



- (51) **International Patent Classification:** Not classified
- (21) **International Application Number:**
PCT/US2012/045846
- (22) **International Filing Date:**
6 July 2012 (06.07.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/284,129 28 October 2011 (28.10.2011) US
- (71) **Applicant (for all designated States except US):** **YAHOO! INC.** [US/US]; 701 First Avenue, Sunnyvale, CA 94089 (US).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** **GOYAL, Prabhakar** [US/US]; 1342 Fisher Hawk Drive, Sunnyvale, CA 94087 (US). **GUPTA, Sanjay, Kumar** [US/US]; 1267 Rembrandt, Sunnyvale, CA 94087 (US). **DODDA, Rampradeep** [IN/US]; 3421 Granada Avenue, Santa Clara, CA 95051 (US).
- (74) **Agent:** **PENILLA, Albert, S.**; Martine Penilla Group, LLP, 710 Lakeway Drive, Suite 200, Sunnyvale, CA 94085 (US).

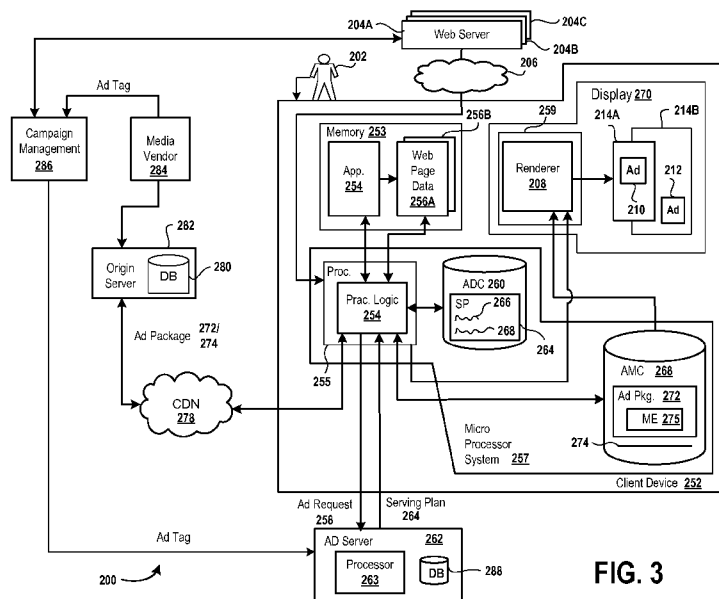
(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, 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, 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).

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) **Title:** METHODS AND SYSTEMS FOR FACILITATING CACHING OF ADVERTISEMENTS

**FIG. 3**

(57) **Abstract:** Methods and Systems for facilitating caching of advertisements are described. The methods include receiving an ad request and generating a serving plan in response to said receiving the ad request. The serving plan includes multiple cache lines that identify corresponding advertisements. Each of the cache lines is associated with a corresponding expiration. Also, each cache line is readable by logic to display one of the advertisements corresponding to the cache line. The cache lines include a first cache line. The first cache line includes ad information, which is associated with a first one of the advertisements. Moreover, the ad information includes one or more parameters for expiration of the first cache line of a client device. The client device is configured to display the plurality of advertisements. The method includes sending the serving plan to the client device to store in the client device.

**METHODS AND SYSTEMS FOR FACILITATING CACHING OF
ADVERTISEMENTS
TECHNICAL FIELD**

[0001] The present disclosure relates generally to methods and systems for facilitating caching of advertisements.

BACKGROUND

[0002] The rapidly expanding presence of the Internet has produced an increased recognition of the importance of web advertising. As compared to more traditional media such as television or radio, advertising on the Web is based on web page views and is more easily quantifiable. In large part, each page view represents a transaction between a client (or user's) computer and a web server. These individual client-server interactions permit more deterministic measures of the reach of particular advertising campaigns. Accordingly, it is important that communication between a client and a server is maintained at all times.

[0003] However, sometimes, communication is lost as a result of a variety of conditions, such as operation of the client at a location where signal strength is zero, damage to the server, or damage to a communication link between the client and the server. Moreover, although there is communication between the client and the server, advertisements can consume a large amount of network bandwidth.

[0004] It is in this content that various embodiments of the present invention arise.

SUMMARY

[0005] The following detailed description together with the accompanying drawings will provide a better understanding of the nature and advantages of various embodiments of the present invention.

[0006] In one embodiment, a client sends an ad request to an ad server. The ad server receives the ad request to generate a serving plan. The serving plan is a plan to serve a number of advertisements for a number of expiration time periods, which are indicated in the serving plan. The ad server receives the serving plan and executes one or more cache lines in the serving plan to retrieve one or more ad media packages locally, such as from the client device, or from an origin server. The one or more ad media packages are provided to a Tenderer that renders the one or more ad media packages. One or more advertisements are generated upon rendering the one or more corresponding ad media packages and are

displayed on the client device. If communication is lost or is intermittently lost between the client device and the ad server after the serving plan is received by the client device, the client device can execute the serving plan to display the one or more advertisements on the client device.

[0007] In another embodiment, a method for facilitating caching of advertisements is described. The method includes receiving an ad request and generating the serving plan in response to receiving the ad request. The serving plan is generated by the ad server. The serving plan includes multiple cache lines that identify corresponding advertisements. Each of the cache lines is associated with a corresponding expiration. Also, each cache line is readable by logic to display one of the advertisements corresponding to the cache line. The plurality of cache lines includes a first cache line. The first cache line includes ad information, which is associated with a first one of the advertisements. Moreover, the ad information includes one or more parameters for expiration of the first cache line of a client device. The client device is configured to display the plurality of advertisements. The method includes sending the serving plan to the client device to store in the client device.

[0008] In one embodiment, a method for facilitating caching of advertisements is described. The method includes determining whether web page data is received. The method further includes generating an ad request in response to determining that the web page data is received. The method also includes sending the ad request and receiving the serving plan in response to sending the ad request. The method includes storing the serving plan within the client device and periodically updating the serving plan based on an ad refresh.

[0009] In some embodiments, a system for facilitating caching of advertisements is described. The system includes a client device for executing an application to allow access to web page data. The client device is configured to generate an ad request upon downloading the web page data. The client device includes an ad decision cache (ADC). The system further includes an ad server for receiving the ad request and for generating the serving plan in response to receiving the ad request. The serving plan includes the first cache line. The first cache line includes ad information, which includes one or more parameters for expiration of the first cache line from the ADC.

DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a flowchart of a method for generating and sending a serving plan, in accordance with one embodiment of the present invention.

[0011] Figure 2A is a flowchart of a method for generating an ad request and receiving a serving plan in response to the ad request, in accordance with one embodiment of the present invention.

[0012] Figure 2B is a flowchart of a method for generating an ad request and receiving a serving plan in response to the ad request, in accordance with another embodiment of the present invention.

[0013] Figure 3 is a block diagram of a system for facilitating caching of advertisements, in accordance with one embodiment of the present invention.

[0014] Figure 4 is a flowchart of a method for determining whether to request an ad media package and for requesting the ad media package, in accordance with one embodiment of the present invention.

[0015] Figure 5 is a flowchart of a method for determining whether to request a subsequent version of an ad media package and generating the request, in accordance with one embodiment of the present invention.

[0016] Figure 6 is a flowchart of a method for executing an ad refresh, in accordance with one embodiment of the present invention.

[0017] Figure 7 illustrates suitable components of a computer, in accordance with one embodiment of the present invention.

[0018] Figure 8 is a diagram illustrating components of a server, in accordance with one embodiment of the present invention.

DESCRIPTION OF EXAMPLE EMBODIMENT(S)

[0019] The following example embodiments and their aspects are described and illustrated in conjunction with apparatuses, methods, and systems which are meant to be illustrative examples, not limiting in scope.

[0020] FIG. 1 is an embodiment of a flowchart of a method 100 for generating and sending a serving plan. The method 100 is performed by an ad server, described below with reference to Figure 3. In operation 102, it is determined whether an ad request is received. In various embodiments, an ad request includes a request for identification of advertisement data and a space identification (ID) categorizing web page data. Examples of categories of web page data include arts, business, computers, arts-movies, and arts-music. In one embodiment, an ad request includes an ad position of embedding the advertisement data within the web page data, a number of ads to be embedded within the web page data, age of content of the web

page data, a user ID identifying a user of a client device, and/or type of the content. In some embodiments, the user ID is provided by the user via an input device to log on to the client device. In one embodiment, the user ID is provided by the user via the input device to access an application, which is described below. The input device is also described below.

Examples of content types include audio, video, text, graphic, animation, and mixed media.

[0021] In some embodiments, the web page data is rendered to display a web page on the client device. In one embodiment, advertisement data includes an ad media package that is associated with an advertisement. For example, advertisement data is rendered to display an advertisement on the client device. An advertisement may be in the form of a graphical ad, such as, a banner ad, a text only ad, an image ad, an audio ad, a video ad, or a combination thereof. An ad media package is described below.

[0022] It should be noted that in several embodiments, categorizing and identifying are used interchangeably. For example, web page data is categorized to a level to allow identification of the web page data. As another example, advertisement data is categorized to a level to distinguish the advertisement data from other advertisement data. In such embodiments, a category is subcategorized to additional levels to identify web page data or advertisement data. In one embodiment, an arts category of web page data is further categorized to an arts-music category, the arts-music category is further categorized to an arts-music-artist name category, which is further categorized to arts-music-artist name-album category. The arts-music-artist name-album category is further categorized into arts-music-artist name-album-song category, which identifies web page data or advertisement data.

[0023] In some embodiments, content type is used in addition to category type to identify advertisement data or web page data. For example, a song may be in the form of text, animation, video, or audio, or a combination thereof.

