Measuring Ajax Performance on a GPRS Mobile Platform

F. Xie[*], D. Parsons[**]
Institute of Information and Mathematical Sciences, Massey University, New Zealand
Xiefeng401@gmail.com[*] ; d.p.parsons@massey.ac.nz[**]

Abstract

Today, mobile technology is rapidly affecting our society, with increasing numbers of services supported by mobile phones, including mobile Internet access. However, mobile Internet access performance over GPRS networks is often unacceptably slow. A new web development model, Ajax, may help to address this problem. Ajax (Asynchronous JavaScript and XML), is a new desktop approach to web application development that uses client-side scripting to provide a seamless user application experience and reduce traffic between client and server. In this paper we address the question of whether mobile Ajax provides measurable performance advantages over non-Ajax mobile applications. A study of web application performance over a GPRS network was undertaken, based on comparing an Ajax application and an Active Server Pages (ASP) application with identical functionality. Our results suggest that mobile Ajax over GPRS can reduce the bandwidth requirement by about 70% and cut the server's response time in half.

.

***Keyword(s)**: Ajax performance; Mobile Phone; GPRS network; Mobile Internet*

1. Introduction

Over the past few years, an increasing number of people have been using mobile Internet access, but due to the limitations of mobile phone web applications and hardware, it is still very problematical. There are a number of reasons, for example; small screen, high network latency, low bandwidth, interface complexity and so on. To address these issues, we may look to recent developments in desktop browser technologies that might also be implemented in the mobile environment to improve the mobile Internet access experience.

In February 2005, Garrett [1] introduced a new web application approach, Ajax (Asynchronous JavaScript and XML), on the desktop. Compared with the traditional web application model, Ajax can significantly increase a web page's download speed in the desktop environment. It has been applied by White [2] in the business realm, and in academia by Smullen and Smullen [3]. Ajax reduces the data transmission volumes between the server and client device, and improves the user experience on the desktop. These are relevant issues for mobile Internet access, which needs a solution to reduce transmission volumes in the context of expensive and low speed connections, and provide a better user experience in browsers with limited screen real estate and navigation tools. Therefore we might ask the question; if we deploy an Ajax approach to the mobile environment, what might be the result? Does Mobile Ajax provide measurable advantages over a non-Ajax mobile application?

The goal of this research is to evaluate the benefits of Ajax over non-Ajax applications, when users access websites via mobile devices through a General Packet Radio Service (GPRS) network. First, we developed two websites with identical functionality, using both an Ajax approach and a non-Ajax approach (using ASP). These two websites have the same user interface and can be accessed both by desktop and mobile browsers. Second, we measured these websites' performance over a GPRS network based on the data collected from the Web server's log files.

2. Related Work

2.1 The Ajax Approach

Garrett [1] defined Ajax as a set of powerful, widely-used, well-known and mature technologies, combining them together to create a powerful new interaction approach:

- ✓ Standards-based presentation using XHTML and CSS;
- ✓ Dynamic display and interaction using the Document Object Model;
- ✓ Data interchange and manipulation using XML and XSLT;
- ✓ Asynchronous data retrieval using XMLHttpRequest;
- ✓ JavaScript binding everything together.

The Ajax web application approach altered the following two features of the traditional web model to produce high performance and more interactive web applications.

"Partial screen update" takes the place of the "Click, wait, and refresh" user interaction model.
Traditional web pages refresh the whole page, every time some new information arrives from the server. During this time users must wait for the page to refresh. Ajax only updates the part of the user interface on the screen that contains the new information, retaining the current web page in the browser [4]

The "synchronous request/response model" is replaced by an asynchronous communication model.
The other key feature for AJAX technology is using an asynchronous communication model, using JavaScript to manage all requests to the server. As a result, requests will continue to be sent out if necessary, but users do not need to wait for the responses. All communication will be done in the background; meanwhile the user can continue to use the Web application online. When the new information arrives, JavaScript will partially update the user interface to provide the user with the latest information so the user can enjoy a seamless browsing experience [4].
In summary, Ajax can theoretically increase a web page's interactivity, speed and usability.

2.2 GPRS network (Bandwidth and Latency)

GPRS (General Packet Radio Service) is the world's most ubiquitous wireless data service, available now with almost every GSM network. GPRS is a connectivity solution based on Internet Protocols that support a wide range of enterprise and consumer applications [5]. GPRS can be utilized for services such as WAP (Wireless Access Protocol) access, SMS (Short Message Service) and MMS (Multimedia Message Service), but also for Internet communication services, such as email and web access. Unfortunately, web access over GSM (GPRS) using the TCP/IP protocol is problematical, with users experiencing very poor performance [6]. There are various reasons for this, for example high and variable latency, fluctuating bandwidth, occasional link 'blackouts' [7], packet loss, and link outages. Sometimes, a simple request can take several seconds [8].

