



- (51) International Patent Classification:  
*H04L 29/08* (2006.01)
- (21) International Application Number:  
PCT/CN2015/087418
- (22) International Filing Date:  
18 August 2015 (18.08.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
201410411906.2 20 August 2014 (20.08.2014) CN
- (71) Applicant: **HANGZHOU H3C TECHNOLOGIES CO., LTD.** [CN/CN]; 466 Changhe Road, Binjiang District, Hangzhou, Zhejiang 310052 (CN).
- (72) Inventors: **JIN, Xiaoting**; Room 730, Oriental Electronic Bld., No. 2 Chuangye Road, Shangdi Information Industry Base, Haidian District, Beijing 100085 (CN). **WANG, Wei**; Room 730, Oriental Electronic Bld., No. 2 Chuangye Road, Shangdi Information Industry Base, Haidian District, Beijing 100085 (CN).

(74) Agent: **BEIJING BESTIPR INTELLECTUAL PROPERTY LAW CORPORATION**; Room 1209, B Block of Jiahua Building, No. 9 Shangdi 3rd Street, Haidian District, Beijing 100085 (CN).

(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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, 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, SA, 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, ST, 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,

[Continued on next page]

(54) Title: SERVICE PROVISION

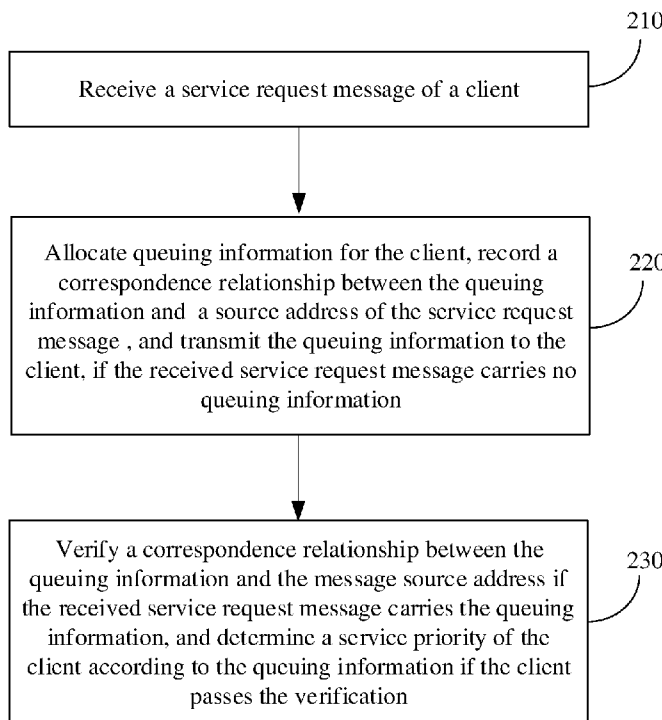


Fig.2

(57) Abstract: There is a service providing method applicable to a server, comprises: receiving a service request message of a client; allocating queuing information for the client, recording a correspondence relationship between the queuing information and a source address of the service request message, and transmitting the queuing information to the client, if the service request message carries no queuing information; and verifying a correspondence relationship between the queuing information and the source address of the service request message if the service request message carries the queuing information, and determining a service priority of the client according to the queuing information if the client passes the verification.



SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG). **Published:**

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

## SERVICE PROVISION

**Background**

5 **[0001]** The emergence of cloud computing has imposed a revolutionary influence upon the development of the information industry. Due to the high-speed transmission capacity of the Internet, Cloud computing enables processing of data from a personal computer or a private server to be transferred to a large cloud computing center, and the computing capacity and the storage capacity to be provided for a user as services, so that the user can purchase and access the computing capacity like a power grid, a water supply and other public services.

10

**Summary****Brief Description of the Drawings**

15 **[0002]** Fig.1 illustrates a schematic structural diagram of a deployed network of a cloud service in an example;

**[0003]** Fig.2 illustrates a flow chart of a service providing method applicable to a server in an example;

**[0004]** Fig.3 illustrates a flow chart of a service providing method applicable to a server on a server in an example;

20 **[0005]** Fig.4 illustrates a flow chart of a method for accessing a service by a client in an example;

**[0006]** Fig.5 illustrates a schematic flow chart of processing by and interaction between a client and a server in an example;

25 **[0007]** Fig.6 illustrates a logic structural diagram of a service providing device applicable to a server in an example;

**[0008]** Fig.7 illustrates a logic structural diagram of a device for accessing a service in an example; and

**[0009]** Fig.8 illustrates a schematic hardware architectural diagram of a service providing device applicable to a server in an example.

### Detailed Description of the Embodiments

**[0010]** Referring to Fig.1, the network illustrated in Fig.1 includes a server 120 providing a service, accessed by a user using a browser on a computer client 111, and by another user using  
5 an application on a handset client 112 in the form of a webpage over the network, for example, if the user submits a browse request or an order, then the server 120 provides a corresponding service in response to the request of the client, for example, by returning a webpage to be accessed by the client or processing the order of the client.

**[0011]** If an enterprise purchases cloud service resources supporting 10, 000 online users  
10 concurrently, then when there are 10, 000 clients accessing concurrently, the server 120 will reject any service to be provided for a client subsequently requesting for an access. The client rejected for the service typically will request again for the service repeatedly, and processing resources of the server 120 will be consumed for rejecting these accesses beyond the capacity, thus degrading the efficiency of processing the accesses of the first 10, 000 clients. Thus too  
15 many concurrent connections will tend to come with a vicious circle in that the processing speed may become lowered with an increasing number of clients requesting for an access.

