

Application Development Trends

When Milliseconds Make Millions: Why Wall Street Programmers Earn the Big Bucks

- 07/29/2011

A recent article I wrote about [the highest paid programmers](#) provided a peek into the world of big-money, automated trading on Wall Street. In high-frequency trading (HFT), programmers eke out every last incremental tick in performance to build algorithms that battle other algorithms for computational supremacy and millions in profits -- and earn a lot in the process.

So it wasn't surprising when, in response to the article, programmers on this site [and others](#) started talking about the tools, languages and techniques used in HFT in financial markets.

My previous article noted that an HFT company, Teza Technologies, offered \$1.2 million to a C++ developer named Sergey Aleynikov to leave his \$400,000-per-year position at Goldman Sachs and work on its own trading system. Aleynikov was [convicted of illegally downloading](#) proprietary code from Goldman Sachs' servers, which landed him in prison.

Aleynikov's trial provided a rare glimpse into the rarefied air of HFT -- sometimes known as algorithmic trading or black box trading-- which reportedly accounts for more than 70 percent of equity trading these days. It's an elite niche of computer science that few in the industry seemingly want to talk about in specific terms, probably because hundreds of millions of dollars are at risk. When the stakes are that high, the players keep their cards pretty close to their vests. As an AdvancedTrading.com article [described HFT firms](#), "They are, as a rule, secretive, stealthy, smart and relatively unknown." In fact, the presiding judge in the Aleynikov trial [sealed the courtroom](#) to hear testimony about the proprietary HFT source code in question.

There are many different aspects to HFT, but basically it comes down to building algorithms and computer systems that can monitor and digest huge amounts of financial data in order to automatically conduct transactions in much less time than it would take a human trader to glance from his BlackBerry to his Bloomberg Terminal. Or, as [Wikipedia](#) explains:

"In [HFT], programs analyze market data to capture trading opportunities that may open up for only a fraction of a second to several hours. HFT uses computer programs and sometimes specialized hardware to hold short-term positions in equities, options, futures, ETFs, currencies and other financial instruments that possess electronic trading capability. High-frequency traders compete on a basis of speed with other high-frequency traders, not long-term investors."

HFT was blamed for the huge drop in the stock market in May 2010, called the "[flash crash](#)." The *Wall Street Journal* [reported](#) that, "At one point, HFTs traded more than 27,000 contracts in just 14 seconds."

In this world, speed kills -- or lets your program live another day, if it's working well and beating other systems. Tackling latency is of primary importance, with programmers using every last tool and trick in the book -- and continually making new ones -- to squeeze out every last millisecond of performance.

Or would you believe every last nanosecond, even?

A nanosecond is 1 billionth of a second. One company, Fixnetix, [recently announced](#) that its "microchip for ultra-low latency trading is now the world's fastest trading appliance for the financial markets. Customers are now seeing latencies as low as 740 nanoseconds through the stack (wire-to-wire)."

So what kind of language tools and techniques are programmers using to develop such high-performance systems? I suspected it would be something as close to the metal as possible, perhaps even assembly language for certain parts or some esoteric proprietary language. But some surprises emerged in readers' discussions of the previous article, with languages like [Haskell](#) and [Smalltalk](#) mentioned prominently.

However, it's no coincidence that Aleynikov was a C++ guy. As HFT expert Matt Davey, chief technology officer at [Lab49](#), a financial services industry consulting firm, explained in an interview: "From a HFT platform perspective, C/C++ is the language of choice due to the latency requirements," he said. "The lower the latency, the more C/C++ is important."

A contract programmer for a New York City firm doing "trading strategy back testing" I talked with who didn't want to be identified confirmed this. "I work mostly in C++," said "Jack" (a pseudonym), "but also Java, Lisp, Matlab, R, Fortran and JavaScript."

Programmer/entrepreneur [Jason Roberts](#) told me the languages/tools used by HFT coders "really varies from firm to firm. Some use the Microsoft stack (.NET, C#, etc.), some use Java or C++ and others use more exotic languages like [Ocaml](#)."

Roberts has a unique perspective on the HFT industry, having worked as a founding partner for a series of small, proprietary trading companies and for a hedge fund, developing the trading technology. "I built everything from the order managers and feed parsers to the low latency messaging systems, trading UIs and tick capture databases, machine learning libraries and trading engines," Roberts said. "In most cases, I developed everything (or most everything) entirely myself."

Surprisingly, Roberts said he used C++ sparingly. "I never really used anything very esoteric and, in fact, tended to rely on tools that were fairly common and unsophisticated. For example, back between 2001 and 2003, I wrote two entire HFT trading systems (one for option market making and the other for equity pairs trading) using VB6, which even back then would've struck many as silly. The reality was that the applications worked, they were reliable and they were fast (since VB6 is compiled)."

"Moreover," he continued, "I was very productive using VB6, which enabled me to get a lot more done than if I had attempted to build everything using C++. In fact, I would only use C++ for the very rare algorithm that needed an extra boost of speed, and the one I remember best as being slow in VB6 was the square root function, of all things!"

