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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 October 2013 (31.10.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/163648 A2

(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/US2013/038647

(22) International Filing Date:
29 April 2013 (29.04.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/639,120 27 April 2012 (27.04.2012) US

(71) Applicant: **F5 NETWORKS, INC.** [US/US]; 401 Elliott Avenue West, Seattle, WA 98119 (US).

(72) Inventors: **WALKER, Anthony**; F5 Networks, Inc., 401 Elliott Avenue West, Seattle, WA 98119 (US). **EARNHART, Michael**; F5 Networks, Inc., 401 Elliott Avenue West, Seattle, WA 98119 (US).

(74) Agents: **GALLO, Nicholas, J.** et al.; LeClairRyan, A Professional Corporation, 70 Linden Oaks, Suite 210, Rochester, NY 14625 (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, BN, 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, PA, 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))*

WO 2013/163648 A2

(54) Title: METHODS FOR OPTIMIZING SERVICE OF CONTENT REQUESTS AND DEVICES THEREOF

(57) Abstract: A method, non-transitory computer readable medium, and network traffic management apparatus that receives a request for content from a client computing device. A length of the content is determined. A plurality of requests for a portion of the length of the content is sent to a plurality of server computing devices, wherein the portion of the length of the content is specified as a byte range in a range header of each of the plurality of requests. A plurality of responses to the plurality of requests is received. At least a subset of the plurality of responses is output to the client computing device.

METHODS FOR OPTIMIZING SERVICE OF CONTENT REQUESTS AND DEVICES THEREOF

[0001] This application claims the benefit of U.S. Provisional Patent Application Serial No. 61/639,120, filed April 27, 2012, which is hereby incorporated by reference in its entirety.

FIELD

[0002] This technology generally relates to network traffic management apparatuses and methods and, more particularly, to methods for optimizing service of content requests by server computing devices in a distributed network environment and devices thereof.

BACKGROUND

[0003] Network resource utilization and traffic distribution in a distributed network environment can be optimized using a network traffic management apparatus configured to communicate with client computing devices and content provider devices, such as a plurality of server computing devices in a server pool. The network traffic management apparatus is utilized to receive requests from client computing devices and communicate with the server computing devices to open connections that can be utilized to service the requests.

[0004] As an intermediary or proxy device, the network traffic management device can distribute client computing device requests across the server computing devices of the server pool. One such method of distribution is to maintain the number of open connections with each server computing device and distribute new requests from client computing devices to the server computing device having the least number of open connections.

[0005] However, the connection count is not a robust metric and server computing device load can remain relatively unbalanced using connection-based techniques which assume all connections are equal with respect to utilization of server computing device resources. For example, some requests will likely be for content of a relatively large size which will require more resources to service and

other requests will likely be for content of a relatively small size requiring fewer resources and resulting in faster service.

[0006] Accordingly, unbalanced loads or hotspots associated with substantial memory, processor cycle, and/or bandwidth usage for one or more server computing devices as compared to other server computing devices in the server pool can result, thereby negatively affecting response time and associated user experience.

SUMMARY

[0007] A method for optimizing service of one or more content requests includes receiving at a network traffic management apparatus a request for content from a client computing device. A length of the content is determined with the network traffic management apparatus. A plurality of requests for a portion of the length of the content is sent with the network traffic management apparatus to a plurality of server computing devices, wherein the portion of the length of the content is specified as a byte range in a range header of each of the plurality of requests. A plurality of responses to the plurality of requests is received at the network traffic management apparatus. At least a subset of the plurality of responses is output with the network traffic management apparatus to the client computing device.

[0008] A non-transitory computer readable medium having stored thereon instructions for optimizing service of one or more content requests comprising machine executable code which when executed by a processor, causes the processor to perform steps including receiving a request for content from a client computing device. A length of the content is determined. A plurality of requests for a portion of the length of the content is sent to a plurality of server computing devices, wherein the portion of the length of the content is specified as a byte range in a range header of each of the plurality of requests. A plurality of responses to the plurality of requests is received. At least a subset of the plurality of responses is output to the client computing device.

[0009] A network traffic management apparatus includes at least one of configurable hardware logic configured to be capable of implementing or a processor or a network interface controller coupled to a memory and configured to execute programmed instructions stored in the memory including receiving a request for content from a client computing device. A length of the content is determined. A plurality of requests for a portion of the length of the content is sent to a plurality of server computing devices, wherein the portion of the length of the content is specified as a byte range in a range header of each of the plurality of requests. A plurality of responses to the plurality of requests is received. At least a subset of the plurality of responses is output to the client computing device.

[0010] This technology provides a number of advantages including methods, non-transitory computer readable medium, and network traffic management apparatus that optimize service of content requests by server computing devices to thereby balance server load and reduce the likelihood of a hotspot developing in the server pool. Additionally, with this technology, latency can be reduced, a maximum size of content requests sent to the server computing devices can be guaranteed, and the elapsed time from the client computing device request to receipt by the client computing device of the first or last byte of the requested content can be reduced.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 is a block diagram of a network environment which incorporates an exemplary network traffic management apparatus;

[0012] FIG. 2 is a block diagram of the exemplary network traffic management apparatus; and

[0013] FIG. 3 is a flowchart of an exemplary method for optimizing service of content requests.

DETAILED DESCRIPTION

[0014] An exemplary network environment 10 with a network traffic management apparatus 12, client computing devices 14(1)-14(n), and server

computing devices 16(1)-16(n) is illustrated in FIGS. 1 and 2. In this example, the network traffic management apparatus 12, client computing devices 14(1)-14(n), and server computing devices 16(1)-16(n) are coupled together by local area networks (LANs) 18 and 20 and wide area network (WAN) 22, although other types and numbers of systems, devices, components and other elements in other configurations which are coupled together in other manners can be used. This technology provides a number of advantages including methods, non-transitory computer readable medium, and apparatus that optimize service of content requests to balance server computing device load and reduce the risk of developing a hotspot in a server pool.

[0015] Referring more specifically to FIGS. 1 and 2, the network traffic management apparatus 12 is coupled to the client computing devices 14(1)-14(n) by the LAN 18 and WAN 20. In this example, the network traffic management apparatus 12 is further coupled to the server computing devices 16(1)-16(n) by the LAN 20. Although network traffic management apparatus 12 is shown in this example, other network devices configured to generate, send, and receive network communications and coupled together via other topologies can also be used. While not shown, the environment 10 also may include additional network components, such as routers, switches and other devices, which are well known to those of ordinary skill in the art and thus will not be described here.