[0024] In an operation 104, a serving plan is generated in response to reception of an ad request. The serving plan includes one or more cache lines of a collection of cache lines. In one embodiment, the serving plan includes a first cache line associated with a first advertisement and a second cache line associated with a second advertisement. In some embodiments, the first cache line includes a first space ID identifying a web page in which advertisement data, which is rendered to display the first advertisement, is to be embedded; an ad position indicating a position of embedding the advertisement data within the web page; an ad tag categorizing the advertisement data; a maximum number of impressions of the first advertisement; a constraint on the first advertisement; and an expiration time of the first cache

line. The first space ID is the same as that received within the ad request. In various embodiments, an expiration time of a cache line is a time of deletion of the cache line from a cache of the client device.

[0025] In one embodiment, the first cache line includes a maximum number of impressions of the first advertisement, an expiration time of the first cache line, or a constraint on the first advertisement.

[0026] In some embodiments, the ad server avoids sending an ad media package to a processing logic of the client device in response to the ad request. Rather, the ad server sends the serving plan. Such avoidance facilitates to reduce a delay in reception of the ad media package from the ad server. The delay may be a result of a limited amount of network bandwidth between the ad server and the client device. Moreover, any delay in communicating an ad media package is reduced by caching the ad media package in an ad media cache (AMC), described below.

[0027] An example of a serving plan is provided as follows:

```
2345667 1:column:Chevy:1 00:freq_cap=5 per hour: 1304626643 (6 hours)
2345667 1:column:Ford:100:None:1304626643 (6 hours)
2345667 1:full_page:Darfur:100:freq_cap=2 per hour: 1304626643 (6 hours)
2345667 1:full_page:HP:100:None:6 hours
33456286:column:Chevy:100:freq_cap=5 per hour: 1304626643 (6 hours)
2345667 1:column:Chevy:1 00:freq_cap=5 per hour: 1304626643 (6 hours)
2345667 1:full_page:Darfur:100:freq_cap=2 per hour: 1304626643 (6 hours)
2345667 1:full_page:HP:100:None:1304626643 (6 hours).
```

It should be noted that each cache line in the serving plan has a format of [content category/spaceid] :[ad position] :[ad_tag] :[max_impressions] :[constrain] :[expiration time]. It should further be noted that 'Chevy', 'Ford', 'Darfur', and 'HP' are examples of ad codes received from the ad server. An ad code is an example of an ad tag.

[0028] In one embodiment, an expiration time is in an absolute format. In this embodiment, the expiration time is measured from a certain point in time in history. For example, an expiration time is 6 hours from 8 am on October 11, 2011.

[0029] In some embodiments, the format of each cache line may be different than that shown. For example, the order of parameters in each cache line is different than that shown in the serving plan example above.

[0030] In various embodiments, all expiration times of all cache lines within a serving plan

are the same. In other embodiments, an expiration time of a cache line in a serving plan is different than an expiration time of any remaining cache lines in the serving plan.

[0031] In some embodiments, an ad position is a center position within a web page, a top left position within the web page, a bottom right position within the web page, a left column position within the web page, a right column position within the web page, a top row position within the web page, or a bottom row position within the web page. In one embodiment, an ad position is a position of an entire web page. In such an embodiment, an advertisement overlays and covers an entire web page. An ad tag may take one or more forms, such as, a numerical form, an alphanumeric form, or an alphabetical form. In one embodiment, an ad tag identifies a version, such as a current version or a subsequent version, of an advertisement. A media vendor, described below, creates a link between an ad tag and an ad media package, which is further described below.

[0032] An impression is an instance of an advertisement. In several embodiments, the same advertisement data may be rendered to display several impressions of the same advertisement. A maximum number of impressions within a cache line is a limit on a number of impressions of an advertisement to be displayed. A number of impressions of an advertisement displayed cannot exceed a maximum number of impressions. Once a maximum number of impressions of an advertisement is reached, a client device, which is described below, ceases to display any more impressions of the advertisement.

[0033] In one embodiment, a constraint on an advertisement includes a frequency of display of the advertisement on an associated web page. A web page is associated with an advertisement when the advertisement is designated, by a campaign management system, to be displayed on the web page. The campaign management system is described below.

[0034] In some embodiments, a cache line is deleted from an ad decision cache (ADC) by the processing logic when an expiration time mentioned in the cache line is reached or a maximum number of impressions mentioned in the cache line is reached. The ADC is described below. A maximum number of impressions is reached when an advertisement is displayed on a display device for the maximum number of times.

[0035] In several embodiments, until the expiration time is reached, the processing logic processes the serving plan and avoids sending an ad request to the ad server. In other embodiments, the processing logic periodically checks whether the maximum number of impressions of a cache line or the expiration time of the cache line is reached to determine whether to delete the cache line from the ADC. In some embodiments, a cache line is

deleted, by the processing logic, from the ADC regardless of whether an expiration time mentioned in the cache line is reached or a maximum number of impressions mentioned in the cache line is reached. In such embodiments, a storage capacity of the ADC is exceeded. In one embodiment, the ADC has a storage capacity of storing 256 to 512 cache lines.

[0036] In various embodiments, the processing logic also counts a number of impressions displayed on the display device and stores the count in a cache line. In various embodiments, the processing logic determines a time at which a latest impression was displayed on the display device and stores the count in a cache line.

[0037] It should also be noted that an expiration time of expiration of a cache line and a maximum number of impressions of an advertisement are parameters for expiration of the cache line. The cache line includes the maximum number of impressions.

[0038] In various embodiments, the second cache line includes a second space ID of a web page in which additional advertisement data is to be embedded, an ad position of the additional advertisement data, an ad tag categorizing the additional advertisement data, a maximum number of impressions of the second advertisement, a constraint on the second advertisement, and an expiration time of the second cache line from the ADC. In one embodiment, the second cache line includes a maximum number of impressions of the second advertisement, a constraint on the second advertisement, or the expiration time of the second cache line.

[0039] The additional advertisement data is associated with the second advertisement. In one embodiment, the additional advertisement data is rendered to display the second advertisement. In one embodiment, the second space ID is associated with the first space ID. For example, the first space ID and the second space ID are associated with web pages having the same category. As another example, the first space ID and the second space ID are associated with web pages having the same content type. As yet another example, the first space ID and the second space ID are associated with web pages having the same category and content type.

[0040] In some embodiments, an association between web pages, such as the first and second web pages, is created within an application, such as a digital newsstand application, a digital newspaper application, or a digital magazine application. In one embodiment, the first space ID and the second space ID identify web pages that are received upon execution of the application. The application is downloaded to and executed on the client device to display web pages, which belong to a category, such as, news or investment.

[0041] Each of the first and second cache lines is associated with a corresponding expiration parameter. In one embodiment, the first cache line has a first expiration time and the second cache line has a second expiration time. The second expiration time may be the same or different than the first expiration time.

[0042] In operation 106, a serving plan is sent, via a network, such as the Internet or the Intranet, to the client device to store in the client device. Method 100 then ends. The method also ends when in operation 102, an ad request is not received.

[0043] Figure 2A is a flowchart of an embodiment of a method 118 for generating an ad request and receiving a serving plan in response to the ad request. The method 118 is performed by the processing logic. In operation 120, it is determined whether web page data is received from a web server, which is described below. Web page data may be received in response to sending a uniform resource locator (URL) request for the web page data. A processor of the client device sends the URL request via a network. It should be noted that instead of a processor, a microcontroller, a computer, a central processing unit (CPU), a server, or any other integrated circuit that executes processing logic can be used. The web page data is received via a network. The web page data is identified with a space ID and includes an ad position. Advertisement data is embedded within an ad position.

[0044] Upon determining that the web page data is received, an ad request is generated in operation 122. The ad request is then sent in operation 124 to the ad server via a network. In operation 126, a serving plan is received from the ad server via a network. The serving plan is stored in operation 128 within the ADC. In operation 128, the serving plan is stored in the ADC. The method 118 then ends. In one embodiment, in response to determination of lack of reception of web page data from a web server, the method 118 ends.

[0045] Figure 2B is a flowchart of another embodiment of a method 132 for generating an ad request and receiving a serving plan in response to the ad request. Operations 120, 122, 124, 136, 140, 142, 144, 146, 148, and 150 of the method 132 is performed by the processing logic. Operations 120 and 122 are described above. Furthermore, in operation 136, an ad request is sent to the ADC.

[0046] Upon sending the ad request, in operation 140, the processing logic determines whether a serving plan is received from the ADC in response to sending the ad request. In one embodiment, upon receiving the serving plan, the processing logic determines whether a space ID, sent within the ad request, is present within a cache line of the received serving plan.

[0047] If the space ID is present within the cache line, an ad tag of the cache line is sent, by the processing logic, to the AMC of the client device to obtain an ad media package from the AMC. In one embodiment, the ad media package includes content of an advertisement, geometry of the advertisement, texture of the advertisement, and/or color of the advertisement. The ad media package is identified by the ad tag.

[0048] The processing logic determines, in operation 140, whether an ad media package is received in response to sending the ad tag. If the ad media package is received from the AMC, the processing logic determines that the ad media package is received. Otherwise, in response to lack of reception of the ad media package from the AMC, the processing logic determines that the ad media package is not received.

[0049] Upon determining that the ad media package is received, in operation 148, the processing logic provides the ad media package to a renderer, described below. The renderer renders the ad media package to display an advertisement on the client device. The advertisement is displayed in operation 150. The method 132 then ends.

[0050] On the other hand, upon determining that the ad media package is not received, in operation 142, it is determined whether a web page, identified by a space ID within the ad request, is to be rendered without the ad media package. If so, in operation 144, the processing logic sends a request to the renderer to display, on the client device, a web page lacking an advertisement. The advertisement is associated with the ad media package and the web page is identified by the space ID in the ad request that was sent in operation 136.

[0051] In operation 146, the processing logic queues the ad request to send the ad request to the ad server and sends, in operation 124, the ad request. The remaining operations 126 and 128, shown in Figure 2A, are executed in response to sending the ad request to the ad server. It should be noted that in another embodiment, the processing logic determines to avoid queuing to send the ad request and the method 132 ends. In such embodiments, the processing logic avoids performing operations 146, 124, 126, and 128.

