## Speed Test

We intend on creating a speed test between a non-colocated trader (Remote Trader, or rTrader) and our internal system (iTrader). Ideally, we remove as much software latency as possible, and focus solely on the rTrader latency vs the iTrader latency, and the fact that iTrader is physically closer to the exchange. We want to know how important that is, quantified by time: the delta between the two trade latencies.

We are funded in the following exchanges:

|  |  |  |
| --- | --- | --- |
| GDAX | Binance | Kraken |
| Bithumb | hitBTC | Gemini |
| BTCC | Bitstamp | Huobi Pro |

Our hypothesis is that the further away the exchange is (from the order-sending-machine), the larger the latency. This is simple physics, but the true question is, are there other speed-bumps within the cloud we are not accounting for, on the virtual machine, within the cloud provider’s network, because of cloudflare or a load balancer, etc.

The goal is to definitively **win any speed test, every time**. And if we lose, know exactly why.

To create the speed test, we will write a very simple and fast app.

1. It can be command line.
2. It will pull prices from one exchange, one pair.
3. It will take 2 prices as input, which can be hard coded.
   1. Eg: Input Low = 9001.01, Input High = 9001.50
4. Once one of those prices is breached (See below), we will fire a market order buy into a “slow” market (in GDAX, this might be LTC/BTC).
5. Breached an either mean:
   1. Best Ask < Input Low OR Best Bid > Input High
   2. Last traded price (LTP) < Input Low OR LTP > Input High
   3. We would choose the best feed for our speed test (trade vs top of book)
6. The rTrader uses the same inputs for:
   1. Input Low
   2. Input High
7. But the other trader uses a different, unique, quantity for the trade.
8. We see which trades first, by watching the book and recording the market data.

A future test will include receiving market data from a different exchange and then triggering a trade. This will require us to find the best path for the market data (over the google network, possibly).