TEMBO CASE STUDY

TEMBO Enterprise Modernisation Gains Momentum

Synopsis

Applying skills acquired by attending an advanced modernisation workshop presented by TEMBO, the author was able to redesign a problematic data integration project and reduce the runtime from 20 hours to 20 minutes.

Introduction

The case study deals with a frequently encountered integration issue, namely one of duplication and synchronization of data.

In South Africa the Financial Services Board (FSB) grants accreditation and maintains a central database of intermediaries and associated product types they are licensed to provide advice on and sell.

Regulation in the financial services industry requires Financial Service Providers (FSPs) to validate the accreditation of business written by intermediaries prior to accepting and issuing a contract.

Each FSP is obliged to have a local (satellite) copy of a subset of the FSB's master database to validate against during a new business process. The FSB's master data is continually being updated as the status and accreditations of intermediaries are updated.

An independent third party organization, Astute, receives nightly updates and is responsible for disseminating changes to FSPs using an xml messaging infrastructure based on IBM MQ series. Although the schema of these messages is fixed, FSP's are at liberty to store the data in whatever format they choose and using their own database preferences.

The Problem

Each FSP deals only with a subset of the financial intermediaries in the FSB's master database. They therefore generate a list of the intermediaries that their systems require and create an xml subscription message that is sent to Astute. Astute stores and acknowledges the subscription.

On a daily basis Astute sends each FSP any changes made to the entries contained in a FSP's subscription list. These updates are in the form of an xml message(s), and can potentially be several megabytes in size.

The process is reasonably straight forward, parse the xml message, split it into individual nodes/components, extract the data, and generate SQL statements to create, update or delete satellite database records.

Momentum's line of business systems runs on IBM iSeries platform. Based on available resources, a choice was made to implement a Java J2EE solution running on IBM Websphere Application Server, updating an iSeries DB2 database. (In hindsight, given recent enhancements in iSeries V7R1 DB2's and RPG XML capabilities, an RPG/DB2 only solution may prove to be a better implementation choice).

The first implementation of the project encountered significant performance problems when placed in production. Daily processing took 20 hours and consumed 5-10 % of the CPU load on an already powerful machine.

momentum

Analysis of the problem revealed a poorly designed schema resulting in badly performing SQL queries.

The Modernisation Course

Prior to being asked to investigate the performance issues relating to this project, the author had attended an Application Modernisation course, presented by TEMBO, a company whose sole focus is to provide tools and best practices for application modernisation.

> ***"Applying skills acquired by attending an advanced modernisation workshop presented by TEMBO, Momentum was able to redesign a problematic data integration project and reduce the runtime from 20 hours to 20 minutes."***
>
> Carl Conradie, IT Architect
> Momentum

The Process

The process first involved understanding the domain, the entities and relationships between these entities to make it easier to identify primary keys. This was followed by a complete redesign of the schema to ensure that a complete history of changes could be maintained and audits generated.

Lastly, design of referential integrity, using foreign keys appropriately, and check constraints ensured complete integrity of the data irrespective of the source of changes.

To address auditing requirements, a set of history tables were added to the schema and database triggers added to move changed/deleted records to history tables. The separation of history records into separate tables resulted in smaller datasets of active records. This had a significant impact on the performance of SQL query processing.

Once the database redesign was implemented using SQL's DDL (as opposed to the older DDS/ISAM), with RI and check constraints' applied, testing commenced. Moving referential constraints out of application code and into the database - where it belongs - proved to be critical in catching data errors during the testing stage. By allowing the SQL engine to deal with cascaded deletes simplified and reduced application coding.

The Result

The refactored solution was deployed and resulted in a daily runtime of less than 20 minutes – down from the original 20 hours. The solution has been in production for several months and not a single change has been necessary.

As always there are enhancements that could be made to improve performance further. One such refinement is to move the xml parsing and MQ processing to run on iSeries. Recent enhancements to RPG's XML processing capabilities in V7R1 release makes this attractive.

A new appreciation for, and understanding of IBM's enhancements to the SQL Engine has resulted. These insights have proved invaluable on subsequent modernisation projects.