Block Gas Limits vs. Transactional Throughput: A Performance Analysis of the Ubiq Platform

Whiteblock, Inc.

***Abstract*— The following report outlines a performance benchmarking analysis for the Ubiq blockchain platform under different block gas limits in terms of block time, difficulty, throughput and gas consumption. The primary purpose of these testing initiatives focus on exploring the correlation between block gas limit and transactional throughput. Testing was conducted using the Whiteblock platform with a network of 30 nodes, 10 of which conducted transactions in a round robin manner while the the remaining nodes were configured as miners. Further details concerning network architecture and topology can be found in the METHODOLOGY section. The results of these testing initiatives imply that optimal throughput was achieved at block gas limits of 40 million. Further details and data concerning results of these testing initiatives can be found in the RESULTS section.**

I. Introduction

Ethereum is an open-source, public blockchain and smart contract platform. It supports a modified version of Nakamoto consensus via transaction-based state transitions. Ubiq is an early hard fork of Geth, the Go implementation of Ethereum, which offers a unique smart contract ecosystem and operates autonomously of its predecessor. By focusing on stability in order to prioritize methodical, bug-free upgrades, Ubiq's core developers hoped to address what they identified as essential shortcomings encountered within the Ethereum platform; primarily, inconsistencies in blocktime resulting from its difficulty algorithm. This is important for enterprise applications where a constantly changing platform presents an inherent degree of technological risk.

While Ethereum's difficulty algorithm refers to the parent block in order to maintain an approximate 12-17 second blocktime, Ubiq implements a custom difficulty algorithm, aptly named *Flux*, which targets an 88-second blocktime and refers to the average blocktime of the previous 88 blocks in order to further adjust difficulty.

A low blocktime, such as that of Ethereum, has certain implications on the performance of a blockchain network, which can consecutively present particular security concerns. For example, a blocktime of 12 seconds incurs a higher uncle rate due to the presence of network latency. In addition to having a negative effect on transactional throughput, a high uncle rate also decreases network resilience against 51% attacks and further increases the likelihood of centralization within the network.

With this in mind, the hypothesis prior to conducting this test series held that the 88-second blocktime in Ubiq would result in a far lower uncle rate without sacrificing transactional throughput, even with block gas limits far exceeding default values.

This following presents the performance benchmark analysis for Ubiq under different block gas limits in terms of block time, difficulty, throughput and gas use. This paper also evaluates the correlation between block gas limit and transaction throughput within the Ubiq network.

II. Methodology

A. Network Configuration

The following testing initiatives were conducted using the Whiteblock blockchain testing framework to provision a private test network consisting of 30 independent nodes. Each of these 30 nodes were configured with equal computational resources and network conditions. Each node stored a wallet which was pre-allocated with 100 UBQ tokens in the custom genesis file. Within each test series, initial gas limits were set within the custom genesis file in accordance with the gas limit tested within that particular series.

Node 1 through node 10 acted as transaction nodes while the remaining 20 nodes were configured as miners. Further details concerning transactional flow are presented in the following section. Fig. 1 illustrates the network topology.

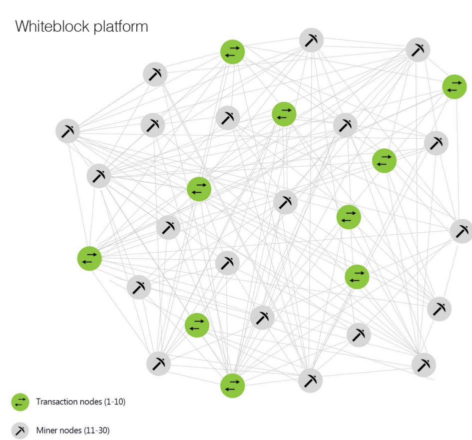


Fig. 1: Network Setup

B. Transaction Flow

In order to avoid the depletion of wallet balances, the transactional flow was automated in an round-robin manner using transaction values which were selected at random. A single node between 1 and 10 was also randomly selected to act as the transaction node and each node from the transaction group took turns sending transactions as illustrated in the figure below:

