



Reuters Develops High-Performance Financial Analytics Applications Within SQL Server

Overview

Country or Region: United States
Industry: Financial Services

Customer Profile

Providing content, analytics, trading, and messaging capabilities to financial services, media, and corporate markets, Reuters employs 14,500 people in 91 countries and serves 330,000 professionals.

Business Situation

As Reuters planned a many-fold increase in the companies, formulas, and history it covered, it needed to replace its historic financial analytic systems with new, faster, and more scalable solutions.

Solution

Reuters developers created new Microsoft® SQL Server™-based systems, with all business logic implemented in stored procedures written entirely in Transact SQL.

Benefits

- Enhanced developer productivity
- Scalability
- Faster performance
- Data-driven business logic
- Smaller code base
- Audit trail

“The SQL language has been an undiscovered jewel for almost 30 years.... [Using] T-SQL, we can get performance that is from 10 to 100 times faster than with a third-generation language.”

David Rozenshtein, Ph.D, Consulting Database Specialist, Reuters

Reuters' global information services help drive business decisions around the globe. When the company prepared for a manyfold increase in the number of analytic formulas and the depth of history it covered, it knew that it needed a new generation of financial analytics systems. The company had been using a collection of systems, created using various technologies, including Oracle and Sybase, with calculation engines built into the middle-tier with formulas hard-coded into the programs. By creating its new solutions using Microsoft® SQL Server™-based systems, putting all the business logic into stored procedures written entirely in Transact-SQL, and by treating formulas as metadata instead of hard-coding them in, the company has been able to increase performance by a factor of 10 to 100, while reducing the code base significantly and greatly simplifying code maintenance.

“Significantly faster performance can be obtained by simply staying inside the database. In the RFA system, we are making about 10,000 analytic computations per second on a basic four-processor computer.”

Sandip Mehta, Senior Database Architect,
Reuters

Situation

Reuters, which was founded in 1851, is a global information company providing data tailored for professionals in the financial services, media, and corporate markets. The company has a reputation for speed, accuracy, and freedom from bias; its information is trusted and drives decision making across the globe. (For a case study on how Reuters enjoys high availability for 10 terabytes of financial information, please see www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=16984)

Although Reuters is best known as the world's largest international multimedia news agency, more than 90 percent of its revenue derives from its financial services business. Some 330,000 financial market professionals working in the equities, fixed income, foreign exchange, money, commodities, and energy markets around the world use Reuters products. They rely on Reuters services to provide them with the information and tools they need to help them be more productive.

Reuters' core strengths lie in providing the content, analytics, trading, and messaging capabilities needed by financial professionals. With 57 percent of its revenue coming from Europe, 27 percent from the Americas, and 16 percent from Asia, Reuters offers financial services that span the globe. As Reuters prepared to significantly increase the number of companies it covered, as well as the depth of history it provided, it saw that the number of analytic computations would increase more than 10,000 percent. Reuters determined it would need to replace its existing historical financial analytics systems.

The company had been using a collection of systems, created using various technologies, including Oracle and Sybase, with calculation engines built into the middle-tier with formulas hard-coded into the application

programs written in C++, Java, and other third-generation languages (3GLs). The systems had served the company well, but Reuters determined that they lacked the flexibility and scalability to support the hundred-fold expansion. Reuters was concerned that the existing systems would suffer from slower performance because the new data volumes would amplify the time spent shuttling data between the application and the database. A new solution was needed that would:

- Scale to accommodate current and future growth.
- Enhance performance to satisfy the company's global customers who expect real-time and near real-time responses.
- Provide flexibility to easily change formulas.
- Maintain an audit trail of calculations.

Solution

Reuters was able to support its expanded coverage of global companies and at the same time provide more sophisticated equity analytics, by developing two new applications:

- **The Price Volume Analytics (PVA)** system collects and maintains history of daily prices and trading volumes from stock exchanges around the world, adjusts them for splits and other corporate actions and currency changes, and performs a wide range of calculations on the data.
- **The Reuters Fundamentals Analytics (RFA)** system computes more than 1,300 analytic formulas over the history of annual and interim financial reports as well as historic prices, historic foreign exchange rates, and other data sets. RFA stores the results in the database for further use.

These applications—which are architected as independent pipelined systems, that also feed data into other Reuters systems—were developed internally by Sandip Mehta, Senior

“Freeing our analysts from having to wait weeks to see the result of their ideas gives them what had been unimagined power to explore concepts and to make ever more precise analytical tools.”

