



Multi-Core Optimization for Market Surveillance

Maximizing Performance to Help Financial Institutions Meet their Goals



EXECUTIVE SUMMARY

Financial institutions have several key challenges when it comes to effective market surveillance, including both the quantity and diversity of the data and fast-changing data definitions. A firm might need to represent the same cash equity in several different formats when it trades on different exchanges and orders flow through different parts of the business. Consolidating all of these different message formats makes oversight possible in real time.

The high speed of financial markets makes fast response times a necessity. Next-day response is not good enough. Real-time systems enable fast response times, including through circuit breakers, which can stop a certain algorithm from trading.

This real-time message consolidation process was the focus of a proof of concept Ancoa conducted with Intel.

Ancoa* is a high-end surveillance platform for financial markets that enables regulators, exchanges, brokers, and trading desks to both protect their reputations and keep on top of regulatory requirements.

A linearly scalable architecture lets Ancoa process large volumes of trading data across both markets and asset classes. The solution is designed for fast and flexible deployment, runs either in the cloud or on-site, and comes with a regulatory update service to help it stay ahead of regulatory developments.

Ancoa contacted Intel to find ways to make further performance gains. Working with Intel Labs UK, Ancoa found significant room for improvement at each individual application instance by increasing utilization of the available multi-core technology. Ancoa and Intel conducted the analysis at the Intel data center on the latest Intel® Xeon® processor E5 product family, using Intel's performance analysis tools.

Reducing Risk

Real-time market surveillance reduces operational risk in capital markets and increases market integrity at trading venues. The overall goal of market surveillance is to increase trust and fairness and protect firms from damage to their reputations. In recent years, regulators have issued large fines for insider trading and various forms of market manipulation such as spoofing.

Faster market surveillance and circuit breakers could have significantly reduced

some specific market integrity and operational problems. For example, in the "Flash Crash" of May 6, 2010, USD 1 trillion in market value temporarily disappeared in less than 10 minutes. The market regained most of these losses 20 minutes later, but this event seriously undermined confidence in financial markets.

In another example, on August 1, 2012, Knight Capital Group lost USD 440 million in 30 minutes due to a runaway trading algorithm. In the immediate aftermath of this event, Knight faced bankruptcy and



Multi-Core Optimization for Market Surveillance

was only saved by a USD 400 million injection from Blackstone Group, TD Ameritrade, Getco and Stife/ Nicolaus. Knight could have avoided this by improved internal surveillance.

Another example where (internal) oversight failed was in 2013, when SAC Capital Advisors, a hedge fund, was indicted for USD 275 million of illegal profits through insider trading. The firm's management was blamed for insufficient checking on insider trading by its staff. Market surveillance could have automated this checking and avoided the resulting damage to the organization's reputation.

Distributed Architecture

Ancoa's architecture is linearly scalable across any number of markets, so it can facilitate cross-market and cross-asset-class market surveillance. Linear scalability, on its own, is not enough to maximize throughput and minimize latency, since each node needs optimization to achieve maximum performance.

Figure 1 shows Ancoa in operation in a distributed configuration. Each blue box represents a network node which is a single application instance, fulfilling a specific task (e.g., message routing, consolidation, indexing, analytics, request solving, or query aggregation). In this

specific example, there are six segments (columns), each handling one data feed. The distributed nature of Ancoa forms the foundation of its linearly scalable architecture, capable of handling any number of messages and data feeds.

Figure 2 shows a conceptual diagram of Ancoa. The blue circles at the top represent different types of data feeds which Ancoa handles such as trading messages, reference data, news messages, payment messages, and unstructured data (e.g. emails, direct messaging, or Twitter* feeds). This diverse range of information is consolidated inside

