(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 October 2007 (25.10.2007)

PCT

(10) International Publication Number
WO 2007/120411 A1

(51) International Patent Classification:
G06F 17/00 (2006.01)

(21) International Application Number:
PCT/US2007/006024

(22) International Filing Date: 8 March 2007 (08.03.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/404,412 14 April 2006 (14.04.2006) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **VOLODARSKY, Michael D.**; One Microsoft Way, Redmond, WA 98052-6399 (US). **OLSON, Erik B.**; One Microsoft Way, Redmond, WA 98052-6399 (US). **RUIA, Anil K.**; One Microsoft Way, Redmond, WA 98052-6399 (US).

(74) Common Representative: **MICROSOFT CORPORATION**; Attention : Sharon Rydberg, (sharonr-21- 2029), LCA, International Patent Department, One Microsof t Way, 21/2029, Redmond, Washington 98052 - 6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, 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, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, 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, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, 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))*

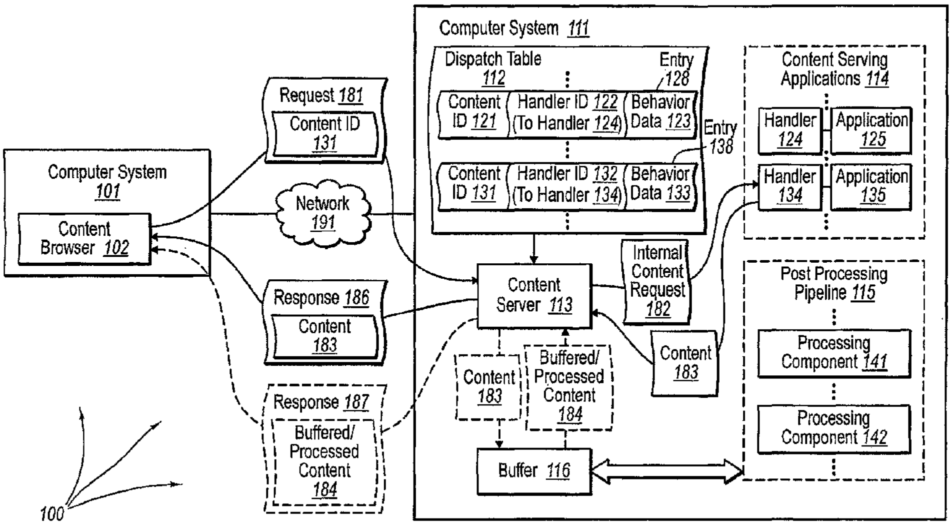
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report*

[Continued on next page]

(54) Title: MANAGING NETWORK RESPONSE BUFFERING BEHAVIOR



(57) **Abstract:** The present invention extends to methods, systems, and computer program products for managing network response buffering behavior. A computer system receives a request for content from a client. The computer system has a default response buffering behavior used when transferring content. The computer system maps the request to a handler configured to serve the requested content. The computer system accesses buffering behavior data for the handler. The computer system determines that the requested content is to be transferred in accordance with altered response buffering behavior based at least on the buffering behavior data. The altered response buffering behavior corresponds to the requested content as an exception to the default response buffering. The computer system accesses a portion of the requested content from the handler. The computer system transfers the portion of requested content to the client in accordance with the altered response buffer behavior.

WO 2007/120411 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

MANAGING NETWORK RESPONSE BUFFERING BEHAVIOR

BACKGROUND

[0001] 1. **Background and Relevant Art**

[0002] Computer systems and related technology affect many aspects of society. Indeed, the computer system's ability to process information has transformed the way we live and work. Computer systems now commonly perform a host of tasks (e.g., word processing, scheduling, and database management) that prior to the advent of the computer system were performed manually. More recently, computer systems have been coupled to one another and to other electronic devices to form both wired and wireless computer networks over which the computer systems and other electronic devices can transfer electronic data. As a result, many tasks performed at a computer system (e.g., voice communication, accessing electronic mail, controlling home electronics, Web browsing, and printing documents) include the communication (e.g., the exchange of electronic content) between a number of computer systems and/or other electronic devices via wired and/or wireless computer networks.

[0003] In many computing environments content is exchanged in a request/response format. For example, a Web browser at a requesting computer system sends a request for content to a Web server at a server computer system. The server computer system receives the request and the Web server processes the request (e.g., passing the request through a pipeline of cooperative serve side components) to identify requested content. The Web server then formulates a response to include identified requested content. The server computer system sends the response back to the requesting computer system. The requesting computer system receives the response and the Web browser presents the requested content.

[0004] Web servers can utilize various different techniques when formulating a response to include requested content and sending a response that includes requested content. Streaming is one technique for formulating and sending a response that includes requested content. Streaming includes a Web server sending portions of requested content to a Web browser as the portions of requested content become available (e.g., from application components). For example, a Web server may stream audio/video file to a Web browser at a specified rate until transfer of the audio/video file is complete.

[0005] Response buffering is another technique for formulating a response that includes requested content. Response buffering includes a Web server storing portions of requested content in memory to collect a complete response in memory before sending any requested content to a Web browser. After collection of requested content is complete, a complete response is sent to the Web browser. For example, a Web server may store various different HTML portions in memory until all the HTML portions for a Web page are collected. After all of the HTML portions are collected, the Web server can send a completed Web page to a Web browser.

[0006] Response buffering can be beneficial for a variety of reasons. For example, response buffering can improve server performance in a variety of ways. Response buffering makes more efficient use of network bandwidth and reduces the overhead associated with sending frequent responses that include small amounts of requested content (e.g., relative to an optimal packet size). This can be of particular benefit for server applications that build dynamic content because these applications frequently generate a high frequency of smaller portions of content.

[0007] Additionally, response buffering enables post-processing of a response prior to sending the response. For example, a Web server can compress, encrypt, filter, cache, etc., a response prior to sending the response to a Web browser.

