February 13, 2019

Speeding up web page content updates using server-side page versioning

Alessio Centazzo

Speeding up web page content updates using server-side page versioning

ABSTRACT

In response to a page refresh request from a user, a web browser application often requests and downloads a previously loaded page again, even when the amount of updated content is small. As a result, bandwidth and time is wasted. This disclosure describes use of page versioning techniques to reduce the amount of data and time needed to update a web page. A web server implements page versioning to assign a unique identifier to different versions of a web page. The unique identifier is sent to the web browser along with the page content. Refresh requests received from the browser include the unique identifier of the page. The web server identifies the content update based on a comparison of the current version with that corresponding to the unique identifier and sends data to the browser that is usable to transform the page to the current version. The described techniques save data usage costs and reduce latency for page refresh.

KEYWORDS

- web server
- browser
- web page refresh
- content update
- page versioning
- page hash
- page identifier
- content delta

BACKGROUND

Web page downloads are slow on slow network connections and lead to a frustrating user experience. When a user requests an update of the contents of a web page that is already loaded in the web browser, the browser requests and downloads the entire page again, even when the amount of updated content is minimal. As a result, bandwidth and time is wasted in reloading the contents of the page that are already available locally. On a slow network, such operation also causes higher data costs (e.g., on mobile networks that are metered) and leads to user frustration due to the page load time.

DESCRIPTION

This disclosure describes use of page versioning techniques to reduce the amount of data and time needed to update a web page. A web server implements page versioning to assign a unique identifier to different versions of a web page. The unique identifier is sent to the web browser along with the page content. Refresh requests received from the browser include the unique identifier of the page. The web server identifies the content update based on a comparison of the current version with that corresponding to the unique identifier and sends data to the browser that is usable to transform the page to the current version.

The data is used by the browser to update only the content within the currently loaded web page that has changed since the page was previously loaded. The identifier of the page loaded within the browser is updated to the updated identifier received from the server. In case the page is unchanged since it was loaded within the browser, no updated content or identifier is transferred,. Alternatively, if the more than a threshold portion of the page content has changed since it was last loaded, the complete page along with the new identifier is sent to the browser.

