Z-DAG: A Practical Performance Analysis

Zachary Cole
Whiteblock, Inc.
zak@whiteblock.io

Nathaniel Blakely
Whiteblock, Inc.
nate@whiteblock.io

Daniel Choi
Whiteblock, Inc.
daniel@whiteblock.io

Abstract— **The following document outlines a performance benchmarking analysis of Syscoin's Z-DAG implementation. This paper describes the testing methodologies employed by the Whiteblock team within these initiatives and presents the results of these efforts. All tests were conducted on the Whiteblock Genesis blockchain testing platform. Each test series was engineered to observe the effects of various environmental conditions on application performance, specifically in regard to capabilities concerning transactional throughput which consistently presented in the range of 14,000 - 30,000 transactions per second (TPS) in an environment which emulated high degrees of activity under realistic Wide Area Network (WAN) conditions. While results are substantially higher in comparison to those achieved by alternative contemporary blockchain systems, it is important to note that these tests were designed only to observe the practical performance of Syscoin's Z-DAG implementation and not the Syscoin platform as a whole. As such, these results should only be considered to reflect the capabilities of said implementation and may not be indicative of the holistic performance of the platform itself.**

I. INTRODUCTION

Inspired by Satoshi Nakamoto's novel concept of a Bitcoin snack machine [1], the Syscoin platform is a Dash/Bitcoin fork which provides an additional masternode system (similar to that presented by Dash) in order to provide an asset layer suitable for implementation within Point-of-Sale systems. Although it is a mature client which was first presented in 2014[2], additional features have been continually developed and introduced with the latest version including the Z-DAG implementation analyzed within this report." (as Z-DAG was added to Syscoin 3,0)

The Zero-Confirmation Directed Acyclic Graph (Z-DAG) implementation intends to achieve higher degrees of transactional throughput by providing near instant settlement in a manner that substantially reduces the costly overhead associated with Bitcoin's transactional confirmation processes [3]. Most blockchain systems are able to provide such performance at the cost of security which often consecutively presents some degree of vulnerability and the potential to execute double-spending attacks. This issue appears to be adequately accounted for within the Z-DAG implementation which eliminates the possibility to engage in such attacks through its mechanism to flag any potentially questionable transactions and defer further validation to Proof-of-Work (PoW). Syscoin is merge-mined with Bitcoin, a function which allows for the mining of more than one cryptoasset without the need for additional proof-of-work. [4]. This two-tiered consensus mechanism probablistically allows for real-time settlements which enable a significantly higher degree of transactional throughput without sacrificing the robust security provided by PoW consensus [4].

The intent of these tests was to validate Z-DAG performance in regard to transactional throughput under a variety of environmental and network conditions that generally inhibit the capabilities of most blockchain systems. It is important to note, however, that these test initiatives were focused entirely on the practical performance of this implementation and do not account for any additional analysis pertaining to the security of the Syscoin platform as a whole.

II. METHODOLOGY

A. Testing Methodology

Each test case was designed [5] to observe the effects of particular environmental conditions on the performance of the Z-DAG implementation. This iterative process of isolating particular variables within a highly controlled environment ensured the results were repeatable, objective, and deterministic.

After defining the primary metrics of interest, a list of performance-impacting variables was identified and a baseline profile for performance was established to act as a control in order to provide a point of reference for future test results. This baseline profile can be referenced in Table II and was established by running the tests in an environment free of additional network conditions and impairments.

Within consecutive tests, a new network environment and blockchain was provisioned for each test series. Every test was conducted within an identical and controlled environment where the specified topology provided the same constant performance with the exception of the variables being observed.

In order to ensure the integrity of the resulting data, each test series environment consisted of a newly deployed network, an appropriate amount of nodes which were properly peered with one another, and one dedicated master node which minted the Genesis block. The role of each node was specified as either asset sender or receiver. Assets were then evenly distributed among sender nodes according to the demands of each test case.

Upon the completion of each test, average test start time, average total time, amount of assets processed in receiver's array, percentage completed, and average throughput were observed and documented for reference.

