SYBASE®

WHITE PAPER

Analyze and Act on Fast Moving Data: An Introduction to Complex Event Processing

Learn the underlying concepts and benefits that can be gained by using complex event processing technology to address the high performance needs of today's real-time enterprise.

INTRODUCTION

In the capital markets, things happen fast. The markets move fast, positions change fast. Reacting quickly is the key to increasing profit and/or managing risk. Whether it's an automated trading application or market making application that needs to react to market movements instantly, or a risk manager that wants to see the firm's exposure updated continuously throughout the day, it all comes down to the ability to analyze data arriving from multiple sources, at very high rates, in real-time.

This phenomenon is by no means exclusive to the capital markets. In addition to financial services, many industries, including telecommunications and networking, logistics and transportation, and government, are experiencing the same challenges of growth in data volumes, acceleration of rates at which data is received or created, along with increasing business pressure to be able to analyze the data and act on it in real-time to maximize profit and/or to reduce risk.

Complex event processing technology (CEP) provides an innovative approach to deriving intelligence from event data in real-time. As a platform for application development, it provides high level tools for defining how events will be processed and analyzed. As an engine for an Event Driven Architecture (EDA) it provides the "brains" to absorb, aggregate, correlate and analyze events, producing new high-level events that can trigger a response as well as producing high-level information that shows the current state of the business. It lets you easily define logic that will be applied to incoming events (i.e. messages) to do things such as:

- Combine data from multiple sources, producing derived streams with richer and more complete information.
- Compute value-added information to enable rapid decision-making.
- Watch for specific conditions or patterns to enable instantaneous response.
- Produce high-level information, such as summary data, statistics, and trends to be able to see the big picture, or the net effect, of many individual events.
- Continuously re-compute key operating values based on complex analysis of incoming data.
- Collect raw and/or result data into a historical database for historical analysis and/or compliance.

This paper is intended to introduce you to the concepts underlying complex event processing and the benefits that can be gained by building event processing applications using the Sybase Aleri Streaming Platform, the most complete enterprise-level complex event processing technology available for today's demanding requirements.

WHAT IS COMPLEX EVENT PROCESSING?

Consider the following situations...

- An automated trading application that scans massive amounts of incoming market data to spot trading opportunities, where the trigger to trade has to be instantaneous or the opportunity is missed.
- A market making application that has to adjust internal or published rates in response to market movements—delays either mean lost business or lost profit.
- A risk management application that continuously updates aggregate position and risk information, combining data from multiple systems to provide a single consolidated view that is always current.

These are just a few examples of the types of applications that can benefit from complex event processing technology. The common denominator among these applications is that they share the need to continuously collect, process, and analyze data in real-time, producing results without delay, even when the data arrives at very high rates.

While traditional databases were designed to process individual transactions at very high rates; analyzing the data to look for specific conditions or deriving higher level summary data were tasks that had to happen "off-line," using query tools that were never designed to produce actionable intelligence in real-time. This made them unsuitable for applications that had to analyze data in real-time (such as trading applications), and while they were adopted for data analysis in other areas, the fact that the analysis is done on "historic" data means that the business insight is derived after-the-fact, in many cases missing an opportunity to react quickly to the results of the analysis.

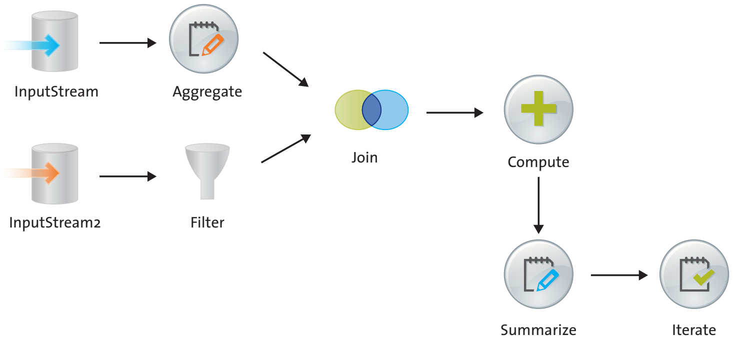
Complex event processing technology delivers the data analysis tools traditionally provided by relational databases or even spread sheets, but in a real-time event-driven implementation that is capable of processing incoming data at very high rates and producing results with near-zero latency.

Think of it as taking some of the fundamental concepts of a relational database and turning them upside down: a traditional relational database is designed to collect data and store it, where you can then analyze it to filter the data, combine it, group it, search for patterns, derive high level summary data, etc. The analysis happens off-line, not in response to incoming events. An event processor (the heart of CEP technology), in contrast, takes incoming messages and runs them through a set of pre-defined continuous queries to produce derived streams or sets of data.