[0016] The network traffic management apparatus 12 may perform any number of functions, such as optimizing, securing, and accelerating the network traffic between the client computing devices 14(1)-14(n) and the server computing devices 16(1)-16(n). The network traffic management apparatus 12 includes at least one processor or CPU 24, a memory 26, optional cache memory 28, optional configurable hardware logic 30, an input and/or display device interface 32, and a network interface controller 34 which are coupled together by a bus 36, although the network traffic management apparatus 12 may include other types and numbers of elements in other configurations. In this example, the bus 36 is a hyper-transport bus, although other bus types and links may be used, such as PCI.

[0017] The processor 24 of the network traffic management apparatus 12 may execute one or more computer-executable instructions stored in the memory 26 of the network traffic management apparatus 12 for managing network traffic and/or optimizing service of content requests. The processor 24 of the network traffic management apparatus 12 may comprise one or more central processing units (CPUs) or general purpose processors with one or more processing cores, such as AMD® processor(s), although other types of processor(s) could be used (e.g., Intel®).

[0018] The memory 24 of the network traffic management apparatus 12 stores these programmed instructions for one or more aspects of the present technology as described and illustrated herein, although some or all of the programmed instructions could be stored and executed elsewhere. A variety of different types of memory storage devices, such as a random access memory (RAM) or a read only memory (ROM) in the system or a floppy disk, hard disk, CD ROM, DVD ROM, or other computer readable medium which is read from and written to by a magnetic, optical, or other reading and writing system that is coupled to the processor 24, can be used for the memory 26. The optional cache memory of the network traffic management apparatus 12 can be a static random access memory (SRAM) device for example, although other forms of memory can also be used.

[0019] The optional configurable hardware logic 30 of the network traffic management apparatus 12 may comprise specialized hardware configured to be capable of implementing one or more steps of this technology as illustrated and described with reference to the examples herein. By way of example only, the optional configurable logic 30 may comprise one or more of field programmable gate arrays (FPGAs), field programmable logic devices (FPLDs), application specific integrated circuits (ASICs) and/or programmable logic units (PLUs).

[0020] The input and display device interface 32 of the network traffic management apparatus 12 enables a user, such as an administrator, to interact with the network traffic management apparatus 12, such as to input and/or view data and/or to configure, program and/or operate it by way of example only. Input

devices configured to communicate with the input and/or display device interface 32 may include a keyboard and/or a computer mouse and display devices configured to communicate with the input and/or display device interface 32 may include a computer monitor, although other types and numbers of input devices and display devices could also be used.

[0021] The network interface controller 34 operatively couples and communicates between the network traffic management apparatus 12, the client computing devices 14(1)-14(n), and server computing devices 16(1)-16(n), which are all coupled together by the LANs 18 and 20 and WAN 22, although other types and numbers of communication networks or systems with other types and numbers of connections and configurations to other devices and elements. By way of example only, the LANs 18 and 20 and WAN 22 can use TCP/IP over Ethernet and industry-standard protocols, including NFS, CIFS, SOAP, XML, LDAP, and SNMP, although other types and numbers of communication networks, can be used.

[0022] The LANs 18 and 20 in this example may employ any suitable interface mechanisms and network communication technologies including, for example, teletraffic in any suitable form (e.g., voice, modem, and the like), Public Switched Telephone Network (PSTNs), Ethernet-based Packet Data Networks (PDNs), combinations thereof, and the like. The WAN 22 may comprise any wide area network (e.g., Internet), although any other type of traffic network topology may be used.

[0023] Each of the client computing devices 14(1)-15(n) and server computing devices 16(1)-16(n) includes a central processing unit (CPU) or processor, a memory, a network interface device, and an I/O system, which are coupled together by a bus or other link, although other numbers and types of network devices could be used. The client computing devices 14(1)-14(n) may run interface application(s), such as a Web browser, that may provide an interface to make requests for and receive content stored on one or more of the server computing devices 16(1)-16(n) via the LANs 18 and 20 and/or WAN 22.

[0024] Generally, the server computing devices 16(1)-16(n) process requests received from requesting client computing devices 14(1)-14(n) via the LANs 18 and 20 and/or WAN 22 according to the HTTP-based application RFC protocol or the CIFS or NFS protocol for example. Various network processing applications, such as CIFS applications, NFS applications, HTTP Web Server applications, and/or FTP applications, may be operating on the server computing devices 16(1)-16(n) and transmitting content (e.g., files, Web pages) to the client computing devices 14(1)-14(n) in response to requests for the content from the client computing devices 14(1)-14(n).

[0025] The server computing devices 16(1)-16(n) may provide data or receive data in response to requests directed toward applications on the server computing devices 16(1)-16(n) from the client computing devices 14(1)-14(n). The server computing devices 16(1)-16(n) may be hardware or software or may represent a system with multiple server computing devices 16(1)-16(n) in a server pool, which may include internal or external networks. In this example the server computing devices 16(1)-16(n) may be any version of Microsoft® IIS servers or Apache® servers, although other types of server computing devices 16(1)-16(n) may be used. Further, additional server computing devices 16(1)-16(n) may be coupled to one of the LANs 18 and 20 and many different types of applications may be available on each of the server computing devices 16(1)-16(n).

[0026] Although an exemplary network environment with the network traffic management apparatus 12, client computing devices 14(1)-14(n), server computing devices 16(1)-16(n), LANs 18 and 20 and WAN 22 are described and illustrated herein, other types and numbers of systems, devices, components, and elements in other topologies can be used. It is to be understood that the systems of the examples described herein are for exemplary purposes, as many variations of the specific hardware and software used to implement the examples are possible, as will be appreciated by those skilled in the relevant art(s).

[0027] In addition, two or more computing systems or devices can be substituted for any one of the systems or devices in any example. Accordingly, principles and advantages of distributed processing, such as redundancy and

replication also can be implemented, as desired, to increase the robustness and performance of the devices and systems of the examples. The examples may also be implemented on computer system(s) that extend across any suitable network using any suitable interface mechanisms and traffic technologies.

[0028] The examples may also be embodied as a non-transitory computer readable medium having instructions stored thereon for one or more aspects of the present technology as described and illustrated by way of the examples herein, as described herein, which when executed by a processor, cause the processor to carry out the steps necessary to implement the methods of the examples, as described and illustrated herein.

