연구 보고서

작성자	김한호	작성일자	2021.04.18.

1. 연구 계획

- Hyperledger caliper를 이용해 각 Peer에 속한 유저의 성능을 측정.
- 다른 Peer에 다른 함수를 실행하여 성능을 측정.

2. 논문 연구 진행

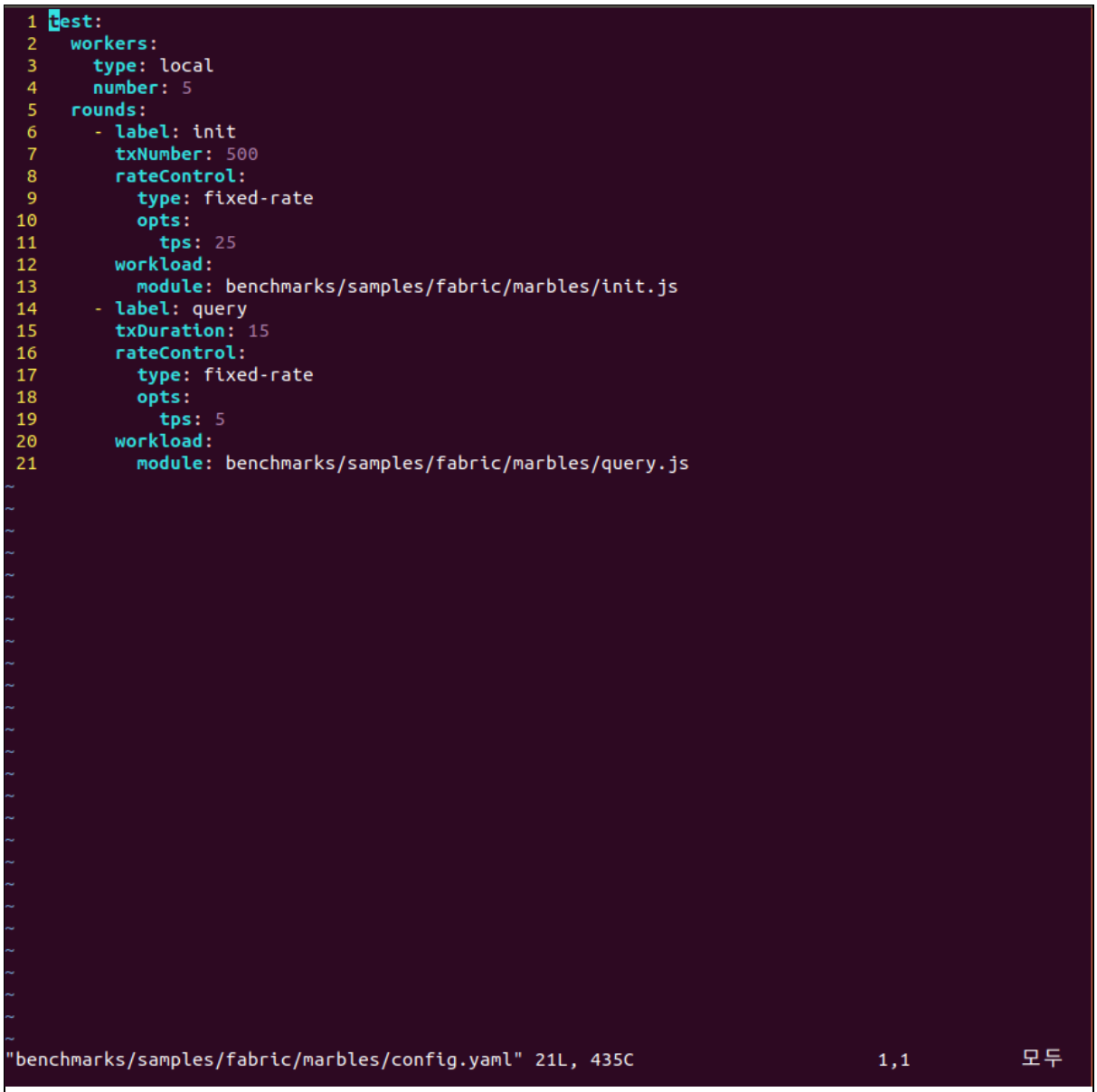
하이퍼레저 패브릭 테스트 네트워크를 생성함.

```
h2kim@h2kim-VBox:~/fabric-samples/test-network$ ./network.sh up createChannel -s couchdb
Creating channel 'mychannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds a
nd using database 'couchdb with crypto from 'cryptogen'
Bringing up network
LOCAL_VERSION=2.2.0
DOCKER_IMAGE_VERSION=2.2.0
/home/h2kim/fabric-samples/test-network/../bin/cryptogen
Generate certificates using cryptogen tool
Create Org1 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org1.yaml --output=organiza
tions
org1.example.com
+ res=0
Create Org2 Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-org2.yaml --output=organiza
tions
org2.example.com
+ res=0
Create Orderer Org Identities
+ cryptogen generate --config=./organizations/cryptogen/crypto-config-orderer.yaml --output=organ
izations
+ res=0
Generate CCP files for Org1 and Org2
/home/h2kim/fabric-samples/test-network/../bin/configtxgen
Generating Orderer Genesis block
+ configtxgen -profile TwoOrgsOrdererGenesis -channelID system-channel -outputBlock ./system-gene
sis-block/genesis.block
2021-04-11 17:12:43.789 KST [common.tools.configtxgen] main -> INFO 001 Loading configuration
2021-04-11 17:12:43.837 KST [common.tools.configtxgen.localconfig] completeInitialization -> INFO
 002 orderer type: etcdraft
2021-04-11 17:12:43.837 KST [common.tools.configtxgen.localconfig] completeInitialization -> INFO
 003 Orderer.EtcdRaft.Options unset, setting to tick_interval:"500ms" election_tick:10 heartbeat_
tick:1 max_inflight_blocks:5 snapshot_interval_size:16777216
2021-04-11 17:12:43.837 KST [common.tools.configtxgen.localconfig] Load -> INFO 004 Loaded config
uration: /home/h2kim/fabric-samples/test-network/configtx/configtx.yaml
2021-04-11 17:12:43.838 KST [common.tools.configtxgen] doOutputBlock -> INFO 005 Generating genes
is block
2021-04-11 17:12:43.839 KST [common.tools.configtxgen] doOutputBlock -> INFO 006 Writing genesis
block
+ res=0
Creating network "net_test" with the default driver
Creating volume "net_peer0.org1.example.com" with default driver
Creating volume "net_peer0.org2.example.com" with default driver
Creating volume "net_orderer.example.com" with default driver
Creating orderer.example.com ...
Creating couchdb0 ...
Creating couchdb1 ...
```

