

Introduction

This document describes the roadmap of a project to compare the rewards between the different Ethereum Consensus Layer (CL) clients in all their configurations, mostly focused on block proposal rewards. The main goal is to measure if any client is consistently generating more block proposer rewards, as suggested by Attestant. The object of this study is not to publicly point fingers to the “best” nor the “worst” client, but to compare them and report privately if we find any major differences and possible improvements.

Block rewards are mainly based on which attestations are included by the client. Individual votes from validators are bundled together into an attestation by committee aggregators. Inside the block, only votes that have not been included before will increase the block reward. The more new included votes, the more rewards.

It is the client's responsibility to decide which attestations to include when one of our validators is chosen to be the next block proposer. Therefore, the reward might be higher or lower, depending on which attestations the client includes into the proposed block.

Objective

In our previous projects we have studied how all CL clients have implemented the needed features to fully participate in the Eth2.0 network, and how switching from one to another is possible and simple. Of course each client has its pros and cons.

We now dig into CL rewards to, once again, study the differences between the 6 main clients from an unbiased perspective. Our goal is to understand how each client processes / packages the attestations and blocks, so as to help them build a more standard and efficient implementation across all of them.

Methodology

In this section we describe the methodology planned for this study and what software tools need to be developed in order to correctly measure client rewards.

Architecture

We will run an instance of all beacon-nodes in Ethereum Mainnet, that will be connected to a fork of Vouch. All data will be dumped into a database to be analyzed afterwards. This will be considered a single instance, and we will deploy several instances in different geographical locations.

[
image.X0WSQ1
1431×1044 213 KB
](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/d/d0c7f008d80f5e83c4f705cdae93dd709cb1a203.jpeg)

Figure 1 - Tool Architecture

Block proposal score

The first logical step is to do a deep dive about block proposal rewards, by developing a custom piece of software that is able to get block proposals from all the clients at each slot and evaluate its score. The idea would be to store these blocks in a database, so as to analyze them offline.

To do this, we are creating a new open-source tool

(a fork of Vouch) that collects the block proposal of each client and score each of them, which should objectively output which blocks would give a greater reward. The score function will also be evaluated offline, by comparing scores given by the tool to actual rewards observed on the mainnet.

In this way we can measure if the block proposal of one client has a better / worse score than the others. If so, it would be good to understand in which situations, one client proposes a better scored block than the others.

Therefore, we aim to analyze:

- The score of the proposed block for all clients

- Score change depending on different chain state scenarios

Geographical impact

Latency plays an important role in the performance of the beacon node, in particular at the beginning of an epoch, when there are many duties to do. Not only this, the more nodes a client is connected to and closer, the easier and faster it will be receiving new blocks and aggregations.

Therefore, as part of this project, we will analyze if running validators in different parts of the world could impact their performance. Decentralized systems should also be decentralized geographically around the world.

To do this, we will deploy the same instance in different locations. After this, we will monitor what data each instance, can see and how fast they see it since the start of the slot.

We aim to measure:

- How latency impacts block proposal rewards
- Whether the same client performs differently in different locations

Latency Study

As expressed previously, the latency within a slot plays an important role in validators' performance. The time of arrival of blocks and attestations can have a significant impact on how much rewards can a validator receive. This is a crucial technical problem that has been underresearched for a long time, in part due to the complexity of the study. It is our goal to provide answers to this challenge, as part of this project.

We will collect latency raw data during an extended period of time in multiple locations over the world. Basically, we will record the timestamps of all blocks and attestations received by p2p in different locations. From this, we will try to deduce the time difference between issuing a block/attestation on a node and it being seen on the node in a different location. In order to measure when a block/attestation is issued on a node, we need to deploy nodes that participate as validators and record the timestamp of every issuance. Thus, this study is easier to be done on a testnet such as Kiln, Prater or Goerli. This should not influence in any way the accuracy of the latency data recorded, as the network layer is exactly the same whether we run on a testnet or on mainnet. What is crucial is to be able to deploy nodes in multiple locations over the world and be able to sync and record timestamps accurately.