David Larr, CFA, Product Architect for
Fundamental and Technical Analytics, Reuters

Database Architect, and David Rozenshtein, Ph.D., Consulting Database Specialist. The applications, which have been delivered for customers, are hosted with Microsoft® SQL Server™ 2000 Enterprise Edition running on Microsoft Windows Server™ 2003 Enterprise Edition operating system. Both Windows Server 2003 and SQL Server 2000 are part of Microsoft Windows Server System™ integrated server software.

Solution Architecture

The systems are architected to be loosely coupled with each other and with other Reuters systems. They each run on their own computers, asynchronously from other systems, and exchange data with each other and with other systems in a pipelined fashion.

This loosely coupled, pipelined architecture was a key element in allowing one of the systems to be easily updated as formula definitions and data values changed, while protecting downstream systems from massive and sudden consequent changes. One of the specific challenges was incorporating features to support maintenance of adjusted historical prices through stock splits, dividends, currency changes, and other events.

“Turkey and Romania recently went through significant revaluation of their currencies,” says Mehta. “That makes comparisons between the raw historical prices in old and new currency units completely meaningless. Similarly, one cannot simply compare numeric values of prices denominated in old European currencies such as the German deutschmark or French francs to those in euros. As another example, today’s share of Microsoft is not directly comparable to the one of ten years ago because of the many splits that occurred. Unless you can perform the necessary adjustments to the data, your historical information will be useless for analytics.”

Exchanges regularly publish corrections to previously released data. Without the ability to incorporate these corrections into historical data, the PVA application would lose its relevancy to analysts. The RFA application also dealt with similar challenges, such as companies restating earnings.

The architecture allows for results to be automatically recomputed, and in near real-time, rather than in batch processes.

“We architect our solutions as a pipeline of loosely coupled systems,” says Rozenshtein. “Whenever a system experiences a change in state, it posts information about what it has just recomputed to a log. Other systems, interested in changes by that system, monitor the log and incorporate the new data into their own computations. For example, if the foreign exchange system updates the value of the euro against the dollar, any systems that are dependent upon this information will recalculate using the new euro value.”

“It is important to note that this is all done from within SQL Server,” says Mehta. “Our solutions are created using a number of cooperating SQL Server processes running on several SQL Server instances. Every line of code is SQL. In a sense, the computation never leaves SQL Server. The logs are just tables. If we have to move data from one server to another, we use replication.”

Developed Using T-SQL

The key to the solution—including achieving the required flexibility, performance, and scalability—is that it was developed completely within SQL Server. The applications were written entirely in Transact-SQL (T-SQL) with all of the business logic implemented in stored procedures. T-SQL is Microsoft’s enhanced version of the Structured Query Language (SQL). System developers Mehta and Rozenshtein both say it would not have been possible to have

“The new RFA system has full audit trails so our customers can click on a figure and see exactly how it was actually computed.... [This] brings transparency to the decision-making process.”

Aref Rahman, Product Manager for Reuters Fundamentals, Reuters

created the functionality with the same speed, scalability, and ease of maintenance using conventional programming languages that operate outside of SQL Server.

“The SQL language has been an undiscovered jewel for almost 30 years,” says Rozenshtein. “More than 95 percent of developers facing such a project would opt to work in a third-generation language such as C++, Java, or C#. But as soon as you start moving large volumes of data from the database to the application and back, everything slows down and the overall costs become prohibitive. By creating data-centric solutions, which for us means writing the entire application in T-SQL, we can get performance that is 10 to 100 times faster than with a third-generation language.”

The data-centric approach to programming eliminates (or at least greatly reduces) movement of large data volumes between the database and other system layers. The data-centric approach also eliminates or minimizes the need for data transformations between them.

“The main principle is to keep computations near the data,” says Mehta. “You need to put the business logic into the database. Typically, creating the complete application using T-SQL gives you 10 to 100 times more efficiency than systems using the database as just a persistent store. You also benefit from using built-in database optimizers, which usually provide better results than the efforts of typical programmers.”

To support frequent formula changes, the RFA application treats formulas as metadata instead of hard-coding them into its source code. To do this, Mehta and Rozenshtein had to create their own formula interpreter, an effort that involved such original research that Reuters has applied for a patent covering their development. In creating the

formula interpreter, the developers again decided to work within T-SQL. “If we had created the interpreter within a 3GL, it would typically be capable of computing just one formula per company per historic period at a time,” says Rozenshtein. “The database support for such a computation would then be very expensive in terms of moving the data and results back and forth. On the other hand, when you develop in T-SQL, the data and computation never leave the database.”