[0052] In some embodiments, communication is lost or is intermittently lost between the processing logic and the ad server after the serving plan is received by the processing logic. The communication may be lost or intermittently lost as a result of a faulty communication medium between the processing logic and the ad server, the ad server being faulty, or a processor having the processing logic being faulty. A fault may be a malfunction or any other inoperable state. Regardless of whether the communication is lost after the serving plan is received by the processing logic, an ad media package corresponding to the serving plan is

obtained and rendered to display an advertisement on the display device.

[0053] In various embodiments, an ad media package is deleted from the AMC when a storage capacity of the AMC is reached. The processing logic determines an ad media package from multiple ad media packages to delete based on a variety of factors, such as a time of use of the ad media package, a number of cache lines identifying the ad media package, and/or a size of the ad media package. For example, an ad media package that is used latest is deleted instead of other ad media packages.

[0054] Figure 3 is a block diagram of an embodiment of a system 200 for facilitating caching of a serving plan 264. A user 202 performs an action, such as a selection of an icon or writing a script and selecting an 'Enter' button, to execute an application 254, which is stored in a memory device 252. In one embodiment, the memory device 252 is a read-only memory (ROM), a random access memory (RAM), or a combination of ROM and RAM. In some embodiments, the application 254 is a digital newsstand, a digital newspaper, a digital magazine, or a web browser application. Examples of the web browser application include Internet Explorer web browser, Mozilla Firefox web browser, or Netscape Navigator web browser. In one embodiment, the application 254 is downloaded to a client device 252 via a network 206, such as the Internet or Intranet, from a web server 204A.

[0055] The application is executed to download web page data 256A and 256B from a web server 204B via the network 206. In one embodiment, the web page data 256 includes web page data describing a newspaper or a magazine. In some embodiments, the web page data 256 includes web page data downloaded from web server 204B as a result of the user 202 selecting a hyperlink of a search result. The search result is generated when the user provides a search query via a search engine, such as a Google search engine or a Yahoo search engine.

[0056] The web page data 256A includes an ad position for embedding advertisement data and a space ID of a web page 214A. In one embodiment, the web page 214A is to be displayed on a display device 278 by rendering the web page data 256A. Similarly, the web page data 256B includes an ad position for embedding advertisement data and a space ID of a web page 214B. In one embodiment, the web page 214B is to be displayed on the display device 278 by rendering the web page data 256B. The display device 278, in some embodiments, is a liquid crystal display (LCD), a light emitting diode (LED) display, a plasma display, or a cathode ray tube (CRT) display.

[0057] A processing logic 276 of processor 255 determines whether a serving plan 264 associated with the web page data 256 is stored in ADC 260 based on the space IDs of the

web pages 214 and the ad positions. In one embodiment, ADC 260 has a storage capacity of the order of kilobytes. In some embodiments, ADC 260 has a storage capacity ranging from 1 kilobyte to 2 kilobyte.

[0058] In one embodiment, processor 255, ADC 260, and AMC 268 are parts of a microprocessor system 257. In some embodiments, AMC 268 has a storage capacity of the order of kilobytes or megabytes. In various embodiments, ADC 260 has a storage capacity ranging from 100 kilobytes to megabytes. In one embodiment, the processing logic 276 includes a set of instructions that are executed by processor 255.

[0059] The ADC 260 includes the first and second cache lines 266 and 268. The first cache line 266 includes the first space ID and an ad position of the first advertisement and the second cache line 268 includes the second space ID and an ad position of the second advertisement.

[0060] To determine whether the serving plan 264 is associated with the web page data 256, processing logic 276 determines whether the space IDs of the web page data 256 matches the first and second space IDs. If there is a match with the first and second space IDs, there is a cache hit and the processing logic 276 executes the first cache line 266 and if there is a match with the second space ID, there is a cache hit and processing logic 276 executes the second cache line 268.

[0061] On the other hand, in one embodiment, in response to determining that the serving plan 264 associated with the web page data 256 is not stored in ADC 260, a cache miss occurs. In such a case, in some embodiments, the processing logic 276 determines to send any ad media package stored in AMC 268 to Tenderer 208. The Tenderer 208 renders the ad media package to display an ad on the web page 214A or 214B. In other embodiments, in response to determining that the serving plan 264 associated with the web page data 256 is not stored in ADC 260, the processing logic 276 sends an ad request 258 to an ad server 262.

[0062] When the first cache line 266 is executed, the processing logic 276 determines whether an ad media package 272 identified by an ad tag of the first cache line is stored in AMC 268. The ad media package 272 includes media elements 275, such as content of an advertisement, geometry of the advertisement, texture of the advertisement, and/or color of the advertisement. In some embodiments, a media element has a format of a graphics interchange format (gif) file, a video file, or a javascript file.

[0063] Upon determining that the ad media package 272 is stored in the AMC 268, a cache hit occurs. On the other hand, upon determining that AMC 268 lacks the ad media package

272, a cache miss occurs and the processing logic 276 requests, via a content delivery network (CDN) 278, the ad media package 272 from a database 280 of an origin server 280. In one embodiment, the CDN 278 is the Internet or an Intranet. Upon reception of the ad media package 272 from origin server 282, the processing logic 276 stores the ad media package 272 in the AMC 268.

[0064] When the second cache line 268 is executed, the processing logic 276 determines whether an ad media package 274 identified by an ad tag of the second cache line is stored in the AMC 268. It should be noted that in some embodiments, the ad tags of the first and second cache lines 266 and 268 identify the same ad media package 272.

[0065] Upon determining that the ad media package 274 is stored in the AMC 268, a cache hit occurs. On the other hand, upon determining that AMC 268 lacks the ad media package 274, a cache miss occurs and the processing logic 276 requests, via the CDN 278, the ad media package 274 from the database 280. Upon reception of the ad media package 274 from origin server 282, the processing logic 276 stores the ad media package 274 in the AMC 268.

[0066] The processing logic 276 provides the ad media package 272 or 274, retrieved from AMC 268, to provide to Tenderer 208. It should be noted that Tenderer 208 is executed by a graphical processing unit (GPU) 259 that is a part of display device 270. In some embodiments, the GPU 259 may be located outside the display device 270. Renderer 208 renders media elements 275 of the ad media package 272 to display an advertisement 210 on web page 214A and renderer 208 renders media elements of the ad media package 274 to display an advertisement 212 on web page 214B.

[0067] It should be noted that when advertisements 210 and 212 are the same, the media elements of the first and second advertisements 210 and 212 are the same. Otherwise, when ads 210 and 212 are different, one or more of the media elements 275 is different than one or more of media elements of the ad 212. It should further be noted that in some embodiments, both first and second advertisements 210 and 212 may be displayed on the same web page 214A when a space ID of the web page 214A is mentioned in the first and second cache lines, which also include ad tags identifying the first and second advertisements 210 and 212. Also, in one embodiment, advertisement 210 or 212 has an image format, a video format, an animation format, or an audio format.

[0068] In case of a cache miss in which serving plan 264 is absent from ADC 260 or ad media package 268 is absent from AMC 268, in several embodiments, processing logic 276 performs one of three operations based on a configuration of the processing logic 276. In a

first configuration, the processing logic 276 determines to send web page data 256A to renderer 208 to display the web page 214A that lacks even a single advertisement. In the first configuration, the processing logic 276 avoids sending the ad request 258 to an ad server processor 263 of the ad server 262.

[0069] In a second configuration, the processing logic 276 determines to send web page data 256A to renderer 208 and the renderer 208 displays the web pages 214 on display device 270. In this configuration, the web pages 214 lack even a single advertisement. Moreover, in the second configuration, the processing logic 276 queues and sends the ad request 258 to ad server processor 263 after sending the web page data 256A to renderer 208. In the second configuration, the processing logic 276 sends the web page data 256A to renderer 208 without waiting to send the ad request 258 to ad server processor 263. The ad server processor 263 generates the serving plan 264 to send to the processing logic 276. The processing logic 276 receives the serving plan 264 and stores the serving plan 264 in the ADC 260.

[0070] In the second configuration, after display of the web page 214A, in case web page data 256A is received again from one of web servers 204 having a space ID and an ad position that match the first space ID and the ad position in the first cache line 266, the web page data 256A and ad package 272 identified by an ad tag of the first cache line 266 are provided by processing logic 276 to renderer 208. The renderer 208 renders the ad package 272 to display the ad 210 on web pages 214A. In the second configuration, if the ad media package 272 is not stored in AMC 268 when serving plan 264 is received from ad server 262, the ad media package 272 is requested, as described above, from origin server 282.

[0071] In a third configuration, the processing logic 276 waits to send web page data 256A to renderer 208 before sending the ad request 258 to ad server processor 263. The processing logic 276 first sends the ad request 258 to ad server processor 263. In response to receiving the serving plan 264 based on the ad request 258, the processing logic 276 retrieves the ad media package 272 from AMC 268 and sends the ad media package 272 and the web page data 256A to renderer 208. The renderer 208 renders the ad media package 272 and the web page data 256A to display the ad 210 on web page 214A. Again, in the third configuration, if the ad media package 272 is not stored in AMC 268 when serving plan 264 is received from ad server 262, the ad media package 272 is requested, as described above, from origin server 282.

[0072] A media vendor server 284 is managed by a media vendor entity. In one embodiment, an entity is a person or a company. The media vendor server 284 is used by the media vendor

entity to create the ad packages 272 and 274. In one embodiment, the media vendor server 284 is used by the media vendor entity to create an ad that displays a black Ford Taurus adjacent to text, "Best of the Best". In some embodiments, the media vendor server 284 creates the media elements 275 and the media elements of the ad package 274. The media vendor server 284 is used by the media vendor entity to generate an ad tag that identifies the ad media package 272 and another ad tag that identifies the ad media package 274.

