

(51) International Patent Classification:  
**G06F 17/30** (2006.01)(21) International Application Number:  
PCT/US2012/020106(22) International Filing Date:  
3 January 2012 (03.01.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
12/985,799 6 January 2011 (06.01.2011) US(71) Applicant (for all designated States except US): **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, M/S 50P7, Redwood Shores, California 94065 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SULLIVAN, Blake** [US/US]; 1729 Whipple Avenue, Redwood City, California 94062 (US). **GAO, Yuan** [CN/US]; 52 East 41st Place, #5, San Mateo, California 94403 (US). **WANG, Hongbing** [US/US]; 998 Cartier Lane, Foster City, California 94404 (US).(74) Agents: **LEE, Andrew** et al.; Kilpatrick Townsend & Stockton LLP, Two Embarcadero Center, 8th Floor, San Francisco, California 94111 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

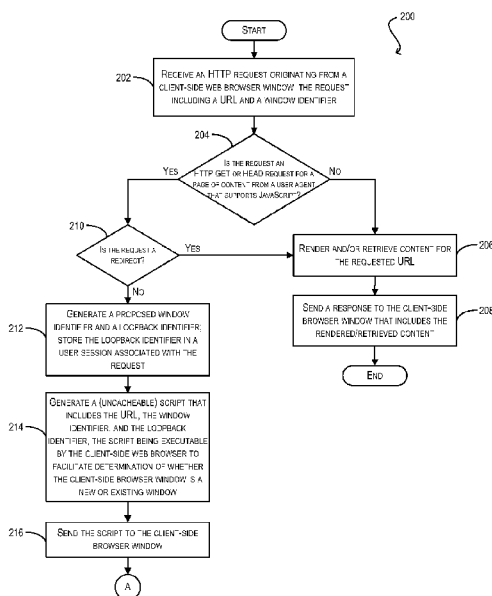
**Declarations under Rule 4.17:**

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

**Published:**

— with international search report (Art. 21(3))

(54) Title: TECHNIQUES FOR DETECTING NEW BROWSER WINDOWS

**FIG. 2**(57) **Abstract:** Embodiments of the present invention provide techniques for tracking the lifecycles of client-side browser windows in the context of a web application. In one set of embodiments, a server component of a web application can receive a request originating from a client-side browser window, where the request includes a resource locator (e.g., a URL) and a window identifier. In response, the server component can send a script to the window for retrieving the window's actual identifier/name. Based on the actual identifier returned by the script and the window identifier included in the initial request, the server component can determine whether the window is a new window or an existing window and can process the request as appropriate.

## TECHNIQUES FOR DETECTING NEW BROWSER WINDOWS

### CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] The present application claims the benefit of U.S. Application No. 12/985,799, filed  
5 January 6, 2011, entitled "TECHNIQUES FOR DETECTING NEW BROWSER  
WINDOWS," the entire contents of which are incorporated herein by reference for all  
purposes. The present disclosure is related to the following commonly assigned, co-pending  
U.S. Patent Application, which is incorporated herein by reference in its entirety for all  
purposes: Application No. 12/985,805, filed January 6, 2011, entitled "TECHNIQUES FOR  
10 DETECTING INACTIVE BROWSER WINDOWS."

### BACKGROUND

[0002] The present disclosure relates in general to web applications, and in particular to  
techniques for tracking the lifecycles of client-side browser windows in such applications.

15 [0003] Generally speaking, a web application is a software application that is hosted on a  
server and is accessible by users via a client-side web browser. In a typical use case, a user  
opens a client-side browser window and navigates to an initial web page of the application.  
From there, the user can input information (e.g., via form elements embedded in the page  
markup) and/or navigate to additional pages of the application to complete some task or  
20 transaction.

[0004] In some instances, the user may open a new browser window from the context of an  
existing browser window while accessing the application. This causes the browser to send a  
new request to the server for the application page displayed in the existing window. In  
current implementations, the server generally cannot distinguish between this new window  
25 event and a situation where the user has reloaded/refreshed the existing window – in either  
case, the request received by the server appears to originate from the existing window. This  
can be problematic since the server typically maintains separate application state information  
for each browser window, and accordingly needs to know which window it is serving before  
rendering a page. For example, if the server erroneously determines that a client request  
30 corresponds to a reload of an existing window (when it actually corresponds to the opening of

a new window), the server may unintentionally modify the application state of the existing window when processing the request.

[0005] In other instances, the user may close a browser window that he/she has used for accessing the application, or navigate away from the application (e.g., to a third party web site) within the window. In current implementations, the server generally cannot distinguish between these situations and a situation where the user has navigated (within the window) to another web page in the same application – in either case, the server simply receives a notification from the browser that the application page in the window has been unloaded/destroyed. This can make it difficult for the server to determine when to free memory resources (e.g., application state information) that are maintained by the server for each browser window. For example, if the server erroneously determines that a browser window is still being used to access the application (when the user has actually closed the window or navigated away from the application), the server may keep the resources for the window in memory longer than necessary, which is undesirable from both an efficiency and security standpoint.

#### BRIEF SUMMARY

[0006] Embodiments of the present invention provide techniques for tracking the lifecycles of client-side browser windows in the context of a web application. In one set of embodiments, a server component of a web application can receive a request originating from a client-side browser window, where the request includes a resource locator (e.g., a URL) and a window identifier. In response, the server component can send a script to the window for retrieving the window's actual identifier/name. Based on the actual identifier returned by the script and the window identifier included in the initial request, the server component can determine whether the window is a new window or an existing window and can process the request as appropriate.

[0007] In another set of embodiments, the server component can receive a notification that an application page in a first client-side browser window has been unloaded/destroyed. In response, the server component can associate the first window with an expiration time and can add the first window to an "expiration" list of windows that are potentially inactive (i.e., closed or no longer being used to accessing the application). If a new request is received by the server component from the first window while the first window is in the expiration list, the server component can identify the first window as active (i.e., still being used to accessing

the application) and remove it from the list. If a new request is received by the server component from a second window while the first window is in the expiration list, the server component can compare the time at which the request from the second window was received with the expiration time associated with the first window. If the time of the request is later  
5 than the expiration time, the server component can identify the first window as inactive and free all resources associated with the first window.

[0008] In yet another set of embodiments, the server component can receive a notification that an application page in the last remaining window for the client-side browser has been unloaded/destroyed. In response, the server component can modify a session timeout  
10 variable for a user session associated with the browser from a first duration (e.g., one hour) to a second duration shorter than the first duration (e.g., ten minutes). The server component can then begin a countdown of the session timeout variable. If a request is received by the server component from the last remaining window before the second duration expires, the server component can identify the window as active and reset the session timeout variable to the  
15 first duration. If no request is received from the last remaining window before the second duration expires, the server component can identify the window as inactive, free all resources associated with the window, and free all resources associated with the user session.

[0009] According to one embodiment of the present invention, a method is provided that comprises receiving, by a server computer system, a first request originating from a window  
20 of a web browser running on a client computer system, the first request including a Uniform Resource Locator (URL) and a first window identifier; generating, by the server computer system, a loopback identifier; and storing, by the server computer system, the loopback identifier in a user session associated with the request. The method further comprises generating, by the server computer system, a script that includes the URL, the first window  
25 identifier, and the loopback identifier; and sending, by the server computer system, the script to the window that originated the first request.

[0010] In one embodiment, the script is executable by the web browser running on the client computer system to facilitate determination of whether the window that originated the first request is a new window or an existing window.

30 [0011] In one embodiment, the loopback identifier is a randomly generated number.

[0012] In one embodiment, the first request is a Hypertext Transfer Protocol (HTTP) request.

[0013] In one embodiment, the script is sent to the window that originated the first request only if the first request is an HTTP GET or HEAD request.

[0014] In one embodiment, the script is configured to, on the client computer system, determine whether the first window identifier matches an identifier of the window that originated the first request, and based on the determining, send a second request to the server computer system.

[0015] In one embodiment, if the first window identifier matches an identifier of the window that originated the first request, the second request includes the URL, the loopback identifier, and the first window identifier.

[0016] In one embodiment, if the first window identifier does not match an identifier of the window that originated the first request, the second request includes the URL, the loopback identifier, and a second window identifier distinct from the first window identifier.

[0017] In one embodiment, the method further comprises receiving, by the server computer system, the second request; matching, by the server computer system, the loopback identifier included in the second request with the loopback identifier stored in the user session; and removing the loopback identifier from the user session.

[0018] In one embodiment, the method further comprises, if the second request includes the first window identifier, determining, by the server computer system, that the window that originated the first request is a new window; and if the second request includes the second window identifier, determining, by the server computer system, that the window that originated the first request is an existing window.

[0019] In one embodiment, the method further comprises rendering, by the server computer system, content for the URL based on whether the window that originated the first request is a new window or an existing window; and sending, by the server computer system, the rendered content to the window that originated the first request.

[0020] According to another embodiment of the present invention, a non-transitory computer readable storage medium is provided having stored thereon instructions executable by a processor. The instructions comprise instructions that cause the processor to receive a first request originating from a window of a web browser running on a client computer system, the first request including a Uniform Resource Locator (URL) and a first window identifier; instructions that cause the processor to generate a loopback identifier; and

instructions that cause the processor to store the loopback identifier in a user session associated with the request. The instructions further comprise instructions that cause the processor to generate a script that includes the URL, the first window identifier, and the loopback identifier; and instructions that cause the processor to send the script to the window that originated the first request.

