Building Distributed Applications

Complex Event Processing with Coral8

Applies to: Financial Services Architecture .NET Framework

Abstract

This paper describes how Coral8 complex event processing software can be integrated with a broad .NET/Windows platform stack to form the foundation for algorithmic trading and other real-time analytics solutions, including real-time dashboards, advanced data visualization, dynamic OLAP, event-driven business rules, and real-time business process management.

John Morrell
Coral8, Inc.

Stevan D. Vidich
Microsoft Corporation

June 2008

Table of Contents

Introduction

Complex event processing (CEP) solutions are particularly suited to high-volume, low-latency applications such as those in financial trading, real-time best execution metrics, real-time Web commerce traffic analysis, RFID and sensor network processing, communications network record processing, and more. CEP software provide the platform to quickly analyze the high-speed, high-volume event streams to uncover the opportunities and threads buried inside, and drive immediate actions.

CEP engines provide an easily programmable, reusable infrastructure platform to increase the speed and lower the cost of developing and deploying real-time event-driven applications, much in the same way relational databases made it easier to store, manage, and access data. In fact, some CEP engines have a great deal in common with relational databases, including common SQL language constructs, data indexing, and relations between different data sets. This commonality reduces the learning curve for adopting CEP platforms, allowing programmers to reuse existing skills.

As more businesses move closer to real-time operations and business processes, CEP will offer a familiar, yet powerful means for IT teams to create event-driven applications to monitor and act upon critical enterprise “data-in-motion”.

Coral8 Overview

Coral8 CEP engine is designed for high-volume, low-latency applications where data analysis must occur in seconds, milliseconds, or even microseconds, either to provide important information to key stakeholders in a timely fashion or drive instantaneous actions. In the securities industry, such applications can be found in the Front Office, where algorithmic trading requires extraordinary performance. Some scenarios call for processing of hundreds of thousands of messages per second. Coral8 can connect to multiple streaming data feeds from leading market data providers. It can perform complex pattern matching analyses on incoming streams and implement execution strategies including volume-weighted average price (VWAP), time-weighted average price (TWAP), etc. Moreover, Coral8 CEP is equally well suited to other event driven solutions, including real-time dashboards, advanced visualization with Windows Presentation Foundation, dynamic online analytical processing (OLAP), event-driven business rules, real-time business process monitoring (BPM), and predictive analytics. Figure 1 provides a high-level overview of Coral8 architecture.

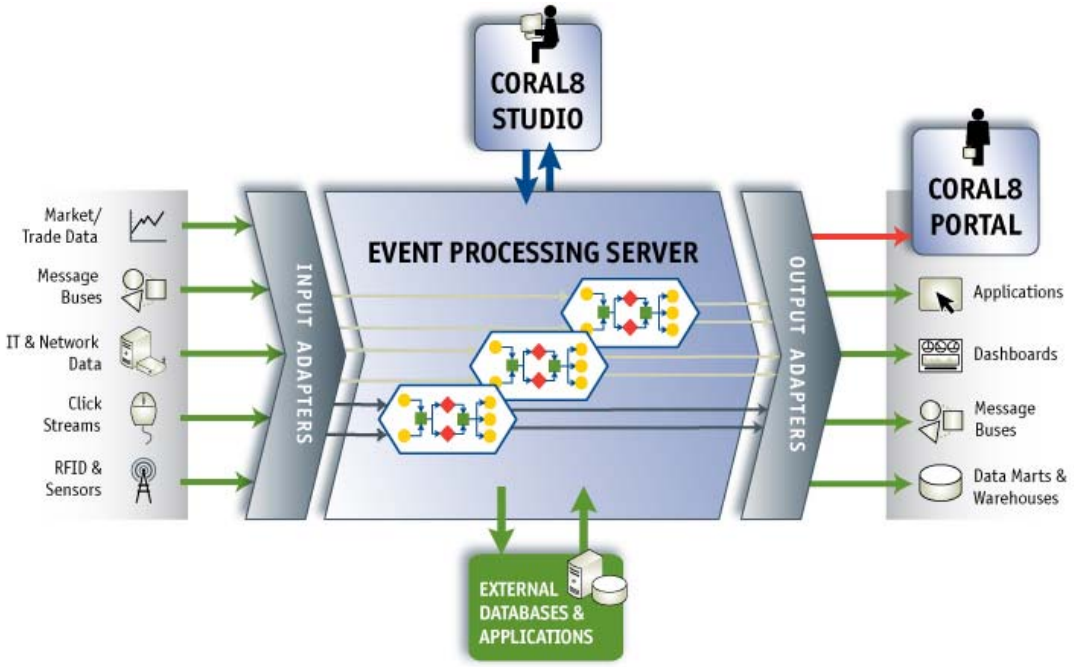

Figure 1. Coral8 Architecture

The Coral8 product suite consists of three main components:

- Coral8 Server - The Coral8 Server is the high-throughput, low-latency runtime server for Coral8 applications. It offers enterprise-class features to easily deploy, integrate, and manage Coral8 applications in the most scalable and mission-critical environments. It comes with a number of

packaged adapters for common high-speed data sources such as market data, messaging software, databases and more.

