FSA Integration Partner

United States Department of Education

Federal Student Aid

Integrated Technical Architecture
FAFSA 7.0 and ED PIN
Performance Test Results

Version 1.0 DRAFT

December 31, 2002

Table of Contents

1 Introduction

The U.S. Department of Education's Office Federal Student Aid Programs (FSA) administers and operates the "Free Application for Federal Student Aid" (FAFSA). While available in paper form, FSA also provides this service through a web site. U.S. college students seeking student financial aid use the FAFSA program. During the academic year 2002-2003, over three million students used the web site to apply for federal financial aid. FSA anticipates that the number of users / applicants will increase by 50% during the 2003-2004 academic year, and will continue to rise in future years as the number of paper submissions decreases. This anticipated growth makes it imperative that FSA maximize the capacity and availability of the FAFSA web infrastructure while at the same time minimizing the amount of support FSA's representatives will have to provide for questions by students or difficulties with completing the form. Due to the planned increase in Internet application submissions, an intensive performance testing effort was planned and conducted, as summarized in this document.

FAFSA and PIN coexist in the same environment. PIN is used by Department of Education to associate students with their records. The number of applications that use PIN is growing. FSA anticipates the usage of the PIN application by FAFSA will increase by 50%. There are three major components for PIN: FAFSA PIN, HTML PIN and PIN Web Services. FAFSA PIN is the authentication functionality of FAFSA that allows users to securely access FAFSA business processes. PIN Web Services is authentication used only by the DLSS application. PIN HTML authentication is used by other FSA applications. All three components were performance tested to ensure that they could handle the FAFSA peak load.

2 Executive Summary

2.1 Current Test Results

The purpose of the performance test was to identify and resolve bottlenecks in the application, architecture, and infrastructure. The performance test also helped determine the optimal configuration for the production environment. In addition, the testing effort helped to determine capacity requirements and to prove that the application would be able to scale and handle the 2003 peak traffic.

All of the goals outlined in the performance test plan were met. There were several key bottlenecks that were identified and resolved, which improved the performance of the application.

2.2 Results Achieved

The performance problems and recommended fixes are categorized by component ranging from Web Server to Mainframe. The specific changes made are included in Section 3 of this document.

- FAFSA Application Code
- IBM HTTP Server - Web Server
- IBM WebSphere Application Server
- MQ
- Database Access
- Mainframe
- Infrastructure

In addition to these enhancements, the performance testing resulted in an optimized configuration of the environment.

2.3 Recommendations

The performance testing effort found areas for enhancements to the application, architecture, and infrastructure. Below is the high level list of items that could be evaluated in the future for potential performance improvements to the system:

- Serving HTML images in non-secure way instead of through SSL.
- Reducing session size to 2 ~ 4 KB.

The FAFSA application and environment should continue to be tested and tuned. In the future, tests should target an integrated system (possibly in production) with all of the components that make up FAFSA on the Web.

3 Performance Test - High Level Summary

Overall, twenty-three tests were executed from September through December. Each performance test cycle was focused on a specific area of the FAFSA Application. A series of infrastructure tests were run to identify issues with the network, hardware, Web servers, and Application servers. The business process scripts focused on the backend, including WAS, Oracle, DB2, Shadow Direct, MQ, and the Mainframe. The details for each run with conclusions are outlined in Section 4 of this document.

The performance problems and recommended fixes have been outlined below. The recommended performance enhancements are categorized by component ranging from Web Server to Mainframe.

FAFSA Application Code:

- FAFSA Application Code was updated to invalidate sessions. The code update was made to minimize memory usage and disk space in the database.
- The code was updated to resolve invalid time stamps. The invalid time stamps were causing application submissions to fail under heavy load.