Mike O'Hara, publisher of the [High Frequency Trading Review](#), also listed the "C" languages as being prominent in the industry, along with Java, [Matlab](#) and [Cuda](#). The latter, he said in an interview, is used for programming GPUs for an extra boost in performance, which seems to be an increasing trend (more on that later).

A Slashdot.org reader named "TheRaven64" said in [the comments](#) about my highest-paid-programmers article that "I periodically get contacted by recruiters for banks, because my CV mentions Haskell, and there's a massive shortage of Haskell programmers."

The use of Haskell indicates that the new darling of the software development world, functional programming, plays a large part in the HFT industry. TheRaven64 said in his comments that recruiters who contacted him for possible employment in the financial industry "wanted people with any functional programming experience. A lot of these companies use their own in-house languages, but functional programming is popular because it's easy to verify functional programs, and because languages like Haskell facilitate rapid development. Cincom does good business selling Smalltalk to trading houses for a similar reason. Typically, a small improvement in a trading algorithm gives you an advantage for a day -- maybe less. Being able to go from idea to deployment in under an hour is something that Smalltalk and Haskell give you, and that's something that the financial industry values highly."

When one reader asked about the role of testing in such quick deployments, he was answered by yet another reader who said, "I think that was the point. Writing in one of these languages means that you can prove the correctness of the code, presumably automatically, so presumably very quickly."

Of course, there's much more to the nuts and bolts of HFT than the choice of a programming language. "From an OS perspective, the default is Linux, a custom build that has removed all the fat," Davey said. "From a back-testing perspective, there will be some kind of tick database -- Kdb or similar." But other languages come into play also, Davey said when I asked him about needed skills for HFT programming. "On the UI and data side, C# and Java are relevant, coupled with database knowledge."

High-performance hardware and a tight marriage with software are also important. A reader on [Hacker News](#) with the handle "actionbrandon" [noted in the comments](#) that, "From the CS side, some of the more cutting-edge stuff I've heard of being worked on involves [hacking graphics cards](#) to get a speed edge and [building data centers](#) in the middle of the ocean for the same reason."

That hardware angle was echoed by Davey: "GPU/FPGA [graphics processing units/Field-Programmable Gate Array] usage is not uncommon from a hardware perspective," he said.

That was confirmed by O'Hara, who said "Cuda is increasingly used for programming parallel processing algorithms [pattern matching] on GPUs. [Impulse C](#) is used for programming FPGAs."

But the highest-performing systems in the world are worth nothing if the underlying logic is unsound. Or, as Davey worded it, "Having low latency, but a strategy/model that generates no profit, isn't really going to help, and likewise the reverse."

Roberts agreed. Besides a thorough understanding of data structures, algorithms and concurrency issues, programmers "need to understand what it is they're building," he said. "One of the biggest hurdles for HFT firms is that the programmers, quants and traders don't understand one another, which tends to cause a lot of problems. An HFT programmer who really groks trading and some of the theory and math behind finance will be regarded as extremely valuable and can usually write their own ticket."

That strategic/planning aspect somewhat blurs the lines between coders and "quants" (quantitative analyst) in the financial trading industry.

"From my understanding, programmers making [\$300,000 to \$400,000] in the financial industry are typically quantitative analysts or financial engineers with masters degrees or Ph.Ds in these fields," a Slashdot reader with the handle "ThomasFlip" commented. "Their primary duties are things like modeling complicated financial scenarios or finding statistical anomalies to exploit in high-frequency trading. Yes they code their strategies, but I don't know if I'd put them in the same category as your typical programmer."

Or, as Davey said, "If you have a head for maths, then you are more than likely to be able to move towards strategy/model implementation, rather than just developing the platform."

Experts describe the HFT environment as a pressure-cooker for programmers tasked with these high-performance, ad-hoc apps, perhaps requiring them to work on-site with traders to continually tweak or build new apps as market conditions change. O'Hara, when asked about the traits needed by HFT programmers, cited the "ability to work under pressure when the traders are screaming at you!"

There are ways to get away from that stressful atmosphere, however. "Jack," though he works for a firm in New York City, works from his home in the Midwest for three weeks out of the month -- and has pulled in \$270,000 per year for the past three years. He said, "\$1,000 per day is easy to achieve in this environment." Of course, having around 25 years of experience helps him out.

But as attractive as the salaries may be in HFT, the work isn't for everybody. One Slashdot reader said, "It is soulless work only for the black-hearted. Lucky if you get to work on interesting algorithms. Most work is in excruciatingly painful market data or regulatory related work."

And Slashdot reader "TheLink" summed up the feelings of many when he said, "In some jobs you earn twice as much, but end up looking twice as old and feeling twice as old...."

Do you have experience with HFT programming? Is it soul-destroying, highly lucrative, neither, both? We'd love to hear your thoughts. Post in our comments area or [drop me a line](#).

--- More articles from David Ramel on ADTmag.com --

- [*What Do the Highest-Paid Programmers Make?*](#)
- [*Developers Gone Bad: Fired Programmers Strike Back...With Nasty Code*](#)
- [*Job-Killing Decisions by Techies: What Not To Do*](#)

out the Author David Ramel is an editor and writer for 1105 Media.