[0008] However, response buffering may not be appropriate for some applications and content types. For example, an application that generates large amounts of content may not be able to buffer the content due to memory overhead and/or constraints. Further, some applications may require streaming behavior (e.g., video, audio, etc.) and thus are not compatible with response buffering.

[0009] Web server platforms often support multiple content types and multiple application technologies that each has different buffering requirements. However, there are limited mechanisms for controlling response buffering per content type and/or per application. Additionally, buffering requirements can change over time, for example, depending on run-time characteristics (e.g., size) of a response.

[0010] Unfortunately, a Web server typically has no way to adapt to changed run-time characteristics. Thus, a Web server may select less appropriate response buffer behavior based on prior configuration and/or typical characteristics of a content type or application, even when more appropriate response buffer behavior is possible for a particular response. For example, a Web server may be configured to buffer portions of requested content even though an application handle for the requested content provides its own buffer (resulting in at least partial double buffer). On the other hand, a Web server may be configured to not buffer requested content based on a content type, even when the requested content includes a number of smaller portions of content.

[0011] Further, in some environments, response buffering can cause transfer requirements for content to be violated. For example, buffering content that has a time to first byte requirement or a specified transfer rate requirement can cause the transfer of content to exceed the time to first byte requirement or cause content to be transferred at a rate less than the specified transfer rate. Thus, to avoid violating transfer requirements, a Web server may default to never buffering content of any type (or may not include response buffering functionality at all). Accordingly, even those content types that would benefit from response buffering are not allowed to or cannot utilize response buffering.

BRIEF SUMMARY

[0012] The present invention extends to methods, systems, and computer program products for managing network response buffering behavior. A computer system receives a request for content from a client. The computer system has a default response buffering behavior used when transferring content. The computer system maps the request to a handler configured to serve the requested content.

[0013] The computer system accesses buffering behavior data for the handler. The computer system determines that the requested content is to be transferred in accordance with altered response buffering behavior based at least on the buffering behavior data. The altered response buffering behavior corresponds to the requested content as an exception to the default response buffering. The computer system accesses a portion of the requested content from the handler. The computer system transfers the portion of requested content to the client in accordance with the altered response buffer behavior.

[0014] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0015] Additional features and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0017] Figure 1 illustrates an example computer architecture that facilitates managing network response buffering behavior.

[0018] Figure 2 illustrates a flow chart of an example method for managing network response buffering behavior.

[0019] Figure 3 illustrates another example computer architecture that facilitates managing network response buffering behavior.

DETAILED DESCRIPTION

[0020] The present invention extends to methods, systems, and computer program products for managing network response buffering behavior. A computer system receives a request for content from a client. The computer system has a default response buffering behavior used when transferring content. The computer system maps the request to a handler configured to serve the requested content.

[0021] The computer system accesses buffering behavior data for the handler. The computer system determines that the requested content is to be transferred in accordance with altered response buffering behavior based at least on the buffering behavior data. The altered response buffering behavior corresponds to the requested content as an exception to the default response buffering. The computer system accesses a portion of the requested content from the handler. The computer system transfers the portion of requested content to the client in accordance with the altered response buffer behavior.

[0022] Embodiments of the present invention may comprise a special purpose or general-purpose computer including computer hardware, as discussed in greater detail below. Embodiments within the scope of the present invention also include computer-readable media for carrying or having computer-executable instructions or data structures stored thereon. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer. By way of example, and not limitation, computer-readable media can comprise computer-readable storage media, such as, RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

[0023] In this description and in the following claims, a "network" is defined as one or more data links that enable the transport of electronic data between computer systems and/or modules. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a computer-readable medium. Thus, by way of example, and not limitation, computer-readable media can comprise a network or data links which can be used to carry or store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

[0024] Computer-executable instructions comprise, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

[0025] Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, pagers, and the like. The invention may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

[0026] Figure 1 illustrates an example computer architecture 100 that facilitates managing network response buffering behavior. Computer architecture 100 includes computer system 101 and computer system 111. Each of computer system 101 and computer system 111are connected to network 191. Network 191 can be any type of network, such as, for example, a Local Area Network ("LAN"), a Wide Area Network ("WAN"), or even the Internet. Thus, the various components at computer system 101 and computer system 111 can receive data from and send data to each other, as well as other components connected to network 191. Accordingly, the components can create message related data and exchange message related data (e.g., Internet Protocol ("IP") datagrams and other higher layer protocols that utilize IP datagrams, such as, Transmission Control Protocol ("TCP"), Hypertext Transfer Protocol ("HTTP"), etc.) over network 191.

[0027] Generally, computer system 101 is configured to request, receive, and present content. For example, content browser 102 (e.g., a Web browser) can send a request for content including a content ID (e.g., a Uniform Resource Locator ("URL")) and a return electronic address onto network 191. Components of network 191 (e.g., routers) can route the request to a server computer system (e.g., including a Web server) identified in the content ID. The computer system 101 can receive requested content in a response from the server computer system (e.g., sent to the return electronic address). Computer system 101 can process the requested content and content browser 102 can present the requested content.

[0028] Generally, computer system 111 is configured to receive requests for content, process the requests to identify requested content, and return identified content in a response to a received request. For example, content server 113 (e.g., a Web server) can receive a request for content including a content ID (e.g., a Uniform Resource Locator ("URL")) and a return electronic address from network 191. Content server 113 can refer to dispatch table 112 to identify an application handler in content serving applications 114 that can produce the requested content. Content server 113 can forward an internal content request (e.g., a program call) to an identified application handler.

[0029] The identifier application handler can return at least a portion of the requested content to content server 113 in response to the internal content request. Depending on content type (e.g., a HypterText Markup Language ("HTML") document, image data, audio data, video data, etc.), an application handler may return requested content differently. For example, when returning an HTML document, a handler may return the complete HTML document as a single portion of content. On the other hand, for video data, a handler may return portions of the video data as the portions of video data are generated.

