MyInfoVault

MIV Version 2
Load Test Results

Version 2.0
June 20, 2008

Prepared by: Rick Hendricks
Information and Educational Technology
University of California, Davis

Table of Contents

MyInfoVault Load Test Results

Introduction

This document summarizes the load testing results performed on the MyInfoVault (MIV) application using Borland's SilkPerformer 2007.

The purpose of the load testing is to evaluate MIV application performance under load to for the following criteria:

- Response Time – The response time under load must be evaluated to insure the user experience will not be adversely affected.
- Defect Identification – Performance under load can often expose deficiencies in the code that would not otherwise be detected through the normal QA process.
- Performance Tuning - Application performance under load can serve to highlight areas in the application configuration which may need to be adjusted in order to optimize performance.

Testing Approach

Functional Areas Tested

Since it was not realistic to test every page within the MIV application, several key functional areas were identified to be the target of the load testing. The functional areas identified are as follows:

- Login
- Publications
- Grants and Contracts
- List Of Service
- Manage Format Options
- Switch User
- Annotations
- Create a Packet
- Design a Packet

Separate load tests were conducted on each individual functional area with a final test being done with all functional areas included in a single load test.

User Group Profiles

Actual MIV user accounts and data were used for the load tests with the users divided amongst 9 profiles as shown below:

User Group	**Profile**					
	Connection Speed	**Browser**	**Cache Enabled**	**Cookies Enabled**	**User %**	**Users**
Level 1 (Staff)	T1	IE7	Yes	Yes	2%	10
Level 2 (Staff)	T1	IE7	Yes	Yes	8%	40
Level 3 (Staff)	T1	IE7	Yes	Yes	50%	250
Level 3a (Staff)	DSL	IE7	Yes	Yes	2%	10
Level 3b (Staff)	T1	Firefox 2	Yes	Yes	3%	15
Level 4 (Candidate)	T1	IE7	Yes	Yes	23%	115
Level 4a (Candidate)	T1	IE6	Yes	Yes	8%	40
Level 4b (Candidate)	T1	Firefox 2	Yes	Yes	3%	15
Level 4c (Candidate)	DSL	Opera 9.2	Yes	Yes	1%	5
					100%	**500 users**

Each testing run, unless otherwise noted, was conducted with 500 users over a period of two hours. For each test, the users were divided amongst the 9 profiles listed above.

Testing

The actual load tests were executed for each functional area for a period of two hours, with the startup of all 500 users occurring within approximately the first 60-90 seconds of the test. Random “think times” are used for each action the user may perform to simulate the amount of time a real user may pause on a page before taking an action.

The tests are characterized as “steady state”, which means all 500 users execute for the entire duration of the test. However, because of the rapid rate at which all 500 users are introduced into the test, the tests also served as a “stress test” to demonstrate the ability of the application to survive and recover from an extreme high load condition.

Results

Each functional area tested is comprised of multiple pages. SilkPerformer load tests generate large amounts of statistical data, broken down by each user. In the interest of simplifying the results and saving space, the results for each functional area include only a breakdown of the individual page response time averages with all times being given in seconds. The tabular results for each test are sorted in descending average page time order.

Each set of tabular results contains the following data:

- **Page Name** – The name of the Web page as defined in SilkPerformer.

- **Count** – The total number of requests for the page.

- **Total** – The cumulative number of seconds for all page requests.
- **Min** – The minimum page request time in seconds. At least one page request resulted in the **Min** page time.
- **Avg** – The average page request time in seconds. (Total / Count = Average)
- **Max** – The maximum page request time in seconds. At least one page request resulted in the **Max** page time.

Login

Test consists of simply having all 500 users hit the MIV Login and MIV Main pages repeatedly for the duration of the test.

Page Name	Count	Total	Min	Avg	Max
MIV Logout	500	910.72	1.03	**1.82**	4.98
MIV Main	626360	194402.44	0.00	**0.31**	14.19
MIV Login	626360	167354.38	0.00	**0.27**	29.66

Publications

Test simulates all users randomly adding, editing or deleting any one of the randomly selected publication categories for the duration of the test. The MIV ItemList page is the page from which items to add, edit or delete are selected.

Page Name	Count	Total	Min	Avg	Max
MIV ItemList	119732	30846.94	0.00	**0.26**	102.82
MIV Add Publications (Form)	39564	5087.47	0.00	**0.13**	47.25
MIV Edit Publications (Form)	32226	3704.80	0.00	**0.11**	44.81
Edit Record (Save)	76056	7098.86	0.00	**0.09**	24.99
Remove Record (Delete)	32333	3020.46	0.00	**0.09**	25.03

Grants and Contracts

Test simulates all users randomly adding, editing or deleting Grants and Contracts data for the duration of the test. The MIV ItemList page is the page from which items to add, edit or delete are selected.

