October 2021

SYSTEM AND METHOD FOR PROVIDING SERVICE FOR A SEARCH REQUEST

GIRISH BALAKRISHNA HEGDE
Visa

PRIYADARSHINI SARODE
Visa

RAHUL POWAR
Visa

TITLE: “SYSTEM AND METHOD FOR PROVIDING SERVICE FOR A SEARCH REQUEST”

VISA

GIRISH BALAKRISHNA HEGDE

PRIYADARSHINI SARODE

RAHUL POWAR

TECHNICAL FIELD

The present subject matter relates to field of transmission of digital information, more particularly, but not exclusively to a system and method for providing service for a search request.

BACKGROUND

In current scenario, consider a user enters a search request or a Uniform Resource Locator (URL) on a search engine to obtain results for the search request. The search engine fetches domain of the URL and then subdomain of the URL to find the results for the search request. The URL is a mechanism used by a browser to retrieve any published resource on web. In another words, the URL is an address of a unique resource on the web. In existing methods, time taken to obtain the results for the search request is more. For example, if the user enters a URL: "http://mydomain.visa.com/us/east/keygenerator/oauth/merchantxyz?rid=<UUID>", here "mydomain.visa.com" is the domain. The domain is alternatively called as primary domain. The domain is a string of text that maps to a numeric Internet Protocol (IP) address used to access a website. Further, "us", "east", "keygenerator", "oauth" and "merchantxyz" are the subdomains for the entered URL. The subdomain is an additional part to the main domain. The subdomains are created to organize and navigate to different sections of the website. The search engine for the entered search request will go through six tiers of extraction and query since we have five subdomains to obtain the results for the search request. Thus, the time complexity for existing methods is O(n)*m, where m is the number of subdomains for the entered URL. Therefore, more the number of subdomains, more time is taken to obtain the results for the search request. Also, in the existing method, the domain and the subdomain-based routing is slow and there is no optimal way for splitting the subdomains for routing. As such, a solution is required to reduce the time complexity and obtain the results quickly for the search request.

The information disclosed in this background of the disclosure section is only for enhancement of understanding of the general background of the invention and should not be taken as an acknowledgement or any form of suggestion that this information forms the prior art already known to a person skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, serve to explain the disclosed principles. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of device or system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

Figure 1 illustrates exemplary environment of a system for providing service for a search request, in accordance with some embodiments of the present disclosure;

Figures 2 and 3 illustrate an example showing implementation for providing service for a search request, in accordance with some embodiments of the present disclosure;

Figure 4 illustrates flow diagram showing method for providing service for a search request, in accordance with some embodiments of the present disclosure; and

Figure 5 illustrates a block diagram of an exemplary computer system for implementing embodiments consistent with the present disclosure.

The figures depict embodiments of the disclosure for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the disclosure described herein.

DESCRIPTION OF THE DISCLOSURE

In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all

modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise.

The present disclosure proposes a system and method for providing service for a search request. The proposed system extracts address of the search request entered by a user based on length and hashes the address. Further, the proposed system looks into a hash table and routes the search request to the service to obtain search result upon successful matching of the hash address. The proposed system reduces time complexity and provides search results more quickly for the entered search request.

Figure 1 illustrates an exemplary environment 100 of a system 101 which is configured to provide service for a search request entered by a user 102 on a browser. In an embodiment, the search request may also be referred as an URL. The URL is a mechanism used by the browser to retrieve any published resource on web. In an embodiment, the browser is a software which is used by the user 102 to view or visit different webpages or websites. The environment 100 for providing service for the search request includes the user 102, a communication network 103 and the system 101. For example, consider the user 102 enters a URL “http://mydomain.visa.com/us/east/keygenerator/oauth/merchantxyz?rid=<UUID>” on the browser. The system 101 receives the URL and adds custom header to the URL. In an