[0030] Computer system 111 also includes buffer 116. Depending on various inputs content server 113 may or may not buffer requested content in buffer 116 before sending a response that includes requested content. Some inputs can be runtime information, such as, for example, response size and handler content generation speed. Other inputs can be rules, such as, for example, default buffering behavior and handler specific buffering behavior data, that indicate response buffering behavior that is to occur based on other inputs, for example, runtime information.

[0031] When requested content is not buffered, content server 113 can send a response (e.g., using a return electronic address), including the requested content, to the computer system that requested the content. When requested content is buffered in buffer 116, processing components of post processing pipeline 115 can process (e.g., encrypt, filter, buffer, cache, etc.) the requested content. When processing is complete, content server 113 can retrieve buffered and/or processed content from buffer 116. Content server 113 can send a response (e.g., using a return electronic address), including the buffered and/or processed content, to the computer system that requested the content. Web browser 102 can receive and display any buffered and/or unbuffered content.

[0032] Figure 2 illustrates a flow chart of an example method 200 for managing network buffer response behavior. The method 200 will be described with respect to the components and data depicted in computer architecture 100.

[0033] Method 200 includes an act of receiving a request for content from a client (act 201). For example, content server 113 can receive request 181, including content ID 131 (e.g., a URL), from content browser 102. Content browser 102 can send request 181, which can also include an electronic address for computer system 101, to request content for presentation at content browser 102.

[0034] Content server 113 can have default response buffering behavior used when transferring content. For example, a default configuration for content server 113 can be to completely buffer requested content in buffer 116 before sending a response that includes the requested content. Thus by default, a complete response can be available to processing components 141, 142, etc., for post processing. Alternately, a default configuration for content server 113 can be to buffer a specified amount of content (even if the specified amount is less than all of the requested content) before sending a response that includes requested content. For example, the specified amount can be set to a packet size used on network 191. On the other hand, content server 113 can also be configured by default to send requested content without buffering the requested content.

[0035] Content server 113 can have varied default behavior based on content type. For example, default behavior for HTML documents can be complete buffering. On the other hand, default behavior for audio data can be no buffering. Image data may have partially buffering up to the packet size of network 191. However, it should be understood that the expressly described default buffering behaviors based on content type are merely examples, and that other default buffering behaviors for these and other content types are possible.

[0036] Content server 113 can have varied default behavior based on application (whether the applications produce the same or different types of content). In some embodiments, different application handlers are configured to produce the same content type. For example, handler 124 may be a HyperText Preprocessor ("PHP") handler configured to produce HTML documents (e.g., by executing PHP script code). Likewise, hander 134 may be an Active Server Pages ("ASP") handler configured to produce HTML documents (e.g., by executing VB script or Jscript code). A further handler may even be a Cold Fusion handler configured to produce HTML documents (e.g., by processing executing tags written in Cold Fusion Markup Language ("CFML")).

[0037] The default buffering behavior for HTML produced by handler 124 can differ from the default behavior for HTLP produced by handler 134 and other handlers that produce HTML. For example, default behavior for HTML documents produced by handler 124 can include complete buffering. On the other hand, HTML documents produced by handler 134 can include buffering up to a specified size.

[0038] In other embodiments, different application handlers are configured to produce the different content types. The default buffering behavior for these applications handlers can also be varied. However, it should be understood that the expressly described default buffering behaviors based on application are merely examples, and that other default buffering behaviors for these and other applications are possible.

[0039] Method 200 includes an act of mapping the request to a handler configured to server the requested content (act 202). For example, in response to receiving request 181, content server 113 can map request 181 to handler 134. As depicted, dispatch table 112 can have one or more entries that include a content ID, a handler ID, and behavior data. A handler ID represents the identity of a handler at computer system 111 that is configured to produce content requested by a corresponding content ID. Behavior data indicates the response buffering behavior (potentially altering default response buffering behavior) that is to be associated with content produced at the identified handler.

[0040] To map a request, content server 113 can refer to dispatch table 112 to identify a handler ID corresponding to a received content ID. For example, content server 113 can refer to dispatch table 112 to determine that entry 138 corresponds to content ID 131. Within entry 138, handler ID 132 can be a pointer to handler 134. Behavior data 133 indicates the response buffering behavior (potentially altering default response buffering behavior) that is to be associated with content produced at handler 134.

[0041] Entry 128 includes content ID 121, handler ID 122, and behavior data 123. Thus, when appropriate, content server 113 can refer to dispatch table 112 to determine that entry 128 corresponds to content ID 121. Within entry 139, handler ID 122 can be a pointer to handler 124. Behavior data 123 indicates response buffering behavior (potentially altering default response buffering behavior) that is to be associated with content produced at handler 124.

[0042] Method 200 includes an act of accessing buffer behavior for the handler (act 203). For example, content server 113 can access behavior data 133 for handler 134. Behavior data 133 can indicate that handler 134 is to buffer content produced at handler

134 in accordance with default response buffering behavior. For example, when default response buffering behavior is to completely buffer requested content, behavior data can include a value that indicates compliance with default response buffering behavior. On the other, behavior data 133 can indicate that handler 134 is to buffer content produced at handler 134 in accordance with response buffer behavior that differs from default response buffering behavior. For example, when default response buffering behavior is to completely buffer requested content, behavior data can include a value (e.g., zero) that indicates content produces at handler 134 is not to be buffered at all.

[0043] Generally, behavior data can include any data indicative of how response buffering behavior is to be managed, for example, based on content type, application, response size, handler content generation speed, etc.

[0044] Method 200 includes an act of determining that the requested content is to be transferred in accordance with altered response buffering behavior based at least one the buffering behavior data (act 204). For example, content server 113 can determine that requested content produced at handler 134 is to be transferred in accordance with behavior data 133. Behavior data 133 can indicate that requested content produced at handler 134 is to be transferred in accordance with altered response buffering behavior. Thus, behavior data 133 can indicate response buffer behavior for content from handler 134 is an exception to the default response buffering behavior of content server 113.

