Mobile Application Performance Testing

Introduction

Purpose

This discussion paper aims to educate the reader about mobile applications and demystify their performance testing needs by explaining the anatomy of, and outlining a sensible performance testing framework for, mobile applications.

Scope

The scope of this paper covers end to end performance testing in relation to mobile applications. It does not cover testing the performance or suitability of specific mobile hardware or operating systems, and it does not cover performance testing in general. Both functional testing and wider non-functional testing are excluded from the scope of this paper.

Context

*"**Online spending** with Visa is now at an **unprecedented** scale across all sectors...largely **fuelled by** the explosion in **smartphones and tablets**."* Marc O'Brien, MD, Visa UK

Mobile** phones and tablets **now account for 23%** of all UK **online sales – IMRG and Capgemini

*By 2016, **mobile** app **projects** will **outnumber traditional** app **projects** by **4:1*** – HP

29 versions** of **Android** released since 2007; **420** active **models** of Android mobile phones; **18 versions** of **iOS** released since 2007; **7** common operating **platforms – HP

*"**mobile** and **tablets** are becoming the **dominant** devices used to **interact** with a brand **online**"* – Kate Smyth, Director of E-commerce, Dune

Background

Mobile applications on smartphones and tablets have been around for several years, mainly led by social networking and gaming. Corporate business, now realising that mobile applications offer both the potential for increased income and the defence against brand-obscurity in this increasingly mobile oriented world, is pushing its traditional web presence onto mobile applications.

As a result of this, corporate governance and application lifecycle best practices are being applied to the once 'wild-west' of mobile application development, and one of these key disciplines is performance testing. However, many organisations are struggling to map their existing knowledge and methodologies for performance testing traditional web applications to the paradigm of the mobile application.

Common Questions about Mobile Application Performance Testing

This discussion paper aims to answer the following common questions:

- How do you do it?
- What tools should I use?
- How do I test the network?
- How do I test using different devices?
- How do I assess the performance of my application on different mobile operating systems?
- How do I get the user perception of using the application on a real device?
- Where do performance bottlenecks exist in my infrastructure?
- How many concurrent mobile users can my application support?

Contents

Anatomy of a Mobile Application

Before we can consider how to performance test a mobile application we should take a little time to understand what makes up a mobile application.

Types of Mobile Application

Generally speaking, mobile applications are deployed in one of the following three flavours:

Native

Applications that are installed and run on the mobile device. A network connection is not required for these applications to run. Typical examples include utilities and games

Browser

Applications that are accessed through the mobile device’s web browser. Browser applications require an active network connection. Typical examples include mobile-tailored[1] versions of web-based email, news and shopping sites.

Hybrid

Applications that are installed and run on the mobile device and require a network connection to perform actions on, and access data and content from, a paired-service on the internet.

Some Hybrid applications are designed to cache network data and therefore can operate for a period of time or with limited features without an active network connection, others require an active connection to operate.

Typical examples include social networking applications, banking applications and video streaming applications.

Although the performance of Native mobile applications is important, given the wide range of available mobile handsets and tablets with varying degrees of processing power and memory, the focus of this discussion is the end to end performance of mobile applications that include a network based service. Native mobile applications are client-side only hence this discussion will concentrate on Browser and Hybrid mobile applications.

[1] It should be noted that Browser applications can also include full-desktop versions of internet sites, but for the purpose of this discussion these are excluded.

Architecture of Browser and Hybrid Mobile Applications

A simplified typical client<>network<>server architecture for a Browser or Hybrid mobile application is presented in Figure 1 below.

