access ON DEMAND

WHITE PAPER

Benefits of Citrix NetScaler for Ajax Applications

CiTRIX®

Table of Contents

Ajax and Web Applications

Ajax is emerging as the preferred development approach for the next generation of Web-based applications. Correctly implemented, Ajax frees the end user from the clunky HTML page-refresh interaction style commonly associated with browser-based applications, and delivers an experience much more like the rich client of a traditional client/server application. Yet Ajax-based applications are still built using Web standards. Thus, they are generally cross-platform and cross-browser, and are ubiquitously delivered via a Web browser (albeit with many of the same types of caveats affecting other Web applications). Therefore, Ajax is attractive not only to developers, but more importantly to businesses that rely upon the positive user experience and productivity their applications deliver to employees and customers to achieve business goals. Indeed, this is why Ajax development is used to build the Web applications of many of the emerging “Web 2.0” businesses.

What is Ajax?

“Ajax” stands for “asynchronous JavaScript and XML.” The term refers not to a specific technology or standard, but to an evolving design approach for creating richer and more interactive Web applications. Whereas traditional Web applications are composed of a series of HTML pages that must be reloaded if content is changed, Ajax applications rely upon client-side JavaScript code to asynchronously call the server to fetch additional information. The JavaScript code then takes the server’s response and updates the existing HTML page on the client. The net effect is that developers are able to create Web clients that look and behave much more like traditional client/server applications.[1]

While there are certain technologies — some fairly recent — that are commonly associated with Ajax (e.g., JavaScript, XML, the DOM[2], the XMLHttpRequest object) there is no strict definition of an Ajax application; there is no standard, specification or governing body that defines whether an application is Ajax or not. In fact, many commonly cited Ajax applications do not use XML or XMLHttpRequest at all. Ajax applications span a continuum of development practices and technologies. A simple example is an application that uses hidden frames to make requests back to a server and then receives HTML content in return. A more advanced Ajax application will use JavaScript to make SOAP[3] requests of an application; the application returns XML documents that are then transformed by client-side JavaScript using XSL-T. The results are then handed off to the user interface via the DOM to be rendered by the browser (see Figure 1).

[1] For more information on Ajax, the seminal paper that coined the term is available at http://www.adaptivepath.com/publications/essays/archives/000385.php. A highly influential piece from Apple on the XMLHttpRequest object is at http://developer.apple.com/internet/webcontent/xmlhttpreq.html. Sun has several good introductory documents on Ajax available at http://developers.sun.com/ajax/.

[2] Document Object Model: W3C standard defining the tree structure of HTML or XML documents. It enables individual elements of a page to be updated independently.

[3] W3C standard transport protocol for exchanging XML messages over a network.