[0045] Method 200 includes an act of accessing a portion of requested content (act 205). For example, content server 113 can send internal content request 182 (e.g., a program call) to handler 134. In response to receiving internal content request 182, handler 134 can invoke application 136 to produce requested content. Application 136 can then pass any produced content to handler 134. Handler 134 can return content 183 to content server 113. When appropriate, handler 124 can similarly invoke application 126 to produce requested content and can return requested content to content server 113.

[0046] Method 200 includes an act of transferring the portion of requested content in accordance with the altered response buffer behavior (act 206). For example, content server 113 can transfer content 183 in accordance with altered response buffer behavior indicated in behavior data 133.

[0047] Thus, depending on default response buffer behavior, content server 113 may or may not buffer content 183. It may be that altered response buffer behavior indicates that content 183 is to be sent to content browser 102 without any buffering, even though default response buffer behavior indicates (at least partial) buffering for content 183. Thus, content server 113 can send response 186, including content 183, to content browser 102 without buffering content 183.

[0048] On the other hand, it may be that altered response buffer behavior indicates that content 183 is to be (at least partially) buffered in buffer 116, even though default response buffer behavior indicates no buffering for content 183. Thus, content server 113 buffer content 182 in buffer 116. Once buffered, processing component 141, processing component 142, as well as other processing components in post processing pipeline 115 can access and process content 183. In some embodiments, content is buffered in buffer 116 but does not undergo any post processing.

[0049] When appropriate, for example, when content 183 along with other content represents a complete response to request 181, buffered/processed content 184 can be returned to content server 113. In response, content server 113 can send response 187, including buffered/processed content 184, to content browser 102.

[0050] Figure 3 illustrates an example computer architecture 100 that facilitates managing network response buffering behavior. Computer architecture 300 includes computer system 301 and computer system 311. Each of computer system 301 and computer system 311 are connected to network 391. Accordingly, the components can create message related data and exchange message related data (e.g., Internet Protocol ("IP") datagrams and other higher layer protocols that utilize IP datagrams, such as, Transmission Control Protocol ("TCP"), Hypertext Transfer Protocol ("HTTP"), etc.) over network 391.

[0051] Generally, computer system 301 is configured to request, receive, and present Web based content. For example, Web browser 302 can send a request for content including a URL and a return electronic address (e.g., an IP address) onto network 391. Components of network 391 (e.g., routers) can route the request to a Web server identified in the content ID. Computer system 301 can receive requested content in a response from the Web server (e.g., sent a return IP address). Computer system 301 can process the requested content and Web browser 302 can present the requested content.

[0052] Generally, computer system 311 is configured to receive requests for Web based content, process the requests to identify requested Web based content, and return identified Web based content in a response to a received request. For example, Web server 313 can receive request 381, including URL 331, form Web browser 302. Request 381 can also include a return IP address to Web browser 302. Content server 313 can refer to dispatch table 312 to identify an application handler in content serving applications 314 that can produce content requested in request 381. For example, Web server 313 can identify entry 338 as corresponding to URL 331. From entry 338, Web server 313 can determine that handler ID 322 is pointer to handler 324 that can produce the requested content.

[0053] Computer system 311 also includes buffer 316. Web server 313 can be configured with a default response buffer behavior to buffer all content types from all applications in buffer 316 in order optimize response bandwidth and enable content post processing features. However, depending on various inputs (e.g., response size, handler content generation speed, handler specific buffering behavior data, etc.) Web server 313 may or may not buffer requested content in buffer 316 before sending a response that includes requested content.

[0054] In response to receiving request 381 and identifying handler ID 322, Web server 313 can access buffer behavior data 333 (for handler 334). The granularity for buffering behavior data can be a handler mapping. The granularity can ease configuration and management since a handler mapping corresponds to a particular application/content type that has specified response buffering behavior. In addition, handler mappings can be defined per URL such that per URL granularity response buffering behavior is supported.

[0055] For example, entries 328 and 329 correspond to URLs 321 and 331 respectively. However, URLs 321 and 331 both correspond to handler ID 322. Thus, content for URLs 321 and 331 is produced at handler 324. Nonetheless content produced for URL 321 and content produced for URL 331 are indicated to have different response buffering behavior. Content for URL 321 corresponds to behavior data 323 and content for URL 321 corresponds to behavior data 333. Accordingly different portions of content produced at handler 324 can have different response buffering behavior.

[0056] For example, it may be that handler 324 produces image data. URL 321 may be associated with a plurality of smaller images. Thus, behavior data 323 may indicate that for these smaller images the default response buffering behavior (complete buffering) is to be utilized. Web server 313 can access behavior data 323 when a request for content at URL 321 is received.

[0057] On the other hand, URL 331 may be associated with one very large image. Thus, behavior data 333 may indicate that for this larger image the default response buffering behavior (complete buffering) is to be turned off (e.g., buffer size = 0). Web server 313 can access behavior data 333 when a request for content at URL 331 is received.

[0058] When a request for content at URL 371 is received, Web server 313 can also access behavior data 373 to identify response buffering behavior for content produced at handler 374.

[0059] Based on behavior data in an entry, Web server 313 can determine if requested content is to be transferred in accordance with default response buffer behavior or in accordance with altered response buffer behavior indicating an exception to default response buffer behavior. For example, based at least in part on behavior data 333, Web server 313 can determine if content form handler 324 is to be buffered (default) or not buffered (altered).

[0060] In some embodiments, requirements for turning off response buffering can be viewed as hard requirements and soft requirements. A hard requirement is a requirement dictated by a handler. For example, a handler can indicate (through behavior data) that the hander is required to stream data. A soft requirement can balance various inputs to determine if turning response buffering off is appropriate. For example, an administrator can configure behavior data to potentially turn response buffering off balancing whether content is to be streamed, the rigidity of time to first byte requirements, memory usage, network bandwidth usage, need for post processing, response size, handler response generation speed, etc.