Page Name	Count	Total	Min	Avg	Max
MIV ItemList	109438	32786.28	0.02	**0.30**	114.98
MIV Edit Grants and Contracts (Form)	35435	2944.09	0.00	**0.08**	41.53
MIV Add Grants and Contracts (Form)	36493	3087.59	0.00	**0.08**	93.74
Edit Record (Save)	71928	5817.51	0.00	**0.08**	4.78
Remove Record (Delete)	36279	2646.89	0.00	**0.07**	3.11

List Of Service

Test simulates all users randomly adding, editing or deleting List Of Service data for any one of the randomly selected List Of Service categories for the duration of the test. The MIV ItemList page is the page from which items to add, edit or delete are selected.

Page Name	Count	Total	Min	Avg	Max
MIV ItemList	105123	21085.92	0.00	**0.20**	108.01
MIV Add Service (Form)	34928	4274.74	0.00	**0.12**	95.13
MIV Edit Service (Form)	31242	3318.60	0.00	**0.11**	56.25
Edit Record (Save)	66169	5363.27	0.00	**0.08**	30.00
Remove Record (Delete)	31695	2651.37	0.00	**0.08**	27.03

Manage Format Options

Test simulates all users repeatedly accessing the Manage Format Options page and executing a save. No adds or deletes are done as this functionality required additional complexity in the scripts which could not be provided in the limited script development time available.

Page Name	Count	Total	Min	Avg	Max
MIV Manage Format Options (Form)	90832	9841.64	0.00	**0.11**	7.40
ManageFormatOptions (Save)	90832	4892.41	0.00	**0.05**	99.39

Switch User

Test simulates all users repeatedly selecting a random user account to switch to and then switching back to themselves. If a user is not authorized to switch accounts, they are eliminated from the test resulting in 166 users being eliminated from the test and the test being completed with 334 users. In this SilkPerformer test, there is a web page named for each user switched to, making it impractical to show each page in the table. Therefore the overall page time average is shown. The MIV Main page is the page from which the user switch is selected.

Page Name	Count	Total	Min	Avg	Max
MIV Main	334	2395.16	0.05	**7.17**	16.67
Overall Page Time MIV Manage Format Options	19319	9437.00	0.01	**1.38**	33.08

Annotations

Test simulates all users repeatedly editing a randomly selected annotation, adding/changing the Label Above text and executing a save.

This test was performed with 100 users. More users produced response times which *were unacceptable. It is understood that further analysis is needed to determine the refactoring and improvements which are needed to bring the performance of this test to an acceptable level. Such analysis and refactoring must be deferred to later phases of the MIV project.*

Page Name	Count	Total	Min	Avg	Max
MIV Manage Publication Annotations #2 (Save)	5965	10855.31	0.45	**1.82**	9.84
MIV Manage Publication Annotations (List Page)	5965	10411.84	0.38	**1.75**	22.39
MIV Manage Publication Annotations #1 (Form)	5965	10404.59	0.39	**1.74**	22.2

Create My Packet

Test simulates all users repeatedly creating their packet.

This test was performed with 50 users over a period of 1 hour. More users produced response times which were unacceptable.

The response time for Create My Packet is related to the time it takes for the FOP processor to actually process the XML and create the documents, which is a CPU intensive process. Therefore, when many users simultaneously create a packet, the response times will increase.

It is understood that further analysis is needed to determine the refactoring and improvements which are needed to bring the performance of this test to an acceptable level. Such analysis and refactoring must be deferred to later phases of the MIV project.

Page Name	Count	Total	Min	Avg	Max
MIV Create My Packet	2103	87400.12	1.77	**41.56**	74.48

Design My Packet

Test simulates all users repeatedly accessing the Design My Packet page and executing a save. No adds or deletes are done as this functionality required additional complexity in the scripts which could not be provided in the limited script development time available.

This test was performed with 25 users over a period of 1 hour. More users produced response times which were unacceptable.

The response time for Design My Packet is related to the large number of requests to retrieve and release connections from the database connection pool. When many requests are simultaneously made of the single connection pool, the requests are placed in a queue and acted upon one at a time resulting in a backlog of requests, which slows response times.

It is understood that further analysis is needed to determine the refactoring and *improvements which are needed to bring the performance of this test to an acceptable level. Such analysis and refactoring must be deferred to later phases of the MIV project.*

Page Name	Count	Total	Min	Avg	Max
MIV Design My Packet (Item List)	719	37404.01	2.33	**52.02**	96.33
DesignPacket (Save)	698	2823.04	0.16	**4.04**	19.13

All Transactions

Test simulates all users repeatedly executing any of the above transactions randomly, with the exception of the Annotations, Design My Packet, and Create My Packet transactions. These three tests were excluded do to their current performance. For further details, see the results sections for each.

As part of the test, each user randomly switches users (Switch User) and executes tests until such time as the Switch User transaction is randomly executed again, which returns the user to his own login (Return To Admin).

