acmqueue

Barbarians at the Gateways

High-frequency Trading and Exchange Technology

Jacob Loveless

I am a former high-frequency trader. For a few wonderful years I led a group of brilliant engineers and mathematicians, and together we traded in the electronic marketplaces and pushed systems to the edge of their capability.

HFT (high-frequency trading) systems operate and evolve at astounding speeds. Moore's law is of little comfort when compared with the exponential increase in market-data rates and the logarithmic decay in demanded latency. As an example, during a period of six months the requirement for a functional trading system went from a "tick-to-trade" latency of 250 microseconds to 50. To put that in perspective, 50 microseconds is the access latency for a modern solid-state drive.

I am also a former and current developer of exchange technology. The *exchange* is the focal point of HFT, where electronic buyers and sellers match in a complex web of systems and networks to set the price for assets around the world. I would argue that the computational challenges of developing and maintaining a competitive advantage in the exchange business are among the most difficult in computer science, and specifically systems programming. To give you a feeling of scale, the current exchange technology is benchmarked in nightly builds to run a series of simulated market data feeds at 1 million messages per second, as a unit test. There is no such thing as premature optimization in exchange development, as every cycle counts.

The goal of this article is to introduce the problems on both sides of the wire. Today a big Wall Street trader is more likely to have a Ph.D from Caltech or MIT than an MBA from Harvard or Yale. The reality is that automated trading is the new marketplace, accounting for an estimated 77 percent of the volume of transactions in the U.K. market and 73 percent in the U.S. market. As a community, it's starting to push the limits of physics. Today it is possible to buy a custom ASIC (application-specific integrated circuit) to parse market data and send executions in 740 nanoseconds (or 0.00074 milliseconds).[4] (Human reaction time to a visual stimulus is around 190 million nanoseconds.)

In the first of the other two articles in this special section on HFT, Sasha Stoikov and Rolf Waeber explain the role of one-pass algorithms in finance. These algorithms are used throughout the industry as they provide a simple and very fast way of calculating useful statistics (such as correlation between two streams). One-pass algorithms are also easier to implement in hardware (using Verilog or VHDL) because they require only a few bits of memory, compared with a vector of historical events.

In the other article, Stephen Strowes discusses a method for estimating RTT (round-trip time) latency from packet headers. Estimating RTT is a key issue for both HFT firms and exchanges, and the method presented here solves an interesting problem when you cannot install a tap on every interface or on the other side of the wire.

MY TIME IN HFT

When I began in HFT, it was a very different world. In 2003, HFT was still in its infancy outside of U.S. equities, which two years earlier had been regulated into *decimalization*, requiring stock exchanges to quote stock prices in decimals instead of fractions. This decimalization of the exchanges changed the minimum stock tick size (minimum price increment) from 1/16th of a dollar to $0.01 per share. What this meant was that "overnight the minimum spread a market-maker (someone who electronically offered to both buy and sell a security) stood to pocket between a bid and offer was compressed from 6.25 cents…down to a penny."[5]

This led to an explosion in revenue for U.S. equity-based HFT firms, as they were the only shops capable of operating at such small incremental margins through the execution of massive volume. Like the plot of *Superman III*, HFT shops could take over market-making in U.S. equities by collecting pennies (or fractions of a penny) millions of times a day. I wasn't trading stocks, however; I was trading futures and bonds. Tucked inside a large Wall Street partnership, I was tackling markets that were electronic but outside the purview of the average algorithmic shop. This is important as it meant we could start with a tractable goal: build an automated market-making system that executes trades in under 10 milliseconds on a 3.2-GHz Xeon (130 nm). By 2004, this was halved to 5 milliseconds, and we were armed with a 3.6-GHz Nocona. By 2005 we were approaching the one-millisecond barrier for latency arbitrage and were well into the overclocking world. I remember bricking a brand-new HP server in an attempt to break the 4.1-GHz barrier under air.

By 2005, most shops were also modifying kernels and/or running realtime kernels. I left HFT in late 2005 and returned in 2009, only to discover that the world was approaching absurdity: by 2009 we were required to operate well below the one-millisecond barrier, and were looking at tick-to-trade requirements of 250 microseconds. *Tick to trade* is the time it takes to:

1. Receive a packet at the network interface.
2. Process the packet and run through the business logic of trading.
3. Send a trade packet back out on the network interface.

To do this, we used realtime kernels with bypass drivers (either InfiniBand or via Solarflare's Onload technology). At my shop, we had begun implementing functionality on the switches themselves (the Arista switch was Linux based, and we had root access). We must not have been alone in implementing custom code on the switch, because shortly after, Arista made a 24-port switch with a built-in FPGA (field-programmable gate array).[1] FPGAs were becoming more common in trading—especially in dealing with the increasing onslaught of market-data processing.

