TL;DR

We are working on building the first ZKVM based on a parallel execution architecture and achieving higher TPS through the improvement of ZK-friendly design and ZK algorithms. The technical features are as follows:

- Fast proof generation
- ZK-friendly: smaller circuit scale and simplified bottom constraint units
- Fast ZK: further optimization on Plonky2
- ZK-friendly: smaller circuit scale and simplified bottom constraint units
- Fast ZK: further optimization on Plonky2
- Fast execution: Utilizing parallel execution to significantly shorten the proof generation time

Current progress:

1. In July 2022, we released the OlaVM Whitepaper.
2. November 2022, completed instruction set design and development, and realized the OlaVM execution module of the virtual machine, you can check the link: GitHub - Sin7Y/olavm: A pure Rust Olavm implementation. to view our code, continuously updated.
3. For the ZK algorithm with the fastest execution efficiency, we have completed the circuit design and algorithm research of plonky2. You can check the link: plonky2/plonky2/designs at main · Sin7Y/plonky2 · GitHubto learn more about the design of plonky2, we will optimize and improve it in the next step. Please stay tuned.

Coming soon

2022 Early December:

1. OlaVM DSL design.
2. Pre-Compilation Contract.
3. OlaVM Instruction Constraint, Context Constraint Pre-Compilation Contract Constraint.
4. First Upgrade of Plonky2.