One exploratory research analysis we could do with this latency data is an attempt at a triangulation "attack" in which we try to guess the geographical location of the validators depending on where we receive the block/attestations first. For example, we can divide the world into a few regions (i.e., Europe, North America, Oceania, etc), deploy nodes in all these regions, and then record a vast amount of data related to block proposal arrival from different locations around the world. Then, based on where the block proposal was observed first, we could guess the most likely region where the validator is located. Over a long period of time and with sufficient nodes, we could collect enough data to have a relatively good estimate of the geographical distribution of the validators. We could then compare this to network data obtained from our crawler Armirma to check whether we observe a similar distribution.

Data Release

This study will produce a significant amount of data that could be of interest to the community, thus it is our aim to make the data collected available for further analysis by other research teams and developers. However, we have to be careful about what data is released and when it is released, as this could have an important impact on the client diversity on the network. Therefore, it is our plan to be in constant communication with CL client developer teams and provide feedback to them about the performance differences observed during our study. We will set a grace period of three months after the end of our study

, during which the CL client teams will have the time to investigate and fix any performance issues observed.

This should allow sufficient time to apply corrective measures before any data is released to the public. After the grace period, the data will be released for the entire community to use and analyse. The data will be published on a platform such as BigQuery in order to allow easy access and fast API. The data will be available for access for a period of at least 1 year

, from the moment it is made public.

Attestation inclusion (Extra)

This part of the project will consist of analysing:

1. What votes does an attestation have, as it may collect votes already included in the previous blocks?

2. Which aggregations should better be chosen to be included in the block proposal, in terms of rewards. This should be a result of point 1, after analysing which votes each client may include.

All these points could impact the efficient use of the block storage and aggregations storage, as already included votes could be dismissed and this space could be used to include new votes. This could also impact the rewards of the network.

A block proposer gets rewards from the attestations it includes in its proposed block. However, if these votes were already included in previous blocks, then it is likely they do not contribute to the block proposer rewards.

This part of the analysis could be divided into:

1. Analyze how the client chooses which attestations to include in the next block to propose, and how long it takes to send the block.
2. Analyze how each client packages the attestation and sends it to the network to be used by the block proposer.

Configuration

Clients in scope of the analysis

- ★Prysm
- ★Lighthouse
- ★Teku
- ★Nimbus
- ★Lodestar
- ★Grandine

Environment

- ❖ Ethereum Mainnet

Hardware

Fat cloud node

- >32 vCores 2.3 GHz
- >120 GB RAM
- >2 TB SSD
- >10,000 Mbit/s

Each instance should be running multiple clients and the new tools for block archival, therefore a sufficiently powerful machine is needed to make sure we are not limited by hardware. These machines will be running for months in order to get statistically robustness.

Roadmap

→Initial study (Month 01)

- ◆ Read Vouch code.
- ◆ Deploy some instance.
- ◆ Test how it works.

→ Modify Vouch (Month 02)

- ◆ We want to get one block per slot independently from the validator duties. This way we can analyze data at every slot.

→ Data gathering (Months 03 - 05)

- ◆ Put the tool to run and store all the data into a database.

→ Data analysis (Month 06)

- ◆ After we have the data, it is time to analyze if one client performs better than the other when it comes to block proposal rewards.

Budget

For this project, we will hire three engineers for 6 months (~\$3,500/month) for a cost of \$63,000 in compensation. We will also use cloud resources in the order of \$1,500/month for 6 months, thus \$9,000 on cloud instances for the entire project. In addition, we plan to invest in some good servers and we will need a couple of work supplies (laptop, screen, etc), for a cost of \$8,000 on hardware. Plus, we will also have a reserve of about \$7,000 to pay for travel to related events where we can meet with partners from Lido, the Ethereum Foundation, the CL client teams, Attestant team, among many others. Finally, in order to publish the data in a proper format and make it accessible through an easy API, we will dedicate \$6,000 to BigQuery. The total budget for this 6-months project is then \$93,000 paid in DAI to be paid at the beginning of the project at the following address 0x492d683a51613aBcef3AD233149d69b7FE60FBd7.