**[0012]** Referring to Fig.2, a service providing method applicable to a server as illustrated in Fig.2 can be applicable to a server and include the following blocks:

**[0013]** The block 210 is for receiving a service request message of a client;

20 **[0014]** The block 220 is for allocating queuing information for the client, to record a correspondence relationship between the queuing information and a source address of the service request message , and to transmit the queuing information to the client, if the received service request message carries no queuing information; and

**[0015]** The block 230 is for verifying a correspondence relationship between the queuing  
25 information and the message source address if the received service request message carries the queuing information, and to determine a service priority of the client according to the queuing information if the client passes the verification.

**[0016]** Thus in this disclosure, the server can provide the client with the different services according to the received service request message carries the queuing information to thereby

better satisfy a demand of the client and improve the efficiency of providing the client with the service by the server.

**[0017]** Referring to Fig.3 and Fig.4, a client generates a service request message carrying queuing information transmitted by a server if the queuing information is stored locally at block 410.

**[0018]** The client generates the service request message before requesting the server for a service. The client firstly searches locally for queuing information received from the server, and if there is queuing information received from the server, then the client encapsulates the stored queuing information into the service request message to be transmitted to the server, in a predetermined format; otherwise, then the client generates a normal service request message carrying no queuing information. No matter whether the service request message carries the queuing information, the service request message carries an address of the client, e.g., an IP address, so that the service request message will be transmitted normally over a network.

**[0019]** In an example, the queuing information stored locally by the client includes a queuing sequence number transmitted by the server, and the generated service request message carries the queuing sequence number; and in another example, the queuing information stored locally by the client includes a queuing sequence number, and a queuing password corresponding to the queuing sequence number, transmitted by the server, and the generated service request message carries the queuing sequence number and the corresponding queuing password.

**[0020]** The service request message can include different contents in different formats dependent upon a particular protocol, and a particular type of service, applied between the server and the client, although the invention will not be limited to this example.

**[0021]** The client transmits the service request message carrying the queuing information to the server, so that the server verifies the queuing information therein and determines a service priority at block 420.

**[0022]** The server receives the service request message of the client at block 310.

**[0023]** The server allocates queuing information for the client, records a correspondence relationship between the queuing information and a source address of the service request message, and transmits the queuing information to the client, if the service request message

from the client carries no queuing information at block 320.

**[0024]** If the service request message transmitted by the clients carries no queuing information, then the server determines that this is a service request initially initiated by the client, allocates queuing information for the client, and transmits the allocated queuing information to the client. In order to enable the client to be identified correctly in a service request subsequently initiated by the client, the correspondence relationship between the queuing information and the source address of the service request message (e.g., an IP address) to be recorded on the server.

**[0025]** In an example, the server allocates the queuing information including a queuing sequence number for the client. Typically only one queuing sequence number is allocated on the server for the same source address so that respective clients can be provided with a fair queuing environment. Alternatively more than one queuing sequence number can be allocated for the same source address.

**[0026]** The server can allocate queuing sequence numbers sequentially in an ascending order for the respective clients in an order that they initiate requests; or can queue the clients into different queues according to user levels of the clients, types of requested services, etc., and allocate queuing sequence numbers for the clients in the respective queues.

**[0027]** In another example, in order to further enhance security, the queuing sequence number can be configured with a corresponding queuing password, and the queuing information allocated by the server for the client includes the queuing sequence number and the corresponding queuing password. If the service request message transmitted by the client carries no queuing information, then the queuing sequence number is allocated for the client, and the queuing password corresponding to the queuing sequence number, and the source address of the service request message is generated. In this example, the server records a correspondence relationship between the queuing sequence number, the queuing password, and the source address of the service request message, and transmits the queuing sequence number and the corresponding queuing password to the client, so that the client will initiate a service request next time by carrying the queuing sequence number and the queuing password in the service request message. The server can generate the queuing password in a number of ways, for

example, several queuing passwords can be preset and applied alternatively on the server; and in another example, the queuing password can be generated on the server with the queuing sequence number and/or the source address of the service request message being parameters in a predetermined encryption algorithm.

5 **[0028]** In an example, the server can enable a queuing mechanism after the system utilization ratio exceeds a preset threshold, and after the queuing mechanism is enabled, the server will allocate the queuing information for the client and transmit the queuing information to the client, records the correspondence relationship between the queuing information and the source address of the request message, and also provide the service by determining a service order of  
10 the client according to the queuing information. If the system utilization ratio does not exceed the preset threshold, then the server will provide the client with the service instead of allocating the queuing information. Different parameters can be selected as an index to evaluate the system utilization ratio, and corresponding thresholds can be preset, for particular application scenarios; for example, the system utilization ratio can be calculated from the maximum processing  
15 capacity of the system which is the largest number of requests in the SPEC Web (criteria to evaluate the processing capacity of a Web server) series, or which is the number of transactions processed in a unit period of time in the Transaction Processing Performance Council (TPC)-C, or which is the number of users accessing the service to which they subscribe, or the like.

**[0029]** The server can estimate a period of time in which the service request of the client can  
20 be processed, according to the processing speed of the server, and the queuing sequence number of the client, and transmit a possible wait period of time to the client together with the queuing information.

**[0030]** The server can process the service request message of the client by putting it into a queue of tasks waiting for processing as in the prior art.

25 **[0031]** The client can store the queuing information transmitted by the server locally upon reception of the queuing information.

**[0032]** When the service request message transmitted by the client carries no queuing information, the server can return the queuing information to the client. If the client receives the queuing information transmitted by the server, then the client will store the queuing information

locally, so that the client will generate a service request message locally by carrying the queuing information.

**[0033]** The queuing information can include the queuing sequence number or include the queuing sequence number and the queuing password.

5 **[0034]** The client can display the queuing sequence number to the user. If the message returned by the server further carries the wait period of time estimated by the server, then the client can display it to the user together with the queuing sequence number; and the client can also initiate a service request again automatically according to the estimated wait period of time.

