Federal Student Aid (FSA)

FP Data Mart – Release 2

Stress Test

Creation Date: 05/28/02
Last Updated: 06/04/02

FP Data Mart – Release 2
Stress Test

Table of Contents

FP Data Mart – Release 2
Stress Test

1. Stress Test

1.1. Test Scope and Objective

While other testing efforts have verified both the functional and technical viability of the FP Data Mart business processes, the stress test verifies that the system can handle extremities in volume of the business process. The Stress Test will focus on the 'front end' user loads via the Microstrategy Web server and the Intelligence server.

Objectives

- Simulate production conditions in the testing environment, and then measure performance.
- Verify that high levels of transactions do not cause performance degradation beyond acceptable levels.

1.2. Approach

The approach details the tasks, owners, and associated outputs for the major steps covered during the stress test phase.

1. Test Team prepares a stress test plan, metrics, and configures Rational Load Test tool.
2. Development and the Test teams execute stress tests. Results and any problems encountered are documented.
3. Work Products:
    a. Stress Test Plan
    b. Stress Test Results

1.3. Definitions and Assumptions

1.3.1 Definition of Concurrency

One of the goals of stress tests is to determine how many concurrent users can be supported by the application. Thus, it is important to understand exactly what we mean by "concurrency." Concurrency can mean either:

1. The number of users simultaneously submitting report requests to Intelligence Server and / or Web Server; or
2. The number of users simultaneously logged in and submitting report requests within a given timeframe.

While the second definition of concurrency is more realistic, the first example is used for testing purposes to verify 'worst-case' scenario performance. For example, in the Data Mart, it is highly unlikely that 45 users would all request reports at the very same instant. At most, those 45 users might make different report requests within a five- or ten-minute span. Thus, the results from doing a stress test that starts 45 reports at the same instant represent a load larger than the system should ever have to handle.

1.3.2 What is Scalability?

A solution is scalable if, for all desired ranges of operation:

- **The architecture is free of structural limits.**

Scaling is effected through modular upgrades to existing architectural components rather than wholesale replacement of individual components. In other words, FSA will establish a baseline architecture that will grow as the requirements grow without having to redefine the architecture.

- **The cost to perform a marginal unit of work is no higher than the average**

For example, consider a solution that can service one query per second. If this solution can be enhanced to service 10 queries per second for less than 10 times the original price, we could conclude that it is a scalable solution. However, if the cost to increase capacity requires an incremental investment that is more than 10 times that of the original, the solution could not be considered scalable.

Given the above generally accepted principles, what exactly are the determinants of scalability? An environment must demonstrate consistent performance while supporting growth in:

1. Analytical Complexity
2. Atomic Data Volume
3. Number of Users

Analytical Complexity

The Data Mart applications must be able to support a range of analytical requirements, which generally fall into three categories:

Simple analytical requirements represent those reports that are populated from simple warehouse queries, such as SELECT statements including one or several columns from a table, with little or no runtime calculation.

Medium Complexity: More complex reports often require multiple queries (multipass) and generation of intermediate result sets, with mathematical and logical operations performed on the data before a final result is presented. For the purposes of this test, queries with a couple of passes of SQL are considered medium complexity.

High Complexity: reports require multiple queries and intermediate result set processing, combine several derived formulae into a single analysis, and perform advanced query generation, cross-tabulation, and report formatting at run time. For the purposes of this test reports with more than a couple of passes of SQL were considered high complexity.

As the complexity of reports for the FP Data Mart increases, so will the query times. The following table lists the FP Data Mart Reports with their level of Complexity and how often the reports will be run in production (Frequency).