IBM HTTP Server - Web Server:

- MaxSpareServers was changed from 100 to 10. The Spare Servers setting was used for handling load spikes. However, leaving a large number of web servers idle drains system resources. IBM recommended keeping this setting maxed at 10 spare servers.
- MinSpareServers was changed from 10 to 5. The change was also made for the same reason as MaxSpareServers.
- MaxRequestPerChild was changed from 10,000 to 0. This setting allows for an upper limit on the number of requests a particular child process can process before it terminates. By setting this to 0, we made sure that no individual process can potentially hold all system resources.

IBM WebSphere Application Server:

- Updated HP Kernel Parameters – HP Systems need maintenance on their kernel parameters as a result of these systems' interaction with MQ and WebSphere.
- HP Operating System patches were updated to bring the system up to IBM recommended levels.
- The version of WebSphere was updated to WAS to 3.5.6.
- Increased JVM Memory Size from 512Mb to 1Gb. This allowed us to better utilize the memory on each application server.
- Decreased number of clones per server from 4 to 2, which allowed us to better utilize the memory on each application server without creating problems with large number of clones.

Database Access:

- A dedicated session database was used for PIN business processes. For a high performance application, it is recommended by IBM, that it have its own session database.
- More space was allocated for the FAFSA session database so that the database will not fill up fast.

MQ:

- CPS region stopped responding to requests when bad data (reversed byte) was sent to the DPL Bridge (bridge between MQ and CICS) and requiring the CICS to be recycled. An IBM patch was applied to fix this issue. The patch will allow the DPL Bridge to handle bad data properly.
- WAS recycled unexpectedly during performance tests. It was determined that AMI, the MQ application interface, caused the recycle. AMI was removed and native java code was added to interface with MQI. This fixed the WAS recycling issue.
- MQ queue manager used 10 pairs of channels in its cluster to send a receive messages. Once a connection was made in the queue manager, the connection was not reused until the transaction was completed (sent or received). Connection pooling code was implemented to resolve this issue. This code reuses the connections made to the queue manager, which increased the speed of message transfer.
- No security measures were placed on MQ queues, which could result in one bad message shutting down the application. A security measure was put on MQ queues that controlled who may put messages from specific IP addresses.
- MQ had a parameter that requested operator acknowledgement to offload logs. Exceptions occurred when operator acknowledgement was not provided in time. In order to resolve this, Auto Ops updated the parameter so MQ did its offload automatically rather than requesting a response.
- During data transmission on the bridge between MQ and the mainframe, errors messages were occurring for transactions called CKB5. These errors would occur when pending transactions would timeout and appropriately produce transaction expired messages. Instead of handling the expiration messages, MQ would sustain errors. IBM provided a fix to resolve this issue.
- Messages were expiring on the bridge due to duplicated messages from MQ parameter, sync q. CSC & EAI ran a utility program that was provided by IBM to resolve this issue.

Mainframe:

- DB2 transactions timed out due to DB2 deadlocks. An IBM provided patch was applied to resolve DB2 deadlocks.
- A stress test for CICS and DB2 showed that the backend needed to be configured for the FAFSA peak. The following is the recommended DB2 configuration: set Max Task to 75, Thread Limit to 40, and TCB Limit to 50.

Performance Tests – Detailed Description

5 Capacity Planning

The capacity planning and performance test efforts work hand-in-hand. Actual peak data from last year was used to come up with targets for the number of concurrent users to test. Based on the results of the tests, the capacity and utilization requirements were estimated for the following year. Many of the calculations and estimates are based on an assumption that there will be a 50% increase from FAFSA 6.0 in utilization during peak 2003.

5.1 Number of Users

The following tables specify the number of expected users for FAFSA 7.0. Included are tables for the average daily number of users and hits, as well as the estimated peak number of users and hits, and several significant calculations based on the peak number of users estimated.

