

XML Processing for Investment Banks:

Rapid time to market for new products and scalable derivatives processing using open standards and MarkLogic Server

Table of Contents

- 1 | Synopsis
- 2 | Derivatives Warehouse – A Reflection of Reality
- 3 | The Derivative as Content
- 3 | An XML Content Platform
- 4 | Native XML Persistence
- 4 | Universal Indexing
- 4 | High Volume Transaction Processing
- 4 | Conformance to XML standards
- 5 | ACID compliance
- 5 | High availability
- 5 | Summary

XML Processing for Investment Banks:

Rapid time to market for new products and scalable derivatives processing using open standards and MarkLogic Server

Synopsis

In the front and back offices of the investment bank, the speed of processing XML can have immediate impact on the success of the business. In the front office, the time to market for implementation of new products in trade capture systems has direct impact on the profitability of the bank. Similarly, in the back office, as the volume of trades has grown, and as the variety of derivative instruments has flourished, the prospect of rapidly analyzing all positions has become unattainable for many banks. This paper describes an approach for the storage, processing, and analysis of derivatives trades that will benefit the investment bank in the following ways::

- Rapid implementation of new products using XML and open standards
- Visibility for all positions across all trading systems
- Transparency for counterparties, auditors, and fund administrators
- Reducing the time for trade confirmations and full documentation completion, thus lowering risk and increasing profitability
- Allowing all business units to gain meaningful information quickly to aid in their various tasks, such as a Client Director validating the status of his customer conversation, or compiling a new deal
- Improving efficiencies in trade processing, from improved trade capture to portfolio reconciliation



Derivatives Warehouse – A Reflection of Reality

As the trading of derivatives has shifted over the past decade from a low-volume, high-margin business to a high-volume, low-margin industry, several opportunities for increased efficiency and predictable accuracy have materialized in the back office of investment banks. However, the sheer number of derivatives instruments, their ever-increasing levels of complexity, and the fast-changing structure of trades, poses significant challenges to firms who wish to provide real-time analysis of their reconciled positions.

Many investment banks conduct their derivatives trading business with partners using multiple, heterogeneous systems. Each system typically encodes the trade in a specific representation, ranging from compressed binary formats, to industry standard schemas such as FpML. Standardization of all trading systems is, in most cases, impractical. The reality is that the analysis of positions necessarily requires the ability to accommodate multiple trade representations simultaneously.

Should the bank wish to gain a holistic view of trades across all systems, consolidation of the different trade representations can result in significant latency in availability of information, in turn resulting in lost opportunities and increased exposure to risk. These limitations are introduced by the design rigors of relational database

systems, whose performance and functionality are predicated on predictable information structures, and predictable analysis patterns. As the information becomes more complex, and the volume of trades grows, adherence to these rigors increases in criticality.

Even in the case of a single trading system, based on FpML, it is realistic to expect that during the retention period of the trading information (e.g. several decades), the underlying schema will change. Historically, FpML has produced a new schema at least once per year over the past five years. A bank therefore may find itself in the invidious position of choosing between an endless data migration exercise to ensure access to all trades with a partner over time, or to accommodate the increasing risk that goes hand in hand with increased latency in trade analysis.

Furthermore, the time to develop each type of analysis can be substantial, and must be defined well in advance of the consolidation. Ad hoc analysis of the reconciled positions can be impractically slow, limited in options, or both. It is clear that these types of analysis are beneficial to the bank, and in many cases critical. The questions then is how to enable such beneficial analysis, and how to continuously adapt to the changes of the derivatives instruments.

The Derivative as Content

In this paper we distinguish between data and content. Data is understood to be predictable in type and structure. A purchase order, at least in its simple form, is considered to be a good example of data – the parties are defined, as is the item, its price, and quantity. All of these characteristics of the purchase are relatively simple to express in the relational model.

Content, on the other hand, is understood to be unpredictable in type and/or structure. A contract falls more on the side of content; some of its types may be known, such as dates and monetary values, and some of its structure may be known, such as parties, clauses, and the ordering of paragraphs, but on the whole “valid” contracts vary enormously from instance to instance. Expressing a contract in a relational model is possible, but difficult and complex. Creating one model that could encompass all contracts is impractical, unless that model is to store a few pieces of metadata about the contract, and to store the contract itself as an opaque object (ie, LOB) whose valuable contents are sequestered from the system’s functional abilities.

FpML, and many other XML schemas, are much closer in complexity to the contract described above than they are to a purchase order. They too are difficult and complex to model in a relational model and present especially difficult challenges to implement at large scale with substantial system load. This is not because there is something broken in the relational database management system; rather, it is because they were not designed for this type of information.

An XML Content Platform

It turns out that XML content appears in many different industries, not only financial services. Information providers, companies who generate profits by buying and selling information, have been addressing the challenges of XML content for over a decade.

The success of these companies can in many ways be determined by the agility and efficiency with which they address change in the information they process as well as the ways in which it is assembled as product. Explosive growth in demand for electronic information over the same period has added a dimension of scalability to this complex problem in the range of terabytes of XML and hundreds of queries per second, with expectations for the precision of a relational database and the speed of an internet search engine.