[0061] In some embodiments it may be appropriate completely turn off response buffering for a response. For example, when a response is relatively large (e.g., exceeds available memory in buffer 316) it may be detrimental to the performance of computer system 311 or impossible to store in memory. Thus, in these embodiments, buffer behavior data for a handler and/or content type can indicate that response buffering is to be turned off.

[0062] Further, content may have delivery rate requirements and/or time to first byte requirements. Buffering content can impact delivery rate requirements and time to first byte requirements. Thus, in these other embodiments, buffer behavior data for a handler and/or content type can indicate that response buffering is to be turned off.

[0063] Additionally, a handler may provide its own buffering and does not intend for Web server 313 to buffer content again. Double buffering content is an inefficient use of resources. Thus, in these further embodiments, buffer behavior data for a handler and/or content type can indicate that response buffering is to be turned off.

[0064] Web server 313 supports the use of a response buffering threshold that indicates whether content produced at a handler is or is not to be buffered. An administrator can set response buffer thresholds for handlers to implement desired response buffering behavior. A response buffering threshold can be a number that indicates how much data is to be buffered. A response buffering threshold of zero can indicate that response buffering is turned off.

[0065] If the size of produced content is equal to or less than a corresponding buffering threshold, the produced content can be buffered. On the other hand, if the size of produced content is greater than a corresponding buffering threshold, the produced content may not buffered. In some embodiments, the response buffering threshold indicates a specified amount of content that is to be buffered before flushing the content to the client. When content is larger than an available buffer, the buffer can be repeatedly filled and flushed until all content is sent. Web server 313 can be configured to automatically resolve the buffering threshold for each handler mapping.

[0066] Web server 313 can send internal content request 382 to handler 324 to request that handler 324 produce content accessed through URL 331. In response handler 324 can return Web content 383 to Web server 313. Subsequently, handler 324 can also return Web content 384 to Web server 313.

[0067] If buffering is off (e.g., behavior data 333 has a response buffer threshold = 0), Web content 383 and 384 are sent to Web browser 302 as they are received. For example, Web server 313 can send response 386, including Web content 383, to Web browser 302. Subsequently (after receiving Web content 384), Web server 313 can send response 396, including Web content 384, to Web browser 302.

[0068] On the other hand, if buffering is on, Web server 313 can buffer Web content 383 in buffer 316. Subsequently (after receiving Web content 384), Web server 313 can also buffer Web content 384 in buffer 316. Thus, Web content 383.384 is made available to post processing pipeline 315, including compression module 341, encryption module 342, filter module 343, and cache module 344. Accordingly, one or more of these modules can process Web content 383/384 resulting in buffered processed Web content 388. .

[0069] When processing is complete, Web server 313 can retrieve buffered and/or processed Web content 388 from buffer 316. Web server 313 can send response 387 (e.g., using a return IP address), including buffered and/or processed content 388, to Web browser 302. However, in some embodiments, Web content 383 and 384 are buffered without being subjected to any post processing. Web browser 302 can receive and display any buffered and/or unbuffered Web based content.

[0070] In some embodiments, response buffering APIs are used to control response buffering behavior from an application. For example, in Figure 1, application 126 or application 136 can call response buffering APIs to control their own response buffering behavior. Similarly, in Figure 3, application 326 or application 376 can call response buffering APIs to control their own response buffering behavior.

[0071] Response buffering APIs can be used to control and/or enforce hard response buffering requirements. For example, response buffering APIs can be used to enforce streaming behavior based on arbitrary rules applied by an application producing requested content. Response buffering APIs can be used to override default response buffer behavior as well as altered response buffer behavior included in a dispatch table.

[0072] Response buffering APIs can include:

WriteResponse() – used to send data to the client, and may internally use response buffering based on the configuration information for the particular handler

Flush() – used to force the flush of the response buffer to the client. Applications can used this to control the buffering behavior of the server.

DisableResponseBuffering() – used to turn off response buffering for the remainder of a request.

[0073] Accordingly, embodiments of the invention facilitate the ability to control response buffering for different content types and application parts associated with a content server. Further, embodiments provide options to define response buffering behavior in terms of runtime information, such as, for example, response size and handler response generation speed. Thus, for example, a content server can be generally configured to take advantage of response buffering, but still achieve streaming behavior and/or prevent large memory usage by a response buffer.

[0074] Additionally, response buffering behavior for an application can be controlled through configuration settings set by an administrator. Thus, application code does not have to be aware of response buffer semantics of a content server. Alternately, through response buffering APIs, an application can take control of its own response buffering behavior.

[0075] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

What is claimed:

1. At a computer system, a method for managing network response buffering behavior, the method comprising:

an act of a content server receiving a request for content from a client, the content server having a default response buffering behavior used when transferring content;

an act of mapping the request to a handler configured to serve the requested content;

an act of accessing buffering behavior data for the handler;

an act of determining that the requested content is to be transferred in accordance with altered response buffering behavior based at least on the buffering behavior data, the altered response buffering behavior corresponding to the requested content as an exception to the default response buffering;

an act of accessing a portion of the requested content from the handler; and

an act of transferring the portion of requested content to the client in accordance with the altered response buffer behavior.

2. The method as recited in claim 1, wherein the act of receiving a request for content comprises an act of receiving a URL.

3. The method as recited in claim 1, wherein the act of receiving a request for content comprises an act of a Web server receiving a request for content.

4. The method as recited in claim1, wherein the act of mapping the request to a handler configured to serve the requested content comprises an act of mapping a URL to handler ID for the handle.

5. The method as recited in claim 1; wherein the act of accessing buffering behavior data for the handler comprises an act of accessing buffering behavior data that indicates response buffering for the handler is to be turned off.

6. The method as recited in claim 1, wherein the act of accessing buffering behavior data for the handler comprises an act of receiving a response buffer API call from an application corresponding to the handler.