$ ./network.sh deployCC -ccn marbles -ccl go -ccp ../chaincode/marbles/go/ 명령어를 사용하여 피어에 체인코드를 배포함.

```
name: Calier test
version: "2.0.0"

caliper:
  blockchain: fabric

channels:
  - channelName: mychannel
    msps: ['Org1MSP', 'Org2MSP']
    contracts:
    - id: marbles

organizations:
  - mspid: Org1MSP
    identities:
      certificates:
      - name: 'user1.org1.example.com'
        clientPrivateKey:
          path: '../fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/users/User1@org1.example.com/msp/keystore/priv_sk'
        clientSignedCert:
          path: '../fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/users/User1@org1.example.com/msp/signcerts/User1@org1.example.com-cert.pem'
    connectionProfile:
      path: '../fabric-samples/test-network/organizations/peerOrganizations/org1.example.com/connection-org1.yaml'
      discover: true
  - mspid: Org2MSP
    identities:
      certificates:
      - name: 'user1.org2.example.com'
        clientPrivateKey:
          path: '../fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/users/User1@org2.example.com/msp/keystore/priv_sk'
        clientSignedCert:
          path: '../fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/users/User1@org2.example.com/msp/signcerts/User1@org2.example.com-cert.pem'
    connectionProfile:
      path: '../fabric-samples/test-network/organizations/peerOrganizations/org2.example.com/connection-org2.yaml'
      discover: true
~
~
~
~
~
~
"networks/networkConfig.yaml" 35L, 1438C                    1,1           모두
```