[0021] According to another embodiment of the present invention, a server computer system is provided comprising a processor configured to receive a first request originating from a window of a web browser running on a client computer system, the first request including a Uniform Resource Locator (URL) and a first window identifier; generate a loopback identifier; and store the loopback identifier in a user session associated with the request. The processor is further configured to generate a script that includes the URL, the first window identifier, and the loopback identifier; and send the script to the window that originated the first request.

[0022] A further understanding of the nature and advantages of the embodiments disclosed herein can be realized by reference to the remaining portions of the specification and the attached drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0023] FIG. 1 is a simplified block diagram illustrating a system in accordance with an embodiment of the present invention.

[0024] FIG. 2 is a flow diagram illustrating a process for determining whether a client-side browser window is a new window or an existing window in accordance with an embodiment of the present invention.

[0025] FIG. 3 is a flow diagram illustrating a process performed by a client-side script in accordance with an embodiment of the present invention.

[0026] FIG. 4 is a flow diagram illustrating a further process for determining whether a client-side browser window is a new window or an existing window in accordance with an embodiment of the present invention.

[0027] FIG. 5 is a flow diagram illustrating a process for determining whether a client-side browser window is inactive in accordance with an embodiment of the present invention.

[0028] FIG. 6 is a flow diagram illustrating a further process for determining whether a client-side browser window is inactive in accordance with an embodiment of the present invention.

[0029] FIG. 7 is a flow diagram illustrating a process for determining whether the last remaining window of a client-side browser is inactive in accordance with an embodiment of the present invention.

[0030] FIG. 8 is a simplified block diagram illustrating a system environment in accordance with an embodiment of the present invention.

[0031] FIG. 9 is a simplified block diagram illustrating a computer system in accordance with an embodiment of the present invention.

#### DETAILED DESCRIPTION

[0032] In the following description, for the purposes of explanation, numerous details are set forth in order to provide an understanding of embodiments of the present invention. It will be apparent, however, to one of ordinary skill in the art that certain embodiments can be practiced without some of these details.

[0033] Embodiments of the present invention provide techniques for tracking the lifecycles of client-side browser windows in the context of a web application. In one set of embodiments, a server component of a web application can receive a request originating from a client-side browser window, where the request includes a resource locator (e.g., a URL) and a window identifier. In response, the server component can send a script to the window for retrieving the window's actual identifier/name. Based on the actual identifier returned by the script and the window identifier included in the initial request, the server component can determine whether the window is a new window or an existing window and can process the request as appropriate.

[0034] In another set of embodiments, the server component can receive a notification that an application page in a first client-side browser window has been unloaded/destroyed. In response, the server component can associate the first window with an expiration time and can add the first window to an "expiration" list of windows that are potentially inactive (i.e., closed or no longer being used to access the application). If a new request is received by the server component from the first window while the first window is in the expiration list, the

server component can identify the first window as active (i.e., still being used to accessing the application) and remove it from the list. If a new request is received by the server component from a second window while the first window is in the expiration list, the server component can compare the time at which the request from the second window was received with the expiration time associated with the first window. If the time of the request is later than the expiration time, the server component can identify the first window as inactive and free all resources associated with the first window.

[0035] In yet another set of embodiments, the server component can receive a notification that an application page in the last remaining window for the client-side browser has been unloaded/destroyed. In response, the server component can modify a session timeout variable for a user session associated with the browser from a first duration (e.g., one hour) to a second duration shorter than the first duration (e.g., ten minutes). The server component can then begin a countdown of the session timeout variable. If a request is received by the server component from the last remaining window before the second duration expires, the server component can identify the window as active and reset the session timeout variable to the first duration. If no request is received from the last remaining window before the second duration expires, the server component can identify the window as inactive, free all resources associated with the window, and free all resources associated with the user session.

[0036] FIG. 1 is a simplified block diagram illustrating a system 100 according to an embodiment of the present invention. As shown, system 100 can include a client 102 in communication with a server 104. Client 102 can be any type of electronic device or system (e.g., desktop computer, laptop computer, tablet computer, smartphone, personal digital assistant (PDA), set-top box, video game system, etc.) capable of running a web browser 106. Examples of available web browsers include Mozilla Firefox, Microsoft Internet Explorer, Google Chrome, Apple Safari, and the like. In various embodiments, web browser 106 can be operated by one or more users of client 102 to access web resources, such as a web application hosted by server 104.

[0037] In certain embodiments, web browser 106 can enable users to access multiple web resources simultaneously via multiple browser windows 108. For example, a user can open a first browser window in browser 106 and navigate to a first web page "A". From the context of the first browser window, the user can open one or more additional (new) browser windows, thereby causing additional instances of web page A to be retrieved and rendered in the new windows. From there, the user can navigate to other web pages using any of the

open windows. The user can also close one or more of the open windows once the user has finished his/her browsing session.

[0038] As used herein, the terms “browser window” and “window” can refer to any viewport of browser 106 capable of displaying web content identified by a resource identifier (e.g., a URL). In a particular embodiment, the terms “browser window” and “window” can refer to a browser “tab,” although such tabs are not typically considered to be standalone windows.

[0039] Server 104 can be any type of electronic device or system capable of hosting a web application such as application 114. In one set of embodiments, server 104 can include an application server component 110 configured to handle interactions between web browser 106 and web application 114 and to provide various services such as dynamic web page creation, load balancing, security, and the like. In a particular embodiment, application server 110 can include a web server component 112 configured to receive Hypertext Transfer Protocol (HTTP) requests from browser 106 of client 102 and to pass those requests to application server 110 and/or application 114 for processing. Web server 112 can also return HTTP responses (e.g., rendered HTML) to browser 106.

[0040] In one set of embodiments, application server 110 can be configured to track the lifecycles of client-side browser windows, such as windows 108 of web browser 106. For example, in one embodiment application server 110 can detect when a user of browser 106 has opened a new browser window while accessing application 114, and can distinguish that new window event from a reload/refresh event on an existing browser window. In another embodiment, application server 110 can detect when a user of browser 106 has inactivated a browser window by closing the window or navigating away from application 114, and can distinguish those situations from a situation where the user has merely navigated to another web page of application 114. The specific processing performed by application server 110 in these scenarios is described in detail with respect to FIGS. 2-7 below.

[0041] In addition to client 102 and server 104, system 100 can also include a data store 116. Data store 116 can be communicatively coupled with server 104 and can provide data storage services for various components of server 104, such as web application 114.

Although server 104 and data store 116 are shown as separate entities in system 100, one of ordinary skill in the art will appreciate that server 104 and data store 116 can correspond to a single physical machine or separate physical machines.

[0042] It should be appreciated that FIG. 1 is illustrative and not intended to limit embodiments of the present invention. For example, the various entities of system 100 may each have additional capabilities or include additional components that are not specifically described. In addition, although only one client, one server, and one data store are depicted, any number of such entities can be supported. One of ordinary skill in the art will recognize many variations, modifications, and alternatives.

#### Detection of New Client-side Browser Windows

[0043] As indicated above, in certain embodiments application server 110 can determine when a user of browser 106 has opened a new browser window while accessing application 114, and can distinguish that new window event from a reload/refresh event on an existing browser window. FIG. 2 illustrates a process 200 that can be performed by application server 110 for implementing this new window detection according to an embodiment of the present invention. In various embodiments, process 200 can be implemented in hardware, software, or a combination thereof. As software, process 200 can be encoded as program code stored on a machine-readable storage medium.

[0044] At block 202, application server 110 can receive an HTTP request originating from a window of client-side web browser 106, where the request is for a resource (e.g., web page) provided by web application 114. In various embodiments, the request can include an identifier of the resource (e.g., a URL), as well as an identifier of a particular browser window. For example, the request can include a URL “X” and a window identifier “Window1.” In most situations, the window identifier included in the request will correspond to the identifier of the browser window that originated the request. However, if the request originated from a new window created from the context of an existing window, the window identifier included in the request will correspond to the identifier/name of the existing window rather than the new window. Accordingly, in these cases application server 110 cannot readily discern, based on the window identifier alone, whether the originating window is a new or existing window.

[0045] At block 204, application server 110 can determine whether the request is an HTTP GET or HEAD request for a page of content from a user agent that supports JavaScript. If the request does not fall into this particular category, application server 110 can determine that the originating browser window is not a new window. Accordingly, application server 110 can performing its normal processing flow of rendering/retrieving content for the

requested URL and sending a response to the browser window with the rendered/retrieved content (blocks 206, 208).

[0046] If the request is an HTTP GET or HEAD request for content from a user agent supporting JavaScript, application server 110 can further determine whether the request is a  
5 redirect (block 210). If the request is a redirect, application server 110 can perform the processing flow of blocks 206, 208 as described above.

