

The Solution of Web Font-end Performance Optimization

Bangzhong Cao¹,Minyong Shi²,Chunfang Li³
School of Computer, Faculty of Science and Engineeringline
Communication University of China
Beijing,China

Abstract—With the rapid development of the Internet, people have a higher demand for the performance experience of the Web site. To improve page utilization and user interaction experience, web page capacity becomes bigger and bigger and use a lot of JavaScript code and its framework technology in web pages which makes web pages more expressive and richer.^[1] However, it also affects the resolution efficiency of the Web page in the browser, and directly delays the loading time of the Web page. To improve the user experience, web optimization becomes extremely necessary. Web site optimization is divided into front-end optimization and back-end optimization of two ways. We found that only 10%~20% of the time was used to get HTML documents from the Web server to the front-end browsers and the 80%~90% of the time was used in response to the front end^[2]. Front-end optimization requires less time and resources, and modifying a small number of programs can quickly improve the speed of your site^[3]. This article focuses on the front-end, from five aspects of the reduction of HTTP requests, page element optimization, Client caching optimization, DNS optimization and compression technology to propose some optimization methods to achieve the performance of the Web site to improve without changing the database and background code^[4].

Keywords—Web front-end; front-end optimization; response speed

I. INTRODUCTION

The client usually requires an HTTP protocol and a TCP protocol to complete the communication.^[5] The HTTP protocol is the web application layer protocol, which is the core of the web. Processing data generally takes a long time. In the process of HTTP transmission of messages, HTTP delay is caused by TCP network latency unless the browser concurrency limit, or server overloading.^[7] Here's a detailed explanation of the steps contained in the HTTP protocol's client/server request Response mechanism: (1) Enter the Access URL in the address bar of the browser, and the browser calls the domain name resolution library for the IP address^[6]. (2) The client browser sends the request URL to the server. (3) The server side receives and processes requests sent by the browser to generate HTML. (4) server-side response browser request^[8]. (5) The client browser accepts and parses the data returned by the browser. (6) The server side returns the related resource files to the client. (7) browser-side parsing related resource files, rendering pages from top to bottom^[9]. (8) Asynchronous communication.

We found that when the user gets a page, the waiting time is: (1) Domain name resolution time, The time required for client to resolve domain name to find server IP. (2) The time

taken by TCP to establish a connection : the time required for TCP to connect three times when a client establishes a connection with the server. (3) The response time for HTTP requests : The time the HTTP request was sent and the time the server responds to HTTP requests. (4) The time that the requested content is downloaded to the client, after the server responds to the request. (5) The time that the client resolves the content requested from the server side.

This paper focuses on reducing the delivery time of server response and the time of client resolution response to introduce some optimization methods of Web front-end.

II. REDUCE HTTP REQUESTS

Each request is cost-inclusive, which containing both time and resource costs. A complete request requires a "lengthy" and complex process of DNS addressing, connecting to the server, sending data, waiting for the server to respond, and receiving data. The Web page contains many other elements, in addition to the HTML code that includes JavaScript, CSS, and back-ground images. To render the entire page, you need to down-load all the components on the page, and each component needs to send an HTTP request to the server. The large number of requests generated by these components is the main reason to affect the speed of the website. The easiest way to improve response time is to reduce the number of components. But simply reducing the number of components reduces the richness of the Web page, thereby reducing the user experience. The following two scenarios reduce HTTP requests:

A. CSS Sprite Technology

CSS Sprites is a kind of image integration technology, it combines some scattered images on the page into a large picture, using CSS properties to navigate to the location of the desired picture. This approach converts many of the HTTP requests corresponding to a large number of scattered images into an HTTP request corresponding to a larger image, greatly reducing the number of HTTP requests, and the image is loaded once, achieving a visually better user experience.

B. Merging scripts and style sheets

Most Web sites use JavaScript and cascading style sheets. There are two options for Web page design: One is to put JavaScript code and cascading style sheets into an HTML page file, and the other is to write separate JS files and CSS files, and then refer to them in an HTML file. In general, good

performance can be achieved by using the second method. However, multiple JS files and style sheets increase the number of HTTP requests. Merging style sheets and JS script files can effectively reduce the number of HTTP requests and thus reduce the time to download Web pages.

III. OPTIMIZATION OF PAGE ELEMENTS

A. Optimization of HTML

HTML language is the most important language that makes up a Web document. HTML files in a Web page are descriptive text that consists of many HTML commands. The main methods of HTML optimization include the following: (1) The use of "DIV + CSS" mode layout can make the code streamlined, the structure is clear, make the page smaller. (2) CSS and JavaScript files outside the chain. The outer chain is referenced by the CSS or JavaScript code in the relevant page. General HTML documents will always be refreshed when the page loads, the contents of the page will be reloaded, embedded, inline write the contents of the reload together. And when the CSS and JavaScript files outside the chain, you can cache the component, the page will be loaded again to significantly reduce the download component time. (3) Delete the HTML element redundant initial default properties, many HTML elements will have some initial default properties, so you can delete these redundant attributes.