Core programming practices that Mehta and Rozenshtein follow when developing with T-SQL include:

- Avoiding procedural programming and using loops only when absolutely necessary.
- Avoiding row-at-a-time programming, and in particular avoiding database cursors.
- Using a few complex SQL statements in place of many simple ones, thus minimizing the use of intermediate tables.
- Minimizing the number of passes through the data.
- Relying on the optimizer in the vast majority of cases, and doing custom optimization only when needed.
- Putting all code in stored procedures
- Using, generally, the full relational power of SQL.

Their best practices for designing application programming interfaces (APIs) include:

- Never directly access the database; instead, do everything through stored procedures.
- One business action equals one stored procedure call.
- Use XML as the in and out interface between the stored procedures and the outside layer.
- Write for a specific database, rather than attempt vendor-neutral applications.

"We like SQL Server Profiler because it allows us to see exactly what is happening in the server."

David Rozenshtein, Ph.D, Consulting
Database Specialist, Reuters

Mehta and Rozenshtein say that the efficiencies they gain from developing applications with T-SQL are so significant that they are surprised the development strategy isn't more widely used. "Everyone argues with our architecture until they see how it performs," says Mehta. "Then they want to know how we get such speed. We tell them not to leave the database."

Benefits

Developing applications using T-SQL gives Reuters a number of benefits, including enhanced developer productivity, scalability, faster performance, data-driven business logic, a smaller code base, and an audit trail.

Enhanced Developer Productivity

In addition to their most recent work on the PVA and RFA systems, Reuter developers Mehta and Rozenshtein have built a number of other important applications using T-SQL, including:

- Trading systems
- News management systems
- Data warehouses

The two like working with Microsoft SQL Server so much that they use it as their development environment even when working on projects that involve other databases. "If we have to work with Oracle, Sybase, or DB2, we still use SQL Query Analyzer to develop our queries, and then transfer the code over to the other environment," says Mehta. "We also like T-SQL more than Oracle's PL/SQL, because of the open result sets. T-SQL makes it much easier to debug stored procedure code."

Mehta and Rozenshtein use a development methodology based on progressive prototyping and continuous code refactoring. The database design, SQL coding, and logical optimization are all done by the same team—frequently just themselves.

"We tell our internal clients that if they can describe what they want to do with data, and then repeat the description without changing it [meaning that they have a firm concept in mind], we can create a solution that will meet their needs," comments Rozenshtein. "We can usually come back with a prototype of the application within just a couple of weeks, which is why we appreciate the simplicity with which SQL Server can be deployed. It configures itself right out of the box and is ready to go."

Mehta agrees. He adds, "With other databases we need to get a professional database administrator (DBA) involved just to set up the database for us to work with it. From the very beginning with other databases you have to start setting caches and other parameters. But we don't want to get a DBA involved until our project is well on the way. So we like that SQL Server is ready to go right out of the box. We can set up the database with just a few clicks."

They also enjoy working with SQL Server Profiler, which can be used to monitor and log server activity. "We like SQL Server Profiler because it allows us to see exactly what is happening in the server," says Rozenshtein. "With other database systems you have to go through significant work to gain the same information. Profiler makes it fast and easy, which helps speed up the overall development effort."

Scalability

Reuters benefits from the easy scalability of SQL Server. The developers often prototype on a laptop, simulating all of the additional databases and other systems that will be involved in the solution.

"Migrating from a single laptop to a room full of computers is simple," says Mehta. "Nothing breaks and nothing needs to be reprogrammed. That's the beauty of SQL

“We needed to create a solution that would support constant formula changes.... So we created our RFA system as an interpreter, so that any formulaic changes would simply be a matter of changing some metadata.”

Sandip Mehta, Senior Database Architect,
Reuters

Server. We set up replication and reverify the query plans for the more complex SQL statements, but beyond that there are no changes to the logic, no changes to the tables, no changes to SQL, and no additional programs have to be written. We simply hand it off to the DBAs, and they place it into production.”

“SQL Server allows us to seamlessly scale up from the laptop to full production because all the tools are already built in to it,” observes Rozenshtein. “It’s not like some systems in which the development environment is significantly different from the production environment, and additional code needs to be written to move from one to the next. With Microsoft SQL Server, it all just works. Promotion simply is not an issue.”

Faster Performance

The developers are sometimes asked to examine existing data-heavy applications in an effort to boost performance. What they find routinely is that the delays emerge because the application was written outside of the database so information needs to be shuttled back and forth between the application and database—sometimes numerous times to resolve a complex query.

“We recently helped with a conventional application that launched a query in Java. It retrieved a large data set, brought it back into the application, composed a new query based on the results, brought back a new data set, and so on,” says Rozenshtein. “Each business request involved three separate passes across the database, and three returns to the application. We pulled all of the application’s business logic into a single stored procedure, removed the code that was there just to support generation and use of intermediate results, and optimized it. The resulting application ran about 80 times faster.”