Caliper측정을 위해 네트워크 설정파일을 작성함. msps 항목에 Org1MSP, Org2MSP 추가함.



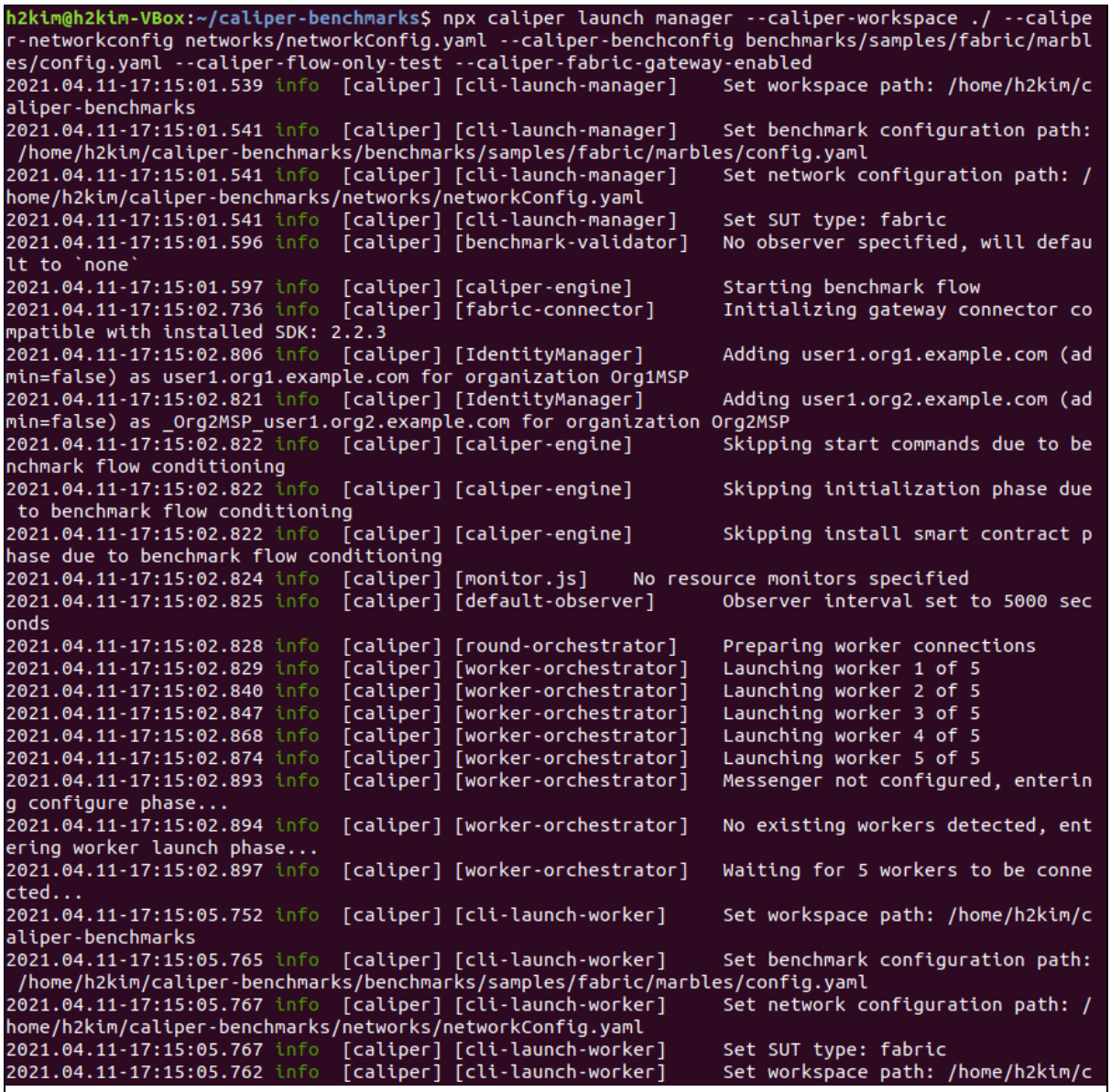
benchmarks/samples/fabric/marbles/config.yaml에 초기화 함수와 읽어오는 함수 파일 위치를 설정함.

```
17 const { WorkloadModuleBase } = require('@hyperledger/caliper-core');
18
19 const colors = ['red', 'blue', 'green', 'black', 'white', 'pink', 'rainbow'];
20 const owners = ['Alice', 'Bob', 'Claire', 'David'];
21
22 /**
23  * Workload module for the benchmark round.
24  */
25 class InitWorkload extends WorkloadModuleBase {
26     /**
27      * Initializes the workload module instance.
28      */
29     constructor() {
30         super();
31         this.txIndex = 0;
32     }
33
34     /**
35      * Assemble TXs for the round.
36      * @return {Promise<TxStatus[]>}
37      */
38     async submitTransaction() {
39         this.txIndex++;
40         let marbleName = 'marble_' + this.txIndex.toString() + '_' + this.workerIndex.toStrin
   g();
41         let marbleColor = colors[this.txIndex % colors.length];
42         let marbleSize = (((this.txIndex % 10) + 1) * 10).toString(); // [10, 100]
43         let marbleOwner = owners[this.txIndex % owners.length];
44
45         const args = {
46             contractId: 'marbles',
47             contractVersion: 'v1',
48             contractFunction: 'initMarble',
49             invokerIdentity: 'user1.org1.example.com',
50             contractArguments: [marbleName, marbleColor, marbleSize, marbleOwner],
51             timeout: 30
52         };
53
54         await this.sutAdapter.sendRequests(args);
55     }
56 }
57
58 /**
59  * Create a new instance of the workload module.
60  * @return {WorkloadModuleInterface}
61  */
62 function createWorkloadModule() {
                                                                    62,1          80%
```