We call them continuous queries because the data analysis logic is similar to what might be included in a traditional database query. For example:

- Show me which events meet this criteria
- Tell me if this pattern of events (or non-events) occurs
- Show me the current total of all events matching this criteria
- Group events by these values and calculate the average for each group

While complex event processing logic may be similar to a traditional database query, the implementation is anything but. Event processing uses a dataflow architecture to pass incoming messages through the continuous query operators as soon as the message arrives so that the result sets are instantly updated. These functions used within the continuous queries have been implemented in a way to maximize throughput and minimize latency.



An Alternative to Custom Code

Complex event processing technology provides an alternative approach to building high performance enterprise-class applications that have to process event data in real-time. Custom applications written in C++ or Java are expensive, time consuming to build, and are typically inflexible and therefore expensive to maintain since the processing logic is hard coded and tightly bound to the data structures. What's more, designing and writing highly efficient code for real-time processing requires specialized programming skills.

A New Concept, New Terms

What often happens with the emergence of a new technology, is that it takes awhile for the industry to adopt common terminology and agree on common definitions for the various terms being used. This is certainly the case with complex event processing (CEP). Over the past two years as this technology has emerged, it has been referred to as "Event Stream Processing", "Stream Processing", or simply "Event Processing". While "event processing" is still an appropriate term and is technically the most general term, encompassing all types of event processing, the phrase "complex event processing" has become the label most industry participants are using to refer to this class of technology.

In his book "The Power of Events", David Luckham of Stanford defines complex event processing as "a set of techniques and tools to help us understand and control event-driven information systems". This definition applies just as easily to "event stream processing". We could expand the definition, however, to look beyond "information systems" and apply it to event-driven business processes.

A Spectrum of Uses and Requirements

Regardless of the specific terms used, all event processing applications set out to do one or more of the following:

- **Situation Detection:** Monitor incoming events to detect patterns that indicate the existence of an opportunity or a problem—i.e. a situation that warrants a response or that needs to be recorded. This can range from a simple filter to a complex set of rules that correlate incoming events and screen for sets of conditions that may include missing events. High level events indicating the existence of the situation are generated as the result.
- **Data Aggregation and Analysis - Continuous Computation:** Data is correlated, grouped and aggregated, and computations are then applied to produce new information such as summary data, high level statistics, or adjustments to key operating parameters. Examples of this type of CEP include:
  – continuously adjusting prices based on movements in the market or other real time inputs
  – continuously updated key performance indicators (KPIs)
  – continuously update valuations, exposures
  – continuous aggregation of data from multiple sources to see “the big picture”
- **Data Collection:** A by-product of CEP is often the collection of raw event data and/or higher level summary data. The collected data can be used as context for processing newly arriving events and can also be stored in an historical database for off-line analysis, reporting or to have an audit trail.
- **Application Integration, Intelligent Event Handling:** Many applications are built on an Event Driven Architecture, but the basic tools for EDA provide the mechanisms for the exchange of event data without providing the ability to analyze event data. CEP can provide intelligence within an event driven architecture to analyze events in the context of other events and a knowledge of the state of various systems to determine what new events need to be generated or to determine the action to be taken based on an event.

This is an important point, since in a particular context you may find the focus to be on one particular aspect of event processing. Yet recognizing that different applications have different needs will help you ensure that you select the optimal tool or tools for the job.

Real-Time Data Analysis

When we talk about the ability to analyze incoming event data in real-time, we are actually referring to a variety of functions that can be applied to the data, alone or in combination, to derive high level intelligence and/or to trigger a response. Examples include:

- Filter data to apply simple or complex filters to detect conditions of interest. This can include correlation of events across multiple sources, correlation of events across time, and watching for sets of events that match a defined pattern.
- Combine data from multiple sources, including the ability to combine streaming and static data or to combine data that arrives at different times. Define data retention “windows”, either based on time or number of elements, across which the computations will be performed.
- Group and aggregate data, producing high level summary data and statistics. This can include trends (moving averages), net positions/exposures, etc.
- Compute new data elements: enrich simple event data by adding new fields that are computed based on context, data from other sources, etc.
- Transform data format and structure. This can go beyond simple message-level transformation and can create entirely new events based on individual or multiple events using rules that take into account context, reference data, etc.
- Generate high level events from patterns or groupings of low level events.