- Coral8 Studio - The Coral8 Studio is an interactive graphical environment for developing, testing, and deploying Coral8 components and modules. The look and feel of the Coral8 Studio is very similar to that of Microsoft Visual Studio. The Studio also acts as a central management console for a distributed network of Coral8 Servers.

- Coral8 Portal - The Coral8 Portal is a dashboard and visualization server that allows users to dynamically query and work with real-time CEP output in a secure, manageable environment. It offers an easy-to-use, self-service environment that quickly and efficiently puts powerful, real-time information from CEP applications in the hands of business users.

The Coral8 SDK provides a fully featured interface to develop custom adapters, utilities or programs that interact with the Coral8 platform. The entire set of Coral8 functionality – input/output adapters, control, and administration – is exposed through the Coral8 SDK APIs.

Coral8 – Microsoft Technology Touch Points

Coral8 provides a .NET API for integrating CEP with custom line of business applications developed using the Microsoft .NET Framework 3.0. It integrates with leading messaging backbones from TIBCO and IBM, which in turn enables .NET applications to receive messages via TIBCO's and IBM's native support for Microsoft .NET. By way of seamless integration with Windows Communication Framework (WCF), Coral8 CEP engine can leverage Microsoft's investments in a unified programming model for building service-oriented applications. The most important aspects of WCF are

- Unification of existing Microsoft distributed computing technologies, including ASP.NET Web Services (ASMX), Web Services Enhancements (WSE), .NET Remoting, Enterprise Services, and Microsoft Message Queuing (MSMQ)

- Support for advanced Web services via standard WS-* protocols that enable cross-vendor interoperability for secure, reliable, transacted communications

- Explicit support for service oriented development

With respect to graphical user interface (GUI) development, Windows Presentation Foundation (WPF) in .NET Framework 3.0 is a natural way to handle visualization of management and real-time information. WPF enables applications with rich, interactive displays, large datasets, and it takes advantage of hardware-accelerated, three-dimensional graphics to enable new modes of displaying the vast quantities of flow produced by algorithmic trading. Using a host of services provided by WPF "out-of-the-box", such as data binding, layout, animation, imaging, and effects, developers can easily provide robust, highly interactive and innovative user interfaces to control and monitor algorithmic trading systems. Moving beyond the simple paradigms of tickets and blotters, WPF-enabled applications can add value to the trader by displaying enhanced context around the important dynamics of multiple

ongoing strategies. At the same time, using techniques such as filtering, zooming, and 3D graphics, WPF can reduce or eliminate the need for traders to focus on the minutiae of their execution flow (cutting down on distraction and maximizing screen real estate) while still providing tools to dig deeper and drill down to the underlying data should they need to.

The Coral8 Portal, the self-service query, dashboard, and visualization server, can be easily integrated into Windows SharePoint Services. All query visualization components created in the Coral8 Portal can be accessed via a standard URL reference. SharePoint Web parts can directly call and display these query visualization components by referencing the appropriate URL.

For storage of events and real-time analytical results, Coral8 provides built-in support for SQL Server 2005 database. Once the data is stored in the database, it becomes very easy to use built in business intelligence (BI) features of SQL Server such as Analysis Services and Reporting Services. Moreover, customers can build very effective portals using Windows SharePoint Services in Windows Server 2008.

Real-time data processed by Coral8 can be output to Excel 2007 via Excel real-time data (RTD) components, and displayed using the familiar spreadsheet look and feel. RTD servers are COM DLLs that implement a specific interface provided by Excel. Displaying real-time data works via push-pull mechanism, whereby RTD server notifies Excel of data changes and Excel in turn requests changed data from RTD server. Coral8 provides an Excel adapter that implements this functionality. Table 1 and Figure 2 provide an overview of technology touch points between Coral8 and Microsoft's broad platform stack.

