BitShares Testnet Stress-test

Fabian Schuh BitShares Europe, bitshares.eu[*]
Erlangen (Bavaria), Germany
fabian@bitshares.eu

***Abstract*—BitShares 2.0 is an industrial-grade decentralized platform built for high-performance financial smart contracts. In order to show its capabilities in the field, we have conducted a stress test on the public testnet. The testnet has been deployed with the same code base that is used for the BitShares network and has nodes around the globe. Only margin changes have been made to distinguish the testnet from the productive network, properly. A multi-phase stress-test has been proposed and accepted that modifies the maximum transaction size, maximum block size and block confirmation times in the live network during the stress test. Validators have been kept up to date by means of stake-weighted voting [1].**

1 Introduction

The Graphene Platform has been developed by Cryptonomex specifically for BitShares. It has undergone many changes and has been enhanced to stay on top of blockchain technology. One of its public forks, BitShares, has been publicly traded for over 2 years and has yet to show its maximum potential with respect to throughput. For this reasons, we have deployed a testnet, that uses the very same technology that BitShares is built on, with the specific purpose of testing algorithms, implementations, and also scalability.

Here, the term *scalability* refers to the amount of transactions that can be applied to the blockchain at scale. Several factors need to be taken into account to correctly interpret any results obtained through testing. Among these are the specification of the deployed servers (CPU/RAM) that produce the blocks (validators), the interconnectivity of the nodes in the Peer-2-Peer network, the round-trip times and latency between nodes, as well as the geographical distance between nodes.

The number of validators, furthermore, does not affect the throughput as long as all active validators can keep up with the requirements of the network. All active validators are treated equally and are given a slot to produce their blocks in each round [1]. Increasing the number of validators (a.k.a. witnesses) comes with an increased robustness against server failures yet also results in a longer duration to reach transaction finality [1] which describes the absolute irreversibility of transactions after 2/3 of all validators have approved a given block and its transactions.

The goal of the public stress-test performed on the graphene-based testnet was to identify the limiting factors and bottlenecks at scale with multiple parties involved. We furthermore wanted to demonstrate scalability of current state of the art blockchain technologies.

2 Testnet Setup

We have deployed a separated blockchain for the public testnet that has been launched in January 2016. That blockchain was open to the public and every new account has been donated some of the core assets called `TEST`. By the time of the stress-test, the blockchain had over 6,800,000 blocks, and over 25,000,000 operations processed.

2.1 Software

The software for the testnet backend has been maintained by BitShares Europe and is available for review on github.com. It is an almost identical fork of the Graphene code base that BitShares is built on with only marginal changes to accommodate the separated blockchain.

The tools that have been developed for the stress-test and the analysis are mostly available in separate repositories as well.

2.2 Validators

The validators' job is to collect unconfirmed and pending transactions as received through a Peer-2-Peer network, verify and validate them and publish your approval over those transactions in form of a signed block that carries those transactions.

Given that the Graphene testnet also uses Delegated Proof of Stake (DPOS) as its consensus mechanism, the validators (a.k.a. block producers) can be voted in and out by means of the stake-based governance system [1]. This allowed us to 1. modify the number of validators 2. replace failing validators by standby validators 3. pick validators that are closer geographically (if location is known) every maintenance interval. This interval is used to tally all votes made since the previous maintenance block and is set to $5\,\mathrm{min}$ on the test-network.

At the beginning of the stress-test, we will have these validators active:

blckchnd-x Intel Xeon E3, 32GB RAM, bare bone, Germany

[*] This work was supported by ChainSquad GmbH, BitShares-Munich IVS and honorable members of the bitsharestalk.org community.

