**ENFA US Migration – Performance Testing Approach**

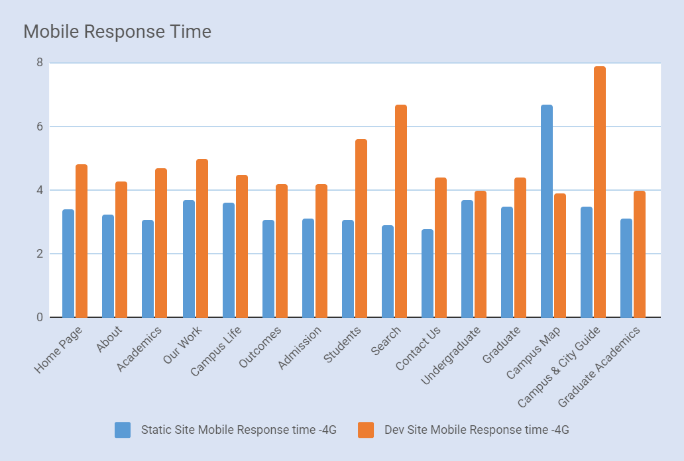
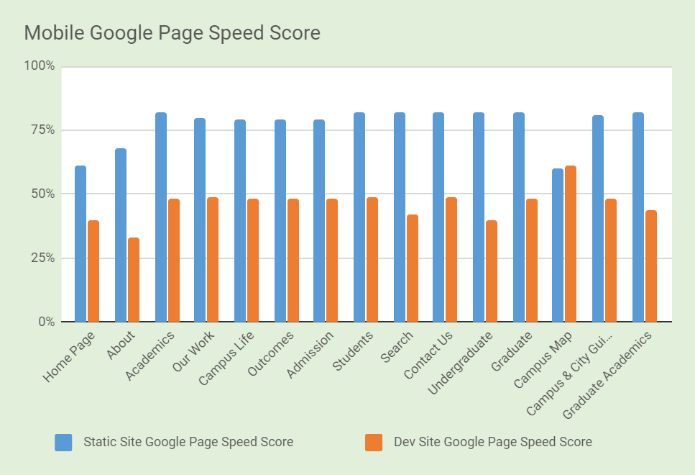
**Objective:**

* To measure the end user performance capabilities of the application in mobile and desktop
* Evaluate the business transactions (end-user) conducted within the application

**Tools Identified for Client-Side Metrics:**

|  |  |
| --- | --- |
| **Tools** | **Features** |
| GT Metrix | Google Grade & Recommendations |
| Page Speed Insights | Google Page Speed Score |
| Webpage Test | Page Size and No of Requests |
| Light House | Automated tool for improving the quality of web pages |
|  |  |

**Sample Report:**

**Performance Metrics**

We must monitor two different metrics are client side and server side

Client side:

* Average response time
* Transactions per second
* Hits per second
* Throughput in bytes
* Success/Failed statistics
* Error statistics

Server-side metrics:

* CPU Utilization
* Memory Utilization
* Application – Requests Count, CPU Usage, Memory Usage
* Database – CPU, Memory, MySQL slow queries count

**Client-Side Performance testing:**

In today's digital era, gaining the 100% customer experience (CX) on the application most difficult one. In-order achieve effective customer experience we need to concentrate on the three major area

i) Functionally workable product, ii) User-friendly interface and the last most import measure is work fast (i.e.) performance for application in the client-side environment (browser).

Currently, most of the performance efforts like performance testing and optimizing is spent on the server side code to reduce response time (TTFB - time to first byte). As a result, while we have very

good TTFB on the server side the end-user experience might be very disappointing due to the delays occurring on the client-side(in browser). Page load time and time until UI is interactive are crucial

for positive CX.