Table 1. Coral8 and Microsoft product mapping

Application	Coral8 Functions	Microsoft Products
Algorithmic Trading	• Market data processing • Algorithm execution • Trade signals	• SQL Server • WCF • Platforms
Trader Desk Support	• Position monitoring • Risk models • Pricing models	• SQL Server • SharePoint Server • WCF, WPF
Trader Monitoring	• Behavior Patterns • Real-time analytic capture	• SQL Server • SharePoint Server • WCF, WPF
Best Execution	• Real-time analytic capture • Warnings/Alerts • Client report data	• SQL Server • Business Intelligence • SharePoint Server • WCF, WPF

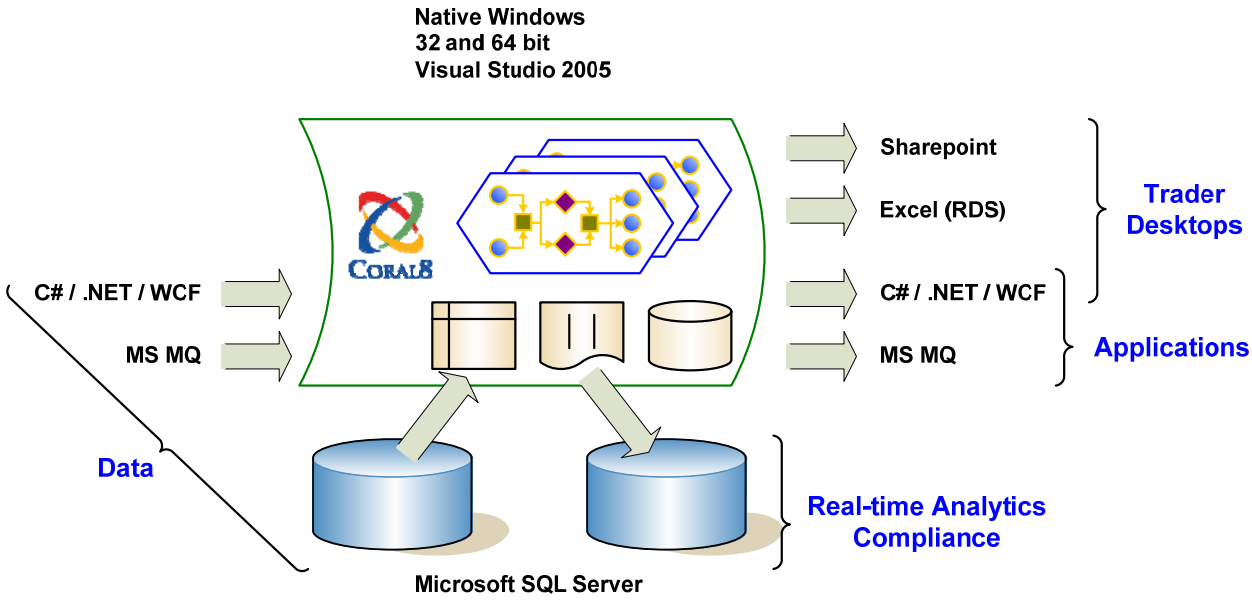

Figure 2. Coral8 and Microsoft product integration in financial services

Complex Event Processing with Coral8

In the highly competitive investment services sector, differentiation can often be measured in milliseconds. The firms that can process and analyze the ever growing volumes of market and trade data the fastest can execute trades more swiftly and pass this advantage on to their customers. CEP software enables the competitive edge investment services firms seek. The Coral8 CEP Engine enables high-throughput, low-latency processing and analysis of investment data for applications such as:

- Real-time market data enrichment for faster, more effective trading decisions
- Low-latency algorithm calculations to drive trading signals to algorithmic trading, order management, and execution management systems
- Inline best execution analytics to drive the lowest cost and fastest trade execution
- Real-time capture of trade execution data for simpler, lower cost reporting on MiFID and RegNMS requirements

The Coral8 CEP Engine replaces the lengthy and costly process of custom coding real-time market and trading applications. This dramatically lowers the cost of building these applications, speeds time-to-

market, and enables IT teams to support the needs of business units through faster delivery of trading applications.

The Coral8 CEP Engine focuses on several critical areas to enable large-scale, high-speed trading applications.

Performance

The native, high performance Coral8 architecture provides high-throughput and low latency for the complex workloads driven by sophisticated trading applications wanting true tick-by-tick analysis. The Coral8 Engine was built in C/C++ and runs natively on Windows 32- and 64-bit operating systems, ensuring maximum performance.

The Coral8 Server does not merely generate static runtime executables; instead, it manages run-time execution via its highly tuned optimizer and scheduler that considers many static and dynamic conditions for optimal throughput and performance. This allows the Coral8 Engine to optimize performance in real-world financial trading environments where market and trade data rates can be "bursty" and volatile, and multiple query modules are running, each with varying degrees of complexity.

Powerful, familiar language and tools

Coral8's development language, CCL, is based on SQL, providing a familiar and easy-to-use programming environment. The Coral8 Studio and Server combine to allow developers to create, add, change and adjust trading application modules "on demand".

In 2005, CCL emerged as the first declarative SQL-based CEP language, and has evolved to offer very powerful CEP language options, including sophisticated window policies (ways to manage CEP data sets), fine-grained event pattern matching, unique native XML processing functions, and inline database sub-queries to easily integrate real-time streams with SQL Server data.