“Significantly faster performance can be obtained by simply staying inside the database,” says Mehta. “In the RFA system, we are making about 10,000 analytic computations per second on a basic four-processor computer. Everything we have seen makes us believe this simply wouldn’t be possible using conventional development practices of creating the application apart from the database—no matter how powerful the hardware you deployed on.”

Rozenshtein agrees, saying, “You can’t make up with hardware what you lose by treating the database as a simple repository.”

Data-driven Business Logic

Using a data-driven approach is an essential element of success when creating solutions for financial analytics. That’s because the underlying algebraic formulas are being changed continuously as analysts tweak formulaic variables seeking an ever more precise result.

“We needed to create a solution that would support constant formula changes,” recalls Mehta. “We discovered that it was hopeless to try to get a definitive set of formulas written down on a piece of paper with everybody agreeing to them, and with some assurance that the formulas would not change for a substantial period of time. Financial analytics is an area in which when you get two people together they’ll have three opinions. The analysts are continuously debating, ‘Is this the right formula, or maybe we should change this variable?’ This is good. This is what analysts do. Our goal was to make it easy for the analysts to change the formulas—which immediately excludes a conventional programming approach in which formulas are hard-coded into the application. So we created our RFA system as an interpreter, so that any formulaic changes would simply be a matter of changing some metadata.”

“The new data-centric analytic applications that are being created with SQL Server are directly benefiting Reuters’ customers, because they allow our analytic tools to continually be refined.”

David Larr, CFA, Product Architect for Fundamental and Technical Analytics, Reuters

The previous solution, which had been created with a conventional architecture, was difficult to change. “With the old system, changing a formula was a source-code event,” says Rozenshtein. “A spec had to be written, a programmer needed to be allocated, and after the change was made, the code had to be sent through regression testing to ensure that the changes didn’t create any unexpected problems elsewhere in the application. With the new solution, we simply change a piece of data representing the formula. No spec is needed. No code testing is required. The whole process might take an hour.”

“New analytic systems have given us a dynamic ability to reconfigure formulas,” says Aref Rahman, Product Manager responsible for Reuters Fundamentals. “This means that formulaic changes that used to take weeks can now be accomplished in an hour or two. And the developers are also creating tools that will make such changes self-service, so that our analysts can tweak the formulas on their own.”

Giving analysts the ability to experiment with their formulas unrestrained by the need to involve programmers, gives them the intellectual freedom to make their analytical tools ever better.

“Freeing our analysts from having to wait weeks to see the result of their ideas gives them what had been unimagined power to explore concepts and to make ever more precise analytical tools,” says David Larr, CFA, Product Architect for Fundamental and Technical Analytics at Reuters. “The new data-centric analytic applications that are being created with SQL Server are directly benefiting Reuters’ customers, because they allow our analytic tools to continually be refined.”

Smaller Code Base

Creating applications with T-SQL results in solutions with a substantially smaller code base compared to developing with other languages. The smaller code base, in turn, contributes to easier debugging and smaller project teams.

“We believe that SQL in general, and T-SQL in particular, are very powerful, very elegant, and well put together programming languages,” says Rozenshtein. “Through the use of joins, sub-queries, and other efficiencies of T-SQL you get very compact, clear, and maintainable code. We generally find that the code base is dramatically smaller than if the same solution worked outside the database—and it performs 10 to 100 times faster.”

“T-SQL is an incredibly powerful programming language, well suited for some very complex computations,” says Mehta. “Unfortunately, few programmers understand this. Many advanced data structure problems such as those on trees and graphs have remarkably beautiful T-SQL solutions.”

Reducing the overall code size produces a number of benefits—including making it easier for the developers to understand the complete functionality of the program. “We are often asked to do ports of applications that were written in C++, Java, or some other conventional language,” says Rozenshtein. “When we reduce the code base from 500 pages to 20 pages, we not only make it faster, we also make it easier for one person to get a complete view of what the application is doing and how it is doing it. It is hard to get your mind around 500 pages of code.”

“Reducing the size of the code also greatly simplifies debugging,” says Mehta. “You simply have fewer bugs in 20 pages of code than you do in 500 pages. And you have

“The customer reaction has been phenomenal. Our customers had no idea that we could provide an audit trail that was so simple to use and so deep and complete.”

Aref Rahman, Product Manager for Reuters Fundamentals, Reuters

fewer developers working on the code. All of this makes it easy to localize, identify, and solve issues early. We have a policy that whenever we have an application ready to release, we get some crisp, new one-dollar bills and offer a dollar for each bug found. We use our own money to do this, and it never costs us more than a few dollars.”