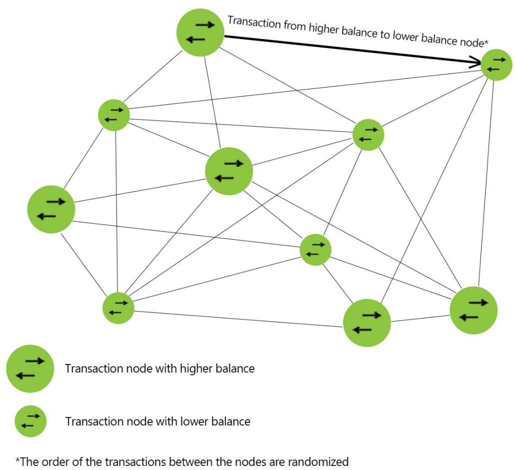


Fig. 2: Transaction flow

C. Testing Parameters (Ubiq vs Ethereum)

In order to objectively analyze the performance of the Ubiq client, the following testing parameters were selected and defined as such:

1) Total Transactions: The number of transactions included within the current block.
2) Difficulty: The total mining difficulty up to current block.
3) Block Time: The time difference between current block and the parent block
4) Block Gas Usage: The gas consumed in processing transactions for the current block

Each of the aforementioned metrics were plotted against block number on the x-axis to illustrate the behavior of the network as the blockchain continues to grow. The following gas limits were implemented in order to observe their effects on network performance:

1) 4 Million
2) 8 Million
3) 16 Million
4) 20 Million
5) 40 Million
6) 80 Million

D. Finding Equilibrium and Optimization

Block gas limit is a primary bottleneck that limits any Ethereum based blockchain protocol. Although the relationship between gas limit and throughput is not linear, as displayed in the formula below, an increase in block gas limit increases the block size, allowing for more transactions per block, and positively correlates with overall throughput. However, as the blocks grow larger in size, mining nodes require a larger amount of time to import and validate the block information, resulting in longer block propagation time.

$$Gas\ Limit \uparrow \rightleftharpoons Transactions\ Per\ block \uparrow \tag{1}$$

$$Gas\ Limit \uparrow \rightleftharpoons Block\ Propagation\ Time \uparrow \tag{2}$$

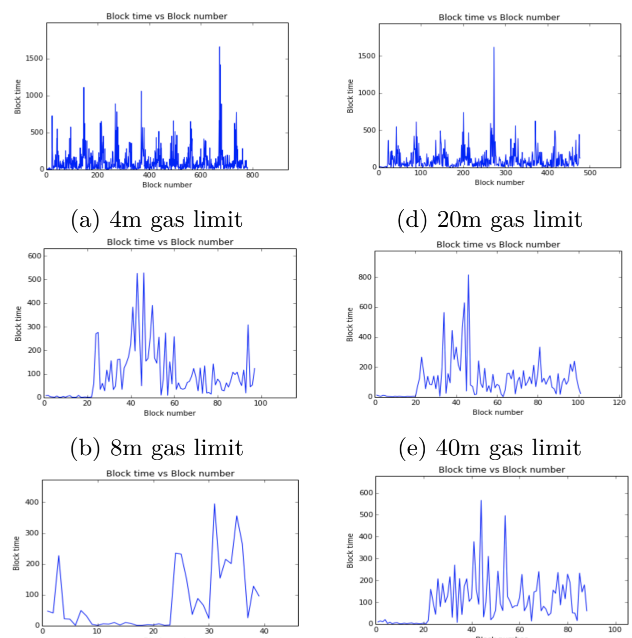
$$Throughput = \frac{Total\ Number\ of\ Tx/Block}{Total\ Block\ Time} \tag{3}$$

III. Results

The transaction throughput was tested by implementing a gas limit over a set range (4 million to 80 million). Transaction interval rate was set to 1 every 300 ms.