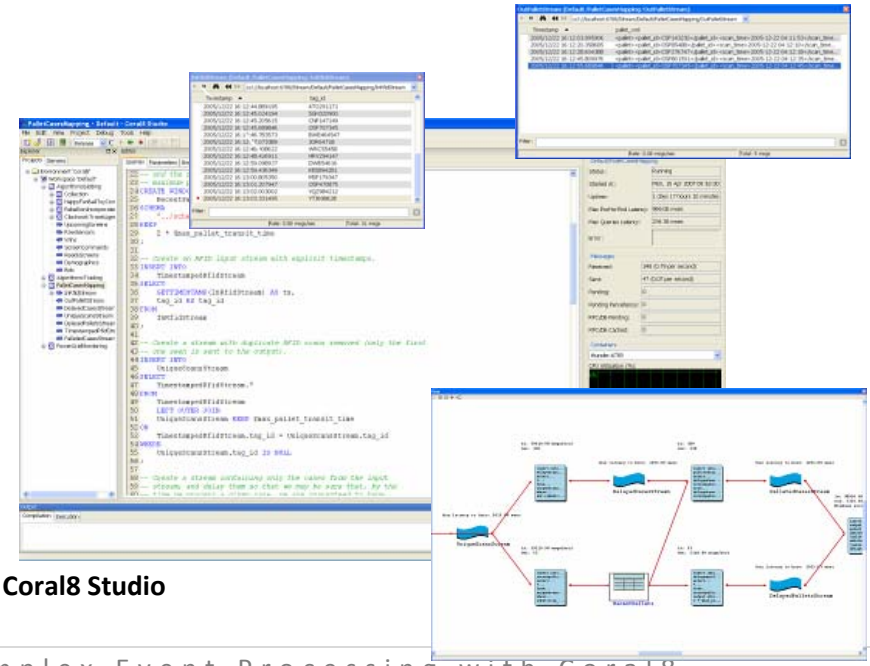

Figure 3: Coral8 Studio

The Coral8 Studio, shown in Figure 3, provides an easy to use environment for developers to build, debug, and manage CEP application modules and projects. Microsoft developers should find Coral8 Studio very familiar because it offers a similar look and style as Visual Studio. It contains a number of advanced features to debug and tune complex CEP applications, including stream viewers, step debugging, speed-up and slow-down, and granular performance statistics.

The Studio and Server also combine for a powerful backtesting environment, a process familiar to financial modelers and quants. The special Read-From-Database and Read-From-CSV Coral8 Adapters can be configured to read historical tick or trade data and treat this data as live streams for backtesting. Accelerated playback can be used to speed the backtesting process by "playing" the data into the system at an accelerated rate, enabling years of historical data to be tested in minutes. Coral8 guarantees deterministic results between backtesting and live environments when using one or multiple input streams and when testing using accelerated playback.

Clustering and high availability

The Coral8 Server supports multiple forms of clustering to maximize availability and to scale for extra performance:

- High Availability (HA) clustering maximizes CEP application uptime in mission-critical environments. IT can configure HA clusters in either active-passive or hot-standby configurations.
- Parallel clustering supports the processing of high-volume data streams in parallel on multiple servers. Event streams are dispersed across multiple instances of a Coral8 module running on different Coral8 servers. The Coral8 Studio makes it easy to deploy a query module across multiple servers and Coral8 Cluster Manager optimizes splitting data to the servers.
- Pipelined clustering can split computations into several pipelined stages. One or more stages can be dispersed across different Coral8 servers. The Coral8 Studio easily connects the pipeline stages together, and allows developers to visually monitor the performance of various stages and optimize use of resources.

A complex application often involves parallel, pipelined, and HA clusters together, as shown in Figure 4. This approach requires a single cluster manager that manages both HA and scalability clusters in an "M+N" model. The Coral8 Cluster Manager allows IT to mix and match various types of clustering for optimal use of computing resources. For example, four active parallel cluster nodes can share one failover/passive HA node, requiring only five systems for a full parallel and HA cluster implementation.