blckchnd-test Intel Xeon E3, 32GB RAM, bare bone, Germany

jim.witness1 Intel XeonÂ® E5, 28-56GB RAM, Azure, South Korea

smailer-5 Intel Xeon E3, 32GB RAM, Germany

init0 Intel Core i7, 32GB RAM, bare bone, Germany

init2 Intel Core i7, 32GB RAM, bare bone, Germany

lafona2 Intel Avoton, 32GB RAM, France

delegate.ihashfury Intel Atom C2750, 32GB RAM, bare bone, France

f0x Intel Xeon E5, 56GB RAM, Azure, USA

alpha-jpn Intel Xeon E5, 56GB RAM, Azure, Japan

bravo-bra Intel Xeon E5, 56GB RAM, Azure, Brasil

charlie-usa Intel Xeon E5, 56GB RAM, Azure, USA

delta-gbr Intel Xeon E5, 56GB RAM, Azure, UK

rngl4b Intel Xeon E5, 32GB RAM, bare bone, Luxembourg

taconator-witness Intel Xeon E5, 24GB RAM, Switzerland

arthur-devling Intel Xeon E5, 39GB RAM, France

fr-blockpay Intel Xeon E5, 256GB RAM, Germany

de-blockpay Intel Xeon E5, 128GB RAM, France

2.3 Reference Full Node and Seed Node

A bare bone reference node is provided by BitShares Europe that allows participants to interface with the Peer-2-Peer network without running their own full-nodes. The endpoint is available at `wss://node.testnet.bitshares.eu` and is powered by an Intel Core i7 with 64GB of RAM which carries two fully balanced full nodes to deal with the traffic.

A frontend-wallet is provided under `https://testnet.bitshares.eu`.

3 Limiting Factors

In this section we briefly discuss the limiting factors we have identified prior to the stress-test. For the sake of proper description, we first need to look into the distinction of *blocks*, *transactions*, and *operations*.

Transactions and Operations In contrast to other blockchains, Graphene-based blockchain *explicitly* allow to bundle multiple *operations* into a single transaction. Similar to Bitcoin allowing to bundle multiple outputs, but with the option to execute different smart contracts offered by the blockchain sequentially. The easiest application is similar to Bitcoin's multi-send feature where a single user sends bitcoins to many addresses. In the case of Graphene/BitShares, the user would also sign a single transaction with a single signature but the transaction would carry many `transfer` operations. Additionally, a user may put a combination of `trade`/`transfer`/`borrow` operations (or any other) into a single transaction to ensure that they are executed sequentially.

As can be seen, the main difference between bundling many transactions into a block and bundling many operations into a transactions is that in the latter case, the operations are guaranteed to be executed sequentially and that only a single authorization (e.g. signature) needs to be validated and verified.

This distinction is important as our test results will clearly distinguish between the throughput of *operations* and the throughput of *transactions*.

Signature Verification Similar to other blockchain technologies (i.e. Bitcoin), a single block may contain multiple transactions. Each transaction is signed by one or multiple entities and thus requires the validators to perform elliptic curve signature verification as well as public key recovery for authorization.

From a scalability point of view, the limiting factors are the time and resources that are required to validate a signature for a given transaction and the time and resources that are required to append a transactions to the blockchain and execute its corresponding smart contract/operation.

Since the Graphene technology is built such that the signature validation can be done independent of the blockchain's current database state, the signature verification and key recovery can be trivially parallelized by means of a cluster or the use of a graphics processing unit (GPU). However, as the required software for this parallel verification was not available at the time of the stress-test, we expect the throughput to be significantly higher once we can be validated on a GPU.

For this reason, and because we want to see the limiting factors when applying validated transactions to the blockchain, parts of our test scenario are focused around producing transactions that carry hundreds of *transfer* operations with just a single signature to be verified.

Connectivity Obviously, a globally distributed network represents the worst case scenario when it comes to latency between nodes limited mostly by the speed of light and the size of the planet. A network that is hosted locally can result in better overall performance but does not offer the same robustness and redundancy as a global network.

Memory Requirements In previous tests, we have identified a significantly increasing need for memory by the blockchain providing back-end software. This was mostly caused by so called *plugins* that take care of historic events, such as account transactions history or the trading histories. Those plugins kept their data in memory for performance reasons. Once we disabled these plugins, the validating software only required a few hundred Megabytes of memory.

Transaction Production In order to stress-test a network, we need to produce actual stress which is not easy to achieve at the scale that we needed for our tests. Furthermore, we wanted to simulate a more or less realistic scenario in which multiple independent parties distributed on the globe produce transactions, each transaction trying to make it into the next block.

For this reasons, we have asked the BitSharesTalk community to participate in the stress-testing not only by means of providng validation nodes across the planet, but also to produce the required stress.

