SFA Modernization Partner

United States Department of Education

Student Financial Assistance

FAFSA on the Web 6.0
Performance Test Plan

DRAFT

September 15, 2001

Table of Contents

1 Executive Summary

1.1. Introduction

The U.S. Department of Education's Office of Student Financial Assistance Programs (SFA) administers and operates the "Free Application for Federal Student Aid" (FAFSA). While available in paper form, SFA also provides this service through a web site. U.S. college students seeking student financial aid use the FAFSA program. During the academic year 2001-2002, over three million students used the web site to apply for federal financial aid. SFA anticipates that the number of users / applicants will double during the 2002-2003 academic year, and will continue to rise in future years as the number of paper submissions decreases. This anticipated growth makes it imperative that SFA maximize the capacity and availability of the FAFSA web infrastructure while at the same time minimizing the amount of support SFA's representatives will have to provide for questions by students or difficulties with completing the form. The web FAFSA product is commonly referred to as FAFSA on the Web and incorporates all requirements related to the paper FAFSA for each school year.

The primary objectives for redesigning FAFSA on the Web are to leverage industry best practices to improve usability and accessibility by customers, performance of the web application during peak periods of FAFSA submissions, and create the foundation for efficient enhancements, as necessary.

1.2. Background

Each year, we are seeing a 100% increase in users and applications submitted. As the number of FAFSA on the Web users increases, the FAFSA on the Web application has to scale to be able to handle the increased capacity. Our N-Tiered architecture is designed to scale vertically and horizontally. It is the responsibility of the application team to validate that this architecture will be able to scale and handle growth in peak periods. Our performance test will not only verify the scalability, but also give us an update to our estimated capacity for peak 2002.

1.3. Objective

The purpose of our performance test is to:

1. Validate the n-tier architecture for FAFSA 6.0.
2. Verify that the application will scale.
3. Verify the performance capability of FAFSA 6.0 relative to users.

1.4. Expected Results

At the conclusion of each test cycle a test report will be prepared with the results of the test cycle and confirmation that our objectives were met. Each subsequent test cycle should get us closer to our overall goal of an optimized application and architecture performance.

2 Overview

2.1. Overview

This document provides the process and details on the performance goals that will be used through the FAFSA 6.0 test effort.

Because of the new performance test environment, web infrastructure, and application, there is no "true" baseline history to set specific application performance goals. For the FAFSA 6.0 performance tests, there will be a four performance tests runs.

This document will be utilized to communicate test plans prior to each cycle execution with all key participants as outlined in Section 3.

2.2. Overall Goals

Performance Test, Active Test runs will be executed with the following goals in mind:

1. Tune application so it is at optimal performance at the conclusion of each test cycle.
    - Run full set of scripts to performance test key functionality
2. Tune hardware / environment so it is at optimal performance at the conclusion of each test cycle.
    - eNetwork dispatcher is correctly balancing the load to the web servers
    - Web server load manager is correctly balancing load to application servers.
    - Right number of connections to database (connections to Shadow Direct and Threads from Shadow Direct to DB2).
    - Tune Shadow Direct – verifying that the timeouts are correctly set to work with connection pooling.
    - Tune each oracle database for optimal performance of FAFSA.
    - Review the performance of the oracle connections and verify that the listener(s) can handle load. Look into MTS, if needed.
    - Web, application, and database queues configured correctly
3. Ensure that the current network infrastructure and 100Mb LAN can handle peak with new application requirements.
4. Validate performance gains of adding an SSL accelerator card
5. Validate performance gains of having caching enabled.
6. Determine saturation point and buckle zone for each test run in order provide necessary bottleneck data.
7. Use results to update, if necessary, the production hardware requirements for FAFSA 6.0.

3 Performance Test Process

This section describes the Performance Test Process.

3.1. Performance Test Process

To achieve the overall goals, establish a FAFSA 6.0 baseline, and set specific goals for tests, a process is required. Below explains the steps that will be taken through the performance test effort:

3.1.1. Establish a list of performance test areas that will be monitored.

Performance test areas include a wide range of items such as user response times, throughput, and the various servers. These areas are established before the test so monitoring points can be derived, and teams have time to schedule the right people to monitor the tests. Specific monitoring points will be analyzed to identify issues such as bottlenecks and will also provide the necessary data for analysis and response times. Table 8.1 includes all the performance test areas that need to be monitored and the responsible party for each.