Init.js 파일에 성능 체크하는 submitTransaction함수를 작성함. 성능 측정을 위한 매개변수에 노드별 측정을 위해 invokerIdentity 변수에 'user1.org1.example.com 값을 추가함.

```
'use strict';

const { WorkloadModuleBase } = require('@hyperledger/caliper-core');

const owners = ['Alice', 'Bob', 'Claire', 'David'];

/**
 * Workload module for the benchmark round.
 */
class QueryWorkload extends WorkloadModuleBase {
    /**
     * Initializes the workload module instance.
     */
    constructor() {
        super();
        this.txIndex = 0;
    }

    /**
     * Assemble TXs for the round.
     * @return {Promise<TxStatus[]>}
     */
    async submitTransaction() {
        this.txIndex++;
        let marbleOwner = owners[this.txIndex % owners.length];
        const args = {
            contractId: 'marbles',
            contractVersion: 'v1',
            contractFunction: 'queryMarblesByOwner',
            invokerIdentity: '_Org2MSP_user1.org2.example.com',
            contractArguments: [marbleOwner],
            timeout: 120,
            readOnly: true
        };

        await this.sutAdapter.sendRequests(args);
    }
}

/**
 * Create a new instance of the workload module.
 * @return {WorkloadModuleInterface}
 */
function createWorkloadModule() {
    return new QueryWorkload();
}

                                                    61,0-1        93%
```