Assuming 50% growth from FAFSA 6.0

	FAFSA 7.0 Predict (17 hours)	FAFSA 7.0 Predict 4% Increase
Hits/day	105,000,000	
Hits Homepage/day	10,500,000	
Page Views/day	21,000,000	
Users/day	980,000	
User session length (min)	15	
Temp Apps/day	720,000	
App Submits/day	127,500	

FAFSA 7.0 Peak Hour Assumptions

Hits/hour	5,558,824	5,786,735
Hits Homepage/hour	555,882	578,674
Page Views/hour	1,111,765	1,157,347
Users/hour	51,882	54,010
User session length (min)	15	15

Column 1 -(90% of hits during 17 hour period 9am-2am, distributed equally)

FAFSA 7.0 Peak Hour Calculations

Hits/sec	1,544	1,607
Hits homepage/sec	154	161
Page views/sec	309	321
# concurrent users	12,971	13,502
Hits/user	107	107
Page views/user	21	21

Capacity Planning Estimates Table for FAFSA 6.0 and FAFSA 6.0 Actual (from the web trend reports)

	FAFSA 6.0 Predict (17 hours)	(24 hours)	FAFSA 6.0 Actual (17 hours)
Hits/day	70,000,000		61492052.5
Hits Homepage/day	7,000,000		
Page Views/day	14,000,000		11378108
Users/day	700,000		636204.00
User session length (min)	25		14
Temp Apps/day	700,000		480000
App Submits/day	70,000		85000
Hits/hour	3,705,882	4,417,000	4136891.5
Hits Homepage/hour	370,588	441,700	
Page Views/hour	741,176	883,400	
Users/hour	37,059	44,170	37423.76471
User session length (min)	25	25	14
			Column 3 represents Peak Hour of FAFSA 6.0 Actual 2/28/02 @ 21:00
Hits/sec	1,029	1,227	1149.136528
Hits homepage/sec	103	123	
Page views/sec	206	245	
# concurrent users	15,441	18,404	8733
Hits/user	100	100	
Page views/user	20	20	
user think time (sec)	75	75	

PIN Capacity Planning:

The following table shows the existing PIN database volume per hour (maximum):

Business Processes	Jan	Feb	Mar	Apr	May	Jun	July	Aug
Authentication	N / A	14882	7807	8254	9572	6949	8844	10140
Registration	N / A	2306	1688	1034	1182	973	1311	1477

After getting the projected volume from other applications for their future releases, the ITA team determined that the PIN should be performance tested with the following number of number of transactions per hour.
Registration: 3,459 transactions per hour (50% growth)
Authentication (FAFSA): 18,975 transactions per hour
Authentication (Other Applications): 3,189 transactions per hour
Authentication (Web Services): 1,000 transactions per hour

5.2 Extrapolations

The spreadsheet in diagram 2.1.1 is the method used to extrapolate the number of application servers needed for peak 2002 processing.

Diagram 2.1.1

Server Extrapolations	
Users Tested	3,000
Clones Used	4
Users per Clone	700
Projected # Peak Concurrent Users	13,500/hour
Clones Needed for Peak	16 for FAFSA 3 for PIN
Projected clones per server (8 x 750)	2
# WAS Servers for peak (20% contingency)	10
# IHS Servers for peak (20% contingency)	10

