

Research and Analysis of Ajax Technology Effect on Information System Operating Efficiency

Xiao Zhang, Yi Zhang and Jun Wu

School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China

newzhangxiao@126.com zhy@bupt.edu.cn junwu@bupt.edu.cn

Abstract. In recent years, information system based on browse/server architecture (namely B/S architecture) received more favor by enterprises. Ajax technology consists of five parts. They are HTML, JavaScript, DHTML, DOM and XML. With the help of cooperation and collaboration of these technologies, they can optimize the conventional enterprise information system by using an asynchronous way. Meanwhile, a quickly-responded and smoother user interface was provided. Enterprise information system with Ajax can be operated in a more efficient way, which means even use the current hardware, it can provide more load capacity, be more stable and serve more clients in parallel. In this paper: we present two kinds of information system models, one use conventional B/S architecture and the other use Ajax enhanced B/S architecture. First, we build both of the systems in accordance with typical business applications (search files, database access, etc.). Second, we use standard web pressure test tool such as Microsoft Web Application Stress tool to test both of the systems to get information like concurrent user number and average response time. Finally, with those experimental data, I compare and found out the difference between the two systems. The results presented in this paper propose a good way for enterprises, to enhance the information system performance, capacity and stability under a definite hardware facilities circumstance.

Keywords: *AJAX, Asynchronous transfer, Enterprise information system, System performance*

1. INTRODUCTION

Ajax (also known as AJAX), shorted for "Asynchronous JavaScript and XML"[1], is a web development technique for creating interactive web applications. The intent is to make web pages feel more responsive by exchanging small amounts of data with the server behind the scenes, so that the entire web page does not have to be reloaded each time the user requests a change. This is meant to increase the web page's interactivity, speed, and usability.

As we know, widely used information systems such as mySAP, IBM WebSphere [2], Microsoft SharePoint [3] are all based on browse/server architecture. Meanwhile, this kind of system is consisted of thousands of web pages. So it is a good practice to combine the Ajax technology with those information system in order to get a more

user friendly UI and other benefits? This is the main issue that we will talk about in this paper.

2. AJAX OVERVIEW

Ajax (Asynchronous Javascript And XML technique) consists of five parts. They are HTML, JavaScript, DHTML, DOM and XML. With the help of cooperation and collaboration of these technologies, they can optimize the conventional web interactive mode (User triggers a HTTP request by clicking hyperlinks -->Server gets the request, processes it and visits the database for data -->server generates a HTML page and sends back to the client) by using a asynchronous way(See Figure 1). In this way, there is no interactive process like "Click -> Wait -> Click -> Wait". Meanwhile, a quickly-responded and smoother user interface was provided.

After using Ajax to optimize the information system of browse/server architecture, clients get better experiences. Furthermore, it's also means a lot for the servers. Because Ajax reduce the redundant and duplicated data that were generated and transferred before, now enterprise information system can be operated in a more efficient way, which means even use the current hardware, it can provide more load capacity, be more stable and serve more clients in parallel.

3. EVALUATION CRITERIA

To compare the two kinds of system, I need to set up an evaluation criteria first.

Considering the aspect of TCO (total cost of ownership) of the information system, I define the less bytes transferred the better, on the premise of finishing the same function. This criterion is set base on the following reasons: firstly, process less data will lower the CPU's workload for generating web pages; secondly, transfer less data can save the bandwidth and cost especially for the network that are charged by volume; thirdly, less data means less energy and less usage of the devices, this in turn reduce the cost of device replacement and energy consumption.

Considering the aspect of user experience, I define the average response time as a parameter to measure. The average response time is calculated by TTFB (Time To First Byte) and TTLB (Time To Last Byte), which shows the delay between system opens the web page and finishes it. Using this parameter, it is easy to judge which is better. Because less opening time means less wait and enhanced user experience.

Considering the aspect of system performance, I define the successful requests per second (RPS) as a key criterion. Due to this criterion can effectively reflect system concurrent performance especially when system works under a very heavy workload.

