

COMPRESSION OF CASCADING STYLE SHEET FILES

Inventor: Lixin Hao

BACKGROUND

5 Field

[0001] The present application relates to the field of computer technologies, and relates generally to a method and system for data compression for network communication.

Related Art

10 [0002] With the advancement of the computer and network technologies, various operations performed by users from different applications lead to extensive access to web services. For example, multiple users may concurrently shop on an e-commerce website via different applications, such as mobile applications running on different platforms as well as web-interfaces running on different browsers in different operating systems. A server hosting the web
15 services presents the corresponding shopping web page to different users, thereby facilitating the e-commerce service.

[0003] However, different applications concurrently accessing the system can cause an increasing number of user activities, consequently, an increasing amount of web pages are accessed from the server. A large number OF these web pages can be represented using a
20 Cascading Style Sheet (CSS). A CSS file is typically sent over a network along with a web page to provide the presentation style for the web page for a browser in a client device. The CSS file is typically in a text format. If a CSS file is compressed, the size of the CSS file becomes smaller. As a result, the compressed CSS file can be transmitted over the network more efficiently.

25 [0004] Although a number of methods are available for CSS file compression, some problems still remain in extensive compression of CSS files.

SUMMARY

[0004a] The invention is defined in the independent claims. Further embodiments of the
30 invention are defined in the dependent claims.

[0005] One embodiment disclosed herein may provide a system for facilitating compression of a Cascading Style Sheet (CSS) file. During operation, the system stores the CSS file in a local storage device and identifies a merge parameter in the CSS file. The merge
35 parameter can be one of: a repeated selector, a repeated property in a selector, and a same value

of a property in different selectors. The system then determines whether a merge condition is satisfied for the merge parameter and merges the merge parameter in the CSS file to obtain a

and together with the description, serve to illustrate the present application, construing no limitation to the present application. In the drawings:

[0016] FIG. 1A illustrates an exemplary compressed CSS file transmission across a network, in accordance with an embodiment of the present application.

5 [0017] FIG. 1B illustrates an exemplary enhanced CSS file compression process, in accordance with an embodiment of the present application.

[0018] FIG. 2A presents a flowchart illustrating an enhanced CSS file compression process of a data compression system, in accordance with an embodiment of the present application.

10 [0019] FIG. 2B presents a flowchart illustrating a validation process of a data compression system for the enhanced CSS file compression, in accordance with an embodiment of the present application.

[0020] FIG. 2C presents a flowchart illustrating an overwrite validation process of a data compression system for the enhanced CSS file compression, in accordance with an embodiment of the present application.

[0021] FIG. 3A illustrates an exemplary graphical selection of common factors for the enhanced CSS file compression, in accordance with an embodiment of the present application.

[0022] FIG. 3B presents a flowchart illustrating an enhanced CSS file compression process of a data compression system based on common factors, in accordance with an embodiment of the present application.

[0023] FIG. 4 is a schematic structural diagram of an exemplary data compression system, in accordance with an embodiment of the present application.

[0024] In the figures, like reference numerals refer to the same figure elements.

25 DETAILED DESCRIPTION

[0025] Embodiments disclosed herein may efficiently compress a CSS file by merging repeated selectors, and repeated properties and values in the CSS file while ensuring compression integrity. In this way, the enhanced compression of the CSS file results in a smaller size, which is conveniently transmitted over a network.

30 [0026] With existing technologies, compressing the CSS file removes the margins, annotations, and repeated characters in the CSS file to reduce the size of the file. Furthermore, some compressors may also remove invalid characters. However, such compression does not significantly reduce redundancy within and across the selectors of the CSS file. Particularly, preprocessing the CSS file can generate a large quantity of redundant CSS code, which may not
35 be directly compressed. Transmitting such a CSS file across a network becomes inefficient.

[0027] Embodiments disclosed herein may provide a system for facilitating enhanced compression of a CSS file. The system compresses the CSS file by merging the repeated selectors (e.g., multiple declaration of the same selector) and removing the repeated properties in the same selector. The system then removes the overwritten style statement in the same selector (e.g., overwritten value of a property) and aggregates the selectors that includes the same properties. The system further extracts and merges the common factors. It should be noted that these five steps can be sequential since a respective step should be performed based on the previous one.

[0028] FIG. 1A illustrates an exemplary compressed CSS file transmission across a network, in accordance with an embodiment of the present application. In this example, a data compression system 102 for CSS file compression is deployed in server 112 for enhanced compression of CSS file 122 at server 112. Server 112 can store the CSS file in a local storage device. Suppose that CSS file 122 is sent over a network 150 (e.g., a local area network (LAN), a wide area network (WAN), the Internet), along with a web page 130 to provide the presentation style for web page 130 in a browser in a client device 114. The presentation style dictates how web page 130 is presented to user 101 in the browser.

