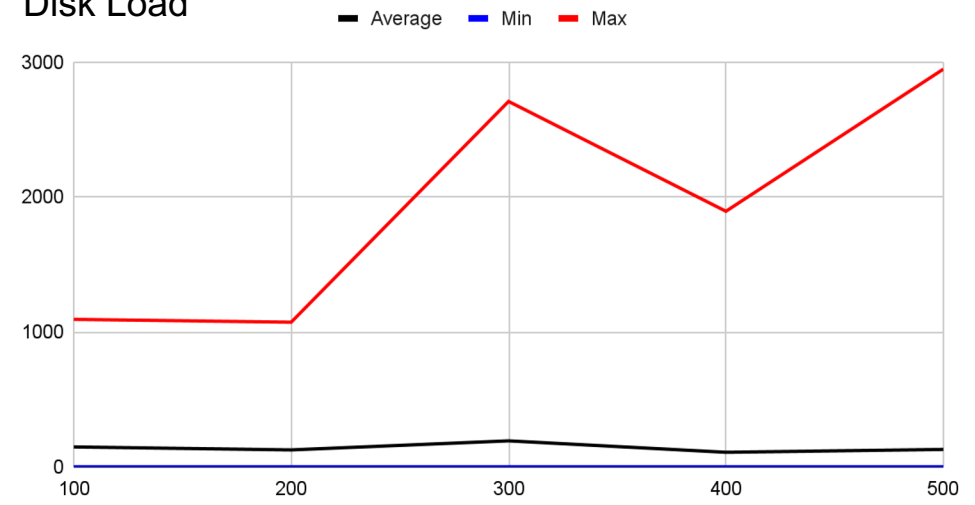
### Read Throughput / Transaction Throughput (TPS)



### CPU使用率 (%) Usage



### ディスク負荷 (KB/s) Disk Load



Since the measurement was performed with the total number of load requests and the target TPS as variables, it was possible to verify the performance limit points and possibilities of Geth.

## Test Results Summary

### 1 Target TPS set to TPS of 50: Increase the total number of transactions

- The ThroughPut comes close to 50TPS for increasing the total number of transactions.
- The latency and MAX CPU/Disk Load values become larger with a unilateral variation increase, but only a slight average value increase.  
The request seems to be handled as usual with a minimal load application even when setting the target TPS to 50 TPS and increasing the number of requests to 10,000.
- However, if there is a further increase in the number of transactions pending transactions will start to accumulate, with a lag in processing at 17,000, so the limit number of requests for a single load tool appears to be 17,500.

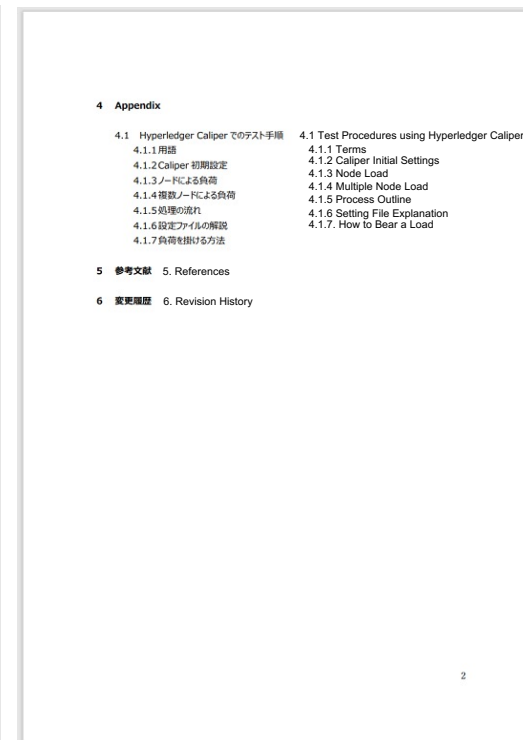
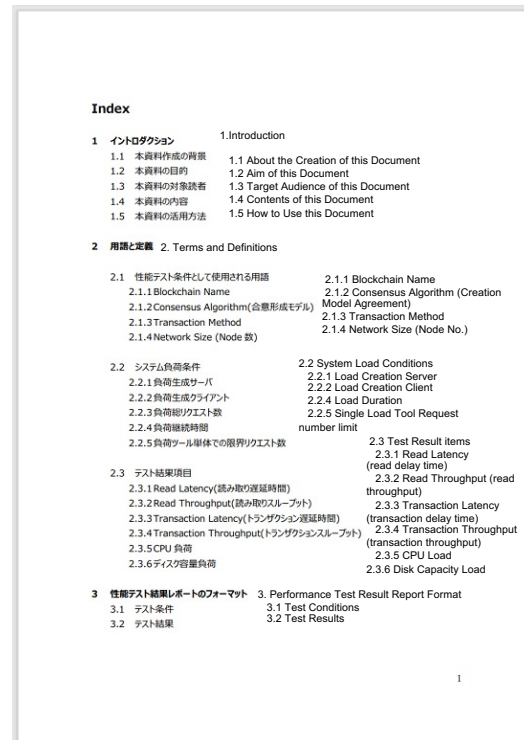
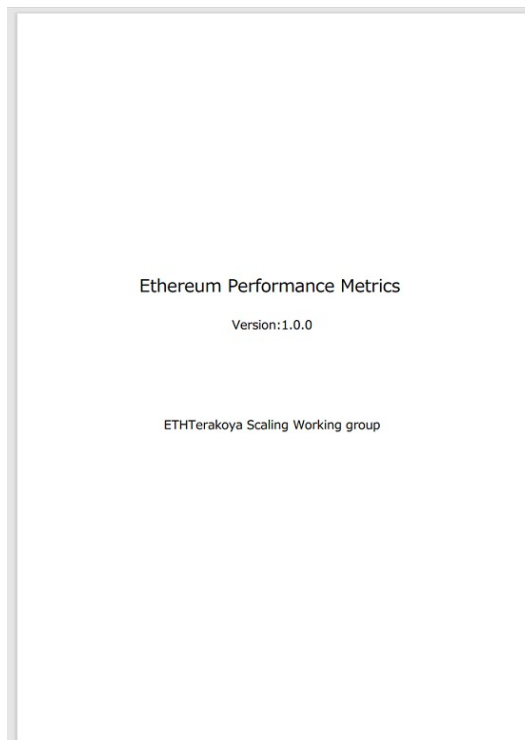
### 2 Total number of transactions set to 10,000: increase the target TPS

- The Maximum ThroughPut is somewhere between 200-300 TPS, with it actually decreasing at over 300 TPS. The resulting CPU usage and disk load increases, but the increased latency and load only increase without an improvement in the ThroughPut, so these seem to be the maximum values.
- Ethereum mainnet is said to run at 15 TPS, so it seems that the speed can be expected to increase by about 20 times when making it a private chain at the expense of decentralization.



Having a common evaluation index allows for verification and performance comparison of a wide variety of chains under the same conditions. I ask for your cooperation if you have comments about the white paper or are considering referencing it.

For Publication/Update of White Paper:



- Github repository : [xxxxxxx](#)
- ETHTerakoyaWEB : <https://ethereum-terakoya.org/scaling/>
- Contact information : ETHTerakoya Executive Office [contact@ethereum-terakoya.org](mailto:contact@ethereum-terakoya.org)