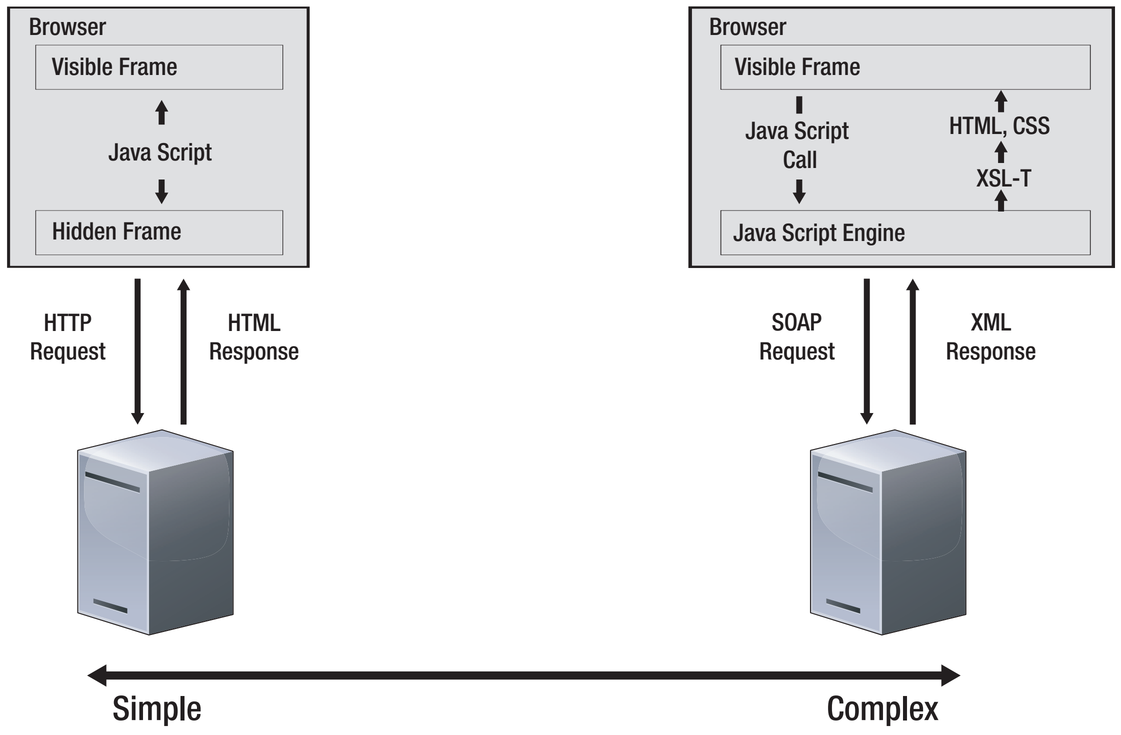

Figure 1. Ajax Application Continuum

Ajax and Citrix NetScaler

To understand how Citrix® NetScaler® works with an Ajax application, it is important to remember that Ajax is not a technology in and of itself. Rather, Ajax is a design approach for how various Web technologies are integrated to deliver a better client experience. Just as it is dangerous to make assumptions about the behavior of "standard" Web applications, it is dangerous to make assumptions regarding the behavior of Ajax applications. However, since Ajax applications are built using the same foundation as standard Web applications, the techniques that Citrix NetScaler uses to optimize standard Web applications are applicable to Ajax applications. Indeed, since there is no strict definition of an Ajax application, and because Ajax and non-Ajax applications will co-exist for some time, a common infrastructure supporting both Ajax and non-Ajax applications is desirable.

The Citrix NetScaler Application Switch:

- Improves data center efficiency
- Enhances application availability
- Accelerates application performance for an improved user experience

These benefits are achieved via the integration of:

- Patented Citrix® Request Switching® technology to optimize TCP protocol-handling operations and offload them from the Web servers
- Static and dynamic content caching to offload redundant content serving from applications
- Content compression to speed data transmission and reduce network bandwidth requirements
- L4-7 DoS/DDoS defenses to provide application availability in the face of these attacks
- L4-7 load balancing to evenly distribute workload and provide high availability/business continuity in the event of Web server failure

Improving Data Center Efficiency

The asynchronous nature of Ajax enables, and even encourages, developers to create applications that incrementally request small amounts of information rather than refreshing large amounts of information at one time. Indeed, Ajax client requests can look more like function calls (albeit asynchronous function calls) than traditional page requests. The repercussions are that an Ajax server is likely to see far more requests, even though the amount of information it is serving for each request is likely to be smaller. Therefore optimizing connection/session handling is critical for Ajax applications, and efficiently managing connection/session handling is the key to driving down hosting costs.

NetScaler Request Switching and static and dynamic content caching offload expensive processing tasks from general-purpose Web and application servers to purpose-built NetScaler Application Switches. It is not uncommon for NetScaler to reduce a Web application's hosting resource requirements by 50% or more. While Citrix has invested heavily in optimizing the offload functionality of NetScaler, it is built around Web standards such as GZIP, HTTP, and TCP. While Ajax applications differ from standard Web applications in their architecture, they are still at heart built upon these same standards, and Request Switching and caching will significantly reduce the data center and network bandwidth requirements of an Ajax application.