[0029] With existing technologies, a compression technique is applied on CSS file 122 prior to transmission over network 150. Typically, compressing CSS file 122 includes removing the margins, annotations, repeated and invalid characters in CSS file 122. However, such compression does not significantly reduce redundancy within and across the selectors of CSS file 122. Transmitting CSS file 122 across network 150 thus becomes inefficient. As disclosed herein, system 102 compresses CSS file 122 to generate a compressed CSS file 124. In some embodiments, system 102 prepares the CSS file for compression by organizing the properties of all selectors in an alphabetic order. Server 112 also stores the compressed CSS file 124 in a local storage device. Whenever client device 114 requests for web page 130, server 112 transmits compressed CSS file 124 over network 150 to client device 114.

[0030] FIG. 1B illustrates an exemplary enhanced CSS file compression process, in accordance with an embodiment of the present application. System 102 generates compressed CSS file 124 based on a number of steps. During the enhanced compression process, system 102 merges the repeated selectors (operation S152). For example, the selectors in `.a{margin:0}.a{padding:0}` are the same. Hence, they can be merged as `.a{margin:0;padding:0}`. System 102 also removes the repeated value assignments for the same property in the same selector (operation S154). Since the value of the property “margin” is repeated in `.a{margin:0;margin:1px}`, system 102 can compress it as `.a{margin:1px}`. System 102 then removes the overwritten value assignment for the same property within the same selector

property have the same priority, the latter value may directly overwrite the former value. For example, `.a{margin:0;margin:1px}` may be directly merged as `.a{margin:1px}`. Here, `.a{margin:1px}` can be considered as a compressed merge parameter. If the values of the same property have different priorities, the value with a higher priority overwrites the value with a lower priority. For example, `.a{margin:0 !important;margin:1px}` is merged as `.a{margin:0 !important}`, or `.a{margin:0;margin:1px !important}` is merged as `.a{margin:1px !important}`.

[0043] If the latter value overwrites the former value, and the latter value has the same or a higher priority over the former value, the merge condition is satisfied. In this case, the latter value overwrites the former one for merging. For example, `.a{background-color:#FFF;background:#000 url(xxx.jpg)}` is merged as `.a{background:#000 url(xxx.jpg)}`. On the other hand, if the former value overwrites the latter value, and the former value has a higher priority over the latter value, the merge condition is satisfied. In this case, the former value with the higher priority overwrites the latter value. For example, `.a{background:#000 url(xxx.jpg) !important;background-color:#FFF}` is merged as `.a{background:#000 url(xxx.jpg) !important}`.

[0044] In some scenarios, the merging decision can be complex. For example, in `.a{margin:0;padding:0;color:#FFF}.b{margin:0;padding:1;color:#FFF}.c{margin:0;padding:1;color:#000}`, each of the three selectors have the same properties with both same and different values. If all possibilities are traversed and all results are calculated to compare the respective lengths, the shortest merged selectors can be selected. However, if there are n selectors and m properties, there are $n!m!$ possibilities. Such extensive computation may not be feasible in practice. Embodiments disclosed herein may provide a graphic selection algorithm for the selection of common factors.

[0045] FIG. 3A illustrates an exemplary graphical selection of common factors for the enhanced CSS file compression, in accordance with an embodiment of the present application. Suppose that there are n selectors 302, 304, and 306 (e.g., selectors `.a`, `.b`, and `.c`, respectively) and m properties 312, 314, and 316 (e.g., properties “margin,” “padding,” “color,” respectively). The graphical selection algorithm generates a two-dimensional table 300 with n row and m columns. During operation, the system first selects the property that maximizes merging. In other words, the system selects a property that maximizes the common area (e.g., the number of grids with the same value for a property) in table 300.

[0046] In table 300, the values of property 312 are the same for a respective selector. Hence, the number of grids that can be merged for property 312 is 3. The system thus determines the common area for property 312 to be 3. In the same way, the system determines the common area for properties 314 and 316 to be 2 and 2, respectively. During the graphical selection

What Is Claimed Is:

1. A method for facilitating compression of a Cascading Style Sheet (CSS) file, the method comprising:

identifying, by a server hosting a web page associated with the CSS file, a first selector and a second selector in the CSS file by identifying a repeated code in the first and second selectors, wherein the repeated code appears in each of the first and second selectors;

determining whether merging the repeated code in the first and second selectors produces a same style for the web page as an original style in the CSS file; and

in response to determining that the merging produces the same style, merging the repeated code in the first and second selectors in the CSS file to obtain a compressed CSS file; and

in response to receiving a request for the web page, generating a web response comprising the web page and the compressed CSS file.