MarkLogic Sever was created to address the complex nature of XML content. Some specific assumptions were involved at the outset of these efforts, assumptions very different from those of 30 years ago when the relational database was invented:

- The schema is unknown
- Multiple schemas may coexist
- The system should be ACID compliant
- Documents may be small (1KB) and large (1GB)
- Collections of documents may be small (thousands) and large (billions)
- XQuery is the query language
- Indexes should be automatic and universal
- The system should be robust on commodity hardware

The result of these efforts is the industry’s leading enterprise class XML content platform. There are a number of functional areas that address the needs of the investment bank working to implement agile XML processing:

The sheer number of derivatives instruments, their ever-increasing levels of complexity, and the fast-changing structure of trades, poses significant challenges to firms who wish to provide real-time analysis of their reconciled positions.



Native XML persistence

Documents are stored as documents; they are not deconstructed into more primitive data types. XML is parsed by the server, indexed, and stored in a proprietary compressed DOM. In MarkLogic, the XML is parsed once – when it is loaded – and never again. All interactions with the document occur with this optimized, compressed form.

Documents in MarkLogic are organized by a variety of means, including directories, collections, security, and metadata. Interacting with documents in MarkLogic is similar to interacting with a file system – take any webDAV client, such as Windows Explorer, and access documents and directories in MarkLogic to read or edit in a familiar fashion.

Universal indexing

Because one of the design assumptions of MarkLogic is that the schema is unknown, documents of multiple schemas may coexist in the same contentbase. When a document is loaded or updated, values of all elements and attributes are indexed, as are the qualified names of all elements and attributes, as well as the hierarchical relationship of each element in the context of the document.

This combination of values and structure is encoded in a patented index called a universal index. By combining what are typically separate indexes into a single universal index, it becomes possible to perform rapid evaluation of arbitrary XQuery expressions against billions of variant structured XML documents.

The universal index is completely automatic. It is unnecessary to explicitly declare which elements and paths should be indexed by the server: all elements in all paths are indexed by default. When new elements appear in documents, they are transactionally added to the index with no intervention.

High Volume Transaction Processing

Excellent and predictable performance is a key characteristic of MarkLogic. Many features come together to enable high performance in a robust system. Rapid in-memory ingestion, asynchronous index optimization, sequential disk IO, the universal index, 64 bit architecture, multiple layers of distributed caches, and other features all collaborate to provide millisecond response time against multi-TB collections of XML content. While there are no standardized benchmarks for XML content, Mark Logic has conducted numerous exercises with customers that include:

- Over 1 billion documents
- Hundreds of simultaneous schemas
- Hundreds of queries per second
- Over 4MB/second per CPU sustained load rate
- Over 1.5MB/second per CPU sustained load rate with simultaneous read/update/load operations

Mark Logic has reference customers in production with multi-TB systems and these characteristics.

Conformance to XML standards

Mark Logic is a key participant in the W3C committees for XQuery, XPath 2.0, XML Schema, and other relevant standards. MarkLogic Server provides the most complete implementation of the XQuery standard, as well as hundreds of extensions to the standard for features such as update, XML search, try/catch, security, triggers, and many other types of semantics.

ACID Compliance

MarkLogic Server provides a transactional system that adheres to the ACID model. Transactions are handled in a fully non-blocking manner. This type of architecture ensures users will never have to wait on the read/write operations of others. In order to provide transactional capabilities MarkLogic Server implements a temporal database. In MarkLogic, all transactions are evaluated at a system timestamp, and documents are valid at a specific timestamp. Updates are not processed in place, thereby eliminating the complex and expensive overhead of disk management incurred by most relational database management systems. All updates are processed in memory and written to disk in an optimized, asynchronous, sequential disk write. Optimization of indexes is automatic, asynchronous, and processed as serialized disk IO.

High Availability

In a MarkLogic deployment high availability is delivered through several mechanisms including journaling, clustering, sub-second auto reboot, automatic failover, and hot backup/restore. The journaling system used in MarkLogic ensures data integrity during update operations. A clustered architecture spreads the storage and query processing load across multiple servers over commodity hardware and gigabit Ethernet. Sub-second auto-reboot reduces system downtime for individual node failure, while automatic hot failover ensures transactional integrity through node recovery. Hot backup and restore capabilities allow for system administration and maintenance without impacting system performance or uptime. Hot cluster administration, including cluster scale out for query throughput and data volume, allow for rapid operational response to the needs of the business.

These features, taken together with the ease of installation, deployment, and administration that are characteristic of MarkLogic Server, provide an exemplary platform for XML processing and analysis.

MarkLogic Server provides a reliable, scalable platform for storing, managing, and analyzing large volumes of trade information in different structures to accelerate the processing of trades, to enable powerful analysis of all positions, and to adapt to the rapidly changing landscape of derivatives instruments.

Summary

Derivative trades are complex, binding legal documents that continue to increase in complexity and kind. The ability of a bank to analyze their positions is critical to their success in managing risk as market conditions change. MarkLogic Server provides a reliable, scalable platform for storing, managing, and analyzing large volumes of trade information in different structures to accelerate the processing of trades, to enable powerful analysis of all positions, and to adapt to the rapidly changing landscape of derivatives instruments.



Mark Logic Corporation

www.marklogic.com

Headquarters

999 Skyway Road, Suite 200
San Carlos, CA 94070

+1 650 655 2300

New York

+1 646 378 2104

United Kingdom

+44 (0) 207 643 1712