[0047] If the request is not a redirect, application server 110 can determine that the request may have originated from a new browser window. In response, application server 110 can generate a proposed window identifier and a loopback identifier, and store the loopback  
10 identifier in a user session associated with the request (block 212). In one set of embodiments, the loopback identifier can be a randomly generated number. Application server 110 can further generate a client-side script that includes the loopback identifier, the proposed window identifier, and the URL and window identifier from the received request (e.g., “X” and “Window1”), and can send the script to the originating browser window  
15 (blocks 214, 216). In one set of embodiments, the client-side script can be configured to be uncacheable by client-side browser 106. The client-side script can then be executed by browser 106 to retrieve the actual identifier of the originating browser window, thereby facilitating determination of whether the originating window is a new or existing window.

[0048] FIG. 3 illustrates a process 300 that can be performed by web browser 106 upon  
20 receiving and executing the client-side script sent at block 214. At block 302, browser 106 can determine whether the window identifier included in the script (i.e., the window identifier extracted from the request received at block 202) matches the identifier/name of the current browser window (i.e., the window that originated the request). If the identifiers match, that indicates that the current window is an existing window. In this case, browser 106 can  
25 initiate a client-side redirect to send a second HTTP request to server 104 that mirrors the original HTTP request (e.g., includes URL “U1” and window identifier “Window1”), with the addition of the loopback identifier (block 304).

[0049] If the identifiers do not match, that indicates that the current window is a new window. In this case, browser 106 can determine whether there already exists a window  
30 identifier for the current window (block 306). For example, browser 106 may have automatically assigned a new name/identifier to the window upon opening it. If such an existing identifier is found, browser 106 can initiate a client-side redirect to send a second

HTTP request to server 104, wherein the second request includes the loopback identifier, the URL, and the existing identifier (block 308). If an existing window identifier is not found, browser 106 can initiate a client-side redirect to send a second HTTP request that includes the loopback identifier, the URL, and the proposed window identifier generated at block 212 of FIG. 2 (block 310).

**[0050]** Upon receiving the second HTTP request sent by browser 106 at block 304, 308, or 310 of FIG. 3, application server 110 can carry out process 400 of FIG. 4. At blocks 402 and 404, application server 110 can receive the second HTTP request and match the loopback identifier included in the second request with the loopback identifier stored in the user's session. If the loopback identifiers match, the application server knows that this second request was generated in response to the client-side script sent at block 214 (rather than in response to a user action). Accordingly, application server 110 can remove the loopback identifier from the session (block 406) and proceed with the remainder of process 400.

**[0051]** At block 408, application server 110 can determine whether the window identifier included in the second request matches the window identifier included in the original request received at block 202. If the identifiers match, application server 110 can conclude that the window that originated the original request of block 202 is an existing window. Alternatively, if the identifiers do not match, server 104 can conclude that the window that originated the original request of block 202 is a new window. Application server 110 can then process the original request by rendering/retrieving the content for the URL and sending a response to the originating browser window that includes the rendered/retrieved content (blocks 414, 416).

**[0052]** In a particular embodiment, the processing of the original request can differ based on whether the originating window is determined to be a new window or an existing window. For example, if the originating window is determined to be an existing window per block 410, application server 110 can freely update/modify application state information associated with that window. However, if the originating window is determined to be a new window per block 412, application server 110 can take appropriate steps to ensure that the states of any existing windows remain unchanged. In this manner, application server 110 can avoid unintentionally trashing/invalidating the states of those existing windows when processing the request from the new window.

[0053] It should be appreciated that processes 200, 300, and 400 are illustrative and that variations and modifications are possible. For example, steps described as sequential may be executed in parallel, order of steps may be varied, and steps may be modified, combined, added, or omitted. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

#### Detection of Inactive Client-side Browser Windows

[0054] In certain embodiments, application server 110 can detect when a user of browser 106 has inactivated a browser window by closing the window or navigating away from application 114, and can distinguish those situations from a case where the user has merely navigated to another web page of application 114. FIG. 5 illustrates a process 500 that can be performed by application server 110 for implementing this inactive window detection according to an embodiment of the present invention. In various embodiments, process 500 can be implemented in hardware, software, or a combination thereof. As software, process 500 can be encoded as program code stored on a machine-readable storage medium.

[0055] At block 502, application server 110 can maintain an “active” list of browser windows that are open in web browser 106. From the perspective of application server 110, a browser window is considered “active” if it is being used to access web application 114. The active list can be stored as part of the user session associated with the browser. In one set of embodiments, the entry for each window in the active list can include the window’s identifier and an indication of the current page being displayed in the window.

[0056] At block 504, application server 110 can receive a notification from browser 106 that an application page in an active browser window has been unloaded/destroyed. Such a notification may be received if, e.g., the window has been closed by a user, or if the user has navigated to a different web page within the window. In response, application server 110 can associate an expiration time with the active window, and can add the window to an “expiration” list of windows that are potentially inactive (i.e., no longer being used to access the application) (blocks 506, 508).

[0057] In addition, application server 110 can determine whether all of the windows in the active list are now in the expiration list (block 510). If not, the steps of blocks 504-510 can be repeated as more unload/destroy notifications are received by application server 110, thereby causing additional active windows to be added to the expiration list. If all of the active windows are now in the expiration list, application server 110 can perform special

processing to determine whether to free user session-level resources; this processing is discussed with respect to FIG. 7 below.

[0058] The processing of FIG. 5 is continued in process 600 of FIG. 6. At block 602, application server 110 can receive an HTTP request originating from a particular browser window that was previously placed in the expiration list at block 508. Since the receipt of this request indicates that the window is still being used to access application 114, application server 110 can conclude that the window is still active and can remove it from the expiration list (block 604). Application server 110 can also update the window's entry in the active list with the latest URL included in the received request.

[0059] At block 606, application server 110 can go through the remaining windows in the expiration list and compare, for each window, the time at which the request was received at block 602 with the expiration time associated with the window. If the time of the request is later than the expiration time for the window, application server 110 can identify the window as being inactive (i.e., no longer being used to access application 114) and can remove the window from both the active and expiration lists. Application server 110 can also free the memory resources (e.g., application state information, etc.) associated with that window.

[0060] If the time of the request is earlier than the expiration time for the window, application server 110 can leave the window in the expiration list. The window can remain in the expiration list until a new request is received from that window (indicating that the window is still active), or until a new request is received from a different window at a time later than the expiration time associated with the window (indicating that the window is most likely inactive).

[0061] In some cases when a user navigates from a first application page to a second application page in a browser window, application server 110 may receive a request for the second page before it receives the notification that the first page has been destroyed/unloaded. This is because HTTP does not guarantee that messages are received in the same order in which they were sent. To account for this, in certain embodiments application server 110 can keep track of the current page displayed in each window and can ignore unload notifications that pertain to outdated pages.

[0062] For example, assume browser window A is currently displaying application page 1. Further, assume that the user of window A navigates from application page 1 to application page 2, but application server 110 receives the request for (window A, page 2) before the

unload notification for (window A, page 1). Once the request for (window A, page 2) is received, application server 110 can update the active window list such that the current page for window A is page 2. When the unload notification for (window A, page 1) is subsequently received, application server 110 can ignore the unload notification since the current page for window A is not page 1 (rather, it is page 2). Thus, this prevents application server 110 from erroneously adding window A to the expiration list upon receiving the unload notification.

[0063] It should be appreciated that processes 500 and 600 are illustrative and that variations and modifications are possible. For example, steps described as sequential may be executed in parallel, order of steps may be varied, and steps may be modified, combined, added, or omitted. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

[0064] As noted above, in certain embodiments application server 110 can employ special processing to detect when the last remaining browser window for a user session has been inactivated/closed. This processing is illustrated in process 700 of FIG. 7. In various embodiments, process 700 can be implemented in hardware, software, or a combination thereof. As software, process 700 can be encoded as program code stored on a machine-readable storage medium.

[0065] At block 702, application server 110 can determine that an application page in the last remaining browser window for a user session (i.e., the last window in the active list) has been unloaded/destroyed. In one set of embodiments, this can be accomplished by comparing the set of windows in the active list and the expiration list are the same. Upon determining that the last remaining browser window has been unloaded/destroyed, application server 110 can modify a session timeout variable associated with the user session from a first time period (e.g., one hour) to a second time period shorter than the first (e.g., ten minutes), and can begin a countdown of the variable (block 704).

[0066] If a HTTP request is received from any window before the session timeout variable expires, application server 110 can conclude that the session is still active and can reset the session timeout variable to the first time period (blocks 706, 708). Further, processing can return to block 602 of FIG. 6.

[0067] However, if no request is received from any window before the session timeout variable expires, application server 110 can conclude that the last remaining window is

inactive (and thus, the entire user session is closed). Accordingly, application server 110 can free all of the resources associated with the window and the user session (block 710).

[0068] It should be appreciated that process 700 is illustrative and that variations and modifications are possible. For example, steps described as sequential may be executed in parallel, order of steps may be varied, and steps may be modified, combined, added, or omitted. One of ordinary skill in the art would recognize many variations, modifications, and alternatives.

[0069] FIG. 8 is a simplified block diagram illustrating a system environment 800 according to an embodiment of the present invention. As shown, system environment 800 can include one or more client computer systems 802, 804, 806, 808, a server computer system 810, a network 812, and a database 814. In various embodiments, client computer systems 802, 804, 806, 808, server computer system 810, and database 814 can be used to implement client 102, server 104, and data store 116 of FIG. 1 respectively.