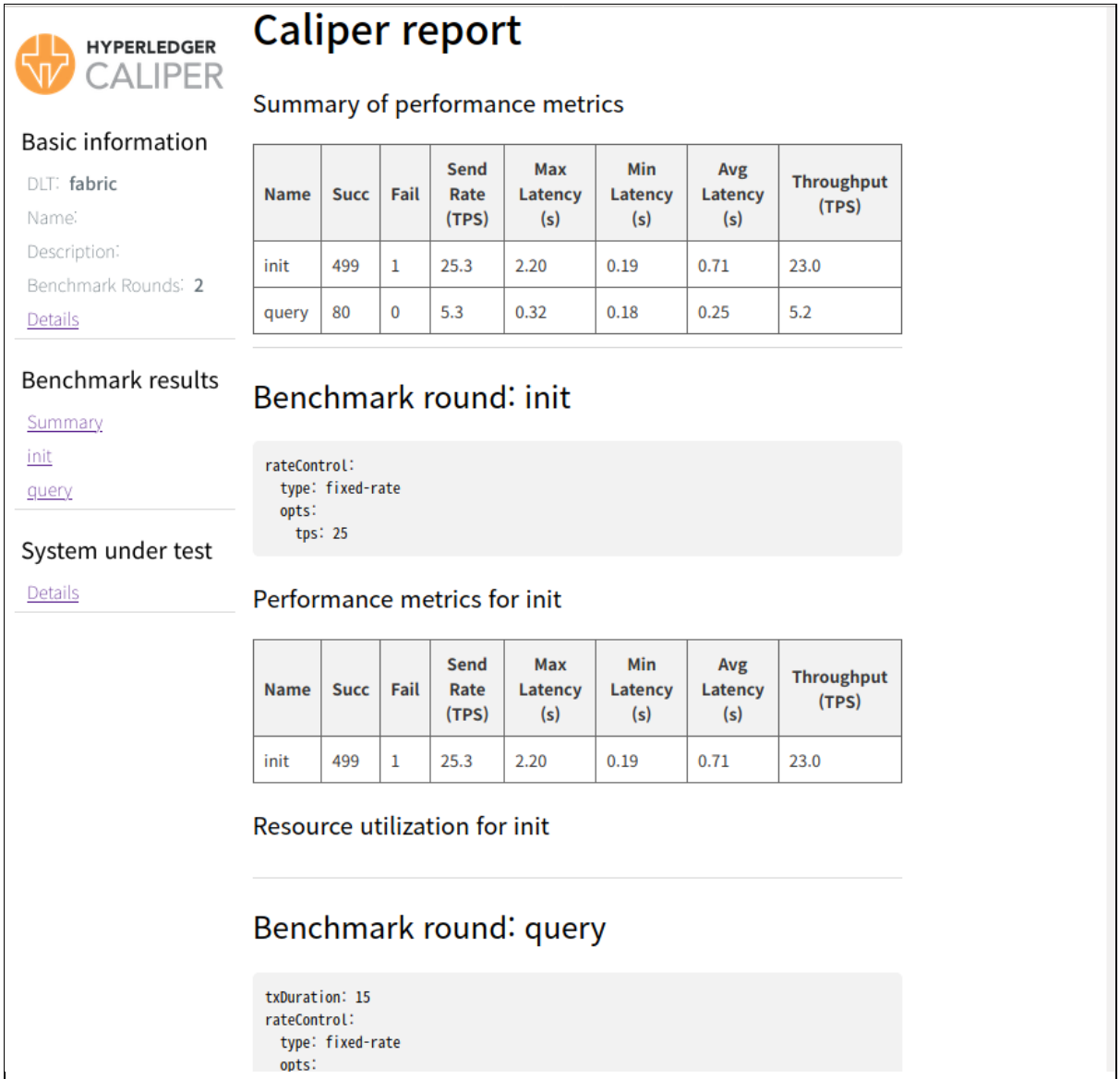
query.js에는 위와 같이 queryMarblesByOwner 함수를 사용하고 init.js다르게 invokerIdentity에는 '_Org2MSP_user1.org2.example.com' 값을 대입함.