3.1.2. Establish the data and parameters that need to be collected.

Data will need to be collected for analysis during and after each test cycle. Table 8.1 outlines the type of data that should be monitored and collected and the responsible party for each monitoring point. Section 4.2 outlines the data analysis deliverables that CSC and Mercury Interactive are to submit at the conclusion of each test cycle.

3.1.3. Planning for the first FAFSA 6.0 Performance Test.

The first FAFSA 6.0 will become the baseline for set of tests that will be run. The goal of the first test is to take the necessary steps to ensure that a proper baseline is created, which will aid in setting more detailed goals for forthcoming tests. The current parameters/configuration settings on the hardware will be recorded before the first test is run. This will enable future tracking of the changes that will be made throughout the tests.

3.1.4. FAFSA 6.0 Active test cycles monitored and data collected.

Mercury Interactive will be running the pre-scheduled performance tests. Mercury Interactive, CSC, ITA team, and the performance test team will be monitoring the test and collecting data. This data will be used to determine what modifications need to be made to the environment and application code.

3.1.5. Fixes and Changes made to the environment and application code.

Based on the analysis and recommendations of the team, configuration changes may need to be made after each test run. These changes should provide higher performance results in the next test run.

3.1.6. Specific Goals set for the next FAFSA 6.0 test run.

Specific goals for each test cycle are outlined in Section 6. After data has been collected, and the required fixes made, there will be the opportunity to modify specific goals for the next test run.

3.1.7. Results are documented and FAFSA 6.0 environment is validated.

Final deliverable is assembled with information and data from each of the test runs. The data is used to estimate any areas that may need close attention in the future. Additionally, the data will be used for capacity planning and establishing a FAFSA 6.0 production environment.

4 Roles, Responsibilities, and Deliverables

This section outlines the roles and responsibilities of all parties involved in FAFSA 6.0 Performance testing.

4.1. Performance Test Roles and Responsibilities

The following list outlines the roles and responsibilities of the FAFSA 6.0 Capacity Planning team.

Role	Assigned	Phone	Responsibility
Performance Test Lead	Amy Schafer	847-714-2813	Define the detailed goals for each test cycle / test types – stress, stability, ssl, cache, db, mainframe. Analyze the application functionality to plan the right mix of tests (submit, renew, correct). Define the expected concurrent users per server at optimized configuration. Work with CSC during ActiveTest to measure the performance of each component (CPU, Mem, IO, Network). Work with Mercury Interactive to plan the length of each run and starting / stopping points. Work with IBM SME to fix bugs and bottlenecks identified during performance test. Work with Mercury Interactive and IBM SME to assess scalability. Use the test results to update the expected physical and logical configuration at peak.
Performance Test SME Team	Accenture, Snap Technologies	TBD	Define the detailed goals for each test cycle / test types – stress, stability, ssl, cache, db, mainframe. Analyze the application functionality to plan the right mix of tests (submit, renew, correct). Define the expected concurrent users per server at optimized configuration. Define the expected transactions per minute at optimized configuration. Define the expected physical architecture based on similar architectures and benchmarks. Use the test results to update the expected physical and logical configuration at peak.
Performance Test Resource	Tien Le	703-947-2802	Verify Existing Test Scripts (Excel Format). Create New Test Scripts (Excel Format). Coordinate recording and testing of scripts with Mercury Interactive.