B. Optimization of CSS

CSS style sheet determines the way the page is displayed, defining the location, size, and effect of the elements in the page, and plays an important role in the rendering of the web. CSS optimization of the main methods are:

(1) the CSS file on the head tag, so that the first step in the loading page to download and parse the CSS file, along with other resource files to achieve progressive rendering and display to enhance the page gradually rendering speed.

Many elements of the page, such as pictures, scripts, etc. are required to download the web page, and CSS style sheet for the components of the page provides its performance style, location and format information. Web pages in the loading page content is generally explained in the order of execution, that is, the page elements in accordance with its order in the page to download. If you place a reference to a CSS file at the bottom of an HTML document, it will cause the CSS code to be downloaded and drawn after all the relevant files and page source code have been parsed. This will cause the page to be loaded with a "white screen" or no style of confusion. So under normal circumstances, the style sheet CSS should be placed at the top of the page.

(2) Write efficient CSS selectors

CSS selector in accordance with the efficiency from low to high: pseudo-class and pseudo-elements, attribute selector, wildcard selector, descendants selector, sub-selector, adjacent brother selector, type selector, class selector, ID selector. There are several general rules for writing efficient selectors:

- Avoid the use of rules, in addition to the traditional sense of the wildcard selector, the brothers selector, sub-selector, descendants of the selector and attribute

selector should be avoided, as far as possible with ID, class and tag selector to replace.

- Do not limit the ID selector. For example `li#list` is not necessary.
- Avoid using qualified class selectors, such as `.div-menu` is more efficient than `div.menu`.
- Try to refine the rules as much as possible, do not try to write long selectors like `ul li span`, it is best to create a new class, such as `.list_span`, add to the appropriate elements.
- Avoid using the offspring selector, usually dealing with the cost of the offspring selector is the highest, you can use the sub-selector instead.
- Avoid using tags - sub-selectors, such as `#wrap > div > a`. This tag-based child selector can be used to define the `.wrap_link` class name instead.
- Replace the sub-selector with a specific class as much as possible.
- Rely on inheritance, to avoid duplication, you can also reduce the size of CSS files.

(3) Avoid using expressions in CSS

The fatal thing about the CSS expression is that it performs more often than you think. Ideally, we want this expression to be executed when the page is loaded or refreshed. Contrary to expectations, as soon as the page scrolls, it repeats itself, even when the mouse moves. You know, 80% of users like to use the mouse to move on the page to help them read, or simply move aimlessly. Then imagine how much the mouse would move when the user was reading it. Because the CSS expression executes when the mouse is moving, so the CSS expression is executed thousands of times, even thousands of times, just looking at this page. This is a disaster for the user experience.

C. Optimization of JAVASCRIPT

Now, Web sites are mostly driven by a lot of JavaScript code, most of the site's user interface interaction and dynamic effects are implemented by JavaScript. Therefore, the optimization of performance not only to shorten the page load time, but also to improve the user in the page when the functional operation of the response speed. Write efficient JavaScript code to enable it to respond quickly to user actions. The main methods of optimizing JavaScript are:

(1) JavaScript script table at the bottom of the page.

The browser's download and rendering order are top-down, and at the same time. As the order of the implementation of the browser, no matter where the script placed in the place, will block the subsequent content and component download, you need to load the script after the completion of the contents will be displayed later. When the script is at the top of the page, loading the script causes the page to stop rendering and rendering blank. Putting the script at the bottom of the page is better than the top of the page, and the script is more effective.

(2) use local variables whenever possible.

In the JavaScript language, local variables are the fastest and identifiable identifiers.

(3) avoid the growth of the role of the chain.

Unnecessary growth of the scope chain will lead to slower code execution.

(4) Use a timer to avoid running scripts that are too long.

JavaScript is a single-threaded language, so in a time period, each window or tab can only execute a script. When the script is too long, in the implementation of the user interaction will lead to interruption. You can set the setTimeout function to set the time to delay the execution of the function.

(1) Careful use of with.

with(obj){ p = 1};The block of code behavior is actually modify the code block in the execution environment, obj will be placed in the front of the scope chain, in the with block access to non local variables are first to start the search from the obj, if not followed by the scope chain to find, so using with equivalent increase the scope chain length. Each time the scope chain is consumed, the long chain of domains leads to a decrease in lookup performance. Therefore, unless you are sure that only the attributes in obj are accessed in the with code, you can use with with caution, and instead you can use the local variables to cache the attributes you want to access.

(2) Do not use '+' for string concatenation.