[0029] An exemplary method for optimizing service of content requests will now be described with reference to FIGS. 1-3. In this example, in step 300, the network traffic management apparatus 12 receives a request for content from one of the client computing devices 14(1)-14(n). The request for content can be a hypertext transfer protocol (HTTP) request for a hypertext markup language (HTML) web page document, a video, music, and/or graphic file, or a portable document format (PDF) file, for example, or any other type of content.

[0030] In step 302, the network traffic management apparatus 12 determines whether the length of the requested content is included in the cache memory 28. The length can be the size of the content as represented by a number of bytes and can be stored in the cache memory 28 in an entry associated with an indication of the content, for example. If the network traffic management apparatus 12 determines that the length of the requested content is not stored in the cache memory 28, or the associated entry in the cache memory is not valid, then the No branch is taken to step 306.

[0031] In step 306, the network traffic management apparatus 12 sends an HTTP head request to one of the server computing devices 16(1)-16(n) and receives the length of the content in response, although other methods for determining the length of the content can be used. Optionally, in step 308, the network traffic management apparatus 12 stores the length of the content in the

cache memory 28 as associated with an indication of the requested content so that it can be retrieved and used by the network traffic management apparatus 12 when processing subsequent requests for the content.

[0032] Referring back to step 302, if the network traffic management apparatus 12 determines that the length is included in the cache memory 28, then the Yes branch is taken to step 310. In step 310, the network traffic management apparatus 12 retrieves the length from an entry of the cache memory 28 corresponding to the requested content.

[0033] In step 312, the network traffic management apparatus 12 optionally determines whether one or more threshold conditions are satisfied. Exemplary threshold conditions include whether the requested content is larger than a specified length, whether a file type of the requested content indicated in the request matches one or more specified file types, or whether at least a portion of a path specified in the content request matches one or more specified paths, although other threshold conditions can also be used. The threshold conditions can be established by a manufacturer of the network traffic management apparatus and/or can be configurable by an administrator of the network traffic management apparatus.

[0034] If at least one of the threshold conditions is not satisfied, then the No branch is taken to step 314. In step 314, the network traffic management apparatus 12 services the request from one of the server computing devices 16(1)-16(n) based on an established policy and/or load balancing technique. While servicing the request for content, or during any of steps 302-312, the network traffic management apparatus 12 can receive one or more additional requests for content from one or more of the client computing devices 14(1)-14(n) in the step 300.

[0035] Referring back to step 312, if the network traffic management apparatus 12 determines that at least one of the threshold conditions is satisfied, then the Yes branch is taken to step 316. Accordingly, a content length, file type, path, and/or any other attribute of the content or the request, for example, can be

specified to filter those requests most likely to benefit from the optimization techniques described and illustrated herein with respect to steps 316-328.

[0036] In step 316, the network traffic management apparatus 12 sends a plurality of requests for a portion of the length of the content to a plurality of the server computing devices 16(1)-16(n). In one example, the portion of the length of the content is specified as a byte range in a range header of each of the plurality of requests. Accordingly, the request received in step 300 is split by the network traffic management apparatus 12 into a plurality of requests sent to a plurality of server computing devices 16(1)-16(n). By using more than one of the server computing devices 16(1)-16(n) to service the original request, the risk that one of the server computing devices 16(1)-16(n) may become a hotspot and/or relatively over utilized, such as in the event the requested content is relatively large, is substantially reduced.

[0037] The byte ranges included in each of the plurality of requests can indicate discrete portions of the length of the content, partially overlapping portions of the length of the content, and/or completely overlapping portions of the length of the content. However, in order to ensure effective service of the request, the network traffic management apparatus 12 must send at least one request for each portion of the length of the content.

[0038] In step 318, the network traffic management apparatus 12 determines whether sufficient portions of the requested content have been received. In order to determine whether sufficient portions of the requested content have been received, the network traffic management apparatus 12 determines whether at least a subset of any received responses include an initial byte range portion of the requested content. If the received responses do not include an initial portion of the requested content greater than a threshold, or if sufficient portions of the content have not been received based on any other criteria, then the No branch is taken to step 322. The condition in step 320 will not be satisfied immediately subsequent to the network traffic management apparatus sending the plurality of requests in step 316 as, generally, insufficient time will have elapsed to receive any responses to the plurality of requests.

[0039] In step 322, the network traffic management apparatus determines whether a failure condition has been satisfied. In one example, the failure condition is a failure of the network traffic management apparatus to receive one or more responses that include one or more byte ranges of the requested content within a specified time period, although other failure conditions can be used. If the network traffic management apparatus 12 determines that a failure condition has not been satisfied, then the No branch is taken to step 322. The condition in step 322 will not be initially satisfied as the elapsed time subsequent to sending the plurality of requests on an initial pass will not generally be greater than the specified time period.

[0040] In step 322, the network traffic management apparatus 12 receives a response to one or more of the plurality of requests. In some examples, the responses will include various byte range portions of the length of the content. In some examples, step 322 is performed in parallel with any of steps 318, 320, 324, 326, or 328. Referring back to step 320, if the network traffic management apparatus 12 determines, generally in a pass subsequent to an initial pass, a failure condition has been satisfied, then the Yes branch is taken to step 324.

[0041] In step 324, the network traffic management apparatus 12 requests at least one missing portion of the length of the content, or portion that has not been received during a specified time period, from at least one of the server computing devices 16(1)-16(n) from which the missing portion was not previously requested. Thereby, the missing portion of the length of the content is not requested twice from one of the server computing devices 16(1)-16(n) that may have failed to promptly respond to the first request for that portion. Subsequent to receiving one or more responses in step 322 or retrieving the missing portion of the length of the content in step 324, the network traffic management apparatus 12 can proceed to step 318.

[0042] If, in step 318, the network traffic management apparatus determines, generally in a pass subsequent to an initial pass, sufficient portions of the requested content have been received, then the Yes branch is taken to step 326. In step 326, the network traffic management apparatus 12 prepares and sends at

least a subset of the responses received in step 322 to the requesting one of the client computing devices 14(1)-14(n). Accordingly, in this example, the network traffic management apparatus 12 can send an initial portion, based on byte range order, of the requested content to the requesting one of the client computing devices 14(1)-14(n) prior to receiving responses including byte ranges representing subsequent portions of the requested content.

[0043] In one example, the initial portion can be a single byte range such that the byte ranges are sent to the requesting one of the client computing devices 14(1)-14(n) as they are received, as long as all lower byte range portions have previously been sent to the requesting one of the client computing devices 14(1)-14(n) and the byte range order is maintained. In another example, the initial portion can be the entire portion of the requested content such that the network traffic management apparatus 12 waits until responses including byte ranges representing all portions of the requested content are received before sending any response to the requesting one of the client computing devices 14(1)-14(n). In yet other examples, the initial portion can be determined based on any other criteria such as a threshold size, number of byte ranges, or percentage of the requested content, for example.