10 **[0035]** At block 330, the server verifies the correspondence relationship between the queuing information and the message source address if the service request message from the client carries the queuing information; and determines the service priority of the client according to the queuing information if the client passes the verification.

**[0036]** If the service request message carries the queuing information, then the server extracts the queuing information in the message and the source address of the message, and compares  
15 them with the recorded correspondence relationship between the queuing information and the source address to verify the client transmitting the service request message. If the queuing information and the source address of the message agree with the record, then the client passes the verification, and the client determines the priority at which the client is provided with the service, according to the queuing information.

20 **[0037]** If the client fails to pass the verification, then the server can reject the service or can treat the client as a client initially initiating a service request, reallocate queuing information for the client, record a correspondence relationship between the queuing information, and a source address of the service request message, and transmit the queuing information to the client.

**[0038]** In an example, the service request message transmitted by the client carries the  
25 queuing sequence number. There may be one or more queues of tasks on the server to store service requests waiting for processing, dependent upon how the server allocates the queuing sequence number. The server puts the queuing request of the client into the matching queue of tasks at a corresponding position according to the queuing sequence number so that the client is provided with the service at the priority at which the queuing sequence number is allocated. If



there are a plurality of queues of tasks, then the client can be scheduled among the plurality of queues of tasks in any one of a number of ways, although this disclosure will not be limited to this example.

**[0039]** In another example, the service request message transmitted by the client carries the queuing sequence, and the queuing password corresponding to the queuing sequence number. At this time the server will compare the source address, the queuing sequence number, and the queuing password in the service request message with the recorded correspondence relationship between the source address, the queuing sequence number, and the queuing password, and only if all of them agree, then the client will pass the verification.

**[0040]** The queuing information can become more temporal due to a survival time. For example, the survival time can be preset on the server for the recorded queuing information, and when the survival time expires, the recorded correspondence relationship between the queuing information and the source address of the service request message can be removed. Thus if the service request message carrying the queuing information for which the survival time expires arrives at the server, then the client will fail to pass the verification, and at this time the server can reallocate queuing information for the client. In another example, a survival time can be set on the client for the stored queuing information, or the survival time of the queuing information can be specified by the server while transmitting the queuing information; and when the survival time expires, the client can remove the locally stored queuing information. Thus after the survival time of the queuing information expires, the client will transmit the service request message carrying no queuing information and obtain queuing information again from the server. A survival time can be preset on both the server and the client for the queuing information for the same effect, so a repeated description thereof will be omitted here. If the queuing information is provided with a survival time, then the server can transmit the queuing information to the client by instructing the client to reinitiate a service request before the survival time expires to thereby avoid the client from being queued again.

**[0041]** After the server issues the queuing information, the client can know a task processing condition on the server to thereby avoid a service request from being initiated repeatedly so as to lower the number of tasks to be processed by the server and improve the response speed of

the server.

**[0042]** In an example of this disclosure, the clients 111 and 112 access the service provided by the server 120 in the form of a webpage in the network illustrated in Fig.1. A secret key is preset on the server 120 to encrypt the queuing sequence number: Key1. An interaction flow between the client 111 and the server 120 is as illustrated in Fig.5.

**[0043]** 1) The server 120 detects its system utilization ratio, and if the system utilization ratio does not reach 90%, then the server 120 responds to the service request of the client, and provides the client with the corresponding service.

**[0044]** 2) The server 120 starts the queuing mechanism upon detecting that the system utilization ratio reaches 90%.

**[0045]** 3) The client 111 accesses the service provided by the server 120 through the browser by transmitting the service request message to the server 120. The client 120 is initially accessed by the client 111, so the client 111 has not stored locally any queuing information, and the service request message transmitted by the client 111 carries no queuing information.

**[0046]** 4) The server 120 receives the service request message of the client 111, and puts the service request of the client 111 into the local queue of tasks to wait for processing. The server 120 extracts the IP address in the message, and allocates the unique queuing sequence number with the survival time for the client. The server 120 allocates the queuing sequence numbers in an ascending order according to the sequential order that the service request messages arrive at the server, so that only one queuing sequence number is allocated for each IP address.

**[0047]** For example, the server 120 allocates the queuing sequence number of 30 for the client 111, and the queuing sequence number of 31 for the subsequently arriving client 112.

**[0048]** 5) The server 120 executes the encryption algorithm with the queuing sequence number of 30 and the Key1 being parameters to derive the unique queuing password of Code-111 corresponding to the queuing sequence number.

**[0049]** 6) The server 120 stores the correspondence relationship between the IP address of IP-111 of the client 111, the queuing sequence number 30, and the queuing password of Code-111.

**[0050]** 7) The server 120 returns the queuing sequence number of 30 with the survival time,

the queuing password of Code-111, and the estimated wait period of time to the client 111.

**[0051]** 8) The client 111 stores locally the queuing sequence number 30, and the queuing password of Code-111, transmitted by the server 120, for example, in cookie. The client 111 displays the queuing sequence number of 30 and the estimated wait period of time to the user, and alerts the user that the service requested by the user is waiting for processing.

**[0052]** 9) The client 111 requests again for the service after the estimated wait period of time elapses, and since the queuing information is stored locally, the client 111 generates a service request message by encapsulating the queuing sequence number of 30 and the queuing password of Code-111 into the message.

**[0053]** 10) The client 111 transmits the service request message carrying the queuing sequence number 30 and the queuing password of Code-111 to the server 120.

**[0054]** 11) The server 120 extracts the source IP address, the queuing sequence number of 30, and the queuing password of Code-11 from the received service request message, and searches the stored correspondence relationships for the matching record to verify the client 111. The server 120 locates the matching record, so the client 111 passes the verification.