**Here are some of the in-sights of the client-side performance metrics;**

* According to Google research (www.thinkwithgoogle.com), if page load time takes 1-3 seconds, the probability of bounce (when a user leaves the first page without further interaction) increases by 32%, whilst page load time of 10 s increases the bounce probability by 123%.
* Another use case from the Amazon, reporting increased revenue of 1% for every 100 milliseconds improvement to their site speed.
* A study by Akamai found that:
* 47% of people expect a web page to load in two seconds or less
* 40% will abandon a web page if it takes more than three seconds to load
* 52% of online shoppers say quick page loads are important for their loyalty to a site
* According to Yahoo, around 80% of application response time is spent downloading elements such as images, stylesheets, scripts, etc.

**Measuring client side performance;**

There're three major factors when we measuring client-side performance. Those are

1. Initial page load performance
2. Server-Side performance
3. Client side execution performance

**Tools we used to take client-side performance metrics are;**

1. Webpage Test
2. Google PageSpeed
3. GT Metrix
4. Yslow
5. Lighthouse
6. Chrome Developer Tools

**Conclusion;**

Based on the above mentioned metrics. Ameex recommendation is for static content page less than 1 sec and for dynamic pages which have images, stylesheets, scripts 2-3 sec. According to 80-20 rule, we derive the pages which generate value and revenue to customer, we strict with less than 2 sec of page load time.

**Server-Side Performance Testing:**

The primary objective of the server-side performance testing is to measure the response time of any application end user each action that is submitted to the server. Application should work accordingly to reach the service level agreement (SLA) in the non-functional requirements point of view.

**Server side performance testing life cycle;**

Server side performance testing life cycle contains the six phase's, each phase has its own significance to go into the next phase.

1. Study phase
2. Design phase
3. Build phase
4. Execution phase
5. Analysis phase
6. Reporting phase

**Different types of server-side performance testing;**

There are six types of performance testing. They are,

* Load Testing
* Stress Testing
* Endurance Testing
* Spike Testing
* Volume Testing
* Fail Over testing

According to the current assessment, we're more concentrate on doing Load and Stress test. For near future we'll concentrate on Endurance and Spike test. Volume and Fail over test are out of scope for this assessment

**Measuring server side performance;**

* CPU Utilization
* Memory Utilization
* Application – Requests Count, CPU Usage, Memory Usage
* Database – CPU, Memory, MySQL slow queries count

**Server-side Performance test strategy;**

**Load Test:**

Test the application with the normal load or production expected number of users to make sure that in the production application will perform well. Ideal time period of execution time is 1 to 2.30 hours mostly.

**Observations:** We need to observe the both client and server side metrics. From client side metrics; No. of users, Throughput, Hits per second, Transaction response time and from the server side metrics; CPU utilization, Memory consumption.

**Stress Test:**

Find out the break point of the application, which means how many users can be handling the application without any problems in it. Ideal time period of execution time is 30 minutes, without any wait time between transactions and also no wait time between the iterations to find out real break of users in the application point of view.

**Observations:** We need to observe the both client and server side metrics. From client side metrics; No. of users, Transaction response time and from the server side metrics; CPU utilization, Memory consumption.

**Endurance Test:**

Find out any bottlenecks is present in the application or not and system can sustain the continuous expected load. During soak tests, memory utilization is monitored to detect memory leaks in the application. Ideal time period of endurance execution time is 8 to 24 hours with 70% load test users will be considered to execute the endurance test.

**Observations:** We need to observe the both client and server side metrics. From client side metrics; No. of users and from the server side metrics; High memory consumption, Network Latency.

**Spike Test:**

Find out the performance issue in the application when sudden increase (or) decrease of the load on the application. Majorly spike testing is to observe any spikes are occurred in the application of memory utilization when the varying the user data dynamically to the application. Ideal time period of execution time is 15 to 20 minutes.

**Observations:** We need to observe the both client and server side metrics. From client side metrics; No. of users and from the server side metrics; Memory consumption, Network Bandwidth.

**Pre-requirements to do server-side performance testing;**

For doing server-side performance testing we need dedicated environment which should equivalent to production like environment (Same as production). If not we must need scale down environment (50% production server) or we need to change work load profile equivalent to the provided environment