

Test Delivery System Performance Improvements

Final Release Report

Prepared for: Smarter Balanced Assessment Consortium
May 20, 2016



Fairway Technologies, Inc.

Table of Contents

Executive Summary	1
Performance Improvements	2
Overview	2
Results Summary	3
Load Testing	5
Criteria	5
Additional Details	6
AWS Environment	6

Executive Summary

Smarter Balanced and Fairway Technologies are pleased to release an update to the SmarterApp test delivery system that includes significant increases in performance and load capacity.

Performance tests used cloud servers hosted on Amazon Web Services. The primary key metric is the time it takes for a student to start a test from the moment they select an assessment until the first set of test questions are displayed. Under a modest load of 20,000 concurrent students, the unenhanced software required 9.3 seconds to present the test. The enhancements reduce this time to 0.7 seconds under identical load, hardware, and configuration.

Another important metric is the number of students that can concurrently login and take a test. According to Fairway's tests, while the unenhanced software could provide tests for up to 20,000 concurrent students with over 99% student completing the test with no errors, this new release allows for up to 50,000 total concurrent students, a 150% increase.

Note: The unenhanced version was retrieved from the BitBucket source code repository and compiled on 12/14/2015.

Performance Improvements

Overview

Fairway started its first round of performance improvements by running profilers and load tests to identify the main bottleneck of the current open source code. The Fairway team found that the TDS MySQL database interaction was the primary area of contention. By viewing the CPU usage, connections, writes/sec and reads/sec, Fairway determined the heaviest workload was produced during student login and the starting of a test.

After reviewing the performance metrics, the development team reviewed the code and profiled the API calls made during the login and start test processes to determine what areas of the code should be considered as opportunities for performance improvements. Fairway discovered an excessive number of database queries are made during these processes, resulting in an increased load on the database.

Improvements for this release have focused on reducing the number of database calls by:

1. Combining multiple queries into a single query
 - a. Several methods query the database to get data that is then utilized in the next query. Where possible, these queries were combined using join operators in a single query.
2. Using domain objects to pass data between methods
 - a. Many methods in the application do not accept enough data to accomplish their task. These methods rely on helper methods that query additional information from the database. Fairway eliminated these additional queries by passing domain objects with the required data.
3. Adding caching
 - a. Fairway eliminated a significant number of queries for data that do not change often by adding caching. This includes system configuration and test metadata.
4. Replacing temporary tables with data structures
 - a. A significant number of database queries were eliminated by replacing temporary tables with application data structures. Although the temporary tables were small each operation on a temporary table required another round trip to the database.
5. Focus on improving the proctor polling logic
 - a. The proctor application polls every 30 seconds to get updates on the student progress and check for actions that a proctor needs to perform. By using the techniques listed above we were able to decrease the number of database calls made by 35 per minute.

Fairway has developed a suite of regression tests to cover the features present in the test delivery system. The final regression testing suite will be delivered in this May 2016 release. Fairway has utilized the regression tests, along with newly developed integration tests and a manual QA process, to ensure the code enhancements have not broken existing functionality or introduced new bugs. The code being delivered has passed all regression tests that are in place.

Results Summary

Fairway ran multiple load tests with the unenhanced open source code, compiled from source on 12/14/2015, along with both the enhanced version from the interim release (Jan 2016) and the final release version (May 2016). The application was tested with 20,000 concurrent students (20K), 35,000 concurrent students (35K) and 50,000 concurrent students (50K).

Two separate runs were executed at 20,000 and 35,000 load tests with different database storage types - one run with general purpose SSD, and the second run with provisioned IOPS, a more performant, albeit more expensive storage type option that Amazon Web Services offers. The 50k test run was executed using a Provisioned IOPS database storage type 30,000 IOPS.

General Test Data

	Unenhanced* 20K	Interim 20K	Release** 20K	Unenhanced* 35K	Interim 35K	Release** 35K	Release** 50K
Concurrent Students	20,000	20,000	20,000	35,000	35,000	35,000	50,000
Student/Proctor Ratio	20/1	20/1	20/1	20/1	20/1	20/1	20/1 (S) 10/1 (I)
# of Proctors	1,000	1,000	1,000	1,750	1,750	1,750	2,500 (S) 5,000(I)
% Proctors Start Session	100%	100%	100%	100%	100%	100%	100%
% Students Complete Test	98.6%	100.0%	100.0%	40.5%	99.9%	100%	99.9%

Database Info

	Unenhanced*	Interim	Release**	Unenhanced*	Interim	Release**	Release**
	20K	20K	20K	35K	35K	35K	50K
Max database CPU	51.0%	17.0%	13.4% (S) 14.3% (I)	41.0%	34.0%	27.1% (S) 23% (I)	14% (S) 13.3% (I)
Max writes per second	1,926	1,509	1,542 (S) 2,602 (I)	1,812	2,280	2,020 (S) 2,723 (I)	2,212 (S) 3,092 (I)
Max reads per second	159	129	173/239	64	272	181 (S) 242 (I)	339 (S) 412 (I)
RDS Instance Type (m4)	4xlarge	4xlarge	4xlarge	4xlarge	4xlarge	4xlarge	10xlarge