III. Tests

A. System Specifications

These testing initiatives were conducted on the Whiteblock testing platform [6], a cloud-based infrastructure engineered specifically for blockchain testing. The system specifications for these particular tests were configured to ensure an optimal run-time environment according to the requirements of the Syscoin client. The following table outlines these specifications.

Item	Description
Operating System	Ubuntu 18.04 LTS
Syscoin Core Version	4.0.0.0-80da881
Whiteblock Genesis Version	Diet-v1.1-stable
Whiteblock Syscoin Library Version	v.3.1
CPU Platform	Intel Xeon Skylake Processor
Base Frequency (GHz)	2
All-Core Turbo Freq (GHz)	2.7
Single-Core Max Turbo Freq (GHz)	3.5
vCPUs	3
RAM	4
Storage	10GB SSD
Max Number of Nodes in Network	30

TABLE I
SYSTEM SPECIFICATIONS

B. Test Conditions & Topology

1) Control Group:

	Case A	Case B	Case C
Total Nodes	30	-	-
Master Nodes	24	-	-
Sending Nodes	3	-	-
Receiving Nodes	3	-	-
# of Assets	9	-	-
Tx per Assets	250	-	-
Waiting Period (s)	11	-	-
Network Latency (ms)	0	-	-

TABLE II
CONTROL GROUP

2) Network Latency:

The purpose of this test was to determine how the Syscoin network would react to varying degrees of latency. Latency is the amount of time it takes for a node to transmit a message across the network to another node. This timing is critical to performance and security as high latency can present adverse effects, such as race conditions, which can present attack vectors within most blockchain systems, and result in suboptimal performance[7], as demonstrated in this test series where increasing round trip latency resulted in the substantial degradation of total transactional throughput.

	Case A	Case B	Case C
Total Nodes	30	30	30
Master Nodes	24	24	24
Sending Nodes	3	3	3
Receiving Nodes	3	3	3
# of Assets	9	9	9
Tx per Assets	250	250	250
Waiting Period (s)	11	11	11
Network Latency (ms)	50	100	150

TABLE III
NETWORK LATENCY

3) Number of Assets:

In Syscoin, an asset transfer is logically equivalent to a transaction, therefore an increase in the number of assets consecutively increases the number of transactions within the network. This increase also results in additional overhead which can further degrade performance and effect the number of transactions which the network can successfully process, which is demonstrable with this test series where an increasing amount of asset transfers within the network resulted in transactions dropping from the memory buffer. For this test series, varying degrees of traffic volume was applied within the network in order to measure this effect on performance.

	Case A	Case B	Case C
Total Nodes	30	30	30
Master Nodes	24	24	24
Sending Nodes	3	3	3
Receiving Nodes	3	3	3
# of Assets	21	30	39
Tx per Assets	250	250	250
Waiting Period (s)	11	11	11
Network Latency (ms)	24	24	24

TABLE IV
NUMBER OF ASSETS GROUP

4) Number of Nodes:

In this test series, the total number of masternodes was manipulated in order to observe the effects of this variance on total throughput. The number of nodes in the network has a direct effect on the propagation of the transactions where a higher population of healthy nodes should result in greater propagation times. A base amount of network latency was also included in these tests to ensure the results were indicative of real-world performance.

	Case A	Case B	Case C
Total Nodes	10	15	20
Master Nodes	4	9	14
Sending Nodes	3	3	3
Receiving Nodes	3	3	3
# of Assets	9	9	9
Tx per Assets	250	250	250
Waiting Period (s)	11	11	11
Network Latency (ms)	24	24	24

TABLE V
TOTAL NODES

5) *Doubled Latency on a Percentage of Nodes:*

The ability for a network to optimally perform under the presence of disparate nodes which exhibit higher degrees of network latency is necessary in order to ensure the long-term health of the supporting application layer. Performance can be substantially affected by high degrees of latency, even if only experienced by a minority of the nodes within a network. This test was designed to analyze Z-DAG's ability to effectively handle communications and messaging when various portions of the nodes within the network experienced high degrees of latency. In this test series, twice the amount of the average network latency was applied to a percentage of nodes in order to observe how these conditions affected performance.

