

# Cinnober on: Latency

November 26, 2007

# Low Latency Is All the Rage Today

Low latency is all the rage today—on the lips of our colleagues and in the press. But a claim one millisecond is actually without meaning, unless one knows how that one millisecond was, indeed, measured.

One of the main purposes of this document is to provide you with the results of Cinnober's research on latency, which we arrived at by carefully measuring latency using Cinnober's TRADExpress in a real-life simulation, and which resulted in establishing a benchmark for latency.

## Simplifying the Mystery of Latency Measurement - A Primer

When a vendor declares that his or her product can provide your firm with one-millisecond latency, what, exactly does it mean?

- Does the vendor mean peak latency?
- Average latency?
- Do they even know how it is defined or measured?

## Why is it Important to Measure Average Latency?

In a real-life situation, orders will not arrive to the trading engine in an evenly-spaced pace but instead in bursts. This applies to many other real-life situations, e.g., phone calls arriving at switches, just as it does to orders arriving to a trading engine.

The fact that orders arrive in bursts affects the latency. The mathematical theory dealing with this phenomenon is called Queuing Theory. One of the predictions of the theory is that the latency experienced by a stream of incoming orders varies in accordance with an exponential distribution.

Therefore, to avoid false claims of low latency, it is important when measuring latency to choose a statistic that truthfully summarizes the measurement. Cinnober uses average latency.

## How Does One Define a Method for Measuring Latency? Three Possible Ways

The industry has a number of different ways of defining and measuring latency. Here we describe three of the most widely-used methods—end-to-end latency, response time latency, and business logic latency—and clarify the differences between them. The key difference between the definitions is the segment of the total transaction chain—from a market participant's computer to the marketplace and back that is taken into account.

Please note that there is no general agreement in the market on the use of these three definitions of latency. Different marketplaces and vendors may assume different definitions.

### **End-to-End Latency**

End-to-end latency is defined as the total latency experienced by market participants, from their own computers to the marketplace and back.

There are specific problems measuring end-to-end latency. One is that the computers at the participant site may not be controlled by the marketplace and may therefore not be uniform in terms of CPU power and usage.

Another problem is that the latency measured depends on the network routing to the marketplace. For participants who are neither located in the vicinity of the marketplace, nor in the same metropolitan area, the network communication latency may well be larger than any other latency component.

### **Response Time (Door-to-Door Time)**

Another latency measurement is the time lapse from the moment an order enters the marketplace to the moment the response is sent back to the participant. This number is often called response time or door-to-door time, and excludes the participant-to-marketplace latency.

This response time measurement has fewer unknown factors than end-to-end latency. However, it is important to know exactly where the measurement is made. For example, is it just upon entry into the gateway, or is it after having unpacked the incoming message? When chasing fractions of milliseconds, this can make a difference.

### **Business Logic Latency**

Sometimes the time spent processing the business logic is used as a latency measurement. The number is interesting since it consists of the application logic and is generally difficult to shrink except by upgrading hardware.

The business logic latency also often corresponds to the component where one cannot multithread any further. That is, normally the business logic in the core engine, for a single instrument, runs sequentially on one single CPU and therefore limits the number of updates per second in a single "hot" instrument.

### **How Does Latency Depend on Business Processing?**

The latency depends on the incoming message and what the resulting actions are. What business processing is associated with the latency measurement?

An order that is entered may or may not update the best price(s). An order that does not update the best price will result in less processing than an order that does. In the latter case, there will be additional processing to generate public

broadcasts and broadcasts to market data vendors. Further, the order may trade in which case additional processing happens.

In today's world of program trading, it is often possible for a participant to cancel all outstanding orders for a product. Such a single transaction can lead to many orders simultaneously being pulled from the market and many associated broadcasts being generated.

Another factor when thinking about latency is that, when comparing latencies between different marketplaces, it is seldom stated what type of action or transaction is being measured. This practice obscures the facts and makes impossible the analysis of latency differences.