7. The method as recited in claim 1, wherein an act of determining that the requested content is to be transferred in accordance with altered response buffering behavior comprises an act of determining that requested content is not to be buffered because the response exceeds available memory in a response buffer, even though the default response buffering behavior is to buffer requested content.

8. The method as recited in claim 1, wherein an act of determining that the requested content is to be transferred in accordance with altered response buffering behavior comprises an act of determining that requested content has at least one of time to first byte and transfer rate requirements.

9. The method as recited in claim 1, wherein an act of determining that the requested content is to be transferred in accordance with altered response buffering behavior comprises an act of determining that the handler provides its own buffering.

10. The method as recited in claim 1, wherein an act of determining that the requested content is to be transferred in accordance with altered response buffering behavior comprises an act of determining that requested content is to be buffered even though default response buffer behavior indicates that requested content is not to be buffered.

11. The method as recited in claim 1, wherein the act of transferring the portion of requested content to the client in accordance with the altered response buffer behavior comprises an act of the content server sending the portion of requested content directly to the client.

12. The method as recited in claim 1, wherein the act of transferring the portion of requested content to the client in accordance with the altered response buffer behavior comprises an act of buffering the portion of requested content in a response buffer.

13. The method as recited in claim 12, wherein the act of buffering the portion of requested content in a response buffer comprises an act of buffering the portion of requested content along with one or more other portions of content in the response buffer.

14. The method as recited in claim 12, further comprising:

an act of performing a post processing operation on the portion of requested content, the post processing operation selected from among compressing the portion of content, encrypting the portion of content, filtering the portion of content, and caching the portion of content.

15. The method as recited in claim 14, further comprising:

an act of the content server retrieving the processed portion of requested content from the response buffer; and

an act of the content server sending the processed portion of requested content to the client.

16. At a computer system, a method for managing network response buffering behavior for Web based content, the method comprising:

an act of a Web server receiving a URL from a client, the content server having a default response buffering behavior used when transferring Web based content;

an act of mapping the URL to a handler configured to serve requested Web based content for the portion of the Web server namespace represented in the URL;

an act of accessing buffering behavior data for the handler;

an act of determining that the requested Web based content is to be transferred in accordance with altered response buffering behavior based at least on the buffering behavior data, the altered response buffering behavior corresponding to the requested Web based content as an exception to the default response buffering;

an act of accessing a portion of the requested Web based content from the handler; and

·an act of transferring the portion of requested Web based content to the client in accordance with the altered response buffer behavior.

17. The method as recited in claim 16, wherein the act of accessing buffering behavior data for the handler comprises an act of accessing buffering behavior data for a portion of the Web server's namespace selected from among a plurality of portions of the Web server's namespace serviced by the handler.

18. The method as recited in claim 16, wherein the act of determining that the requested Web based content is to be transferred in accordance with altered response buffering behavior comprises an act of determining that the requested Web based content is to be transferred in accordance with altered response buffering behavior based on an application corresponding to a handler.

19. The method as recited in claim 16, wherein the act of determining that the requested Web based content is to be transferred in accordance with altered response buffering behavior comprises an act of determining that the requested Web based content is to be transferred in accordance with altered response buffering behavior based on a content type of the requested Web based content.

20. A computer system, comprising:

one or more processors;

system memory;

one or more computer-readable media having stored thereon computer-executable-instructions representing a content server, the content server configured to:

have default response buffering behavior when transferring content;

receive requests for content from a client;

map requests to a handler configured to serve the requested content;

access buffering behavior data for the handler;

determine that requested content is to be transferred in accordance with altered response buffering behavior based at least on the buffering behavior data, the altered response buffering behavior corresponding to the requested content as an exception to the default response buffering;

access a portions of the requested content from the handler; and

transferring the portion of requested content to the client in accordance with the altered response buffer behavior.