	Case A	Case B	Case C
Total Nodes	30	30	30
Master Nodes	24	24	24
Sending Nodes	3	3	3
Receiving Nodes	3	3	3
# of Assets	9	9	9
Tx per Assets	250	250	250
Waiting Period (s)	11	11	11
Network Latency (ms)	24	24	24
Nodes with Double Latency (%)	0	10	20

TABLE VI

DOUBLE LATENCY ON A PERCENTAGE OF NODES

IV. RESULTS

A. *Control*

Parameters	Series A	Series B	Series C
Asset sends per block	250	250	250
Avg tx per second	145,542.92	111,619.33	114,572.75
High % tx completion	0.67	0.67	0.67
Low % tx completion	0.67	0.67	0.67
Minimum completion %	97	97	97
Total time average	16138	20313	19832
Total assets	9	9	9

TABLE VII

CONTROL GROUP RESULTS

B. *Network Latency*

Parameters	Series A	Series B	Series C
Asset sends per block	250	250	-
Avg tx per second	15328.43	8676.77	-
High % tx completion	1	1.07	-
Low % tx completion	1	0.88	-
Minimum completion %	97	97	-
Total time average	151,204.66	269,367	-
Total assets	9	9	-

TABLE VIII

NETWORK LATENCY GROUP RESULTS

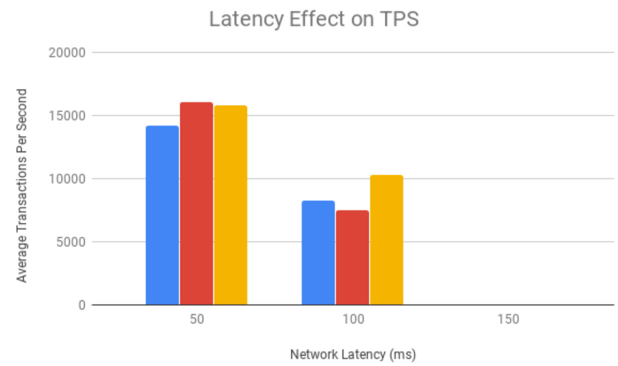

Fig. 1. Effects of Latency on TPS

As demonstrated in the previous tests, network latency had the most significant impact on performance. A large amount of latency between nodes resulted in a longer period of time for transactions to propagate throughout the network and receive confirmation. This reduced the rate at which the network could process these transactions in a secure manner. The results in Fig. 1. indicate that the test failed to complete when network latency was >150ms between each node. It is important to note that this is a very high degree of latency that may not be commonly experienced in a live production network, however, even in this case, it would likely only occur within a smaller percentage of nodes in the network.

C. *Number of Assets*

Parameters	Series A	Series B	Series C
Asset sends per block	250	250	250
Avg tx per second	45,440.54	46,843.70	60,158.00
High % tx completion	1	1	1
Low % tx completion	1	1	1
Minimum completion %	97	97	97
Total time average	117,112.66	173,328	163,221.66
Total assets	21	30	39

TABLE IX

NUMBER OF ASSETS GROUP RESULTS

Contrary to our hypothesis, an increase in the number of assets had a positive correlation with transactional throughput, as demonstrated in Table IX. This is accounted for because the Syscoin client is optimized for higher degrees of transactions. Some of these optimizations include the parallelization of transaction verification and disabling of Nagle's Algorithm, however, further analysis should be conducted in order to objectively determine the cause of these results.

D. Number of Nodes

Parameters	Series A	Series B	Series C
Asset sends per block	250	250	250
Avg tx per second	38,387.39	34,524.72	27,570.61
High % tx completion	1	1	0.96
Low % tx completion	1	1	0.85
Minimum completion %	97	97	97
Total time average	58,985	66,894.33	88,427.66
Total assets	9	9	9

TABLE X
NUMBER OF NODES GROUP RESULTS

A higher population of nodes within the network reduces total throughput since transactional payloads may experience an increase in the average number of network hops in order to reach the receivers. Syscoin compensates for this through its implementation of a broadcast-then-verify method of announcing transactions. Assuming this implementation is optimal (i.e. the Syscoin client broadcast its messages in 0 ns), there is still the delay from increased network hops. Which can be expressed as:

$$\sum_{n=1}^{hops} link(n-1) + logic(n) \tag{1}$$

Where *link* presents the time it takes to send a transaction over link *n* and *logic* presents the time it takes for node *n* to process and relay the transaction to its peers.