Peer-2-Peer Networking The original codebase of Graphene has a hard-coded limit in its Peer-2-Peer networking code (not part of the blockchain consensus protocol), that prevented us from reaching more than 1000 tps due to restriction in the amount of transactions being obtained through the Peer-2-Peer network. After increasing this limit way beyond what we wanted to achieve in our stress-test, we allowed the validating nodes to obtain up to thousands of transactions per second from the Peer-2-Peer network.

However, the Peer-2-Peer networking code has been optimized for a block interval of about 10 s and thus leaves a lot of room for optimizations when talking about high-throughput at low latencies.

4 Phases of the Stress-Test

Since we use Graphene as our underlying technology, most blockchain parameters can be changed in real-time with no need for protocol upgrades or hard forks.

In contrast to the public BitShares network, the majority of the stake in the public testnet is owned by BitShares Europe which allowed us to change the parameters without the need to coordinate with other stakeholders.

This allowed us to setup a multiple phase stress-test that went through a set of different blockchain parameters to gradually increase the amount of processed transactions on the blockchain.

4.1 Modifying Blockchain Parameters

For our stress-test, we decided to focus on three blockchain parameters only:

Max. block size This limit allowed us to modify the size of the blocks that are considered valid by the network. For our test, the size will be between 1 MB and 10 MB. The limiting factor is the supported data rate and connectivity of the validating nodes since blocks need to be produced and broadcasted within a certain time interval. A broadcast block needs to be received by the subsequent validator in time, otherwise the subsequent block cannot be linked to the expected previous block properly.

Max. transaction size Each *block* can carry multiple *transaction* and, in contrast to many other blockchain technologies, transactions on Graphene-based blockchains can carry multiple *operations*. In our test, we assume that most of the operations are simple *transfers* of size 22 bytes. Together with the transaction header, a simple single-transfer (unsigned) transaction is 36 bytes large. During the stress-test, we allowed between 50 and 1000 transfer operations to be bundled into a single transaction.

Block confirmation time The block confirmation time is the expected time between blocks. At the beginning we started with a 3 s block interval and reduced it down to 1 s at which point we expected the network to lose its robustness because of its global distribution, and because round-trip durations together with the need to transmit non-empty blocks might require more time.

These parameters can be changed by a committee decision that is manifested through a dynamic multi-signature account. The members of the committee are voted on by the shareholders and collectively (with 50%+1), the committee can modify blockchain parameters by a single transaction. The procedure requires sufficient committee members to approve/co-sign a transaction that proposes to change parameters. After approval, the parameters are changed during the next maintenance block.

4.2 Tested Parameter Sets

The stress-test contained seven phases. As can be seen, the last two phases changed the block confirmation time and drastically reduce the maximum block size to ensure that blocks can at least be transferred in a timely manner. Whether blocks are propagated to the witnesses in time is is outside of direct control of the testers since we are running on a public and globally distributed Peer-2-Peer network.

phase	max. tx size	max. block size	block interval
1	1 kb	1 MB	3 s
2	1 kb	5 MB	3 s
3	10 kb	5 MB	3 s
4	100 kb	5 MB	3 s
5	100 kb	1 MB	2 s
6	100 kb	1 MB	10 s
7	100 kb	1 MB	1 s

5 Results

5.1 Processed Transactions/Operations

During our 4h-stress-test between 2pm and 6pm, the blockchain processed 4,011,110 transactions and 16,398,274 operations in 4233 blocks. We varied three blockchain parameters including block confirmation times between 1 and 10 s.

5.2 Transaction/Operation Throughput

After the stress-test, when the transactions and operations had been added to the blockchain, we reviewed the blockchain offline. Given that the transactions and operations are recorded on the blockchain, we started by investigatng the overall results first.

As can be seen from fig. 1 (top), there are multiple periods of activity. Clearly, most of the transactions started being broadcast from 15:00 going forward with an average of approximately 1000 transactionsperblock afterwards. We also notice that the throughput was not constant during the stress-test which can be explained by the transactions requiring some time to propagate through the Peer-2-Peer network.

We can further see in fig. 1 (bottom), that the participants have been constantly trying to modify their spam production method (increaseing number of transactions vs. bundling more operations per transaction) until a maximum of processed operations per block was reached at about 16:40. Afterwards, the participants were asked to increase the number of created transactions instead which is why the processed operations per block decreased later on.