embodiment, the custom header comprises priority number and the URL extraction length. In an embodiment, the system 101 is configured to provide priority number 1 for the URL with maximum length. For example, below provided URL is given the priority number 1 and the URL extraction length is 56. "mydomain.visa.com/us/east/keygenerator/oauth/merchantxyz" Further, the system 101 is configured to provide priority number 2 for the URL with second maximum length. For example, below provided URL is given the priority number 2 and the URL extraction length is 44. "mydomain.visa.com/us/east/keygenerator/oauth**"**. Similarly, the system 101 adds the custom header to the URL with the priority number and the URL extraction length until the system 101 reaches the domain. In an embodiment, for the entered URL, the system 101 adds six custom headers as the URL has five subdomains and one main domain. Further, the system 101 is configured to read longest address-size-priority i.e., the URL with the maximum length and the priority number 1 as shown in **Figure 2**. Upon reading, the system 101 is configured to extract the address based on the length and hash the address as shown in **Figure 2**. In an embodiment, hashing is a process of converting the address into another smaller value for O(1) retrieval time. Upon hashing the address, the system is configured to look into a predefined hash table as shown in **Figure 3**. If the hashed address matches with one of the hash values from the predefined hash table, the system 101 is configured to route the URL to service port and provide the service. In an embodiment, the predefined hash table comprises the hash value and service address associated with that hash value. Further, if the hashed address does not match any of the hash values from the predefined hash table, the system 101 is configured to read the address-size-priority with priority number 2 as shown in **Figure 2**. Upon reading, the system 101 extracts the address based on the length and hashes the address with priority number 2 and repeats the process until the system 101 provides the services for the entered search request. In an embodiment, the service provided may be, but limited to, results for the search request, routing to another webpage or website and so on.

In an embodiment, the system 101 and the user 102 may communicate via the communication network 103, for providing the service for the search request entered by the user 102. The communication network 103 may include, without limitation, a direct interconnection, Local Area Network (LAN), Wide Area Network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, and the like. In an embodiment, the system 101 may be implemented in a server configured to provide the service for the search request entered by the user 102. In an embodiment, such server may be a dedicated server or a cloud-based server.

Further, the system 101 may include one or more processor 104, I/O interface 105, and a memory 106. In some embodiments, the memory 106 may be communicatively coupled to the one or more processors 104. The memory 106 stores instructions, executable by the one or more processors 104, which, on execution, may cause the system 101 to provide the service for the search request entered by the user 102, as disclosed in the present disclosure. In an embodiment, the memory 106 may include one or more modules 107 and data 108. The one or more modules 107 may be configured to perform the steps of the present disclosure using the data 108, to provide the service for the search request entered by the user 102. In an embodiment, each of the one or more modules 107 may be a hardware unit which may be present outside the memory 106 and coupled with the system 101. The system 101 may be implemented in a variety of computing systems, such as a laptop computer, a desktop computer, a Personal Computer (PC), a notebook, a smartphone, a tablet, e-book readers, a server, a network server, a cloud-based server and the like. In an embodiment, the system 101 may be a dedicated server or may be a cloud-based server.

Figure 4 illustrates flow diagram showing method for providing service for a search request, in accordance with some embodiments of the present disclosure.

At block 401 of **Figure 4**, the system 101 may be configured to receive the search request or the URL from the user 102. In an embodiment, the URL includes domain and subdomain.

At block 402 of **Figure 4**, upon receiving the search request, the system 101 may be configured to add custom header to the search request or the URL to allow index-based extraction. In an embodiment, the custom header includes the priority number and the URL extraction length.

At block 403 of **Figure 4**, the system 101 is configured to read longest address-size-priority from the search request or the URL i.e., the system reads the URL with the domain and the subdomain having the maximum length and priority number 1.

At block 404 of **Figure 4**, the system 101 is configured to extract the address of the URL based on the length of the search request and hash the extracted address to obtain hash value.

At block 405 of **Figure 4**, the system 101 is configured to look into a predefined hash table. In an embodiment, the predefined hash table includes the hash value and the service address associated with the hash value.