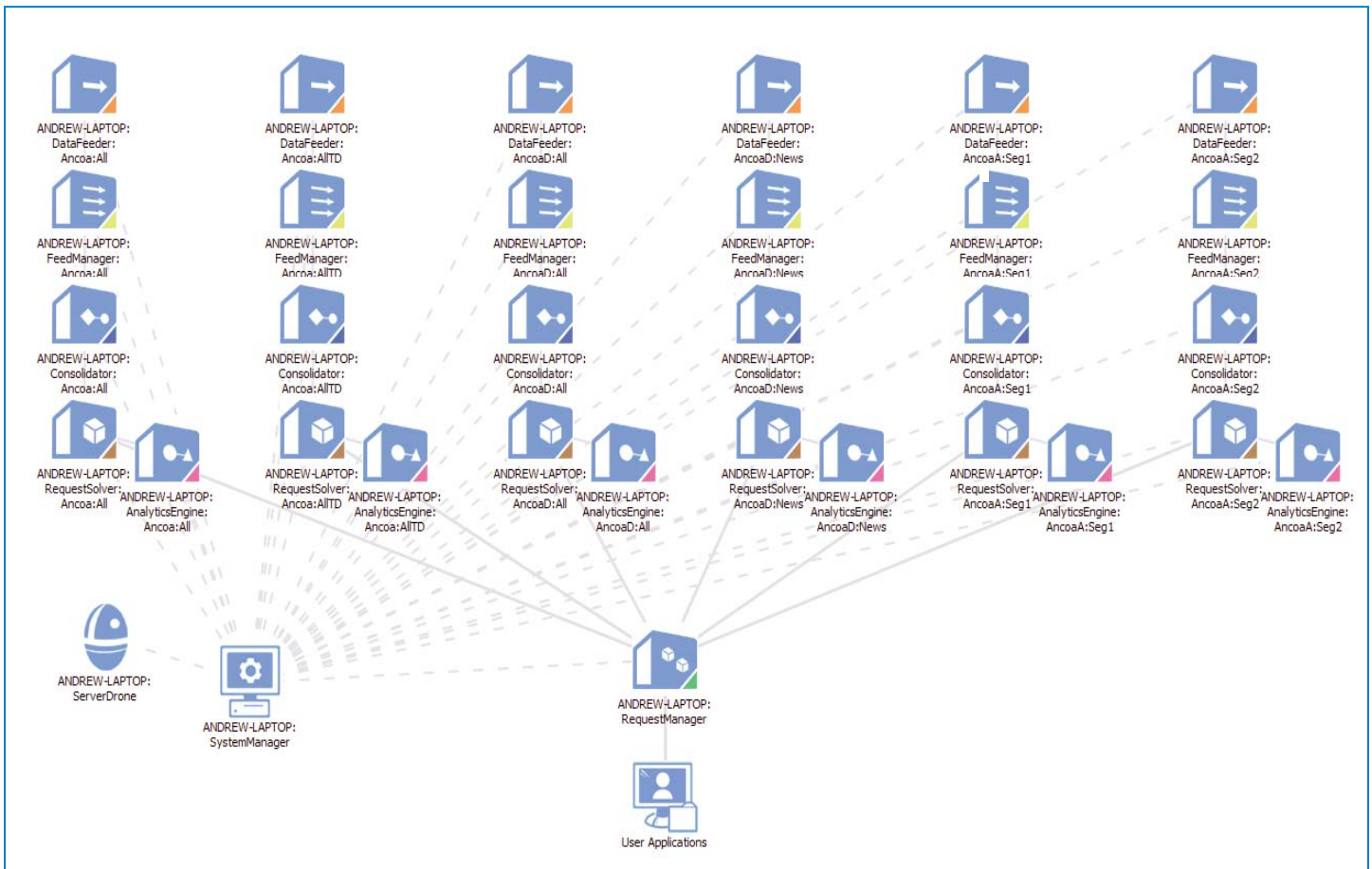


Figure 1. Ancoa Running in a Distributed Computing Network

Ancoa, the large circle, where alerts are triggered in real time based on business rules and reports are updated. The consolidated data is stored in a secure and compressed way. Stored information can be quickly retrieved, since some alerts and reports require both real-time and historical information. Detecting insider information, for example, requires a combination of real-time trading data, news data, and historical trading data to work out who potentially profited in an illegal way from a news event. Further down in the diagram, when an alert has been raised, there is a workflow that allows managing, investigating, and documenting of individual alerts. Reports can be generated ad-hoc or in an automated way. Historical data can be accessed for reporting, research, and visualization.

Challenges

One challenge in scaling up each instance of message processing is to maintain the time sequencing of messages. Using a naïve approach of parallelization would jeopardize the order of the messages after processing, which would be unacceptable.

A second challenge is removing the use of traditional operating system-based locking of data structures while they are being updated.

Ancoa software was developed in C++ for performance reasons. Since C++ needs compiling to machine code on each native platform (i.e., Windows*, Linux*, and UNIX*), any optimizations need to be as generic as possible to avoid platform-specific code, which is hard to maintain.

Ancoa devised and implemented a solution based on lock-free queues and achieved visibility of the scalability bottlenecks at the application level with the help of Intel’s engineers, using Intel® VTune™ Amplifier XE.