$npx caliper launch manager –caliper-workspace ./ --caliper-networkconfig networks/networkconfig.yaml –caliper-benchconfig benchmarks/samples/fabric/marbles/config.yaml –caliper-flow-only-test –caliper-fabric-gateway-enabled 명령어를 사용하여 성능 측정을 실행함.

```
2021.04.11-17:16:12.583 info  [caliper] [report-builder]        ### Test result ###
2021.04.11-17:16:12.585 info  [caliper] [report-builder]
+-------+------+------+-----------------+-----------------+-----------------+-----------------+--
---------------+
| Name  | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | T
hroughput (TPS) |
|-------|------|------|-----------------|-----------------|-----------------|-----------------|--
---------------|
| query | 80   | 0    | 5.3             | 0.32            | 0.18            | 0.25            | 5
.2              |
+-------+------+------+-----------------+-----------------+-----------------+-----------------+--
---------------+

2021.04.11-17:16:12.590 info  [caliper] [round-orchestrator]    Finished round 2 (query) in 15.32
2 seconds
2021.04.11-17:16:12.590 info  [caliper] [monitor.js]    Stopping all monitors
2021.04.11-17:16:12.590 info  [caliper] [report-builder]        ### All test results ###
2021.04.11-17:16:12.592 info  [caliper] [report-builder]
+-------+------+------+-----------------+-----------------+-----------------+-----------------+--
---------------+
| Name  | Succ | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | T
hroughput (TPS) |
|-------|------|------|-----------------|-----------------|-----------------|-----------------|--
---------------|
| init  | 499  | 1    | 25.3            | 2.20            | 0.19            | 0.71            | 2
3.0             |
|-------|------|------|-----------------|-----------------|-----------------|-----------------|--
---------------|
| query | 80   | 0    | 5.3             | 0.32            | 0.18            | 0.25            | 5
.2              |
+-------+------+------+-----------------+-----------------+-----------------+-----------------+--
---------------+

2021.04.11-17:16:12.652 info  [caliper] [report-builder]        Generated report with path /home/
h2kim/caliper-benchmarks/report.html
2021.04.11-17:16:12.655 info  [caliper] [monitor.js]    Stopping all monitors
2021.04.11-17:16:12.656 info  [caliper] [worker-orchestrator]   Sending exit message to connected
 workers
2021.04.11-17:16:12.656 info  [caliper] [worker-message-handler]        Worker#0 is exiting
2021.04.11-17:16:12.657 info  [caliper] [worker-message-handler]        Worker#1 is exiting
2021.04.11-17:16:12.658 info  [caliper] [worker-message-handler]        Worker#2 is exiting
2021.04.11-17:16:12.659 info  [caliper] [worker-message-handler]        Worker#3 is exiting
2021.04.11-17:16:12.660 info  [caliper] [worker-message-handler]        Worker#4 is exiting
2021.04.11-17:16:12.661 info  [caliper] [round-orchestrator]    Benchmark finished in 59.517 seco
nds. Total rounds: 2. Successful rounds: 2. Failed rounds: 0.
2021.04.11-17:16:12.661 info  [caliper] [caliper-engine]        Skipping end command due to bench
mark flow conditioning
2021.04.11-17:16:12.661 info  [caliper] [cli-launch-manager]    Benchmark successfully finished
```

peer0에서는 init.js 성능 테스트를 하고 peer1에서는 query.js 성능 측정을 하고 결과를 도출함.