THE SYBASE ALERI STREAMING PLATFORM

The Sybase Aleri Streaming Platform is a high performance enterprise-class complex event processing engine that can be used to quickly implement and deploy a wide range of applications that need to analyze and act on event data in real-time. This product, represents the state-of-the-art in complex event processing, combining performance, versatility, and ease-of-use in an enterprise-class implementation designed for use in the most demanding environments.

Designed for Performance

The fundamental design objectives of the Sybase Aleri Streaming Platform were, from the beginning, defined to be maximum throughput with minimal latency. On a 2 CPU Linux server, for example, the Sybase Aleri Streaming Platform can process well over 100,000 messages per second, and in some cases over a million messages per second depending on the data and the processing logic being applied. Latency, measured from the time a message arrives until processing is complete and results have been produced, is typically in the range of a fraction of a millisecond to a few milliseconds.

Designed to be highly scalable and to function as an infrastructure component in mission-critical applications, the Sybase Aleri Streaming Platform is a multi-threaded 64-bit application supported on Linux, Solaris, and Microsoft Windows (32 bit). The application is fully multi-threaded to take advantage of the parallel processing capabilities of multiple CPU machines and also has clustering features allowing applications to scale across multiple servers. The net effect is virtually unlimited scalability, as additional CPUs and servers can be added as needed to increase throughput.

Designed for Versatility

As described above, different applications have different needs. Many CEP products are designed to address a single type of application. For example, there are a number of CEP "rules engines" that are designed expressly for situation detection. That's fine if all you need is situation detection, but the technology may not be extensible to other types of applications. The Sybase Aleri Streaming Platform was designed to address the widest possible range of event processing requirements:

- Monitor incoming data streams for conditions that represent opportunities or threats
- Augment data streams with data from other sources and/or computed values
- Group data by different dimensions, producing high level summary data or statistics
- Consolidate data from multiple heterogeneous systems, forming a single aggregate view or stream
- Operate on large data sets spanning large time windows
- Collect raw and/or result data for use in historical analysis, reporting, or to provide an audit trail

Some of the specific aspects of the Sybase architecture that give it this versatility include:

- Incoming messages can be processed as inserts, updates, deletes or upserts. This lets Sybase address situations were incoming messages don't just represent a new data point in a time series, but represent an update to previous information. Many CEP implementations don't handle updates and deletes—they treat all incoming messages as new data points in a time series. The reality is that many data streams produce updates, changes, and cancellations. Whether it's changes to an order book, or a correction to data previously sent, these updates need to be applied to previously received data to maintain an accurate view of the current state.
- Tunable state management with options for stateless, in memory and disk-based retention that can be applied at the individual stream level. This allows full data retention on streams that need it, retaining individual messages according to defined retention rules. For streams where it's essential that data is never lost, with the ability for rapid full-state recovery after a failure, the stream can be configured for disk-based retention. Sybase has developed a proprietary high performance data persistence model to ensure that logging data to disk has minimal impact on throughput and latency.
- Support for dynamic changes to the data model, allowing changes to the computations and "rules" to be applied in flight without affecting the processing of incoming data or streams that are not being changed.

- An on-demand query interface that allows all retained data sets to be queried by external applications as if they were held in a database. The ODBC/JDBC interfaces provide support for off-the-shelf query applications and allow for snapshots of current data sets.
- Built in security including access control, authentication and encryption.

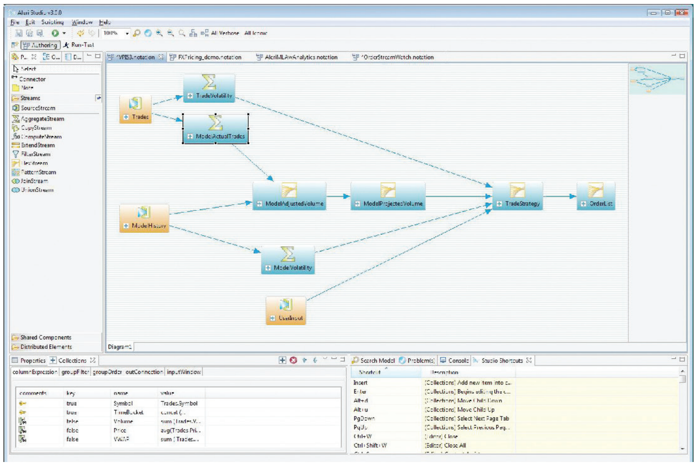
High Level Authoring Tools for Application Development

With the Sybase Aleri Streaming Platform, the data model (i.e. the event processing logic) is defined using any one of three high level authoring environments:

The Sybase Studio: An integrated development environment with a visual editor as well as an execution and test facility. This is based on the widely used Eclipse® framework and provides both authoring and execution perspectives. This provides a visual paradigm for building a model and defining the data flow. The execution perspectives provide a range of tools for testing a model including: record/playback, data simulator, debugger, performance monitor, and streamviewer.

