Aurabit Capital Coding Assignment: Binance Volume Monitor

Version 221111

Problem Statement

Write a program that subscribes to `<symbol>@trade` channels of Binance WebSocket API, receives every incoming message, computes US$ price of cryptocurrencies as a volume-weighted average of EMA of LTPs, and produces textual output from them. You can find their API reference at https://binance-docs.github.io/apidocs/spot/en/#websocket-market-streams

Let's say we are interested in `SOL` (Solana) price. With `exchangeInfo` or `ticker/24hr` REST API, you see the following pairs have `SOL` as their base currency (in other words, these pairs match `SOL/*` wildcard)

SOLBNB, SOLBTC, SOLUSDT, SOLBUSD, SOLEUR, SOLGBP, SOLTRY, SOLBRL, SOLRUB, SOLAUD, SOLBIDR, SOLUSDC, SOLETH

For the ease of discussion, let's make a wild assumption that USDT, a cryptocurrency pegged to US$, operated by Tether, is our primary currency and that all our currency conversions are based on it.

For `SOL/BNB`, we can subscribe to `BNB/USDT` feed and cache its LTP (let's say it's 336) so that if the latest message from `SOL/BNB` channel told us the LTP was 0.088 then we know `SOL/USDT` is 0.088 * 336 = 29.568. We will continuously compute EMA of this `SOL/USDT` time series. Another assumption we make here is the LTP of `BNB/USDT` is always fresh enough for this exercise.

`SOL/USDT`-via-`BNB` is only our first example, and we can do the same with `SOL/BTC` to get `SOL/USDT`-via-`BTC`, and continue with the rest of the symbol from the list above.

The last step is to calculate weighted average of them. Let's just use the `volume` data from `ticker/24hr` REST API as a weight. For `SOL/BNB`, we get something like `“volume": "104583.80000000", "quoteVolume": "9305.61853400",` which are respectively in `SOL` and `BNB`. The two fields should be equivalent for our purpose so if we pick `“volume": "104583.80000000"` which is in SOL, we can use `weightedAvgPrice` of `SOL/USDT` to convert it to US$ volume. If we get `"weightedAvgPrice": "30.59501434",` for `SOL/USDT`, then the 24 hr US$ volume of `SOL/BNB` is 104583 * 30.595 = 3199716. This value will not vary rapidly, so for this exercise, you can compute it just once when the program is starting, and use the cached value throughout the lifetime of the program. Do the same with the rest of the symbols from the list above, and use them as weights for a weighted average:

> weighted average of `SOL` price = [3199716 * `SOL/USDT`-via-`BNB` + … + (`SOL/ETH` volume) * `SOL/USDT`-via-`ETH`] / (3199716 + … + `SOL/ETH` volume)

Notes

EMA: Exponential Moving Average

- For an unevenly spaced time series X[i] whose timestamps are T[i], EMA with the time decay parameter τ can be calculated as

  EMA[i] := α EMA[i-1] + (1-α) X[i]

  where

  α := exp[-(T[i] - T[i-1]) / τ]

LTP: Last Traded Price, the latest data from `<symbol>@trade` channel

You can use any programming languages or frameworks for this coding assignment. However, please keep in mind that the purpose of this assignment is to evaluate your hands-on codings skills as a senior IC (Individual Contributor). That is to say, in an unlikely scenario, if you manage to find an existing app that already implements all the required functionalities and make a submission based on it, that will defeat the purpose of this assignment.

At the same time, if you made some deliberate choices in your implementation specifically for this assignment, maybe in a different way than at an actual job, please make notes on them so we will have a better understanding of your thought process.

Bonus (open ended questions, mainly for discussion):

- Some sensible workaround to deal with lack of info, *e.g.* while waiting for all LTPs to be available.
- Parametrize the target currency, *i.e.* support not just `SOL` but any currency on Binance
- Sensible way to update the 24 hr volume continuously with realtime trade messages
- If we compute a weighted average upon receiving, let's say, a LTP of `SOL/BNB` , then other EMAs like `SOL/ETH` are in general slightly outdated. What's your idea on unifying weighted averaging across symbols with EMA per symbol?

Submission

You have 24 hours to work on this coding exercise. Once you are done, please pack your project as a zip or a compressed tar archive and send it back to our hiring team.

Please include a brief README with steps required to (build and) run your program. If you wrote any unit tests or other automated tests, please also include instructions for running them.

If you used C, C++ or other native compiled language and/or native compiled libraries/dependencies, there's a high chance for your code to fail to build in our environment. Please consider adding Dockerfile or something equivalent to your submission to help us evaluate your submission reproducibly.

Potential Employee Non-Disclosure Agreement

By submitting a solution to this coding exercise, you will be entering a Potential Employee Non-Disclosure Agreement with Aurabit Capital. This document and the information contained herein is and will remain the sole and exclusive intellectual property of Aurabit Capital. You are not allowed to share this document, either in its entirety or in part, with third parties, including, but not limited to, Glassdoor, Leetcode and NodeFlair, without the written permission of Aurabit Capital,

Note also that different candidates may receive different problem statements, depending on various factors such as their past experiences, educational background, and the position they are applying for.

These are an essential part of our constant efforts to make our hiring process fair and effective. Your understanding is greatly appreciated.