			Coordinate test dates with all testing resources (Mercury Interactive, ITA, NCS, CSC, Mod Partner). Work with all teams to verify administrator contact information and attendance. Set up pre-test and test day conference calls. Document Executive Summary of test cycle results.
Mercury Interactive Active Test Manager	Dan Rice	303-301-2616	Record test scripts from excel format scripts. Verify all test scripts and perform calibration test. Lead the Active Test effort
Tech Arch Manager	Jose J. Alvarez	847-714-2813	Coordinate with all teams and resources to ensure that the capacity planning, performance test planning and performance test execution are completed on time.
Application Contact	Matt Kain	319-354-9200 x6852	Responsible for application changes and recommendations. Monitor application performance at each test cycle.
Websphere Administrator	Roshani Bhatt /Bob Wehrle	202-651-3787 202-651-3798	WAS configuration changes and updates. Monitor key areas and deliver summary results.
IHS Administrator	Roshani Bhatt / Bob Wehrle	202-651-3787 202-651-3798	IHS configuration changes and updates. Monitor key areas and deliver summary results.
eNetwork Dispatcher Administrator	Roshani Bhatt / Bob Wehrle	202-651-3787 202-651-3798	Load Balancer configuration changes and updates. Monitor key areas and deliver summary results.
System Adminstrator	Malcolm Waltz / Dave Murdy /Fred Giannetto	203-31704983 203-317-4818 860-513-2305	Monitor hardware to ensure optimal configuration. Monitor CPU, Mem, IO, capacity levels. Record hardware capacity levels at different intervals throughout the test cycle.
Oracle Administrator	Rich Ryan	860-701-1209	Monitor Oracle Database to ensure the optimal configuration. Monitor the Oracle Database performance throughout the test cycle. Make recommendations on ways to improve Oracle performance.
DB2 Administrator	Rich Mincher	817-762-8554	Monitor DB2 Database to ensure the optimal configuration. Monitor the DB2 Database performance throughout the test cycle. Make recommendations on ways to improve DB2 performance.
Mainframe Administrator	Tom Puddicombe / Nancy Mathisen	203-317-5839 817-762-8061	Monitor Mainframe to ensure optimal configuration. Monitor capacity levels at different intervals throughout the test cycle. Monitor Shadow Direct and the communication between App server and DB.
Network Administrator	CSC - TBD		Monitor Network to ensure optimal configuration. Monitor capacity levels at different intervals throughout the test cycle.

4.2. Deliverables

The following deliverables will be submitted to the FAFSA project team within the dates specified. Table 8.1 provides a guide as to the data that needs to be captured and presented to the team at the conclusion of each test cycle.

4.2.1. Mercury Interactive Analysis

Mercury Interactive will deliver a document that will summarize the test, problematic areas, and recommendations. Mercury Interactive is expected to deliver this analysis within 3 days after each test cycle is run.

4.2.2. Mercury Interactive Final Recommendation

Mercury Interactive will deliver final deliverable with an analysis of each test cycle, a summary of the results, and a general recommendation on the performance of the FAFSA application.

4.2.3. CSC Capacity Analysis

CSC will complete a spreadsheet, which captures information on current capacity percentages as well as information from the 2001 peak period. This information will be used to better plan the performance tests and for capacity planning at the conclusion of all the test cycles.

4.2.4. CSC Performance Test Data

The Performance Monitoring Areas, Table 8.1, provides a guide as to the type of data that will need to be collected. CSC should decide on the best format for providing this data. The data should be collected at a 5 minute sampling points, unless otherwise suggested during the test. CSC is expected to deliver this data within 3 days after each test cycle is run.

4.2.5. CSC Performance Test Summary Analysis

At the conclusion of the performance tests, CSC will provide a document containing general comments, summary of capacity issues, and effective capacity.

4.2.6. ITA Parameter Configuration Recommendations

The ITA team will deliver a document containing the performance test environment configuration prior to the start of the first test cycle. The ITA team will deliver their recommendations and changes made during the test. The ITA Team will also be expected to monitor and document the areas outlined in Table 8.1. The ITA team is expected to deliver a document summarizing their recommendation within 3 days after each test cycle is run.

5 FAFSA 6.0 Business Processes

These are the business processes that we will be targeting in our performance testing. These business processes will drive the creation of our scripts for the performance testing (Active Test) cycles.

5.1. Business Processes

The following list outlines the business processes:

1. Fill out a FAFSA– Form to fill out and submit a FAFSA
2. FAFSA Renewal – Renew an application
3. FAFSA Corrections – Correct an existing application
4. Student Access
5. Request Application Status (appstat) – Check application status
6. Electronic Signature – Electronic Signature for an application
7. Request Duplicate SAR – Request another copy of the SAR
8. School Code Search – Look up college
9. Stand Alone Worksheets – Worksheets to aid with application completion

5.2. Business Processes Table

The following table outlines the architecture components that make up the FAFSA Application. These components are web, application, PIN, and Database servers. Each business process test scripts will test the outlined architecture components.

	Web Server	Application Server	Oracle DB	Pin DB	CPS DB2
Fill out a FAFSA	X	X	X	X	X
Renewal	X	X		X	X
Corrections	X	X		X	X
Student Access	X	X			X
Request Application Status	X	X			X
Electronic Signature	X	X		X	
Request Duplicate SAR	X	X			X
College Lookup	X	X			X
Standalone Worksheets	X	X			

6 Detailed Goals per Cycle

This section will include detailed goals for each test cycle.