A. Block Time vs. Block Number

The plots below illustrate the relationship between block time and block number, implying an overall consistency in the Flux difficulty algorithm. Even as the overall hashing power of the network fluctuates, on a long enough timeline, the difficulty algorithm can adequately compensate for this dynamic and provide a relatively consistent blocktime.



(a) 4m gas limit

(d) 20m gas limit

(b) 8m gas limit

(e) 40m gas limit

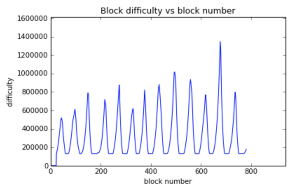
(c) 16m gas limit

(f) 80m gas limit

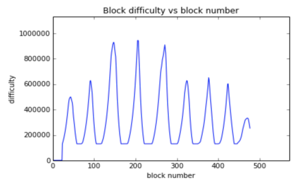
Fig. 3: Block Time vs. Block Number

B. Difficulty Level vs. Block Number

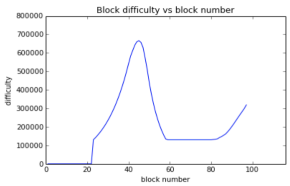
The plots below illustrate the relationship between block difficulty and block number under different gas limits.



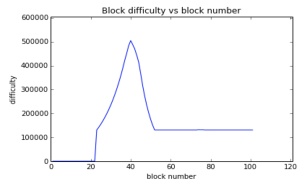
(a) 4m gas limit



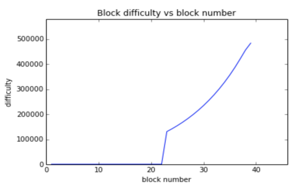
(d) 20m gas limit



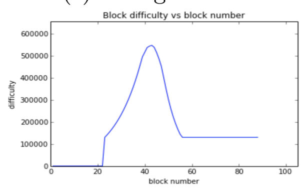
(b) 8m gas limit



(e) 40m gas limit



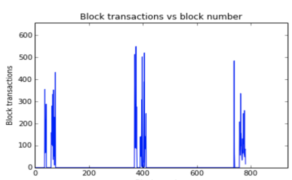
(c) 16m gas limit



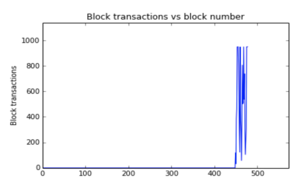
(f) 80m gas limit

Fig. 4: Difficulty Level vs. Block Number

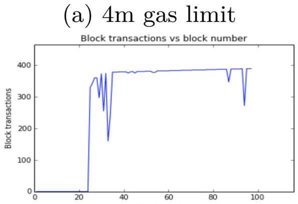
C. Block Transaction Count vs. Block Number



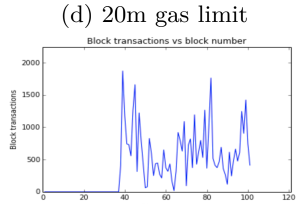
(a) 4m gas limit



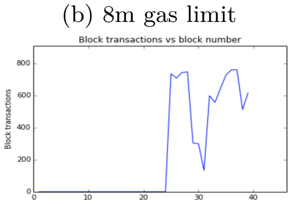
(d) 20m gas limit



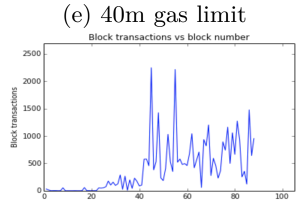
(b) 8m gas limit



(e) 40m gas limit



(c) 16m gas limit

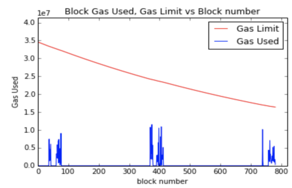


(f) 80m gas limit

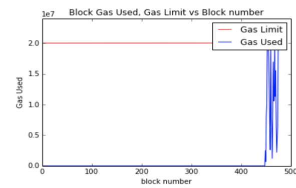
Fig. 5: Block Transaction Count vs. Block Number