In Javascript, the "+" number is used to concatenate strings, the efficiency is relatively low, because each time the run will open up new memory and generate new string variables, and then assign the result of the concatenation to the new variable. In contrast, a more efficient approach is to use the array join method, where the string to be spliced is placed in the array, and finally calls its join method to get the result. However, the use of arrays also has some overhead, so you can consider using this method when you need to concatenate more strings.

IV. THE CLIENT CACHE OPTIMIZATION

As the server and the client there is a bottleneck between the flow, and the use of cache technology can greatly improve the speed of data read. Caching can reduce the amount of data exchanged between the client and the server, thereby improving the performance of the Web system. Caching some of the data on a Web page is to store the result of the first request or the intermediate object, and at the time of the second request, the final data can be read directly from the cache and returned to the client browser. Use the client's cache to optimize Web performance and reduce user latency. But the cache for the first visit to the page does not reduce the response time. Improvements to performance will be reflected when you visit the same page again.

This section provides two ways to optimize the client cache: CDN technology and optimization methods for cookies.

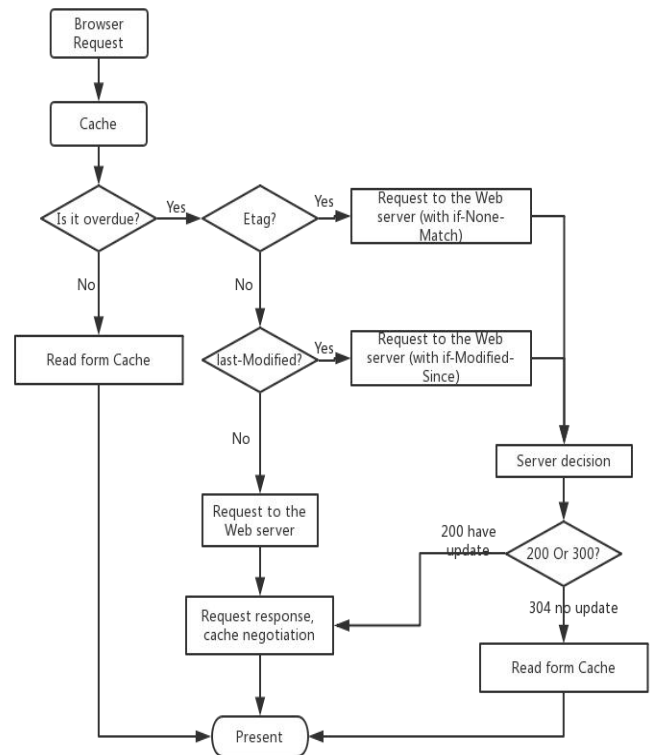
A. Add the Expires header

The first time you access a Web page, you will have a lot of HTTP requests, and you can use the Expires header to cache the components after the initial access and control the cache expiration date. The Web server uses the Expires header to tell the Web client that as long as the component does not expire, the browser will use the cache version, thus avoiding

unnecessary HTTP requests. Cache-Control has a higher priority than Expires, and uses Cache-Control to overcome the limitation of the Expires header to solve the problem that the client server time is not uniform.

B. Use Etag

Etag is a physical tag that is used to confirm that the cached component is valid. The components downloaded by the browser are stored in the browser's cache. In subsequent page views, if the cached component is valid, the browser will read it from the local disk to avoid generating an HTTP request. If the cached component expires, the browser must use a GET request to determine whether the components are still valid before reusing the component.



Flowchart Explanation: The first request, whether it is static files or other files, are read from the server there, then there is no cache. And so the first request finished, the browser has a cache, and then the entire loading process is completely different. The browser second request, the first cache will be read, and then determine whether the cache expired, if not expired, read the cache directly. Otherwise, determine whether the browser returns the presence of Etag. If so, the browser will send a request header with an If-None-Match as the server to compare with the Etag returned by the server. If if-None-Match and Etag are equal, the cache is not updated. The server returns 304, and the browser continues to read the contents from the cache. If the if-None-Match and Etag do not wait, the server returns 200 and the browser needs to retrieve the content from the server again. If there is no Etag in the server's return

message, determine whether there is Last-Modified in the browser's return message. If so, the browser will send an if-Modified-Since request header like the server. Then if-Modified-Since value will be compared with the value of Last-Modified, if if-Modified-Since value is greater than or equal to Last-Modified, the server returns 304, the file is not updated, you can directly read the cache. If the value of if-Modified-Since is less than Last-Modified. It means that the browser cache is not up to date, need to re-read from the server. If the server returns the header information neither Etag nor Last-Modified, the cache has expired and the server is crawling again.