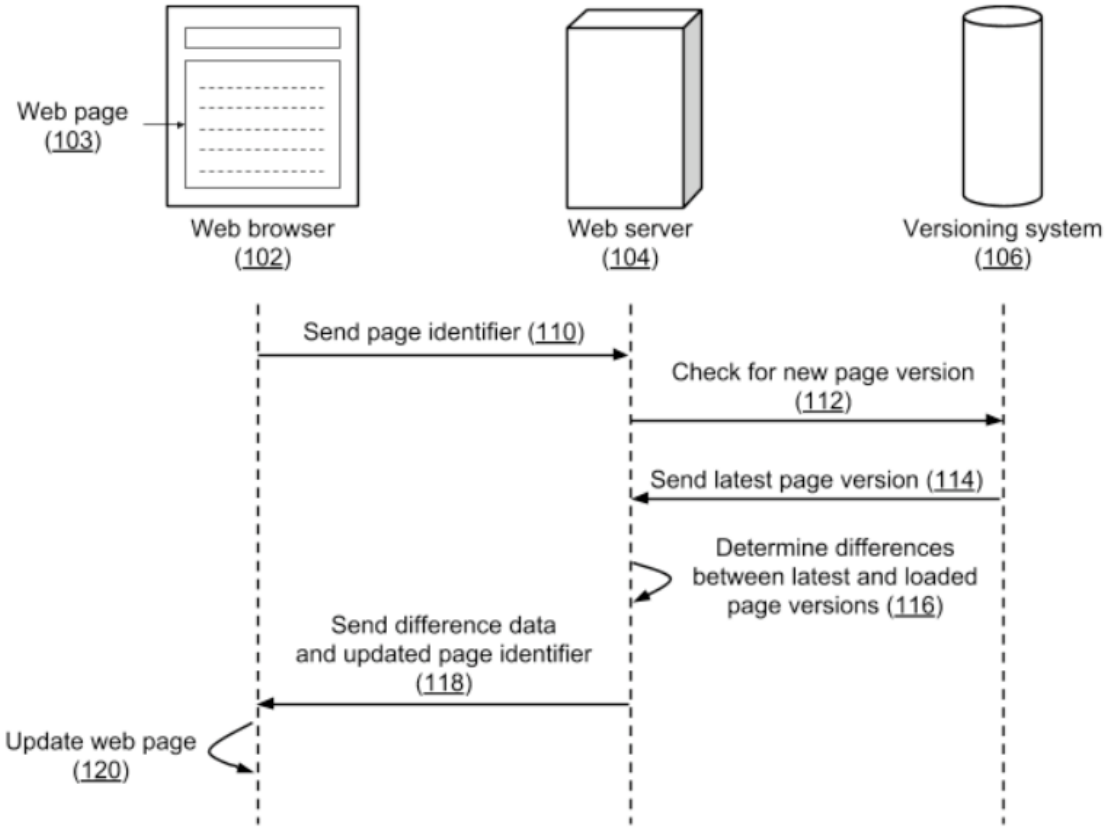

Fig. 1: Operational flow of requesting updates for a web page loaded in the web browser

Fig. 1 shows an example flow of operations per the techniques described in this disclosure. A web browser (102) on a user device has loaded a web page (103). The page version that is loaded is associated with a unique identifier.

At a later time, a page refresh is detected, e.g., triggered by a user command. In response, a refresh request that includes the page identifier (110) is sent to a web server (104). Upon receipt the request, a version check (112) by a versioning system (106), e.g., that is implemented as part of the web server. The latest version of the page (114) is compared with the previously loaded version to determine the content difference (116), if any.

Data corresponding to the difference are sent to the web browser along with an updated identifier associated with the latest version of the page (118). Using the received difference data,

the web page is refreshed in the browser (120) by changing only the pieces of content that have been updated since the last time the page was loaded.

The described techniques can be implemented using commercial source control systems. The Secure Hash Algorithm (SHA) signature of each object corresponding to a web page version can be used as the unique identifier. Upon receiving a request from the browser for updated contents for an already loaded web page, the page version associated with the identifier of the already loaded web page is located and compared with the latest version of the page. A page delta that indicates the difference between the latest and the currently loaded version of the web page is calculated. The page delta along with the SHA identifier of the latest version of the page is sent to the browser. For efficient operation, the entire page can be transmitted, e.g., instead of an extensive page delta, in cases where a majority of the page content has changed since the page was last loaded by the browser.

The described techniques are particularly suitable for serving web pages that involve frequently changing piece of content, such as news, forums, chat, etc. The techniques described in this disclosure can reduce the amount of data that needs to be transferred across the network to update a web page already loaded on a user device. Therefore, the techniques can save data usage costs and reduce latency in refreshing a web page with updated content. The described techniques can enhance the web browsing user experience, especially for client browsers that have slow network connectivity. The use of page versioning and sending page deltas can also lower the carbon footprint of the network operation.

The described techniques are implemented with user permission. If the user denies permission, the techniques are not implemented and an entire page refresh is performed. The user is provided with options to turn off page version based refreshes, e.g., for certain websites, or

turn off the features entirely. For example, such options can be provided as part of browser configuration, within individual web pages, etc. When versioning is enabled, page identifiers are calculated such that no user specific data is utilized.

Further to the descriptions above, a user is provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein enable collection of user information (e.g., information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data is treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity is treated so that no personally identifiable information can be determined for the user, or a user's geographic location is generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user has control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes use of page versioning techniques to reduce the amount of data and time needed to update a web page. A web server implements page versioning to assign a unique identifier to different versions of a web page. The unique identifier is sent to the web browser along with the page content. Refresh requests received from the browser include the unique identifier of the page. The web server identifies the content update based on a comparison of the current version with that corresponding to the unique identifier and sends data to the browser that is usable to transform the page to the current version. The described techniques save data usage costs and reduce latency for page refresh.