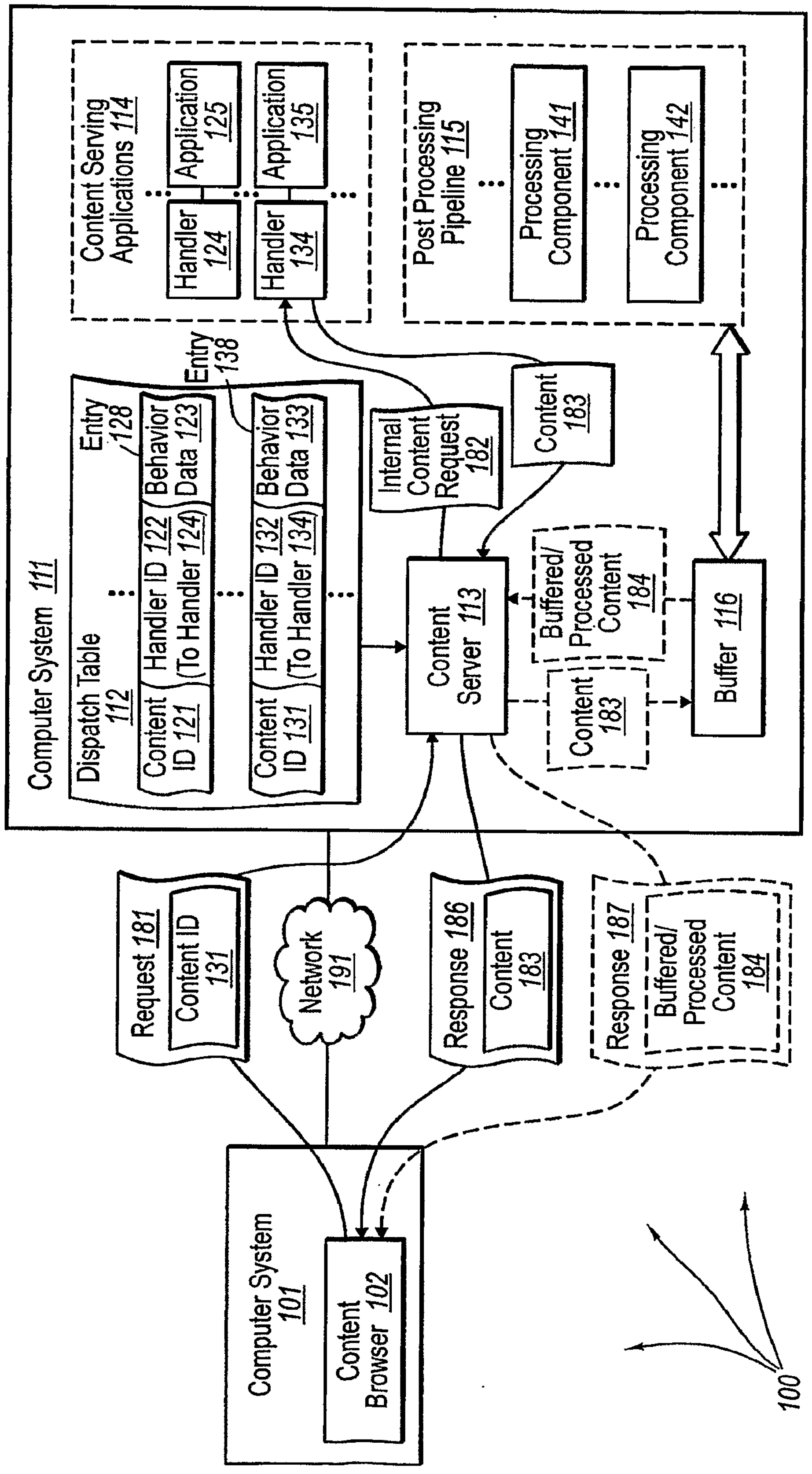

Computer System 111
Dispatch Table 112
Content ID 121
Handler ID 122 (To Handler 124)
Behavior Data 123
Entry 128
Content ID 131
Handler ID 132 (To Handler 134)
Behavior Data 133
Entry 138
Content Serving Applications 114
Handler 124
Application 125
Handler 134
Application 135
Post Processing Pipeline 115
Processing Component 141
Processing Component 142
Internal Content Request 182
Content 183
Content Server 113
Buffered/Processed Content 184
Content 183
Buffer 116
Request 181
Content ID 131
Network 191
Response 186
Content 183
Response 187
Buffered/Processed Content 184
Computer System 101
Content Browser 102
100

FIG. 1

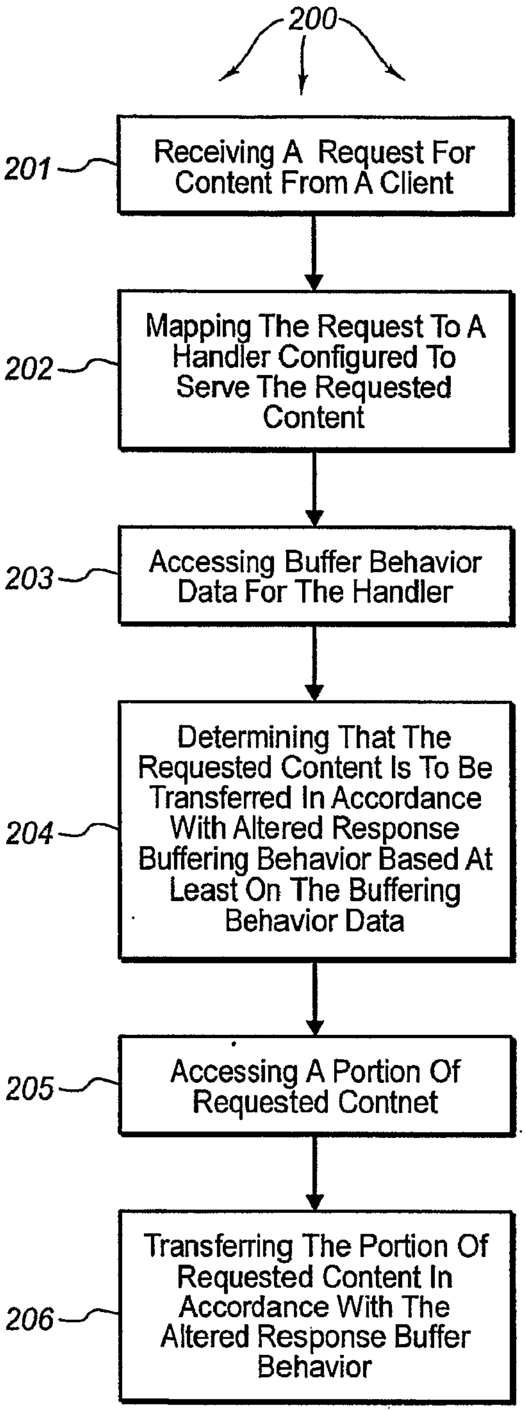

200
201
Receiving A Request For Content From A Client
202
Mapping The Request To A Handler Configured To Serve The Requested Content
203
Accessing Buffer Behavior Data For The Handler
204
Determining That The Requested Content Is To Be Transferred In Accordance With Altered Response Buffering Behavior Based At Least On The Buffering Behavior Data
205
Accessing A Portion Of Requested Contnet
206
Transferring The Portion Of Requested Content In Accordance With The Altered Response Buffer Behavior