### **How Does Latency Depend on the Choice of Transaction Model?**

There is generally a trade-off between latency and data integrity. Yet another factor affecting latency, no matter where it is measured, is what type of transaction model is used by the marketplace. Some marketplaces have models where the response is sent back as soon as the transaction/order entry has been received, but before all the business processing and broadcast messaging have been completed. Naturally, such a model will report lower latency times than a model where all processing associated with the order has been completed before sending the response.

When Cinnober measures latency, all processing is completed before sending the response. This means that all business processing as well as all associated broadcasts and market data interaction is completed. The advantage of this is that the result of the transaction is available to other market participants (through the broadcast) at the same time as the response is sent back.

Different marketplaces may have different levels of data integrity. The main question from a latency perspective is whether the order is secured to disk, to a standby system in a disaster site or to both of these.

Cinnober believes there needs to be flexibility for the customer in choosing the level of data integrity. In this document we present measurements made both with synchronous and asynchronous disks. In the synchronous case, each order is written to disk before committing the response. In the asynchronous case, the disk is allowed to complete the write asynchronously. In both cases, the disk write always happens.

### **How Does Latency Depend on the Choice of Hardware and Network Architecture?**

The power of the hardware selected is a very important factor in determining the latency of the resulting system.

TRADEExpress is 100% Java and can be deployed on a wide range of hardware architectures and associated operating systems.

Another important piece of the technical architecture is the disk subsystem. With full data integrity, each transaction is safely stored to disk before releasing the response. Because disks are relatively slow devices, the disk write is normally the critical component to achieve low latency.

TRADEExpress has no need for any special disk configuration such as redundant disks between sites. Therefore, the most optimal high speed disks can be selected.

The backbone network architecture is also important. If the bandwidth between the primary and standby sites is limited, then this will impact overall latency.

### **How Does Throughput Affect Latency?**

In the best of worlds, one would like both ultra-low latency and extremely high throughput. However, when going to high throughput numbers there is an upwards effect on latency, since it is more likely that queuing effects occur. Therefore, it is important to measure latency at the system's current or planned throughput levels.

In order to achieve ultra-low latency, the CPU in the system boxes cannot be run at high loads. The system will work fine, but there may be queuing and hence some latency effects.

# A Cinnober Study to Establish a Performance Benchmark for Latency

One of the main purposes of this document is to provide you with the results of Cinnober's research on latency, which we arrived at by carefully measuring latency using Cinnober's TRADExpress in a real-life simulation, and which resulted in establishing a benchmark for latency.

## How Cinnober Defines Latency

Cinnober uses response time (door-to-door) latency, as described earlier in this document. Specifically, we measure latency from the time a transaction has been received at the outer system boundaries until the response and result is sent back to the originator at the outer system boundaries. (Refer to the diagram on the following page.)

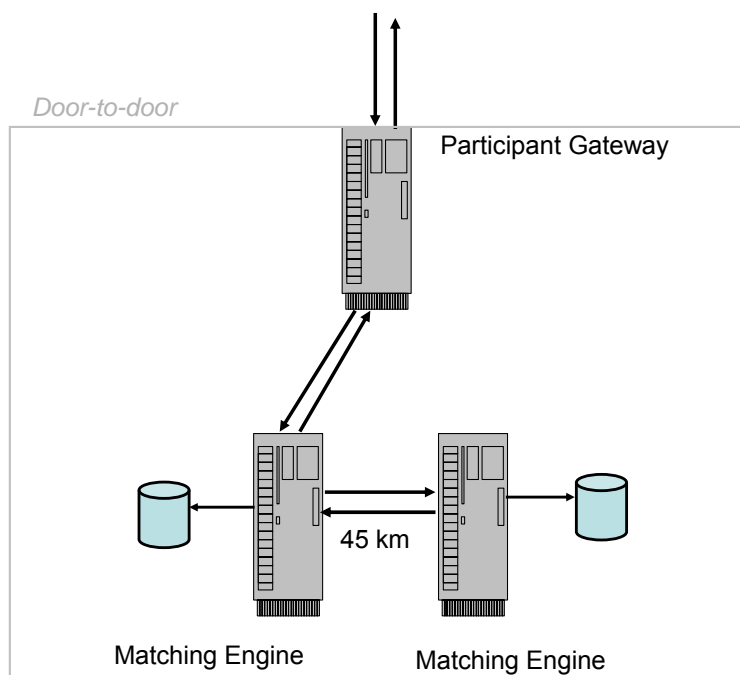
We average these measurements to arrive at final latency number, which will be stated in this section.