At block 406 of **Figure 4**, the system 101 is configured to check if the hash of the address matches with one of the hash values of the predefined hash table. In an embodiment, if the hashed address matches steps at block 407 is performed. In an embodiment, if the hashed address does not match steps at block 403 is performed again.

At block 407 of **Figure 4**, upon successful matching of the hashed address the system 101 is configured to route the search request or the URL to the service address and provide the service for the search request to the user 102.

Advantages of the present disclosure

Embodiments of the present disclosure discloses a system and method to reduce the time complexity for providing the search result by extracting the address based on the length and hashing the address.

Embodiments of the present disclosure provides optimal matching for the search request by using the predefined hash table which stores the hash value and service address associated with the hash value.

Embodiments of the present disclosure reduces subdomain pattern matching complexity to O(1) by hashing the address of the search request based on the length of the address.

Computing System

Figure 5 illustrates a block diagram of an exemplary computer system 500 for implementing embodiments consistent with the present disclosure. In an embodiment, the computer system 500 is used to implement the system 101 for providing the service for the search request entered by the user 102. The computer system 500 may include a central processing unit (“CPU” or “processor”) 502. The processor 502 may include at least one data processor for executing processes in Virtual Storage Area Network. The processor 502 may include specialized processing units such as, integrated system (bus) controllers, memory management control units, floating point units, graphics processing units, digital signal processing units, etc.

The processor 502 may be disposed in communication with one or more input/output (I/O) devices 509 and 510 via I/O interface 501. The I/O interface 501 may employ communication protocols/methods such as, without limitation, audio, analog, digital, monaural, RCA, stereo, IEEE-1394, serial bus, universal serial bus (USB), infrared, PS/2, BNC, coaxial, component, composite, digital visual interface (DVI), high-definition multimedia interface (HDMI), radio frequency (RF) antennas, S-Video, VGA, IEEE 802.n /b/g/n/x, Bluetooth, cellular (e.g., code-division multiple access (CDMA), high-speed packet access (HSPA+), global system for mobile communications (GSM), long-term evolution (LTE), WiMax, or the like), etc.

Using the I/O interface 501, the computer system 500 may communicate with one or more I/O devices 509 and 510. For example, the input devices 509 may be an antenna, keyboard, mouse, joystick, (infrared) remote control, camera, card reader, fax machine, dongle, biometric reader, microphone, touch screen, touchpad, trackball, stylus, scanner, storage device, transceiver, video device/source, etc. The output devices 510 may be a printer, fax machine, video display (e.g., cathode ray tube (CRT), liquid crystal display (LCD), light-emitting diode (LED), plasma, Plasma Display Panel (PDP), Organic light-emitting diode display (OLED) or the like), audio speaker, etc.

In some embodiments, the computer system 500 may consist of the system 101. The processor 502 may be disposed in communication with a communication network 511 via a network interface 503. The network interface 503 may communicate with the communication network 511. The network interface 503 may employ connection protocols including, without limitation, direct connect, Ethernet (e.g., twisted pair 10/100/1000 Base T), transmission control protocol/internet protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc. The communication network 511 may include, without limitation, a direct interconnection, local area network (LAN), wide area network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, etc. Using the network interface 503 and the communication network 511, the computer system 500 may communicate with a user 512 to provide service for the search request entered by the user 512. The network interface 503 may employ connection protocols include, but not limited to, direct connect, Ethernet (e.g., twisted pair 10/100/1000 Base T), transmission control protocol/internet protocol (TCP/IP), token ring, IEEE 802.11a/b/g/n/x, etc.

The communication network 511 includes, but is not limited to, a direct interconnection, an e-commerce network, a peer to peer (P2P) network, local area network (LAN), wide area network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, Wi-Fi, and such. The first network and the second network may either be a dedicated network or a shared network, which represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP), etc., to communicate with each other. Further, the first network and the second network may include a variety of network devices, including routers, bridges, servers, computing devices, storage devices, etc.