2.1 Ajax Performance on the Desktop

Previous research suggests that Ajax can significantly improve web application performance on the desktop. One commonly cited Ajax performance evaluation is the one reported by White [2]. In his study, the Ajax application transferred on average just 27% of the bytes that were transferred by a traditional HTML application. Not only was there an improvement in the transferred byte volume, but there was also an improvement in performance, a 68% overall improvement in data transfer time.

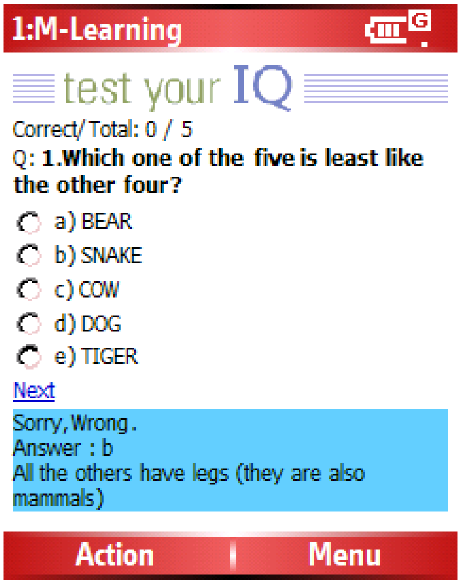
The two applications used in this study, however, did not have the same user interface. The users' skill levels and training were also not assessed in the report, and these factors may have affected the experimental outcomes. A more controlled Ajax performance measurement comes from Smullen and Smullen [3]. They compared the client-side performance of a real-life HTML application and an Ajax application that implemented the same user interface. Later they extended their study by collecting data on a statistically significant sample size and included server performance results. Response size and service time performance measures computed for the applications provided significant performance improvements in response size for the Ajax application (56%), thereby reducing bandwidth requirements. Ajax provided a mean service time improvement of approximately 16% [9].

Although Ajax performance has been measured on the desktop, there is no currently available performance measurement research about Ajax on mobile devices. Therefore we do not know if it can improve mobile Internet performance, or indeed if Ajax is effective in the mobile environment. Therefore this study focuses on measuring Ajax performance in the context of the mobile Internet.

3. Methodology

This study is based on a controlled measurement of data transfer volumes and times, comparing an Ajax web application with a similar non-Ajax implementation. We developed two IQ test (multi-choice) Web applications with the same interface and functionality (Figure 1), to compare the performance of an ASP based website and an AJAX based website in a GPRS mobile network environment. We collected the experimental data from the server and mobile phone, including bandwidth usage, response times, and mobile unit storage requirements.

We initially predicted the websites' performance based on paper calculations, theoretically analyzing the data volumes and response times over the TCP/IP protocol. Second, we analyzed real-life data collected from the system log of the web server.



Correct/ Total: 0 / 5

Q: **1.Which one of the five is least like the other four?**

○ a) BEAR

○ b) SNAKE

○ c) COW

○ d) DOG

○ e) TIGER

Sorry,Wrong.
Answer : b
All the others have legs (they are also mammals)

Figure 1. Website interface

The mobile phone web application was based on a three layer architecture; the mobile phone user interface (web browser), server pages and a database server. The web server was Microsoft Internet Information Services (IIS), the server pages were written using ASP 2.0 and the test database was stored in Microsoft Access. All dynamically generated web pages were valid XHTML 1.1, compatible with the mobile browser (Opera Mobile) used in the test. All data transfer was through the Vodafone New Zealand GPRS (GPRS/GSM 900/1800/1900MHz) network by Class 2 "2+1" multi-slots (2 downlink, 1 uplink channels).

The system logs were activated in the IIS server, so all traffic going to and from the web server was recorded. This enabled us to isolate and monitor all the transferred data (inbound and outbound) between the web server and mobile terminal. The collected data was used to analyze and measure the websites' performance over the GPRS network.

4. Data Collection

4.1 The Web Applications

The ASP Web application only contains three ASP pages; a start web page, a question display webpage, and a finish web page. These pages interact with the database to provide dynamic content. Other supporting files are an image file (in GIF format) and a JavaScript file to manage the question fetching process. When the user clicks on the submit button, it activates a JavaScript function that sends a request to the server to fetch a new question or answer.

The Ajax application version is similar to the ASP web application. On the client side, it includes three HTML pages, a JavaScript file, a GIF image file, and a CSS file. There are also an ASP file and an Access database running on the web server. The ASP file used in the Ajax application is different from those in the ASP application. It is used as both a database connector and XML file generator, because the communication between client and server is via XML.