Based on this performance analysis, Ancoa selected message consolidation as the ideal candidate for optimization. Message consolidation is where transactional messages are normalized so they can be treated uniformly regardless of their origin, asset class, or type. Many firms still use proprietary formats, making comparing messages in real time a challenge. For securities transactions, the industry standard FIX* protocol is the most commonly used. For payment messaging, there are many different standards such as SWIFT MT* and ISO 20022*, as well as regional and proprietary formats. All messages are consolidated before Ancoa analyzes them across markets, classes, and formats.

The performance gains from consolidation were impressive. For example, the throughput increased from 30,000 transactions per second (TPS) on a single core to 140,000 TPS for the parallelized version.

In a second stage of testing, Ancoa achieved further optimization of 15 to 25 percent using Intel® Threading Building Blocks (Intel® TBB), which enabled it to speed up memory allocation. This is available across Windows, Linux, and UNIX, the platforms Ancoa supports.

Results

The parallelization exercise, which used lock-free queues and Intel TBB, managed to deliver excellent scalability for message consolidation. Before analyzing the results, it is worth clarifying that all key components in the message consolidation were discretized and split into different steps.

This explains the performance gain from one core (pre-opt) to one core (post-opt), since some of the consolidation tasks were moved to a different thread, which effectively enabled better management of the workload. The cores in Figure 3 are used for message transformation. One core (post-opt) uses three threads in total

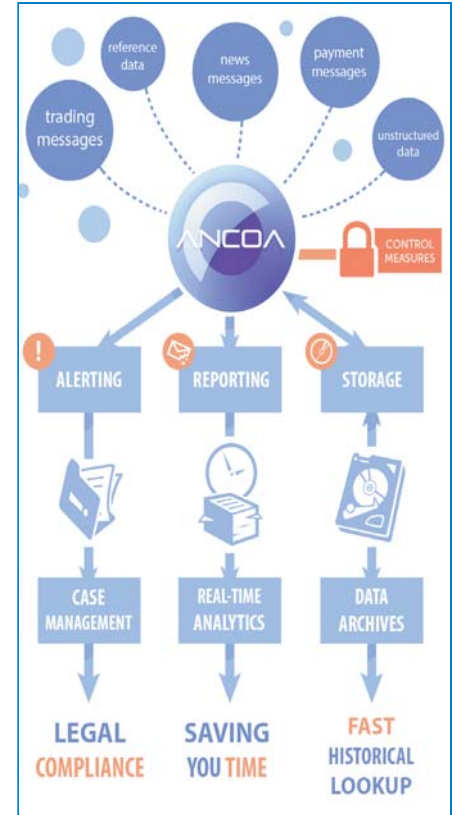


Figure 2. Ancoa Conceptual Diagram

(i.e., one core for message transformation itself and two additional help threads). This explains the performance gain on one core (pre-opt) versus post-opt.

The challenge of keeping the messages ordered with a parallelized pipeline was solved by having circular, lock-free queues feeding into parallel threads and serializing again on the other end using the same mechanism. After this parallelization exercise, memory allocation became the new bottleneck. This was solved by using Intel TBB’s fast memory allocation.

Multi-Core Optimization for Market Surveillance

Figure 4 shows the performance gains measured:

- **For SWIFT MT payment messages**, the performance gain was fivefold (from 2,525 TPS to 14,057 TPS), effectively using one thread before and four conversion threads plus two helper threads (a total of six threads) afterwards.
- **For position-based trading messages**, the performance gain was about fivefold (from 27,106 TPS to 134,987 TPS), effectively using one thread before and four conversion threads plus two helper threads (a total of six threads) afterwards.
- **For FIX trading messages**, the performance gain was approximately 3.5 times (from 40,373 TPS to 136,654 TPS), effectively using one thread before and four conversion threads plus two helper threads (a total of six threads) afterwards.

Each data point in the graph is based on the average of three separate samples, converting one million messages per message format on a single machine with two Intel Xeon processors E5-2680.

A single thread is used for message conversion and serialization, as well as for storage.

Table 1 shows the pre-optimization performance of consolidating messages on a single core, expressed in TPS. The performance was reasonable for a single core, but left the available hardware underutilized.

Separate threads are used for message conversion and serialization, and also for storage. The number of cores in Table 2 represents the number of threads used for conversion, which is the critical operation in the chain.