Important HTTP Requests (time in milliseconds)

	Unenhanced*	Interim	Release**	Unenhanced*	Interim	Release**	Release**
	20K	20K	20K	35K	35K	35K	50K
Start Test	5,682	2,808	463 (S) 131 (I)	38,961	5,395	4,352 (S) 230 (I)	235 (S) 326 (I)
Load TestShell.aspx	215	146	32 (S) 30 (I)	9,233	415	209 (S) 35 (I)	31 (S) 31 (I)
Fetch first 2 pages	3,383	1,915	243 (S) 119 (I)	29,323	5,365	1,274 (S) 221 (I)	152 (S) 112 (I)
Get Page Content	67	65	15 (S) 14 (I)	2,303	128	50 (S) 15 (I)	14 (S) 14 (I)
Total Test Start Time	9,347	4,934	774 (S) 303 (I)	113,430	11,303	4,352 (S) 509 (I)	441 (S) 486 (I)
% Change	-	-47%	-91.7%/-	-	-90%	-96.1%/-	-

* Source code retrieved and compiled from BitBucket repositories on 12/14/2015

** If a single value is shown, value corresponds to both IOPS and SSD database types. Otherwise, results displayed in the form of [SSD]/[IOPS] to differentiate between database types.

*** Please note that 50k test runs were executed using a larger AWS RDS Database instance (m4.10xlarge), direct comparisons to other metrics should consider this difference in hardware.

Load Testing

Since Fairway was not provided the load testing plans used by AIR and Measured Progress, a new load test suite was developed using Apache JMeter. Without any knowledge of the previous plans used, Fairway cannot provide a direct comparison against results obtained by AIR and/or Measured Progress. Instead, Fairway is providing new performance baselines based on the unenhanced open source code retrieved from the BitBucket repositories on 12/14/2015 as well as Fairway's previous interim release (Jan 2016).

The load test plan was created by recording all of the HTTP calls created when a proctor logs in and approves students, as well as the student logging in taking the test. The test includes all API calls to the server in addition to calls made from the browser requesting resources such as CSS files, JavaScript files, images, audio and video resources, etc. Since all of these resources are served from the TDS application it is important to send that load to the server to make the simulation as realistic as possible.

The final round of testing included both fixed form and computer adaptive tests (CAT) assessments, which are both lengthier and are also computationally more complex.

Criteria

From the original RFP, the Client Performance Requirements states “It should take less than 15 seconds to start a new testing session.” There are multiple requests that contribute to this overall time. After the student clicks the final button to start the test, the following HTTP calls are made before the first question is loaded:

- Start Test (/student/Pages/API/MasterShell.axd/startTest)
- Load TestShell with resources (/student/Pages/TestShell.aspx)
- Fetch first 2 question meta-data (/student/Pages/API/TestShell.axd/updateResponses)
- Get page content (/student/Pages/API/TestShell.axd/getPageContent)

In calculating the unenhanced baseline for concurrent students and the final number of concurrent students reached, Fairway evaluated the combined averages of these calls and required them to be below 15 seconds in order to qualify.

Additional Details

AWS Environment

The following environment was configured for use across all load tests performed.

Component	AWS Instance	JVM Settings	Notes
ART Database	c4.2xlarge 8 vCPU, 15 GB RAM		
ART Web Server	c4.2xlarge (load balanced) 8 vCPU, 15 GB RAM	-XX:+UseConcMarkSweepGC\ -Xms2048m -Xmx10240m\ -XX:PermSize=1024m\ -XX:MaxPermSize=2048m\ 	
OpenAM	m4.xlarge (x1 load balanced) 4 vCPU, 16 GB RAM		
OpenDJ	c4.8xlarge 36 vCPU, 60 GB RAM		
Permissions Database	m3.medium 1 vCPU, 3.75 GB RAM		
Permissions Web Server	m3.xlarge 4 vCPU, 15 GB RAM	-XX:+UseConcMarkSweepGC\ -Xms1024m -Xmx10240m\ -XX:PermSize=512m\ -XX:MaxPermSize=2048m\ 	
Program Management Database	m3.medium 1 vCPU, 3.75 GB RAM		
Program Management Webserver	m3.medium 1 vCPU, 3.75 GB RAM	-XX:+UseConcMarkSweepGC\ -Xms512m -Xmx2048m\ -XX:PermSize=512m\ -XX:MaxPermSize=1512m\ 	

TDS Webservice	c4.4xlarge (x3 load balanced) 16 vCPU, 30 GB RAM	-XX:+UseConcMarkSweepGC\ -Xms5120m -Xmx28672m\ -XX:PermSize=512m\ -XX:MaxPermSize=2048m\
Test Spec Bank	m3.medium 1 vCPU, 3.75 GB RAM	
TDS Database	db.m4.4xlarge (20k, 35k runs) 16 vCPI, 32 GB RAM db.m4.10xlarge(50k runs) 40 vCPU, 160 GB RAM	RDS Instance with ~200K tests taken across 3 types of tests.