As with all great technology, using it became easier over time, allowing more and more complicated systems to be built. By 2010, the barriers to entry into HFT began to fall as many of the more esoteric technologies developed over the previous few years became commercially available. Strategy development, or the big-data problem of analyzing market data, is a great example. Hadoop was not common in many HFT shops, but the influx of talent in distributed data mining meant a number of products were becoming more available. Software companies (often started by former HFT traders) were now offering amazing solutions for messaging, market-data capture, and networking. Perhaps as a result of the inevitable lowering of the barriers to entry, HFT was measurably harder by 2010. Most of our models at that time were running at half-lives of three to six months.

I remember coming home late one night, and my mother, a math teacher, asked why I was so depressed and exhausted. I said, "Imagine every day you have to figure out a small part of the world.

You develop fantastic machines, which can measure everything, and you deploy them to track an object falling. You analyze a million occurrences of this falling event, and along with some of the greatest minds you know, you discover gravity. It's perfect: you can model it, define it, measure it, and predict it. You test it with your colleagues and say, 'I will drop this apple from my hand, and it will hit the ground in 3.2 seconds,' and it does. Then two weeks later, you go to a large conference. You drop the apple in front of the crowd…and it floats up and flies out the window. Gravity is no longer true; it was, but it isn't now. That's HFT. As soon as you discover it, you have only a few weeks to capitalize on it; then you have to start all over."

THE HFT TECHNOLOGY STACK

What follows is a high-level overview of the modern HFT stack. It's broken into components, though in a number of shops these components are encapsulated in a single piece of hardware, the FPGA.

COLLOCATION

The first step in HFT is to place the systems where the exchanges are. Light passing through fiber takes 49 microseconds to travel 10,000 meters, and that's all the time available in many cases. In New York, there are at least six data centers you need to collocate in to be competitive in equities. In other assets (foreign exchange, for example), you need only one or two in New York, but you also need one in London and probably one in Chicago. The problem of collocation seems straightforward:

1. Contact data center.
2. Negotiate contract.
3. Profit.

The details, however, are where the first systems problem arises. The real estate is extremely expensive, and the cost of power is an ever-crushing force on the bottom line. A 17.3-kilowatt cabinet will run $14,000 per month.[7] Assuming a modest HFT draw of 750 watts per server, 17 kilowatts can be taken by 23 servers. It's also important to ensure you get the right collocation. In many markets, the length of the cable *within the same building* is a competitive advantage. Some facilities such as the Mahwah, New Jersey, NYSE (New York Stock Exchange) data center have rolls of fiber so that every cage has exactly the same length of fiber running to the exchange cages.[3]

NETWORKING

Once the servers are collocated, they need to be connected. Traditionally this is done via two methods: data-center cross connects (single-mode or multimode fiber) and cross-data-center WAN links.

The Cross Connect and NAT. Inside the data center are multiple exchanges or market-data feeds. Each endpoint must conform to the exchange transit network. This is a simple NAT (Network Address Translation) problem and can be tackled at the switch level, as shown in figure 1. Multiple vendors (for example, Arista[2]) offer hardware-based NAT at the port level. For some marketplaces (such as foreign exchange) there may be as many as 100 such NAT endpoints within a single data center.

The key to the internal NAT problem is the core nature of trading: correlated bursts. These bursts are what make HFT networks so ridiculously difficult. Figure 2 is a screenshot of an internal

FIGURE 1

Using NAT to Conform to the Exchange Transit Network

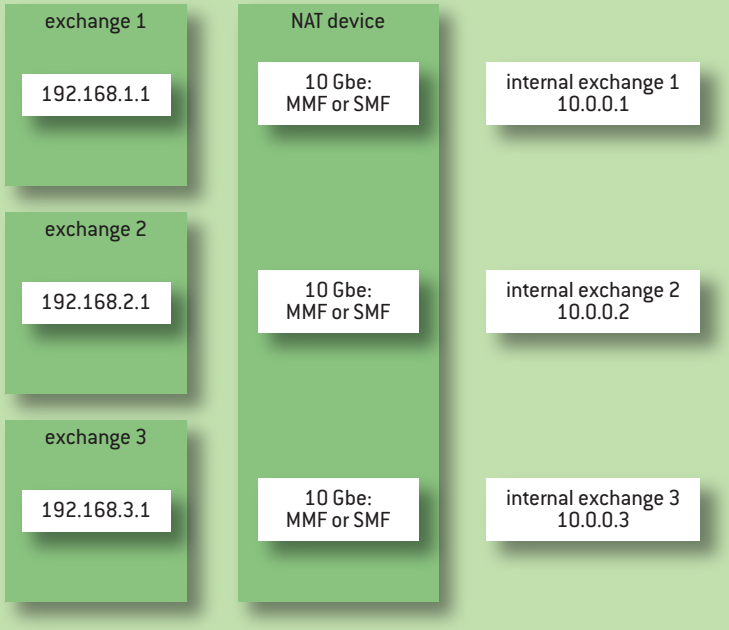


FIGURE 2

Monitoring a Packet Burst

monitoring application that shows the packet-per-second rates for a (small) collection of symbols moving together. The ISM (Institute of Supply Management) manufacturing composite index is a diffusion index calculated from five of the 11 subcomponents of a monthly survey of purchasing managers at roughly 300 manufacturing firms nationwide. At 14:00 EDT the ISM announcement is made, and a burst occurs. Bursts like this are common and happen multiple times a day.