[0044] In order to prepare the at least a subset of the responses, the network traffic management apparatus reformats the responses such as by organizing the byte ranges, discarding any overlapping byte ranges, generating appropriate response headers, for example. The at least a subset of the responses can be sent by the network traffic management apparatus 12 as a single HTTP response, such that each communication includes substantially the same HTTP response header. Accordingly, in one example, the responses can be buffered, organized, and/or arranged, and sent to the requesting one of the client computing devices 14(1)-14(n) in byte range order.

[0045] In step 328, the network traffic management apparatus 12 determines whether a response to the request received from the requesting one of the client computing devices 14(1)-14(n) has been completed. In this example, the network traffic management apparatus 12 determines whether the subset of the

plurality of responses sent to the requesting one of the client computing devices 14(1)-14(n) in step 326 represents the entire requested content. In some examples, the responses from the server computing devices 16(1)-16(n) are prepared and sent to the requesting one of the client computing devices 14(1)-14(n) in byte range order. In these examples, the network traffic management apparatus 12 can determine whether a response has been completed in step 328 based on whether the last byte range of the length of the requested content has been sent to the requesting one of the client computing devices 14(1)-14(n). Other methods of determining whether a response has been completed can also be used.

[0046] If the network traffic management apparatus 12 determines in step 328 that a response has not been completed, then the No branch is taken back to step 322 and one or more additional responses are received by the network traffic management apparatus 12 from the server computing devices 16(1)-16(n). If the network traffic management apparatus 12 determines a valid response has been completed, then the Yes branch is taken to step 300 and a new request for content is received from one of the client computing devices 14(1)-14(n). The network traffic management apparatus 12 can also perform step 300 in parallel to any of steps 302-328 such that multiple requests for content are processed in parallel.

[0047] Thereby, the requesting one of the client computing devices 14(1)-14(n) receives a response to the initial request for content and is unaware that the request has been split into a plurality of requests for various byte range portions of the length of the requested content. In addition to mitigating hotspots, splitting the content request into a plurality of requests provides several advantages as described and illustrated below with respect to several exemplary techniques for optimizing service of content requests received from the client computing devices 14(1)-14(n).

[0048] In one example, steps 300-310, 314, and 318-328 proceed as described and illustrated earlier. However, in step 312, the network traffic management apparatus 12 determines whether the threshold condition of whether the requested content is greater than a specified size is satisfied. If the network traffic management apparatus 12 determines in step 312 that the requested content

is not greater than a specified size, then the No branch is taken to step 314. Accordingly, in this example, the content request is not split into a plurality of requests for portions of the content when the content is of a relatively small size. Thereby, overhead is reduced with respect to requests that can otherwise be serviced relatively quickly and with relatively low consumption of server resources using standard policy and/or load balancing techniques in step 314.

[0049] In this example, if the network traffic management apparatus 12 determines in step 312 that the requested content is greater than a specified size, then the Yes branch is taken to step 316. In this example, the byte range specified in the range header of each of the requests sent in step 316 is based on a total number of server computing devices 16(1)-16(n) in the server pool. Accordingly, each request includes a byte range in a range header which is equal to the length of the content, as determined in step 310 or 306, divided by the number of server computing devices 16(1)-16(n). Additionally, one of the plurality of requests is sent to each of the plurality of server computing devices 16(1)-16(n).

[0050] Thereby, portions of the requested content of substantially the same size are requested from each of the server computing devices 16(1)-16(n). Accordingly, in this example, in addition to reducing the risk of developing a hotspot, the network traffic management apparatus 12 can advantageously guarantee balanced distribution of the load with respect to the size of the content requested from each of the server computing devices 16(1)-16(n).

[0051] In another example, steps 300-314 and 318-328 proceed as described and illustrated earlier. However, in step 316, the byte range specified in the range header of at least a subset of the plurality of requests sent by the network traffic management apparatus 12 is not greater than a maximum byte range size, not greater than a network frame payload size, or substantially equal to a uniform byte range size. The maximum byte range size, network frame payload size, and/or uniform byte range size can be established by a manufacturer of the network traffic management apparatus 12 and/or can be configurable by an administrator of the network traffic management apparatus 12, for example.

[0052] By limiting the byte range size to a maximum size, established network frame payload, or other uniform size in this example, each of the server computing devices 16(1)-16(n) will be servicing requests for relatively small portions of the requested content or for portions that are substantially the same size. Thereby, the potential for the server computing devices 16(1)-16(n) to develop a hotspot will therefore be reduced. Limiting the byte range size included in the range headers of at least a subset of the plurality of requests can have additional advantages described and illustrated below.

[0053] By limiting the byte ranges to a maximum size, the network traffic management apparatus 12 can guarantee that the server computing devices 16(1)-16(n) will never service requests for portions of the content larger than the maximum size. Accordingly, the server computing devices 16(1)-16(n) can advantageously be configured and optimized based on the constraint of the maximum size. For example, the server computing devices 16(1)-16(n) can be configured based on the assumption that memory space larger than the maximum size will never be allocated in order to service a request for a portion of the requested content.

[0054] By limiting the byte ranges to a size not greater than a network frame payload size, network latency, including response time of the server computing devices 16(1)-16(n), can advantageously be reduced. In one example, the network frame payload size can be equal to the maximum frame payload size for Ethernet compatible devices (e.g. 1500 bytes).

[0055] By limiting the bye ranges to a uniform byte range size, the effectiveness of caching techniques utilized by the server computing devices 16(1)-16(n) can be increased. By splitting the original content request into a plurality of requests specifying a uniform byte range size, subsequent requests for the content received from the client computing devices 14(1)-14(n) can also be split into a plurality of requests specifying the same uniform byte range size.

[0056] Accordingly, in examples in which the portions of the content requested from the server computing devices 16(1)-16(n) are cached by the server

computing devices 16(1)-16(n), subsequent requests for the portions of the content can be retrieved from cache rather than slower storage memory of the server computing devices 16(1)-16(n). The uniform byte range size can be the maximum size, the network frame payload size, or the length of the content divided by the number of server computing devices 16(1)-16(n), for example, although other uniform byte range sizes can also be used.