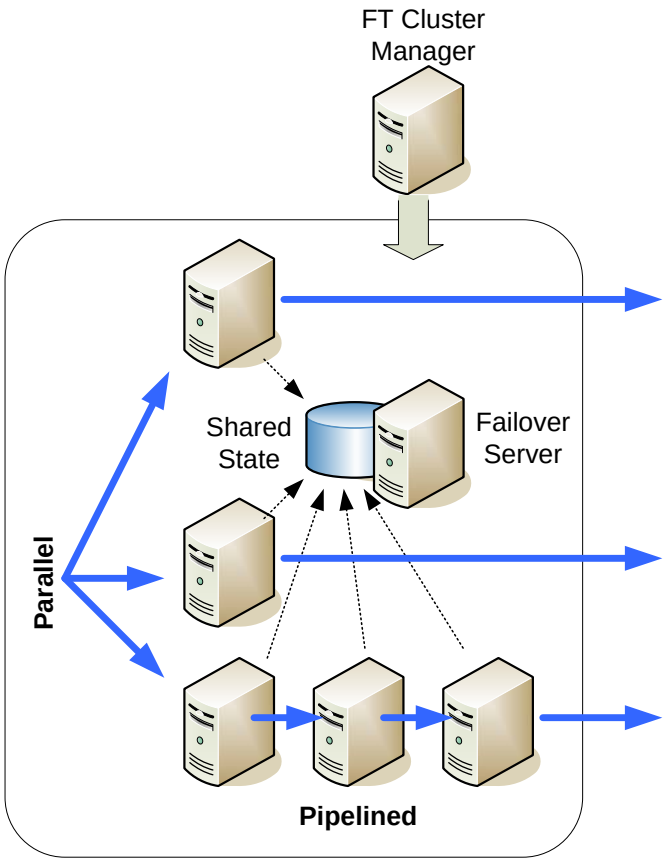

Figure 4: Flexible, M+N clustering model

Security

The Coral8 Engine provides granular security controls allowing IT organizations to easily specify and manage connections to the Coral8 system, and control access to various components within the Coral8 Server. Similar in nature to setting controls on database objects and connections, Coral8 administrators can declare entitlements for connections, access, and execution on workspaces, projects, query modules and streams, all managed by user, role, or IP address.

The Coral8 security system includes authentication plug-ins for Microsoft Active Directory, as well as LDAP and Apache-style htpasswd/htgroup files. An SDK for creating custom user authentication plug-ins is also included. The Coral8 Server access control specification is a simplified version of the eXtensible Access Control Markup Language (XACML), making it very familiar to administrators.

Easy, dynamic deployment

The Coral8 Server supports zero-downtime, mission-critical environments. Developers and IT teams can dynamically deploy new query modules and reconfigure existing components without having to restart servers or disrupt applications.

Business User self-service

The Coral8 Portal provides an easy-to-use environment for business users to access, personalize, execute, and visualize dynamic queries on real-time data. The resulting composite dashboards offer dynamic display of real-time data assembled by CEP queries from multiple sources.

IT administrators define parameterized query templates for business users to run. Query execution wizards allow business users to quickly define personalized queries based on the Query templates. Intelligent charting services provide multiple visualization options for the real-time data.

Algorithmic Trading

Algorithmic trading refers to any quantitative model that automatically routes and/or executes orders according to the parameters built into the model. Algorithms are designed to perform correlation analysis, identify trading opportunities, determine when to run, and generate results based on specific benchmarks (e.g., Volume Weighted Average Price, Time Weighted Average Price, etc.). Aside from algorithms, algorithmic trading systems need to provide functionality for order processing, connectivity to exchanges and ECNs, and integration with existing systems for order management, risk management, compliance, etc. Figure 5 shows a high-level diagram of algorithmic trading implementation.

