

Performance & Scalability Characterization

By Richard Tibbetts
Co-Founder and Chief Architect, StreamBase Systems, Inc.

Motivation for Performance and Scalability in a CEP Engine

CEP engines can be applied to a wide range of problems facing capital markets firms. In each of these problems, it is the CEP engine that performs the application's critical processing and therefore sits at the center of the flow of real-time and historical data streams. A low performing CEP engine can be the bottleneck of any application, unacceptably reducing the performance. Therefore, performance and scalability are important criteria when evaluating a CEP engine.

This report measures the performance and scalability of an Equities Alternative Trading System (ATS) application using the StreamBase CEP engine and illustrates why leading global investment banks, hedge funds and other capital markets firms find StreamBase is the best performing CEP engine for the job.

We measure throughput, latency, and scalability of the application. We conclude that on our baseline quad-core machine, the StreamBase ATS handles 140,000 input messages per second with latency of 86 microseconds. Scaling to an eight-core AMD machine, the StreamBase ATS processes 245,000 input messages per second, with 71 microseconds of latency, demonstrating a scaling factor of 0.875 across multi-core architectures.

Choosing an Application

In order to develop a representative measurement of the performance and scalability of a CEP engine, we must first identify the key attributes of the measurement:

- **Full Application:** CEP engines are used to build whole applications, and it is better to measure the performance and scalability of an entire application rather than focusing on the performance of individual pieces of that application.
- **Heterogeneous Messages:** The application being measured must process different types of messages to be fully representative of a common trading application.
- **State Management:** The application being measured should require the storage and maintenance of state, not just simple data transformation, in order to exercise the capabilities of the CEP engine in this area.

Metrics for Performance

Performance of CEP engines is multi-dimensional. In order to fully characterize performance, three things should be measured:

- **Throughput:** How many messages per second can the application process? This defines the raw speed of the application, how much data it can process in a steady state, as well as how it will deal with load spikes.

- **Latency:** What is the time between the engine receiving a message, and the results of that message being sent out? Latency is critical to trading operations, because a late answer is often a wrong answer, as market conditions change.
- **Scalability:** How do throughput and latency change when the application is moved to different hardware, such as a machine with more processing cores. Data volumes are always increasing, and it is important that a CEP application be scalable so that it does not have to be rewritten in the future.

Equities ATS Benchmark

The application measured in this report is based on an Equities Alternative Trading System (ATS). The ATS is responsible for holding and crossing orders inside a broker/dealer. This application might run inside any major broker, handling both internal and client order flow. The ATS handles heterogeneous messages with both market data and multiple kinds of orders. It also maintains state, in the form of a book of pending orders to be matched against.

The message inputs to an Equities ATS include:

- **Market Data:** In order to perform legal crosses, market data must be monitored and NBBO maintained
- **Orders:** Client order flow must pass through the ATS, in order to create liquidity. Orders are either matched against existing orders in ATS, or passed to downstream markets for execution.
- **Indications of Interest (IOI):** When an internal desk is looking for liquidity, they can place an indication of interest into the ATS, which will rest there waiting for a cross, rather than being sent to the downstream market

The message outputs from an Equities ATS include:

- **Order and IOI Echo:** Orders and IOIs are also sent to downstream systems for record keeping
- **Order Ack to Client:** Upstream clients must receive acknowledgements when their order is resting in the ATS
- **Fill Messages:** When an order fills or partially fills, the ATS must also send notifications to the client

The StreamBase Application

Diagram A below shows the StreamBase Equities ATS application. Input streams on the left receive market data, orders, and indications of interest. These data are normalized before being delivered to the core of the application, where matching occurs. An order book is maintained in memory of pending orders. New orders are matched against the pending orders. If there is a match, a fill message is created, which notifies the client and triggers the booking of a trade.

Other parts of the application are responsible for calculating monitoring statistics, such as throughput, order resting time, and fill rates, and for logging actions to create an audit trail.