As the world becomes more interconnected, and assets are more closely linked electronically, these bursts can come from anywhere. A change in U.K. employment will most certainly affect the USD/GBP rate (currency rates are like relative credit strength). That in turn affects the electronic U.S. Treasury market, which itself affects the options market (the risk-free rate in our calculations has changed). A change in options leads to a change in large-scale ETFs (exchange-traded funds). The value of an ETF cannot be greater than the sum of its components (e.g., the SPDR S&P 500), so the underlying stocks must change. Therefore, the state of employment in London will affect the price of DLTR (the Dollar Tree), which doesn't have a single store outside North America—it's a tangled web.

WAN Links. Outside the data center the systems need WAN links. Traditionally HFT shops ran two sets of links, as shown in figure 3: a high-throughput path and a lower-throughput *fast path*. For the

FIGURE 3

WAN Links: High-Throughput Path and Lower-Throughput Path

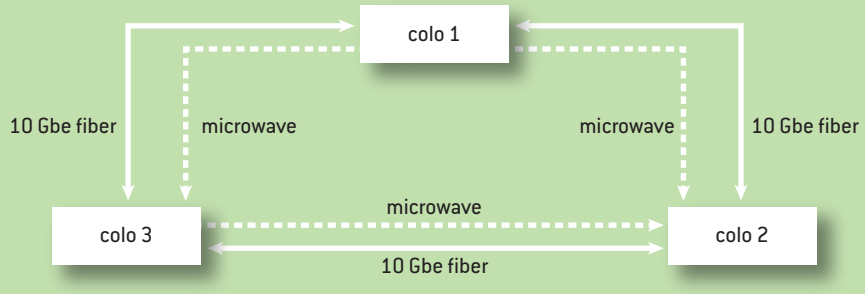
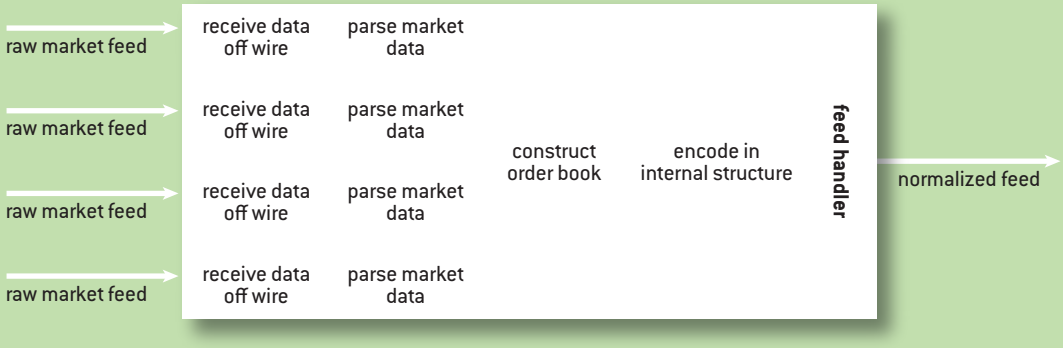


FIGURE 4

Feed Handler Parsing Market-Data Feeds



high-throughput path, private point-to-point fiber—10GbE (gigabit Ethernet) is preferred. For the fast path, each location allows for options. In the New York metro area, both millimeter and microwave solutions are available. These technologies are commonplace for HFT fast-path links, since the reduced refractive index allows for lower latency.

FEED HANDLER

The *feed handler* is often the first bit of code to be implemented by an HFT group. As shown in figure 4, the feed handler subscribes to a market-data feed, parses the feed, and constructs a "clean" book. This is traditionally implemented on an FPGA and has now become a commodity for the industry (http://www.exegy.com). Most feed handlers for U.S. equities are able to parse multiple market-data feeds and build a consolidated book in less than 25 microseconds.