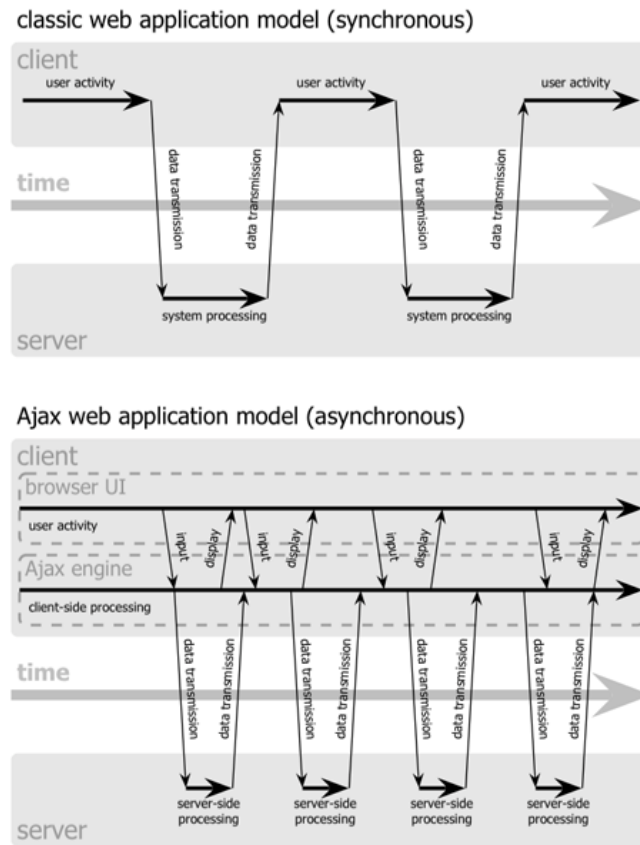


Figure 1. Compare Conventional Web Data Transfer Mode with Ajax Asynchronous Interactive Mode

To display the formula clearly, I set Ax for Ajax optimized information system and Cx for conventional information system based on browse/server architecture. So we have:

$$\text{Data Efficiency Index} = \frac{\sum_{i=1}^x \frac{A_{\text{BytesforFunction}}(i)}{C_{\text{BytesforFunction}}(i)}}{i} \quad (1)$$

$$\text{Page Display Efficiency Index} = \frac{\sum_{i=1}^x \frac{A_{\text{TTFBforPage}}(i)}{C_{\text{TTFBforPage}}(i)}}{i} \quad (2)$$

$$\text{Page Load Efficiency Index} = \frac{\sum_{i=1}^x \frac{A_{\text{TTLBforPage}}(i) - A_{\text{TTFBforPage}}(i)}{C_{\text{TTLBforPage}}(i) - C_{\text{TTFBforPage}}(i)}}{i} \quad (3)$$

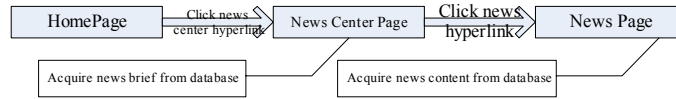
$$\text{System Performance Efficiency Index} = \frac{A_{\text{RPS}}}{C_{\text{RPS}}} \quad (4)$$

These four indicators will show the difference between the two kinds of information system.

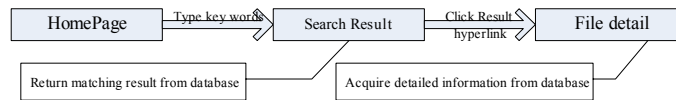
4. SIMULATION

To compare the two kinds of system, I set up several scenes to simulate the actual working action such as looking for internal news update, search for some file in database, surf specific pages in knowledge base, etc [4-7].

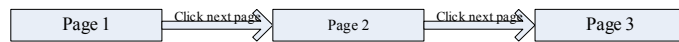
Scene 1: Looking for Internal News Update



Scene 2: Searching for Some File in Database



Scene 3: Surf Specific Pages in Knowledge Base



All the pages which are designed to simulate the conventional information system include the same header and footer part in the very top and very bottom of the page. And the main content was put into a form in the middle of the page. This is a very typical layout because in this way, it's easy for system administrator to maintain the navigation bar and other fixed content on the top and bottom of all the pages.

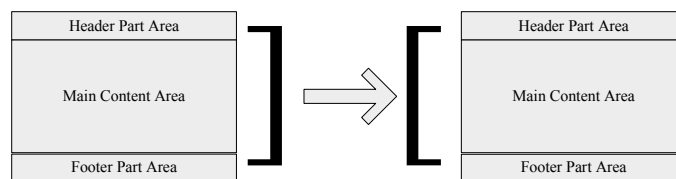


Figure 2. Footer and header layout

All the pages which are designed to simulate the Ajax optimized information system also include the same header and footer part in the very top and very bottom of the page. But the difference between this and the former one is that all the main content was loaded and unloaded dynamically using the Ajax way, that means loading and unloading the main content will not effect the data on the header part and the footer part.