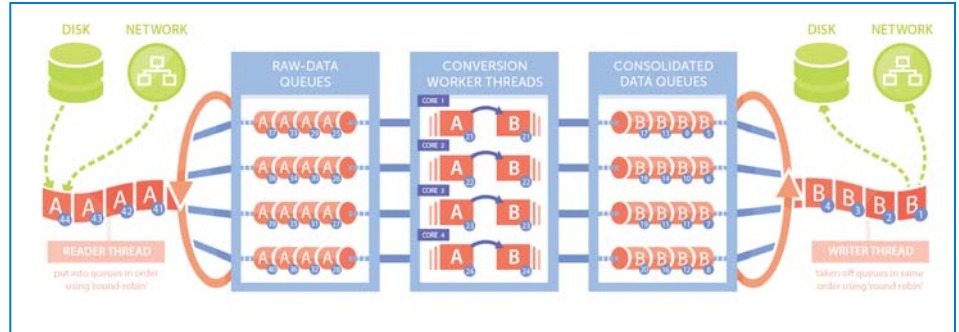


Figure 3. Consolidator Parallelization Mechanism

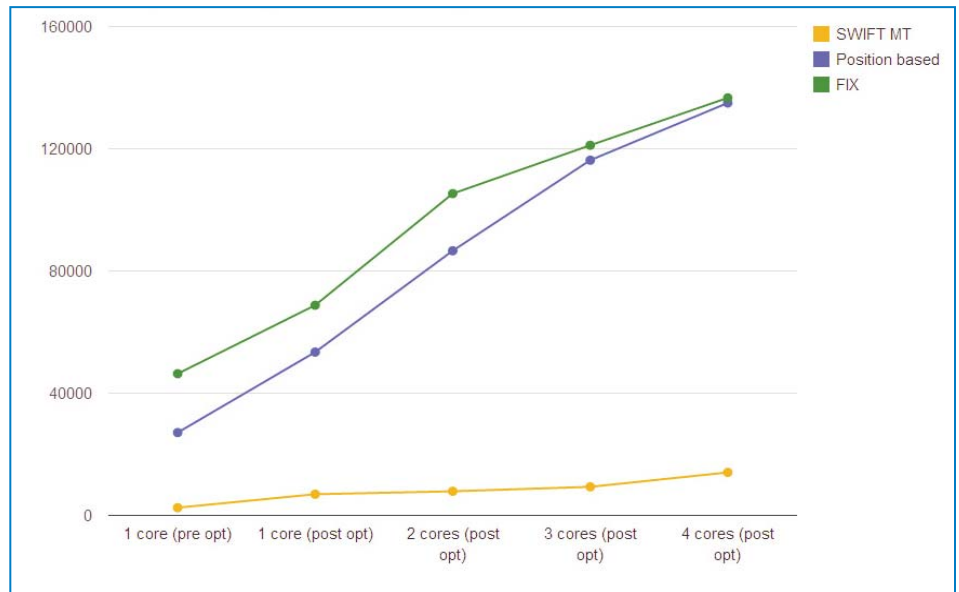


Figure 4. Performance Gains

Table 1. 1 Pre-Optimization Performance Numbers Using One Core

Message Format	TPS Rate
SWIFT MT	2,525
Position-Based	27,106
FIX	46,373

The test environment included:

- Microsoft Windows Server 2012 Standard 64-bit
- Two Intel Xeon processors E5 2680
- 32.0GB DDR3 @ 665MHz (9-9-9-24)

Conclusion

The performance gains from this exercise have been tremendous.

Intel’s tools have helped improve Ancoa’s performance, which will provide benefits across this platform for customers using Windows, Linux, and UNIX. It also shows that by optimizing hardware utilization at each node, Ancoa achieves maximum throughput from any deployment, either on a cloud infrastructure or an on-site deployment.

Achievements included:

- **Near-linear scalability** for Ancoa at the application level.

Table 2. Post-Optimization Performance Numbers, in Function of Number of Cores

Message Type	One Core	Two Cores	Three Cores	Four Cores
SWIFT MT	6,930	7,883	9,351	14,057
Position-Based	53,441	86,617	116,252	134,987
FIX	68,789	105,316	121,139	136,654

- **Ability to confidently handle** large message volumes with widely-used Intel server chipsets. This enables regulators, exchanges, brokers, and trading desks to match their needs for real-time handling of vast amounts of trading data, including both trading messages of different types and payment messages.
- **High infrastructure utilization**, ensuring a minimal hardware footprint.
- **Improved performance** by between 3.5 and five times on a single node in the dis-

tributed architecture, with performance for the widely used FIX industry standard of 136,654 transactions per second.

- **High throughput and scalability**, which let Ancoa put trading venues and financial organizations challenged by transactional volume growth in firm control of the supervision of their operations.

[Learn more about Ancoa here.](#)

[Learn more about the Intel® Xeon® processor E5 product family here.](#)

Copyright © 2013 Intel Corporation. All rights reserved.

Intel, Xeon, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to <http://www.intel.com/performance>.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

0813 GH/SS