**[0055]** 12) The server 120 puts the service request of the client 111 into the queue of tasks at the position corresponding to the queuing sequence number 30 to thereby queue it in sequence.

**[0056]** 13) The survival time of the queuing sequence number 30 expires on the client 111, and the client 111 removes the locally stored correspondence relationship between the queuing sequence number 30 and the queuing password of code-111.

**[0057]** 14) When the survival time of the queuing sequence number 30 expires on the server 120, the server 120 removes the locally stored correspondence relationship between the IP address of IP-111, the queuing sequence number 30, and the queuing password of Code-111.

**[0058]** 15) The server 120 detects that the system utilization ratio drops below 90%, and exits the queuing mechanism after the queue of tasks is emptied. The server 120 will not allocate any queuing information for the client requesting for the service after exits the queuing mechanism.

**[0059]** In this example, the server allocates a unique queuing sequence number for the IP address of the client, and the queuing password corresponding thereto, and notifies the user of the estimated wait period of time, so that the client will not frequently refresh the service

request to thereby alleviate a burden on the server so as to guarantee a response speed to the client connected with the server; and moreover the user provided with the feedback information of the service being queued can anticipate a service period of time to thereby improve the experience of the user.

- 5 **[0060]** Fig.6 illustrates a service providing device applicable to a server in an example of this disclosure, applicable to a server, which includes a request message receiving unit 610, a queuing information generating unit 620, and a queuing information processing unit 630, where:
- [0061]** The request message receiving unit 610 is configured to receive a service request message of a client;
- 10 **[0062]** The queuing information generating unit 620 is configured to allocate queuing information for the client, to record a correspondence relationship between the queuing information and a source address of the service request message , and to transmit the queuing information to the client, if the service request message carries no queuing information; and
- [0063]** The queuing information processing unit 630 is configured to verify a correspondence relationship between the queuing information and the message source address if the received
- 15 service request message carries the queuing information, and to determine a service priority of the client according to the queuing information if the client passes the verification.
- [0064]** In an example, the queuing information generating unit 620 includes a queuing mechanism executing module configured to allocate the queuing information for the client when
- 20 a system utilization ratio of the server exceeds a preset threshold, and the service request message carries no queuing information.
- [0065]** The device can further include a verification failure handling unit configured to reallocate queuing information for the client, to record a correspondence relationship between the queuing information and the message source address, and to transmit the queuing
- 25 information to the client, when the client fails to pass the verification.
- [0066]** In an example, there is a survival time of the queuing information; and in this example, the device further includes a survival time processing unit configured to remove the recorded correspondence relationship between the queuing information and the message source address when the survival time of the queuing information expires.

**[0067]** The queuing information comprises a queuing sequence number; or the queuing information comprises a queuing sequence number and a queuing password.

**[0068]** Fig.7 illustrates a device for accessing a service in an example of this disclosure, applicable to a client, which includes a request message generating unit 710 and a request message transmitting unit 720, where:

**[0069]** The request message generating unit 710 is configured to generate a service request message carrying queuing information transmitted by a server if the queuing information is stored locally; and

**[0070]** The request message transmitting unit 720 is configured to transmit the service request message carrying the queuing information to the server, so that the server verifies the queuing information therein and determines a service priority.

**[0071]** In an example, there is a survival time of the queuing information; and in this example, the device further includes a survival time processing unit configured to remove the stored queuing information when the survival time of the queuing information expires.

**[0072]** The queuing information comprises a queuing sequence number; or the queuing information comprises a queuing sequence number and a queuing password.

**[0073]** Fig.8 illustrates a schematic hardware architectural diagram of a service providing device applicable to a server in an example, which can be applicable to a server.

**[0074]** In this example, the device for providing a service includes a processor such as a CPU 81 and a non-transitory storage medium 82. The non-transitory storage medium 82 may for example be a memory, hard disk or other storage device. The processor 81 and the non-transitory storage medium 82 are communicatively connected with each other by an internal bus 84. The device for providing a service according to this disclosure may include further hardware components, e.g., a communication interface 83 via which data is received or transmitted, etc.

**[0075]** As shown in Fig.8, the non-transitory storage medium 82 may store service providing logic 85 in the form of machine readable instructions, which are executable by the processor 81. The service providing logic 85 may for example correspond to the request message receiving unit 610, a queuing information generating unit 620, and a queuing information processing unit

630 in Figure 6. In one example, the processor 81 reads the instructions corresponding to the service providing logic 85 stored in the non-transitory storage medium 82 and executes the instructions to perform the operations of:

**[0076]** Receiving unit a service request message of a client;

5 **[0077]** Allocating queuing information for the client, recording a correspondence relationship between the queuing information and a source address of the service request message , and transmitting the queuing information to the client, if the service request message carries no queuing information; and

10 **[0078]** Verifying a correspondence relationship between the queuing information and the message source address if the received service request message carries the queuing information, and determining a service priority of the client according to the queuing information if the client passes the verification.

15 **[0079]** In an example, the processor 81 reads the instructions corresponding to the service providing logic 85 stored in the storage medium 82 to allocate the queuing information for the client by:

**[0080]** Allocating the queuing information for the client when a system utilization ratio of the server exceeds a preset threshold, and the service request message carries no queuing information.

20 **[0081]** In an example, the processor 81 reads the machine readable instructions corresponding to the service providing logic 85 stored in the storage medium 82 to further reallocate queuing information for the client, to record a correspondence relationship between the queuing information and the message source address, and to transmit the queuing information to the client, when the client fails to pass the verification.

25 **[0082]** In an example, when there is a survival time of the queuing information, the processor 81 reads the machine readable instructions corresponding to the service providing logic 85 stored in the storage medium 82 to further:

**[0083]** To remove the recorded correspondence relationship between the queuing information and the message source address when the survival time of the queuing information expires.