[0057] In yet another example, steps 300-310, 320-324, and 328 proceed as described and illustrated earlier. However, in step 312, the network traffic management apparatus 12 determines whether the threshold condition of whether the requested content is of a specified video file type, as determined based on the file extension (e.g. MPEG or WMV) of the requested content, is satisfied. If the network traffic management apparatus 12 determines in step 312 that the requested content is not of a specified video file type, then the No branch is taken to step 314.

[0058] Accordingly, in this example, content requests are not split into a plurality of requests for portions of the content when the content is not a video file. As video files are likely to be relatively large in size, overhead is reduced with respect to requests that can otherwise be serviced relatively quickly and with relatively low consumption of server resources using standard policy and/or load balancing techniques in step 314.

[0059] If the network traffic management apparatus 12 determines in step 312 that the requested content is not of a specified video file type, then the Yes branch is taken to step 316. In step 316, the network traffic management apparatus 12 sends one or more requests for relatively low byte range portions of the content to one or more relatively high performance ones of the server computing devices 16(1)-16(n). Additionally, the network traffic management apparatus 12 sends one or more requests for relatively high byte range portions of the content to one or more relatively low performance ones of the server computing devices 16(1)-16(n). Optionally, the requests for relatively low byte range portions of the content can be sent to one or more of the server computing devices 16(1)-16(n) in relatively close geographic proximity to the network traffic

management apparatus 12, with relatively fast hardware or storage devices, statically or dynamically categorized or profiled as relatively fast, or otherwise likely to respond more quickly than one or more other of the server computing devices 16(1)-16(n).

[0060] In step 318, the network traffic management apparatus 12 determines whether sufficient portions of the content have been received. In this example, the network traffic management apparatus 12 determines whether a response has been received from one of the server computing devices 16(1)-16(n) that includes the lowest byte range portion of the content that has not previously been sent to the requesting one of the client computing devices 14(1)-14(n). If the network traffic management apparatus 12 determines a response has not been received from one of the server computing devices 16(1)-16(n) that includes the lowest byte range portion of the content that has not previously been sent to the requesting one of the client computing devices 14(1)-14(n), then the No branch is taken to step 320.

[0061] If the network traffic management apparatus 12 determines a response has been received from one of the server computing devices 16(1)-16(n) that includes the lowest byte range portion of the content that has not previously been sent to the requesting one of the client computing devices 14(1)-14(n), then the Yes branch is taken to step 326. In step 326, the network traffic management apparatus 12 sends response(s) to the request(s) for relatively low byte range portions of the content to the requesting one of the client computing devices 16(1)-16(n) in byte range order and irrespective of whether response(s) to the request(s) for relatively high byte range portions of the content have been received.

[0062] Thereby, relatively low byte range portions of the content will be sent to the requesting one of the client computing devices 14(1)-14(n) relatively quickly while the network traffic management apparatus 12 is requesting and/or receiving relatively high byte range portions of the content. In this example, the relatively low byte range portions of the content will likely be initial segments of a video file. Accordingly, the requesting one of the client computing devices 14(1)-

14(n) can process the initial segments and begin playing the video, while the remaining portions of the video are received from the network traffic management apparatus 12. Therefore, a user of the requesting one of the client computing devices 14(1)-14(n) can receive and interact with initial video segments in relatively less time, thereby improving the user's experience and reducing the time required for the requesting one of the client computing devices 14(1)-14(n) to receive the first byte of the content.

[0063] Additionally, in this example, a plurality of the server computing devices 16(1)-16(n) can respond to requests for portions of the content instead of one of the server computing devices 16(1)-16(n) servicing a request for the video file. As the video file is likely to be relatively large in size, distributing the content request can reduce the risk of developing a hotspot in the server pool resulting from one of the server computing devices 16(1)-16(n) servicing the request for relatively large content.

[0064] In yet another example, steps 300-310, 320-324, and 328 proceed as described and illustrated earlier. However, in step 312, the network traffic management apparatus 12 determines whether the threshold condition of whether at least a portion of a path specified in the request for content matches a specified path is satisfied. The specified path can be a URL, for example, identifying a path at which content is located that, when accessed or requested, must be sent to the requesting one of the client computing devices 14(1)-14(n) as quickly as possible. If the network traffic management apparatus 12 determines in step 312 that at least a portion of the path specified in the request for content does not match the specified path, then the No branch is taken to step 314 and the request is serviced as described and illustrated earlier.

[0065] If the network traffic management apparatus 12 determines in step 312 that at least a portion of the path specified in the content request does match the specified path, then the Yes branch is taken to step 316. In step 316, the network traffic management apparatus 12 sends the plurality of requests such that the byte ranges specified in the range headers of at least a subset of the plurality of requests completely overlap. Optionally, each byte range portion of the requested

content can be requested from each of the server computing devices 16(1)-16(n). As each of the server computing devices 16(1)-16(n) receives the same number of requests and the entirety of the content is requested from each of the server computing devices 16(1)-16(n), the load will be evenly distributed and the risk of developing a hotspot will be reduced.

[0066] In step 318, the network traffic management apparatus 12 determines whether sufficient portions of the content have been received. In this example, the network traffic management apparatus 12 determines whether a response is received that includes the lowest overlapping byte range portion of the content that has not previously been sent to the requesting one of the client computing devices 14(1)-14(n). If the network traffic management apparatus 12 determines a response is received that does not include the lowest overlapping byte range portion of the content that has not previously been sent to the requesting one of the client computing devices 14(1)-14(n), then the No branch is taken to step 320.

[0067] If the network traffic management apparatus 12 determines a response is received that does include the lowest overlapping byte range portion of the content that has not previously been sent to the requesting one of the client computing devices 14(1)-14(n), then the Yes branch is taken to step 326. In step 326, the network traffic management apparatus 12 sends to the requesting one of the client computing devices 14(1)-14(n) each first received response that includes an overlapping byte range portion of the content in byte range order. Because, in this example, each byte range portion of the requested content is requested from each of the server computing devices 16(1)-16(n), the request for content can be serviced as fast as possible based on the resources of the server pool.

[0068] Accordingly, the network traffic management apparatus 12 can send to the requesting one of the client computing devices 14(1)-14(n) each portion of the requested content in byte range order and as received from the one of the server computing devices 16(1)-16(m) capable of sending a response to each of the requests for the byte range portions in the least amount of time. Thereby, the network traffic management apparatus 12 is able to optimize service

of the content request by reducing the time required for the requesting one of the client computing devices to receive the last byte of the requested content.