HYPERLEDGER CALIPER

Basic information

DLT: fabric

Name:

Description:

Benchmark Rounds: 2

Details

Benchmark results

Summary

init

query

System under test

Details

Caliper report

Summary of performance metrics

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
init	499	1	25.3	2.20	0.19	0.71	23.0
query	80	0	5.3	0.32	0.18	0.25	5.2

Benchmark round: init

```
rateControl:
  type: fixed-rate
  opts:
    tps: 25
```

Performance metrics for init

Name	Succ	Fail	Send Rate (TPS)	Max Latency (s)	Min Latency (s)	Avg Latency (s)	Throughput (TPS)
init	499	1	25.3	2.20	0.19	0.71	23.0

Resource utilization for init

Benchmark round: query

```
txDuration: 15
rateControl:
  type: fixed-rate
  opts:
```

$ firefox report.html 명령어를 입력하면 성능측정한 값과 설정한 정보를 확인할 수 있음.

하이퍼레저 패브릭을 이용한 전자출입명부를 개발을 위해 private-data를 이용하기로 계획하고 비교를 위해 공기 키 기반구조(PKI)를 이용하여 성능측정 비교를 하기로 계획을 세움.

```
h2kim@h2kim-VBox:~/fabric-samples/test-network$ ./network.sh up createChannel -ca -s couchdb
Creating channel 'mychannel'.
If network is not up, starting nodes with CLI timeout of '5' tries and CLI delay of '3' seconds
and using database 'couchdb with crypto from 'Certificate Authorities'
Bringing up network
LOCAL_VERSION=2.2.0
DOCKER_IMAGE_VERSION=2.2.0
CA_LOCAL_VERSION=1.4.8
CA_DOCKER_IMAGE_VERSION=1.4.8
Generate certificates using Fabric CA's
Creating network "net_test" with the default driver
Creating ca_orderer ...
Creating ca_org2 ...
Creating ca_org1 ...
Creating ca_orderer
Creating ca_org1
Creating ca_org2
```

./network.sh up createChannel -ca -s couchdb 명령어를 사용하여 ca노드를 사용함.

```
h2kim@h2kim-VBox:~/fabric-samples/test-network$ ./network.sh deployCC -ccn private -ccp ../asset
-transfer-private-data/chaincode-go/ -ccl go -ccep "OR('Org1MSP.peer','Org2MSP.peer')" -cccg ../
asset-transfer-private-data/chaincode-go/collections_config.json
deploying chaincode on channel 'mychannel'
executing with the following
- CHANNEL_NAME: mychannel
- CC_NAME: private
- CC_SRC_PATH: ../asset-transfer-private-data/chaincode-go/
- CC_SRC_LANGUAGE: go
- CC_VERSION: 1.0
- CC_SEQUENCE: 1
- CC_END_POLICY: OR('Org1MSP.peer','Org2MSP.peer')
- CC_COLL_CONFIG: ../asset-transfer-private-data/chaincode-go/collections_config.json
- CC_INIT_FCN: NA
- DELAY: 3
- MAX_RETRY: 5
- VERBOSE: false
Vendoring Go dependencies at ../asset-transfer-private-data/chaincode-go/
~/fabric-samples/asset-transfer-private-data/chaincode-go ~/fabric-samples/test-network
~/fabric-samples/test-network
Finished vendoring Go dependencies
Using organization 1
+ peer lifecycle chaincode package private.tar.gz --path ../asset-transfer-private-data/chaincod
e-go/ --lang golang --label private_1.0
```