Page Name	Count	Total	Min	Avg	Max
Select Candidate Account (Switch User)	4482	3368.43	0.02	**0.75**	96.29
MIV ItemList	97370	49252.93	0.00	**0.51**	105.2
MIV Manage Format Options (Form)	10838	4925.13	0.00	**0.40**	93.71
ManageFormatOptions (Save)	10476	1640.26	0.00	**0.16**	26.63
MIV Edit Grants and Contracts (Form)	9691	1464.71	0.00	**0.15**	93.38
MIV Add Service (Form)	10487	1422.43	0.00	**0.14**	95.34
MIV Add Grants and Contracts (Form)	10475	1516.07	0.00	**0.14**	95.55
Return to Admin	2391	274.73	0.01	**0.11**	13.39
MIV Add Publications (Form)	10472	1100.06	0.00	**0.11**	35.45
MIV Edit Publications (Form)	8066	889.71	0.00	**0.11**	40.02
Edit Record (Save)	58928	5580.9	0.00	**0.09**	24.58
MIV Edit Service (Form)	8518	869.29	0.00	**0.01**	29.2
Remove Record (Delete)	26934	2808.67	0.00	**0.01**	45.53

* *Note: Edit Record and Remove Record are shared between the Publications, Grants and Contracts, and List of Service transactions.*

Afterward

Prior to compiling the final data presented in this report, many tests were conducted which produced unacceptable results due to poor performance. Repeatedly running loads tests in conjunction the JProfiler application profiling tool as well as monitors on the database and other troubleshooting techniques aided in uncovering the root causes of the poor performance. As a result, areas in the MIV code as well as web server and database configurations were modified in order for the MIV application to perform acceptably and successfully complete the tests.

Discovery of such application and configuration deficiencies could only be accomplished as a result of exercising the MIV application under heavy load. Otherwise, such problems would only manifest themselves under a heavy load event in production. Once in production problems related to load are often difficult to isolate and correct because of the somewhat transient nature of heavy load events.

What follows is a summary of changes made as a direct result of load tests. The changes were made over the course of many test and observation cycles

Apache Configuration

Bottleneck was observed in the concurrent connections to the Apache server. The following changes were made in the */etc/httpd/conf/httpd.conf* file:

- ServerLimit and MaxClients parameters were increased from 256 to 500 to handle spikes in the load.

```
ServerLimit     500

MaxClients      500
```

- KeepAlive parameters put in place to allow pages to be downloaded more efficiently.

```
KeepAlive On

KeepAliveTimeout 2

MaxKeepAliveRequests 80
```

Tomcat Configuration

Error messages in the tomcat logs indicated the thread pool was being exhausted under heavy load. The following changes were made in the /usr/*local/tomcat/conf/server.xml* file:

- MaxThreads for the tomcat server was increased from the default 150 to 300.

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009"
 maxThreads="300" minSpareThreads="25" maxSpareThreads="75"
 enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
```

Database Configuration

With 500 users hitting the application simultaneously, it was observed that the database connection pool was being exhausted under load. The following changes were made to the */usr/local/tomcat/conf/Catalina/localhost/miv.xml* file:

- MaxActive connection count was ultimately increased to 700 from 200.

  MaxActive “700”

- MaxWait time to wait for a new connection was increased to 30 seconds, up from from 15.

  MaxWait “30000”

- LogAbandoned parameter set in order to log when database connections are not being properly closed in order to isolate connection leaks.

  LogAbandoned “true”

Code Changes

Below is a summary of the coding changes made to MIV as a direct result of load testing. Changes were made over several iterations of a test and are not necessarily shown in the order in which they were made.

- MIVConfig object contained a synchronized method which is used throughout the application. Under load access to the synchronized method was a bottleneck. The code was re factored in such a way as to avoid the use of a synchronized method.

- Finally blocks added to all database access code in the SectionFetcher and RecordHandler classes. The use of the finally blocks insures that all database result set, statement and connection objects are properly closed. Leaving database objects in a open state was causing connection leaks and contributing to the exhaustion of the database connection pool.

- Refactor SectionFetcher and RecordHandler classes to use connection pooling rather than holding on to a single connection for the lifetime of those objects. While such a strategy removed the overhead of obtaining a connection, it also imposed a “hard limit” on the number of users which could be accommodated on the system since each user held a connection for as log as logged in. The hard limit is such a case would have been the maximum number of connections which are allowed in the database connection pool. Currently the maximum connection limit is set to 700.

- Refactor SectionFetcher and RecordHandler further to overcome threading issues introduced with the way in which the connection pooling was initially implemented.

- Update FieldFormatData and QuarantineData objects to close database connections retrieved from the RecordHandler object. Previously these connections retrieved in these two objects were being leaked.

- No longer synchronize the getConnection and releaseConnection method in the MIVConfig singleton object as it is not necessary.

- Refactor QueryTool to handle the pooled connection object in the same manner as the refactored SectionFetcher and RecordHandler.