

Caché and Data Management in the Financial Services Industry

A white paper by:

InterSystems Corporation

Caché and Data Management in the Financial Services Industry

Executive Overview

One way financial services firms can improve their operational efficiency is to revamp their data management infrastructure. Creating a central repository for data that is used by multiple applications can ensure data consistency and quality across the enterprise, ease integration bottlenecks, and lower the number of failed trades.

However, different applications have different database usage patterns. To satisfy them all, any central data repository must:

- Support a large number of transactions with high performance
- Provide fast response to queries, using up-to-date information
- Provide reliable, scalable persistence for data
- Be easily accessible by several disparate applications

This paper discusses various database technologies and how well they meet the criteria for a central data repository. Traditional relational databases generally do not have the performance needed to support running queries against “live” transactional data. Also, it can be difficult to connect them with applications written in object-oriented technologies. “In-memory” databases are fast, but not very scalable. Nor do they provide reliable data persistence.

Caché, a high-performance, massively scalable, multidimensional database, is an ideal choice for financial services firms looking to streamline the way they manage critical data. Its transactional bit-map indexing technology supports simultaneous querying and transaction processing. Plus, Caché’s wide-open connectivity lets it communicate with many disparate applications, including existing data stores, thus allowing organizations to make incremental changes to their data management infrastructure.

Caché and Data Management in the Financial Services Industry

Introduction

Within many financial services organizations, there is a growing trend towards internal data consolidation. Executives at buy-side and sell-side firms are realizing that since many of their applications work off the same body of data, they can improve their operational efficiency by keeping common data in a central data repository. Such an approach will improve the consistency and quality of data used by critical applications. It will alleviate the “integration bottlenecks” caused by the need to synchronize data between disparate applications within the enterprise. It will enable real-time reporting and analysis using current data. The result will be fewer failed trades, faster performance, and better decision-making.

But creating an internal central data repository can be a challenging task. Ideally, a central data repository should combine the attributes of both a transactional database and a data warehouse. It must be able to process a high volume of updates, while at the same time providing fast responses to queries involving large amounts of current and historical data. Another challenge is that different applications may be written in different languages, use different data access protocols, etc. Any central database must be compatible with them all. And finally, the ideal central data repository will allow a modular approach to data consolidation. It will work with existing messaging systems, and with existing data stores, allowing organizations to leverage the investment they have already made in these areas.

This paper will discuss several database technologies commonly used within the financial services industry, and how well they meet the requirements of a central data repository. It will explain why Caché, the high-performance, multidimensional database from InterSystems, is the best choice for data consolidation, and give an example of how a financial services firm might use Caché improve their operational efficiency by transforming their data management infrastructure.

The Challenge of Data Consolidation

The need to continuously improve operational efficiency, as well as mandates to provide straight-through processing and real-time compliance monitoring, is driving many financial services organizations to revamp their data management infrastructure. Since many applications within buy-side and sell-side firms access and update a common body of data, it makes sense to consolidate that common data in a central repository. Having one source for critical data ensures that the information used by disparate applications within the enterprise is both consistent, and up-to-date. This will result in fewer failed

trades, better performance (because there will be less need to spend time reconciling data between systems), and decisions based on current data.

Although consolidation of data has great benefits, some firms are not ready to embark on a large data centralization program because of the perceived cost and complexity. Ideally, a central data repository would interface with many systems, including applications that handle risk management, performance measurement, accounting, and trading, as well as reference data and corporate action services. In addition, the central repository should provide fast response to queries from analysts, data integrity users, and compliance users. To control costs and maintain smooth operations, many firms will want to take a gradual, modular approach to changing their data management infrastructure.