In some embodiments, the processor 502 may be disposed in communication with a memory 505 (e.g., RAM, ROM, etc. not shown in **Figure 5**) via a storage interface 504. The storage interface 504 may connect to memory 505 including, without limitation, memory drives, removable disc drives, etc., employing connection protocols such as, serial advanced technology attachment (SATA), Integrated Drive Electronics (IDE), IEEE-1394, Universal Serial Bus (USB), fibre channel, Small Computer Systems Interface (SCSI), etc. The memory drives may further include a drum, magnetic disc drive, magneto-optical drive, optical drive, Redundant Array of Independent Discs (RAID), solid-state memory devices, solid-state drives, etc.

The memory 505 may store a collection of program or database components, including, without limitation, user interface 506, an operating system 507, web browser 508 etc. In some embodiments, computer system 500 may store user/application data, such as, the data, variables, records, etc., as described in this disclosure. Such databases may be implemented as fault-tolerant, relational, scalable, secure databases such as Oracle ® or Sybase®.

The operating system 507 may facilitate resource management and operation of the computer system 500. Examples of operating systems include, without limitation, APPLE MACINTOSH® OS X, UNIX®, UNIX-like system distributions (E.G., BERKELEY SOFTWARE DISTRIBUTION™ (BSD), FREEBSD™, NETBSD™, OPENBSD™, etc.), LINUX DISTRIBUTIONS™ (E.G., RED HAT™, UBUNTU™, KUBUNTU™, etc.), IBM™ OS/2, MICROSOFT™ WINDOWS™ (XP™, VISTA™/7/8, 10 etc.), APPLE® IOS™, GOOGLE® ANDROID™, BLACKBERRY® OS, or the like.

In some embodiments, the computer system 500 may implement a web browser 508 stored program component. The web browser 508 may be a hypertext viewing application, such as Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Apple Safari, etc. Secure web browsing may be provided using Hypertext Transport Protocol Secure (HTTPS), Secure Sockets Layer (SSL), Transport Layer Security (TLS), etc. Web browsers 508 may utilize facilities such as AJAX, DHTML, Adobe Flash, JavaScript, Java, Application Programming Interfaces (APIs), etc. In some embodiments, the computer system 500 may implement a mail server stored program component. The mail server may be an Internet mail server such as Microsoft Exchange, or the like. The mail server may utilize facilities such as ASP, ActiveX, ANSI C++/C#, Microsoft .NET, Common Gateway Interface (CGI) scripts, Java, JavaScript, PERL, PHP, Python, WebObjects, etc. The mail server may utilize communication protocols such as Internet Message Access Protocol (IMAP), Messaging Application Programming Interface (MAPI), Microsoft Exchange, Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), or the like. In some embodiments, the computer system 500 may implement a mail client stored program component. The mail client may be a mail viewing application, such as Apple Mail, Microsoft Entourage, Microsoft Outlook, Mozilla Thunderbird, etc.

Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer-readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term “computer-readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., be non-transitory. Examples include Random Access Memory (RAM), Read-Only Memory (ROM), volatile memory, non-volatile memory, hard drives, Compact Disc (CD) ROMs, DVDs, flash drives, disks, and any other known physical storage media.
media.

The described operations may be implemented as a method, system or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The described operations may be implemented as code maintained in a “non-transitory computer readable medium”, where a processor may read and

execute the code from the computer readable medium. The processor is at least one of a microprocessor and a processor capable of processing and executing the queries. A non-transitory computer readable medium may include media such as magnetic storage medium (e.g., hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, DVDs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, Flash Memory, firmware, programmable logic, etc.), etc. Further, non-transitory computer-readable media may include all computer-readable media except for a transitory. The code implementing the described operations may further be implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.).

The illustrated steps are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments. Also, the words "comprising," "having," "containing," and "including," and other similar forms are intended to be equivalent in meaning and be open ended in that an item or items following any one of these words is not meant to be an exhaustive listing of such item or items or meant to be limited to only the listed item or items. It must also be noted that as used herein and in the appended claims, the singular forms “a,” “an,” and “the” include plural references unless the context clearly dictates otherwise.

Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term “computer readable medium” should be understood to include tangible items and exclude carrier waves and

transient signals, i.e., are non-transitory. Examples include random access memory (RAM), read-only memory (ROM), volatile memory, non-volatile memory, hard drives, CD ROMs, DVDs, flash drives, disks, and any other known physical storage media.

Finally, the language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the embodiments of the disclosure is intended to be illustrative, but not limiting, of the scope of the disclosure.

With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

SYSTEM AND METHOD FOR PROVIDING SERVICE FOR A SEARCH REQUEST

ABSTRACT

The present disclosure provides a system and a method for providing service for a search request. The proposed system receives the search request entered by a user on a browser. The proposed system adds custom header to the search request to allow index-based extraction of address of the search request. Further, the proposed system extracts the address based on the length and hashes the address. The proposed system looks into a predefined hash table and upon successful matching of the hash address from the hash table, the proposed system routes the search request to a service port and provides the service. Thus, the proposed system reduces time complexity and provides the search result at a faster rate.

Figure 4

1/5

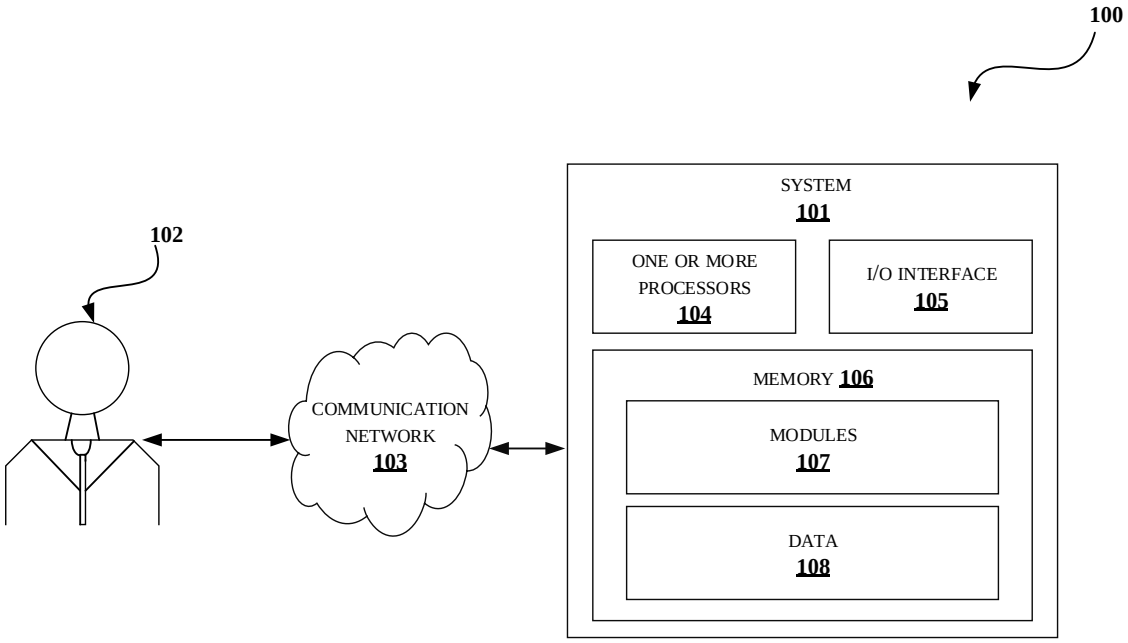
100
102
SYSTEM
101
ONE OR MORE PROCESSORS
104
I/O INTERFACE
105
MEMORY 106
MODULES
107
DATA
108
COMMUNICATION NETWORK
103