## Exactly What Processes Does Response Time Latency Include?

In the TRADExpress Trading System, the response time latency includes the following processes:

- At the participant gateway at the marketplace's site:
  - The request being received from the participant application is unmarshalled and validated.
- The transaction is marshalled and is routed to the appropriate matching engine partition, depending on its data contents. There are multiple matching engine partitions, each handling a subset of the market.
- At the Matching Engine:
  - The request is received from the participant gateway.
  - The request is unmarshalled.
  - The request is recovery-logged. The recovery logging is done to sequential flat files. The tests are run both while waiting for the disk (synchronously), and allowing the disk to potentially complete later (asynchronously).
- The transaction is replicated to a standby matching engine. The standby matching engine is a hot-standby server that immediately takes over, in the event that the primary matching engine fails.
  - After that, the request is processed from a business logic point-of-view.
  - The request and results are audit-logged to sequential flat files.

- The result of the request is marshalled and sent back to the participant gateway after receiving confirmation that:
  - the transaction has been physically saved to disk.
  - the standby matching engine has confirmed receipt of the request.
- Finally, the participant gateway:
  - receives the result from the matching engine,
  - locates the participant session on which the initial request was received, and
  - posts the response back to the participant.



A diagram of the path of each transaction message. Cinnober defines latency as the time it takes a transaction message to complete this path, from its arrival at the participant gateway, to its exit through the participant gateway.

## The Test Environment

To establish a benchmark for latency, we measured latency under the following conditions. We employed two sites: one primary site and a secondary, standby site. The distance between the sites was 45 kilometers (28 miles), with a one Gigabit/second fiber connection.

The primary site hosted all primary matching engine servers and the participant gateway servers. The secondary site hosted the hot-standby matching engine servers.

The market was divided into 8 partitions. In total, 16 servers were used to host the primary and secondary matching engine processes. In total, 8 servers were allocated to host the participant gateway server instances. However, only the 4 machines on the primary site handled the participant interaction.

All participant applications were located within the primary site.

### Hardware

All servers used—the matching engine as well as the participant gateway servers—were of the same model: SUN x4600 M2 with AMD 4 dual core 2.8 GHz, 16 GB memory with a Qlogic HBAs.

At each site there was a SAN: SUN StorageTek 6540, 4 ports x 4 GB. The physical disks being used were 146 GB, 15,000 rpm.

### Software

- Operating System: SUN Solaris V10 Update 4
- Java JRE V1.6.0-2
- TRADExpress V 6.0

### Network

All network connections were 1 Gigabit/second.

## Tests and Results

Two different types of tests were conducted, a concentration test and a distribution test.

### The Concentration Test

We call this test the concentration test because its objective is to simulate what happens when the system is bombarded with orders wanting to buy/sell a single instrument. How will the system cope with this concentration of traffic?

The test conditions were that:

- Twenty (20) participants inserted order into one partition and one single order book.
- Orders were injected with the following mix:
  - 50% insert order
  - ~ 25% cancel order
  - 25% updates
- Of the insert and update transactions, 50% resulted in an execution.

### Results of the Concentration Test

Single partition, single order book:

Order events/s	Disk	Users	Orderbooks	Average latency
1,200	Asynchronous	20	1	0.700 ms
1,200	Synchronous	20	1	2.900 ms

Please notice the difference between running with the disk asynchronously compared with synchronously.