Request Switching and Ajax

The NetScaler Request Switching[4] engine manages client and server connections separately from one another, yet maintains the appearance of an end-to-end connection. By arbitrating client and server connection handling, NetScaler is able to:

- Multiplex multiple client connections through a single server connection. This frees servers from having to perform relatively expensive TCP connection processing and allows them to concentrate on serving content.
- Buffer server responses, insulating the server from widely variable and often unreliable WAN/Internet connections. This allows the server to respond to client requests at LAN speed, enabling it to complete a request and move on to the next at a faster rate.

[4] Please refer to the Citrix NetScaler white paper,"The Next Generation of Traffic Management," for more details on NetScaler Request Switching.

- Implement clientkeepalive to maintain browser TCP sessions. Clientkeepalive was originally used in conjunction with HTTP1.0 servers, and most Ajax applications will use HTTP1.1. However, due to the relatively large number of requests associated with most Ajax applications, the application or server may arbitrarily close client TCP sessions in an attempt to prevent itself from becoming overloaded. Citrix NetScaler maintains the client TCP sessions independent from the server infrastructure, preventing servers from arbitrarily closing established client TCP sessions and thus minimizing unnecessary connection set-up times.

CACHING AND AJAX

As with a standard Web application, caching content on the NetScaler switch can offload significant processing from the back-end Web and application servers[5]. Given the large size of the first page of many Ajax applications[6], caching just this initial page is likely to free up significant server resources. Once the initial page has loaded, the extent to which caching offloads content from subsequent requests is application-dependent. Any Ajax application that serves up the same content (whether this content is static or dynamically generated) to a large number of users will benefit from caching.

Enhancing Ajax Application Availability

Fundamental to the design of many Ajax applications is the concept that the client pre-fetches data it believes the user will eventually request. Since the client-to-server interactions of an Ajax application are asynchronous, it can conduct these pre-fetches without impacting how the user interacts with the application. And, if the server responds to a pre-fetch before the user actually requests the information, it will appear to the user that the application responded to his/her request instantaneously. However, overly aggressive pre-fetching can also subject servers to unnecessary load. Also, if an Ajax application is upgraded and this upgrade increases how much information the application pre-fetches, this will expose the servers to additional and possibly unanticipated load, impacting overall application performance.

To provide for continuous availability of Web applications, NetScaler provides L4 network load balancing, L7 content switching, global server load balancing (GSLB), surge protection, DoS/DDoS defense and priority queuing. The general use cases of these functions are as applicable to an Ajax application (i.e., using GSLB to direct a user to the geographically closest server farm) as they are for a standard application. Indeed, one example of using NetScaler's content switching is to direct traffic between the Ajax and non-Ajax version of the same application, depending upon what browser and browser version the user is using[7]. However, the asynchronous nature of an Ajax application can create certain scenarios not encountered in standard Web applications, where NetScaler surge protection, priority queuing and L7 DDoS defenses offer particular benefits.

[5] Please see the Citrix NetScaler white paper, "Optimizing Web Applications," for more details on static and dynamic content caching w/NetScaler

[6] The first page of an Ajax application typically is composed of large amounts of JavaScript, HTML/DHTML, CSS and/or XML/XSL-T

[7] Currently, Ajax applications require IE6 or the latest versions of Mozilla or Netscape. Therefore, many organizations have one version of an application that is built using Ajax, and another standard version of the application for users that have not upgraded to the latest browser versions.

Surge Protection, Priority Queuing and Ajax

Surge protection buffers client requests to prevent servers from becoming overloaded, and priority queuing is used to prioritize incoming requests, depending upon a defined priority policy. Surge protection and priority queuing are both used to insulate applications from intermittent usage spikes. General usage spikes are an issue for Ajax and non-Ajax applications alike, but surge protection and priority queuing are also useful for managing repercussions of the asynchronous nature of Ajax applications.

NetScaler Surge Protection will buffer the application from an overload of asynchronous pre-fetch requests. At the same time, priority queuing can be used to define policies prioritizing requests within this buffered pool. For example, surge protection can be used to prevent an e-commerce site from becoming completely overwhelmed, while priority queuing can be used to ensure that a certain types of customers (e.g., “gold” customers, customers with items in their shopping baskets) are given priority.

