Technical Disclosure Commons

Defensive Publications Series

December 11, 2017

Preferential Resource Delivery Via Web Proxy or Web Browser

Dean Kenneth Jackson

Daniel Klein

Preferential Resource Delivery Via Web Proxy or Web Browser

Abstract

The subject disclosure relates to prioritizing the loading of files referenced by the HyperText Markup Language (HTML) of a webpage, so that the time required to load the webpage is reduced, for example, as perceived by an end user. The HTML for a webpage can be parsed to identify what types of resources are referenced (e.g., a cascading style sheets (CSS) file, image files, JavaScript), and the bandwidth associated with downloading the resources can be assessed. Based on requested resources and bandwidth, the system can prioritize the loading of resources, in order to load portions of the webpage quickly so that the end user perceives an improved loading time.

Background

Image heavy websites can load slowly due to the network load of downloading the images themselves. However, the user often sees nothing of the webpage until such time as the "essential" components of the page are downloaded. For example, the height and width of the images are often required before the page can be constructed, and the images can later be included as they are downloaded. Additionally, the content of the CSS may also be needed in its entirety before the page can be constructed.

In some systems, image data is included in the HTML attributes and is downloaded first. In other systems, the image attributes are contained in a CSS file, which must be downloaded separately, in competition with all of the images. This can result in slow page display, for example, when a device faced with limited bandwidth attempts to load images before all HTML/CSS has been loaded.

Typical HTML includes text and references to images. The text is formatted via inline HTML tags, while images are referenced by the HTML and stored in separate files. Both text and images can be formatted using CSS, which is also stored in a separate file or files. A page request causes the HTML to be loaded, and then every file which it references is in turn loaded. When a page requires auxiliary files to be loaded in order to be displayed, it is usually necessary for those auxiliary files to be loaded before displaying any part of the page.

A well-designed page will typically have all of the necessary information stored in the HTML, for example, image attributes such as height/width. However, many webpages are not well-designed. For example, webpages that are not well-designed may: (1) be missing height/width attributes in an image tag, (e.g., which requires that the image be downloaded to determine its size), (2) include height/width attributes contained in the CSS specification, and/or or (3) include HTML that may be included into other HTML by methods like an iframe or server side inclusion.

The problem is that once an HTML page is loaded, which is usually a relatively quick operation, the webpage will make a large number of parallel requests for the auxiliary image (e.g., JPEG) and CSS files. These parallel requests heavily load the network for a short period of time.

Once the browser knows all of the images sizes, the page can be rendered with placeholders for images that have not yet been fully downloaded. In other words, if the image size is known, the image is not needed for rendering. However, until such time as the image headers are downloaded, or the entire CSS file is loaded, the browser may not have enough information about the page to begin rendering it. While the CSS is needed for rendering and the image itself may not be needed, both the CSS and the images are typically given the same

priority across the network, with images potentially blocking, stalling or otherwise slowing CSS loading.

Features of the subject technology

The subject technology proposes (a) assigning different priority classes for files (e.g., CSS, JavaScript files, image files) referenced by the HTML of a webpage, and (b) assessing the bandwidth associated with downloading the referenced files. Based on the assigned priority classes and the assessed bandwidth, the system can prioritize downloading some files, or parts thereof, over other files, or parts thereof. In doing so, the subject technology can provide for a faster loading of webpages as perceived by an end user.

Assigning Different Priority Classes for Files

This functionality can be implemented at the web browser. Alternatively, or in addition, the functionality can be implemented at a web proxy positioned between the browser and a network (e.g., the Internet).

The web proxy or web browser parses HTML in order to identify whether a webpage references a CSS file, JavaScript and/or any images files. For example, a web proxy and/or web browser can examine the data stream associated with a web browser of a client device. If the client device requests a webpage, the web proxy and/or browser can: (1) prioritize traffic to the client device in favor of the CSS file over any image files, (2) prioritize traffic to the client device in favor of the first block of data from any image file (that is, containing the header, which contains the image size) over the remainder of all image files, and (3) prioritize traffic such that images which are on the visible portion of the display window are loaded before images which are above, below or behind the currently visible portion of the display window. In example aspects, the web proxy and/or web browser prioritizes requests for CSS files, so that

CSS files are downloaded first and possibly with limited or no competition with other files (e.g., image files).

Regarding JavaScript files, the web browser or web proxy may load external JavaScript before or after images, depending on where the JavaScript appears on the webpage. For example, if external JavaScript is loaded in the header of the webpage, the JavaScript can be loaded with or after the downloading of the CSS. On the other hand, if the JavaScript is loaded in the footer, the JavaScript can be loaded with or ahead of the image files.