TICKERPLANT

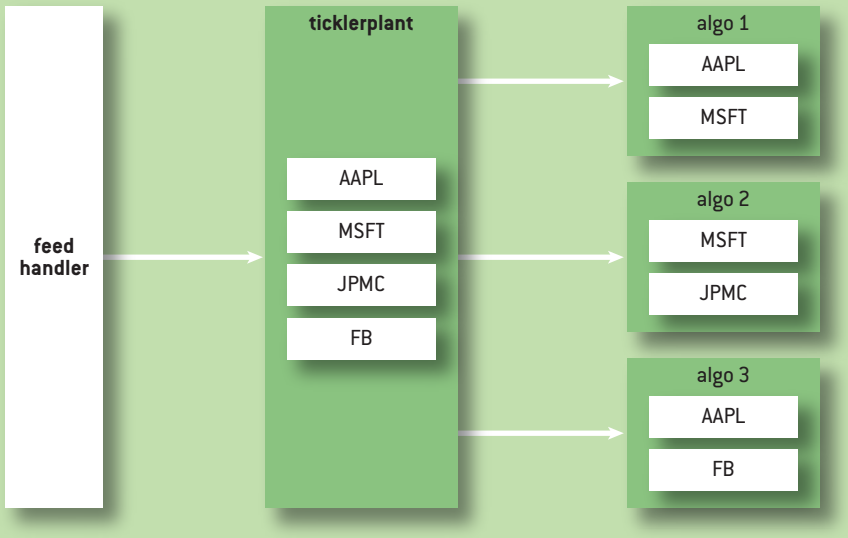
The *tickerplant* is the system component responsible for distributing the market-data feeds to the internal systems based on their subscription parameters (topic-based subscriptions), as shown in figure 5. In these scenarios, the tickerplant is like a miniature version of Twitter, with multiple applications subscribing to different topics (market-data streams).

In addition to managing topic-based subscriptions, advanced tickerplants often maintain a cache of recent updates for each instrument (to catch up subscribers), calculate basic statistics (for example, the moving five-minute volume-weighted average price), and provide more complicated aggregate topics (for example, the value of an index based on the sum of the underlying 500 securities).

LOW-LATENCY APPLICATIONS

In my experience, most high-frequency algorithms are fairly straightforward in concept—but their

FIGURE 5 **Tickerplant Distributing Market-Data Feeds**



success is based largely on how quickly they can interact with the marketplace and how certain you can be of the information on the wire. What follows is a simple model (circa 2005), which required a faster and faster implementation to continue to generate returns.

To begin, let's review some jargon. Figure 6 shows the first two levels of the *order book*. The order book is split into two sides: *bids* (prices at which people are willing to buy); and *asks* or *offers* (prices at which people are willing to sell). A *queue* consists of the individual orders within a price.

QUEUE LIFE: GETTING TO THE FRONT

Orders are executed in a first-in, first-out manner (in most marketplaces). When describing the life of an individual order in an individual queue, we often say that X is *ahead*, and Y is *behind*. More generally, we say we are in the top X percent of the queue. This is called the *queue position*. Figure 7 shows an order in the middle of the queue by the last time step: there are six shares in front of