6.1. Active Test Cycles

6.1.1. Active Test Cycle 1 – Scheduled 9/27/2001

Detailed Goals for test cycle one:

- Target at least 500 Concurrent Users
- Stress Test Index Pages (Non-SSL)
- Stress Test – Index Page (SSL with no Encryption Accelerator)
- Fill out FAFSA Business Process Script

Exit Criteria for test cycle one:

- Test at least 500 Concurrent Users
- Complete Stress test of non-SSL
- Complete Stress test of SSL
- Complete a Fill out FAFSA Business Process Script

6.1.2. Active Test Cycle 2 – Scheduled 10/11/2001

Detailed Goals for test cycle two

- Target at least 1000 Concurrent Users
- Stress test Index Pages (Non-SSL)
- Stress Test – Index Page (SSL)
- First 4 Business Processes Script

Exit Criteria for test cycle two:

- Test at least 1000 Concurrent Users
- Complete Stress test of non-SSL
- Complete Stress of SSL
- Complete 4 Business Process Script

6.1.3. Active Test Cycle 3 – Scheduled 11/1/2001

Detailed Goals for test cycle three

- Target at least 1500 Concurrent Users
- Stress test Index Pages (Non-SSL)

- Stress Test – Index Page – SSL (with SSL Encryption Card)
- First 4 Business Processes Script

Exit Criteria for test cycle one:

- Test at least 1500 Concurrent Users
- Complete Stress test of non-SSL
- Complete Stress of SSL
- Complete 4 Business Process Script

6.1.4. Active Test Cycle 4 – Scheduled 11/15/2001

Detailed Goals for test cycle four

- Target at least 2000 Concurrent Users
- Stress test Index Pages (Non-SSL)
- Stress Test – Index Page – SSL (with SSL Encryption Card)
- First 4 Business Processes Script

Exit Criteria for test cycle one:

- Test at least 2000 Concurrent Users
- Complete Stress test of non-SSL
- Complete Stress test of SSL
- Complete 4 Business Process Script

6.1.5. Active Test Cycle 5 – Optional on 11/29/2001

Detailed Goals for test cycle four

- Target at least 3000 Concurrent Users
- Stress test Index Pages (Non-SSL)
- Stress Test – Index Page – SSL (with SSL Encryption Card)
- Full Application

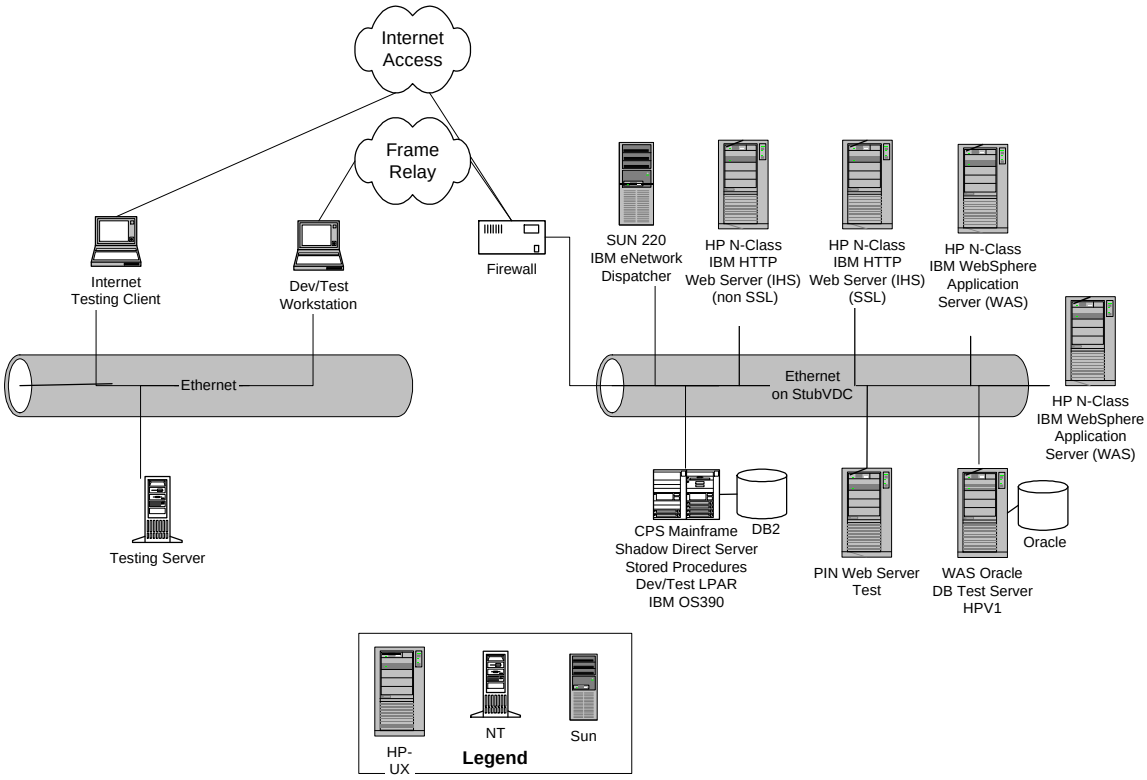
Exit Criteria for test cycle one:

- Test at least 3000 Concurrent Users
- Complete Stress test of SSL
- Complete Stress of non-SSL
- Full Application

7 Technical Infrastructure

7.1. FAFSA 6.0 Performance Test Environment

FAFSA 6.0 Performance Test Environment Diagram



7.1.1. Performance Test Environment Requirements

- This environment mirrors the proposed production environment.
- Oracle Test Server should be configured to handle peak session usage.
- Mainframe test LPAR will be configured exactly like Production – DB and Shadow Direct tweaking required.
- Network Bandwidth will handle at least 2000 concurrent users.
- eNetwork Dispatcher Load Balancing will handle at least 2000 concurrent users.

- Testing IHS Workflow server management

7.1.2. Assumptions

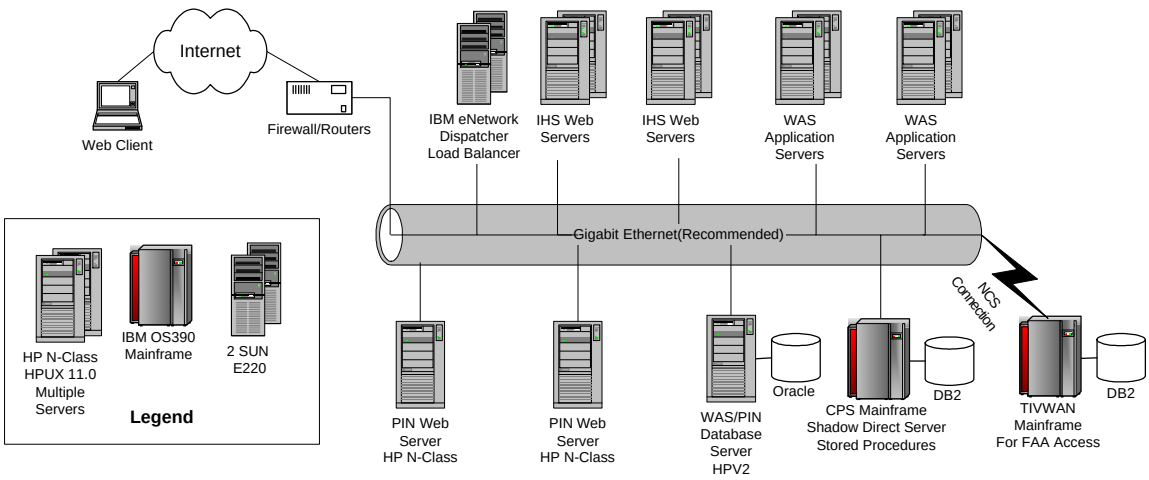
- Testing 2000 concurrent users
- Testing SSL and Non SSL connections
- Network Dispatcher: no proxy caching
- Web Server (non-SSL): disk, network intensive
- Web Server (SSL): CPU intensive
- App Server: CPU and memory intensive
- HP-N Class servers are all 750Mhz

7.1.3. Configurations

- Network Dispatcher: Sun E220 / 1 CPU / 1 GB RAM
- Web Server (non-SSL): HP N-Class 8 Way CPU – 750Mhz
- Web Server (SSL): HP N-Class 8 Way CPU – 750Mhz
- App Server: HP N-Class 8 way CPU – 750Mhz

7.2. FAFSA 6.0 Production Environment

FAFSA 6.0 Production Environment Diagram



8 Performance Monitoring

Performance monitoring detailed information

8.1. Performance Monitoring Table

The table lists each performance area, its associated performance monitoring points, a general description of the performance area, and the group responsible to monitor the area. This document will be supplied to those monitoring the test. It will ensure that all required areas will be monitored.