The web proxy and/or browser can determine which images (or JavaScript) will initially be off-screen (e.g., for the client device) and defer loading of those images. In other words, content such as image content or JavaScript that is not within the initial view of the client display can be assigned low priority, for loading at a later time (e.g., based on a scroll action by the user or after all currently visible images are loaded). In other words, the system can alter the priority if the user scrolls, but can download everything that might be seen by the user after dealing with what is actually seen first.

In example aspects, the above-described prioritization is in addition to any caching that the browser or proxy may already do, for example, so that cached images and cached CSS files are also prioritized. Additionally, based on previously determined image data, a caching proxy may rewrite the HTML to include missing image attributes, so that the browser will be presented with as-complete rendering information as possible.

Assessing Bandwidth Needs

Bandwidth availability can be used in determining whether or not the loading of some files (e.g., CSS files) should be prioritized over other files (e.g., image files). Bandwidth can correspond to the speed at which a hosting website can deliver traffic, and the speed at which the

browser at the client device can accept traffic. In example aspects, the above-described functionality for prioritizing the loading of CSS, JavaScript and images may be enabled for low bandwidth connections (e.g., for a 2G mobile device), and disabled for higher bandwidth connections (e.g., a wired connection or 4G mobile).

In addition, for content (e.g., images) from different servers, the bandwidth at which the respective server delivers traffic can be used in prioritizing the order of loading images. For example, rather than loading images from different servers in parallel, the web proxy and/or browser can favor the image from the server known to have a higher download speed.

In example aspects, the above-described functionality for prioritizing the loading of CSS, JavaScript and images may only be enabled for certain sites, with known large file counts/file sizes. Alternatively or in addition, the functionality may be turned on/off by a response header, such that servers can disable this functionality.

An example of prioritized loading of files will now be described with reference to FIG. 1 shown below:

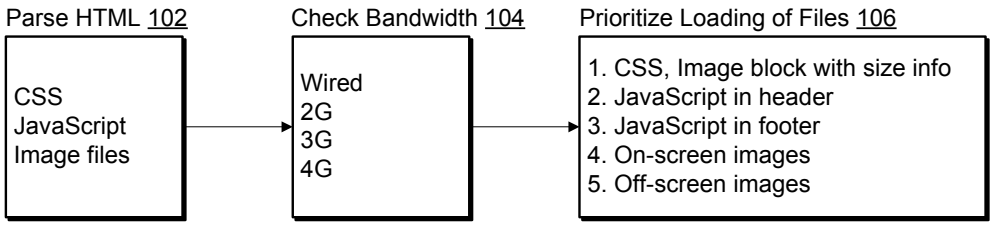

FIG. 1

In the example of FIG. 1, the system (e.g., a web proxy and/or web browser) at block 102 parses the HTML of a webpage to determine whether or not the HTML of a webpage references the loading of CSS, JavaScript and/or images files.

At block 104, the system determines the bandwidth for the traffic associated with loading the referenced files (e.g., CSS, JavaScript and/or images files). For example, the determination of bandwidth may be based on the speed at which a hosting website can deliver traffic, and the speed at which the browser at the client device can accept traffic. In the example of FIG. 1, the connection for delivering content to the client device can be one or more of a wired connection, 2G, 3G or 4G. With a wired or a 4G connection, the prioritized loading of files may be disabled, whereas a 2G or 3G connection may enable prioritized loading of files.

At block 106, the loading of files (e.g., CSS, JavaScript and/or images files) is prioritized. In the example of FIG. 1, CSS and the header blocks of image files (e.g., for determining size) are favored or prioritized over other content such as JavaScript and other image content. For example, the system prioritizes requests for CSS files so that CSS files are downloaded first and possibly with limited or no competition with other files (e.g., image files). In example aspects, the header blocks of image files (e.g., for determining size) may be treated in a similar manner, and are requested together with or immediately after the loading of CSS.

As shown in block 106, external JavaScript in the header of the webpage can be loaded with or after the downloading of the CSS, and JavaScript in the footer can be loaded with or ahead of the image files. In addition, the system (e.g., web proxy and/or browser) can determine which images (or JavaScript) will initially be on-screen or off-screen (e.g., for the client device), and defer the loading of off-screen images in favor of on-screen images. Off-screen images (or JavaScript) can be loaded at a later time, for example, based on a scroll action by the user.

Accordingly, based on the referenced resources in HTML and bandwidth, the system can prioritize the loading of resources, in order to load portions of the webpage quickly so that the end user perceives an improved loading time.