FIGURE 6

The Order Book

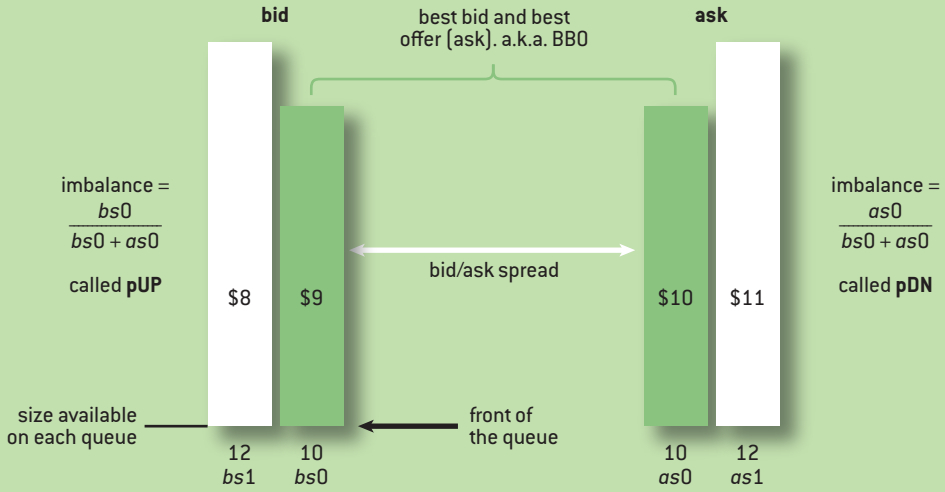


FIGURE 7

The Queue Position

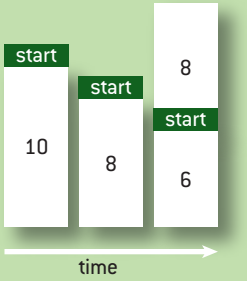
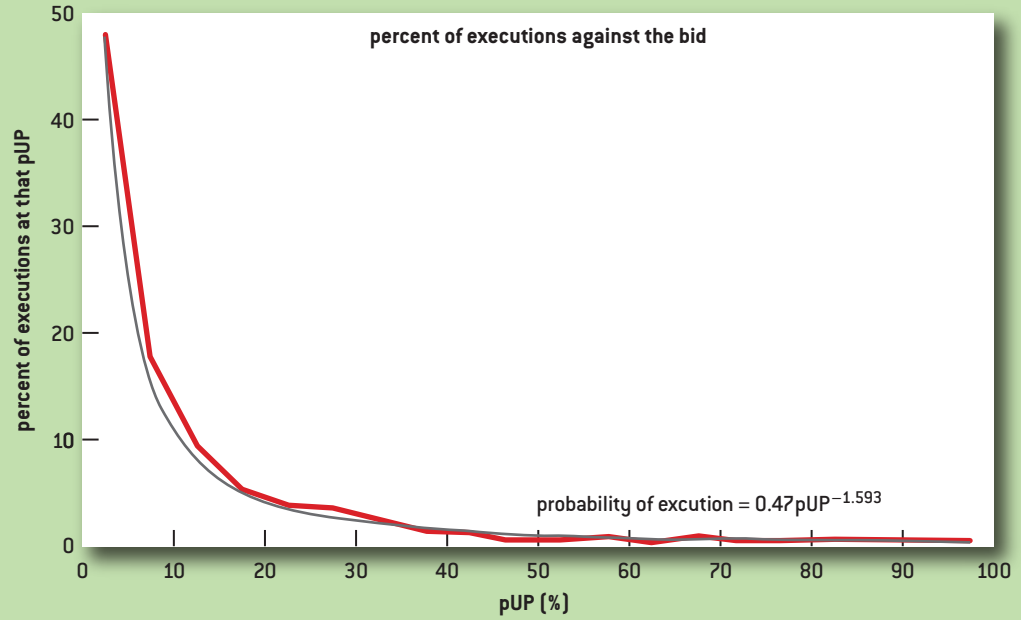


FIGURE 8

The Trading Rate



the order, and eight shares behind it. The longer an order is in the queue, the more likely it is to get to the front. The speed at which an order gets to the front is a function of two things: the rate of trading (trades take orders off the front of the queue); and the rate of cancelling of other orders.

TRADING RATES

Trading rates are somewhat difficult to estimate, but there is a clear relationship between the probability an order will be executed and the ratio of its queue size to the opposite queue size (e.g., the bid queue size versus the ask queue size). This is shown in figure 8 as pUP.

CANCEL RATES

If trading rates are hard, then cancel rates are even harder. The question is, given your place in the queue, what is the probability a cancel will come from in front of you (thereby allowing you to move up)? This is very difficult to estimate, but our own trading and some historical data provides the basis for an engineering estimate.

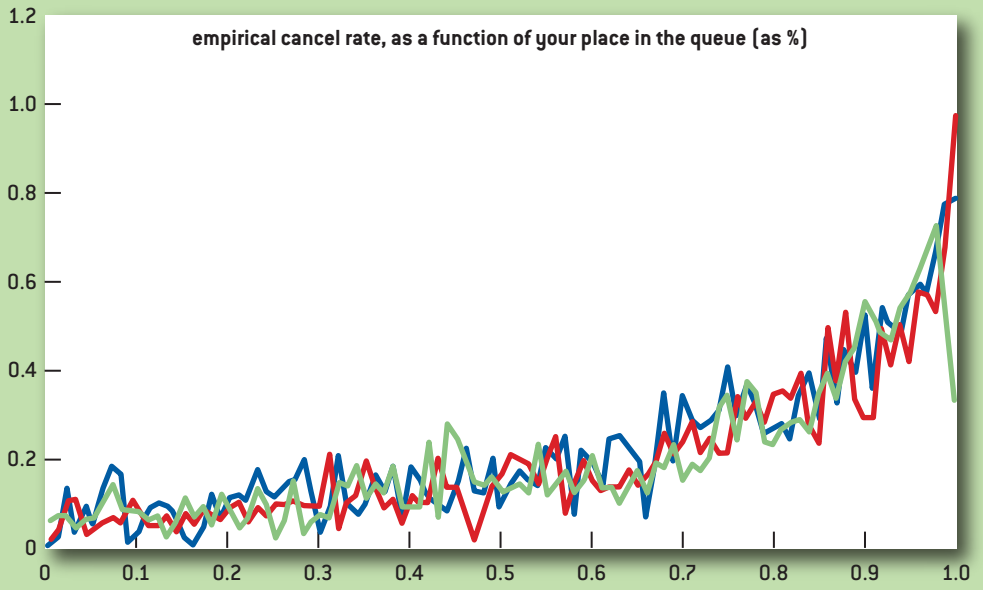
To start, we know that if we are in the back of the queue, the probability of a cancel coming from in front of us is 100 percent. If we're in the front, it's 0 percent.

Figure 9 is a chart of the empirical percentage of time a cancel comes in front of your order; it is a function of the percentage of orders that are behind you in the queue. For example, if you are at 0 on the x-axis, then you are at the very front of the queue, and cancels must come from behind you. The key takeaway is this: The closer your order gets to the front of the queue, the less likely an order in front of it will cancel (so your progression slows).

We often say you get to the front of the queue via two methods: *promotion,* which is a second-level

FIGURE 9

The Empirical Cancel Rate



queue becoming a first-level queue, and *joining,* which is, exactly as it sounds, joining the newly created queue. Figure 10 illustrates the difference between promotion and joining.

PROFIT AND THE BIG QUEUE

If your order is executed on a large queue, then you have a free option to collect the spread. If one of your orders on the opposite queue is executed, then you collect the difference between the price at which you bought the asset (bid side of $9) and the price at which you sold the asset (offer side of $10).