D. Gas Limit vs. Block Number

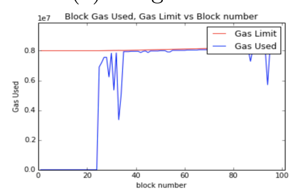
Fig 6(a-f) plots the gas used versus block number. The horizontal red line indicates the gas limit and the blue line indicates the total gas expended within each block. For Figure 1(a), the gas limit is not a horizontal line because the gas limit was not set for the initial test. The drop in gas limit is due to the network's ability to dynamically alter gas limit. Each of the successive plots test a different gas limit. As reflected by the plots, when a lower gas limit is set, the closer the block's gas expenditure comes to reaching the gas limit.



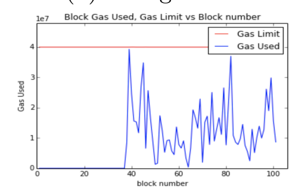
(a) 4m gas limit



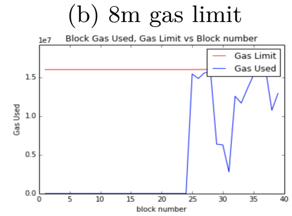
(d) 20m gas limit



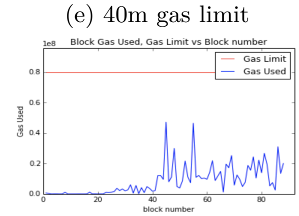
(b) 8m gas limit



(e) 40m gas limit



(c) 16m gas limit



(f) 80m gas limit

Fig. 6: Gas Limit vs. Block Number

E. Uncle Rate vs Block Number (80mil. Gas Limit)

Fig 7. below illustrates Ubiq with an average uncle rate of 2.6% at a gas limit of 80 million versus Ethereum (Fig 8). which presented an uncle rate of 27.6%, approximately 10x higher than that of Ubiq.

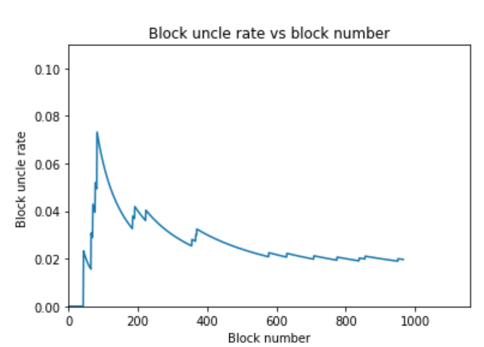


Fig. 7: Ubiq Uncle rate at 80 mil. Gas Limit

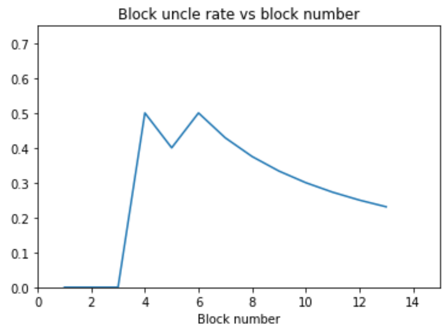


Fig. 8: Ethereum Uncle rate at 80 mil. Gas Limit

F. Identifying the Ceiling (Gas limit vs Throughput)

As mentioned in the methodology section, the relation between throughput and gas limit is non-linear. The increase of the gas limit allows more transactions to fit in each block, but simultaneously results in an increase in block propagation time. The plot below illustrates the identified equilibrium between gas limit and throughput. As can be observed in Fig. 9 below, the Ubiq network reaches optimal throughput at around 40 million block gas limit. This can be considered the ceiling.