**[0084]** In an example, the processor 81 reads the machine readable instructions corresponding

to the service providing logic 85 stored in the storage medium 82, where the queuing information comprises a queuing sequence number; or the queuing information comprises a queuing sequence number and a queuing password.

**[0085]** The foregoing disclosure is merely illustrative of preferred embodiments of the disclosure but are not intended to limit the disclosure, and any modifications, equivalent  
5 substitutions, adaptations, thereof made without departing from the spirit and scope of the disclosure shall be encompassed in the claimed scope of the appended claims.

## CLAIMS

1. A service providing method applicable to a server, comprises:

receiving a service request message of a client;

5 allocating queuing information for the client, recording a correspondence relationship between the queuing information and a source address of the service request message, and transmitting the queuing information to the client, if the service request message carries no queuing information; and

10 verifying a correspondence relationship between the queuing information and the source address of the service request message if the service request message carries the queuing information, and determining a service priority of the client according to the queuing information if the client passes the verification.

2. The method according to claim 1, wherein said allocating queuing information for a client comprises:

15 allocating the queuing information for the client if a system utilization ratio of the server exceeds a preset threshold and the service request message carries no queuing information.

3. The method according to claim 1, further comprises:

20 reallocating queuing information for the client, recording a correspondence relationship between the queuing information and the source address of the service request message, and transmitting the queuing information to the client, if the client fails to pass the verification.

4. The method according to claim 1, wherein

there is a survival time of the queuing information; and

25 the method further comprises removing the recorded correspondence relationship between the queuing information and the source address of the service request message when the survival time of the queuing information expires.

5. The method according to any one of claims 1 to 4, wherein



the queuing information comprises a queuing sequence number; or

the queuing information comprises a queuing sequence number and a queuing password.

6. A service providing device, comprising a processor and a non-transitory storage medium storing machine readable instructions executable by the processor to perform the operations of:

5 receiving a service request message of a client;

allocating queuing information for the client, recording a correspondence relationship between the queuing information and a source address of the service request message, and transmitting the queuing information to the client, if the service request message carries no queuing information; and

10 verifying a correspondence relationship between the queuing information and the source address of the service request message if the service request message carries the queuing information, and determining a service priority of the client according to the queuing information if the client passes the verification.

7. The device according to claim 6, wherein the instructions to allocate the queuing information  
15 for the client include instructions to:

allocate the queuing information for the client if a system utilization ratio of the server exceeds a preset threshold and the service request message carries no queuing information

8. The device according to claim 6, wherein instructions further includes instructions to:

20 reallocate queuing information for the client, record a correspondence relationship between the queuing information and the source address of the service request message, and transmit the queuing information to the client, if the client fails to pass the verification.

9. The device according to claim 6, wherein when there is a survival time of the queuing information, the instructions includes instructions to:

25 remove the recorded correspondence relationship between the queuing information and the source address of the service request message when the survival time of the queuing information expires.

10. The device according to claims 6, wherein

the queuing information comprises a queuing sequence number; or

the queuing information comprises a queuing sequence number and a queuing password.

5 11. A non-transitory storage medium storing machine readable instructions which are executable by a processor to:

receive a service request message of a client;

10 allocate queuing information for the client, record a correspondence relationship between the queuing information and a source address of the service request message, and transmit the queuing information to the client, if the service request message carries no queuing information; and

15 verify a correspondence relationship between the queuing information and the source address of the service request message if the service request message carries the queuing information, and determine a service priority of the client according to the queuing information if the client passes the verification.

