Grant Title:

General Purpose Bidirectional Data Synchronization: Dune ↔ Local DB

Author:

The implementation of this project will be carried out by

Github: @bh2smith & @mooster531

About the Authors:

- bh2smith

is an experienced blockchain engineer and former contributor to CoW Protocol. He was the original author and primary maintainer of Dune Sync V1.

- mooster531

is a seasoned python developer eager to dip their toes into blockchain via data engineering.

Grant Category:

Other (Operation Automation)

Grant Description:

Dune Sync V1 is narrowly scoped and uni-directional. Version 2 aims to improve on by implementing bidirectional data “syncing” and furthermore separating the fetch-insert logic from the Source-Destination configuration. This would allow anyone to run the service from a plain configuration file and preexisting binary image.

The general-purpose tool will enable syncing:

1. Dune to Local:

Archives data from Dune (e.g., historical prices or arbitrary aggregated data) via their API. This allows users to save on API costs and provides quick, easy, and local access to well-structured, readily available.

1. Local to Dune:

Uploads off-chain data collected by an application-specific server (such as CoWSwap’s APP Data, Solver Competition, or ERC-4337 UserOperation requests) to Dune for the sake of transparency, availability, application insight, and overall protocol-data completion.

An outline for Version 2 is sketched in Dune-Sync v2 · Issue #112 · cowprotocol/dune-sync · GitHub. It shows how the final result will use a configuration file to specify the Source and Destination (along with the credentials for read/write access), allowing the tool to implement fetch and insert logic for both DuneAPI and Postgres.

Milestones:

For the first two milestones Local DB

refers to Postgres

Milestone 1:

Dune to Local DB

This milestone will archive data from Dune via its API, inserting it into Postgres. For example, CoW Protocol may want to archive some of the CoW AMM Data. The docker image delivered will:

• Parse a configuration file containing Dune queries

• Fetch query results from Dune

• Insert records into Postgres, mindful of column type inference

• Avoid duplication and support scheduled syncing (e.g., daily or hourly)

A proof of concept script has already been prepared atGitHub - bh2smith/dune-sync (where the project development will take place).

Milestone 2:

Local to Dune

Uploads off-chain data to Dune via itsUpload CSV endpoint. This generalizes the functionality implemented in Version 1, allowing users to upload data from a local DB to Dune using only a configuration file.

The container image should be capable of the following flow:

- User provides read access to a database and a raw SQL statement
- Data retrieved from Postgres will be transformed into CSV and uploaded to Dune via the upload_csv

endpoint

Milestone 3 [Optional]:

Additional Features

This milestone is reserved for the purpose of further maintenance, improvements, extra features or any unforeseen requirements that couldn't fit into the first two.

This could include

- support for other databases
- asyncronous query executions (performance)
- more robust insertion conflict resolution

but is ultimately left open to the desires of CoW DAO after delivery of Milestones 1 & 2.

Timeline:

Milestones 1 & 2 aim to ship by early November 2024 (in time for DevCon VII).

Funding Request:

Funding request summary: 10000 COW + (optional 5000 COW for Milestone 3) paid as 5000 COW per milestone.

Gnosis Chain Address (to receive the grant):

gno:0x1ac70B075D431379c3eaF18C130764B2f609C503

Referral:

@fleupold

Additional References & Resources:

- Sketch configuration file and inspiration for this proposal: Dune-Sync v2 · Issue #112 · cowprotocol/dune-sync · GitHub
- Implementation notes on type mapping: DuneSync.V2.md · GitHub

Terms and conditions:

By applying for this grant, I agree to be bound by the CowDAO Participation Agreement and the COWDAO Grant Terms and Conditions.