2. The method of claim 1, further comprising determining whether the compressed CSS file is valid based on a length of the merged repeated code.

3. The method of claim 1, wherein determining whether merging the repeated code in the first and second selectors produces the same style for the web page as the original style in the CSS file comprises:

determining whether the first selector and the second selector are adjacent; and

in response to the first and second selectors not being adjacent, determining whether a third selector between the first and second selectors includes a property included in the first and second selectors.

4. The method of claim 3, wherein determining whether merging the repeated code in the first and second selectors produces the same style for the web page as the original style in the CSS file further comprises determining whether the property has one or more of:

a same value in the first, second, and third selectors;

different priorities in the first, second, and third selectors.

5. The method of claim 1, further comprising determining that merging the repeated code in the first and second selectors produces the same style for the web page as the original style in the CSS file in response to determining that:

a latter value of a property of the repeated code overwrites a former value of the property and the latter value has an equal or higher priority; or

a former value of the property overwrites a latter value of the property and the former value has a higher priority.

6. The method of claim 1, wherein merging the repeated code comprises removing a second value of a property for a selector, wherein the second value appears before a first value, has a lower priority than the first value, or has an overwriting relationship with the first value.

7. The method of claim 1, further comprising:
representing the first and second selectors of the CSS file and corresponding properties in a table; and
determining a common area in the table for a respective property.

8. The method of claim 7, further comprising:
identifying a current maximum common area in the table for a property;
extracting the property as a common factor;
merging the common factor across a respective selector comprising the property.

9. The method of claim 1, further comprising generating the compressed CSS file based on a plurality of merged repeated codes.

10. The method of claim 1, further comprising preparing the CSS file for compression by organizing properties of all selectors in the CSS file in an alphabetic order.

11. A non-transitory storage medium storing instructions, which when executed by a processor cause the processor to perform a method for facilitating dynamic load-based merging, the method comprising:

identifying, by a server hosting a web page associated with a Cascading Style Sheet (CSS) file, a first selector and a second selector in the CSS file by identifying a repeated code in the first and second selectors, wherein the repeated code appears in each of the first and second selectors;

determining whether merging the repeated code in the first and second selectors produces a same style for the web page as an original style in the CSS file; and

in response to determining that the merging produces the same style, merging the

repeated code in the first and second selectors in the CSS file to obtain a compressed CSS file; and

in response to receiving a request for the web page, generating a web response comprising the web page and the compressed CSS file.

12. The non-transitory storage medium of claim 11, wherein the method further comprises determining whether the compressed CSS file is valid based on a length of the merged repeated code.

13. The non-transitory storage medium of claim 11, wherein determining whether merging the repeated code in the first and second selectors produces the same style for the web page as the original style in the CSS file comprises:

determining whether the first selector and the second selector are adjacent; and

in response to the first and second selectors not being adjacent, determining whether a third selector between the first and second selectors includes a property included in the first and second selectors.

14. The non-transitory storage medium of claim 13, wherein determining whether merging the repeated code in the first and second selectors produces the same style for the web page as the original style in the CSS file further comprises determining whether the property has one or more of:

a same value in the first, second, and third selectors;

different priorities in the first, second, and third selectors.

15. The non-transitory storage medium of claim 11, wherein the method further comprises determining that merging the repeated code in the first and second selectors produces the same style for the web page as the original style in the CSS file in response to determining that:

a latter value of a property of the repeated code overwrites a former value of the property and the latter value has an equal or higher priority; or

a former value of the property overwrites a latter value of the property and the former value has a higher priority.

16. The non-transitory storage medium of claim 11, wherein merging the repeated code comprises removing a second value of a property for a selector, wherein the second

value appears before a first value, has a lower priority than the first value, or has an overwriting relationship with the first value.

17. The non-transitory storage medium of claim 11, wherein the method further comprises:

representing the first and second selectors of the CSS file and corresponding properties in a table; and
determining a common area in the table for a respective property.

18. The non-transitory storage medium of claim 17, wherein the method further comprises:

identifying a current maximum common area in the table for a property;
extracting the property as a common factor;
merging the common factor across a respective selector comprising the property.

19. The non-transitory storage medium of claim 11, wherein the method further comprises generating the compressed CSS file based on a plurality of merged repeated codes.

20. The non-transitory storage medium of claim 11, wherein the method further comprises preparing the CSS file for compression by organizing properties of all selectors in the CSS file in an alphabetic order.