Sybase SQL: A version of standard ANSI SQL, with a few minor extensions for real-time data handling, this provides a familiar environment for anyone used to working with relational databases. This is a complete textual language allowing the author to work in the text editor of choice. Alternatively the Sybase Studio includes an Sybase SQL editor.

XML: Both the Sybase Studio and Sybase SQL produce a data model specification in XML which gets read by the event processor. The XML elements can be created or edited directly using an XML editor, without the need to use either the Studio or SQL. The Sybase XSD defines the elements and attributes that are used to define the data model. The XML data model provides an ideal base for implementation of custom GUIs that can be used to create or modify a model or even for the implementation of specialized languages.



comments	key	name	value
	true	Symbol	Trades.Symbol
	true	TimeBucket	concat (...
	false	Volume	sum (Trades.V...
	false	Price	avg(Trades.Pri...
	false	VWAP	sum (Trades...

Shortcut	Description
Insert	[Collections] Add new item into c...
Enter	[Collections] Begins editing the c...
Alt+d	[Collections] Move Child Down
Alt+u	[Collections] Move Child Up
PgDown	[Collections] Select Next Page Tab
PgUp	[Collections] Select Previous Pag...
Ctrl+W	[Editor] Close
Ctrl+Shift+W	[Editor] Close All

Sybase Aleri Streaming Platform Product Features

Corrections and Updates: While most event processors treat all incoming data as a time series, appending the latest message to the history of the stream, the Sybase processor uses a more sophisticated model that can also apply incoming messages to the stream history as updates or deletes occur.

Data Capture: Raw and/or derived data can be captured, and transferred to an off-the-shelf historical database, providing a historical record for future analysis or audit purposes.

On-Demand Queries: In addition to producing streaming output, all of the raw and derived event data can be queried via standard ODBC and JDBC interfaces. To a reporting tool, it looks like a database where the views are always current.

Configurable Data Retention: Individual streams can be configured as stateless or for data retention. Data retention rules can be defined based on time or number of elements and can scale for retaining large data sets. Individual events and/or summary events can be retained.

Optional Data Persistence: While data is processed in memory for high performance, some or all of the retained data can be designated for disk-based persistence, ensuring that critical data is not lost in the case of a system failure. Upon recovery, the state of each persisted stream is completely restored. Sybase's proprietary high speed log store ensures minimal performance impact when disk-based persistence is used.

Dynamic Data Models: Data models can be changed in-flight, without interrupting the processing of incoming data and without affecting streams that are not being changed.

FlexStreams: Custom stream operators can be implemented using FlexStreams. These programmable stream operators use a simple scripting language, providing the ability to implement procedural logic within a relational data model.

Asynchronous and Synchronous Message Options: While most CEP implementations assume full asynchronous operation, Sybase recognizes that some applications need to ensure data integrity. Thus synchronous message transfer options are available for input streams with tunable parameters to balance the needs for data integrity and performance.

Built-in Security: Stream level access control, authentication and optional encryption of input and output streams. Cluster configurations: Support for clustered hardware configurations include the ability to run a single data model across multiple machines with a cluster manager that will manage the distributed process.

High Availability Options: Hot standby facilities provide for automatic failover from a primary to a live secondary. Alternatively a cold-spare configuration can be used to automatically start a new server on available hardware. Disk-based persistence, if turned on, allows for a new instance to be brought up with the full data set reinstatiated.

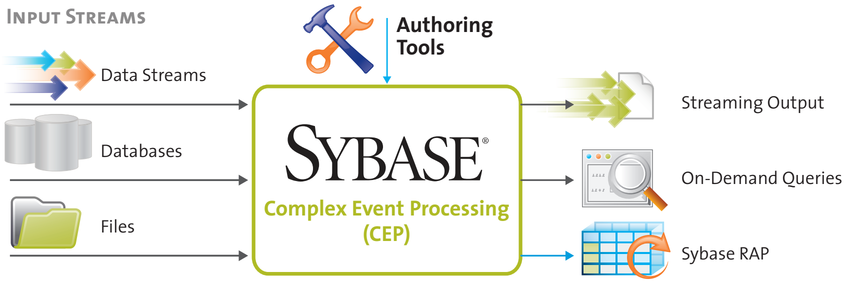
User Defined Functions: External function libraries can be referenced as part of the event processing, meaning that you are not limited to the built-in functions.