[0069] In other examples, a plurality of techniques for optimizing service of the request for content is utilized by the network traffic management apparatus 12. In these examples, the threshold conditions can also be used to determine which technique(s) should be utilized and the various parameters of implementing the technique. Parameters for implementing the optimization techniques can include which of the server computing devices 16(1)-16(n) to utilize, the byte range size to utilize, the level of response buffering, whether multiple requests are to be made for the same content, and the level of redundancy, for example, although other permutation of threshold conditions, methods of determining which technique to apply, and the parameters can also be used.

[0070] By this technology, a network traffic management apparatus generates a plurality of requests to a plurality of server computing devices in response to a request for content received from a client computing device. Thereby, portions of the requested content are retrieved from a plurality of server computing devices instead of the content being retrieved from one server computing device. Accordingly, the risk of a hotspot developing in the server pool due to one or more requests for relatively large content is reduced. Several other advantages include the ability to reduce the time to first byte and/or time to last byte for the client computing device and guaranteeing a maximum or uniform size of requests sent by the network traffic management device to the server computing devices.

[0071] Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only, and is not limiting. Various alterations, improvements, and modifications will occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other

designations therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

CLAIMS

What is claimed is:

1. A method for optimizing service of one or more content requests, the method comprising:

receiving at a network traffic management apparatus a request for content from a client computing device;

determining with the network traffic management apparatus a length of the content;

sending with the network traffic management apparatus a plurality of requests for a portion of the length of the content to a plurality of server computing devices, wherein the portion of the length of the content is specified as a byte range in a range header of each of the plurality of requests;

receiving at the network traffic management apparatus a plurality of responses to the plurality of requests; and

outputting with the network traffic management apparatus at least a subset of the plurality of responses to the client computing device.

2. The method as set forth in claim 1 further comprising:

determining with the network traffic management apparatus the byte range based on a total number of server computing devices in the plurality of server computing devices;

sending with the network traffic management apparatus one of the plurality of requests to each of the plurality of server computing devices; and

outputting with the network traffic management apparatus each of the responses to the plurality of requests to the client computing device in byte range order.

3. The method as set forth in claim 1 wherein the byte range specified in the range header of at least a subset of the plurality of requests is greater than a maximum byte range size, not greater than a network frame payload size, or substantially equal to a uniform byte range size.

4. The method as set forth in claim 1 wherein:

the sending further comprises sending at least one request for a relatively low byte range portion of the content to a relatively high performance one of the plurality of server computing devices and sending at least one request for a relatively high byte range portion of the length of the content to a relatively low performance one of the plurality of server computing devices; and

the outputting further comprises outputting a response to the at least one request for a relatively low byte range portion of the content to the client computing device in byte range order and irrespective of whether a response to the at least one request for a relatively high byte range portion of the content has been received.

5. The method as set forth in claim 1 wherein:

the byte ranges of at least a subset of the plurality of requests at least partially overlap; and

the outputting further comprises outputting at least partially overlapping byte ranges included in each first received response to one of the subset of the plurality of requests.

6. The method as set forth in claim 1 further comprising:

determining with the network traffic management apparatus whether one or more threshold conditions are satisfied, wherein the threshold conditions are selected from whether the requested content is larger than a specified length, whether a file type of the requested content matches one or more specified file types, or whether at least a portion of a path specified in the request for content matches one or more specified paths; and

outputting with the network traffic management apparatus the plurality of requests only when it is determined that the one or more threshold conditions are satisfied.

7. The method as set forth in claim 1 wherein the outputting further comprises outputting the at least a subset of the responses to the plurality

of requests as a single hypertext transfer protocol (HTTP) response and in byte range order.

8. The method as set forth in claim 1 wherein the receiving further comprises:

determining whether the plurality of responses includes all portions of the length of the content; and

requesting at least one portion of the length of the content that has not been included in any of the responses from at least one of the plurality of server computing devices not including one or more of the plurality of server computing devices from which the at least one portion of the length of the content was previously requested, when it is determined that the plurality of responses does not include all portions of the length of the content.

9. A non-transitory computer readable medium having stored thereon instructions for optimizing service of one or more content requests comprising machine executable code which when executed by a processor, causes the processor to perform steps comprising:

receiving a request for content from a client computing device;

determining a length of the content;

sending a plurality of requests for a portion of the length of the content to a plurality of server computing devices, wherein the portion of the length of the content is specified as a byte range in a range header of each of the plurality of requests;

receiving a plurality of responses to the plurality of requests; and

outputting at least a subset of the plurality of responses to the client computing device.

10. The medium as set forth in claim 9 further having stored thereon instructions that when executed by the processor cause the processor to perform steps further comprising:

determining the byte range based on a total number of server computing devices in the plurality of server computing devices;

sending one of the plurality of requests to each of the plurality of server computing devices; and

outputting each of the responses to the plurality of requests to the client computing device in byte range order.

11. The medium as set forth in claim 9 wherein the byte range specified in the range header of at least a subset of the plurality of requests is not greater than a maximum byte range size, not greater than a network frame payload size, or substantially equal to a uniform byte range size.

12. The medium as set forth in claim 9 wherein:

the sending further comprises sending at least one request for a relatively low byte range portion of the content to a relatively high performance one of the plurality of server computing devices and sending at least one request for a relatively high byte range portion of the length of the content to a relatively low performance one of the plurality of server computing devices; and

the outputting further comprises outputting a response to the at least one request for a relatively low byte range portion of the content to the client computing device in byte range order and irrespective of whether a response to the at least one request for a relatively high byte range portion of the content has been received.

13. The medium as set forth in claim 9 wherein:

the byte ranges of at least a subset of the plurality of requests at least partially overlap; and

the outputting further comprises outputting at least partially overlapping byte ranges included in each first received response to one of the subset of the plurality of requests.

14. The medium as set forth in claim 9 further having stored thereon instructions that when executed by the processor cause the processor to perform steps further comprising:

determining whether one or more threshold conditions are satisfied wherein the threshold conditions are selected from whether the requested content is larger than a specified length, whether a file type of the requested content matches one or more specified file types, or whether at least a portion of a path specified in the request for content matches one or more specified paths; and

outputting the plurality of requests only when it is determined that the one or more threshold conditions are satisfied.

15. The medium as set forth in claim 9 wherein the outputting further comprises outputting the at least a subset of the responses to the plurality of requests as a single hypertext transfer protocol (HTTP) response and in byte range order.

16. The medium as set forth in claim 9 wherein the receiving further comprises:

determining whether the plurality of responses includes all portions of the length of the content; and

requesting at least one portion of the length of the content that has not been included in any of the responses from at least one of the plurality of server computing devices not including one or more of the plurality of server computing devices from which the at least one portion of the length of the content was previously requested, when it is determined that the plurality of responses does not include all portions of the length of the content.