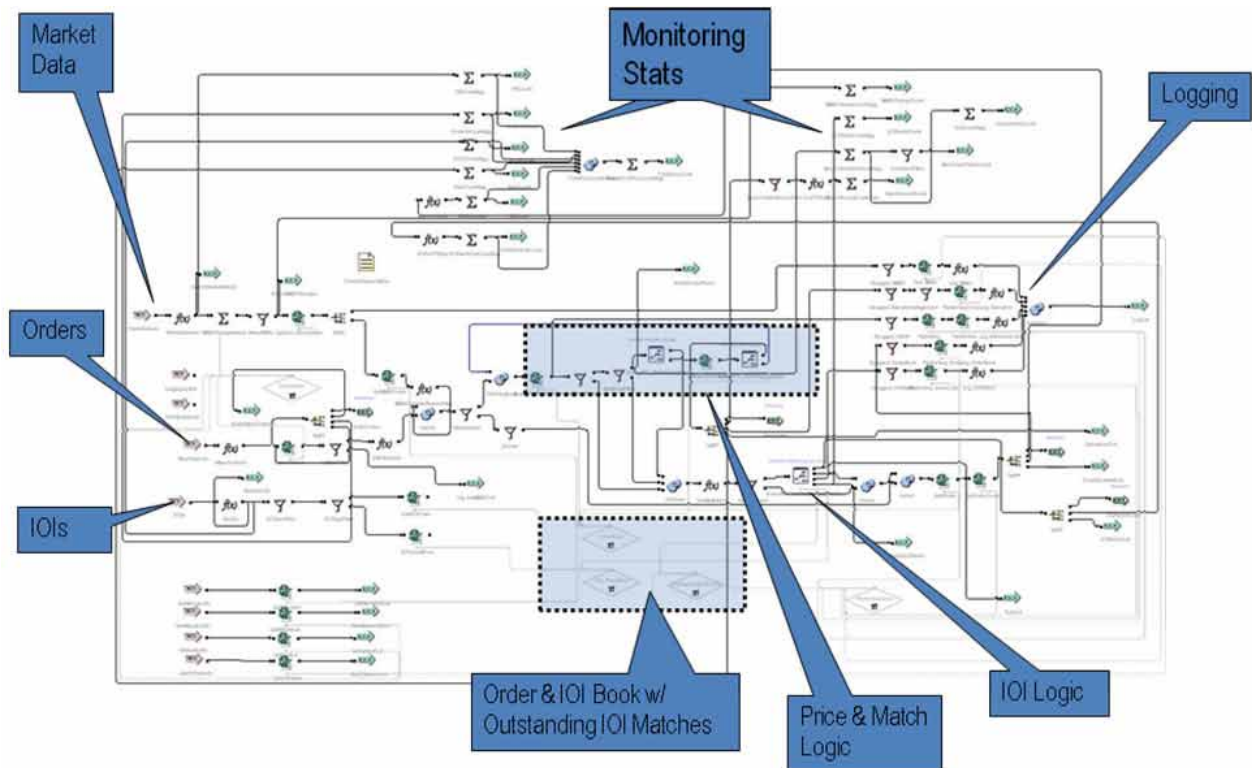


Diagram A: StreamBase Equities ATS Application

Hardware Specifications

We use a 4-core Dell PowerEdge 1360 server and an 8-core AMD whitebox server to measure the performance and scalability of the StreamBase Equities ATS application.

Chart A shows the hardware specifications of these multi-core machines.

	4 core	8 core
Model	Dell PowerEdge 1360	AMD Whitebox
CPU	2 Xeon 5110	2 Opteron 2358
RAM	8GB	8GB
OS	RHEL AS4	RHEL AS4

Chart A: Hardware Specifications

Performance and Scalability Results

Overall Results

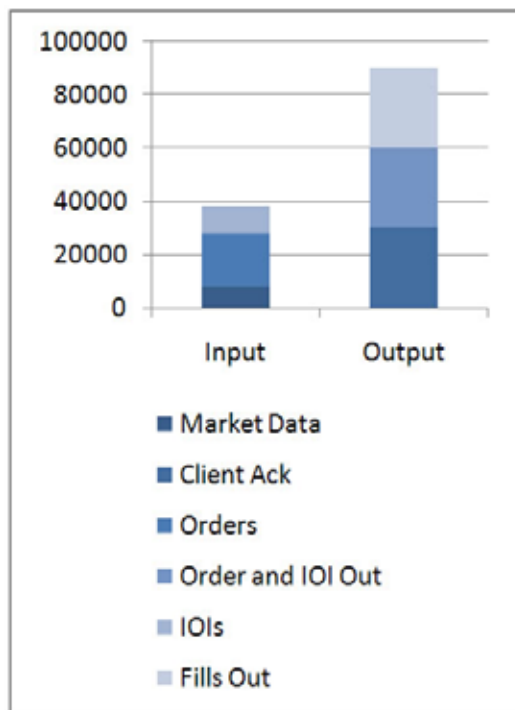
We measured throughput, latency, and scalability across the two hardware configurations. On our baseline quad-core machine, the StreamBase ATS handles 140,000 input messages per second with latency of 86 microseconds. Scaling to an eight-core AMD machine, the StreamBase ATS processes 245,000 input messages per second, with 71 microseconds of latency, demonstrating a scaling factor of 0.875 across multi-core architectures.

