Determinism is the new latency

It's hard to think of an execution service for which latency doesn't matter at all. There are, however, any number of situations in which latency does not need to be minimized – merely controlled and understood.

Every entity involved with trading should care a lot about latency, but perhaps not the way you'd immediately think. While "Electronic Trading" and "latency-sensitive" are related, it's clear that not all traders who trade with a high frequency are latency-sensitive; equally, not all traders who are sensitive to latency trade at a high frequency. In fact, nearly all traders, automated or not, are latency-sensitive, and therefore so are service providers – exchanges, routers, brokers and clearers.

To illustrate this, consider that an event trader's response to a GDP estimate might only occur quarterly but, since every other trader has access to that information, the time taken to respond to it is absolutely critical. These are therefore low-frequency, but extremely latency-sensitive strategies.

Execution services are also low-frequency participants – orders arrive infrequently from investors, when compared with automated traders. However, as was so emphatically argued in "Flash Boys," each of those orders must be placed on multiple trading venues within a very small and precise window of time in order to avoid giving other traders an arbitrage opportunity. Execution services are low frequency, but extremely latency-sensitive.

In fact, it's hard to think of an execution service for which latency doesn't matter at all, if not for the performance of the service, then the fairness of the service. There are, however, any number of situations in which latency does not need to be minimized – merely controlled and understood. The idea of latency control, rather than latency minimization, gives rise to a new mantra: Determinism is the new latency. Even if a minimal delay isn't important, yet the control and consistency of that delay are critical.

Let's look at another group of organizations that should care more about determinism than latency reduction: stock exchanges. If exchanges, dark pools and other venues are attempting to be fair to all participants, as they should be, then their technology must ensure that an order that is sent to the exchange first, is executed first. After the order is entered, the time taken to get a response is important to many trading organizations, but not critical to fairness.

IEX's "speed bump" is a great example – the venue introduces approximately a 350us delay using a highly deterministic solution: a long length of optic fibre. Every order takes exactly the same amount of time to traverse the fibre. Similarly, the size of the delay in sending orders to other venues is not important, but it is extremely important that the latency is consistent and known (i.e., deterministic) so that forwarded orders can be delivered simultaneously, avoiding exposure to arbitrage.

Delay determinism and delay reduction are linked, but the relationship is not an easy one. Often, the best way to reduce the average latency is to provide a "fast path" mechanism. Computer designers understand this idea well – they provide a mechanism to improve the performance of common tasks, but leave infrequently executed or hard-to-implement tasks to the "slow path." This makes perfect sense when optimizing for bandwidth, and does great things for average latency, but it destroys a system's determinism because the worst-case delay via the slow path is now very different from the best-case delay via the fast path.

Caches are a great example of such a technique – all modern microprocessors implement a "cache," which temporarily stores recently or frequently accessed information from RAM. It's usually implemented using some very high-performance memory on the same piece of silicon as the processor, which allows the processor to get access to cached information very quickly if it has been used recently. This works pretty well if the data is used regularly, and computer performance engineers work hard to ensure that as much of the frequently used data as possible will fit inside the cache.

However, this does not improve the worst-case performance of a system, since information that is not stored in the cache takes orders of magnitude longer to access than information that is stored there. This means that nearly all software systems exhibit "long tails" – the worst-case performance is orders of magnitude worse than the average case, or the frequently executed case.

The same is true of network switches, which must cache commonly used routing information in order to improve their average latency and total bandwidth, but which therefore introduce non-determinism: If a required route is not cached, the latency will be much higher than if it is. General-purpose network devices are designed for bulk data transfer; and for transferring a file, caching is fantastic – a lot of messages with the same source and destination addresses will be sent in a short timeframe. But for a trader who sends one packet per quarter, the caching mechanisms in the switches do not help – they will always hit the worst case latency of the switches.

This obviously affects the fairness of trading venues – should participants be forced to trade regularly simply to keep the caches in the network "hot"? What about the caches in the exchange? Or the order router? Should participants need to understand these subtleties, or should the venues themselves be able to provide guarantees?

A similar tension between latency reduction and determinism is seen in the internal implementation of the venues. Most will use more than one server to implement their matching engine and order gateways. Often, exchanges are stratified into tiers – order gateways accept messages from participants in the market, check them for validity, and then forward those messages to an appropriate matching engine. There may be multiple order gateways forwarding to multiple matching engines. Gateways may be allocated to participants, and herein lies a new source of non-determinism: contention. If the time required to process an order in a gateway is not deterministic, it is possible that two orders that are sent to the exchange in one sequence may be executed in a different sequence. If a particular gateway is processing more orders than another, the delay through that gateway may increase, and so some participants may have an advantage over others. The savvy latency-sensitive firm therefore has to connect to many gateways to ensure that it is always able to send orders to the one with the best performance.

It's hard for many participants to understand what's going on. So much complexity and variation in performance makes it a real challenge to understand what happens to an individual trade.

Often the trade that matters the most is the one that happens rarely – some event occurs, resulting in a flurry of activity. It is during these market bursts that performance really matters, but it's also the time when buffers fill, queues lengthen, and determinism goes out the window.

A surprising amount can be done to eliminate this non-determinism.

- There has to be a way to see what's happening. That means good quality network monitoring and hardware-based time stamping, so that the amount of time taken for a message to traverse the trading stack and the response can be traced and analysed. If the time taken by each process is measured, the information can be used to detect components that are introducing non-determinism and the conditions under which this happens.
- There are technological solutions to these problems. FPGA technology and Layer-1 switching allow us to avoid the pitfalls of the one-size-fits-all fast-path. By nature and design, these technologies allow customized solutions that fit the financial services industry.
- There must be an onus on vendor disclosure of the determinism of their products. Many vendors will provide average latency numbers, since this is what a more typical data centre requires; but the worst-case numbers, which are more important for financial services, are omitted. Benchmarks are often constructed to measure the latency when transmitting thousands or millions of packets per second, masking the performance of the once-per-day message that really matters.

We are forced to work with what we have available. Rightly, many network device vendors must cater for markets larger than financial services. Those markets require products that are designed to deal with large volumes of data, rather than providing low, deterministic latency. Financial markets flip that optimization on its head. The upshot is that financial markets require infrastructure designed specifically for the purpose.

These determinism challenges can, and should be solved. Given the desire, these challenges can be met.

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390
Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062