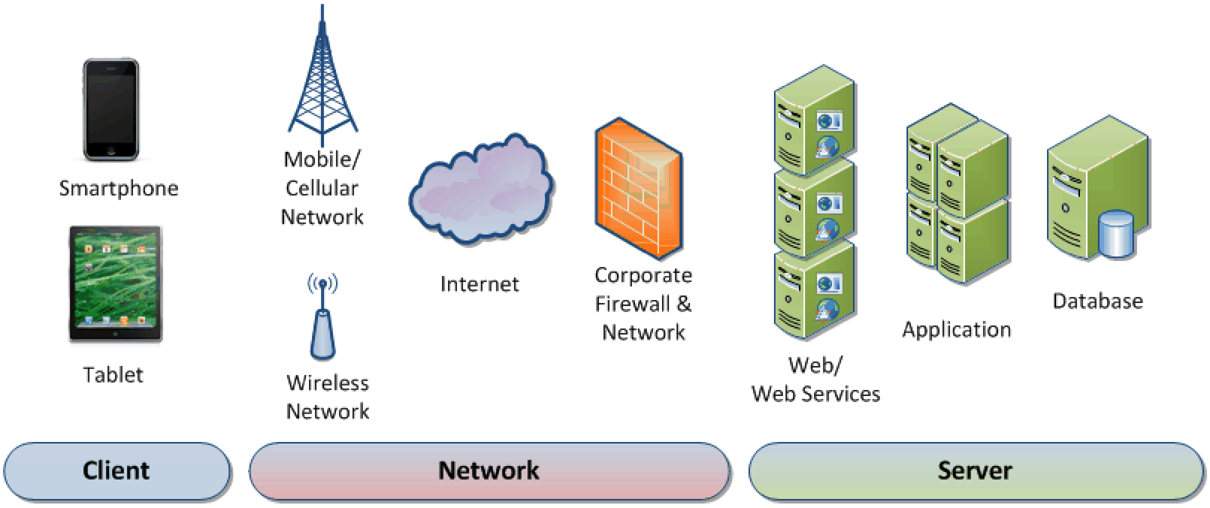


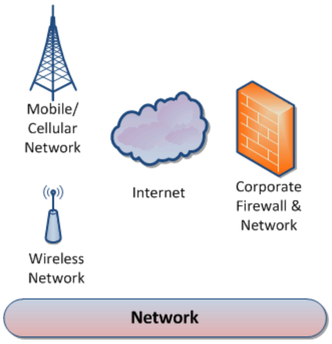
Figure 1 Mobile Application Architecture

As far as mobile application performance is concerned there are three elements to any mobile application:

Client

The presentation of the application to the end user. This will include:

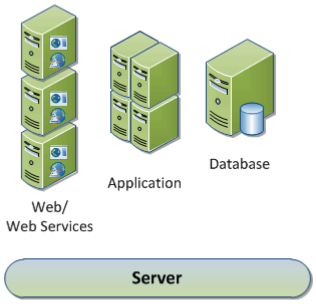
- Device – Type (handset or tablet), hardware (processor, memory, screen, network capability (3G, 4G, etc.), etc.), operating system (iOS, Android, Windows, BBE)
- Application Type – Browser or Hybrid
- Application – The code deployed
- Other Applications and Services on the device competing for resources



Network

The transport of data between the client and server:

- Mobile/Cellular Network – Signal strength, connection type (3G, 4G, etc.), network congestion, black-spots, dropouts, MNP's internal network for the mobile routing
- Wireless Network – Speed, signal strength, congestion.
- Internet
- Corporate Firewall and Network



Server

The back-end paired service. Arguably in a well-designed mobile application the majority of processing should be performed on the server-side.

Performance Testing a Mobile Application

The first question to ask is why are you performance testing the mobile application? The chances are the answer will be one of the following:

- To ensure that it can support *n* number of users without degradation or failure
- To ensure that its response time is less than *n* seconds when under peak load conditions
- To ensure that the new release performs at least as well as the current release
- To ensure that the application will work with a 'mobile' network connection

You should then ask yourself what you are trying to prove or establish. This may be more difficult to answer:

- Are you trying to prove the performance of the handset or tablet?
- Are you trying to establish what happens to application if the mobile signal is weak or lost?
- Are you trying to prove that your network has enough bandwidth to support the mobile application?
- Are you trying to establish if the server-side application is scalable enough to support the peak usage?

In reality you are probably trying to prove all of the above. No single method of testing can achieve this, so a third question to ask is how much are you willing to invest in performance testing? The answer to this will inform how much of the following you will want to or can afford to do:

- Server-side performance testing
- Network performance testing
- Client-side performance testing

Mobile Application Server-Side Performance Testing

Ensuring that your server-side application and systems are scalable is a good place to start in performance testing a mobile application and in many cases will be all that is required to give confidence for go-live.