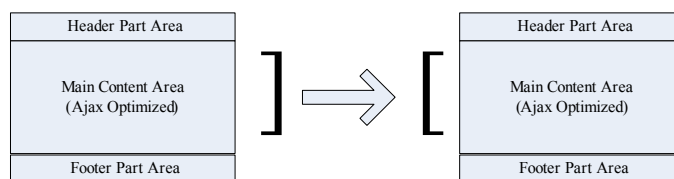


Figure 3. Simulating the Ajax Optimized Information System

Except this, all the content and database are the same, which helps to minimize the effect that different programming method brings to the coming simulations.

5. SIMULATION RESULT

In the simulation, I use Microsoft Web Application Stress Tool (MFAST) [8] as my simulation engine. I choice MFAST because it is designed to realistically simulate multiple browsers requesting pages from a web site. Use this tool I can gather performance and stability information about the web application of the two kinds of information systems. This tool simulates a large number of requests with a relatively small number of client machines. It creates an environment that is as close to production as possible.

Simulation environment is:

Intel Pentium M 1700MHZ with 768MB Memory running Windows XP SP2 and Internet Information Service.

Simulation Results: (All the test were tested for three times, I use the average of the three as the final result)

Table 1. Simulation Result

Scene	Total Hits	Bytes Sent	Bytes Received		
			Bytes/Hit	time	
	KB				KB
Scene1(1st)/C	1494	697.34	1778.5	1.657	
Scene1(2nd)/C	1475	688.02	1754.34	1.656	
Scene1(3rd)/C	1448	675.42	1722.2	1.656	
Average			1.656		
Scene1(1st)/A	1523	553.3	1548.15	1.380	
Scene1(2nd)/A	1454	528.24	1477.99	1.380	
Scene1(3rd)/A	1520	552.21	1545.1	1.380	
Average			1.380		
Scene2(1st)/C	1460	737.62	1611.27	1.609	
Scene2(2nd)/C	1392	703.25	1536.35	1.609	
Scene2(3rd)/C	1510	763.38	1667.6	1.610	
Average			1.609		
Scene2(1st)/A	1443	527.97	1203.48	1.200	
Scene2(2nd)/A	1469	537.52	1225.34	1.200	
Scene2(3rd)/A	1399	512.27	1167.77	1.201	
Average			1.200		
Scene3(1st)/C	18343	8365.85	15411.24	1.296	
Scene3(2nd)/C	18584	8475.78	15613.73	1.296	
Scene3(3rd)/C	18988	8659.54	15952.29	1.296	
Average			1.296		
Scene3(1st)/A	19701	6772.62	11072.94	0.906	
Scene3(2nd)/A	20170	6933.81	11336.3	0.906	
Scene3(3rd)/A	20146	6925.56	11322.81	0.906	

Average	0.906
---------	-------

Table 2. Simulation Result (Continue)

Scene	TTFB	TTLB	Page Load Time	Request/Second	
	ms				ms
Scene1(1st)/C	37.573	37.640		0.067	
Scene1(2nd)/C	37.957	38.013		0.057	
Scene1(3rd)/C	38.943	38.993		0.050	
Average	38.158	38.216		0.058	
Scene1(1st)/A	37.013	37.043		0.030	
Scene1(2nd)/A	38.753	38.807		0.053	
Scene1(3rd)/A	36.900	36.940		0.040	
Average	37.556	37.597		0.041	
Scene2(1st)/C	21.463	38.603		17.140	
Scene2(2nd)/C	22.263	40.700		18.437	
Scene2(3rd)/C	20.997	37.360		16.363	
Average	21.574	38.888		17.313	
Scene2(1st)/A	39.030	39.070		0.040	
Scene2(2nd)/A	38.290	38.330		0.040	
Scene2(3rd)/A	40.340	40.377		0.037	
Average	39.220	39.259		0.039	
Scene3(1st)/C	1.480	1.507		0.027	247.4
Scene3(2nd)/C	1.440	1.467		0.027	238.23
Scene3(3rd)/C	1.403	1.420		0.017	244.35
Average	1.441	1.464		0.023	243.33
Scene3(1st)/A	1.330	1.343		0.013	275.32
Scene3(2nd)/A	1.295	1.303		0.008	305.67
Scene3(3rd)/A	1.315	1.330		0.015	289.81
Average	1.313	1.325		0.012	290.27
Data Efficiency Index					0.759
Page Display Efficiency Index					1.238
Page Load Efficiency Index					0.405
System Performance Efficiency Index					1.193