[0073] The media vendor server 284 sends the ad tags identifying the ad media packages 272 and 274 to a campaign management server 286. The campaign management server 286 is used to associate an ad tag identifying an advertisement with a space ID identifying a web page. In one embodiment, an ad tag is linked with a space ID of a web page based on content of the web page, content of the advertisement, a bid for displaying the advertisement on the web page, and other factors. In some embodiments, a web page is associated with an advertisement to display the advertisement at an ad position on the web page. Various embodiments of a campaign management system are described in U.S. Patent Application Publication No. 2007/0027765, which is incorporated by referenced herein in its entirety. The associations between ad tags, ad positions, and space IDs are provided by campaign management server 286 to ad server 262.

[0074] In one embodiment, the application 254 includes associations between multiple web pages. For example, a digital newsstand application includes an association between a first web page of a digital newspaper and a second web page of the digital newspaper so that the second web page is displayed after the first page is displayed on display device 270. As another example, based on a browsing history of user 202, a fourth web page is displayed after a third web page is displayed on display device 270. In this example, the user 202 has a browsing history of frequently downloading the fourth web page from one of web servers 204 after downloading the third web page. In this embodiment, the associations are retrieved from application 254 by campaign management system 286 to provide to ad server 284.

[0075] Processor 263 of ad server 262 stores the associations between web pages and associations between space IDs, ad positions, and ad tags in an ad server database 288.

[0076] When the ad request 258 is received, ad server 262 generates a serving plan based on the associations between web pages, ad tags, and ad positions and based on the ad request 258. In one embodiment, if the ad request 258 includes the first space ID identifying web page data 256A and an ad position within the web page data 256A, the ad server processor 263 looks up the database 288 to determine an ad tag associated with the first space ID and

associated with the ad position. The ad server processor 263 generates a cache line, of the serving plan, having the ad tag and other parameters, such as, the first space ID, the ad position, the ad tag, a maximum number of impressions of an advertisement identified by the ad tag, a constraint on the advertisement, and an expiration time of the first cache line. The ad server 262 sends the serving plan 264 to the processing logic 276.

[0077] In some embodiments, when the ad request 258 is received, ad server 262 generates serving plan 264 based on the associations between web pages, based on the ad request 258, and based on the associations between web pages, ad tags, and ad positions. In one embodiment, if the ad request 258 includes the first space ID identifying web page data 256A and an ad position within the web page data 256A, the ad server processor 263 looks up the database 288 to determine an ad tag associated with the first space ID and the ad position. The ad server processor 263 generates the first cache line including the first space ID, the ad position, the ad tag, a maximum number of impressions of advertisement 210 identified by the ad tag, a constraint on the advertisement 210, and an expiration time of the first cache line. Moreover, the ad server processor 263 looks up database 288 to determine whether the first space ID is associated with the second space ID stored in database 288. If so, ad server processor 263 looks up the database 288 to determine an ad tag associated with the second space ID stored in the database 288 and with an ad position, of second advertisement 212, stored in the database 288.

[0078] The ad server 262 generates the second cache line including the second space ID, the ad position of second advertisement 212, the ad tag identifying ad media package 274, a maximum number of impressions of the second advertisement 212, a constraint on the second advertisement 212, and an expiration time of the second cache line. The ad server 262 sends the serving plan 264 including the first and second cache lines to processing logic 276.

[0079] It should be noted that although the various embodiments, described herein, are described using two cache lines, in some embodiments, any number of cache lines is used. It should also be noted that although some embodiments are described using a single processor or a single server, in various embodiments, multiple servers or processors are used instead. Moreover, although various embodiments are described using two web pages, in other embodiments, any number of web pages can be used.

[0080] Figure 4 is a flowchart of an embodiment of a method 300 for determining whether to request an ad media package and for requesting the ad media package. Operations 302, 304, 306, 308, 310, and 312 of the method 300 are executed by processing logic 276. In operation

302, a cache line is read from ADC 260. In another operation 304, it is determined whether an ad media package associated with the cache line is stored in AMC 268. In one embodiment, an ad media package is associated with a cache line when the cache line includes an ad tag, which identifies the ad media package. If so, operations 148 and 150, described in Figure 2B, are performed.

[0081] On the other hand, if the ad media package is unassociated with the cache line, a request for the ad media package is generated in operation 306. In operation 308, the request for the ad media package is sent via a network, such as CDN 278, to origin server 282. In response to receiving the request, the ad media package is received from the origin server 282 in operation 310. The ad media package is stored in the AMC 268 to be displayed on display device 270. Operations 148 and 150 of Figure 2B are then performed.

[0082] Figure 5 is a flowchart of an embodiment of a method 350 for determining whether to request a subsequent version of an ad media package and generating the request. Operations 352, 354, 356, 358, and 360 of the method 350 are performed by the processing logic 276. In operation 352, the processing logic 276 receives an ad media package having a current version from the origin server 282 via CDN 278. The current version is a version developed before the subsequent version. In one embodiment, the current version is version 1.0 of an ad media package and the subsequent version is version 1.1 of the ad media package. In another embodiment, the current version is version 1.0 and the subsequent version is version x.y, where x is an integer greater than zero and y is an integer greater than 0. In the subsequent version, there is a change in one or more media elements compared to the current version. In some embodiments, in the subsequent version, there is a change in one or more but not all media elements compared to the subsequent version. For example, there is a lack of change in geometry between the current and subsequent versions and a change in color between the current and subsequent versions.

[0083] In operation 354, the processing logic 276 determines whether an indication that the subsequent version is available is received. In one embodiment, the media vendor server 284 generates the subsequent version and sends the subsequent version to the origin server 282 and to campaign management server 286. The campaign management server 286 replaces an association between a space ID, an ad tag and the current version of an advertisement with an association between the space ID, the ad tag and the subsequent version of the advertisement, also replaces the ad tag identifying the current version with an ad tag identifying the subsequent version, and sends the replaced associations with the space ID and the ad tag

identifying the subsequent version to the ad server 262. In response to receiving the subsequent version, the origin server 282 sends an indication to processing logic 276 that the subsequent version is received. In some embodiments, an indication that the subsequent version is generated is sent by ad server 262 to processing logic 276.

[0084] In response to receiving the indication, processing logic 276 determines, in operation 356, whether one or more cache lines in ADC 280 refers to the current version. A cache line refers to the current version when the cache line includes an ad tag identifying the current version. If so, in operation 360, the processing logic 276 avoids generating and sending a request to the origin server 282 for obtaining the subsequent version from the origin server 282. In such a case, when the current version is not replaced with the subsequent version, processing logic 276 sends, in operation 148 of Figure 2B, the current version to Tenderer 208. In operation 150 of Figure 2B, the Tenderer 208 renders the current version to display advertisement 210, having the current version, on web page 214A on display device 270. The web page 214A is associated with the space ID received with the replaced associations.

[0085] On the other hand, upon determining that one or more cache lines in the ADC 280 lack reference to the current version, in operation 358, the processing logic 276 generates and sends a request to the origin server 282 via CDN 278 for the subsequent version. In response to sending the request, the subsequent version is received from the origin server 282 via CDN 278. The processing logic 276 replaces the current version in AMC 268. When the current version is replaced with the subsequent version, processing logic 276 sends, in operation 148 of Figure 2B, the subsequent version to Tenderer 208. In operation 150 of Figure 2B, the Tenderer 208 renders the subsequent version to display an advertisement, having the subsequent version, on display device 270.

[0086] Figure 6 is a flowchart of an embodiment of a method 362 for executing an ad refresh. Operations 372, 374, 376 and 378 of the method 362 are performed by the processing logic 276. In operation 372, it is determined whether an ad refresh is to be executed. An ad refresh is to be executed when an indication is received from the user 202 via an input device that the ad refresh is to be executed. In one embodiment, the indication is received when the user 202 selects an "ad refresh" button on display device 270. In another embodiment, the indication is received when the user 202 enters a script on display device 270 that an ad refresh is to be performed. In another embodiment, an ad refresh is to be executed periodically, such as every z time units, where z is an integer greater than zero. Examples of the time units include seconds, minutes, and hours. An example of periodic execution of an ad refresh is an

execution every 15 or 20 minutes.

[0087] If an ad refresh is not to be executed, the method 352 ends. On the other hand, if an ad refresh is to be executed, an additional ad request is generated in operation 374. The additional ad request includes an ad position for embedding advertisement data into web page data 256A and further includes a space ID identifying the web page data 256A. In operation 376, the additional ad request is sent to ad server 262. Further, in operation 378, an additional serving plan is received in response to the additional ad request.

[0088] The additional serving plan is the same or different than serving plan 264, which is received in response to the ad request sent in operation 124 of Figure 2A. In one embodiment, the additional serving plan includes an ad tag that identifies the new version compared to the serving plan 264, which is associated with the current version. In another embodiment, the additional serving plan includes a different parameter than that included in the serving plan 264. For example, the additional serving plan includes a different number of maximum impressions than that included in the serving plan 264.

[0089] Upon receiving the additional serving plan, the serving plan 264 is updated, such as replaced, by the processing logic 276 with the additional serving plan in ADC 260. If the ad refresh is executed periodically, the serving plan 264 is also updated periodically. Moreover, operations 302, 304, 306, 308, 310, 312 of Figure 4 and operations 148 and 150 of Figure 2B are repeated with respect to the additional serving plan. In one embodiment, in operation 302, a cache line of the serving additional plan is read. Moreover, in operation 304, it is determined whether an ad media package identified by an ad tag of the cache line of the additional serving plan is stored in AMC 268. If so, the ad media package is sent, in operation 148 to the Tenderer 208. Renderer 208 renders the ad media package to display an advertisement corresponding to the additional serving plan on web page 214A.

[0090] On the other hand, if the ad media package associated with the cache line of the additional serving plan is not stored in the AMC 268, a cache miss occurs and a request for the ad media package is generated in operation 306. The request is sent via the CDN 278 to the origin server 282. The ad media package is received in operation 310 in response to the request and is stored, in operation 312. The ad media package, stored in operation 312, is sent to renderer 208 in operation 148 and the renderer 208 renders, in operation 150, the ad media package to display an advertisement.

