

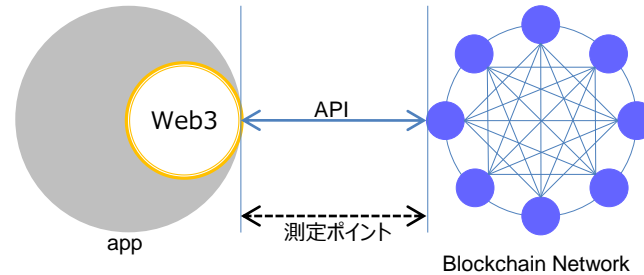


## Ethereum性能評価に関して

株式会社 日立ソリューションズ デジタルシフト開発支援本部  
モダン開発支援ソリューション部 先端技術グループ  
吉田 一省

# 1. 測定ポイント

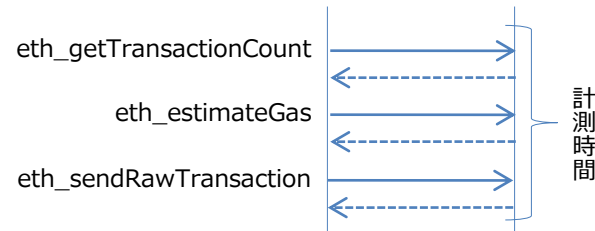
## ■ 測定ポイント



## ■ 測定種別

- ① 台帳更新系: eth\_sendRawTransaction
- ② 台帳参照系: eth\_call

①は、Transactionを作成する為に以下のセットがよりappからの視点で値が近づく。

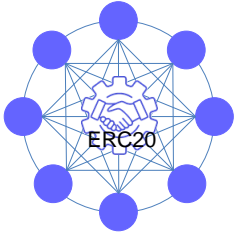


また、①はトランザクションへの署名(クライアント側)の処理時間も含まれる。

純粋なeth\_sendRawTransactionの応答を測定したい場合、TransactionCountは測定アプリ側でカウントアップしガス量も固定にしておき、トランザクションのデータの署名後のストリームの送受信時間を計測する。連続してトランザクション投入する場合は署名時間をどう切り離すか検討要。

## 2. 固定項目と変更項目

### ■ 測定するスマートコントラクト



Blockchain Network

測定するスマートコントラクトは固定化する。(ERC20など)  
また、スマートを変える事によってスマートコントラクト単位での測定が可能。

ERC20などでは初期発行量を決める必要がある。また、測定メソッドも決める必要あり。例えば、①:balanceOf(), ②:transfer()など。

### ■ genesis.json

性能測定用に前提条件として固定値を決めておく必要がある。パラメータによって、コンセンサスアルゴリズム単位に決めておく必要がある。



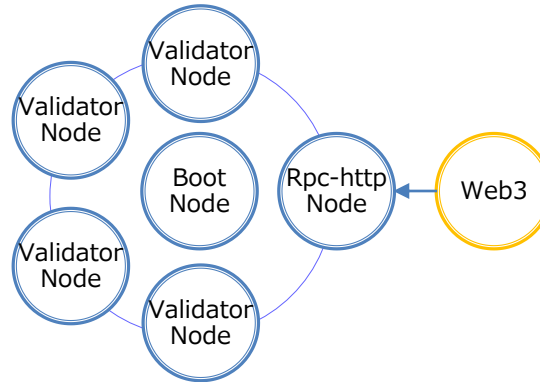
```
{
  "config": {
    "chainId": 2018,
    "constantinoplefixblock": 0,
    "ibft2": {
      "blockperiodseconds": 2, 最小ブロック時間
      "epochlength": 30000, 投票リセットブロック数
      "requesttimeoutseconds": 4, コンセンサスラウンドのタイムアウト
      "blockreward": "報酬額(WEI)",
      "miningbeneficiary": "0x + アドレス" ブロック決定報酬先のアドレス
    }
  },
  difficulty: "0x100", マイニング難易度
  alloc: {
    "0x+address": { "balance": "WEI量" }, 性能評価用のアドレスとETH量
  }
  coinbase: "0x+address" ETH消費の為性能評価用のアドレスと一致させる
  gasLimit: "0x1fffffffffffff", ガス代を不要にする場合、本値を設定(Besu)
}
```

## 2. 固定項目と変更項目(2)

### ■ ノード数

コンセンサスアルゴリズムにより最低必要なノード数で測定する。

例) IBFT2.0 の場合、 $3F+1$ ノードのうちF台がNGでもトランザクション確定したがって、Validatorノードは4台。



### ■ ブートノード

Bootnodeを配置した構成にするか、各ノードへ各ノードのIPアドレスとポートとアドレスを静的に与えるか決めておく必要がある。bootnodeへの問い合わせをなくした環境での時間を計測する場合は静的に与える。

### ■ コンセンサスアルゴリズム

	Ethash	Raft	Clique	IBFT	IBFT2	QBFT
Besu	○		○		○	○
GoQuorum		○	○	○		○
GoEthereum	○		○			

## 3. 測定項目

### ■ Hyperledger Besu がPrometheusへ公開するメトリクス(ご紹介)

#### プロセスメトリクス(4項目)

process_cpu_seconds_total	Counter	Total user and system CPU time spent in seconds
process_max_fds	Gauge	Maximum number of open file descriptors
process_open_fds	Gauge	Number of open file descriptors
process_start_time_seconds	Gauge	Start time of the process since Unix epoch in seconds

#### ブロック関連メトリクス(4項目)

ethereum_best_known_block_number	Gauge	Estimated highest block available (JSON-RPC equivalent: highestBlock of eth_syncing, or eth_blockNumber if not syncing)
ethereum_blockchain_height	Gauge	Current height of the canonical chain (JSON-RPC equivalent: eth_blockNumber)
ethereum_peer_count	Gauge	Current number of peers connected (JSON-RPC equivalent: net_peerCount)
ethereum_peer_limit	Gauge	Maximum number of peers this node allows to connect

#### Besu関連メトリクス(45項目)

besu_blockchain_chain_head_gas_limit	Gauge	Block gas limit of the current chain head block
besu_blockchain_chain_head_gas_used	Gauge	Gas used by the current chain head block
besu_blockchain_chain_head_ommers_count	Gauge	Number of uncles in the current chain head block (JSON-RPC equivalent: eth_getUncleCountByBlockHash or eth_getUncleCountByBlockNumber)
besu_blockchain_chain_head_timestamp	Gauge	Timestamp from the current chain head
besu_blockchain_chain_head_transaction_count	Gauge	Number of transactions in the current chain head block (JSON-RPC equivalent: eth_getBlockTransactionCountByHash or eth_getBlockTransactionCountByNumber)
besu_blockchain_difficulty_total	Gauge	Difficulty of the chain head (JSON-RPC equivalent: difficulty of admin_peers)

#### JVM関連メトリクス(22項目)

jvm_buffer_pool_capacity_bytes	Gauge	Bytes capacity of a given JVM buffer pool
jvm_buffer_pool_used_buffers	Gauge	Used buffers of a given JVM buffer pool
jvm_buffer_pool_used_bytes	Gauge	Used bytes of a given JVM buffer pool
jvm_classes_loaded	Gauge	Current number of classes loaded in the JVM
jvm_classes_loaded_total	Counter	Total number of classes loaded since the JVM started execution

### ■ 測定項目

- (1) APIの発行と戻り時間(受付, トランザクション確定<ファイナリティの扱いをどうするか?>)
- (2) 一定期間でのトランザクション発行可能数

- (a) CPU利用率
- (b) FD数
- (c)
- (d)

**HITACHI**  
Inspire the Next 