High Frequency Trading [closed]

Asked 15 years, 5 months ago Modified 12 years, 1 month ago

Viewed 20k times

▲

62

▼

Closed. This question needs to be more focused. It is not currently accepting answers.

Want to improve this question? Update the question so it focuses on one problem only by editing this post.

Closed 10 years ago.

Improve this question

Over the last couple of weeks i have come across lots of articles about high frequency trading. They all talk about how important computers and software is to this but since they are all written from a financial point of view there is no detail about what does software do?

Can anyone explain from a programmers point of view what is high frequency trading? and why is computer/software so important in this field?

finance hft

Share edited Jun 1, 2010 at 18:42

Improve this question

Follow

Itay Karo
18.3k ● 4 ● 42 ● 59

asked Jul 24, 2009 at 11:05

Hamza Yerlikaya
49.3k ● 46 ● 151 ● 243

There have been a bunch of (non-technical) articles on this topic recently, and I was wondering exactly the same thing. +1 – Julien Chastang Aug 24, 2009 at 15:44

10 Please consider voting for the quantitative finance stackexchange site idea here: meta.stackexchange.com/questions/5786/…. That would be a very appropriate place for this kind of question. – Shane Apr 28, 2010 at 18:02

you can't go at this unless you have a lot of money or working at a financial institution. – KJW Jun 19, 2011 at 0:48

@Kim: Not necessarily. You probably won't be able to run a latency arbitrage strategy due to the hundreds of man years firms like GETCO have poured into their platform but there lots of short durations strategies that can be run by small firms. – Steve Severance Jul 28, 2011 at 20:59

@Steve Latency arb is still possible :) – Foo Bah Sep 10, 2011 at 5:13

6 Answers

Sorted by: Highest score (default)

▲

65

To expand on what Paul said:

The server executing HFT or UHFT are almost always collocated in the exchange's data center. This minimizes latency and also allows the algos use Flash orders (which might be banned soon) to get first look at order flow before the order is broadcast into the market. many algo's will evaluate an order in just a few milliseconds and this is a game where milliseconds matter. Trading groups have been known to pull out all the stops including hiring kernel developers to build custom OS components to better optimize the time between when an order hits the NIC and when the resulting action is taken.

There are a couple of large buckets of strategies which are being commonly used today:

The first is trading in front of large block orders. To use Paul's example of buying a million shares of IBM, HFT algo's will be looking for buying pressure. A firms computers at different exchanges and dark pools will need to share information since the order will be divided up and typically executed across multiple exchanges and dark pools. An HFT algo will use statistical/machine learned models to predict the size of the buying pressure and if it determines that there is enough it will also accumulate shares from across markets and attempt to sell them for a slightly higher price.

The second is liquidity rebate trading where exchanges will pay market participants to add liquidity. (See Direct Edge Pricing) Shares that are bought or sold may only be held for a very short period of time. The goal is just to collect the rebate and break even on everything else.

In both of these strategy types the idea is to make pennies (or fractions) on a trade and do this many times per day.

As you may have noticed there are a lot of HFT jobs available and thus the trades are becoming more crowded. I see this as kind of like stat arb from the early 2000s and eventually the trade will not be very profitable since so many players are trying to make it.

As for why software is important: milliseconds matter. Latency is super important and the code needs to be tight, fast and rock solid stable. Having an algo crash and being caught with shares when the market moves against you is not very profitable. Engineering for these requirements is necessarily different and requires different skills. Crunching the full order book in real time does requires some horsepower and good algorithms. It is fun and interesting though.

Share Improve this answer Follow

edited Jan 4, 2010 at 19:24

answered Aug 14, 2009 at 19:48

Steve Severance

6,646 ● 2 ● 35 ● 44

6 +1 an excellent answer and one which would have helped me greatly when I architected and implemented a black-box arb engine that (sometimes) made money!It could assess about 4.5 million arb routes a second on a desktop PC.We colocated to the US and, in the end - it broke even.I often had

to deal with a break-down half way through some convoluted trading route and plug up the gap manually.My worst was a few hundred million yen sitting out in the cold whilst the market was sliding.That mistake cost our investor $10000 USD. Scary stuff but very, very exciting. (laments)Oh, with a bit more time... – Andras Zoltan Aug 17, 2010 at 12:03

how did you plug the gap?? – Nikos Dec 18, 2012 at 12:55

@Nikos I would assume through manual orders or some other custom scripts that adjust orders. – user9903 Jan 20, 2016 at 21:32

20

There are two parts to any HFT system:

1. Real time super low latency trading - subscribe to real time order book and price information from lots of different sources, execute calibrated algorithms designed to either carry out a large order with minimal slippage (i.e. you want to buy 1 million shares of IBM by the end of the day without moving the market too much), or just to try to statistically make money based on short term arbitrage. This system also has to provide good risk and position management tools to allow one or more human operators to effectively monitor and control what the system is doing.

2. Overnight/Weekly etc. analysis of large quantities of "tick data" (price, time and order book information, and historical data on the systems previous trading activity), looking to optimize and "search for" the best algorithms to be executed in real time by part #1. i.e.

"calibrate" and test the algorithms that will execute in #1.

The first one requires low latency and extremely good access to markets (i.e. a direct network connection to the exchange with minimal hops). This part usually has to be written in a non-GC language like C or C++ (a half second delay while the garbage collector stops the world could be very costly). The second usually requires a grid and lots of good simulation and statistical analysis software, AI algorithms etc.

Share Improve this answer Follow

answered Jul 24, 2009 at 11:22

Paul Hollingsworth
13.4k ● 12 ● 53 ● 69

▲

9

▼

I would just add, that the most prevalent applications in this kind of trading tend to be CEP (complex event processing). Some examples are Streambase, Apama, and Aleri. On the other end, to deal with the massive quantities of data, people use high-speed databases, such as KDB, OneTick, and Vhayu.

If you want to understand the kind of technical challenges, I suggest looking at these vendors first. Their marketing materials will give you a good sense of the business applications as well as the technical challenges.

Share Improve this answer

Follow

answered Sep 10, 2009 at 6:19

Shane

100k ● 35 ● 228 ● 219

▲

7

▼

At certain times (for example on a futures expiry) it is necessary to do thousands of trades a minute - obviously humans can't do this unaided. This BTW is a very stressful time for the programmer, as if anything goes wrong, there is almost no chance of recovery - programmers tend to watch their log files go streaming by with their hearts somewhat in their mouths.

Share Improve this answer

Follow

answered Jul 24, 2009 at 11:16

anon

Haha, I know this is an older question but just reading this made me glad to know there are others on SO who can relate to my day-to-day experience developing for an HFT company. – Dan Tao Nov 19, 2009 at 16:21

4 Very true. When things go wrong, you have people coming up to you asking "whats going on, whats wrong, why is not working, why are we losing money!", usually very loudly while you desperate try to make sense of the huge log file! – Luhar Feb 25, 2010 at 18:15

▲

3

You need to track prices, quickly decide what is going up and down and buy and sell accordingly. Since there're lots of different positions traded the better software you use for that analysis and performing deals the more money you can potentially make.

Better would mean frequently updating data, pinpointing interesting tendencies in such a way that you can react to them quickly, being easy to use when performing frequently required operations.

Share Improve this answer

Follow

answered Jul 24, 2009 at 11:10

sharptooth

170k ● 108 ● 535 ● 1k

1

> why is computer/software so important in this field?

The highest performance and lowest latency is desirable, since the faster that you can react to things, the more money you can potentially make.

Share Improve this answer

Follow

answered Jul 24, 2009 at 11:09

xyz

27.8k ● 30 ● 106 ● 126