[0070] Client computer systems 802, 804, 806, 808 can be general-purpose personal computers (e.g., desktop computers and/or laptop computers), tablet computers, smart phones, PDAs, and/or workstation computers. Alternatively, client computer systems 802, 804, 806, 808 can be any other type of electronic device capable of running a web browser such as browser 106 of FIG. 1. Although system environment 800 is shown with four client computer systems, it should be appreciated that any number of client computer systems can be supported.

[0071] Network 812 can be any type of network familiar to those skilled in the art that can support data communications using a network protocol, such as TCP/IP, SNA, IPX, AppleTalk, and the like. Merely by way of example, network 812 can be a local area network (LAN), such as an Ethernet network; a wide-area network; a virtual network, including without limitation a virtual private network (VPN); the Internet; a wireless network (e.g., a network operating under any of the IEEE 802.11 suite of protocols, the Bluetooth protocol known in the art, and/or any other wireless protocol); and/or any combination of these and/or other networks.

[0072] Server computer system 810 can be a general purpose computer, a specialized server computer (e.g., PC server, UNIX server, mainframe server, rack-mounted server, etc.), a server cluster, or any other appropriate arrangement and/or combination of computers. Server computer system 810 can run any consumer or server operating system such as

Microsoft Windows, Apple OS X, Linux, Solaris, and the like. Server computer system 810 can also run any of a variety of server applications and/or mid-tier applications, including application server 110 and web application 114 of FIG. 1.

[0073] Database 814 can reside on a storage medium local to (and/or resident in) one or more of computer systems 802, 804, 806, 808, and 810. Alternatively, database 814 can be remote from any or all of computer systems 802, 804, 806, 808, 810, and/or in communication (e.g., via network 812) with one or more of these. In one set of embodiments, database 814 can reside in a storage-area network (SAN) familiar to those skilled in the art. Similarly, any necessary files for performing the functions attributed to computer systems 802, 804, 806, 808, 810 can be stored locally on the respective computer and/or remotely on database 814, as appropriate. In one set of embodiments, database 814 can be a relational database, such as Oracle 11g available from Oracle Corporation, that is adapted to store, update, and retrieve data in response to SQL-formatted commands.

[0074] FIG. 9 is a simplified block diagram illustrating a computer system 900 according to an embodiment of the present invention. In various embodiments, computer system 900 can be used to implement any of computer systems 802, 804, 806, 808, and 810 described with respect to system environment 800 above. As shown, computer system 900 can include hardware elements that are electrically coupled via a bus 924. The hardware elements can include one or more central processing units (CPUs) 902, one or more input devices 904 (e.g., a mouse, a keyboard, etc.), and one or more output devices 906 (e.g., a display device, a printer, etc.). Computer system 900 can also include one or more storage devices 908. By way of example, the storage device(s) 908 can include devices such as disk drives, optical storage devices, and solid-state storage devices such as a random access memory (RAM) and/or a read-only memory (ROM), which can be programmable, flash-updateable and/or the like.

[0075] Computer system 900 can additionally include a computer-readable storage media reader 912, a communications subsystem 914 (e.g., a modem, a network card (wireless or wired), an infra-red communication device, etc.), and working memory 918, which can include RAM and ROM devices as described above. In some embodiments, computer system 900 can also include a processing acceleration unit 916, which can include a digital signal processor (DSP), a special-purpose processor, and/or the like.

[0076] Computer-readable storage media reader 912 can be connected to a computer-readable storage medium 910, together (and, optionally, in combination with storage device(s) 908) comprehensively representing remote, local, fixed, and/or removable storage devices plus storage media for temporarily and/or more permanently containing computer-readable information. Communications system 914 can permit data to be exchanged with network 812 and/or any other computer described above with respect to system environment 800.

[0077] Computer system 900 can also comprise software elements, shown as being currently located within working memory 918, including an operating system 920 and/or other code 922, such as an application program (which may be a client-side web browser, middle tier/server application, etc.). It should be appreciated that alternative embodiments of computer system 900 can have numerous variations from that described above. For example, customized hardware can be used and particular elements can be implemented in hardware, software, or both. Further, connection to other computing devices such as network input/output devices can be employed.

[0078] Computer readable storage media for containing code, or portions of code, executable by computer system 900 can include any appropriate media known or used in the art, such as but not limited to volatile/non-volatile and removable/non-removable media. Examples of computer-readable storage media include RAM, ROM, EEPROM, flash memory, CD-ROM, digital versatile disk (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, and any other medium that can be used to store data and/or program code and that can be accessed by a computer.

[0079] Although specific embodiments of the invention have been described above, various modifications, alterations, alternative constructions, and equivalents are within the scope of the invention. For example, although embodiments of the present invention have been described with respect to certain flow diagrams and steps, it should be apparent to those skilled in the art that the scope of the present invention is not limited to the described diagrams/steps.

[0080] Further, although embodiments of the present invention have been described using a particular combination of hardware and software, it should be recognized that other combinations of hardware and software are also within the scope of the present invention.

[0081] The specification and drawings are, accordingly, to be regarded in an illustrative rather than restrictive sense. It will be evident that additions, subtractions, and other modifications may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the following claims.

WHAT IS CLAIMED IS:

- 1                   1.       A method comprising:  
2                   receiving, by a server computer system, a first request originating from a  
3 window of a web browser running on a client computer system, the first request including a  
4 Uniform Resource Locator (URL) and a first window identifier;  
5                   generating, by the server computer system, a loopback identifier;  
6                   storing, by the server computer system, the loopback identifier in a user  
7 session associated with the request;  
8                   generating, by the server computer system, a script that includes the URL, the  
9 first window identifier, and the loopback identifier; and  
10                  sending, by the server computer system, the script to the window that  
11 originated the first request.
- 1                   2.       The method of claim 1 wherein the script is executable by the web  
2 browser running on the client computer system to facilitate determination of whether the  
3 window that originated the first request is a new window or an existing window.
- 1                   3.       The method of claim 1 wherein the loopback identifier is a randomly  
2 generated number.
- 1                   4.       The method of claim 1 wherein the first request is a Hypertext Transfer  
2 Protocol (HTTP) request.
- 1                   5.       The method of claim 4 wherein the script is sent to the window that  
2 originated the first request only if the first request is an HTTP GET or HEAD request.
- 1                   6.       The method of claim 1 wherein the script is configured to, on the client  
2 computer system:  
3                   determine whether the first window identifier matches an identifier of the  
4 window that originated the first request; and  
5                   based on the determining, send a second request to the server computer  
6 system.
- 1                   7.       The method of claim 6 wherein if the first window identifier matches  
2 an identifier of the window that originated the first request, the second request includes the  
3 URL, the loopback identifier, and the first window identifier.

1                   8.       The method of claim 7 wherein if the first window identifier does not  
2 match an identifier of the window that originated the first request, the second request includes  
3 the URL, the loopback identifier, and a second window identifier distinct from the first  
4 window identifier.

1                   9.       The method of claim 8 further comprising:  
2                   receiving, by the server computer system, the second request;  
3                   matching, by the server computer system, the loopback identifier included in  
4 the second request with the loopback identifier stored in the user session; and  
5                   removing the loopback identifier from the user session.

1                   10.      The method of claim 9 further comprising:  
2                   if the second request includes the first window identifier, determining, by the  
3 server computer system, that the window that originated the first request is a new window;  
4 and

5                   if the second request includes the second window identifier, determining, by  
6 the server computer system, that the window that originated the first request is an existing  
7 window.

1                   11.      The method of claim 10 further comprising:  
2                   rendering, by the server computer system, content for the URL based on  
3 whether the window that originated the first request is a new window or an existing window;  
4 and  
5                   sending, by the server computer system, the rendered content to the window  
6 that originated the first request.

1                   12.      A non-transitory computer readable storage medium having stored  
2 thereon instructions executable by a processor, the instructions comprising:  
3                   instructions that cause the processor to receive a first request originating from  
4 a window of a web browser running on a client computer system, the first request including a  
5 Uniform Resource Locator (URL) and a first window identifier;  
6                   instructions that cause the processor to generate a loopback identifier;  
7                   instructions that cause the processor to store the loopback identifier in a user  
8 session associated with the request;

9 instructions that cause the processor to generate a script that includes the URL,  
10 the first window identifier, and the loopback identifier; and  
11 instructions that cause the processor to send the script to the window that  
12 originated the first request.

1 13. The non-transitory computer readable storage medium of claim 12  
2 wherein the script is executable by the web browser running on the client computer system to  
3 facilitate determination of whether the window that originated the first request is a new  
4 window or an existing window.

1 14. A server computer system comprising:  
2 a processor configured to:  
3 receive a first request originating from a window of a web browser running on  
4 a client computer system, the first request including a Uniform Resource Locator (URL) and  
5 a first window identifier;  
6 generate a loopback identifier;  
7 store the loopback identifier in a user session associated with the request;  
8 generate a script that includes the URL, the first window identifier, and the  
9 loopback identifier; and  
10 send the script to the window that originated the first request.