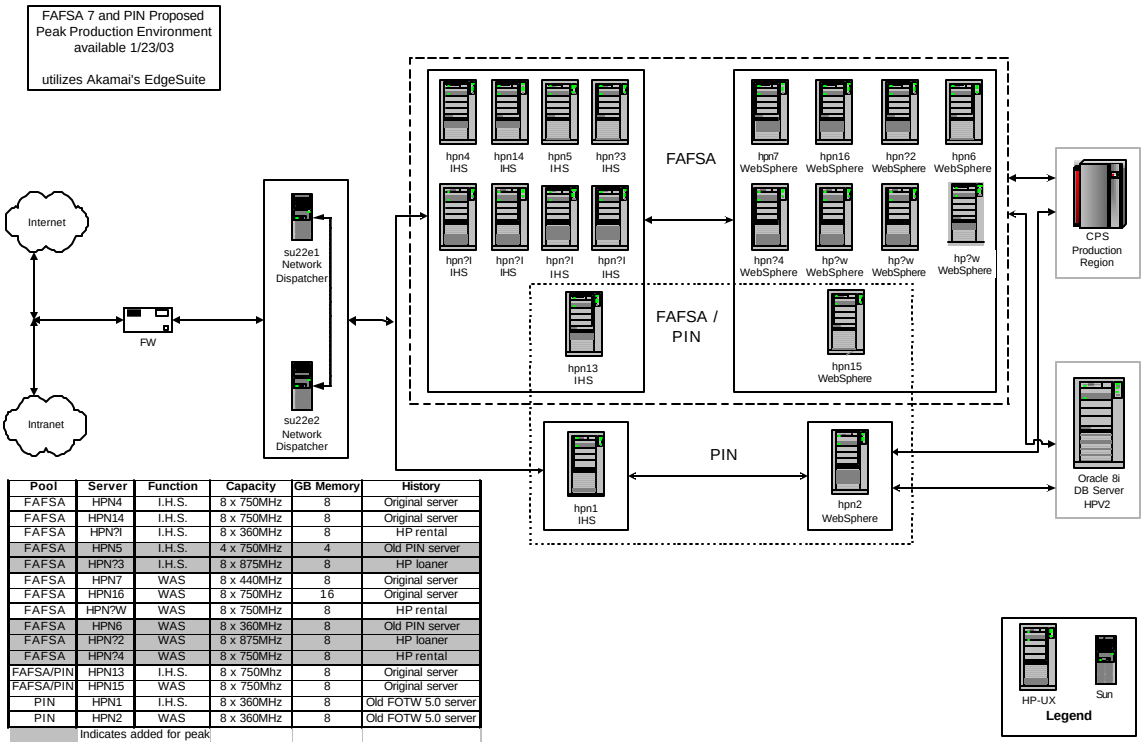
6 Environment Recommendations

Based on the performance testing and tuning, the following configuration should be able to handle the FAFSA peak load with 13,500 concurrent users.

- Web Servers: CPU: 36,000 Mhz, Memory: 40 GB Total
- Application Servers: CPU: 75,000 Mhz, Memory: 80 GB Total

Diagram 2.1.2 illustrates the projected production logical technical architecture requirements.

Diagram 2.1.2



Pool	Server	Function	Capacity	GB Memory	History
FAFSA	HPN4	I.H.S.	8 x 750MHz	8	Original server
FAFSA	HPN14	I.H.S.	8 x 750MHz	8	Original server
FAFSA	HPN?I	I.H.S.	8 x 360MHz	8	HP rental
FAFSA	HPN5	I.H.S.	4 x 750MHz	4	Old PIN server
FAFSA	HPN?3	I.H.S.	8 x 875MHz	8	HP loaner
FAFSA	HPN7	WAS	8 x 440MHz	8	Original server
FAFSA	HPN16	WAS	8 x 750MHz	16	Original server
FAFSA	HPN?W	WAS	8 x 750MHz	8	HP rental
FAFSA	HPN6	WAS	8 x 360MHz	8	Old PIN server
FAFSA	HPN?2	WAS	8 x 875MHz	8	HP loaner
FAFSA	HPN?4	WAS	8 x 750MHz	8	HP rental
FAFSA/PIN	HPN13	I.H.S.	8 x 750Mhz	8	Original server
FAFSA/PIN	HPN15	WAS	8 x 750Mhz	8	Original server
PIN	HPN1	I.H.S.	8 x 360MHz	8	Old FOTW 5.0 server
PIN	HPN2	WAS	8 x 360MHz	8	Old FOTW 5.0 server
	Indicates added for peak				

7 Conclusions

7.1 Ongoing Performance Testing

Performance tuning is an on-going process. A system that has been tuned does not guarantee that the tuning enhancements will always be suitable for a growing system. Everyday occurrences such as changes in usage patterns, data volumes and distribution, and applying patches and fixes can make a marked impact on the overall performance of a system. It is important to have a production performance strategy in place. The strategy should include proactive performance monitoring and capacity planning.