./network.sh deployCC -ccn private -ccp ../asset-transfer-private-data/chaincode-go/ -ccl go -ccep "OR('Org1MSP.peer','Org2MSP.peer')" -cccg ../asset-transfer-private-data/chaincode-go/collections_config.json 접근 권한과 변수를 설정함.

```
h2kim@h2kim-VBox:~/fabric-samples/asset-transfer-private-data/application-javascript$ node app.j
s

--> Fabric client user & Gateway init: Using Org1 identity to Org1 Peer
Loaded the network configuration located at /home/h2kim/fabric-samples/test-network/organization
s/peerOrganizations/org1.example.com/connection-org1.json
Built a CA Client named ca-org1
Built a file system wallet at /home/h2kim/fabric-samples/asset-transfer-private-data/application
-javascript/wallet/org1
An identity for the admin user already exists in the wallet

--> Fabric client user & Gateway init: Using Org2 identity to Org2 Peer
Loaded the network configuration located at /home/h2kim/fabric-samples/test-network/organization
s/peerOrganizations/org2.example.com/connection-org2.json
Built a CA Client named ca-org2
Built a file system wallet at /home/h2kim/fabric-samples/asset-transfer-private-data/application
-javascript/wallet/org2
An identity for the admin user already exists in the wallet

*************** As Org1 Client ***************
Adding Assets to work with:
--> Submit Transaction: CreateAsset asset1

--> Submit Transaction: CreateAsset asset2
```

$ node app.js 명령어를 사용해 asset-transfer-private-data 예제 프로그램을 실행함.

```
--> Evaluate Transaction: ReadAsset asset1
  result: {
  "objectType": "ValuableAsset",
  "assetID": "asset1",
  "color": "green",
  "size": 20,
  "owner": "eDUwOTo6Q049YXBwVXNlcjIsT1U9Y2xpZW50K09VPW9yZzIrT1U9ZGVwYXJ0bWVudDE6OkNOPWNhLm9yZzIu
ZXhhbXBsZS5jb20sTz1vcmcyLmV4YW1wbGUuY29tLEw9SHVyc2xleSxTVD1IYW1wc2hpcmUsQz1VSw=="
}
  Asset owner: x509::CN=appUser2,OU=client+OU=org2+OU=department1::CN=ca.org2.example.com,O=org2
.example.com,L=Hursley,ST=Hampshire,C=UK

--> Evaluate Transaction: ReadAssetPrivateDetails
  result:

--> Evaluate Transaction: ReadAsset asset2
  result: {
  "objectType": "ValuableAsset",
  "assetID": "asset2",
  "color": "blue",
  "size": 35,
  "owner": "eDUwOTo6Q049YXBwVXNlcjEsT1U9Y2xpZW50K09VPW9yZzErT1U9ZGVwYXJ0bWVudDE6OkNOPWNhLm9yZzEu
ZXhhbXBsZS5jb20sTz1vcmcxLmV4YW1wbGUuY29tLEw9RHVyaGFtLFNUPU5vcnRoIENhcm9saW5hLEM9VVM="
}

********* Demo deleting asset **************
--> Attempt Transaction: as Org2 DeleteAsset asset1
--> Attempt Transaction: as Org2 DeleteAsset asset2
2021-04-18T04:24:53.653Z - error: [Transaction]: Error: No valid responses from any peers. Error
s:
    peer=peer0.org2.example.com:9051, status=500, message=asset not found in owner's private Col
lection Org2MSPPrivateCollection: asset2
  Successfully caught the error:
    Error: No valid responses from any peers. Errors:
    peer=peer0.org2.example.com:9051, status=500, message=asset not found in owner's private Col
lection Org2MSPPrivateCollection: asset2
```

Asset를 생성하고 PrivateData를 읽고 소유권을 변환하고 삭제하는 과정을 보여줌.

3. 차주 계획
- Hyperledger fabric asset-transfer-private-data를 참고하여 전자출입명부 개념적 설계 - 공개키 기반 구조를 체인코드 적용방안 구상 후 테스트
4. 참고 웹페이지
[1] https://hyperledger-fabric.readthedocs.io/en/release-2.2/private-data/private-data.html [2] https://hyperledger-fabric.readthedocs.io/en/release-2.2/private_data_tutorial.html [3] https://hyperledger-fabric.readthedocs.io/en/release-2.2/membership/membership.html