V4 dYdX Orderbook

Author: Lawrence Chiu (lawrence@dydx.exchange)
Twitter: https://twitter.com/LawrenceChiu14
Date: 9/1/2023
Tested on: Ubuntu Server 20.04.6 LTS

Let's start off by looking at the end result first. This is the order book for ETH-USD. There are 5 areas to talk about, a to e.

    a.   This is the last trade from the v4_trades websocket channel.

From left to right, we got: 1) local server date and time, then 2) the trade's date and time, 3) It's either 'N/A' here or the *tradecreatedatheight* value if it is a long-term trade, 4) the price, 5) side, and 6) size.

b. This is the list of bids from high price to low price.


```
2023-09-02 04:22:37 Last trade: 2023-09-02T04:22:29.386Z
Bid
1634.4      (13.695)              7212 2023-09-02 04:21:59
1634.0      (0.001)              7302 2023-09-02 04:22:22
1633.3      (0.016)              7304 2023-09-02 04:22:23
1633.1      (0.014)              7332 2023-09-02 04:22:29
1632.7      (0.014)              7333 2023-09-02 04:22:30
1632.5      (0.016)              7317 2023-09-02 04:22:27
1632.4      (0.016)              7267 2023-09-02 04:22:14
1631.7      (0.016)              7303 2023-09-02 04:22:23
1631.6      (0.015)              7289 2023-09-02 04:22:19
1631.4      (0.015)              7275 2023-09-02 04:22:18
maxbid    : 1634.4
```

From left to right, we got: 1) price, 2) size, 3) the 'offset' (we'll talk about this below), and 4) the local server date and time when the price was received on websocket.

c. Same thing as above except these are asks.


```
36Z N/A 1634.5 BUY (0.014)
Ask
1634.5      (0.003)              7326 2023-09-02 04:22:29
1634.6      (0.015)              7241 2023-09-02 04:22:06
1634.8      (0.015)              7350 2023-09-02 04:22:36
1635.2      (0.015)              7290 2023-09-02 04:22:19
1635.8      (0.016)              7319 2023-09-02 04:22:27
1636.1      (0.017)              7312 2023-09-02 04:22:26
1636.3      (0.015)              7298 2023-09-02 04:22:20
1636.5      (0.015)              7288 2023-09-02 04:22:19
1636.8      (0.015)              7250 2023-09-02 04:22:08
1637.3      (0.016)              7266 2023-09-02 04:22:14
```

d. This is some stats from the bids/asks above.


```
1631.4      (0.015)              7275 2023-09-02
maxbid    : 1634.4
minask    : 1634.5 (+0.1000) 0.0061%
bidvolume: 13.818
askvolume: 0.14200000000000002
minoffset: 7212
maxoffset: 7350 (+138)
nextFundingRate: 0.0000359431818181818182    202
```

In particular, we see: 1) the highest bid, 2) the lowest ask along with a delta both in amount and percent, 3) the total bid volume, 4) the ask volume, 5) the minimum offset, and 6) the maximum offset along with a delta. Note that these numbers will change if you choose a different depth level. In this example, we set depth = 10 so we see 10 bids and 10 asks (this will display

completely, assuming that many levels are available; if not, we'll see the maximum of what is on the book).

    e.   Some market parameters from the v4_markets websocket.

What you see here depends on what comes through the websocket. In addition to the next funding rate, open interest, and 24 hour trades and USD volume, you will eventually see: priceChange24H, effectiveAt, effectiveAtHeight, marketId, and price (from Oracle). Finally, the last time tells you how long it took to render this data, in this case just 17ms.

Notes

1. Please see the V3 orderbook.pdf first at https://github.com/chiwalfrm/dydxexamples
2. The following are the V4 files with their V3 equivalents. They work exactly the same except on V4 dYdX-Testnet-2

V4 program	V3 program
v4dydxob.py	dydxob.py
v4dydxtrades.py	dydxtrades.py
v4dydxv4markets.py	dydxv3markets.py
v4dydxob2.py	dydxob2.py

3. In addition, the program v4dydxmarkets.py can be used to list every market and their status (equivalent to dydxmarkets.py for V3). There's currently 33 total.

```
vmware@v4dydxorderbooks:~/extra$ python3 v4dydxmarkets.py
BTC-USD ACTIVE
ETH-USD ACTIVE
LINK-USD ACTIVE
MATIC-USD ACTIVE
CRV-USD ACTIVE
SOL-USD ACTIVE
ADA-USD ACTIVE
AVAX-USD ACTIVE
FIL-USD ACTIVE
AAVE-USD ACTIVE
LTC-USD ACTIVE
DOGE-USD ACTIVE
ICP-USD ACTIVE
ATOM-USD ACTIVE
DOT-USD ACTIVE
XTZ-USD ACTIVE
UNI-USD ACTIVE
BCH-USD ACTIVE
EOS-USD ACTIVE
TRX-USD ACTIVE
ALGO-USD ACTIVE
NEAR-USD ACTIVE
SNX-USD ACTIVE
MKR-USD ACTIVE
SUSHI-USD ACTIVE
XLM-USD ACTIVE
XMR-USD ACTIVE
ETC-USD ACTIVE
1INCH-USD ACTIVE
COMP-USD ACTIVE
ZEC-USD ACTIVE
ZRX-USD ACTIVE
YFI-USD ACTIVE
vmware@v4dydxorderbooks:~/extra$
```

4. Because V4 does not have a centralized order book, there is no longer a global 'offset' parameter (like we had in V3) for order book price updates. At any given time, the 'correct' order book is sitting in the mempool of the current block proposer. The block proposer changes every block, so that means a new canonical mempool and therefore a new canonical order book. Therefore you may see crossed prices from the indexer's websocket. The way we deal with that is we assign every order book price, our own local offset (using the websocket's *message_id*) and we assume that later prices have higher precedence. If you do not want to remove crossed prices, you can change **v4dydxob2.py**'s *remove_crossed_prices* to False

```
import os
import signal
import sys
import time
from datetime import datetime
remove_crossed_prices = True

widthmarketstats = 24
widthprice = 10
widthsize = 10
widthoffset = 11
```

5. Let's take a look at an order book with crossed prices (by setting the parameter above).

Here you can see the highest bid is 1633.3 which is higher than the lowest ask of 1629, but you can also see the offset of the bid is 11176 while the offset of the ask is much lower at 9973. Therefore, our logic is to assume the higher offset takes precedent which allows us to remove 1629 from the list of asks. The **dydxob2.py** program does that automatically by default, as you can see below. No crossed prices!

6. One last thing is that by default, the order book display a price in color if it matches the last trade.

In this example, the asks is the same price as the last trade.