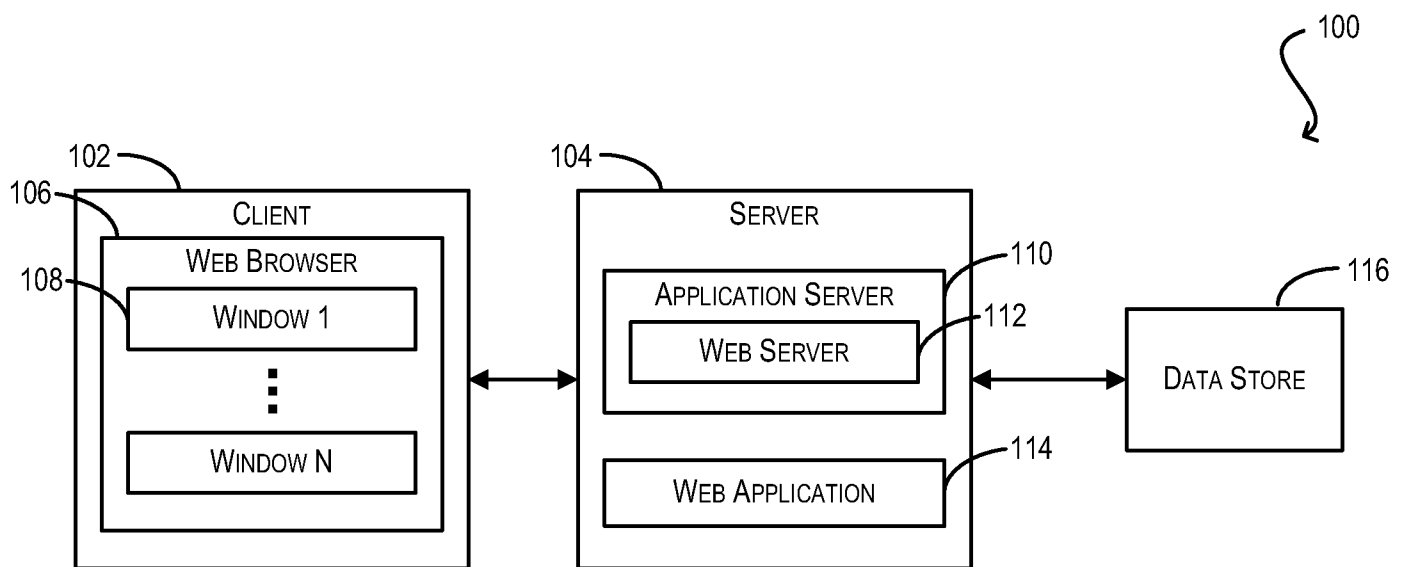
1 15. The server computer system of claim 14 wherein the script is  
2 executable by the web browser running on the client computer system to facilitate  
3 determination of whether the window that originated the first request is a new window or an  
4 existing window.

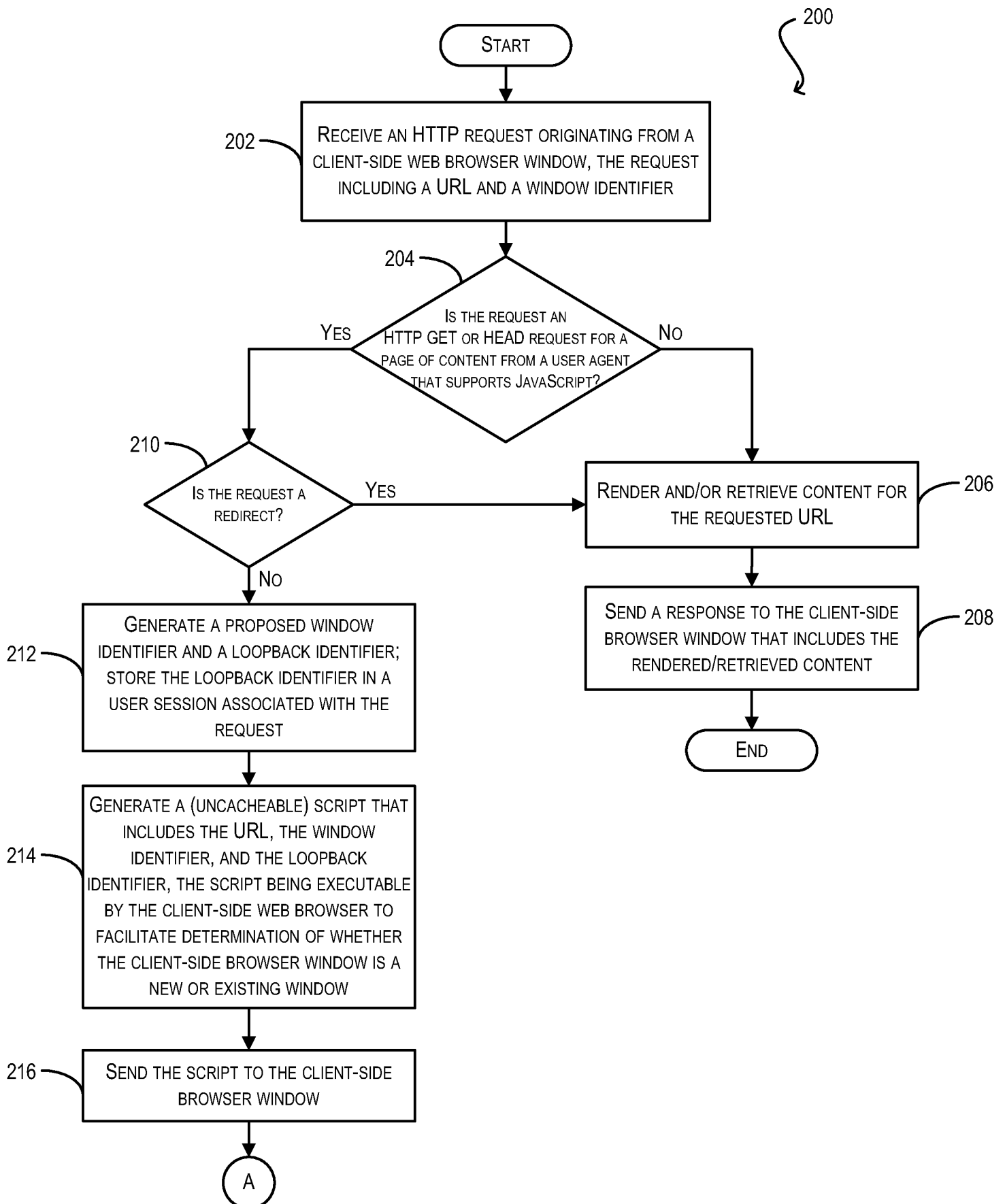
1 16. A system comprising:  
2 means for receiving a first request originating from a window of a web  
3 browser running on a client computer system, the first request including a Uniform Resource  
4 Locator (URL) and a first window identifier;  
5 means for generating a loopback identifier;  
6 means for storing the loopback identifier in a user session associated with the  
7 request;  
8 means for generating a script that includes the URL, the first window  
9 identifier, and the loopback identifier; and

10 means for sending the script to the window that originated the first request.

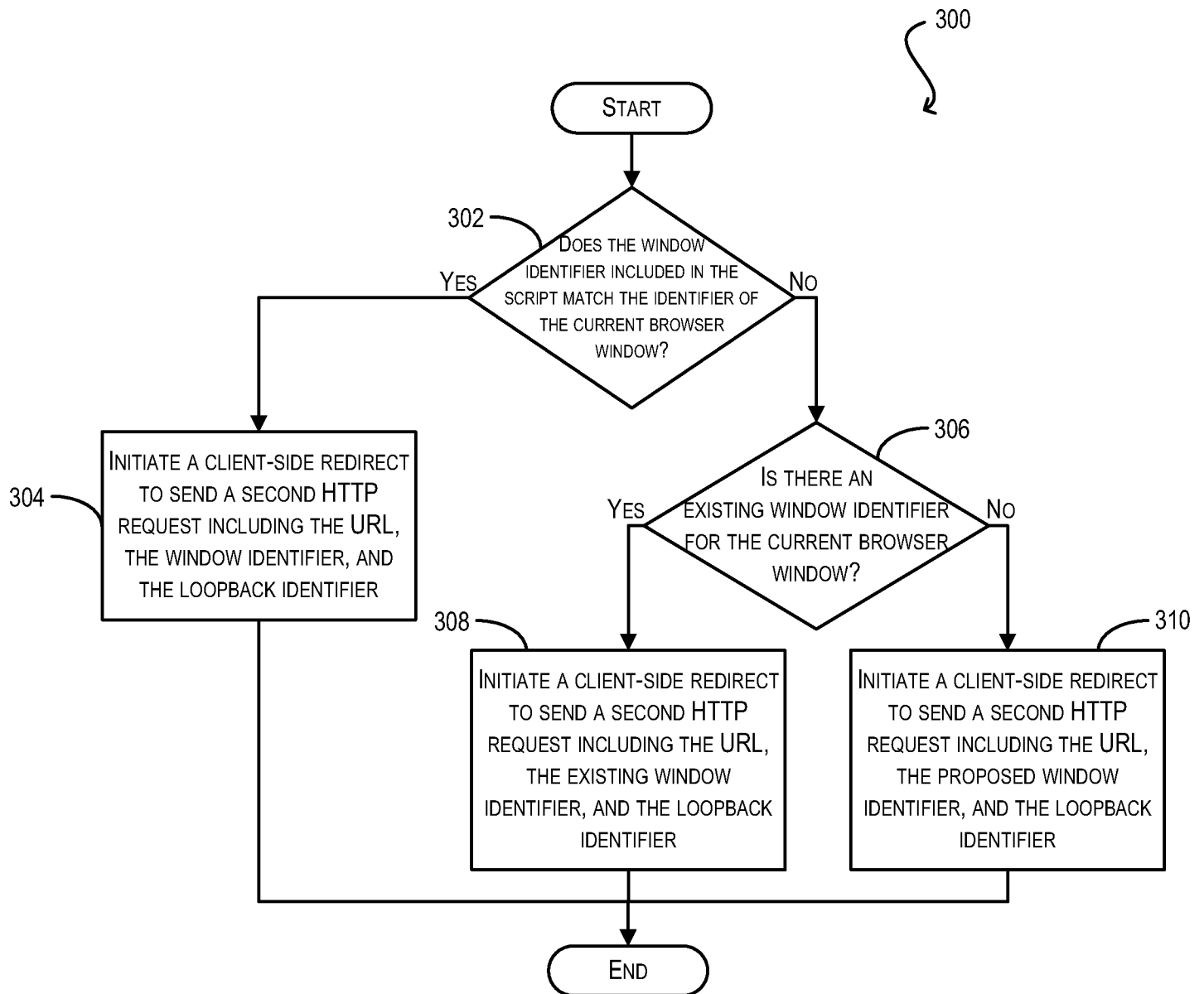
1 17. The system of claim 16 wherein the script is executable by the web  
2 browser running on the client computer system to facilitate determination of whether the  
3 window that originated the first request is a new window or an existing window.