Audit Trail

The banks and other financial institutions who subscribe to Reuters financial analytic solutions are making decisions each day involving millions, and sometimes billions, of dollars. The people making these decisions need as much information as they can get, which is why Reuters created its Reuters Fundamentals Analytics service with built-in audit trails that show exactly how calculations are made.

“The old way of doing analytics gave customers a recommendation or other guidance, but didn’t show what exactly went into any given calculation,” says Rahman. “The new RFA system has full audit trails so our customers can click on a figure and see exactly how it was actually computed. This is very important to our customers. Even if a customer doesn’t agree with a specific finding, they can see exactly how it was created, and there is a lot of value in that. The audit trail brings transparency to the decision-making process.”

The audit trail means Reuters customers can spend less time on accumulating and verifying data on their own, and more time on the real value-add of determining how information will impact individual companies, business sectors, and markets in general moving forward.

“The audit trail also provides great validation of our services,” says Larr. “Our customers have always known that they were spending too much time on tasks that could be done

just as reliably by others, but with such big stakes, trusting the data was always an issue. With audit trail available, we are telling our customers that if they want they can check every number, because it is only a mouse click away. With the ability to trust in what we do, our customers can stop expending their own funds and time on duplicating the effort, and turn their attention to determining the best ways in which to act upon and add value to the knowledge.”

The audit trail is proving very popular with customers. “People do not like magic,” says Rozenshtein. “They don’t want black boxes. They want to see exactly how our work was done, and this is precisely what they can see by clicking on hyperlinks, drilling down as deep into the process as they like.”

“The customer reaction has been phenomenal,” says Rahman. “Our customers had no idea that we could provide an audit trail that was so simple to use and so deep and complete.”

Looking Ahead to SQL Server 2005

The developers are planning an early transition to SQL Server 2005 because of several new features and benefits, including:

- **Enhanced Performance** – The developers visited the SQL Server lab at Microsoft corporate headquarters in Redmond, Washington, to test run their systems on the beta edition of SQL Server 2005. Tests using the same data on the same hardware found that their PVA application ran twice as fast compared to SQL Server 2000. Processing speed doubled again when tested on the 64-bit version of SQL Server 2005 running on a 64-bit AMD computer.
- **Common Language Runtime (CLR)** – With the data-centric programming model used by Reuters, CLR integration with SQL Server 2005 is important because it

For More Information

For more information about Microsoft products and services, call the Microsoft Sales Information Center at (800) 426-9400. In Canada, call the Microsoft Canada Information Centre at (877) 568-2495. Customers who are deaf or hard-of-hearing can reach Microsoft text telephone (TTY/TDD) services at (800) 892-5234 in the United States or (905) 568-9641 in Canada. Outside the 50 United States and Canada, please contact your local Microsoft subsidiary. To access information using the World Wide Web, go to:

www.microsoft.com

For more information about Reuters products and services, visit the Web site at:

www.reuters.com

makes it easy to extend the functionality of T-SQL. CLR integration provides the option of performing mathematical calculations in a procedural language such as C#. CLR can also be used for creating custom aggregations.

- XML Enhancements – Reuters deals with a significant amount of semi-structured text-based data perfectly suited for XML representation. The developers are also using XML extensively in their APIs between stored procedures and other system layers. SQL Server 2005 includes XML as a native data type, as well as numerous and significant enhancements to XML parsing, generation, and querying.
- T-SQL Enhancements – SQL Server 2005 provides numerous T-SQL enhancements, including Ranking and Tiling functions, PIVOT and UNPIVOT operators and Recursive Queries, all of which allow for more efficient and compact native solutions to many classes of important business problems.

Microsoft Windows Server System

Microsoft Windows Server System integrated server infrastructure software is designed to support end-to-end solutions built on Windows Server 2003. It creates an infrastructure based on integrated innovation, Microsoft's holistic approach to building products and solutions that are intrinsically designed to work together and interact seamlessly with other data and applications across your IT environment. This helps you reduce the costs of ongoing operations, deliver a more secure and reliable IT infrastructure, and drive valuable new capabilities for the future growth of your business.

For more information about Windows Server System, go to:

www.microsoft.com/windowsserversystem

Software and Services

- Microsoft Windows Server System
 - Microsoft Windows Server 2003 Enterprise Edition
 - Microsoft SQL Server 2000 Enterprise Edition

Hardware

- Intel-based servers

© 2005 Microsoft Corporation. All rights reserved. This case study is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. Microsoft, the Windows logo, Windows Server, and Windows Server System are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.