

Basic Arbitrage Trading Strategy

Overview

Your task is to develop a system that monitor two cryptocurrency exchanges in real-time, tracking their order books to identify and exploit arbitrage opportunities. The goal is to execute virtual trades that capitalize on price discrepancies between these exchanges.

Initial Setup

- **Starting Capital:** Simulate beginning with an initial balance of 1000 USDC on each exchange.
- **Asset Focus:** Apply the trading strategy to a single cryptocurrency, either BTC or ETH, by buying the asset at a lower price on one exchange and selling it at a higher price on another.
- **Profit and Loss Tracking:** Maintain and display an accurate account of your simulated profit and loss (P&L) with each trade action.

Data Sources

You are encouraged to use the following APIs for streaming order books. Alternatively, if you prefer using different exchanges, please inform us in advance to ensure compatibility and alignment with the challenge objectives.

- Aevo Exchange API: [Documentation](#)
- dYdX Exchange API: [Documentation](#)

Technical Requirements

- **Concurrency:** Demonstrate your ability to work with threads by using `tokio` for asynchronous runtime.
- **Streams:** Deponstrate your ability wot work with streams. Your solution should efficiently handle and process streaming data, particularly the real-time order book feeds from the exchanges.
- **Memory:** Demonstrate your ability to work with memory in multi-threaded environment.

Task Objectives

- **Arbitrage Detection and Simulation:** Calculate the spread between the two exchanges for your chosen asset. When a favorable arbitrage opportunity arises, simulate executing a buy

order on the exchange with the lower price and a sell order on the exchange with the higher price.

- **Profit and Loss (P&L) Tracking:** Continuously update and display your P&L after each simulated trade to monitor the effectiveness of your trading strategy.

Try to complete this task in no more than 4 hours, prioritize core functionality and simplify where you can.