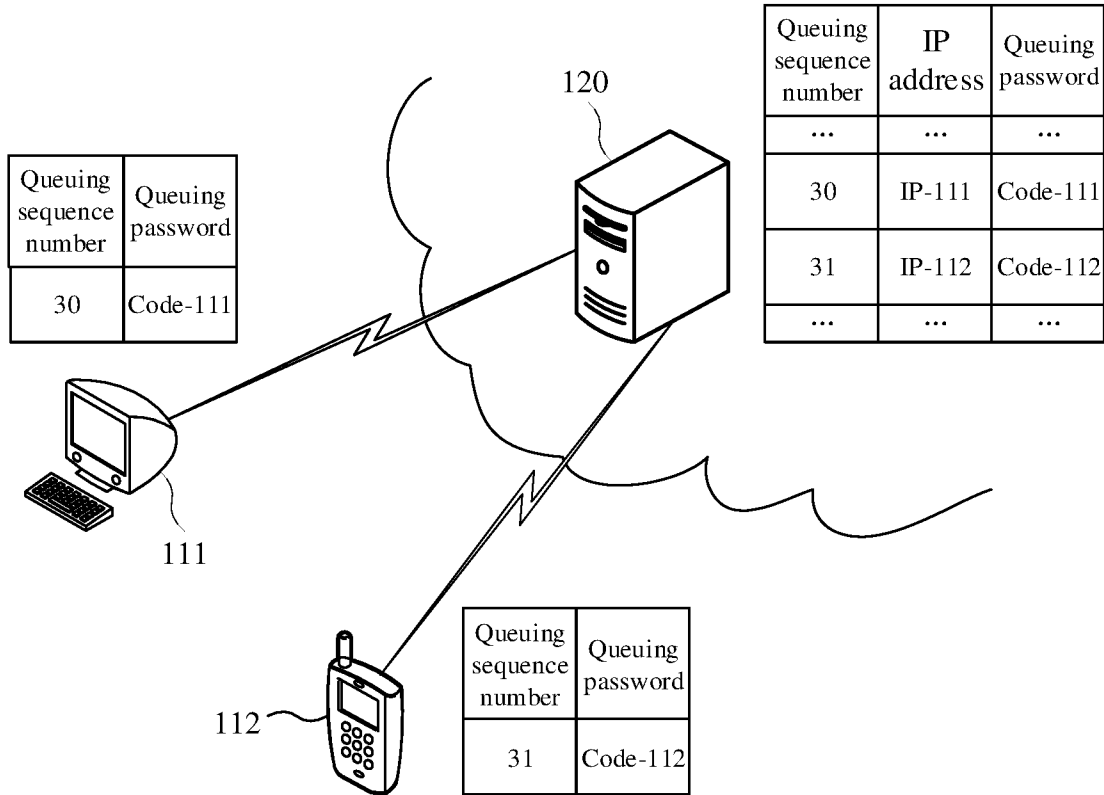


Fig.1

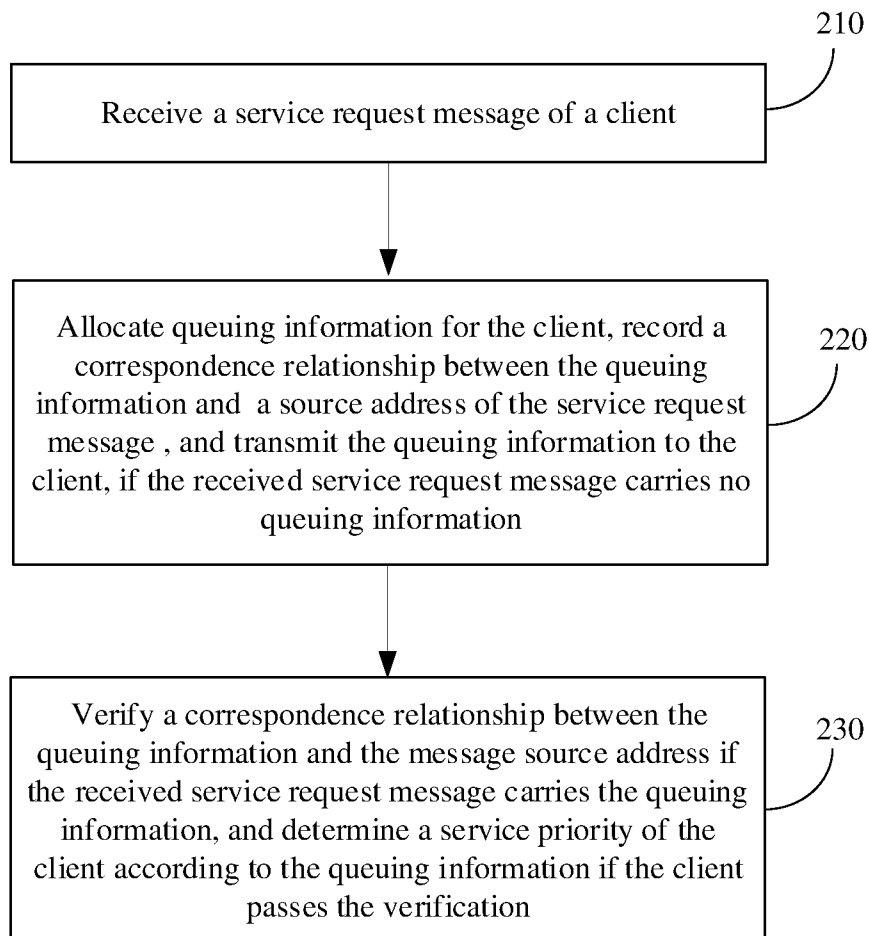


Fig.2

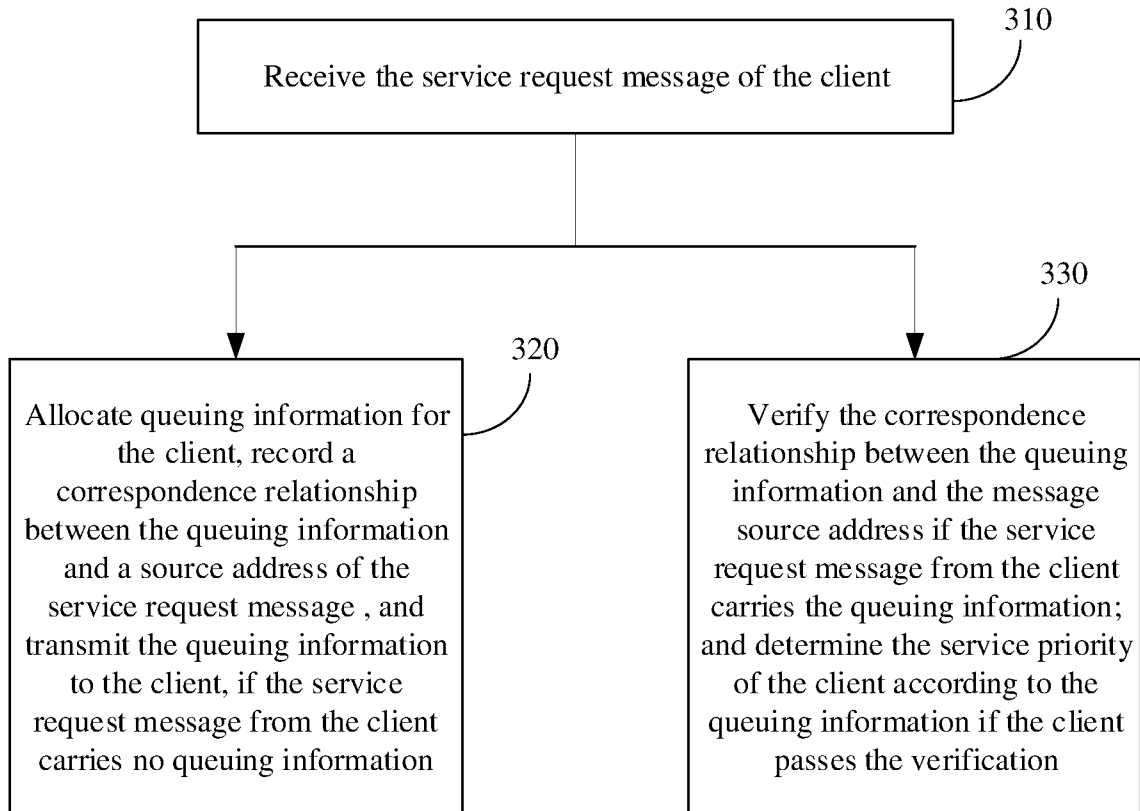


Fig.3

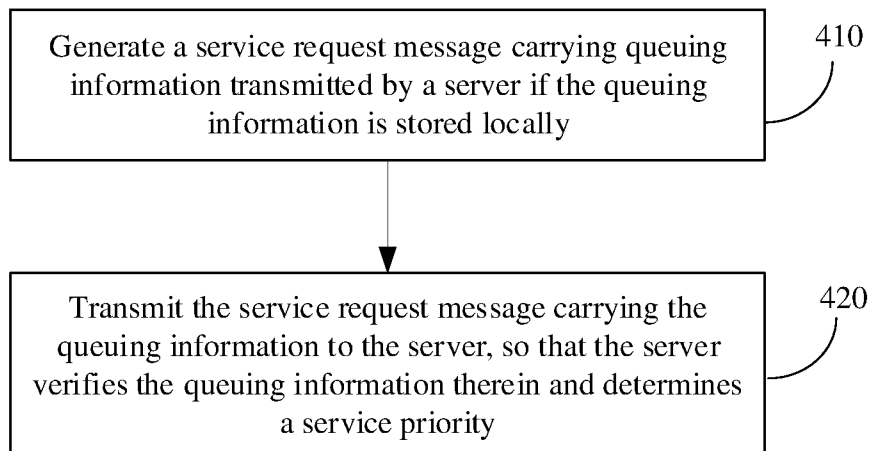


Fig.4

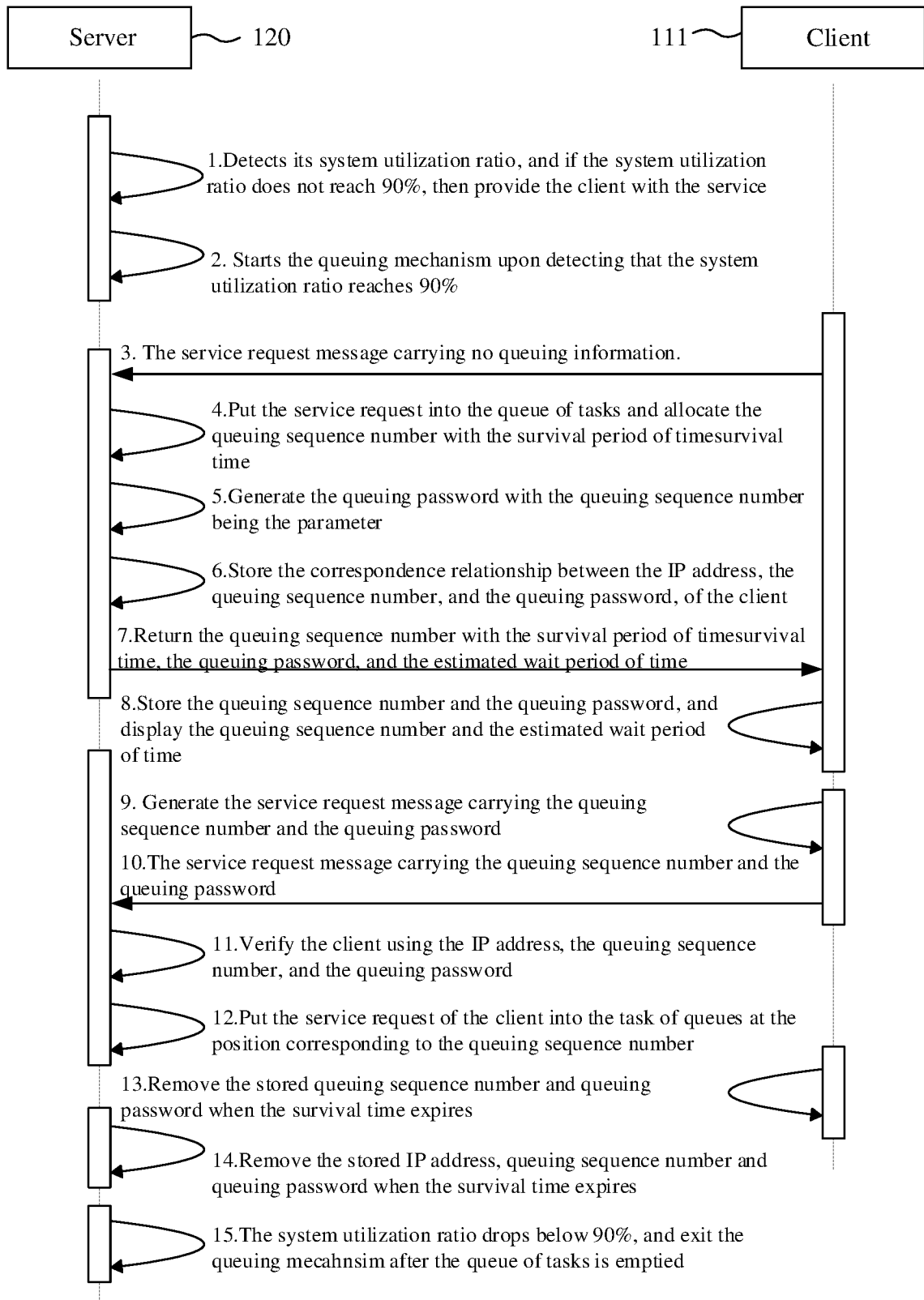


Fig.5

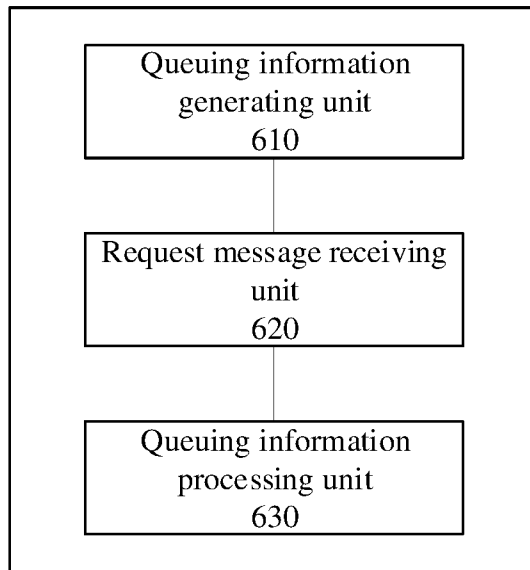


Fig.6

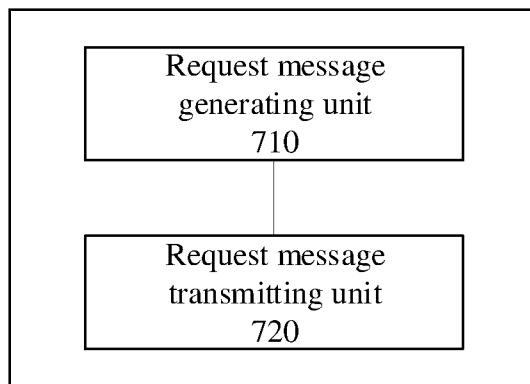


Fig.7

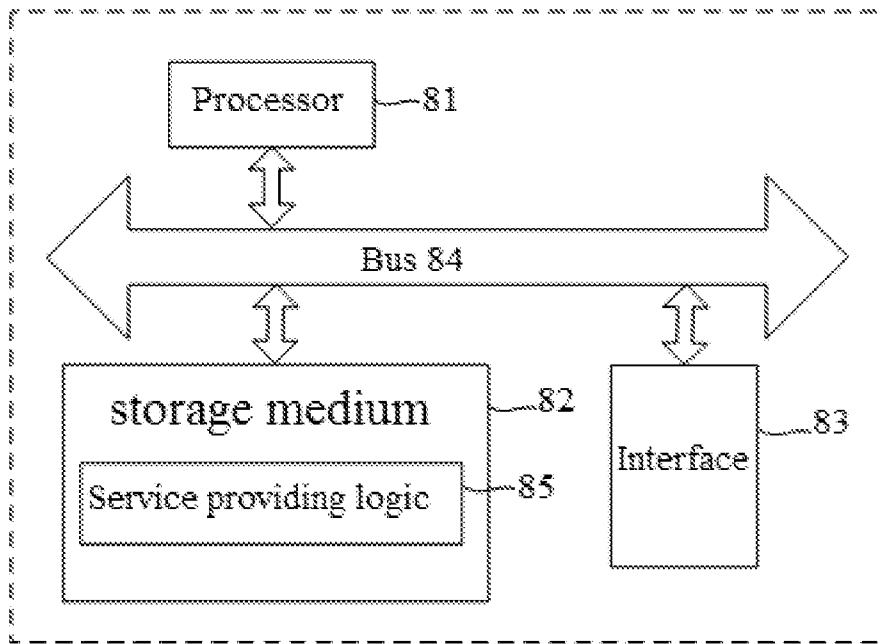


Fig.8

## INTERNATIONAL SEARCH REPORT

International application No.

**PCT/CN2015/087418**

<b>A. CLASSIFICATION OF SUBJECT MATTER</b>		
H04L 29/08(2006.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b>		
Minimum documentation searched (classification system followed by classification symbols)		
H04L; H04M; G06F; H04W; H04Q; G07C		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