All of the needs mentioned above place large demands on the database technology that powers a financial services firm's central data repository. The choice of database technology for the new data infrastructure becomes critical. It must enable:

- **Reliable persistence for large amounts of data**
Data is the lifeblood of financial services organizations. They must be able to store and retrieve huge volumes of data in such a way that it is always available, and recoverable in the case of system failure.
- **High-performance completion of large volumes of transactions**
Some of the applications accessing a central data repository need it to be a *transactional* database. It must allow very fast reading, inserting, and updating of data. For these applications performance is key, because the faster the central database can process changes, the more transactions it can complete in a given time.
- **Quick response to complex queries of current and historical data**
Some of the applications accessing a central data repository will be running queries for the purposes of trend analysis, reporting, and compliance monitoring. Many of these applications will benefit by being able to access up-to-date information. (In the case of compliance monitoring, this is a legally mandated requirement.) Typically, database systems use indexes to shorten response times to queries, so an important attribute of any central database will be the ability to build indexes on rapidly changing transactional data.
- **Seamless and high-speed interaction with a variety of different applications**
Different applications within an enterprise may use different technologies, protocols, and standards. In order to interact with them, a central database may be called upon to communicate using Java, C++, .NET, XML, Web services, FTP, ODBC, RV, e-JMS, or just about any other technology available. However, merely interacting is not enough. Interfaces to disparate applications must be efficient and fast, so as not to slow overall performance.
- **Easy adaptation and extensibility**
Since many financial services firms will take an evolutionary approach to revamping their data management infrastructure, it is important that their central repository be easily adaptable, in order to accommodate new data as

applications are integrated into the system. Extensibility helps ensure that the central database will be compatible with emerging technologies.

In the remainder of this paper, we will discuss how various database technologies stack up against the needs of a central data repository.

Relational Databases

Relational database technology has been used to provide reliable data persistence for over 25 years. It has the advantage of using easily understood data structures – all information is stored in tables using a simple rows-and-columns format. Plus, relational technology has a query language (SQL) that has become the *de facto* standard throughout the database world.

However, relational technology falls short in situations when the very highest performance is needed. Its tabular data structures are not well-suited for storing complex, real-world information. Complex data is split into several interrelated tables, and must be “joined” before it can be accessed and used. The constant deconstructing and reconstructing of data increases processing overhead, and transactional performance tends to suffer.

The same sort of problem occurs when a relational database must interact with object-oriented technologies like Java and C++. In such cases it is necessary to create a “map” that translates data from the objects used by target applications and the tables used by the relational database. Not only does mapping add to the processing overhead, it also lengthens the development cycle.

The performance drawbacks of a relational database may not be apparent in simple applications. But they will become evident if relational technology is used to build a central data repository, especially when the attempt is made to build indexes in order to speed query response times. Relational databases are not designed to act as a transactional system and a data warehouse simultaneously.

That being the case, some organizations take the approach of building *two* central data repositories. One is a transactional system that periodically “feeds” the data warehouse, against which queries are run. However, in order to maintain adequate performance of the transactional system, downloads to the “query” system cannot be done very often. Thus, the applications that access the query system are usually working with day-old, even week-old, data. That situation is undesirable at best. And in the face of the new mandate for real-time compliance monitoring, it is unacceptable.

“In-Memory” Databases

So-called “in-memory” or “main memory” databases are exactly what they sound like. Rather than writing data to disk, they hold it all in local memory. Because there is no reading from or writing to disk, transactions can be processed very quickly. High performance is the main advantage of using in-memory database technology.

However, in-memory databases do not provide data persistence, and the mechanisms for data recovery after failure are not practical for most applications. In addition, because they require all data to be resident in memory, these systems cannot scale to handle extremely large volumes of data. In-memory databases may be useful for small applications where transaction speed is of utmost importance, but they are not suitable for building a central data repository.

The Caché Multidimensional Database

InterSystems Caché is a high-performance multidimensional database. It stores data in flexible, efficient multidimensional arrays, rather than in a simplistic rows-and-columns format. Caché’s unique architecture and capabilities make it an ideal choice for financial services firms that are looking to build a central data repository.

High Performance

Because it does not constrain data to a rows-and-columns format, Caché can store complex information without “deconstructing” it. By eliminating the joins (and associated processing overhead) common to relational technology, Caché can provide reliable data persistence *and* transactional performance that rivals the speed of in-memory databases.