In the event the queue you were executed on gets too small, you can *aggress* the order that was behind you. This means crossing the bid/ask spread and forcing a trade to occur. If you get executed *passively,* you are aggressed upon by another order sitting on a queue. As long as another order is behind you, you can *unwind* the trade, meaning you can aggress the order behind.

Aggressively unwinding a trade is called *scratching* the trade. You didn't make a spread; you didn't lose a spread. It's a zero-sum trade.

EXCHANGE TECHNOLOGY

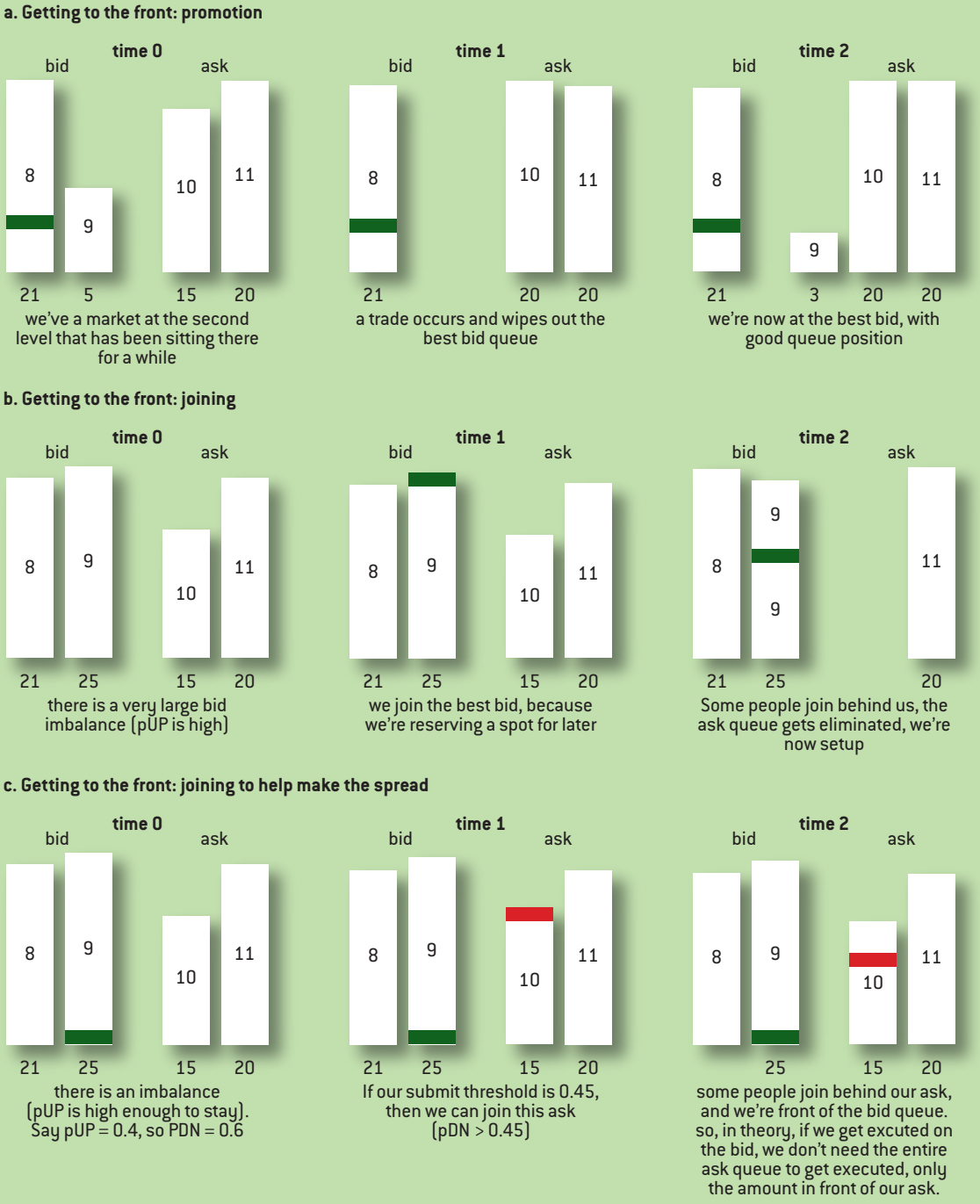
An exchange is the collection of systems that facilitate the electronic execution of assets in a centrally controlled and managed service. Today, exchanges are in a fight to offer faster and faster trading to their clients, facing some of the same latency issues as their newer HFT clients.

COLLOCATION

For exchanges, collocation can be an invaluable source of revenue. The more incumbent exchanges

FIGURE 10

Two Methods of Getting to the Front of the Queue



run their own data centers (such as NYSE and Nasdaq), and customers pay collocation fees to the exchanges directly. Newer exchanges must collocate in third-party data centers that house financial customers (for example, Equinix's NY4 in Secaucus, New Jersey).