[0091] In other embodiments, when the ad refresh is to be executed, instead of sending the additional ad request to the ad server 262, the processing logic 276 sends the additional ad

request to the ADC 260 to determine whether the ADC 260 includes a serving plan that includes a cache line including a space ID that is provided in the ad request. If so, the processing logic 276 executes the serving plan to retrieve an ad media package identified by an ad tag of the cache line from AMC 260. The ad media package is provided to renderer 208 to render the ad media package on display device 270. On the other hand, in case of a cache miss in which the ad media package is not found within AMC 260 or the serving plan is not found within ADC 260, the processing logic sends the additional ad request to ad server 262.

[0092] Figure 7 illustrates suitable components in an exemplary embodiment of a computer 400. The exemplary embodiment is only one example of suitable components and is not intended to suggest any limitation as to the scope of use or functionality of various embodiments of the present invention. Neither should the configuration of components be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary embodiment of computer 400.

[0093] Various embodiments of the present invention may be described in the general context of computer-executable instructions, such as processing logic 276, being executed by a CPU 402. Generally, processing logic includes routines, programs, objects, components, and/or data structures, which perform particular tasks. Various embodiments of the present invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, processing logic 276 may be located in local and/or remote computer storage media including memory storage devices.

[0094] Components of the computer 400 may include, but are not limited to, CPU 402, a system memory 404, and a system bus 420 that couples various system components including the system memory 404 to the processing unit 402. The system bus 420 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0095] The computer 400 may include a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by the computer 400 and includes both volatile and nonvolatile media. For example, computer-readable media may

include volatile and nonvolatile computer storage media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer-readable media includes, but is not limited to, cache, RAM, ROM, flash memory, compact disc ROM (CD-ROM), digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 400.

[0096] The system memory 404 includes computer-readable media in the form of volatile and/or nonvolatile memory such as ROM 406 and RAM 410. An input/output system 408 (BIOS), containing routines that help transfer information between elements within computer 400, such as during start-up, is typically stored in ROM 406. Additionally, RAM 410 may contain operating system 412, application programs 414, other executable code 416 and program data 418. RAM 410 typically contains data and/or processing logic 276 that are immediately accessible to and/or presently being operated on by CPU 402.

[0097] The computer 400 may also include other removable/non-removable, volatile/nonvolatile computer-readable media. By way of example only, Figure 7 illustrates a hard disk drive 422 that reads from or writes to non-removable, nonvolatile magnetic media, and storage device 434 that may be an optical disk drive or a magnetic disk drive that reads from or writes to a removable, nonvolatile storage medium 444 such as an optical disk or magnetic disk. In some embodiments, a hard disk and/or storage medium 434 includes one or more databases. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary computer 400 include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 422 is coupled with ADC 260 and AMC 268. It should be noted that although two separate caches are shown in Figure 7, in some embodiments, ADC 260 and AMC 268 may be part of the same cache, which is partitioned into ADC 260 and AMC 268. In one embodiment, each of ADC 260 and AMC 268 or both AMC 260 and 268 are parts of a database, such as a structured query language (SQL) database, a Datalog database, or a .QL database. It should further be noted that, in one embodiment, CPU 402 is able to retrieve instructions and/or data from a cache faster than from hard drive 424 or from storage medium 444.

[0098] The hard disk drive 422 and the storage device 434 may be typically connected to the system bus 420 through an interface such as storage interface 424. In some embodiments, the

drives and their associated computer storage media provide storage of computer-readable instructions, executable code, data structures, processing logic 276 and other data for the computer 400. In Figure 7, for example, hard disk drive 422 is illustrated as storing operating system 412, application programs 414, other executable code 416 and program data 418. The user 202 may enter commands and information into the computer 400 through an input device 440 such as a keyboard and pointing device, commonly referred to as mouse, trackball or touch pad tablet, electronic digitizer, or a microphone. Other input devices may include a joystick, game pad, satellite dish, scanner, and so forth. These and other input devices are often connected to CPU 402 through an input interface 430 that is coupled to the system bus 420, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A display device 438 or other type of video device may also be connected to the system bus 420 via an interface, such as a video interface 428. In some embodiments, the display device 438 is an input device. In addition, an output device 442, such as speakers or a printer, may be connected to the system bus 420 through an output interface 432 or the like.

[0099] The computer 400 may operate in a networked environment using a network 436 to one or more remote computers, such as a remote computer 446. The remote computer 446 may be a personal computer (PC), a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 400. A network, such as the network 436, may include a local area network (LAN), a wide area network (WAN), or other type of network. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet. In a networked environment, executable code and application programs may be stored in the remote computer 446.

[00100] By way of example, and not limitation, Figure 7 illustrates remote executable code 448 as residing on remote computer 446. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used. Those skilled in the art will also appreciate that many of the components of the computer 400 may be implemented within a system-on-a-chip architecture including memory, external interfaces and operating system. System-on-a-chip implementations are common for special purpose client devices, such as hand-held devices. Examples of hand-held devices include mobile phones, tablets, digital music players, personal digital assistants, and the like.

[00101] Referring to Figure 8, a diagram illustrating the components of an exemplary server 500 is shown. The server 500 includes a CPU 502, a ROM 504, a RAM 506, a hard disk drive or storage memory 508, and input/output (I/O) interface 510 for coupling input/output devices, such as, a keyboard, monitor, printer, electronic pointing device (e.g., mouse, trackball, stylus, etc.), or the like. In one embodiment, the ROM 504 and/or RAM 506 includes one or more databases. It should be noted that in some embodiments, the server 500 can have more than one CPU, ROM, RAM, HD, I/O, or other hardware components.

[00102] It should be noted that although the flowcharts are described with a sequence of operations, in various embodiments, operations in a flowchart are performed in a different sequence than that show or are performed in parallel.

[00103] Although various embodiments of the present invention have been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications can be practiced within the scope of the appended claims. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.

CLAIMS

What is claimed is:

1. A method for facilitating caching of advertisements, comprising:
 - receiving an ad request;
 - generating a serving plan in response to said receiving the ad request, wherein said generating is performed by an ad server, the ad server including a processor, the serving plan including a plurality of cache lines identifying a plurality of corresponding advertisements, each of the cache lines associated with a corresponding expiration, each cache line readable by logic to display one of the advertisements corresponding to the cache line, the plurality of cache lines including a first cache line, the first cache line including ad information, the ad information associated with a first one of the advertisements, the ad information including one or more parameters for expiration of the first cache line of a client device, the client device configured to display the plurality of advertisements; and
 - sending the serving plan to the client device to store in the client device.
2. The method of claim 1, wherein the ad request includes a request for identification of advertisement data, a space identification (ID) categorizing web page data, and an ad position for embedding the advertisement data within the web page data, the web page data associated with a web page to be displayed on the client device, the first advertisement to be displayed on the web page, the advertisement data associated with the first advertisement.
3. The method of claim 1, wherein the plurality of cache lines includes a second cache line, the first cache line including a first space identification (ID) identifying a web page in which advertisement data is to be embedded, an ad position indicating a position of embedding the advertisement data within the web page, an ad tag categorizing the advertisement data, and a constraint on the first advertisement, the advertisement data associated with the first advertisement, the constraint including a frequency of display of the first advertisement, the parameters including a maximum number of impressions of the first advertisement and an expiration time of deletion of the first cache line from the

client device.

4. The method of claim 3, wherein the plurality of advertisements includes a second advertisement, the second cache line including a second space ID of a web page in which additional advertisement data is to be embedded, an ad position of the additional advertisement data, an ad tag categorizing the additional advertisement data, a maximum number of impressions of the second advertisement, and a constraint on the second advertisement, the constraint on the second advertisement including a frequency of display of the second advertisement.
5. The method of claim 1, wherein said sending the serving plan comprises sending the plurality of cache lines via Internet, wherein the ad information identifies the advertisement data and provides a frequency of display of the first advertisement on a web page.
6. A method for facilitating caching of advertisements, comprising:
 - determining whether web page data is received;
 - generating an ad request in response to determining that the web page data is received;
 - sending the ad request;
 - receiving a serving plan in response to said sending the ad request, the serving plan including a plurality of cache lines identifying a plurality of corresponding advertisements, each of the cache lines associated with a corresponding expiration, each cache line readable by logic to display one of the advertisements corresponding to the cache line, the plurality of cache lines including a first cache line, the first cache line including ad information, the ad information associated with a first one of the advertisements, the ad information including one or more parameters for expiration of the first cache line from a client device, the client device configured to display the plurality of advertisements;
 - storing the serving plan within the client device, wherein said storing is performed by processing logic of the client device; and
 - periodically updating the serving plan based on an ad refresh.
7. The method of claim 6, wherein the web page data is associated with a space

identification (ID) and includes an ad position, said method further comprising determining whether the ad information associated with the space ID and with the ad position is stored in the client device in response to determining that the web page data is received, wherein said generating an ad request comprises generating the ad request in response to determining that the client device lacks the ad information.

8. The method of claim 6, further comprising:
 - sending a uniform resource locator (URL) request for the web page data;
 - receiving the web page data in response to the URL request; and
 - determining that the web page data is received upon receiving the web page data.
9. The method of claim 6, wherein said generating an ad request comprises generating a request for an identification of the first advertisement to be displayed.
10. The method of claim 6, wherein said receiving the serving plan comprises receiving the plurality of cache lines, the cache lines including ad information identifying advertisement data of the plurality of advertisements to be displayed, and a plurality of frequencies of display of the advertisements, wherein the parameters include a maximum number of impressions of the first advertisement associated with the first cache line and an expiration time of deletion of the first cache line from the client device.
11. The method of claim 6, further comprising:
 - reading the first cache line;
 - determining whether an ad media package associated with the first cache line is stored within an ad media cache (AMC) of the client device, said determining whether ad media package is stored is performed in response to said reading the first cache line, the ad media package including media elements for rendering the first advertisement at an ad position, the ad position specified in the first cache line;
 - generating a request for the ad media package in response to determining that the AMC lacks the ad media package;
 - sending the request via a network;
 - receiving the ad media package via the network in response to the request for the ad media package; and

storing the ad media package in the AMC.