Massive Scalability

Caché’s efficient multidimensional data structures are inherently more compact than relational representations of the same information. Because data is kept “intact” (that is, not split up into several interrelated tables), transactional performance does not suffer as the size of the database grows. In addition, Caché’s Enterprise Cache Protocol (ECP) dramatically reduces network traffic between application servers and the data server in distributed architectures, making it possible to add application servers and support even more users. Caché-based applications can support thousands of concurrent users without sacrificing performance.

Transactional Bit-Map Indexing

Caché is the only data management technology fast enough to enable the building of indexes on dynamic data. That means it can function simultaneously as a transactional database supporting large volumes of updates, and as a data warehouse providing quick responses to complex queries.

Seamless Connectivity

Another advantage of Caché's multidimensional data structures is that they can be automatically projected in a variety of forms, providing connectivity to other technologies, without "mapping" and the processing overhead that goes with it.

- **Relational connectivity**
Multidimensional arrays can be projected as two-dimensional arrays (tables), thus allowing Caché to "look" like a relational database. The Caché database is ODBC- and JDBC-compliant, and can be queried using SQL. In addition, Caché's Relational Gateway feature lets it access data stored in your existing relational databases.
- **Object connectivity**
Caché's multidimensional arrays are a good match for the complex data structures used by object-oriented technologies. Data can be stored as objects and projected in a variety of formats including Java, C++, COM, EJB, and XML. Plus, any Caché object method can be published as a Web service with just a few clicks of a mouse.
- **Connectivity with other technologies**
Caché can communicate via a wide variety of other protocols and standards including TCP sockets, FTP, and SMTP. It is also compatible with popular middleware and messaging technologies.

Such wide-open connectivity will allow a Caché-based central data repository to be seamlessly accessed by any application within your organization.

Rapid Development

Caché's object model supports concepts such as multiple inheritance and polymorphism that enable the reuse of components and speed application development. Object-oriented development techniques also make any Caché-based system easily adaptable and extensible.

A Sample Architecture

Figure #1 gives an example of how Caché might be used as a central data repository for an asset management firm. It shows Caché acting as a "transactional data warehouse" that supports STP efforts, risk management, performance measurement, compliance, and trading activities. A data infrastructure like the one shown will provide better data consistency and quality, and enhance operational efficiency.