DDoS Defenses and Ajax

The server side of an Ajax application will be designed to handle asynchronous requests. While this asynchronous design improves the application’s ability to scale the processing of requests from multiple clients, it also opens up additional vectors for DoS/DDoS attacks, especially at Layer 7. This may be exacerbated by the fact that since so much of the application is exposed on the client and JavaScript code, Ajax applications are highly susceptible to reverse engineering. While issues associated with client-side JavaScript are outside the scope of the NetScaler product, NetScaler’s L4-7 DDoS defenses will effectively isolate an Ajax application from many DDoS attacks.

Accelerating End-user Performance

Citrix NetScaler compression and content caching can certainly improve Ajax application response times. This will be most readily apparent while loading the first Web page of an Ajax application, since the initial pages are generally quite large given the large amount of JavaScript, CSS, XSL-T, etc., they contain. Since these types of content are all highly compressible, Citrix® AppCompress™[8] can greatly reduce page load times (and also save on network bandwidth requirements).

Content caching and compression can also improve response time for subsequent interactions between an Ajax client and the server. However, given the asynchronous nature of many Ajax applications, how much of this improvement will be perceptible to the user will depend upon the efficacy of the application’s design. If pre-fetching is well-implemented, the user may not realize that individual requests are being fulfilled faster, since data presentation seems instantaneous regardless. However, an Ajax application that does not effectively leverage pre-fetch will benefit from NetScaler acceleration functionality.

Also, it is important to remember that much of the same functionality that accelerates applications (i.e., caching, request switching) also offloads and thus improves data center efficiency. An interesting side benefit of this offload is that, given a fixed amount of time and server resources, NetScaler will enable an Ajax application to fulfill more

[8] Please see the Citrix NetScaler white paper, “Optimizing Web Applications,” for more details on Citrix compression technology.

pre-fetches. This can result in the perception that the application is responding faster since all the pre-fetches the client requests will be fulfilled in less time, reducing the likelihood that the user will click on an element that hasn't yet been pre-fetched and thus have to wait for the request to be fulfilled.

Summary

While the asynchronous nature of Ajax applications makes them fundamentally different from standard Web applications, Ajax applications are still built upon standard Web infrastructure. Moreover, this asynchronous behavior does not obviate the benefits of solutions like the NetScaler Application Switch. Indeed, this asynchronous behavior can make NetScaler's Request Switching and high-availability functionality more important to an Ajax application than to a standard Web application. Finally, NetScaler will speed up the initial load of an Ajax application and help ensure that the perceived user experience benefits of asynchronous communication are not thwarted by network and server delays.

CITRIX®

Best Access Experience. Anytime. Anywhere.

About Citrix: Citrix Systems, Inc. (Nasdaq:CTXS) is the global leader and the most trusted name in application delivery infrastructure. More than 180,000 organizations worldwide rely on Citrix to deliver any application to users anywhere with the best performance, highest security and lowest cost. Citrix customers include 100% of the *Fortune* 100 companies and 98% of the *Fortune* Global 500, as well as hundreds of thousands of small businesses and prosumers. Citrix has approximately 6,200 channel and alliance partners in more than 100 countries. Annual revenue in 2005 was $909M.

PDF/1106

Citrix Worldwide

WORLDWIDE HEADQUARTERS

Citrix Systems, Inc.
851 West Cypress Creek Road
Fort Lauderdale, FL 33309 USA
Tel: +1 (800) 393 1888
Tel: +1 (954) 267 3000

EUROPEAN HEADQUARTERS

Citrix Systems International GmbH
Rheinweg 9
8200 Schaffhausen
Switzerland
Tel: +41 (52) 635 7700

ASIA PACIFIC HEADQUARTERS

Citrix Systems Hong Kong Ltd.
Suite 3201, 32nd Floor
One International Finance Centre
1 Harbour View Street
Central
Hong Kong
Tel: +852 2100 5000

CITRIX ONLINE DIVISION

5385 Hollister Avenue
Santa Barbara, CA 93111
Tel: +1 (805) 690 6400

www.citrix.com