Figure 1

2/5

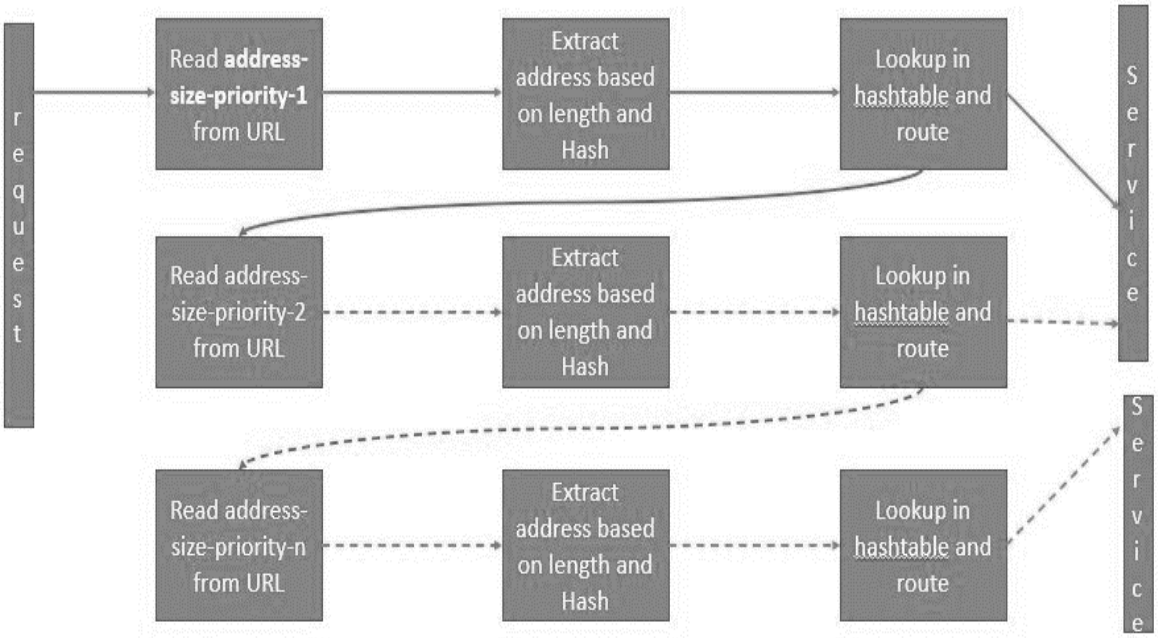

Figure 2

3/5

Header	URL	Hash
address-size-priority-1: 56	mydomain.visa.com/us/east/keygenerator/oauth/merchantxyz	d875aee6
address-size-priority-1: 44	mydomain.visa.com/us/east/keygenerator/oauth	556001d9
address-size-priority-1: 38	mydomain.visa.com/us/east/keygenerator	7147a536
address-size-priority-1: 25	mydomain.visa.com/us/east	4df70e78
address-size-priority-1: 20	mydomain.visa.com/us	947de6e5
address-size-priority-1: 17	mydomain.visa.com	bf3249e5

Hash	Service <IP: Port>
d875aee6	192.158. 1.31 : 8443
556001d9	191.155. 1.42 : 8443
7147a536	193.156. 1.13 : 8443
4df70e78	192.155. 1.34 : 8443
947de6e5	192.156. 1.45 : 8443
bf3249e5	192.155. 1.37 : 8443