These web applications have the same user interface, use the same image file (iq_header.gif) and CSS file (mobile.css) and share the same database.

4.2 Web Application Performance

Although the Ajax web application and ASP web application have much in common, they showed quite different performance in our experiment. The data collected from the IIS web server system log comprised hundreds of entries, so all results stated here are averages.

First, the initial load size of the applications has some impact on performance. When the client-side web browser loads the web application for the first time, some files need to be downloaded. The ASP web application's initial load size is about 6Kb, including an ASP file, a JavaScript file, an image file, and a CSS file. It takes the server about 800ms to respond. For the Ajax web application, the initial load size is about 9Kb, with a 1400ms server response time. In this case an HTML file, a JavaScript file, an image file, and a CSS file will be loaded. The image and the CSS file are the same, and there is a small difference between the sizes of the HTML pages and those generated by the ASPs. The major difference, however, is in the JavaScript, because the Ajax JavaScript file is about 50% larger than in the ASP version. This is because the Ajax JavaScript contains some additional logic and layout control functions.

The second performance difference is evident when the web application is running. For each multi-choice IQ test question, the ASP version needs to fetch the question page first and then refresh the page again with the response, after the user has submitted their answer. This requires two download activities from the server. The Ajax version only requires one download activity for each question, because the JavaScript can verify the user's answer on the client, avoiding the second request-response cycle. The ASP version cannot do this because it can only run on the server.

There are 9 questions in the test application, after the initial question. For the ASP version, each question needs to transfer 4.6Kb of data, with a 570ms server response time. Therefore 9 questions need to transfer 41.4Kb of data and use 5,130ms of server time. For Ajax, an average of only 555 bytes of data needs to be transferred for each question and the server takes 210ms to respond. Therefore 9 questions need 4.995Kb of data and 1,890ms of server time.

Taking both initial load and subsequent page loads into account, the ASP application needs to transfer a total of 47.4Kb of data compared to only 13.5Kb for the Ajax application, and takes a total of 5,987ms to respond compared to only 3,268ms.

5. Results

In previous studies, both White [2] and Smullen and Smullen [3] used the following algorithm to define the percentage of Ajax application performance improvement:

(HTML - Ajax) / HTML

According to this algorithm, the Ajax approach in our experiment provides almost a 50% performance improvement when using a mobile device over GPRS (Table 1).

	Response bytes	Response seconds
ASP	47.4 Kb	5987ms
Ajax	13.5 Kb	3268ms
Improvement	71.51%	45.41%

Table 1: Performance Improvement

6. Conclusion

The results of our experiments demonstrate that the Ajax approach can significantly reduce both the data transmission size and the server's response time. By reducing the required bandwidth for mobile interaction, and speeding up the user interface on the mobile device, a mobile Ajax system is able to provide the user with a better mobile Internet experience. Of course our results are based on small experimental applications, and we have not so far included any usability studies from the mobile user perspective, but we believe that these results are promising and suggest that mobile application developers would benefit from integrating Ajax into their development toolset.

Acknowledgments

Sincere thanks to the Institute of Information and Mathematical Sciences, Massey University, for their generosity in letting us use their computing resources to conduct our experiments.

References

[1] J.J. Garrett, *Ajax: A New Approach to Web Applications*. http://adaptivepath.com/publications/essays/archives/000385.php , 2005. Last accessed December 9th 2007.
[2] A. White, *Measuring the Benefits of Ajax*. www.developer.com/java/other/article.php/3554271, 2005. Last accessed December 9th 2007.
[3] C.W. Smullen and S.A. Smullen, *AJAX Application Server Performance*. Proceedings of the IEEE SoutheastCon, Richmond, VA, USA. March 22-25 2007.
[4] C.K. Wei, *AJAX: Asynchronous Java + XML?* http://www.developer.com/design/article.php/3526681, 2005. Last accessed December 9th 2007.
[5] GSM Association, *GPRS Roaming Guidelines* http://www.gsmworld.com/technology/gprs/index.shtml, 2003. Last accessed December 9th 2007.
[6] R. Chakravorty and I. Pratt, *WWW Performance over GPRS*. Proceedings of the 4th International Workshop on Mobile and Wireless Communications Network, Stockholm, Sweden, September 9-11 2002.
[7] R. Chakravorty, J. Cartwright, and I. Pratt, *Practical experience with TCP over GPRS*. Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02), Taipei, Taiwan, November 2002.
[8] P. Stuckmann, N. Ehlers, and B. Wouters, *GPRS traffic performance measurements*. Proceedings of the 56th Vehicular Technology Conference, Vancouver, Canada, September 2002.