Report Name	SQL Complexity	Frequency
1.1 Lender Scorecard Part I	Medium	Medium (1x per day)
1.1 Lender Scorecard Part II	Medium	Medium
1.1.1 Lender Portfolio Characteristics	Medium	Medium
1.1.2 Lender Default Claim Rate	Medium	Medium

1.1.3 Lender Change in Default Claim Rate	Medium	Medium
1.1.4 Lender Delinquency Rate	Medium	Medium
1.1.5 Lender Change in Delinquency Rate	Medium	Medium
1.1.6 Lender Origination Fee Variance	Medium	Medium
1.1.7 Lender Interest Adjustments	Low	Medium
1.1.8 Lender Cohort Default Rate	Medium	Medium
1.1.9 Lender Change in Loan Status	Low	Medium
1.1.10 Lender Purchases	Low	Medium
1.1.11 Lender Sales	Low	Medium
1.1.12 Lender Capitalized Interest	Medium	Medium
1.1.13 Lender Voids	Medium	Medium
1.1.14 Lender Program Review Summary	Medium	Medium
1.1.14.a Lender Program Review Deficiency	Low	Medium
1.1.15 Lender Audit Results	Low	Medium
1.1.16 Lender ED 799 Late	Medium	Medium
1.1.17 Active Lender ED 799 Missing	Low	Medium
1.1.18 Lender Scorecard Analysis Report	Medium	Medium
1.2.3 GA Federal Fund Report	High	Low (1x every other day)
1.2.4 GA Operating Fund Report	High	Low
1.2.5 GA Restricted Account Report	High	Low
1.2.6 GA Fee Payments	High	Low
1.2.6.1 GA Requested and Paid Fees	High	Low
1.2.7 GA Fee Payment History Report	High	Low
1.2.8 GA Delinquency Aging	Low	High (2x per day)
1.2.13 GA Closed School and False Certification	Low	High
1.2.14 Lender Portfolio Percentage by GA Report	Medium	High
1.3.1 School Change in Loan Status	Low	High
1.5.1 GA VFA Fees	Medium	Low
1.5.3 GA VFA Performance Measures	High	Low
1.6.1 FMS-NSLDS Cross-Check	Low	Low

Atomic Data Volume

The Data Mart should support incremental growth in database size while holding performance levels constant at best or with no more than linear degradation in performance at worst. This performance profile will only be possible if the warehouse can support growth in atomic data volume (the "raw" data providing the lowest possible level of detail) without requiring substantial modification to the warehouse architecture and database. The size of the data warehouse is generally dictated by the size of the atomic data. Along with this raw data, warehouses typically contain some pre-calculated results, descriptive or "lookup" data structures, and indexes. The ratio of the atomic data volume to the total warehouse volume defines the atomic density, i.e., the degree to which the "atomic" tables contribute to the over-all size of the warehouse. A database with low atomic data density typically includes a large amount of pre-calculated results or indexes, increasing the overall size of the database without increasing the overall information content. For the purposes of this test, indexes were created on fact and lookup tables mirroring the eventual production index strategy.

FP Data Mart – Release 2
Stress Test

Number of Users

The Data Mart should not need to support more than 50 users logged in at the same time in production. To meet the challenges of handling this number of users submitting a high number of queries per day and still provide acceptable response times, an architecture must be able to scale linearly as the user population grows and/or if that population uses the application more frequently. Linear scalability must allow for modular platform enhancements, not just at the data warehouse, but at all components of a middle-tier architecture, such as additional web servers to manage increased network loads as the user base increases and becomes more physically dispersed. In these cases, no changes to the underlying data warehouse design or MicroStrategy application architecture should be necessary to deploy the application to the expanded user base.

To test the three determinants of scalability, the following steps will be taken in this test:

1. A mix of simple, medium and high queries will be run against the data warehouse.
2. Canned reports will be used for Intelligence Server and Rational Test Manager for Web server, to simulate a number of concurrent users.

Our tests show a report complexity mix that will mirror production. The development environment with MicroStrategy Intelligence Server and Web Server on only one server will provide a testing environment harsher, or less query-friendly than will be found in production.

1.3.3 Testing Architecture

Below is a diagram showing the applications and operating systems for all the various tools used in this test.

	Client machine	MicroStrategy Web	Web Server	MicroStrategy Intelligence Server	Warehouse
Application	MicroStrategy Desktop 7.1.1	Web Server 7.1.1	Microsoft Internet Information Server (IIS) 4.0	MicroStrategy DSS Agent 7.1.1	Oracle 8.1.6
OS	Microsoft Windows 95, 98, & 2000	Microsoft Windows NT Server	Microsoft Windows NT Server	Microsoft Windows NT Server	HP Unix

1.3.4 Assumptions

The purpose of the stress test is to demonstrate the scalability of MicroStrategy tools in a worst-case scenario; this test is not intended to fine-tune the middle-tier pieces of a pre-production system.