12. The method of claim 11, wherein the serving plan is sent to be stored in a cache of the client device.
13. The method of claim 6, further comprising:
 - receiving an ad media package having a current version;
 - receiving an indication that a subsequent version of the ad media package is available;
 - determining whether one or more of the plurality of cache lines in the client device refers to the current version in response to said receiving an indication;
 - generating and sending a request for the subsequent version in response to determining that the one or more cache lines lacks reference to the current version; and
 - avoiding said generating and sending a request in response to determining that the one or more cache lines refers to the current version.
14. The method of claim 6, further comprising:
 - receiving an ad media package;
 - loosing communication with an ad server; and
 - rendering the ad media package to display the first advertisement after the communication is lost.
15. The method of claim 6, wherein the first cache line includes a first space identification (ID) of the web page data in which the advertisement data is to be embedded, an ad position indicating a position of embedding the advertisement data within the web page data, an ad tag categorizing the advertisement data, a maximum number of impressions of the first advertisement, and a constraint on the first advertisement, the advertisement data associated with the first advertisement, the constraint including a frequency of display of the first advertisement.
16. The method of claim 6, further comprising:
 - determining to execute the ad refresh;
 - generating an additional ad request in response to the determination to execute the ad refresh;

sending the additional ad request; and
receiving an additional serving plan in response to said sending the additional ad request.

17. The method of claim 16, further comprising:

determining whether an ad media package associated with the updated serving plan is stored in an ad media cache (AMC);

determining that a cache miss occurred in response to determining that the ad media package is not stored in the AMC; and

generating a request to obtain the ad media package via a network; and

receiving the ad media package via the network in response to sending the request.

18. A system for facilitating caching of advertisements, comprising:

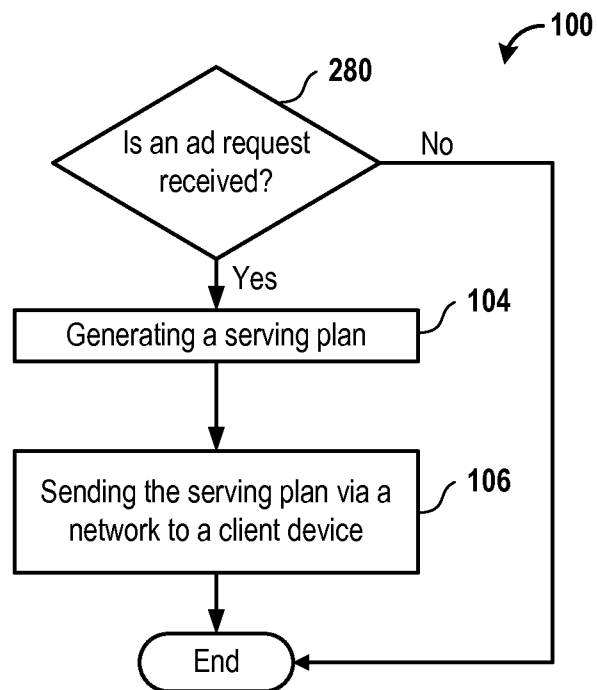
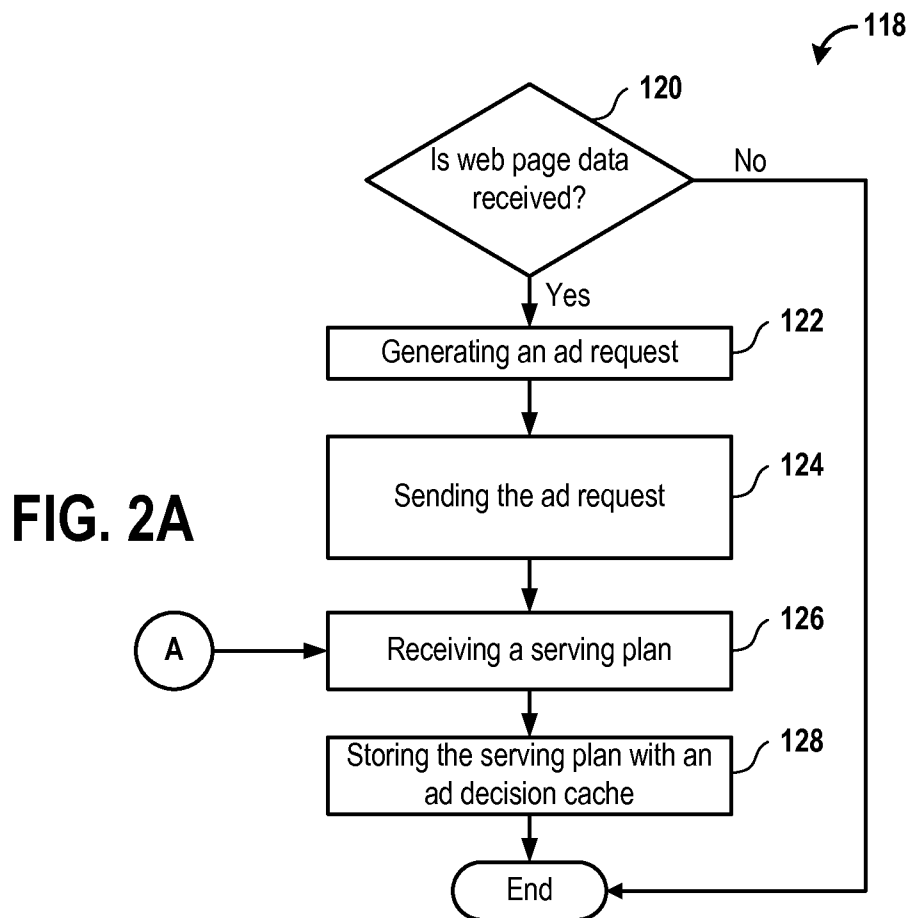
a client device for executing an application to allow access to web page data, the client device configured to generate an ad request upon downloading the web page data, the client device including an ad decision cache (ADC); and

an ad server for receiving the ad request, the ad server for generating a serving plan in response to receiving the ad request, the serving plan including a plurality of cache lines identifying a plurality of corresponding advertisements, each of the cache lines associated with a corresponding expiration, each cache line readable by logic to display one of the advertisements corresponding to the cache line, the plurality of cache lines including a first cache line, the first cache line including ad information associated with a first one of the advertisements, the ad information including one or more parameters for expiration of the first cache line from the ADC.

19. The system of claim 18, wherein the ad request includes a request for the advertisement data, a space identification (ID) categorizing web page data, and an ad position of embedding the advertisement data within the web page data, the web page data associated with a web page to be displayed on the client device, the advertisement to be displayed on the web page, the advertisement data associated with the first advertisement.

20. The system of claim 18, wherein the client device comprises an ad media cache and a display device, the client device configured to request an ad media package in response to

receiving the serving plan, the ad media package including media elements, the display device configured to apply the media elements to render the first advertisement.