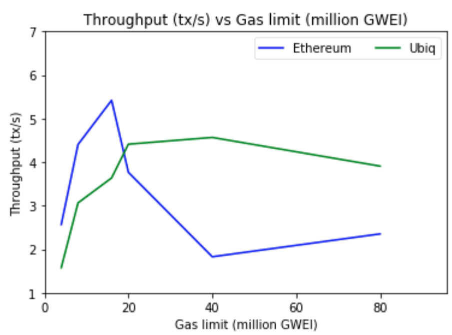


Fig. 9: Gas limit vs Throughput

G. Overall Measurements

Table I displays the correlation between gas limit and transactional throughput of the network, indicating a positive correlation between the two metrics until a certain point. The results of an 80,000,000 gas limit implies a lower transaction throughput resulting from the relationship between gas limit and block size. The higher the gas limit, the larger the block. The larger the block, the longer the block will take to propagate throughout the network and consequentially results in a lower transaction throughput. Table II displays the corresponding measurements for Ethereum.

H. The Effects of Latency on Throughput

High amounts of latency within a network can increase block propogation time and consecutively impact overall performance, specifically in relation to throughput. Table III displays a comparison of throughput performance between Ethereum and Ubiq under varying degrees of inter-node latency (25ms and 50ms one-way). The tests were conducted using the gas limits reffered to in Fig. 9 when throughput peaks. The test results imply that Ethereum displays a higher degree of stability under the presence of extreme degrees of network latency.

TABLE I: Ubiq Overall Measurements

Gas Limit	Tx Per Second	Avg Block Time (sec)	Avg difficulty
4,000,000	1.578	111.334	325275.890
8,000,000	3.063	95.237	201211.412
16,000,000	3.637	81.128	121134.641
20,000,000	4.401	100.593	308864.746
40,000,000	4.506	114.396	151631.910
80,000,000	3.908	103.738	173546.398

TABLE II: Ethereum Overall Measurements

Gas Limit	Tx Per Second	Avg Block Time (sec)	Avg difficulty
4,000,000	2.569	93.055	131072
8,000,000	4.405	49.730	131081.846
16,000,000	5.420	106.843	131074.0
20,000,000	3.767	63.730	131076.923
40,000,000	1.827	91.250	131078.0
80,000,000	2.352	105.841	131076

TABLE III: Effects of Latency on Throughput

Network Condition	Ethereum (tps)	Ubiq (tps)
No Latency	5.420	4.506
25ms Latency (one-way)	4.407	3.224
50ms Latency (one-way)	4.158	2.675

IV. Ubiq - Ethereum Comparison

The same test cases were conducted on an Ethereum network with an equal number of nodes, computational resources and network conditions. The results indicate that Ubiq presents a higher degree of stability in comparison to Ethereum at higher gas limit intervals. Ubiq transaction throughput is more than twice as high as Ethereum at a 40,000,000 gas limit.

V. Conclusion

The results of the Ubiq testing series indicate a high volume of transactional throughput which increases until the 40 million gas limit, after which it begins to gradually decline as the block size increases.

Although transactional throughput declines, the overall uncle rate of the network in every test series never rose to significant levels and remained quite low in comparison to Ethereum's which enforces a lower block time. It should be noted, however, that this test series only benchmarked performance using simple transactions. Further tests should be conducted using more complex smart contracts which require higher gas limits in order to objectively validate these metrics in a deterministic manner.

Should the Ubiq network decide to implement a higher default gas limit, it should be safe to raise this limit to approximately 40 million without sacrificing throughput. Depending on the overall health of the network, it can be assumed that block propagation time will present the only bottleneck to further scalability. Additional tests should be conducted to analyze the effects of latency on block propagation time in-depth at a higher scale.

References

[1] Ethereum Whitepaper :A Next-Generation Smart Contract and Decentralized Application Platform,July 30, 2015 from https://github.com/ethereum/wiki/wiki/White-Paper

[2] What is Ubiq (UBQ)? | Beginner's Guide. February 24, 2018, from https://coincentral.com/ubiq-ubq-beginners-guide/

[3] Gubiq github, from https://github.com/ubiq/go-ubiq

VI. Whiteblock, Inc. Legal notice