

Rust Internship Take Home Interview

1. Simple Client

Build a simple client that can run the two following commands:

```
./simple --mode=cache --times=10  
./simple --mode=read
```

The cache mode should listen to a websocket for given number of times(seconds) only for the USD prices of BTC. Example is given here https://binance-docs.github.io/apidocs/websocket_api/en/#symbol-price-ticker, any other websocket is also fine like kucoin, gemini, gateio, bybit etc. Calculate the average of these prices, say XXX. Then print "Cache complete. The average USD price of BTC is: XXX"

Save both the result of the aggregate and the data points used to create the aggregate to a file.

The read mode should just read from the file and print the values to the terminal.

2. Simulated distributed client

Extend the solution to Q1 by instantiating 5 client processes and one aggregator process.

- All client processes start at the same tick of the time, say 10:01:01 AM.
- Client process read values from the websocket for 10 seconds and computes the average and sends it to the aggregator process.
- Aggregator process waits for the average values from all the 5 processes. Upon getting all the values it computes yet another average and displays on the screen.

3. Using signatures

Extend the solution to Q2 where the clients send the signed messages to the aggregator. And the aggregator validates the signatures and then computes the average of averages. Any signature scheme is fine. Set up the premise such that all the processes knows the public keys of all other processes before hand.