1.4. Entry and Exit Criteria

It is critical that all entry criteria are validated before stress testing is initiated.

Entry:

- The stress test plan has been created.
- Rational Test Manager has been installed on a testing PC.
- The testing environment has been configured.
- The automated Rational Test scripts have been created.

Exit:

- All test scripts have been executed successfully.
- All results have been documented.

1.5. Intelligence Server Test

The purposes of the Intelligence Server stress test are to:

- Ensure that the machine running Intelligence Server can handle the load of a large number of simultaneous reports / users, and
- Determine the optimal number of Intelligence Server threads into the Oracle database.

1.5.1 Intelligence Server Stress Test Setup

We will stress the Intelligence Server with a number of simultaneous reports (simulating concurrent users running a report at roughly the same time). The Intelligence Server will be tested with a range of available threads open to the Oracle database.

The total time for a report to be returned is the Intelligence Server queue time plus the SQL execution time. When a report is sent to Intelligence Server, it waits in the Intelligence Server queue until an Intelligence Server thread becomes available, at which time the query is sent to the warehouse. With a given number of reports placed in the Intelligence Server queue, the time a report spends in the Intelligence Server queue drops as the number of Intelligence Server threads is increased. At the same time, as the number of threads increases, more queries are hitting the warehouse at the same time, and the queries will begin to compete with each other for resources on the database server. This will cause the SQL execution time for each query to increase. The purpose of these tests is to determine the middle ground, *i.e.,* the number of threads where the balance between reports in the queue and concurrently executing reports produce the best overall performance. This number of threads will be where the total execution time is lowest.

1.5.2 Intelligence Server Stress Test Procedure

The Intelligence Server Stress Test will track report performance between the Intelligence Server machine and the Test Oracle database.

Testing Architecture

Machines that will be used for the testing:

1. A Windows NT server (SFANT001) running Intelligence Server.
2. One machine running Windows 2000. The desktop will be used to submit reports via MicroStrategy Web.

Testing Procedure/Tools

1. The Rational Tool will be used to create an automated test that kicks off 45 reports at the same time.
2. Report Statistic Logging will be turned on in the FP Data Mart.
3. Using Intelligence Server Administrator, the number of available Intelligence Server threads will be set to an appropriate number for the test.
4. The test will be conducted in 5 passes. Pass one will test 5 open threads, pass two- 10, pass three- 15, pass four- 20 and pass five- 25.
5. When all reports have finished running through Intelligence Server, we will query the IS_REPORT_STATS and IS_REP_STEP_Stats tables to obtain the SQL execution time and the query finish time for each query.
6. The results for the test will be put into an Excel sheet for reporting purposes.

1.6. Web Server Test

The purpose of the Web Server stress test is to determine its capacity, *i.e.*, how many reports the Web Server can effectively process simultaneously.

1.6.1 Web Server Stress Test Setup

This stress test will be performed on SFANT001, which will have both Intelligence Server and Web Server installed. Note: In the production environment there are two Web servers and one Intelligence Server. This dispersed architecture will improve performance.

The purpose of this test is to see how many requests Web Server can handle without losing significant performance capabilities. End-to-end response time in production will be affected by many other variables, including end users' Web connections, VDC operations, and other factors beyond the scope of this test and control of FSA.

1.6.2 Web Server Stress Test Procedure

To conduct the test, we will use Rational Test Manager. We will use Test Manager to simulate varying numbers of users submitting different reports at various time intervals.

Test Architecture

Machines to be used for the testing:

1. A Windows NT server (SFANT004) running Intelligence Server.
2. A desktop running Windows NT and Rational Test Manager.

Test Procedure/Tools

1. Rational Test Manager Robot recorder will be used to record keystrokes simulating up to 45 reports running concurrently.
2. Rational Test script will be executed and the Test Manager will record the statistics.
3. The results for the test will be put into an Excel sheet for reporting purposes.