1/9

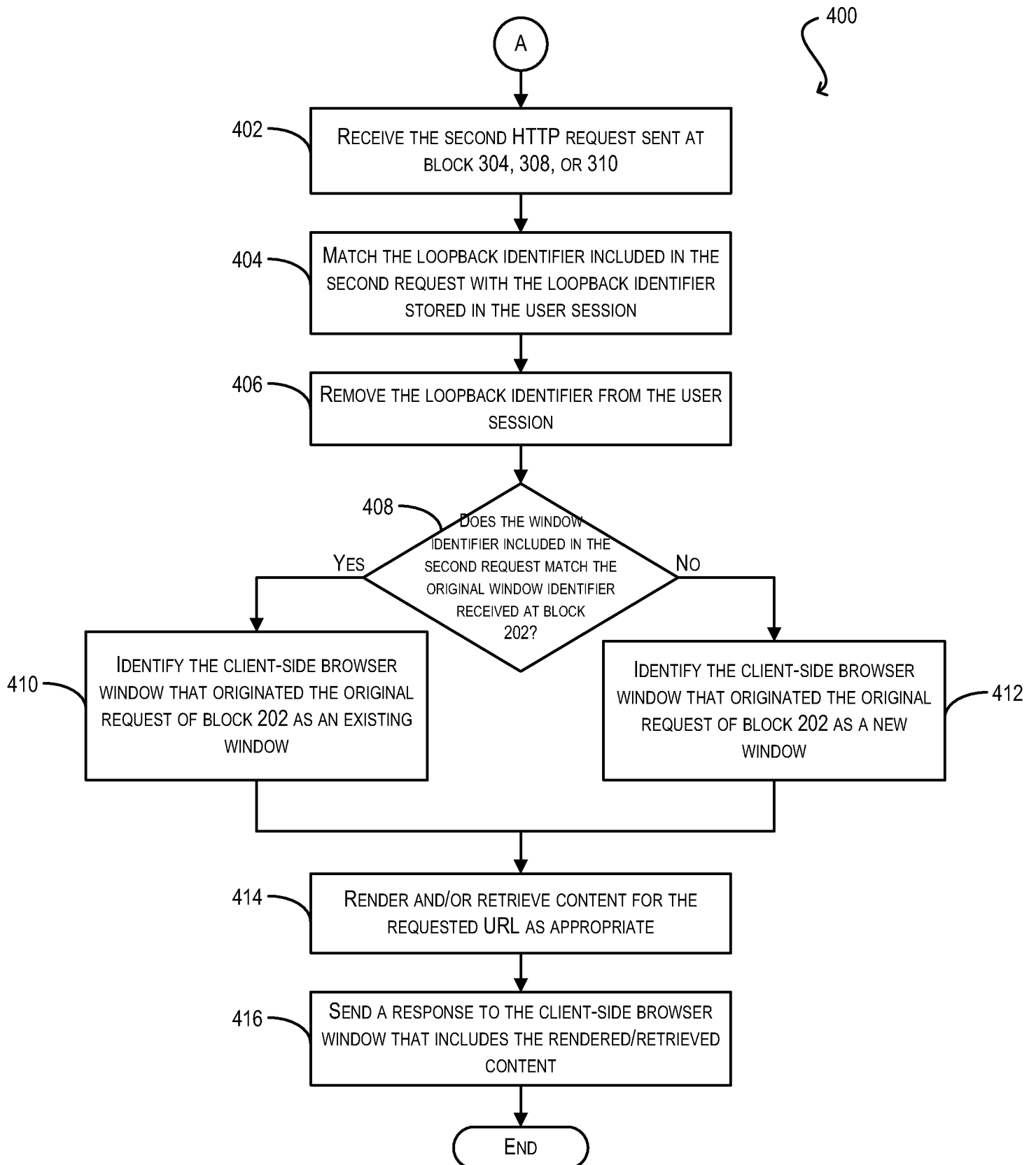
**FIG. 1**

**FIG. 2**

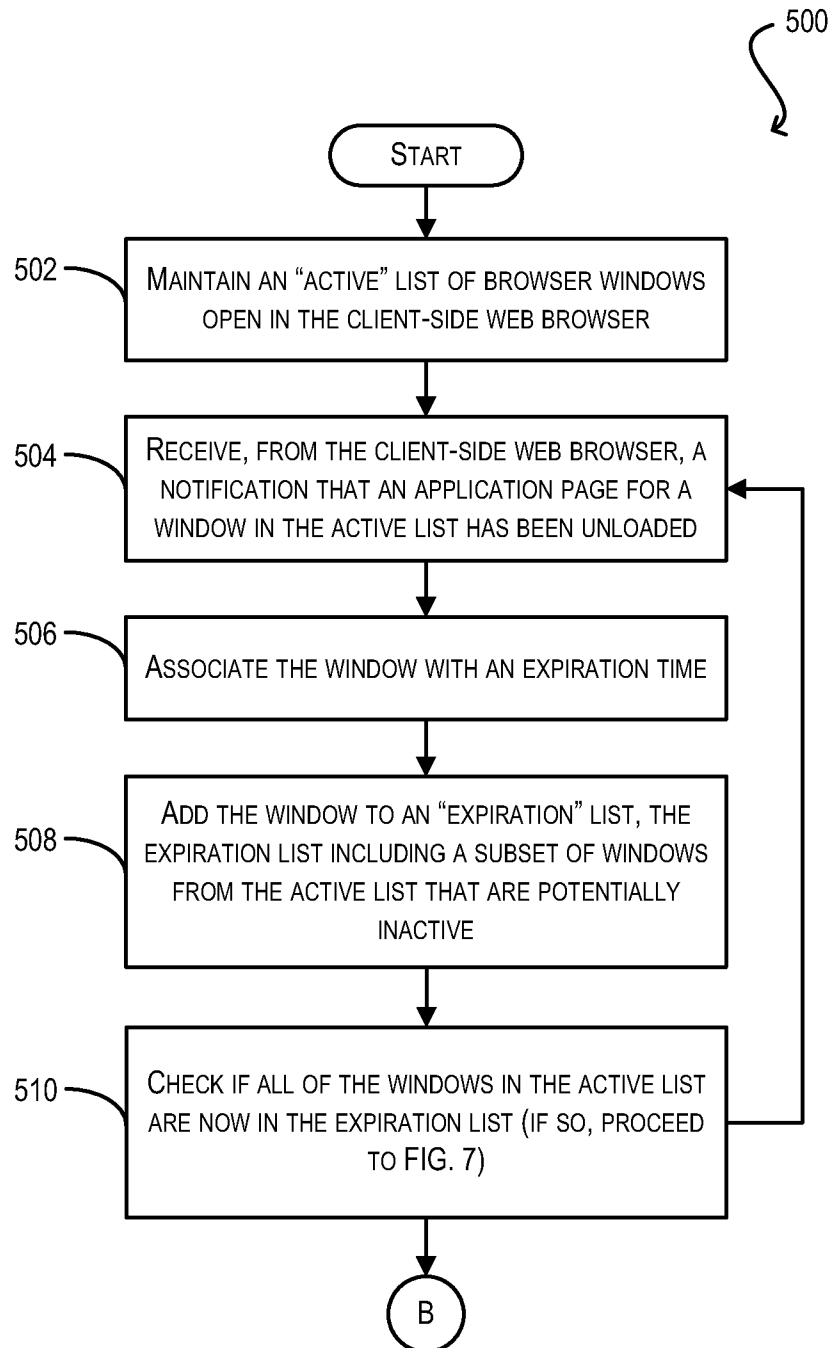
3/9

**FIG. 3**

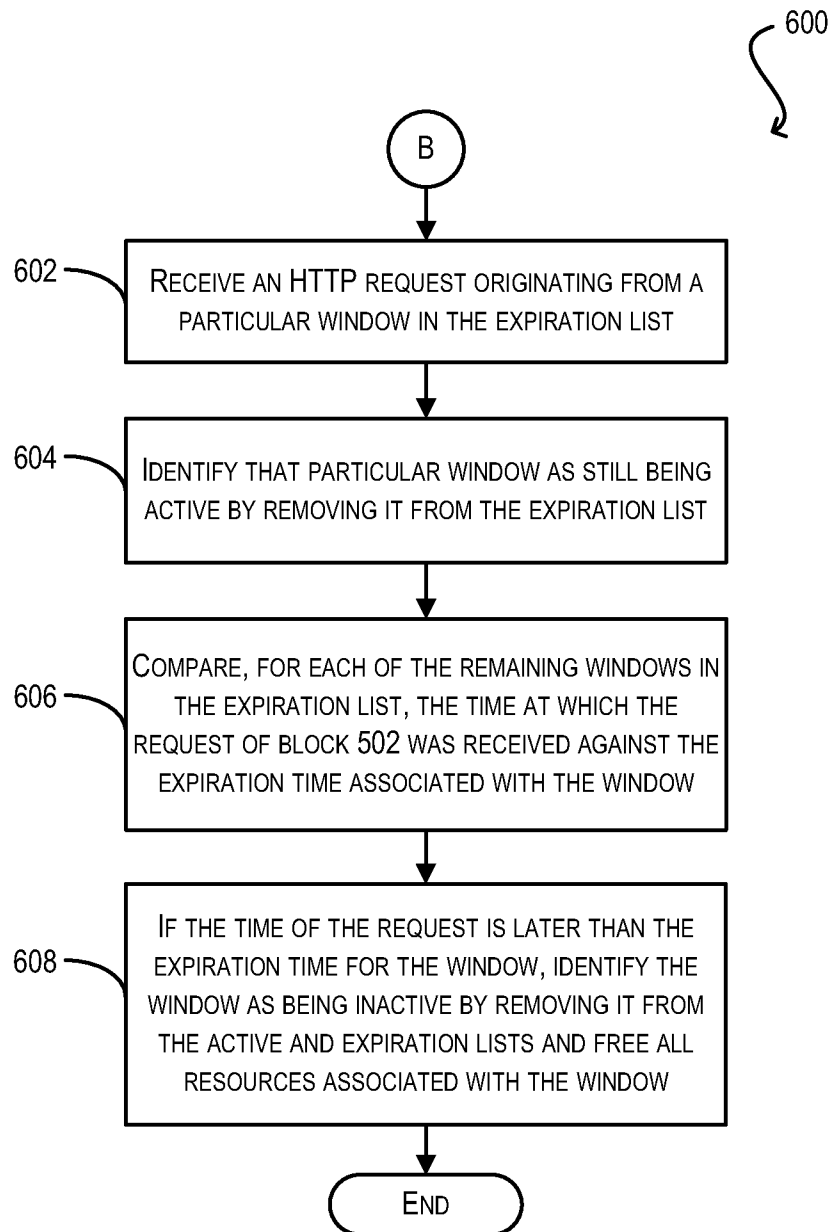
4/9

**FIG. 4**

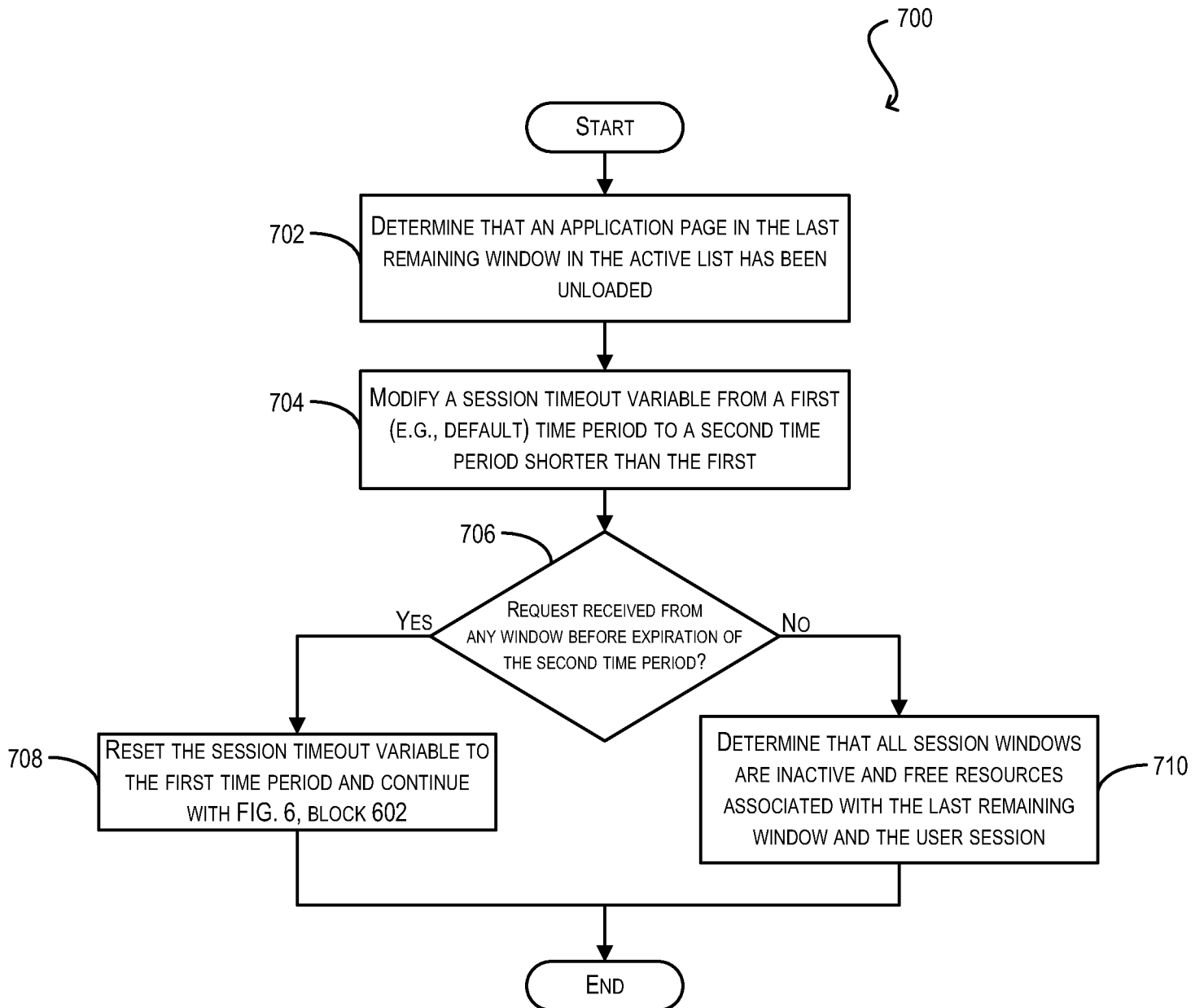
5/9

**FIG. 5**

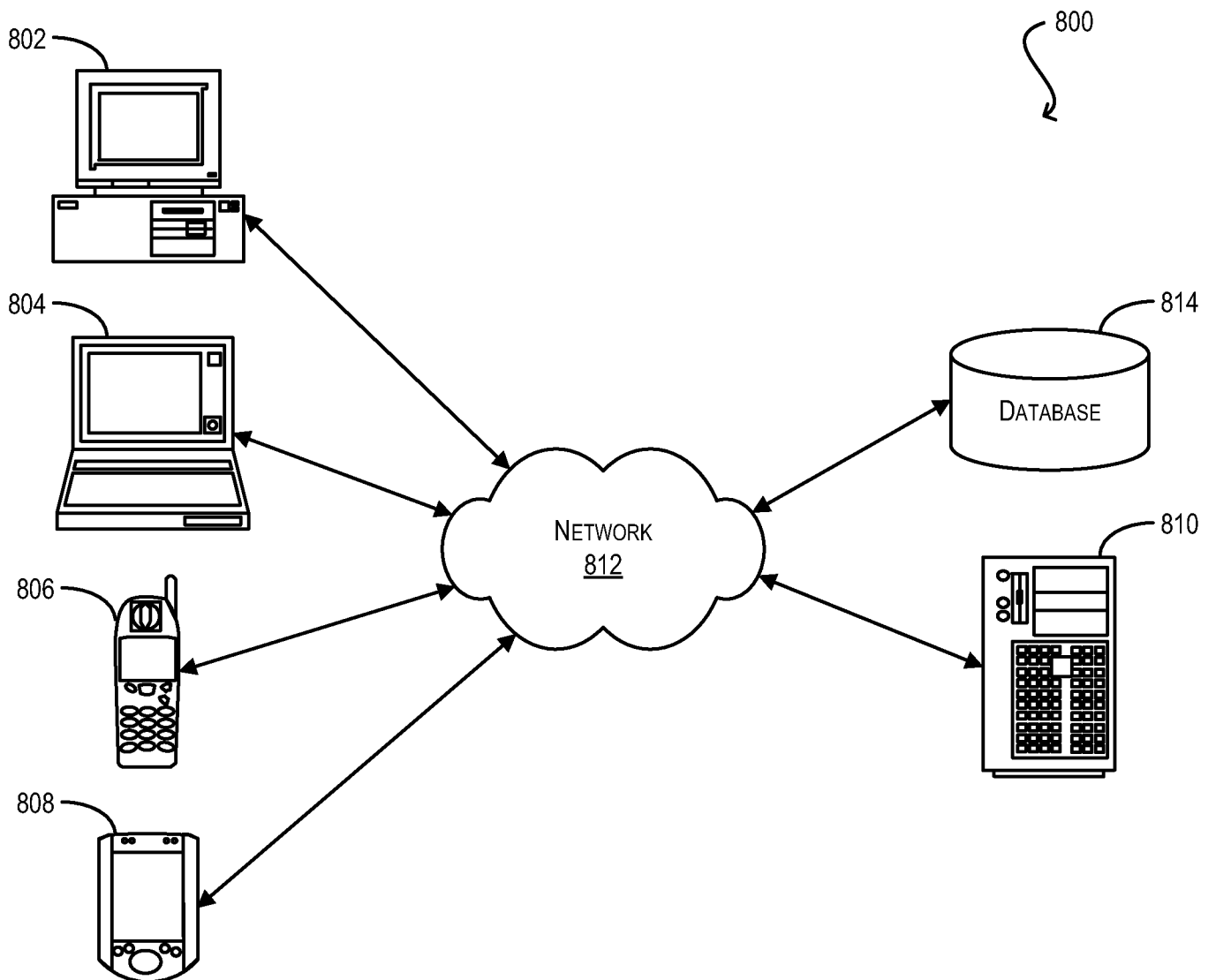
6/9

**FIG. 6**

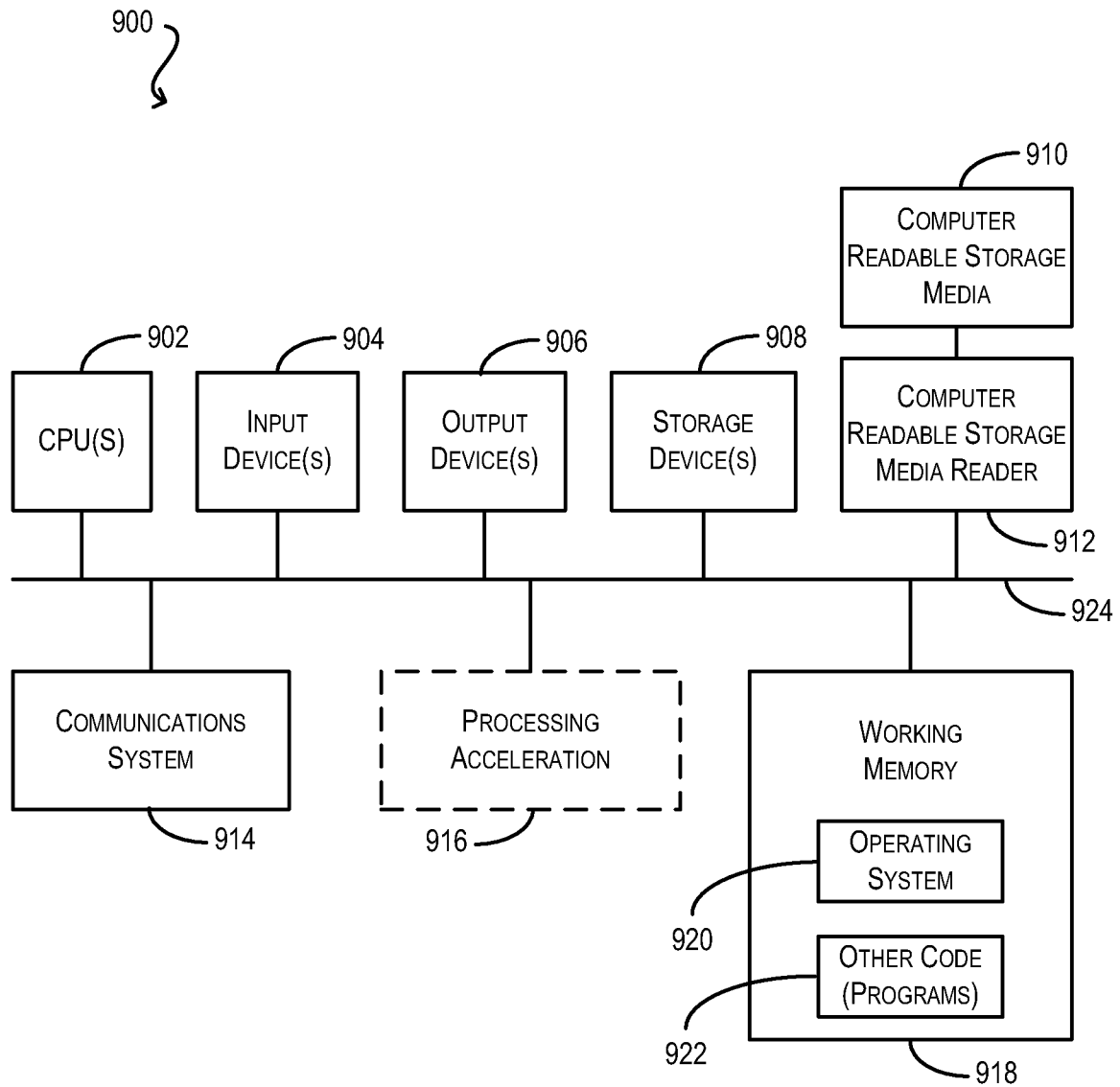
7/9

**FIG. 7**

8/9

**FIG. 8**

9/9

**FIG. 9**

# INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2012/020106

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                          |                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>A. CLASSIFICATION OF SUBJECT MATTER</b><br>INV. G06F17/30<br>ADD.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                          |                                                                                                                           |
| According to International Patent Classification (IPC) or to both national classification and IPC                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                          |                                                                                                                           |
| <b>B. FIELDS SEARCHED</b><br>Minimum documentation searched (classification system followed by classification symbols)<br>G06F                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                          |                                                                                                                           |
| Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                          |                                                                                                                           |
| Electronic data base consulted during the international search (name of data base and, where practical, search terms used)<br>EP0-Internal                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                          |                                                                                                                           |
| <b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                          |                                                                                                                           |
| Category*                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Citation of document, with indication, where appropriate, of the relevant passages                                                                       | Relevant to claim No.                                                                                                     |