C. Cookie optimization

In the process of user access to the site, the server will generate a text file and sent to the browser, this is the cookie. In the browser to the server to send the request process, the cookie will be passed between the two. Cookies are not associated with a particular page, but are associated with the entire site. As a result, the browser and the server exchange cookie information when requesting any page in the site. The biggest drawback of cookies is that every time an HTTP request is brought down, this results in a high cost of bandwidth on the server and an impact on the user's access.

Cookie optimization program:

- Remove unnecessary cookies. If the page does not require a cookie, it will be completely banned.
- Minimize the size of the cookie. Because cookies are sent to the server through HTTP requests when accessing the resources under the corresponding domain name, reducing the size of the cookie can reduce the size of the HTTP request packets and improve the response speed.
- Set the appropriate expiration time, because the longer expiration time can improve the response speed. To add an expiration time to the cookie, the cookie information will be stored on the hard disk, even if the browser exits the cookie. As long as the cookie is not cleared and is still expired, the cookie is sent to the server when the corresponding domain name is accessed.
- Reduce the use of cookies by using different domains. Cookies are sent to the server via HTTP requests when they access the resources under the domain name, but when accessing some resources, such as js, css, and pictures, cookies are redundant in most cases, and different domains can be used to store these static resources, So that access to these resources will not send redundant cookies, thereby increasing the response speed.

V. DNS OPTIMIZATION

When the DNS lookup is cached, the local DNS server will first ask its cache record after a request, and do not need DNS lookup for a short period of time. If the cache is available, it will return directly. Only when this record is discarded by the cache is it necessary to re-initiate the request and take extra time.

A. Add the Expires header

When the DNS lookup is cached, the local DNS server will first ask its cache record after a request, and do not need DNS lookup for a short period of time. If the cache is available, it will return directly. Only when this record is discarded by the cache is it necessary to re-initiate the request and take extra time.

B. Reduce DNS lookup

Because DNS lookups take time, and they are usually cached for a certain amount of time, you should minimize the number of DNS lookups. Reduce the number of DNS lookups, the best way is to put all the content resources are placed under the same domain (Domain) below, so access the entire site only need to conduct a DNS lookup, which can improve performance. But each domain has only a certain amount of parallelism, which will lead to the emergence of resources when the queue phenomenon, it will reduce performance. It is recommended that you use at least 2 domains, but no more than 4 domains, to provide resources on a website. This will ensure that neither the DNS lookup nor the parallel download is allowed.

VI. COMPRESSION TECHNOLOGY

Web page for the first time the load is in the browser without the cache from the server through the network to download the process of the client. There are many factors that affect the speed of the page, the page size is an important factor affecting the loading speed of the page. When the requested page is small, the amount of data is small and the time it takes to return is reduced. Compression is the simplest technology to reduce page size.

A. Gzip compression

Gzip is an abbreviation for GNUzip, and the Gzip encoding on the HTTP protocol is a technique used to improve the performance of Web applications. Gzip compression technology is currently the most widely used a compression technology, it is a GUN free software file compression program. Gzip is now the most widely used a compression technology, patent rights and other aspects without any restrictions, and has become the RFC standard. The Web client informs the server of the compressed file format type it supports through the Accept-Encoding header in the HTTP request. The Web server parses the request header, uses a method provided by the client to compress the response code, and uses the Content-Encoding response header to pass back to the client using the compression method it uses.

SUMMARY

The web front end is a relatively large part of the optimized space. This paper introduces the principle of front-end optimization, and introduces five front-end optimization methods from two aspects: server response time and client response time.

REFERENCES

- [1] LI Meng, JIANG Shan, HUANG Xing, and AN Li-peng “WEB optimization design base on front-end”, Wuhan Research Institute of Posts and Telecommunications, Wuhan 430000, China, May 2016.
- [2] LI Xiaofeng, “Performance Optimization of Web Front-End Engineering”, Human Resources Department, Kunming Shipbuilding Equipment Research Test Center, Kunming 650000, China, May 2015.
- [3] Zhou Lu, “Performance Study and Improment of Localstorage Based on Web Font-End ”, School of Computer Science and Technology, China, May 2014.
- [4] Wang Yun, “The Research on online Learning Platform of University Based on Web Font-End Technology”, Jiangsu Open University, Information Technology Office, Nanjing 210036, China, May 2015 .
- [5] Zhong Xunke, “Research on cross platform mobile Web application and hybrid application based on html5”, Software Department, Guangzhou Unversity 510900 ,China, July 2014.
- [6] LI Xin, “Research on development technology and optimization of Web font-end”, Beijing jiaotong University 061199 china, May 2015.
- [7] Song Guanmin, “The trends of Web font-end”, Shenyang Ligong University 110159 ,China, August 2016.
- [8] Zhu Jubao, “Research of mobile Web font-end performance and image’s optimization”, School of Computer Central China Normal University, May 2013.
- [9] Wang Yutao, “High-performance web server and implement of online tickets system”, School of Computer and Technology Donghua University, May 2015.