17. A network traffic management apparatus, comprising:

at least one of configurable hardware logic configured to implement or a processor or a network interface controller coupled to a memory and configured to execute programmed instructions stored in the memory comprising:

receiving a request for content from a client computing device;

determining a length of the content;

sending a plurality of requests for a portion of the length of the content to a plurality of server computing devices, wherein the portion of the length of the content is specified as a byte range in a range header of each of the plurality of requests;

receiving a plurality of responses to the plurality of requests; and

outputting at least a subset of the plurality of responses to the client computing device.

18. The apparatus as set forth in claim 17 wherein at least one of the configurable hardware logic is further configured to be capable of implementing or the processor or the network interface controller coupled to the memory is further configured to execute programmed instructions stored in the memory further comprising:

determining the byte range based on a total number of server computing devices in the plurality of server computing devices;

sending one of the plurality of requests to each of the plurality of server computing devices; and

outputting each of the responses to the plurality of requests to the client computing device in byte range order.

19. The apparatus as set forth in claim 17 wherein the byte range specified in the range header of at least a subset of the plurality of requests is not greater than a maximum byte range size, not greater than a network frame payload size, or substantially equal to a uniform byte range size.

20. The apparatus as set forth in claim 17 wherein

the sending further comprises sending at least one request for a relatively low byte range portion of the content to a relatively high performance one of the plurality of server computing devices and sending at least

one request for a relatively high byte range portion of the length of the content to a relatively low performance one of the plurality of server computing devices; and

the outputting further comprises outputting a response to the at least one request for a relatively low byte range portion of the content to the client computing device in byte range order and irrespective of whether a response to the at least one request for a relatively high byte range portion of the content has been received.

21. The apparatus as set forth in claim 17 wherein

the byte ranges of at least a subset of the plurality of requests at least partially overlap; and

the outputting further comprises outputting at least partially overlapping byte ranges included in each first received response to one of the subset of the plurality of requests.

22. The apparatus as set forth in claim 17 wherein at least one of the configurable hardware logic is further configured to be capable of implementing or the processor or the network interface controller coupled to the memory is further configured to execute programmed instructions stored in the memory further comprising:

determining whether one or more threshold conditions are satisfied wherein the threshold conditions are selected from whether the requested content is larger than a specified length, whether a file type of the requested content matches one or more specified file types, or whether at least a portion of a path specified in the request for content matches one or more specified paths; and

outputting the plurality of requests only when it is determined that the one or more threshold conditions are satisfied.

23. The apparatus as set forth in claim 17 wherein the outputting further comprises outputting the at least a subset of the responses to the plurality of requests as a single hypertext transfer protocol (HTTP) response and in byte range order.

24. The apparatus as set forth in claim 17 wherein the receiving further comprises:

determining whether the plurality of responses includes all portions of the length of the content; and

requesting at least one portion of the length of the content that has not been included in any of the responses from at least one of the plurality of server computing devices not including one or more of the plurality of server computing devices from which the at least one portion of the length of the content was previously requested, when it is determined that the plurality of responses does not include all portions of the length of the content.

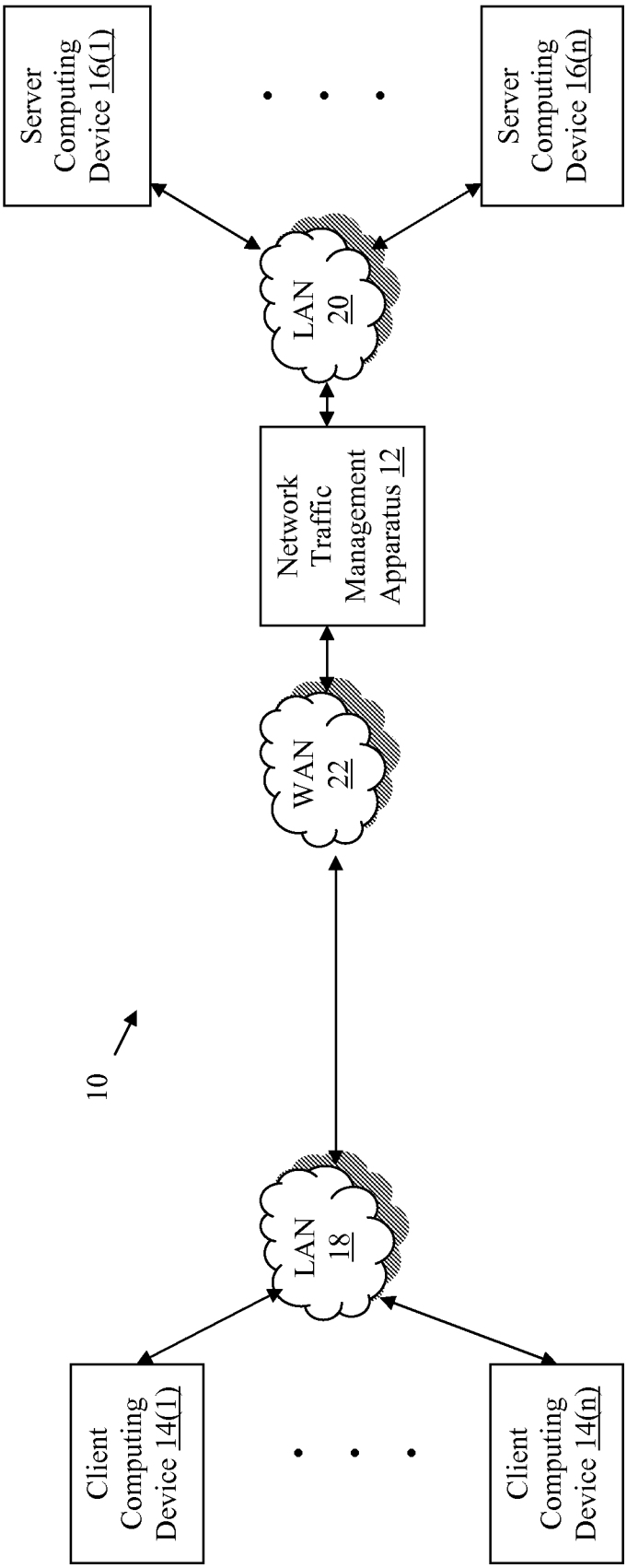
Server Computing Device 16(1)
Server Computing Device 16(n)
LAN 20
Network Traffic Management Apparatus 12
WAN 22
LAN 18
10
Client Computing Device 14(1)
Client Computing Device 14(n)

