

Open Debate on Latency Numbers is 'Healthy', But Just How Fast is Fast?

By Simon Garland, Chief Strategist at Kx Systems

Everyone is keen to climb on the low latency bandwagon. This is due to several factors, not least the exponential growth in market and trade data volumes that exchanges and financial institutions are having to process. Others include the growth in derivatives and algorithmic trading, the scores of new assets and asset classes being traded, and the increase in the number of players in the financial markets, particularly in emerging economies.

Existing systems have difficulty coping. And the extent of the problem is only likely to increase given the projected exponential increases in data volumes. In 1998, the NYSE published trade and quote (TAQ) data at the rate of 5 million records per day. Just 10 years on, this has increased 100-fold to more than half a billion records per day.

Speed – of processing – and plenty of reserve capacity are key to being able to deal with data. Industry figures often say they have new servers, all the lights are on, that they're running at close to 100 per cent CPU and everything looks fine at the moment.

However, if anything happens – a particularly volatile day on the markets, for example – these servers are no longer going to cope. This is not conjecture; this happened several times last year to systems distributing and collecting market data. The chances are that there

are plenty of other instances that we don't get to hear about.

There are two distinct problems that need to be addressed. The first is how to measure latency; the second is how to reduce it, having measured it. Assuming that it has been measured and the source of the problem has been identified, what's the solution?

The debate continues about how best to save milliseconds or even microseconds, but there is a lack of benchmarking of complete systems or even a standard definition of how and where to measure latency. One might even say that there are some who are being creative with their latency numbers.

This 'creativity' is partly understandable, as there are so many factors that need to be considered in order to achieve any meaningful benchmarking. It is healthy to have an open debate about the issue and I applaud independent benchmarks and analysis of vendor performance, such as the STAC initiative, an independent group sponsored by IBM, Intel, Kx and others.

In its recent benchmarking study on latency, Stockholm-based trading platform supplier Cinnober defined three aspects of measuring latency: End-to-End latency, Response Time latency and Business Logic latency (See Cinnober's Latency benchmarking Study on www.a-teamgroup.com 23/01/2008).

Cinnober defines End-to-End latency as mostly hardware-related. Response Time latency might relate to a feed handler, for example. And Business Logic Latency measures the whole process from beginning to end.

Not surprisingly, the conclusion of the report was that Business Logic latency was key. Cinnober is one of the first organisations to point out and emphasise its importance. This is surprising given that this is probably the most important aspect of latency, but one that rarely gets a mention.

Impartial measurement is vital. Comments such as, 'Yes, it's great on speed, but it falls over all the time compared to X or Y', indicate the 'mean time' latency to failure. But they do not provide an objective measure that can be used to compare between systems.

Even if that problem can be solved, how does an institution choose a vendor? Selection can, at times, be as simple as the word getting around that somebody based in Canary Wharf had bought boxes from XYZ and that they worked pretty well. Before you know it, the word has spread to the City, and from there to Wall Street.

This is an inefficient and ineffective way of making a decision. I am very much in favour of organisations such as STAC subjecting all the major suppliers to the same set of standardised benchmarks.

A decade ago, when a bottleneck was discovered the solution was to throw money and servers at the problem. But is that still feasible?

There are many reasons for bottlenecks. Low-level, 'mundane' bottlenecks – such as those presented by a feed handler – can be fairly straightforward to resolve.

So, if it's that easy, why not just put in a hardware-based feed handler, or provide a more powerful server? The trouble is that when that's been done another bottleneck almost invariably rears its ugly head.

An institution may be running the fastest and most efficient feed handlers, but its servers may be running a standard Linux distribution. We've seen an improvement in a number of cases when a switch is made to, for example, the real-time version of Red Hat. It's not as though an institution needs to have its private team of kernel hackers; today one gets pretty standard distributions, which can make a significant difference.

And, when the going gets tough, these versions of Linux still allow the data to continue to stream through, which is what we want.

Then there is the hardware solution.

There is no shortage of hardware on the market, but is this the right solution? There has been a 100-fold increase in the output of the NYSE in only 10 years, and that pace is accelerating. Over the same period, CPUs have become only two to four times faster. Grid computing and dual and quad-core processing are regarded as the way forward in some quarters.

But, while the power may be there, most software has not been optimised to run on grid or even quad core CPUs and is unable to take maximum advantage of the processing power.

The current way of doing things is fast running out of road. What is needed is some original thought and some different approaches to the problem. The requirement is for speed, accuracy, resilience, design optimisation for software and hardware for multi-core chips, availability of support, and speed of development. While all this sounds obvious, it isn't easy to come by.

There is a real gap developing between top tier firms, which are pulling out all the stops to increase capacity and reduce latency – such as adding hardware and writing custom code to

perform very sophisticated calculations and execute strategies – and others that are using shrink-wrapped solutions and struggling to keep up.

Proper multi-threading support – making effective use of all the available cores – is part of the solution. The largest financial firms, those handling billions of time-ordered events a day, want and need to obtain instant intelligence from their streaming data.

Multi-threading allows the whole machine to do things in parallel, saving time, for example with VWAP (volume weighted average price) calculations, or to retrieve history from multiple price feeds.

It is important to consider the best use of hardware and to run on multi-core machines only software that has been optimised to take advantage of that multi-core architecture; otherwise, performance levels may decline.

While some may resist in the short term, I can envisage, and look forward to, a time when it will be possible to have all vendor systems tested for Business Logic latency across industry standard benchmarks against their competitors.