Study on Blockchain Technologies for Enterprise Applications

Author: David Rodriguez Perez
Advisor: Dr. Jeffrey Duffany
Electrical & Computer Engineering and Computer Science Department

Abstract

We are living on a time where emerging technologies are disrupting the way we work in multiple industries. One of these technologies is distributed ledgers called Blockchains. This technology enables the implementation and interaction of transactions/code states in an auditable, immutable and distributed virtual ledger.[1] Within the enterprise, we are analyzing the capabilities of Blockchain technologies that can be implemented in a private and secure environment. Given the fact that we want to benefit from this technology, which Blockchain implementation will be more efficient with less amount of infrastructure within a private network?

Introduction

Blockchain started in 2008 when the pseudonym Nakamoto published a paper describing the theory behind the digital currency Bitcoin.[3] Bitcoin was the first major Blockchain implementation that presented the concept of distributed peer to peer transactions without depending on centralized entities like banks or countries.[1] After Bitcoin we have the emergence of multiple Blockchain technologies that expand from crypto coin transactions into Turing complete scripts called smart contracts, that have infinite amount of implementations.[2]

Background

There are certain industries that depend on the validity and trust of their data for different reasons, but you depend on centralized entities/industries that in some cases end up having slow or complex processes to support this. For example, you have supply chain management between multiple corporations each with their own system to document the same shipments, legal documentations like affidavits or contracts that require a third party entity for documentation/execution, also universal identity that can be trusted between industries and public voting.[1][4] Even though this technology is growing in both support and popularity, we in the enterprise are required to study and validate witch implementations would be more efficient for a private network.

Problem

Currently we have multiple alternatives to implement a private Blockchain network that support smart contracts with different algorithms, architectures and efficiencies. With this study we want to compare two different Blockchain algorithms in order to evaluate which one would be more efficient in a private implementation.

Methodology

In order to compare the two selected implementations: Ethereum with Proof of Work versus Ethereum Proof of Authority we require to implement them in the same or as close as possible infrastructure and implement the same smart contract in order to measure both hardware utilization and customer experience while interacting with the smart contract. This is why we implemented each Blockchain in one server each with the same hardware specifications and configured two nodes each that participated in the network. Before starting the user acceptance test with the smart contract we took measurements of CPU utilization and the speed where the blocks are being created. As seen below the CPU utilization is considerably different from one technology to the other.

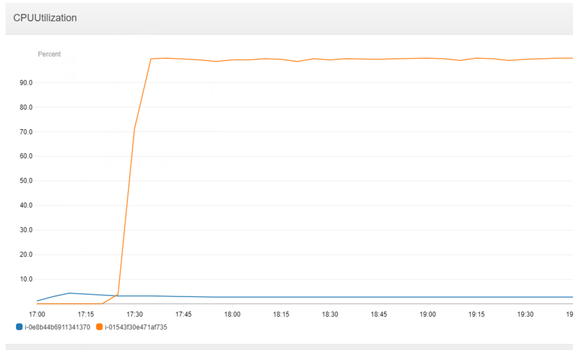

Figure 1
CPU Consumption: AWS Blue (PoA) versus Orange (PoW)

We also measured the velocity where the blocks were being created which its essential for the user experience seen below.

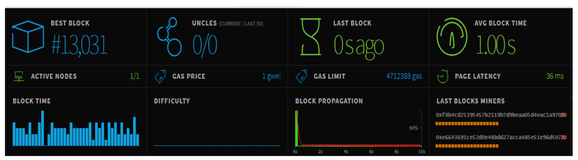

Figure 2
Proof of Authority Netstats one block per second

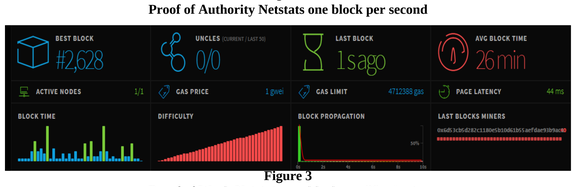

Figure 3
Proof of Work Netstats one block per 26 minutes

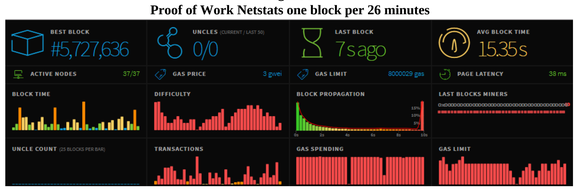

Figure 4
Public Ethereum Netstats one block per 15 seconds

As seen before in figures 2-4 we can compare a major difference in the block creation times, where PoA [figure 2] behaves as expected meanwhile PoW [figure 3] had problems performing with the selected hardware, since we expect it to behave as close as the public Ethereum network of one block per 15 seconds.