FIG. 1

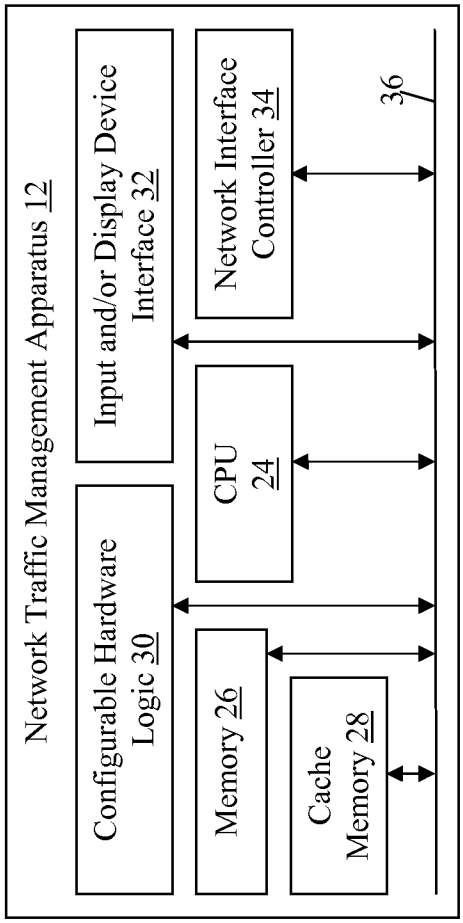
Network Traffic Management Apparatus 12
Configurable Hardware Logic 30
Input and/or Display Device Interface 32
Memory 26
CPU 24
Network Interface Controller 34
Cache Memory 28
36

FIG. 2

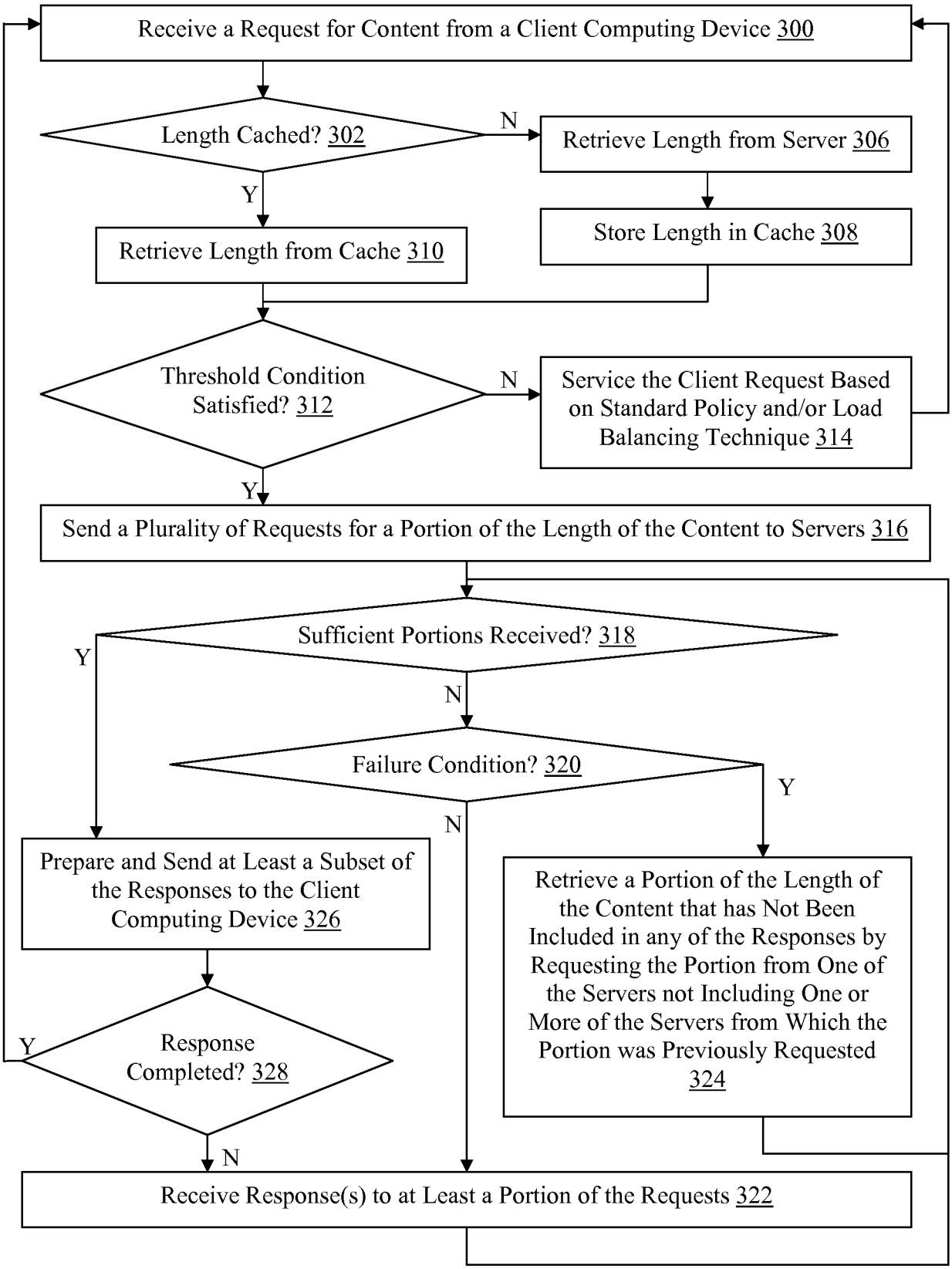
Receive a Request for Content from a Client Computing Device 300
Length Cached? 302
N
Retrieve Length from Server 306
Y
Store Length in Cache 308
Retrieve Length from Cache 310
Threshold Condition Satisfied? 312
N
Service the Client Request Based on Standard Policy and/or Load Balancing Technique 314
Y
Send a Plurality of Requests for a Portion of the Length of the Content to Servers 316
Sufficient Portions Received? 318
Y
N
Failure Condition? 320
Y
N
Prepare and Send at Least a Subset of the Responses to the Client Computing Device 326
Retrieve a Portion of the Length of the Content that has Not Been Included in any of the Responses by Requesting the Portion from One of the Servers not Including One or More of the Servers from Which the Portion was Previously Requested 324
Response Completed? 328
Y
N
Receive Response(s) to at Least a Portion of the Requests 322

FIG. 3