E. Double Latency on % of Nodes

Parameters	Series A	Series B	Series C
Asset sends per block	250	250	250
Avg tx per second	27129.98	29359.55	28574.82
High % tx completion	1	1	1
Low % tx completion	1	1	1
Minimum completion %	97	97	97
Total time average	84107	78109.33	80540.33
Total assets	9	9	9
Percentage of Nodes	0%	10%	20%

TABLE XI
DOUBLE LATENCY GROUP RESULTS

A minority of nodes experiencing suboptimal network conditions does not seem to present a significant performance bottleneck within the Syscoin network. This indicates the resiliance of the Sycoin client when presented a substantial degree of suboptimal nodes in the network, however, further analysis should be conducted in order to objectively identify the specific limitations of this tolerance threshold.

V. CONCLUSIONS

Syscoin's Z-DAG is significantly impacted by the effects of latency on the network, as demonstrated by results B, however, this isn't unexpected within any asynchronous distributed system. It is worth noting that the network was able to compensate for high degrees of latency when these conditions are limited to a minor population of nodes within the network.

Higher throughput was observed with an increase in the number of assets sent to other nodes in the network, demonstrating the network's ability to adapt to a higher volume of network traffic and activity. This can be partially attributed to Syscoin's ability to use multithreading in the process of verifying transactions by allowing workloads to be effectively distributed among available resources.

Contrary to our initial hypothesis, an increase in the number of nodes resulted in a fairly low performance cost. This can be due in part to the propagation times between sending and receiving nodes, however, it is recommended that a more substantial analysis be conducted in order to derive an objective conclusion.

Overall, the Z-DAG implementation provided a higher amount of transactional throughput without sacrificing a significant degree of resilience and reliability to adverse network effects.

ACKNOWLEDGMENT

A number of different people contributed to this research. We would like to thank Jagdeep Sidhu from Blockchain Foundry, Qi Trey Zhong of USC, Eric Lim from the Whiteblock team, and Trenton Van Epps from the Ethereum community.

REFERENCES

[1] Bitcoin Talk Forum. Internet: https://bitcointalk.org/index.php?topic=423.0, Retrieved on Jan. 30, 2019

[2] Bitcoin Talk Forum. Internet: https://bitcointalk.org/index.php?topic=587080.0, Retrieved on Jan. 30, 2019

[3] Quan-Lin Li, Jing-Yu Ma and Yan-Xia Chang, "Blockchain Queueing Theory," arXiv preprint arXiv:1808.01795, 2018.

[4] Judmayer A., Zamyatin A., Stifter N., Voyiatzis A.G., Weippl E. (2017) Merged Mining: Curse or Cure?. In: Garcia-Alfaro J., Navarro-Arribas G., Hartenstein H., Herrera-Joancomartí J. (eds) Data Privacy Management, Cryptocurrencies and Blockchain Technology. DPM 2017, CBT 2017. Lecture Notes in Computer Science, vol 10436. Springer, Cham

[5] Syscoin Whitepaper. Internet: https://www.syscoin.org/assets/whitepaper.pdf, Retrieved on Feb. 13, 2019

[6] Whiteblock GitHub Repo. Internet: https://github.com/Whiteblock/syscoin_test, Retrieved on Mar. 3, 2019

[7] Whiteblock Website. Internet: https://www.whiteblock.io/, Retrieved on Mar. 10, 2019

[8] Q. T. Zhong, Z. Cole, "Analyzing the Effects of Network Latency on Blockchain Performance and Security Using the Whiteblock Testing Platform," Internet. https://www.whiteblock.io/library/analyzing-effects-network.pdf, Retrieved on Mar. 12, 2019.