**FIG. 1**

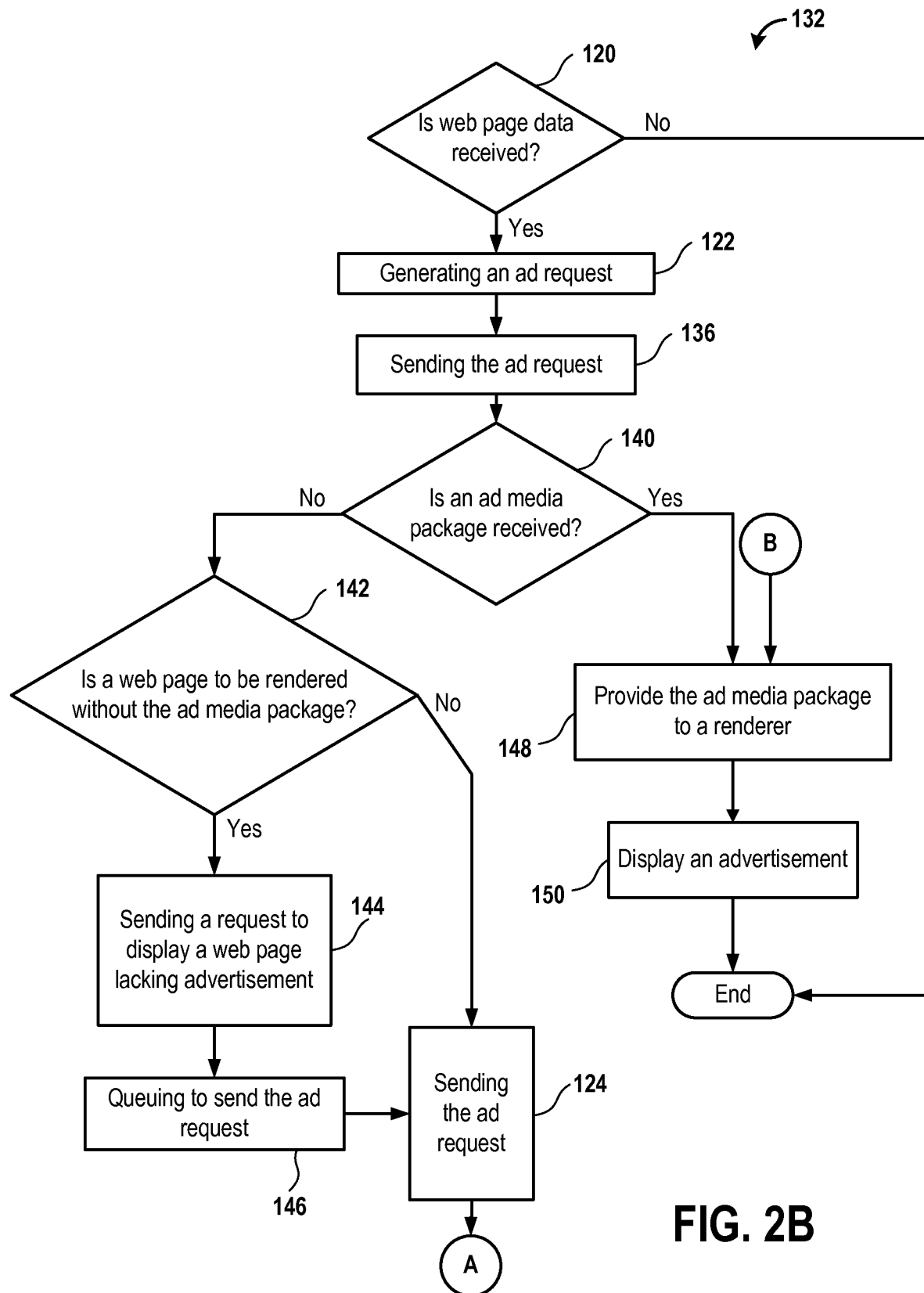


FIG. 2B

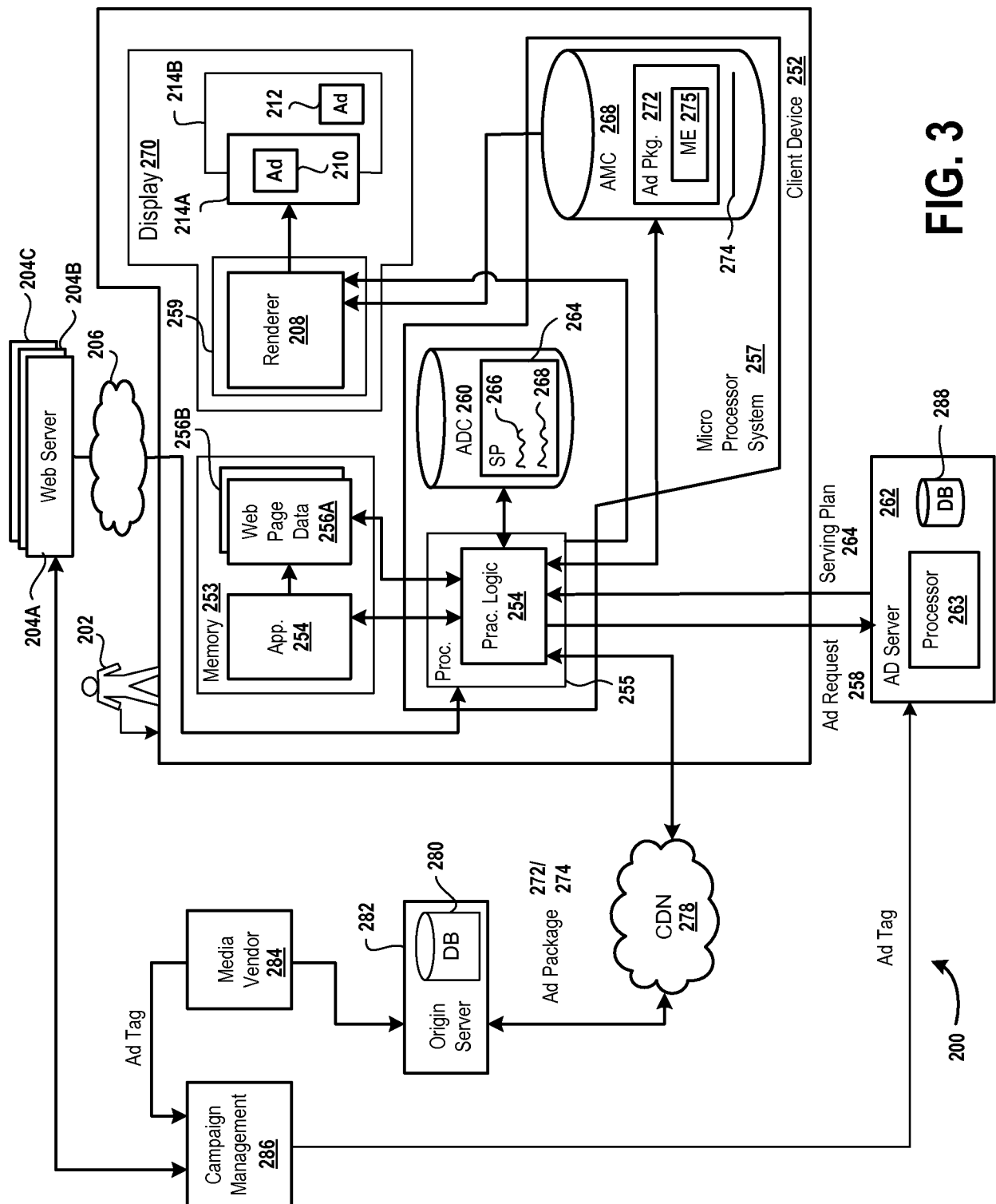
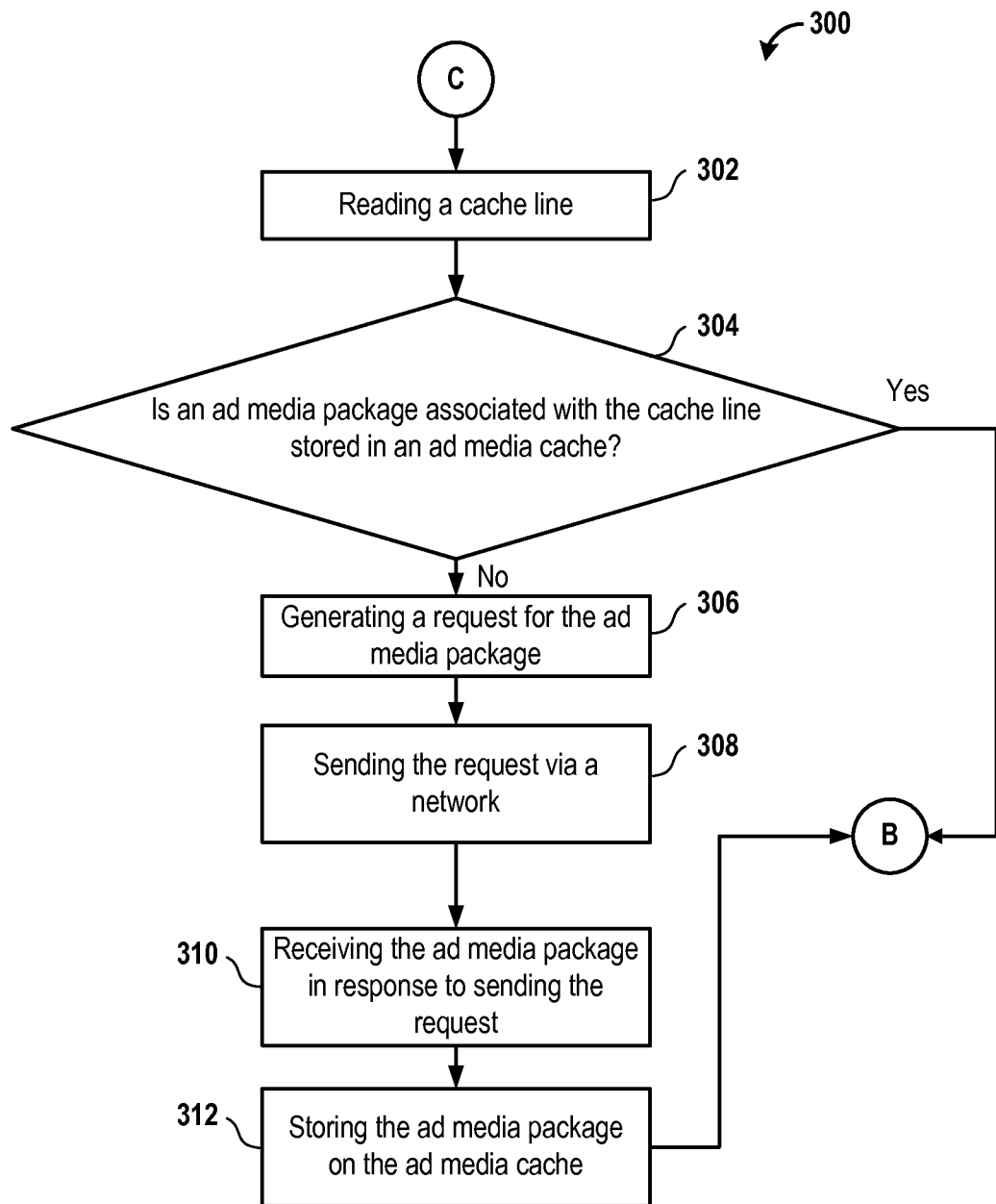