6. RESULT ANALYSIS AND CONCLUSIONS

From the simulation results, I can get the following conclusion:

Compare with the conventional information system (hereinafter referred to as MIS C), the Ajax optimized system (hereinafter referred to as MIS A) enhanced data transfer efficiency remarkably. This enhancement was not caused by the increase of network bandwidth; however, it is because the MIS A saved near 25% of total bytes that system transferred. Meanwhile, MIS A didn't need to transfer the header part and footer part, the same content on all of the pages. But for MIS C, this is a necessity to do on every page. For this simulation, I only put a few texts on the header and footer parts without any picture, but for the real MIS people always use pictures to make the navigation bar and logo, so I believe MIS A will have much better data efficiency than MIS C in real.

Page Display Efficiency Index shows that MIS C has better display effect for users than MIS A, which means the content (partial content, not the whole content of the page) on MIS C comes out quicker than MIS A. The reason for this is Ajax works in an asynchronous way. So in some simulations which Ajax needs to get data from database, the system has to wait until the database finish its work.

Although MIS A has worse result for Page Display Efficiency Index, it also has its advantages. In simulation for Page Loading Efficiency Index, MIS A have excellent result. In average, it only spend 40% of time that MIS C need to load the complete content, which means MIS C will display the partial content very quickly, but visitor has to wait for longer time to get the whole page loaded. Personally, I think the result shown by Page Display efficiency Index is more important because the reason people visit the website is to get the full content, not the partial one. So MIS A will have a better experience for the visitors.

For the stress test part, I simulated numbers of clients to visit the server. Let the server became exhausted in order to get the maximal workload the server can handle. So under this maximal workload, we will know in which architecture (Conventional or Ajax), system can finish more requests from the clients. From the simulation data, we could see, MIS A handles nearly 20% more requests than MIS C. That means, using the same hardware, MIS A can deal with more customers at the same time.

So from these simulation data we could draw this conclusion: the Ajax technology indeed optimized the system. Management information system in Ajax architecture not only reduced the consumption of server resource, enhanced the efficiency and stability of the servers, but also improved the user experience by lessening the page loading time. For the enterprise who will build the new information system, it is reasonable to involve the Ajax in the system. Because this won't increase the cost for building the system, but it will reduce the cost for purchasing hardware and its maintenance. For the enterprise that already has information system, it should compare the cost for maintaining the hardware and the cost for migrating the system from conventional one to Ajax one. Because reduce the total cost of ownership and increase the performance and efficiency is our original goal of deploying Ajax technology.

REFERENCES

1. J.J. Garrett, *AJAX: A New Approach to Web Applications*, Adaptive Path (2005).
<http://www.adaptivepath.com/publications/essays/archives/000385.php> (Accessed May 7, 2007).
2. U. Wahli, *WebSphere Studio Application Developer Version 5 Programming Guide*, IBM Redbooks, International Business Machines Corporation (2003).
3. S. Tu, M. Xiong, X Chen, and W. Du, *Microsoft Office SharePoint Portal Server 2003 Advanced Guide* (Publishing House of Electronics Industry: Beijing, China, 2006).
4. J. Gehlman, B. Galbraith, and D. Almaer, *Pragmatic Ajax: A Web 2.0 Primer* (Pragmatic Bookshelf: Lewisville, TX, 2006).
5. D. Crane, E. Pascarello, and D. James, *Ajax in Action* (Manning Publications: Greenwich, CT, 2005).
6. Z. Ke, *Ajax Development Brief---- Concept, Example and Framework* (Publishing House of Electronics Industry: Beijing, Beijing, 2006).
7. M. Chen and M. Shen, *ASP technology and dynamic web page development* (Tsinghua University Press: Beijing, Beijing, 2007).
8. FECIT Product Research & Development Center, *Practical Software Testing Method and Application* (Publishing House of Electronics Industry: Beijing, China, 2003).