Most interestingly, we observed several short peaks of processed transactions and operations during the whole stress-test. As it turned out, the reasons of those peaks can be explained by a single participant who decided to use his block validating node

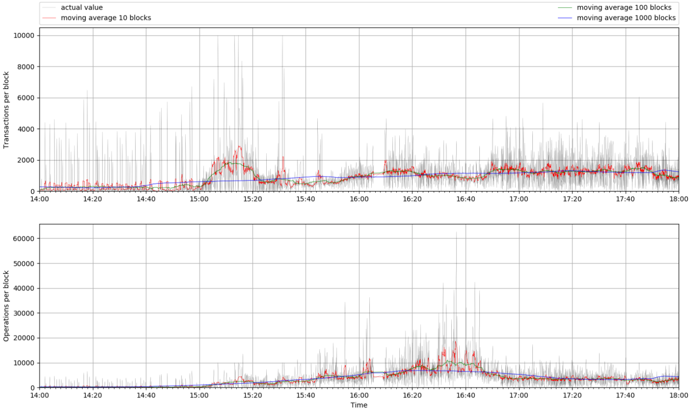

Figure 1: Overview of the throughput ops/s and txs/s during the whole stress-test

to produce transactions. By this, he skipped the propagation of transactions through the Peer-2-Peer network which confirms to us that the networking code represents one of the bottlenecks.

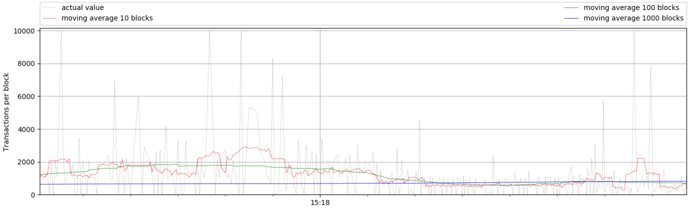

Figure 2: Max ops/s during the stress-test

In fig. 2, we have zoomed into a period of rather high throughput (i.e. 2500 transactionsperblock). Here, we still see the peaks produced by skipping the networking.

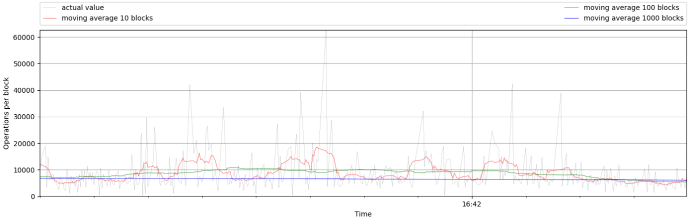

Figure 3: Max ops/s during the stress-test

Finally, the results in fig. 3 show that the blockchain was able to process an average of over 8000 operationsperblock with quite high peaks and an irregular throughput per block.

The irregularity can be resolved by deploying similar hardware that can all process the same amount of transactions/operations. We were able to confirm this behavior later on by replacing all validators except for one, which greatly reduces the variance of the throughput as can be seen in fig. 4. Here, the stress has been produce by a single node connected directly to the validating node while another single validator was only connected through the Peer-2-Peer network.

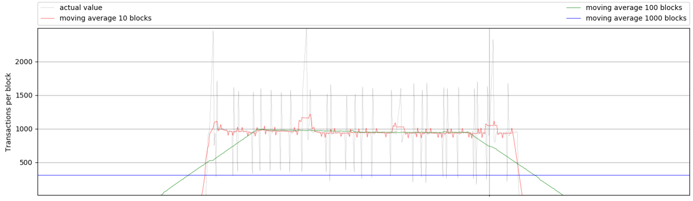

Figure 4: Max ops/s during the stress-test

5.3 Block Confirmation Times

After investigating the throughput of the blockchain, we also tested the Peer-2-Peer network and capability to deal with different block confirmation times. As a reminder, the technology tested here is capable of changing different blockchain parameters on the fly. One of these parameters is the block confirmation time which represents the time between the blocks in the synchronous consensus mechanism delegated proof-of-stake (DPOS).