Methodology (Continued)

For the user experience test we implemented a smart contract manually utilizing the Ethereum Web Interface called remix seen in the figure below.

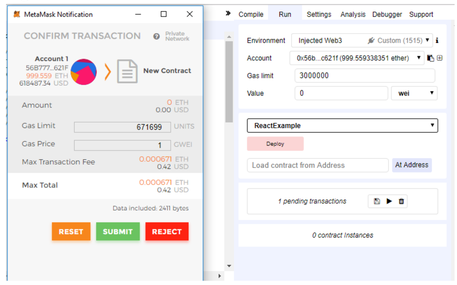

Figure 5
Implementing Ethereum smart contract with remix

With remix, we can also retrieve the contract's address in the Blockchain that can be included in our Reactjs application in order for the application to know which contract it will modify it's state. We executed 20 calls approved by our own private account in MetaMask wallet, to each network using the same Reactjs application seen below in order to measure the user experience between one and another.

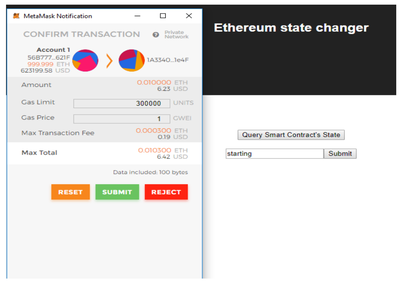

Figure 6
Interacting between the ReactJs application, the smart contract and MetaMask wallet

Results and Discussion

First of the results recompiled is that the instance utilizing the PoW is not the best hardware configuration to process a PoW environment with two nodes running in the same server, because it consumed more than 80% CPU higher than PoA and also it created blocks much more slower than the public Ethereum network that utilizes PoW. Since PoW it was so slow, we proceeded to utilize another server configuration that recommends Microsof Azure with multiple servers to distribute load and roles in the network. Even with this distribution we were able to see a high CPU utilization (90%).The second result recompiled was the user experience where we can see how the PoA network responded faster overall than the PoW network.

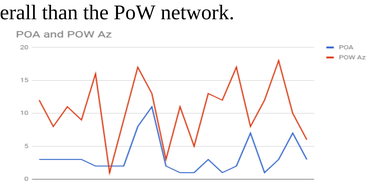

Figure 7
Smart contract state changed time in seconds

Conclusions

In theory PoA is supposed to be more efficient than PoW since the nodes don't need to perform the complex mathematical process of mining, instead they only need to sign and vote to approve the blocks that will be implemented in the Blockchain. But we still had the question of how more efficient is one to another and which hardware specifications work best with this network. These questions definitely were answered for both the time spent to change the state of the contract in the Blockchain and the hardware required to implement this solution. There is one drawback on configuring PoA block creation speed, since each block will take storage space in your Blockchain, you may end up requiring massive storage requirements in order to provide a faster responding network. In conclusion, for a private enterprise network it's recommended to implement PoA or another consensus algorithm even if you end up having the same time to change the contracts, due to the efficiency of these networks in hardware.

Future Work

For future work, we want to expand both the scope of the user experience tests by creating a more realistic and complex smart contracts and automating the user interaction to measure load capacity. We also want to explore the scalability capabilities of each implementatnion, because for PoW specifically we can't add as much nodes as we want and expect it to work more resiliently, because If N sealers are defined in the genesis file, clique will only work if int(N/2+1) nodes are online. So with PoA for 4 and for 5 nodes you will need 3 mining/signin nodes for the network to work. [5] Furthermore, we want to expand the comparison with other Blockchain technologies that have the capacity to run solidity contracts: JP Morgan's Quorum and Hyperledger's Sawtooth.

Acknowledgements

I would like thank my advisor Dr. Jeffrey Duffany for his guidance and support in this project.

References

[1] D. Yaga, et al., "Blockchain Technology Overview" [online] Available: https://csrc.nist.gov/CSRC/media/Publications/nistir/8202/draft/documents/nistir8202-draft.pdf
[2] D. Baars, "Towards Self-Sovereign Identity using Blockchain Technology" [online] Available: http://essay.utwente.nl/71274/1/Baars_MA_BMS.pdf
[3] P. Corten. "Blockchain Technology for Governmental Services: Dilemmas in the Application of Design Principles " [online] Available: https://repository.tudelft.nl/islandora/object/uuid:87709465-b9a1-48da-9ba5-eba98bc263d7/datastream/OBJ1/download
[4] L. Coleman. "Smart Contracts: 12 Use Cases For Business And Beyond" [online] Available: https://www.ccn.com/smart-contracts-12-use-cases-for-business-and-beyond/
[5] Selanfe. "Setup your own private Proof-of-Authority Ethereum network with Geth" [online] Available: https://hackernoon.com/setup-your-own-private-proof-of-authority-ethereum-network-with-geth-9a0a3750cda8