SybaseRT for Microsoft Excel®: This add-in for Excel, included in the base product, provides the ability to publish data from Excel into the Sybase platform, and to subscribe to streaming output from the Sybase Platform within Excel.

SYSTEM INTEGRATION

The Sybase Aleri Streaming Platform runs as a server process that receives and publishes messages via sockets. The data model is loaded at start-up and all incoming messages are processed according to the model. The model can be changed in-flight using the dynamic data model facility.

Data is streamed into the server using the Sybase pub/sub API. It is available in Java, C++ and .NET. The same interface is used to subscribe to output streams (results) from the Sybase Platform. The pub/sub API can be imbedded in applications that produce events that will be processed by the Sybase Platform, consume events produced by the Sybase platform, or both. The API can also be used to build adapters. A range of pre-built adapters is available from Sybase, including adapters for TIBCO, IBM MQ, JMS, Reuters, ODBC, JDBC and others. New adapters are being built all the time—check with Sybase for the availability of specific adapters to meet your needs. Custom adapter can also be built on request.



The User Defined Function (UDF) interface of the Sybase Platform allows functions contained within external function libraries to be invoked within an Sybase data model. The UDF API is currently available for C++ and Java.

The On-Demand Query interface allows snapshot SQL queries to be run against the retained data sets within the Sybase Platform. ODBC and JDBC interfaces are available as well as a C++ API.

Command and control of the Sybase server is via an XML RPC interface.

TYPICAL COMPLEX EVENT PROCESSING APPLICATIONS

The Sybase Aleri Streaming Platform enables rapid development of a wide range of event processing applications. Just a few of the applications that have been built using complex event processing include:

Market Data Enrichment: select or combine data from multiple sources; monitor latency and quality; compute value-added fields; produce custom data streams.

Automated or Algorithmic Trading: process high volume market data to discover and act on trading opportunities; execute large orders using a variety of trading algorithms that track the market; consolidate depth-of-book data from multiple markets, enabling analysis of full market depth and pressure. The Sybase platform can be deployed in conjunction with a tick database to low-latency results that incorporate historical tick data.

Market Making, Auto Pricing: cleanse and validate incoming market data and then apply skew and spread, incorporating trader input to update internal and/or published rates.

Pre-Trade Validation and Compliance: check incoming orders for errors and/or compliance without adding latency.

Best Execution, Smart Order Routing: route orders in compliance with Reg NMS and MiFID taking into account protected orders, customer profile and preferences and venue characteristics. Capture and record data for compliance monitoring and reporting.

Post-Trade Monitoring and Reporting: detect discrepancies, track performance, generate quality statistics, generate alerts when acceptable thresholds are exceeded.

Risk Aggregation, Real-time P&L: aggregate position and risk information, in real-time, from multiple independent trading systems, position keeping systems, and risk management systems, providing a real-time aggregate view that can be analyzed across multiple dimensions; non-intrusive consolidation of information in real-time across organizational "silos" without the need to change or replace the underlying systems. Apply real-time market-data for continuous valuation.

Data Orchestration: intelligently manage data dissemination within a service oriented architecture, defining complex rules that incorporate knowledge of context and state to keep distributed systems in synch.

CONCLUSION

Critical business processes and decisions increasingly rely on having the most up-to-date information possible and the means to instantly react and respond to changing conditions. This is particularly key in the ever-changing financial services arena where speed, data volume, and accuracy is paramount for maximizing profits, minimizing risk, and maintaining compliance with corporate and governmental regulatory requirements. The Sybase Aleri Streaming Platform, high performance complex event processing technology from Sybase, offers next generation technology for meeting these demanding data challenges now and into the future.

For more information, please visit our website at **www.sybase.com**.

Sybase, Inc.
Worldwide Headquarters
One Sybase Drive
Dublin, CA 94568-7902
U.S.A
1 800 8 sybase

www.sybase.com

Copyright © 2010 Sybase, Inc. All rights reserved. Unpublished rights reserved under U.S. copyright laws. Sybase and the Sybase logo are trademarks of Sybase, Inc. or its subsidiaries. All other trademarks are the property of their respective owners. ® indicates registration in the United States. Specifications are subject to change without notice.
03/10

SYBASE®