Figure 3

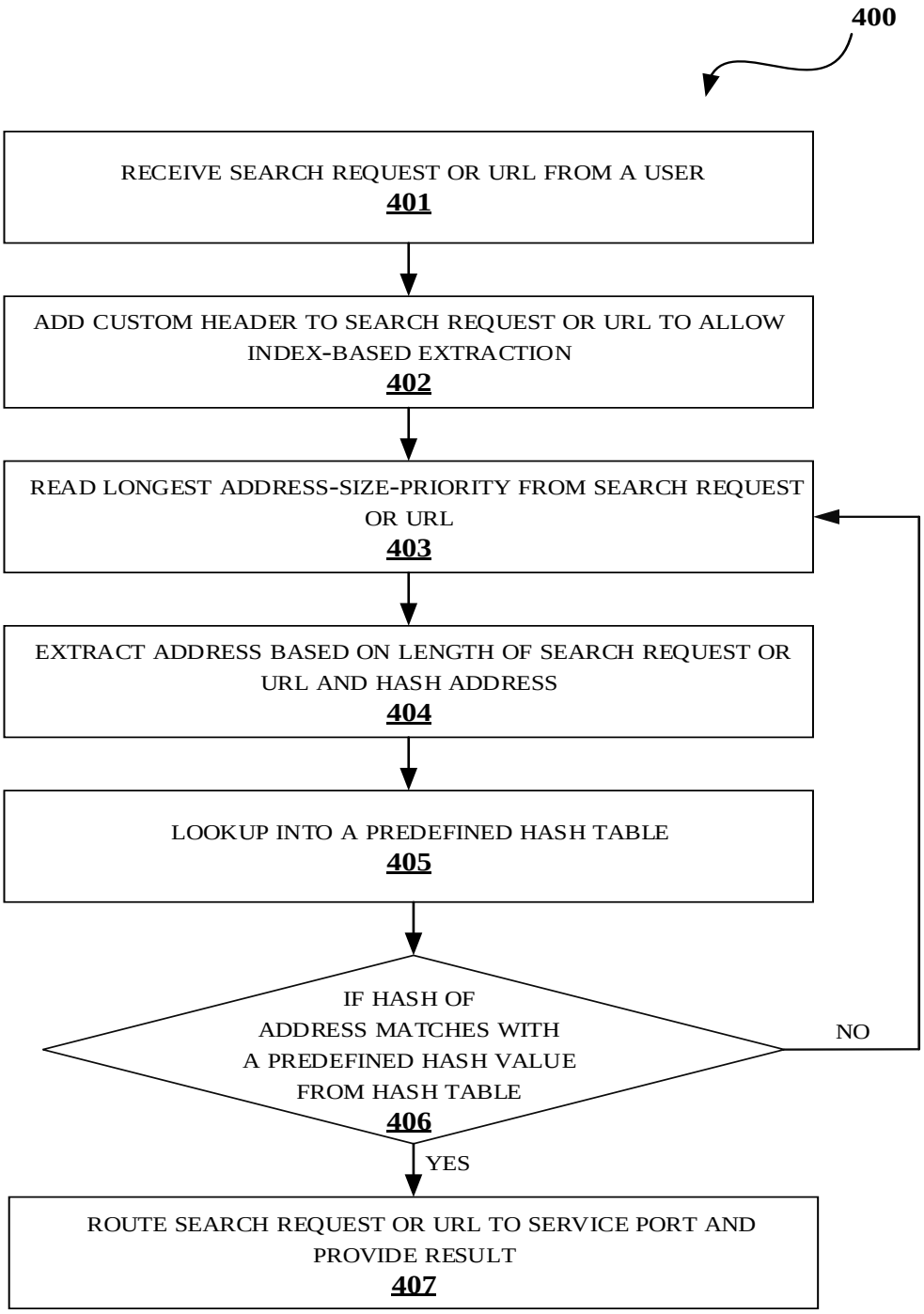
400
RECEIVE SEARCH REQUEST OR URL FROM A USER
401
ADD CUSTOM HEADER TO SEARCH REQUEST OR URL TO ALLOW INDEX-BASED EXTRACTION
402
READ LONGEST ADDRESS-SIZE-PRIORITY FROM SEARCH REQUEST OR URL
403
EXTRACT ADDRESS BASED ON LENGTH OF SEARCH REQUEST OR URL AND HASH ADDRESS
404
LOOKUP INTO A PREDEFINED HASH TABLE
405
IF HASH OF ADDRESS MATCHES WITH A PREDEFINED HASH VALUE FROM HASH TABLE
406
NO
YES
ROUTE SEARCH REQUEST OR URL TO SERVICE PORT AND PROVIDE RESULT
407