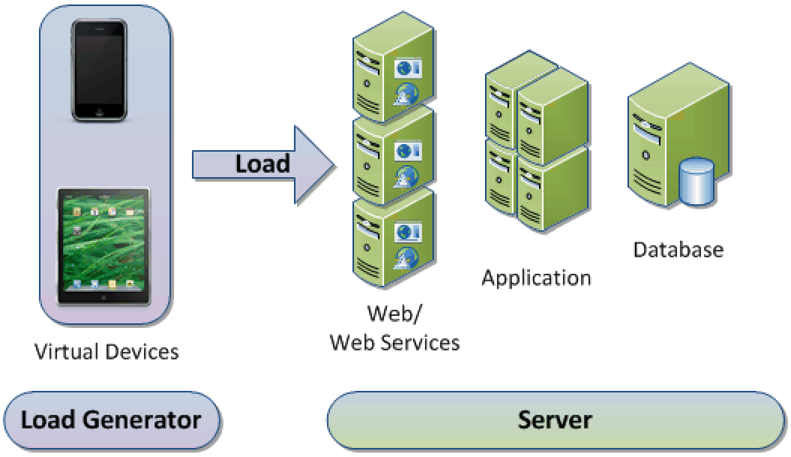


Figure 2 Performance Testing Server Side

Performance testing the server-side of a mobile application is very like testing a traditional web-based application. The protocols a mobile application uses are typically web, web services and streaming, and these are widely supported in performance testing tools.

Using Your Existing Performance Testing Tool Licence

If you are tech-savvy enough, the ‘traditional’ web, web services and streaming protocols, which your existing performance testing tool may already be licensed for, are suitable for doing this type of testing.

Test scripts can be recorded by pointing either a mobile device emulator (of which there are many) or a real mobile at the performance tools proxy recorder and then executing the steps of the business process you wish to record. (Don’t worry - your techies will know how to do this). The recorded script contains the web, web service and streaming calls that the mobile application made.

Test scenarios can be built, executed and analysed in the same way as a traditional web application performance test.

Using a Specific Mobile Application Performance Testing Protocol

Most performance testing tools now come with specific protocols for Mobile Applications. Generally these encapsulate the tech-savvy approach outlined above and integrate a mobile device emulator or install capture software on the mobile device for simplified script recording.

In most cases the recorded script contains the web, web service and streaming calls that the mobile application made.

Some mobile protocols record scripts that contain the actions made on the mobile device (typing, pressing buttons, pinch/pull, swipe, etc.). When performance tests are executed, each ‘virtual’ user loads an emulated device and replays the actions. This arguably results in a more accurate test as it is the real application being used rather than a replay of its communication calls, but its larger footprint requires more load generation hardware and it should not be mistaken for mobile client testing (discussed on page 9).

Mobile Application Network Performance Testing

Once the server-side of the mobile application is tested and tuned for scalability, testing can change focus to the network performance testing[2].

Network Emulation can be introduced to assess the impact of the network on the performance of the mobile application. Most of the enterprise performance tools have built in network emulation capabilities and standalone network emulation tools are available.