| X                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | EP 1 041 496 A2 (IBM [US])<br>4 October 2000 (2000-10-04)<br>paragraphs [0001] - [0049]; figures 1-5<br>-----                                            | 1-17                                                                                                                      |
| A                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | US 6 052 730 A (FELCIANO RAMON M [US] ET AL)<br>18 April 2000 (2000-04-18)<br>the whole document<br>-----<br><div style="text-align: center;">-/--</div> | 1-17                                                                                                                      |
| <div style="display: flex; justify-content: space-between;"> <span><input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C.</span> <span><input checked="" type="checkbox"/> See patent family annex.</span> </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                          |                                                                                                                           |
| * Special categories of cited documents :                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                          |                                                                                                                           |
| <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </div> <div style="width: 45%;"> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&amp;" document member of the same patent family</p> </div> </div> |                                                                                                                                                          |                                                                                                                           |
| Date of the actual completion of the international search<br><div style="text-align: center; font-size: 1.2em;">29 February 2012</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                          | Date of mailing of the international search report<br><div style="text-align: center; font-size: 1.2em;">07/03/2012</div> |
| Name and mailing address of the ISA/<br>European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                          | Authorized officer<br><div style="text-align: center; font-size: 1.2em;">Moon, Timothy</div>                              |

# INTERNATIONAL SEARCH REPORT

International application No  
PCT/US2012/020106

