Performance optimization of agricultural products website front-end based on user experience

Shengyin Zhu 1, Xiangzhen He2* and Honzhi Yu3

1 Shengyin Zhu Key Laboratory of China’s Ethnic Languages and Information Technology of Ministry of Education Northwest Minzu University Lanzhou, Gansu 730000, China 87961192@qq.com
2 Xiangzhen He Key Laboratory of China’s Ethnic Languages and Information Technology of Ministry of Education Northwest Minzu University Lanzhou, Gansu 730000, China 5967148@qq.com
3 Hongzhi Yu Key Laboratory of China’s Ethnic Languages and Information Technology of Ministry of Education Northwest Minzu University Lanzhou, Gansu 730000, China yuhongzhi@hotmail.com
*Corresponding author’s e-mail: 5967148@qq.com

Abstract. With the rapid development of Internet technology, users have higher and higher requirements for the response speed of the website, which makes the performance optimization of the website become extremely important. How to solve this problem is the key to improving the user experience. In this paper, taking the characteristic agricultural product website as an example, we analyze the website front-end performance optimization in CSS optimization, page rendering, HTTP request reduction, Ajax caching and web caching, and offer the corresponding performance optimization methods.

1. Introduction
From the perspective of users, what users intuitively feel on the browser is the response speed of the website, that is, the response time is an important factor that directly affects the user experience. According to the survey, when the response time of page connection exceeds 1s, 5% of users choose to leave; when the response time of connection exceeds 3s, 17% quit; when the response time of connection exceeds 5s, 27% choose to give up. Some studies have shown that there is a positive correlation between website visit delay and user churn rate. Therefore, even if the website achieves better page vision and interaction effects, users will choose to leave and go to other websites because of long time of page loading or response, resulting in losing a large number of users. It is also mentioned in PULSE, a website measurement index, that if a product is frequently interrupted or visited very slowly, it cannot attract users. Therefore, this section introduces the front-end performance optimization scheme for the featured agricultural products website from the following aspects.

2. The design for performance optimization scheme
2.1 CSS optimization
Fine page layout needs to be achieved through a large number of CSS styles, but too many CSS codes will lead to bloated, messy state and uneasy management. You can use abbreviations to keep CSS files of reasonable size and reduce the size of CSS files.

2.2 Page rendering
Browser rendering process of the page: first load the HTML document to parse. After finding the referenced CSS file, send a request to start downloading the CSS file. When the CSS file has been downloaded, the browser starts rendering the page until the HTML tag is executed. During the rendering process, if the browser finds the script tag, it will execute the JavaScript immediately, which will affect the page rendering process and block the page. Therefore, CSS files should be downloaded before HTML files, and JS files should be loaded after HTML files.

2.3 Reduce HTTP requests

2.3.1. Resource consolidation. The main way to reduce HTTP requests is to merge CSS, JavaScript and pictures. The platform can reduce HTTP requests by merging files, merge JavaScript files and CSS files in the production environment, so that the number of browser requests can be reduced, the response speed of web pages can be improved, the waiting time of users can be reduced, and the user experience can be optimized.

2.3.2. HTML, JS, CSS.compression output. The web data stream output by the server to the browser client generally has more useless characters, including spaces, tabs, line breaks, comments, etc. By compressing HTML, JS, CSS and other files through Gzip, the web data package can be reduced, the data volume of communication transmission can be effectively decreased, and the page loading time can be shortened [1].

2.3.3. Reduce cookie transmission. Cookies on the browser side are usually sent to the server, which will greatly affect the data transmission. Therefore, unnecessary data should not be written into cookies so as to reduce the amount of data transmitted in the cookie and the size of the HTTP request. Static resources such as CSS can be accessed by different domain names to avoid sending cookies when requesting static resources and reduce the number of cookie transfers.

2.4 Delayed image loading
Another interpretation of delayed loading is lazy loading. When the user accesses the page, first the picture path of img element should be replaced with the picture of placeholder. When the user scrolls the scroll bar, the picture appears in the visible area of the browser, and then the information and the real path of the picture are displayed [2]. The steps for lazy image loading are as follows:

First, when setting the SRC attribute of img, assign the address of the placeholder image to the SRC attribute, and then give the address of the real image to the data-origin attribute. After the page is loaded, judge whether the user performs the scroll bar operation. If the operation and the picture position are within the user's vision, take out the value in the data-origin attribute and store it in the SRC attribute. If the user repeatedly performs the scroll bar operation, repeat the judgment according to the above steps.

After delayed loading is used, the page starts to load only a part of resources, and only when the user continues to browse later, the subsequent resources will be loaded [3], which can reduce or delay the number of HTTP requests, lessen the memory pressure, improve the loading speed, and give the user a better visual experience [4].

2.5 Using cached Ajax
Reasonable use of Ajax technology can speed up page response and reduce server pressure. From the perspective of users, it can provide users with instant feedback, and only let the page update locally
rather than redraw the page after the white screen, reducing the user's actual and psychological waiting time, and achieving a high level of user interaction [5]. When a traditional web page is submitted, the entire page will be reloaded, which will force users to wait for the entire page to load.

Ajax is a technology that starts from the client and interacts with the server, involving the client and the server. Ajax can interact with the server without refreshing the web page after the web page is loaded [6]. The principle of Ajax is simply to create an XMLHttpRequest object, send the request to the server through the Ajax engine; the server returns the data and receives it, and then use JavaScript to operate the DOM and update the page. The most important thing in the whole operation is to get the data in the server [7]. As shown in Figure 1, the user's request is sent indirectly through the Ajax engine rather than directly through the browser. At the same time, the Ajax engine also receives the data returned by the server, so it will not cause all the pages on the browser to refresh.