Figure 4

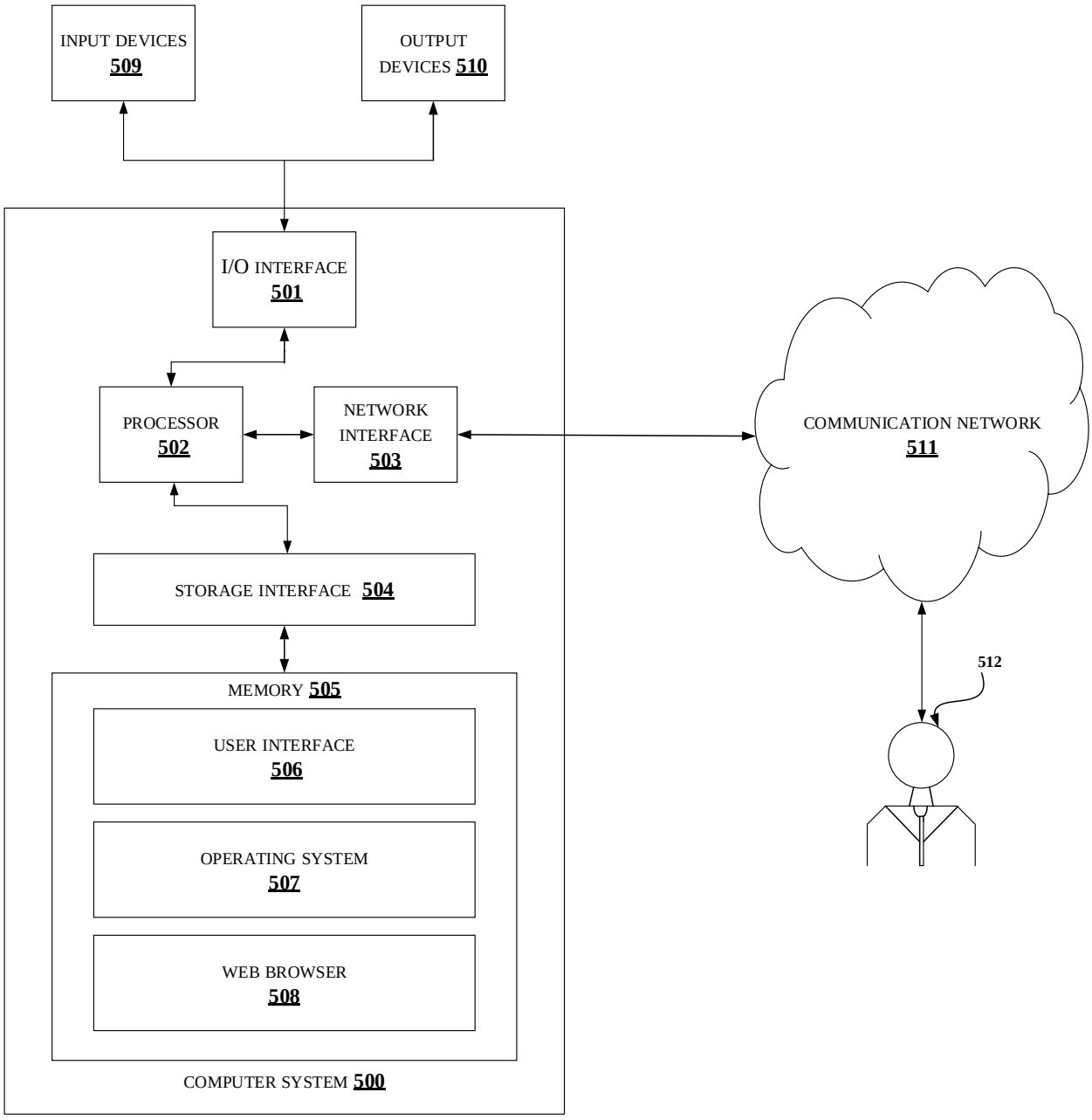
INPUT DEVICES
509
OUTPUT
DEVICES 510
I/O INTERFACE
501
PROCESSOR
502
NETWORK
INTERFACE
503
COMMUNICATION NETWORK
511
STORAGE INTERFACE 504
MEMORY 505
USER INTERFACE
506
OPERATING SYSTEM
507
WEB BROWSER
508
COMPUTER SYSTEM 500
512

Figure 5