When exchanges operate inside third-party hosting facilities, they are subject to the same power and cooling costs as their HFT brethren. As such, many exchanges focus on delivering high-throughput systems using standard x86 designs.

NETWORKING

Exchange networking is as challenging as HFT networking but also has a deeper focus on security. Information arbitrage, or the practice of gaining information about a market that is not directly intended for the recipient, is a major concern. An example of information arbitrage is when an exchange participant "snoops" the wire to read packets from other participants. This practice is easily thwarted with deployment of VLANs for each client.

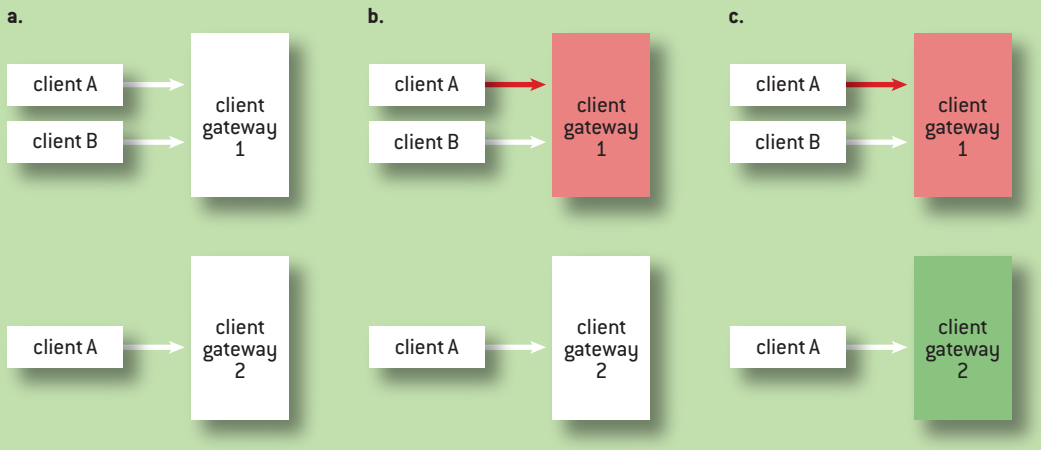
Most exchanges still deploy 1GbE at the edge. This is both a function of history (change management in an exchange is a long process) and practicality. By putting 1GbE edge networks in place, the exchange can protect itself somewhat from the onslaught of messages by both limiting the inbound bandwidth and adding a subtle transcoding hit. For example, a 1GbE Arista 7048 has a three-µsec latency for a 64-byte frame, which is a 350-nsec latency for the same frame on a 7150 (10GbE).

GATEWAY

The gateway is the first exchange subsystem that client flow encounters. Gateways have evolved over the years to take on more responsibility, but at the core they serve as feed handlers and tickerplants. The gateway receives a client request to trade (historically in the FIX format, but as latency became paramount, exchanges have switched to proprietary binary protocols). It then translates the request to a more efficient internal protocol, and routes the request to the appropriate underlying exchange matching engine.

The gateway also serves as the first line of defense for erroneous trades. For example, if a client attempts to buy AAPL for $5,000 (whereas AAPL is offered at $461), the gateway can reject the order.

FIGURE 11 **Gaming the Gateway**



MARKET DATA GATEWAY

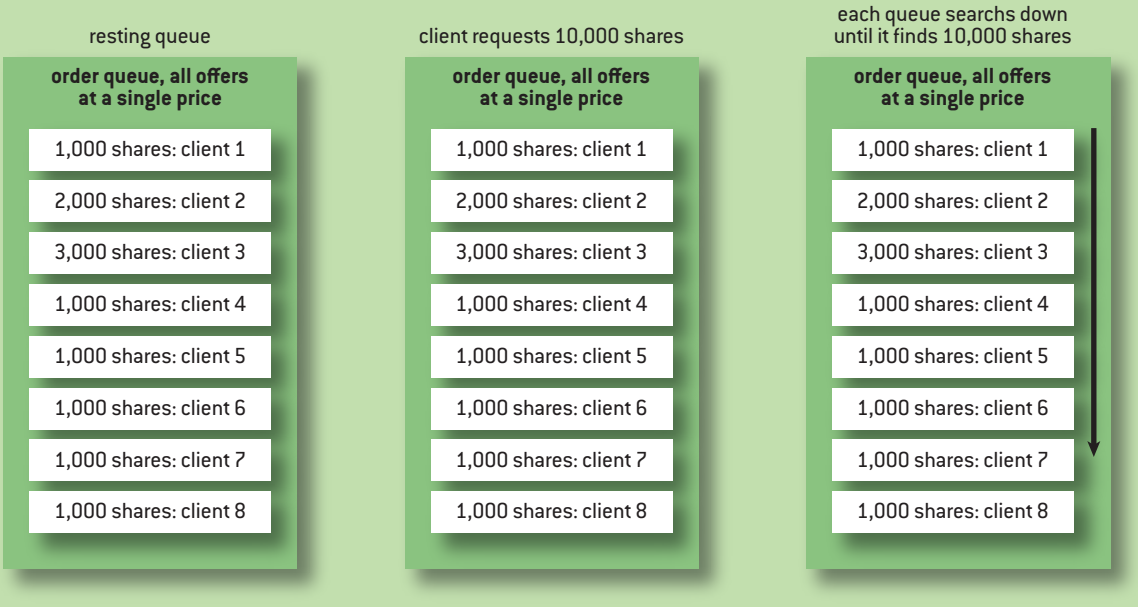
Traditionally the Order Gateway (which receives client requests to trade) and the Market Data Gateway (which distributes market-data feeds) are two separate systems, and often two separate networks. For market-data distribution, two methods are common: UDP (User Datagram Protocol) Multicast for collocated customers, and TCP (Transmission Control Protocol) for noncollocated customers. Customization takes place here as well (for example, Nasdaq's SoupTCP[9]). In some markets (for example, FX), all market data is distributed over TCP in FIX (Financial Information Exchange). For the other markets, data is often distributed over UDP in a binary format or an easy-to-parse text format. The predominant two binary formats are ITCH[6] and OUCH,[8] and both sacrifice flexibility (fixed-length offsets) for speed (very simple parsing).