In the asynchronous case, the order was replicated to the standby site before the response was committed. In the synchronous case, the order was replicated and secured to disk.

### The Distribution Test

The objective of the distribution test was to benchmark the latency of the TRADExpress Trading System during a time of fairly normal use. Specifically, we simulate traffic across a number of instruments, with orders inserted into a large

number of order books distributed over all partitions. Since this simulates what most days are like, this is also interesting to measure.

The test conditions were that:

- One hundred thirty (130) participants inserted orders in 1,000 order books, equally distributed over 8 partitions.
- Orders were injected with the same mix as for the concentration test, above.
- As part of the test, 50,000 National Best Bid and Offer (NBBO) updates were entered into the system. Each matching engine consumed one eighth of the updates, i.e. 6,250 updates/second, and updated their caches with NBBO prices.

#### Results of the Distribution Test

8 partitions; 1,000 order books; 50,000 NBBO updates:

Order events/s	Disk	Users	Orderbooks	Average latency
9,600	Asynchronous	130	1,000	1.628 ms
9,600	Synchronous	130	1,000	3.247 ms

#### Miscellaneous Observations

The logical volumes to which matching engine recovery logs and audit logs were written had the following I/O performance characteristics:

- 2,400 synchronous writes/second, with a record size of 2 KB.
- The network latency on an application level of sending 1,400 bytes from the primary site to the secondary site and back again was 0.670 ms, on average.

# In Conclusion

## Transparent Numbers

The goal of this study was to provide you with the most relevant, useful and transparent definition of latency—tested in a real-world environment. The latency numbers we report comprise the time from the moment the transaction message enters the border of our system, until the time it leaves our system.

Technology has played and will play an even more critical role in answering increasing demands for transaction speed and cost efficiency. This will be significant across all trading areas and all asset classes, and may be especially significant for the low-touch, low-value arbitrage. This will lead to a race between the different marketplaces—a race partly based on speed, partly based on the flexibility and sophisticated service options the different solutions can offer.

Traders will benefit by being able to take advantage of changes in the market without being limited by technology. They will be able to immediately react and respond to information, thus making trading strategies even more profitable. On the opposite end of the spectrum, a delay might mean that execution fails. It is all about the transferring of risk.

## Moving Forward

Cinnober is continually working on improving our latency numbers, and we will continue to release new technology in this area. We ever strive to push the limits of latency.

We would like to be your technology partner, and we will commit to being transparent, innovative, and ahead-of-the-curve.

Cinnober increasingly receives requests for high throughput and low latency, which yesterday would have been considered extreme. We continuously enhance our TRADExpress technology to respond to new demands, as well as to foresee and meet the needs of tomorrow's ever-changing market.

**Passion for change** | Cinnober provides mission-critical solutions to the world's most demanding financial marketplaces.

We are passionate about one thing: applying advanced financial technology to help marketplaces seize new opportunities in times of change.

We build partnerships with our customers based on trust and transparency. We serve banks, exchanges, brokerage firms and other actors that have extreme demands on business functionality, high throughput and low latency.

We currently have product-based offerings for a number of areas, such as marketplaces, post-trade management and binary markets. All solutions use our TRADExpress™ technology, designed for scalability and flexibility.

Among our customers are the American Stock Exchange, Turquoise, Markit BOAT, the Chicago Board Options Exchange, Liffe NYSE Euronext, the London Metal Exchange, Deutsche Bank and Borsa Italiana.

We are an independent provider of marketplace solutions, and do not operate a market of our own, avoiding any conflicts of interest. We are not owned by - nor have any ownership interests in - any market operator.

Our track record says it all. We help our customers turn change into a competitive advantage.



Cinnober Financial Technology AB  
Kungsgatan 36  
SE-111 35 Stockholm  
Sweden  
+46 8 503 047 00  
cinnober.com