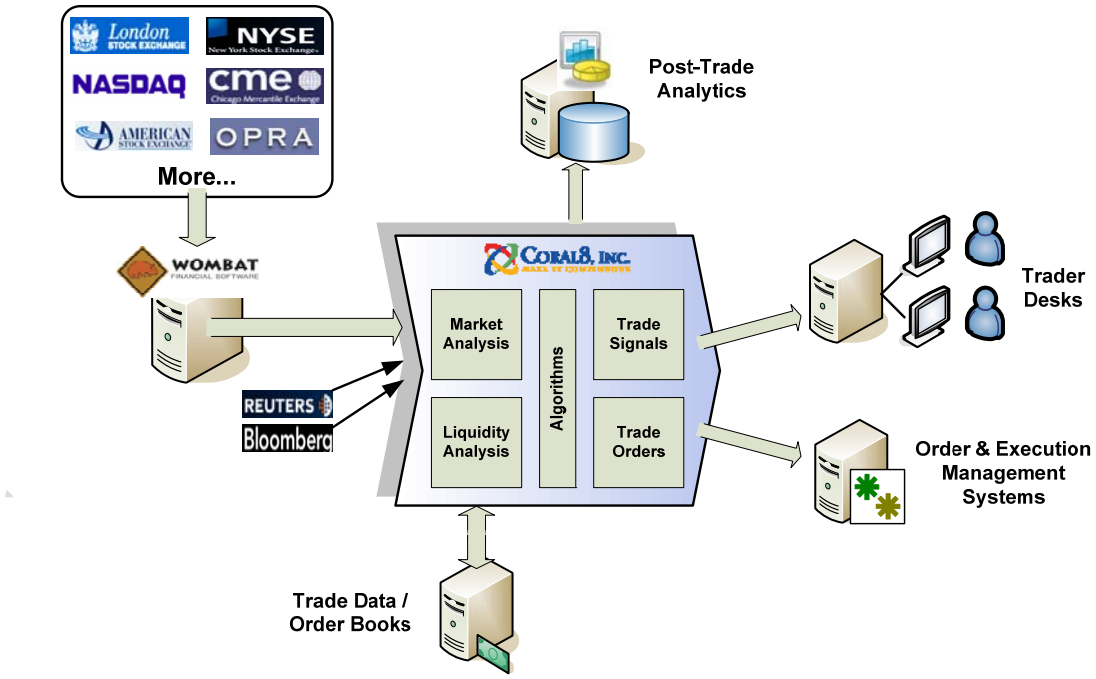

Figure 5. Algorithmic trading with Coral8 CEP

An algorithmic trading solution based on Coral8 CEP provides a modular, extensible approach to slice up trades automatically and execute them in the marketplace. At the heart of the application is a module that manages the "state" of client orders. Client orders are fed from the Order Book. When a new Order arrives, it requests a specific algorithm by which the Order is to be sliced into market trades, for example:

- Direct execution
- Direct execution
- VWAP and TWAP based spreads
- Spray algorithm
- AK47 algorithm
- Volume participation algorithm

The algorithm decides how much of the total order to trade at a given point in time. Each of the algorithms is represented in separate modules, allowing the customer to easily add more algorithmic modules over time. As each trade is completed, the order "state" is updated based on trade data flowing back into the application. Market data inputs are used to constantly calculate market metrics used by the algorithms to decide when and how much of an order to trade.

Financial Market Data Enrichment

Institutional traders and asset managers are continually asking for more sophisticated real-time market analytics to drive their trading strategies. Many of these customers have existing trading applications in place, and want to flow richer market data analytics into these applications. Figure 6 shows a high-level diagram of financial market data enrichment solution architecture.

In this solution, the Coral8 Engine subscribes to the market and trade data, performing any number of value-added calculations, such as VWAP, TWAP, proprietary indexes, arbitrage opportunities, risk models, and more. The results of the calculations are then placed back on the common market data infrastructure, where downstream applications can easily subscribe to the new enriched data.

In this example, the Coral8 Engine subscribes to the market data from the Wombat Financial Software market data backbone, and places the calculation results directly back onto this backbone for downstream applications to use. Existing applications do not need to be changed to take advantage of the new, enriched data. This service-oriented approach to enriched market data allows organizations to add new calculations at will with no disruption to the existing infrastructure.

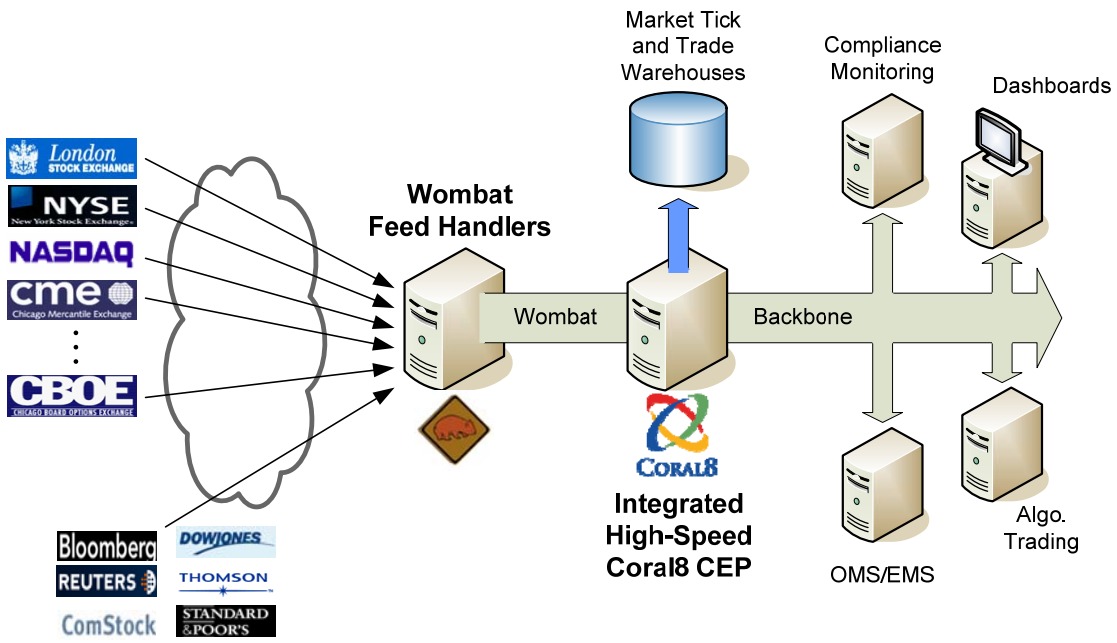

Figure 6. Financial market data enrichment

Dynamic Pre-trade Analytics

Many brokerages and market information service providers offer pre-trade analytics to their clients to help them make better trading decisions. The better the pre-trade analytics data, the more likely the client will use the broker to conduct the trade.