2.6 Web Caching

2.6.1. HTTP file cache. HTTP file caching is a caching mechanism of browser file level based on HTTP protocol. HTTP caching can make browsers read files locally when file caching is in effect. It can not only speed up page resource loading, but also save traffic. Figure 2 is the flow chart of caching judgment mechanism of browser HTTP file. By setting cache-control, HTML content can be cached to reduce requests to the server. When the page cache-control does not expire, the browser will read the content directly from the cache without making requests to the server. If it expires, it will send a request to the server for the next judgement.
2.6.2. Use CDN cache. The full name of CDN is Content Delivery Network, that is, content distribution network. Its purpose is to publish the content of the website to the "edge" of the network closest to the user by adding a new network architecture to the existing Internet, so that the user can get the required content nearby, solve the Internet network congestion, and improve the response speed of the user visiting the website. In terms of technology, it comprehensively solves the root cause of slow response of users to visit the website due to the small network bandwidth, large number of users' visits and uneven distribution of outlets [8]. The essence of CDN is still a cache---caching the data closest to the user and enables the user to obtain the data at the fastest speed, as shown in Figure 3. By using CDN, resources can be obtained nearby. If the resource requested by the browser already exists in CDN, it can be returned to the browser directly through CDN to respond with the shortest path. In this way, the response speed of user access can be improved, and the data center load pressure can be reduced [9].

![Figure 3. CDN mode pattern](image)

3. Performance test
After the optimization of the above performance scheme, under the test environment of Intel (R) core (TM) i5-3470 CPU 3.20ghz, 4G memory, Windows 7 operating system and Chrome browser, a test is conducted through the developer tool Timeline provided by chrome, and the data before and after the optimization is compared to verify the impact of the front-end performance improvement scheme on the page response speed. The test results are shown in Figure 4 and Figure 5.

In Figure 4 and Figure 5, the red parts are the response time and the number of requests before and after optimization. Before the performance optimization, the loading time of the website homepage took 6.61s, and the number of page HTTP requests was 226. After the optimization, the loading time of the website homepage took 3.58s, and the number of page HTTP requests was 131.

![Figure 4. Test results before the performance optimization](image)
Figure 5. Test results after the performance optimization. Figure 6 shows the page loading time of the website after adopting the cache optimization scheme. After cache optimization, page loading time is greatly reduced.

Table 1 shows the data comparison before and after optimization. It can be seen from the data in the table that after optimization, the page loading time is controlled within 3-4s, the loading time is almost halved, the page response speed is significantly improved, and the waiting time of users is reduced; the number of page requests is reduced by about 40%, and the compression rate of documents through Gzip tool is up to more than 80%, and the page loading load is greatly reduced. After the cache scheme is used, the page loading time is obviously reduced. It shows that the optimization scheme has a significant effect on improving the front-end performance of the website.

| Table 1  Performance correlation chart |
|----------------------------------------|
| before performance optimization | after performance optimization | efficiency promotion |
| first Page loading time (s) | 6.61 | 3.58 | 46% |
| Number of requests | 226 | 131 | 42% |
| File compression (bytes) | 221472 | 28079 | 87% |
| Page reloading time (s) | 3.58 | 1.42 | 60% |

4. Conclusion
Performance optimization is a very broad concept. The performance index of a website is generally measured by the time from the user's data acquisition to the actual data acquisition [10]. This paper mainly optimizes and tests the website from the foreground performance, achieves the expected goal, improves the page performance and promotes the user experience.

Reference:
[1] thunderbolt. Performance optimization of web application front end [J]. Mindong agricultural science and technology, 2017 (01): 29-30.
[2] Duan Xu, Huang Tongcheng. Optimization of listview control based on delay loading technology [J]. Software guide, 2015, 14 (11): 93-94.

[3] Dai Sheng, Zhu Lin. Research on website performance optimization based on Web server [J]. Information recording materials, 2018, 19 (06): 181-182.

[4] Song Xiaoyuan, Fang Qiang. Research and application of web front end performance optimization [J]. Information technology, 2016 (10): 198-202.

[5] Ji Huabin, Shang Ying, Li Zheng. Web application asynchronous request timing problem detection [J]. Computer science and exploration, 2018, 12 (01): 101-111.

[6] Traian Turc. Ajax technology for Internet of things [J]. Procedia manufacturing, 2019, 32.

[7] Ying Yu. Ajax based monitoring system stairs technology [a]. Science and Engineering Research center. Proceedings of 2016 3rd International Conference on advanced education technology and Management Science (aetms2016) [C]. Science and Engineering Research Center: Science and Engineering Research Center, 2016:3.

[8] Zhao Hong. CDN content distribution network technology principle [a]. Construction technology and management organizing committee. Paper collection of construction technology and management academic exchange meeting in August 2017 [C]. Construction technology and management Organizing Committee: Beijing Hengsheng Boya International Cultural Exchange Center, 2017:2.

[9] Zhou Qiang, Xie Jing, Zhao Huaming. Architecture research and solution of large websites [J]. Computer science, 2017, 44 (S1): 587-590.

[10] Chen Hongyan, Wan Junwei. Research and application of Web-based enterprise website performance optimization scheme [J]. Internet of things technology, 2018, 8 (02): 67-69-73.