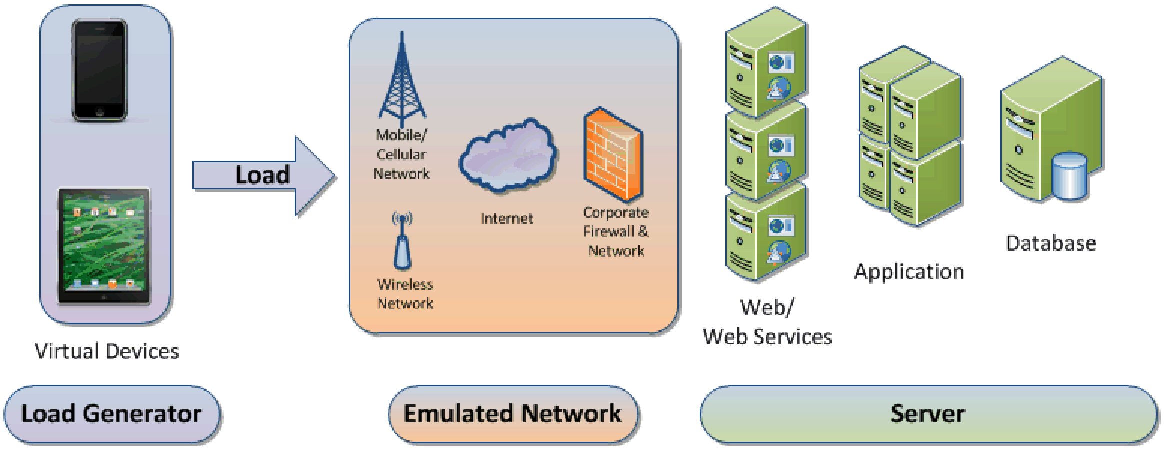


Figure 3 Performance Testing inc. Emulated Network

The same tests used for the server-side performance testing can be repeated and the Network Emulation can be configured to emulate a variety of network conditions (3G, 4G, WiFi, etc., network congestion, black-spots, dropouts, etc.). The mobile application can be tuned for optimal performance and various what-if scenarios can be carried out.

[2] Network performance testing commonly consists of assessing three things:

1. The impact of the network (or other things on the network) on the performance of the mobile application
2. The impact of the mobile application on the performance of the network (or other things on the network)
3. The performance capability of the network

Testing of items 2 and 3 on the list would be limited to the corporate network, and therefore will not be included in this discussion.

Mobile Application Client Performance Testing

Whilst performance testing the server-side and network of a mobile application employ traditional performance testing methodology, that is 'virtual' users apply load on the 'real' server-side application, performance testing the client of a mobile application requires a fresh approach.

Such is the power of perception, and the damage a slow or inconsistent mobile application could do to a brand, many organisations are not satisfied with the 'virtual' results. They want to understand the 'real' user experience, and they want to understand this on a variety of devices, operating systems and mobile networks.

Figure 4 below illustrates how this can be achieved with a 'real' user testing in conjunction with a server-side performance test.