For many brokers, however, the process of generating pre-trade analytics is long, slow, and cumbersome. For example, one major broker took 96 hours to process one day of market data and generate the resulting pre-trade analytics. Smart, highly competitive brokers are moving to real-time generation of pre-trade analytics to serve as a competitive differentiator and a mean to lower the cost of generating the analytics. Figure 7 shows a high-level diagram of a dynamic, real-time pre-trade analytics solution.

Existing analytics generation process typically takes market data extract files, and runs the data through a complex series of sophisticated inter-dependent calculations. In these sequential batch processes, intermediate results are continually stored, and then subsequent processing steps are invoked. The process is painstakingly long, and often uses brittle, custom coded analytics processing modules.

Using Coral8, the CEP engine can directly subscribe to the market data and continuously generate the pre-trade analytics in real-time, feeding real-time dashboards, on-demand operational database, and long-term historical warehouse – all in parallel. The various analytical/computational rollups can be

modularized and pipelined for efficient, in-memory data passing. The high-speed processing model of the CEP engine streamlines the data flow and execution of the analytics.

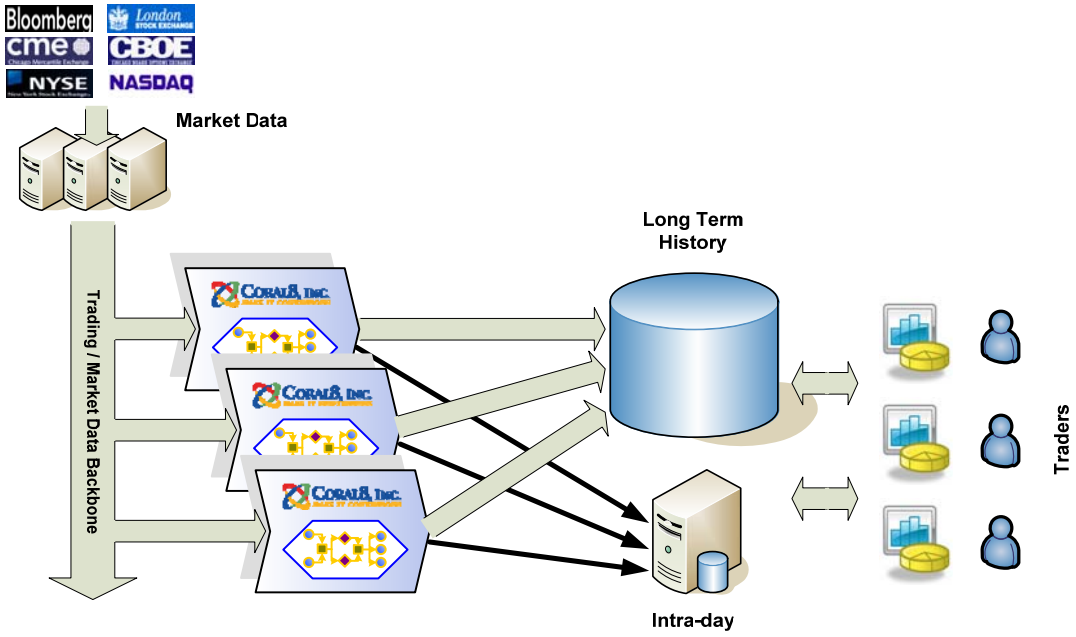

Figure 7. Dynamic pre-trade analytics

Coral CCL, the data-centric SQL-like language, greatly simplifies the coding and maintenance of analytics. The language and modularity also make it fast and easy for the broker development team to add new pre-trade analytics as customers demand, reducing their development and maintenance costs.

Institutional Trading Network Monitoring and Best Execution Analytics

Trading volumes on private trading networks are growing dramatically as institutional investors look to trade blocks of stock without causing significant changes in share price or alerting other investors to their intentions, as typically occurs on traditional stock markets. To compete successfully against open exchanges, private trading networks need to deliver two critical attributes on their trading network: anonymity and execution quality.

Coral8 CEP can be used to drive real-time trading network analytics that keep a constant eye on customer-focused metrics such as execution quality and value added per trade, and continual surveillance on the network for illicit trader activity looking to expose positions. Figure 8 shows a high-level diagram of real-time trade monitoring and best execution analytics for Institutional Trading Networks.