| C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT |                                                                                                                                                                                                                                                                                                                                                                                           |                       |
|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Category*                                            | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                                                                                                                                                        | Relevant to claim No. |
| A                                                    | <p>ZHONG TANG ET AL: "Closed-loop load balancing: comparison of a discrete event simulation with experiments", AMERICAN CONTROL CONFERENCE, 2005. PROCEEDINGS OF THE 2005 PORTLAND, OR, USA JUNE 8-10, 2005, IEEE, PISCATAWAY, NJ, USA, 8 June 2005 (2005-06-08), pages 2721-2726, XP010820208, DOI: 10.1109/ACC.2005.1470380 ISBN: 978-0-7803-9098-0 the whole document</p> <p>-----</p> | 1-17                  |

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2012/020106

| Patent document<br>cited in search report | Publication<br>date | Patent family<br>member(s) | Publication<br>date         |
|-------------------------------------------|---------------------|----------------------------|-----------------------------|
| EP 1041496                                | A2                  | 04-10-2000                 | CA 2299039 A1 30-09-2000    |
|                                           |                     |                            | CN 1268700 A 04-10-2000     |
|                                           |                     |                            | EP 1041496 A2 04-10-2000    |
|                                           |                     |                            | JP 2000330957 A 30-11-2000  |
|                                           |                     |                            | KR 20010014581 A 26-02-2001 |
|                                           |                     |                            | TW 578066 B 01-03-2004      |
| -----                                     |                     |                            |                             |
| US 6052730                                | A                   | 18-04-2000                 | NONE                        |
| -----                                     |                     |                            |                             |