Gateways are often shared across customers, as a gateway for each and every exchange participant would likely require a massive data-center footprint. As such, gateways must be closely monitored for malicious manipulation. An example of gateway "gaming" is shown in figure 11. In figure 11a, client A is connected to two distinct gateways. In 11b, Client A induces extreme load on Gateway 1, causing Client B traffic to slow. In 11c, Gateway 1, not under load, slows all attempts for Client B to cancel resting markets. Client A has an advantage with the self-made fast path.

MATCHING ENGINE

The matching engine is the core of the exchange, and like its HFT cousins is fairly straightforward. A matching engine, shown in figure 12, is a simple queue management system, with a queue of bids and a queue of offers. When a customer attempts to execute against a queue, the exchange searches the queue until it reaches the requested size and removes those orders from the queue.

FIGURE 12 **Matching Engine**



The difficulties arrive in determining who receives notifications first. The aggressing party does not know (for certain) it has traded 10,000 shares until receiving a confirmation. The passive parties do not know they've been executed until they receive a confirmation. Finally, the market as a whole does not know the trade has occurred until the market data is published with the new queue. Problems such as this are becoming increasingly more difficult to solve as we move from milliseconds to microseconds to nanoseconds.

CONCLUSION

The world of high-frequency trading is rich with problems for computer scientists, but it is fundamental to the new marketplace of automated trading, which is responsible for the majority of transactions in the financial markets today. As HFT moves from milliseconds to microseconds to nanoseconds, the problems becoming increasingly more difficult to solve, and technology must strive to keep up.

REFERENCES

1. Arista Networks. 7124FX Application Switch; http://www.aristanetworks.com/en/products/7100series/7124fx/7124fx-development.
2. Arista Networks; 7150 Series 1/10 GbE SFP Ultra-Low Latency Switch; http://www.aristanetworks.com/en/products/7150-series/7150-datasheet.
3. CBS News. 2010. Robot traders of the NYSE. Sixty Minutes Overtime; http://www.cbsnews.com/video/watch/?id=6942497n&tag=related;photovideo.
4. Millar, M. 2011. "Lightning fast" future traders working in nanoseconds. BBC News; http://www.bbc.co.uk/news/business-15722530.
5. Moyer, L., Lambert, E. 2009. Wall Street's new masters. Forbes (Sept. 21): 40–46; http://www.forbes.com/forbes/2009/0921/revolutionaries-stocks-getco-new-masters-of-wall-street.html.
6. Nasdaq. 2013. Nasdaq TotalView-ITCH 4.1; http://www.nasdaqtrader.com/content/technicalsupport/specifications/dataproducts/nqtv-itch-v4_1.pdf.
7. Nasdaq. OMX Co-Location; http://app.qnasdaqomx.com/e/es.aspx?s=453941583&e=9032&elq=4824c6a202f34d00a5e586d106f64cc8.
8. Nasdaq. 2012. OUCH Version 3.1; http://www.nasdaqtrader.com/content/technicalsupport/specifications/TradingProducts/NQBX_OUCH3.1.pdf.
9. Nasdaq. SoupTCP; http://www.nasdaqtrader.com/content/technicalsupport/specifications/dataproducts/souptcp.pdf.

LOVE IT, HATE IT? LET US KNOW

feedback@queue.acm.org

JACOB LOVELESS is the CEO of Lucera and former head of High Frequency Trading for Cantor Fitzgerald. He has worked for both high frequency trading groups and exchanges for the past 10 years in nearly every electronic asset. Prior to a life in finance, Mr. Loveless was a special contractor for the US Department of Defense with a focus on heuristic analysis on things which cannot be discussed. Prior to that, he was the CTO and a founder of Data Scientific, a pioneer in distributed systems analysis.