FIG. 3

**FIG. 4**

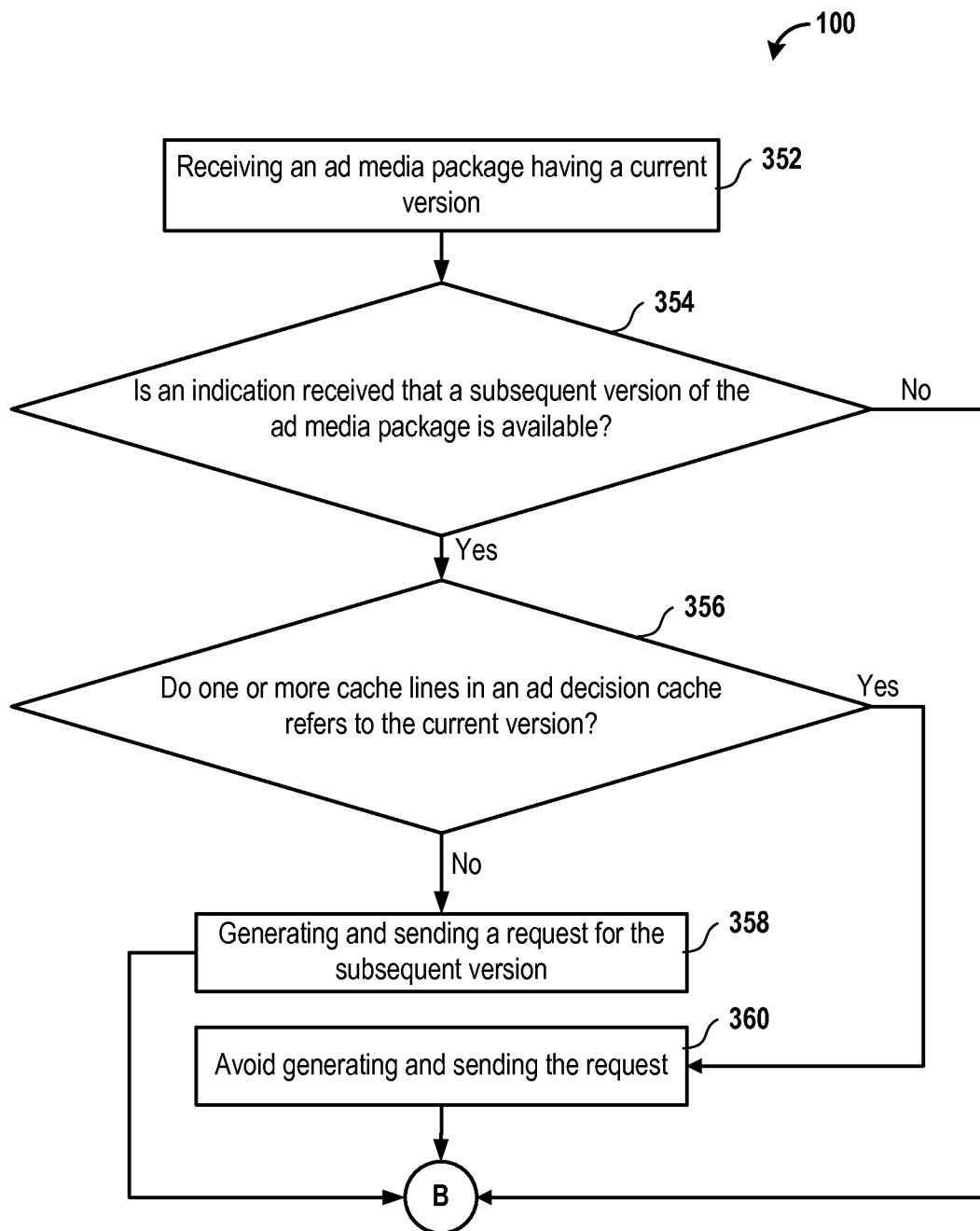
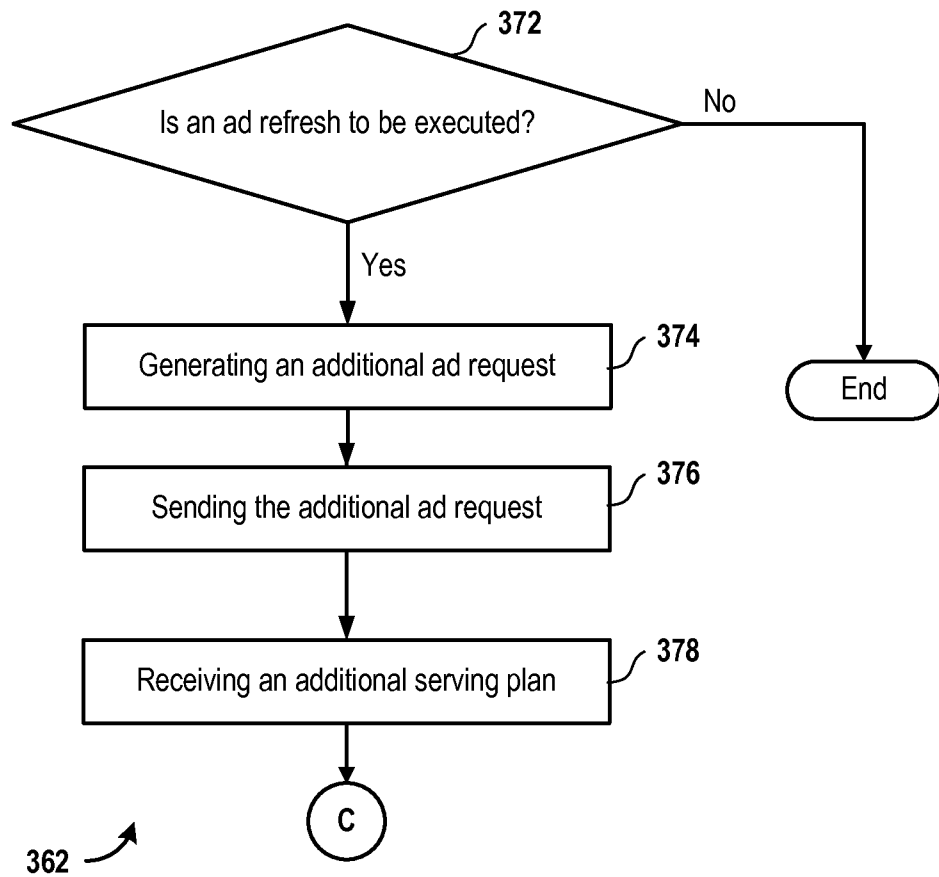


FIG. 5

**FIG. 6**

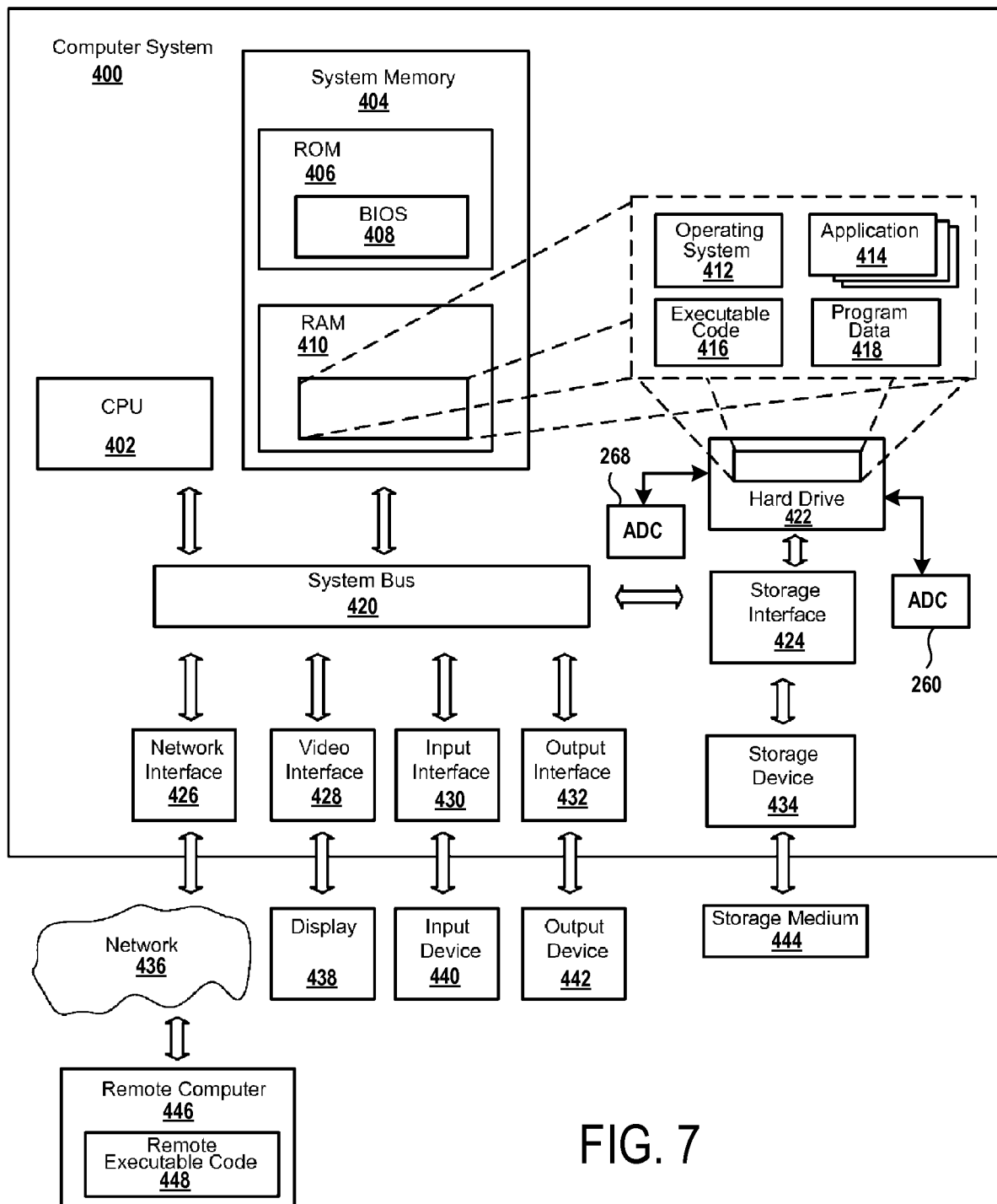


FIG. 7

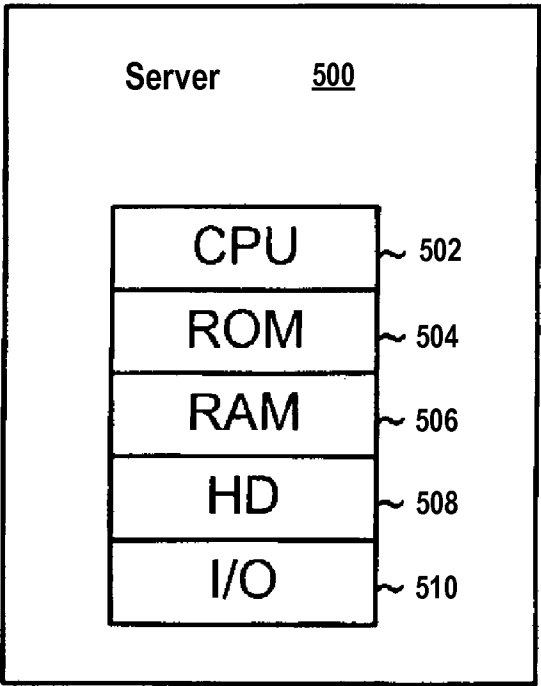


FIG. 8