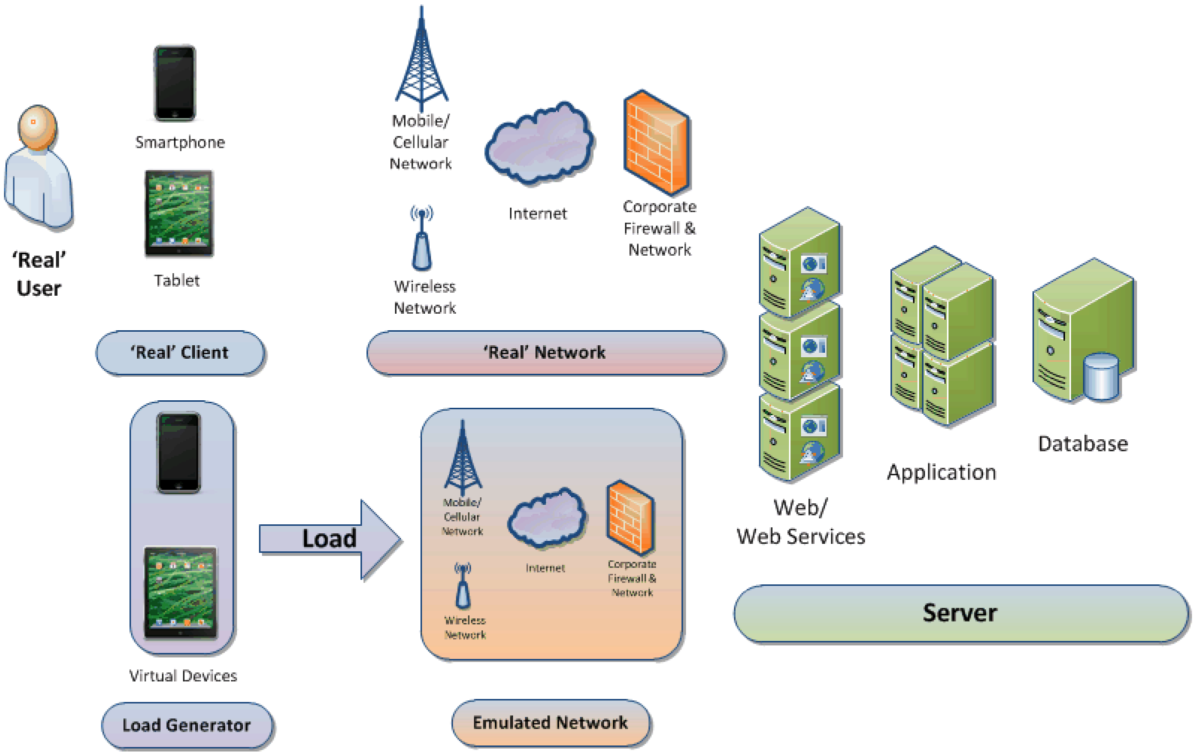


Figure 4 Performance Testing 'Real' User Experience

There are several options for gaining the 'real' user experience:

- Manual testing
- Automated functional testing
- Real-device testing

Use Manual Testing

The simplest form of ‘real’ user testing and probably the easiest concept to understand is to have a tester use the real mobile application, on a real device, over a real mobile network. As the performance test executes, the tester can execute a manual test against the mobile application and report on its response times and usability.

This manual approach is perfect for ad-hoc testing as it is simple to do and requires no special software or hardware, but it falls short if there is a requirement to test multiple devices, on different networks from multiple locations due to logistical complexity.

An alternative to a lone manual tester is to use a Crowd Sourced Testing service. Again, whilst a server-side performance test is executing the crowd of testers can each execute a manual test and report on the performance. Crowed Sourced Testing services can offer a variety of devices, networks and locations, and if manual testing is the preference they offer a good solution.

If the requirement is to test regularly, with exact device specifications and network and location distributions, and provide accurate, not perceived, timings then a Crowed Sourced approach will not be suitable because the same testers will not be available.

Use Automated Functional Testing and Device Emulation

To achieve test repeatability and accurate timings automated functional test scripts can be used, and executed alongside the server-side performance test. Many performance testing tools allow these automated functional testing tools to be plugged in, meaning the entire test process can be controlled by the performance test tool.

This approach offers excellent re-use of exisiting tools and scripts, and the real client-side mobile application will be executed rather than just the web, web service and streaming calls that the mobile application made.

However, the approach uses an emulated mobile device rather than a ‘real’ device and it will not be using a mobile network so it falls short if these are requirements.

Use a Specialist Real-Device Testing Service

Several new services are available that allow testing using real devices on multiple mobile networks in multiple worldwide locations. These services are mainly cloud based, generally charge on a usage basis, and can be accessed without the need of specialist software or hardware. Some suppliers offer their hardware/software for installation on your network.
These real device testing services come in two flavours:

- Real mobile devices with ‘remote-control’ software installed
- Real mobile devices integrated in controlling hardware

This approach gives the ability to frequently run tests, take accurate timings often with further diagnostic information, and on multiple exact device specifications from multiple geographic locations and mobile networks.

The services can run standalone, alongside a server-side performance test. And many of the services offer integration into popular automation tools allowing the entire test process to be controlled by the performance test tool.

Client Testing Options Summary

A summary of the options for testing the ‘real’ user experience is presented in Table 1 below.

	Tests the Real Mobile Application	Provides the User Perception of a Real User	Provides Accurate Transaction Timings	Allows Regular Repeatable Testing	Multiple Device and Operating System Configurations[3] available	Uses Mobile Network[4]	Allows Multiple Locations / Mobile Networks	Uses Industry Standard Automation
Manual Tester	✓	✓				✓		
Crowed Sourced Testing	✓	✓			✓	✓	✓	
Automated Functional Testing	✓		✓	✓		✗		✓
Real-Device Testing	✓		✓	✓	✓	✓	✓	✓

Table 1 'Real' User Testing Options

[3] e.g. Apple, Samsung, HTC, iOS, Android, Windows, BBE

[4] The application will need to be exposed externally in order to test from Mobile networks. This may not be allowed in all development environments

Summary

By understanding what you require to achieve from performance testing a mobile application it is possible to incorporate this testing within your current performance testing framework, which is illustrated in a simple form in Figure 5 below.