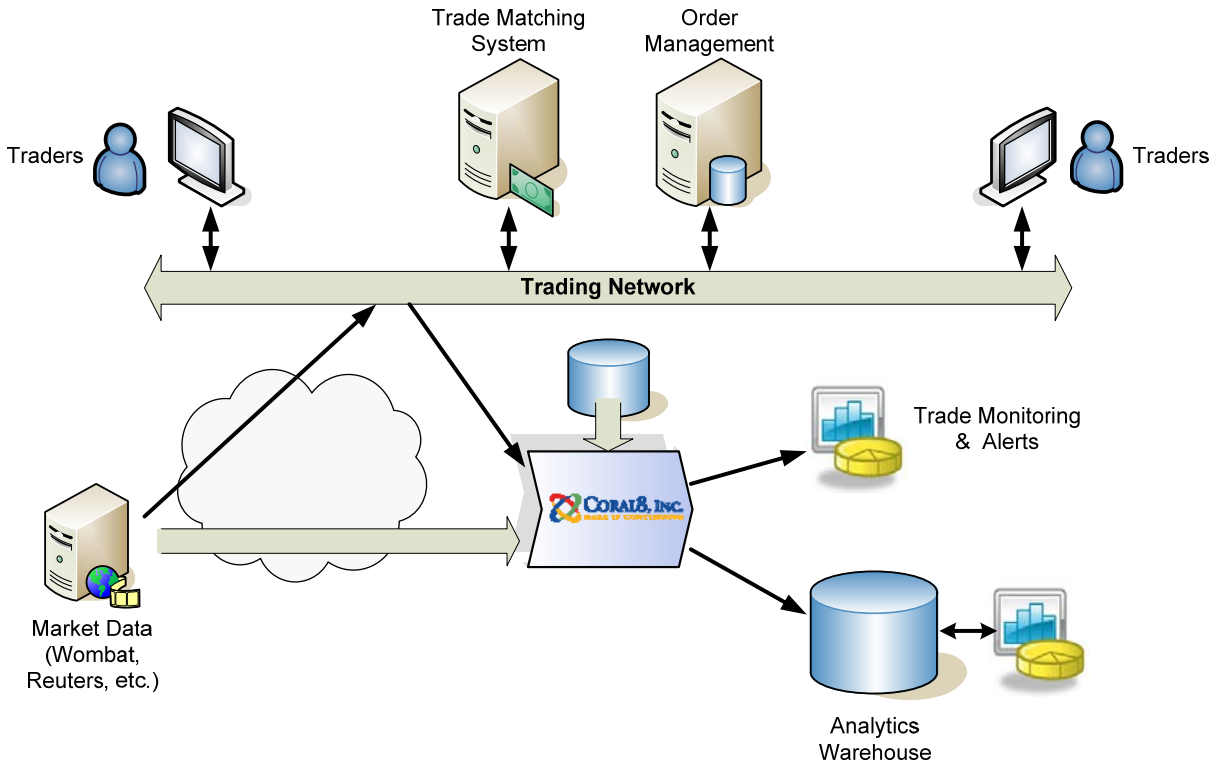

Figure 8. Trade monitoring and best execution analytics on an Institutional Trading Network

In the trader monitoring solution, trader interactions and activity are continually flowing into the CEP engine, where application modules use sophisticated queries with event pattern matching to identify illicit trader activity, such as "peeking" which exposes other trader's positions. When the pattern matching queries identify illicit behavior, warnings are sent to dashboard or via emails to account managers, who are responsible for warning or suspending accounts.

In the best execution analytics solution, market data continually flows into the Coral8 Engine, where open market metrics are constantly maintained for all symbols, such as VWAP, TWAP, Best Bid and Offer (BBO), execution routing and costs, and more. When trades are conducted on the internal institutional network, the Coral8 application matches the symbol with the appropriate set of real-time open market metrics, and captures the best execution, value-added per trade, and other reports. The results are stored in an analytical warehouse for later reporting, and alerts for "out of bounds" trades (for example, outside the BBO) are posted to dashboards.

The real-time analytics capture represents a much lower cost and easier to maintain analytics processing model than the alternative, which is to capture ALL market data and trades into a warehouse and match the data in the reports after the fact. In addition, alerts can indicate immediate actions the team needs to take on out-of-bounds trading.

Conclusion

Complex Event Processing represents a new paradigm by which real-time, event-driven business applications can become a reality in a low-cost, rapid development model. While CEP is a new paradigm, it leverages existing languages, tools, and approaches to give developers a familiar environment that dramatically reduces the learning curve.

Microsoft and Coral8 provide many key technology integration points to assist financial services developers with building the new event-driven applications. Tight integration between Coral8 Server and WCF, WPF, Office Excel, Office SharePoint Server, and SQL Server make it possible for Microsoft developers to leverage their existing skills and products deployed in the enterprise when building CEP solutions.

Capital Markets and financial trading have always pushed the leading edge of real-time applications to create competitive advantage. It is no surprise that many of the application use cases and early adopters of CEP are in financial trading. A number of investment firms and service providers have implemented new real-time trading applications similar to those described in this white paper using the Coral8 CEP platform, creating competitive advantage for their organizations.

References

Coral8 Web site – Provides information about Coral8, its products, and available services

MSDN Financial Services Industry Center