FIG. 2

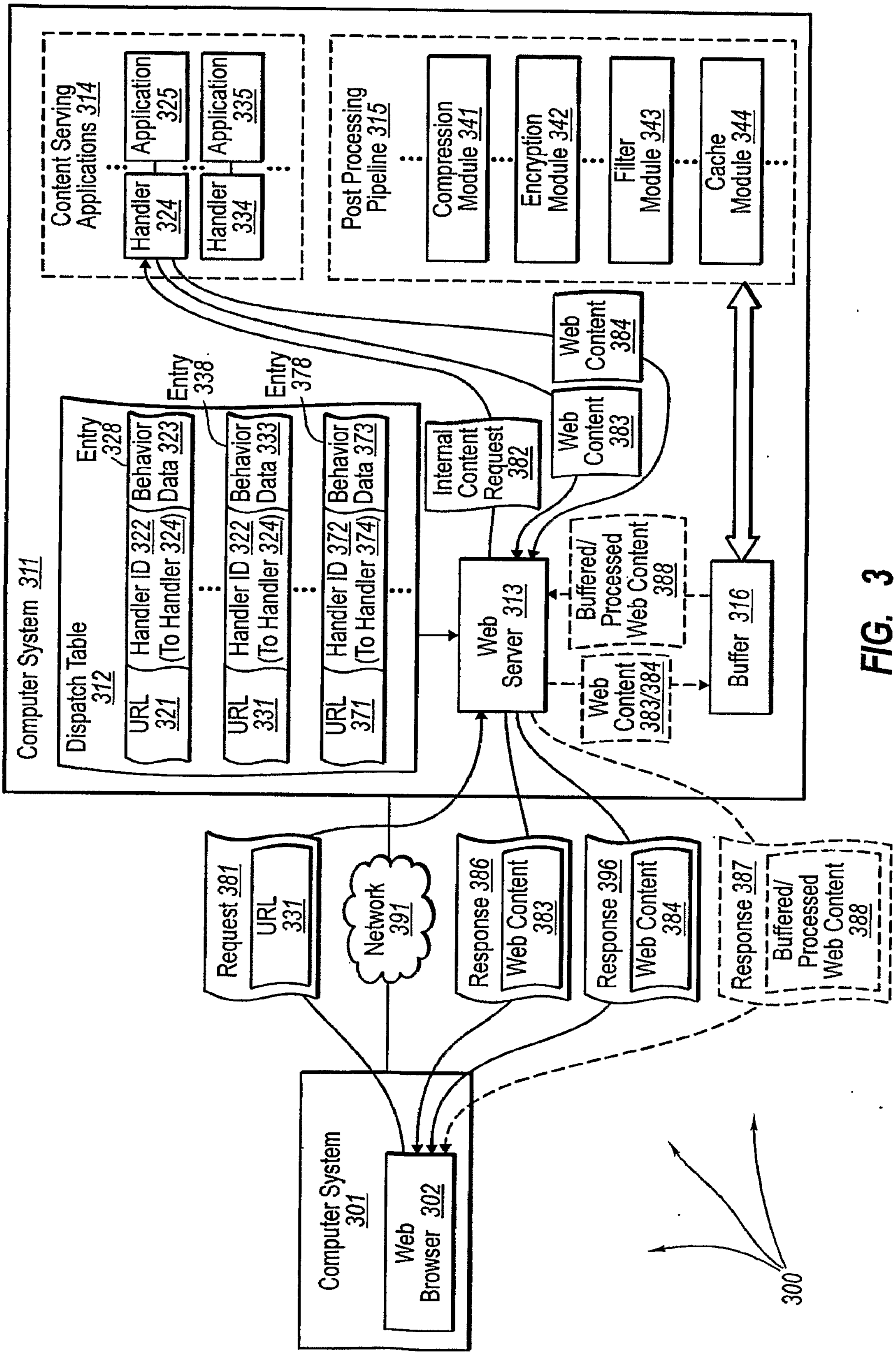

Computer System 311
Dispatch Table 312
URL 321
Handler ID 322 (To Handler 324)
Behavior Data 323
Entry 328
URL 331
Handler ID 322 (To Handler 324)
Behavior Data 333
Entry 338
URL 371
Handler ID 372 (To Handler 374)
Behavior Data 373
Entry 378
Content Serving Applications 314
Handler 324
Application 325
Handler 334
Application 335
Post Processing Pipeline 315
Compression Module 341
Encryption Module 342
Filter Module 343
Cache Module 344
Internal Content Request 382
Web Content 383
Web Content 384
Web Server 313
Buffered/ Processed Web Content 388
Web Content 383/384
Buffer 316
Request 381
URL 331
Network 391
Response 386
Web Content 383
Response 396
Web Content 384
Response 387
Buffered/ Processed Web Content 388
Computer System 301
Web Browser 302
300

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2007/006024

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC8 G06F 17/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean Utility models and applications for Utility models since 1975
Japanese Utility models and applications for Utility models since 1975

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

ekipass "buffering, network, manage, behavior, web, content, URL, handler, response, ID, CDN"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 1020030030050 A (BIZMODELINE CO., LTD.) 18 APR. 2003 See abstract; figures 1,4. claim 1.	1-20
A	KR 1020020043972 A (FEELAMINT NETWORKS CO., LTD.) 12 JUN. 2002 See abstract; figures 2,6,7. claims 1~3,12,15.	1-20
A	KR 1020020076028 A (BIZMODELINE CO., LTD.) 09 OCT. 2002 See abstract; figures 1,2. claims 1,2.	1-20
A	US 20050216569 A1 (CRESCENZO COPPOLA et al.) 29 SEP. 2005 See abstract; figures 3,4. claims 1,7.	1-20

□ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"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
"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
"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
"&" document member of the same patent family

Date of the actual completion of the international search 21 AUGUST 2007 (21.08.2007)	Date of mailing of the international search report **21 AUGUST 2007 (21.08.2007)**
Name and mailing address of the ISA/KR Korean Intellectual Property Office 920 Dunsan-dong, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 82-42-472-7140	Authorized officer KIM, Jung Jin Telephone No. 82-42-481-5962

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2007/006024

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
KR1020030030050	18.04.2003	None	
KR1020020043972A	12.06.2002	None	
KR1020020076028A	09.10.2002	None	
US20050216569A1	29.09.2005	AU2003224096AA	03.11.2003
		BR200304537A	03.08.2004
		CA2482952AA	30.10.2003
		EP1497966A1	19.01.2005
		KR1020040103980	09.12.2004
		WO2003090423A1	30.10.2003