Throughput Results

Executing on the standard 4-core machine, the StreamBase CEP engine was able to maintain in excess of 140,000 messages per second. This input was composed of 118,600 market data updates per second and 21,900 orders per second.

On the 8-core machine, throughput was 245,400 input messages per second. The breakdown was 207,800 market data updates per second and 37,600 orders per second.

Graph A shows the input and output results of the Equities ATS.



Graph A: Equities ATS Input/Output Results

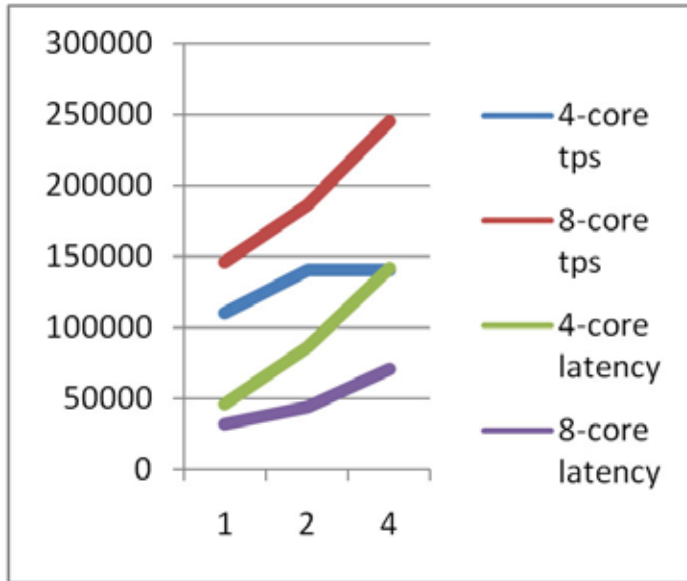
Latency Results

On the 4-core machine, average latency was 86 microseconds. On the 8-core machine, average latency was 71 microseconds.

Scalability Results

In cooperation with AMD, we demonstrate the scaling of the StreamBase Equities ATS application when moving from a 4-core server to an 8-core server. We take advantage of the parallelism in the application to run portions of the load in parallel.

Graph B below shows that the AMD 8-core machine supports higher throughput and greater parallelism with lower latency.



Graph B: Scaling of the Equities ATS Benchmark

About Richard Tibbetts

Richard Tibbetts is co-founder and Chief Architect at StreamBase Systems. Throughout his career, Tibbetts has concentrated on the design and development of database technologies, while also offering specialization in program analysis and distributed computing.

Tibbetts created the Linear Road Benchmark, the database management industry's guide for comparing the performance of real-time streaming databases with alternative systems, like relational databases. In addition, he has been heavily involved with a number of academic database projects, including Aurora at Brown University, and Medusa at the Massachusetts Institute of Technology (MIT). In 2003, the research from these two projects later led to the founding and commercialization of StreamBase's Stream Processing Engine.

As a co-founder and Chief Architect at StreamBase, Tibbetts is directly involved in architecting the foundation for the next-generation textual language based on Structured Query Language (SQL). The objective of this work is to apply the benefits of SQL for stored data to real-time, transitory data. Tibbetts is also charged with global responsibilities in furthering new StreamBase capabilities through platform and technical innovation.

Tibbetts earned both his bachelor of science in computer science and engineering and his master's of engineering degree at MIT. He completed his graduate research under the direction of Dr. Mike Stonebraker.

About StreamBase

StreamBase Systems, Inc, a leader in high-performance Complex Event Processing (CEP), provides software for rapidly building systems that analyze and act on real-time streaming data for instantaneous decision-making. StreamBase's Event Processing Platform(tm) combines a rapid application development environment, an ultra low-latency high-throughput event server, and the broadest connectivity to real-time and historical data and leading EMS/OMS software platforms. Six of the top ten Wall Street investment banks and three of the top five hedge funds use StreamBase to power mission-critical applications to increase revenue, lower costs, and reduce risk. It is also used by government agencies for highly specialized intelligence work. The company is headquartered in Lexington, Massachusetts with European offices in London. For more information, visit www.streambase.com.

Corporate Headquarters
181 Spring Street
Lexington, MA 02421
+1 (866) 787-6227

New York Office
845 Third Avenue, 6th Floor
New York, NY 10022
+1 (866) 787-6227

Virginia Office
11921 Freedom Drive, Suite 550
Reston, VA 20190
+1 (443) 472-0767

European Headquarters
60 Cannon Street
London, EC4N 6NP
+44 (0) 20 7002 1095

www.streambase.com