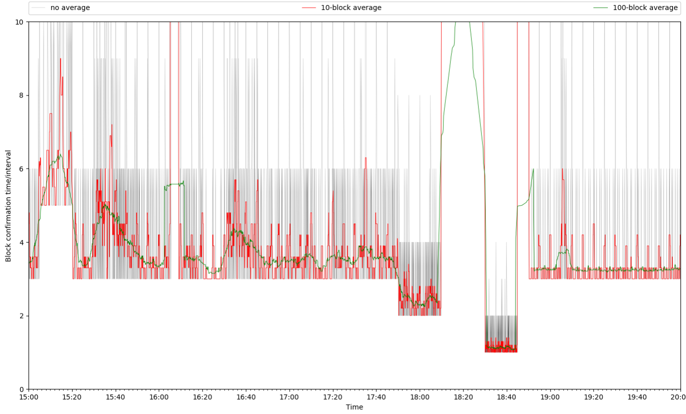

Figure 5: Max ops/s during the stress-test

In fig. 5, we can see the actual block confirmation time as tracked by the blockchain. These values have been obtained from each block header which also contains the time stamp. Little variations are to be expected due to slightly inaccurate synchronization of the local time, while large deviations indicate one or multiple validators to have missed their blocks. The graphs shown in the figure are averaged over 100 blocks to reduce the noise.

We can see that, during the stress-test, multiple different block confirmation times were tested, starting with 5 s, going down to 2 s, then up to 10 s until we decided to also test 1 s block confirmation time. Unfortunately, we observed some witnesses not producing during large parts of the stress-test which is why the averages are above the target rates most of the time. For our next stress-test we plan to be more strict in replacing witnesses when they miss block production blocks.

In **??** highlights the period when the block confirmation time was reduced to 1 s. We can see that most of the blocks were produced in time with only up to 3 validators not having produced in time (i.e. the black peaks going up to at most 4 s). This came

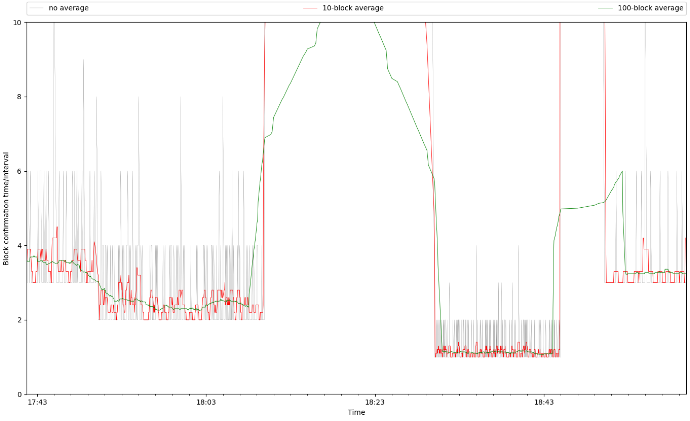

Figure 6: Max ops/s during the stress-test

as a big surprise to us as we had expected a higher block-miss rate all together.

6 Conclusions

With our testing, we have seen that the code base is much more robust than we had expected. The block creation and block validation is sufficiently fast so that we can run a blockchain with synchronous block confirmations times of 1 s without validators dropping out too much.

The Peer-2-Peer code seems to be working nicely but results show that it could be one of the bottlenecks currently preventing us from going further. The computational resources of our validators have more than enough back-off for higher throughputs but the networking code was not able to provide sufficient data (e.g. transactions) to raise it during our stress-test.

We have further identified another bottleneck with respect to transaction production. Signing transactions takes more resources and we will prepare properly for our next stress-test. This result was confirmed in a later private testnet where 36 cores have been activated during the test sequentially. The result was that the blockchain easily dealt with all of the produced transactions and the throughput increasesed accordingly, each time additional cores have been added to transaction production.

To conclude, the current software stack still has a few limitations and edges that need to be optimized in order to improve scalability further, but the foundation has been designed in such a way that it allows for even higher throughput. Our current limitations are purely in the implementation and networking aspects than in the software and protocol architecture.

7 Acknowledgements

We would like to thank every one that has participated in the stress-test for block validation, transaction generation and setup of a robust Peer-2-Peer network. We would like to further acknowledge the assistance during the preparations of the stress-test and the feedback we have received while writing this paper. Last but not least we appreciate the financial support given by each and everyone that has been running a machine for the stress-test and BitShares-Munich for providing the major API node on bitshares.eu.

References

[1] “BitShares 2.0: General Documentation,” *BitShares Whitepapers*, 2015.