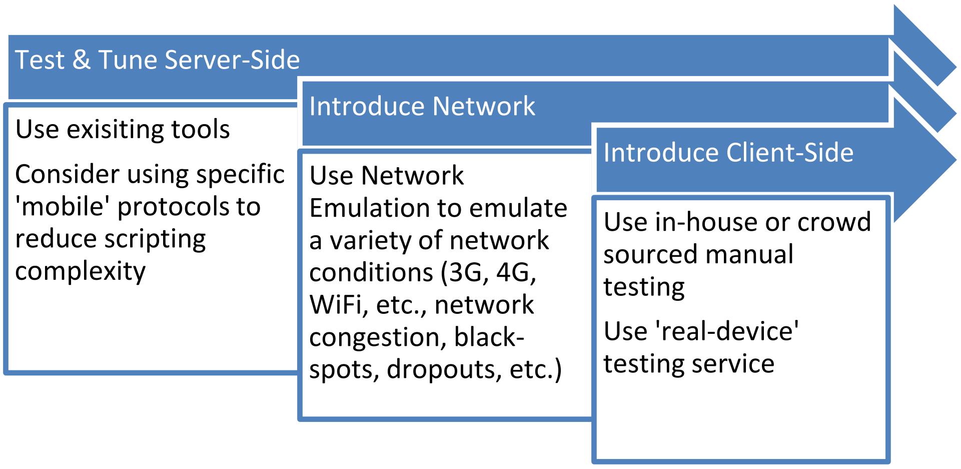

Test & Tune Server-Side
Use exisiting tools
Consider using specific 'mobile' protocols to reduce scripting complexity
Introduce Network
Use Network Emulation to emulate a variety of network conditions (3G, 4G, WiFi, etc., network congestion, black-spots, dropouts, etc.)
Introduce Client-Side
Use in-house or crowd sourced manual testing
Use 'real-device' testing service

Figure 5 Recommended Mobile Application Performance Testing Framework

	Identify Server Side Bottlenecks	Prove Server-Side Scalability	Tune Application for Network Conditions	Evaluate Network What-Ifs?	Understand Real User Experience	Evaluate Performance on Variety of Devices
Test & Tune Server-Side	✓	✓				
Introduce Network			✓	✓		
Introduce Client-Side					✓	✓

Figure 6 Mobile Application Performance Testing Coverage

Conclusion

A new breed of testing tools and services has developed around mobile application testing, and many of these are being marketed towards performance testing. It is worth understanding what you want to achieve from your mobile application performance testing, and if your existing investment in performance testing covers these, before making choices on tools and services.

Traditional testing methodology and tools can be effectively deployed to test and tune the performance of the server-side of mobile applications. It may be advantageous to use the new mobile protocols the tool vendors offer to simplify the test development effort.

Network emulation tools offer cost effective alternatives to undertaking full-scale 'real' user performance tests to establish mobile application performance under 'real' network conditions. If incorporated correctly into the test strategy, the configurable and repeatable nature of network emulation will offer greater testing and tuning possibilities for the mobile application.

If the 'real' user experience is a requirement of performance testing, then manual testing (either in-house or crowd-sourced), automated functional testing or using the new real-device testing services can be used to provide this. It is recommended this is done in conjunction with server-side performance testing and network emulation to give the full picture on mobile application performance.

StarBase is the UK's leading performance testing consultancy, trusted by major enterprises to mitigate IT systems risk since 1992. We ensure the consistent performance of our clients' key systems. As specialists, we possess an unrivalled depth of experience, knowledge and insight, creating smart and value-adding solutions for complex situations.

Performance Testing

Combining the best testing tools with effective processes can improve the bottom line

Testing Tools

Guidance on testing tool selection - the right tools for your technology challenges and to improve productivity

Functional Testing

Smarter processes and greater automation will reduce cost and duration of the project

Managed Testing

Let us take responsibility for all your testing requirements

Address
StarBase Computer Services Ltd
Calleo House
49 Theobald Street
Borehamwood
WD6 4RT
United Kingdom

Telephone
+44(0) 20 8236 7000

Email
sales@starbase.co.uk

Website
www.starbase.co.uk