Performance Areas	Performance Monitoring Point	Description	Monitored By:
Run Time	• Total memory available for the JVM • Amount of free memory for the JVM	Application server - Memory used by a process as reported by the JVM.	ITA
Database connection pools	• Average size of the connection pool (number of connections), • Average number of threads waiting for a connection • Average wait time in milliseconds for a connection to be granted • Average time the connection was in use	Application server - Reports usage information about connection pools for a database.	ITA
Servlet engines	• Average number of concurrent requests for a servlet • Amount of time it takes for a servlet to perform a request • Average number of concurrently active HTTP sessions.	Application server - Reports usage information for Web applications, servlets, Java Server Pages (JSPs), and HTTP sessions.	ITA
Network Dispatcher	• Completed number of connections per web server • Number of connection errors per web server • Active number of connections to the web servers	eNetwork Dispatcher is an IBM load-balancing tool. It balances http requests between web servers.	ITA

CPU utilization	• Network Dispatcher Server (SU22E1) • 2 web servers (?) • 2 application servers (?) • Database server (HPV1) • HPV1 databases (WAS35STG, FAFSASTG, SESSSTG) • PIN test web server • CPS Mainframe DB2 database server	Depicts the utilization of the CPU. High CPU utilization can be an indicator of a CPU bottleneck. CPU bottlenecks may occur when Global CPU utilization exceeds 75%. However, some workloads can operate with very high CPU utilization with the CPU becoming a bottleneck.	CSC
Memory utilization	• Network Dispatcher Server (SU22E1) • 2 web servers (?) • 2 application servers (?) • Database server (HPV1) • HPV1 databases (WAS35STG, FAFSASTG, SESSSTG) • PIN test web server • CPS Mainframe DB2 database server	Indicates the utilization of memory. If memory utilization is below 95%, it is not a bottleneck. Memory can reach 100% utilization without necessarily being a bottleneck. The combination of high memory utilization and Virtual Memory reads & writes indicates that memory is a bottleneck.	CSC
Disk I/O	• Network Dispatcher Server (SU22E1) • 2 web servers (?) • 2 application servers (?) • Database server (HPV1) • HPV1 databases (WAS35STG, FAFSASTG, SESSSTG) • PIN test web server • CPS Mainframe DB2 database server	Illustrates the percentage of time that a disk I/O is pending on a disk device. High disk utilization can be an indicator of a disk bottleneck. Disk utilization greater than 50% may indicate a disk bottleneck. The service times, not charted, will determine if the I/O subsystem is performing poorly.	CSC
Network Utilization	• Connections from firewall to eNetwork Dispatcher • ENetwork Dispatcher to Web Servers • Web servers to Application servers • App Servers to Oracle database servers • App servers to CPS mainframe	The bandwidth required to support an application. It can refer to the application traffic in and out of a data center, or between servers.	CSC

Throughput	• Bytes per second	Volume processed in a specified period by the system or system component	Mercury Interactive
Hits Per Second	• Hits per second	The number of hits per second on the application.	Mercury Interactive
User Response Times	• Average, minimum, and maximum times for SSL pages • Average, minimum, and maximum times for non-SSL pages	Elapsed time between two events measured at specific points	Mercury Interactive
Servlet Performance	• Servlet and JSP performance	The performance of each of the servlets and JSPs as they are accessed by the test script.	Mod Partner

9 Capacity Estimates

Capacity Planning Estimates for 2002- 2003 peak period.

9.1. Capacity Planning Estimates Table

Capacity Planning Estimates Table

FAFSA 6.0 Daily Assumptions

(WebTrends, page views are 20% of total hits)

Hits/day	70,000,000
Hits Homepage/day	7,000,000
Page Views/day	14,000,000
Users/day	700,000
User session length (min)	25
Temp Apps/day	700,000
App Submits/day	70,000

FAFSA 6.0 Peak Hour Assumptions	**9am-2am (Ave)**	**Peak Hour**
Hits/hour	3,705,882	4,417,000
Hits Homepage/hour	370,588	441,700
Page Views/hour	741,176	883,400
Users/hour	37,059	44,170
User session length (min)	25	25

FAFSA 6.0 Peak Hour Calculations	**9am-2am (Ave)**	**Peak Hour**
Hits/sec	1,029	1,227
Hits homepage/sec	103	123
Page views/sec	206	245
# concurrent users	15,441	18,404
Hits/user	100	100
Page views/user	20	20
user think time (sec)	75	75
# concurrent active users	1,541	1840