Figure #1 – Data Management Infrastructure with Caché

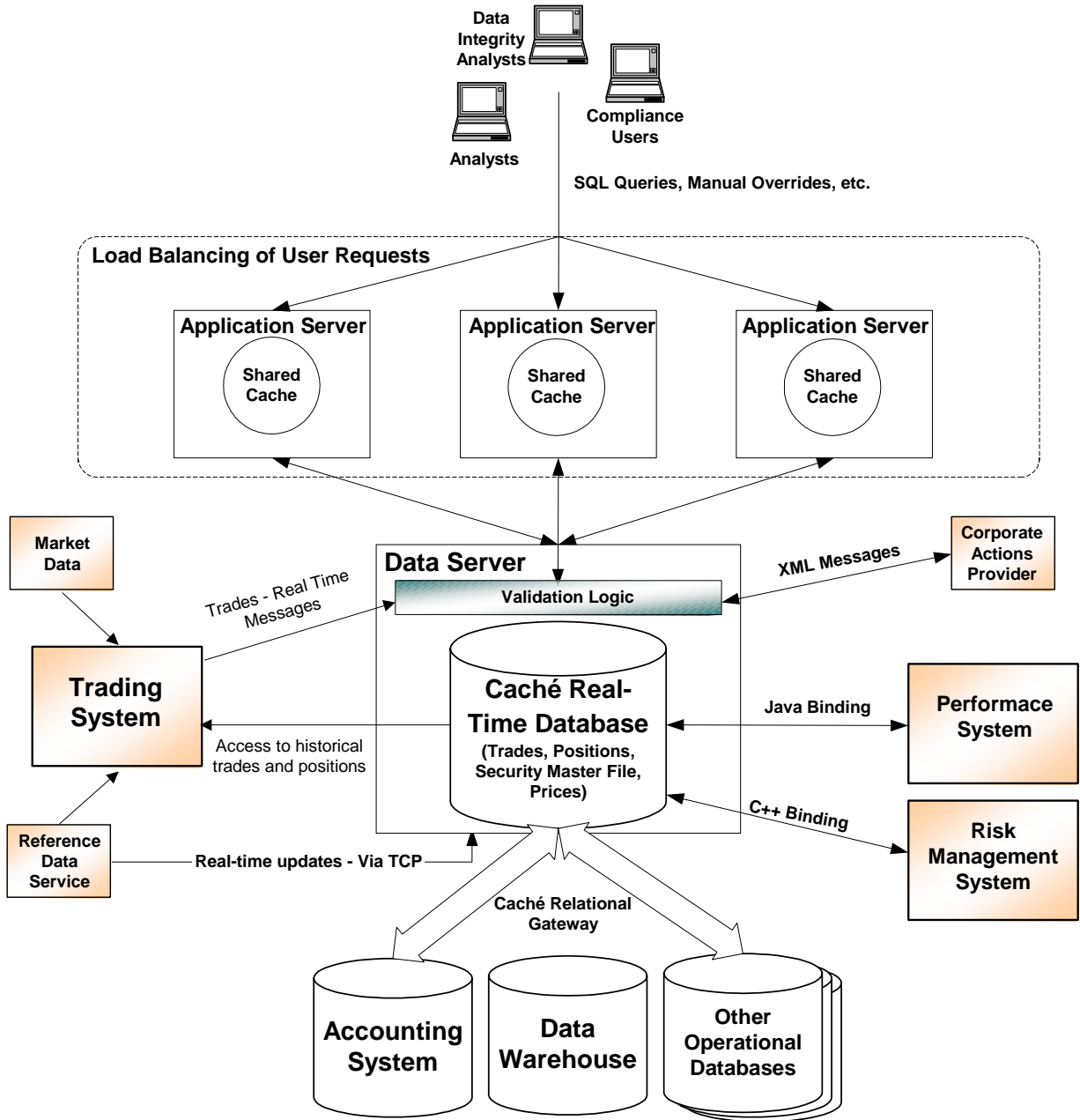


Figure # 1 shows the end result of what will, in most cases, be an evolutionary process of linking applications to a central database. In this example, here is a list of the various steps taken by our hypothetical asset management firm to gradually revamp the firm's data infrastructure:

1. **The risk management data model was implemented in Caché.** This was accomplished by importing the DDL from the existing risk management system and using it to define data structures within Caché. That done, some modifications were made, taking advantage of Caché's object technology to improve the data model. Then historical reference and position data were migrated into Caché. Caché's C++ object binding was used to communicate natively with the risk management system.
2. **The performance measurement system was connected to the central repository.** This required some further modifications to the data model, a task made easy by using inheritance – extending the classes created during step #1. In this hypothetical example, Java binding was required to allow the central database to communicate with the performance system.
3. **Reference data services and corporate action services were directed to feed the Caché database directly.** Thus the Caché repository became the “master copy” for all securities reference data, corporate actions, and securities prices (which were fed from the trading system). Batch feeds to the accounting system could be turned off. (Caché's Relational Gateway was used to periodically update the accounting system. In addition, the Relational Gateway allowed the central database to pull information from the existing data warehouse and other operational systems, as needed.)
4. **The trading system was connected to the central data repository in real time.** This allowed the running of compliance queries for all activity in the fixed income and equities systems. Caché's ECP was used to distribute the processing load created by a large number of analysts, data integrity users, and compliance users.

Conclusion

Financial services organizations can improve their data consistency and operational efficiency by changing their data management infrastructure so that critical applications access and update the same central data repository. To be effective, a central database must have certain capabilities, chiefly the ability to act simultaneously as a transactional database and a data warehouse. It must also provide high-performance connectivity to disparate applications that may be based on a wide variety of technologies, protocols, and standards.

Of the several database technologies commonly used in the financial industry, only Caché, the multidimensional database from InterSystems, has the performance, scalability, flexibility, and reliability to meet the rigorous demands of a central data repository.