**General description of the POA ethereum tests**

Network performance was estimated by mass transaction generation in 2 contracts:

**addchunk** (add entry to array of structures)

**counter** (simple counter)

using the set of scripts described in the scripts section. 2 types of networks were checked:

vcnet5 - (5 sec block\_period, 4700000 GasLimit)

vcnet2 - (1 sec block\_period, 23000000 GasLimit)

The task of the tests was to get the maximum possible tps. In conclusion, theoretical estimates are obtained for the maximum tps depending on the network parameters and the transaction.

The tests varied the number and type of generators for each of the contracts and for each of the networks.

The results were recorded in log files by means of the web3.py library, which were subsequently processed locally in matlab.

Each line in the file logs is:

mined tx pending tx queued tx

recorded every 10 seconds

Significant results are presented below in the report.

**Tests for poa vcnet 5 (5 sec block creation)**

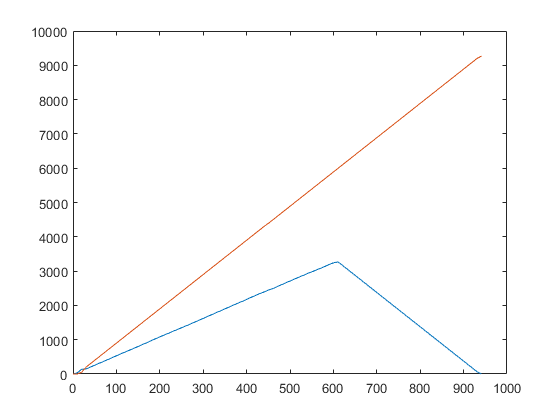
**16 transactions test**

transaction generator produce effectively ~16 ts/sec

transactions are equal and store in contract vcnet 5

generator time - 600 sec

result we can see on the plot



red - mined ts (stored in contract)

blue - pending ts

Process complete. All working good

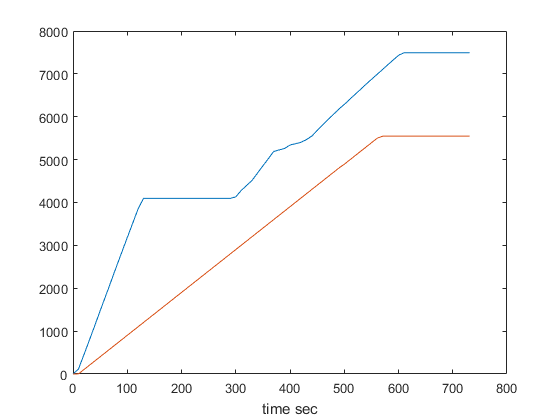
**34 transactions test**

two transaction generators from different nodes produce effectively ~17 ts/sec each

transactions are equal and store in contract vcnet 5

generator time - 600 sec

result we can see on the plot



red - mined ts (stored in contract)

blue - pending ts

Processes not complete. the local geth clients are stopped and need to reboot .

**Testing and selecting POA Network parameters**

vcnet2 tests:

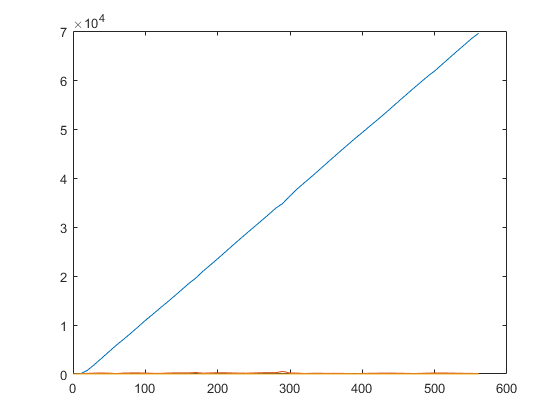
with another contract (counter)

single-threaded transaction generator (python)

multithreaded transaction generator (python)

~ 10 - 12 tests were performed in order to get the maximum tps for this configuration. The best result for three generators (two in a network one local)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| test | ins1 txs (5 w 5proc) | eur1 txs (5w 1 proc) | target | time | net | generator txs | TPS | Pending | queued | effective time | txs |
| 5 | 50000 | 20000 | counter | 560 | vcnet2 | multi thread | 124.0107 | 177.4386 | 0 | 561 | 70000 |



tps chart

Y label transactions

X label time (sec)

blue line - mined

orange line - pending

POA Parameters and TPS

**absolute math limit for POA Ethereum calculated as**:

tps=gaslimit/Gtransaction\*block\_period

for vcnet it’s

tps=4 700 000/21000\*5=44.7

for vcnet2 it’s

tps=23 000 000/21000=1095

**The next calculations for real network strongly depend on:**

1. cost of transaction in accordance with Appendix G yellow paper of ethereum
2. Specific of software client (geth)
3. Physical state of computers and network

**For example**

if we sent transaction like chunk registration in vcnet2 (addchunk contract, 156к gas for transaction) that similar to use in videocoin

the maximum tps will be 23000000/156000 = 147.

if we sent very simple transaction like counter contract (27k gas for transaction)

max tps will be 23000000/ 27000 = 851

**A set of python, bash scripts and code for ethereum is attached:**

**benchmark\_tr.py** Description: generator TX to vcnet using Contract AddChunk. Single thread. Call a functions.addChunkInfo with 20 ms delay in 600s.

**benchmark\_tr2\_thread.py** Description: Advanced generator TX to vcnet2 using Contract AddChunk. Multithread. Call N:10000 function.addChunkInfo() via multi-threading queue with M workers.

**benchmark\_tr2\_counter.py** Description: generator TX to vcnet2 using Contract Counter. Single thread. Call a function incrementCounter() with 2 ms delay in 600s.

**benchmark\_tr2\_thread\_counter.py** Description: Advanced generator TX to vcnet2 using Contract Counter. Multithread. Call N:10000 function incrementCounter() via multi-threading queue with M:2-5 workers.

**benchmark\_tr2\_send.py** Description: generator TX to vcnet2 without smart-contract. Single thread. Send a N wei from eth.coinbase to pre-defined eth-node (vcnet2) using RAW transaction.

**txpool2.py** 2> log.file Description: Measurer state txpool in network (via w3 RPC) and counter mined transaction in blockchain per time. Worked together with generator. log.file source for plot.

**benchmark\_tr2\_counter.sh** start some generator at the same time

**benchmark\_tr2\_counter\_thread.sh** start some generator at the same time

**benchmark\_tr2\_send.sh** start some generator at the same time

AddChunk.json abi of smart-contract

Counter.json abi of samart-contract

All files are here:

https://drive.google.com/drive/folders/1ELnVJoOjt7-pQBEqGGNixdTcls68X5WJ?usp=sharin

g