

Background

The MEV committee is a grants-funded initiative to help the community enforce a social mitigation [strategy](#) against malicious block proposers. The committee proactively monitors, analyzes, and reports on MEV activity on dYdX v4, so that the community can respond appropriately to malicious actors.

In the previous [report](#), we described our process for investigating positive MEV events and highlighted issues with a specific validator. In our investigation, we found the issues to be a result of the validator configuration. To help mitigate similar issues with other validators, the committee has put together a guideline for improving non-malicious performance issues, which could help with discrepancies. In this report, we'll walk through the guideline and share a quick update on the past month.

Guidelines For Improving Validator Performance

The full configuration guideline has been published [here](#). We hope to find a more permanent home for this reference among the formal validator documentation in the near future. For now, we'll share a summarized version below with the goal of highlighting our key areas of improvement.

Purpose

This guideline is intended to help validators review their configurations so as to improve their overall performance. Validators experiencing issues with high discrepancies as reported by Skip's [dashboard](#) should use the below as a reference for areas of potential improvement. If all else fails, validators can contact the council directly for help and discuss next steps.

For reference, [here](#) is an example of what can go wrong for validators when deviating from standard configurations. The validator followed the guidelines below to improve their overall performance and reduce discrepancies.

Summary

Here's a summary of our recommendations in a digestible set of steps that a validator may refer back to should they experience any issues.

Category

Parameter

Configuration

Hardware

CPUs

= 8-core

Memory

= 128 GB RAM

Storage

= 1 TB NVMe SSD

Location

Tokyo, Japan

Node (config.toml)

Version

version = "v1"

Transaction Timeout

ttl_num_blocks = 20

Mempool Size

size = 5000

Cache Size

cache_size = 2000

Signing and Architecture

Signing

Local (not Remote or Threshold)

Architecture

Standard (not Sentry)

Monitoring

MEV Telemetry

—mev-telemetry-enabled=true

Contact the MEV Council

https://twitter.com/say_no_to_mev

Configurations

The guideline highlights a few important areas of consideration for a validator wanting to improve their setup and/or mitigate high discrepancy data. Below, we share quick summaries of each area.

1. Hardware

For any validator experiencing issues with latency and/or discrepancies, we recommend increasing to at least the following:

- 16-core
- 128 GB RAM
- 1 to 2 TB NVMe SSD
- Location

A validator not already in Japan experiencing issues should migrate their node to a server in Tokyo.

If a validator doesn't want to migrate their node to Japan, they could consider adding a, or multiple, Japan based peer(s) to their persistent_peers configuration, allowing for more consistent access to the regional gossip network. However, this may not be enough to resolve latency issues, and could lead to other issues commonly encountered with persistent peers (e.g. downtime from lost connections).

1. Node Configuration

Overall, we recommend using all the default configurations for dYdX found [here](#). The important parameters to keep in mind include:

- Version = "v1"
- Ttl-num-blocks = 20
- Size (mempool) = 5000
- Cache_size = 2000
- Signing and Architecture

A validator running their node through sentry or threshold/remote signing methods that is also experiencing latency should redeploy to a standard validator node. Sentries in particular are known to add latency. Given block times are under a second on dYdX Chain, we expect validators with complex signing and architecture to fall behind.

1. Monitoring and Outreach

2. The dYdX Chain [includes](#) logs and telemetry services that track MEV activity and orderbook discrepancies per block. Validators experiencing issues may want to consider enabling these additional logs by including the flag '—mev-telemetry-enabled=true' to review their block activity and identify issues with their order matching execution. The logs provide additional color on the discrepancies, which may help identify issues with their configuration.
3. The MEV Council, appointed by the community to help monitor and analyze discrepancies among active validators, is

also here to help validators struggling with high discrepancies. We encourage all validators to reach out if they would like to discuss their setups and explore methods of resolving performance related issues.

You can reach out to us through our Twitter profile here: https://twitter.com/say_no_to_mev

What's happened on-chain?

In the past month, we haven't found any significant events of MEV activity or trends among the active validator set. The validator previously experiencing issues, P2P, improved drastically following adjustments to their configuration.

Volatility in the market and high trading volumes have led to occasional spikes in discrepancy data. This can be expected however, and is mostly attributed to deleveraging and liquidation events triggered locally when a validator proposes a new block during large price swings. We hope to find new methods of isolating these events to identify any malicious activity potentially hiding within volatile periods, but for now find no reason to be concerned with the activity seen.

What's next?

We are working on ways to improve our analysis in a more automated manner through open source libraries, such that any community member could perform their own investigation. In the meantime, we'll continue monitoring on-chain activity and investigate findings as needed.