HITACHI
Inspire the Next

Benchmark Tools

2021/07/07 日立製作所

HITACHI
Inspire the Next

Contents

1. Benchmark Tools

HITACHI
Inspire the Next

いくつかのbenchmark toolsが存在する

- ethereum/test-tools
  https://github.com/ethereum/test-tools
- Chainhammer
  https://github.com/drandreaskrueger/chainhammer
- BCTMark
  https://gitlab.inria.fr/dsaingre/bctmark
- Blockbench
  https://www.comp.nus.edu.sg/~dbsystem/blockbench/
- Hyperledger Caliper
  https://github.com/hyperledger/caliper

2-1. 基礎理解

HITACHI
Inspire the Next

- 前提
  - Bitcoinは論理上は最大で7tps（実際は3~4tps）
  - Ethereumは10~15tpsを処理している
    https://ethereum.org/ja/developers/docs/dapps/#implications-of-dapp-development
  - Hyperledger Fabricは論文では2200tpsだが一般的な環境では700~1200tps
    https://ieeexplore.ieee.org/document/9169454
    https://www.ibm.com/blogs/blockchain/2019/01/answering-your-questions-on-hyperledger-fabric-performance-and-scale/
- 論理値と実測値など混在、また測定環境もツールも同じではない
- 誰のために何を測定するのかが重要
  - 開発者であれば多くのメトリクスを取り改善に役立てたい
  - ユーザーはTPSで十分かもしれない

2-2. Hyperledger Performance and Scale Working Group

HITACHI
Inspire the Next

- 異なるブロックチェーン間のパフォーマンスを測定・比較することは非常に困難
- パフォーマンスを正確に一貫して評価するために用語や指標を定義する必要がある
- 主要なメトリクスの定義
  - **Read Latency** = レスポンスを受け取った時間 – 送信時間
  - **Read Throughput** = 総読み取り操作数/総時間
    定義された時間内にどれだけ多くの読み取り操作が完了したかを示す。RPSで表す
  - **Transaction Latency** = (確認時間@閾値) - 送信時間
    トランザクションの結果がネットワーク上で使用可能になるまでの時間。全てのノードで結果を確認できるため伝搬時間やコンセンサスの時間が含まれる。ネットワーク上の50%や90%のノードで結果が反映されるネットワークの閾値も設定できる
  - **Transaction Throughput** = コミットされたトランザクションの総数/コミットされたノードの合計時間@コミットしたノード数
    1つのノードではなくネットワークの全てのノードでコミットされたレート。TPSで表す。無効なトランザクションは含まれない

2-3. Hyperledger Performance and Scale Working Group

HITACHI
Inspire the Next

- テスト環境として考慮すべきこと
  - コンセンサスプロトコル
  - ノードの地理的分布
  - ハードウェア環境
  - ネットワークモデル（FWを利用している等）
  - テストトランザクションに関与するノードの数
  - 依存するソフトウェアコンポーネント
  - テストツール
  - 利用するデータストアの種類（CouchDB, H2, Postgres等）
  - ワークロード
- これらの詳細を明らかにすることで、プラットフォーム間でのパフォーマンステストの比較が容易になるため、テスト結果の一部として記載する必要がある

3 - 1 . Hyperledger Caliper

HITACHI
Inspire the Next

- Blockchainのベンチマークツール、前述のPSWGも開発に協力
- Hyperledger Fabric, Besu, Ethereum, FISCO BCOSがサポートされている
- ネットワークの定義（計測環境）とワークロードの定義（どのようなTxを発行するか）

ネットワーク

```
"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
      }
    }
  }
}
```

ワークロード

```
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. 測定結果

HITACHI
Inspire the Next

- Sampleで用意されているシナリオの測定結果
  - ethereum/client-go
  - 1ノード（m5.2xlarge）で実行

前ページに記載されたワークロード

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

各パラメータを10倍にした結果

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

HITACHI

Inspire the Next