

## Performance Testing on Mobile Applications

By Kelvin Kam

Gartner, a company which deals with technology research for global business leaders, predicted years ago that mobile phones will overtake PCs as the most common web access device worldwide by 2013. Well, in today's society, we can see that their prediction was very much spot-on. With mobile web usage growing at such an unprecedented rate, it is imperative that your mobile applications meet the performance and scalability expectations of consumers.

Mobile users have high performance expectations, whether using a mobile application or accessing a website from a mobile browser. According to Equation Research, 58% of mobile users expect sites to load roughly as quickly on their mobile phones as on their desktops at home.

Sluggish performance, errors, or crashes will send many users to a competitor, and may affect their willingness to do business with you at all, even from fixed workstations. The mobile app marketplaces are filled with competitive offerings; if you want a spot on a customer's smartphone, you have to earn it.

At the same time, you cannot afford to spend months and millions of dollars testing mobile apps before you deploy them. There are too many competitors and the market is growing far too quickly for you to sit on the sidelines; if you've yet to join the mobile app playing field, you're missing out on spreading your services to millions upon millions of potential customers.

In this article, we will look at some of the common issues, challenges, and strategies to tackle performance testing on mobile applications.

### Common issues

#### Server Side:

- Variation of response time
- Stream resource-intensive multimedia packets
- Delays/drops in delivery of message/emails
- Application crash
- Inefficient use of resources

#### Client Side:

- Application behaves differently on various platforms and handsets
- High memory and CPU consumption
- Mobile application too slow to load
- Application causing battery drain out

## Challenges:

- Prevalence of customer protocols
- Rapid scalability
- Lack of mobile monitoring solutions
- Lack of diagnostic tools
- Time to market
- Selection of load testing tool
- Test environment
- High concurrency

## Strategies

### Test in production environment

With the complexity of today's web infrastructure and mobile users scattered around the globe, extrapolating performance based on a limited lab test is dangerous. Testing in the lab or a staging environment is helpful, but is unlikely to reveal the performance and reliability problems that appear in production at scale.

To really test the performance of your mobile application, you have to extend your testing efforts beyond the application itself to the actual production environment. Only by testing in the production infrastructure can you take into account all of the components of your application including those outside the application itself. Production testing does not necessarily mean putting your production application at risk. There are ways to test in the production environment that limit your exposure. For example:

- Test during maintenance window
- Test in production but before the product is announced/released
- Test on slice of the production infrastructure

### Test for 2-3 times expected capacity

No one is really good at predicting peak loads- we're all just guessing. Even companies that have been in the business for years and have history to look back on make the mistake of assuming that past performance predicts the future. That's a dangerous assumption in today's online world. With the help of Twitter, Facebook and many other social networking sites, words can be spread quicker than you can imagine.

Because of this uncertainty, it is recommended that you estimate your peak traffic, and then test at two to three times that peak capacity. Testing at scale like this gives you confidence that your application infrastructure can handle an unexpected load without melting down.

### **Test accurately for the mobile environment**

Make sure that your performance tests can model the actual behavior and environment of your mobile users. In the diverse mobile application environment, you need to be able to accurately reflect the user environment, including web services calls from games or other apps as well as mobile browsers. For mobile web pages, testing should include multiple mobile platforms, including Blackberry devices, Android, iOS, WebOS, and Windows Mobile.

Once you have created accurate scripts from different mobile devices and user behavior profiles, you can then combine, parameterize or manipulate them to create the complex usage patterns and high traffic volumes needed for an accurate load test.

Ideally, you should be able to correlate end user performance on real mobile devices with performance metrics from your supporting backend infrastructure, to discover how infrastructure issues impact the end user performance.

### **Test across geographies**

By definition, mobile users don't stay in one place, and the Internet defies geographical borders; therefore, it is important to test across geographies for global reach. Send traffic from different parts of the country, over different Internet backbones, and from different countries and continents. Make sure that the application performs consistently and reliably across the entire infrastructure.

### **Test with real time results**

Engineers generally will use "mean time to failure" to predict the required maintenance cycle for an application. By estimating the number of transactions per minute, hour, or day for the application, you can extrapolate the number of transactions for a week or a month. The application should then be brought to a high but stable load rate. The site is then allowed to run indefinitely, and the total transactions are recorded. From that number, you can predict how long the site will run given normal conditions.

With fast-moving development cycles and rapid mobile adoption cycles, you cannot take weeks or months to test. You need to see results in real time, so you can make changes on the fly and test those changes as part of an iterative performance test. An example of this is when the site is placed under increasing load levels until a bottleneck is reached (site becomes non-responsive due to a downstream dependency). That bottleneck should then be fixed, either by fixing the code or by scaling up or out.

### **Summary**

When testing mobile apps at scale, you do not actually need to deploy legions of mobile users around the globe on iPhones, iPads, Android phones or other devices. Instead, use a proxy to capture all HTTP or HTTPS from each tested device performing the actions you want to test. Instead, you can use load generators to simulate the behavior of those devices, which are fairly easy to create and don't tend to require complex scripting. New technologies like Cloud testing will also offer the fast test deployment

and real time insight necessary to get the testing done without delaying deployment or adversely affecting production performance.

### References

1. <http://searchsoftwarequality.techtarget.com/tip/Testing-functionality-performance-of-mobile-Web-applications>
2. <http://mobiforge.com/forum/testing/testing-simulators/performance-testing-mobile-applications>
3. <http://batterywalam.wordpress.com/2010/02/16/performance-testing-of-mobile-applications/>