CNPAT, WPI, EPODOC, CNKI: queuing, carr+, allocat+, record, assign+, service, operat+, request, informat+, number, address, IP, correspond+, binding		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 102833257 A (BEIJING XIAOMI TECHNOLOGY CO., LTD.) 19 December 2012 (2012-12-19) description, paragraphs [0033]-[0074]	1-11
A	US 6724885 B1 (LUCENT TECHNOLOGIES INC.) 20 April 2004 (2004-04-20) the whole document	1-11
A	CN 102946350 A (HANGZHOU H3C TECHNOLOGIES CO., LTD.) 27 February 2013 (2013-02-27) the whole document	1-11
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents:		
“A”	document defining the general state of the art which is not considered to be of particular relevance	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“E”	earlier application or patent but published on or after the international filing date	“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“L”	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“O”	document referring to an oral disclosure, use, exhibition or other means	“&” document member of the same patent family
“P”	document published prior to the international filing date but later than the priority date claimed	
Date of the actual completion of the international search		Date of mailing of the international search report
04 November 2015		24 November 2015
Name and mailing address of the ISA/CN		Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA 6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China		GUO,Haibo
Facsimile No. (86-10)62019451		Telephone No. (86-10)61648258



**INTERNATIONAL SEARCH REPORT**  
**Information on patent family members**

International application No.

**PCT/CN2015/087418**

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 102833257 A	19 December 2012	None	
US 6724885 B1	20 April 2004	None	
CN 102946350 A	27 February 2013	None	