SHORT LINK PROCESSING METHOD, DEVICE, AND SERVER

TECHNICAL FIELD

[0001] The present disclosure relates to the field of Internet technologies, and in particular, to a short link processing method, apparatus, and server.

BACKGROUND

[0002] A short link is a short URL string that is obtained by converting a long uniform resource locator (URL) through program calculation, etc., so as to save character space.

[0003] In the existing technology, a plurality of servers are generally needed to support a large number of short link applications and access. In addition, the number of characters of a simplified short link needs to be increased to support more records.

[0004] However, if a plurality of short link applications are made for the same long URL, a plurality of short link records would be generated, which wastes storage space. If short link records are stored in a single database and the number of characters of a simplified short link is increased to support a larger storage size, the efficiency of access to the single database is limited.

SUMMARY

[0005] The present disclosure provides a short link processing method, apparatus, and server, to resolve the following problems in the existing technology: If a plurality of short link applications are made for the same long URL, a plurality of short link records are generated, which wastes storage space. If short link records are stored in a single database and the number of characters of a simplified short link is increased to support a larger storage size, the efficiency of access to the single database is limited.

[0006] According to a first aspect of implementations of the present disclosure, a short link processing method is provided, where the method is applied to any server in

a distributed server cluster and includes: receiving a short link application request from a requester, where the application request includes a first long link URL; calculating a corresponding first database table identifier based on the first long link URL; determining whether a first database corresponding to the first database table identifier falls within an access range of the server; and if it is determined that the first database falls within the access range of the server, obtaining a corresponding first short link URL from the first database, and sending the first short link URL to the requester.

[0007] According to a second aspect of the implementations of the present disclosure, a short link processing apparatus is provided, where the apparatus includes: a first receiving unit, configured to receive a short link application request from a requester, where the application request includes a first long link URL; a first calculation unit, configured to obtain a corresponding first database table identifier based on the first long link URL; a first determining unit, configured to determine whether a first database corresponding to the first database table identifier falls within an access range of a server; and a first processing unit, configured to obtain a corresponding first short link URL from the first database and send the first short link URL to the requester if it is determined that the first database falls within the access range of the server.

[0008] According to a third aspect of the implementations of the present disclosure, a server is provided, where the server includes: a processor, and a memory configured to store an instruction that can be executed by the processor, where the processor is configured to receive a short link application request from a requester, where the application request includes a first long link URL; calculate a corresponding first database table identifier based on the first long link URL; determine whether a first database corresponding to the first database table identifier falls within an access range of the server; and obtain a corresponding first short link URL from the first database, and send the first short link URL to the requester, if it is determined that the first database falls within the access range of the server.

[0009] By using the implementations of the present disclosure, the short link application request is received from the requester, and the application request includes the first long link URL. The corresponding first database table identifier is calculated based on the first long link URL, which is used to determine whether the first database corresponding to the first database table identifier falls within the access

range of the server. If it is determined that the first database falls within the access range of the server, the corresponding first short link URL is obtained from the first database, and the first short link URL is sent to the requester. As such, in a database and table sharding mode, the server converts a long link URL to a corresponding short link URL. In addition, because the corresponding short link URL is unique and short, horizontal scalability of short link storage space is improved, and a short link processing speed is further increased.

BRIEF DESCRIPTION OF DRAWINGS

[0010] FIG. 1 is a schematic diagram illustrating a short link processing scenario according to an implementation of the present disclosure;

[0011] FIG. 2 is a flowchart illustrating an implementation of a short link processing method according to the present disclosure;

[0012] FIG. 3 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure;

[0013] FIG. 4 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure;

[0014] FIG. 5 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure;

[0015] FIG. 6 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure;

[0016] FIG. 7 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure;

[0017] FIG. 8 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure;

[0018] FIG. 9 is a structural diagram of hardware of a device where a short link processing apparatus is located according to the present disclosure;

[0019] FIG. 10 is a block diagram illustrating an implementation of a short link processing apparatus according to the present disclosure;

[0020] FIG. 11 is a block diagram illustrating another implementation of a short link processing apparatus according to the present disclosure; and

[0021] FIG. 12 is a block diagram illustrating another implementation of a short link processing apparatus according to the present disclosure.

DESCRIPTION OF EMBODIMENTS

[0022] Example implementations are described in detail here, and examples of the implementations are presented in the accompanying drawings. When the following description relates to the accompanying drawings, the same numbers in different accompanying drawings represent the same element or similar elements unless specified otherwise. Implementations described in the following example implementations do not represent all implementations consistent with the present disclosure. On the contrary, they are only examples of apparatus and methods that are described in the appended claims in detail and that are consistent with some aspects of the present disclosure.

[0023] The terms used in the present disclosure are merely used for illustrating implementations, and are not intended to limit the present disclosure. The terms "a", "said", and "the" of singular forms used in the present disclosure and the appended claims are also intended to include plural forms, unless otherwise specified in the context clearly. It should also be understood that, the term "and/or" used here indicates and includes any or all possible combinations of one or more associated listed items.

[0024] It should be understood that although the terms "first", "second", "third", etc. can be used in the present disclosure to describe various information, the information is not limited by the terms. These terms are only used to differentiate information of the same type. For example, without departing from the scope of the present disclosure, first information can also be referred to as second information, and similarly the second information can also be referred to as the first information. Depending on the context, for example, words "if" used here can be explained as "while" or "when" or "in response to determining".

[0025] Referring to FIG. 1, FIG. 1 is a schematic diagram illustrating a short link processing scenario according to an implementation of the present disclosure.

[0026] The scenario in FIG. 1 includes a requester and a distributed server cluster. The requester in the scenario in FIG. 1 can initiate a short link application request or a short link access request. In addition, the requester can be a system end device, a terminal device of a common user, a device of a partner, etc. Terminal devices of the common user can include various smart phones, tablet computers, personal computers (PCs), etc.

[0027] Any server in the distributed server cluster in the scenario in FIG. 1 can receive a short link application request or a short link access request from the requester.

[0028] In the present disclosure, a distributed server forwarding mechanism in a database and table sharding mode is used. Details are described below.

[0029] Only a limited database that can be accessed by each server is allocated to the server. If a short link application request or a short link access request that is received by a server does not fall with a range of a database that can be accessed by the server, the server forwards the application request or the access request to a server that can access the corresponding database. One or more databases can be allocated to each server.

[0030] The requester can send a short link application request to any server in the distributed server cluster. The application request includes a long URL, and the long URL can be an actual URL to be converted. The purpose of the application request is to enable a short link server to return a corresponding short link URL.

[0031] Alternatively, the requester can send a short link access request to any server in the distributed server cluster. The access request includes a short link URL. The purpose of the access request is to enable the server to obtain a corresponding long URL based on the short link URL, and jump to the long URL. For example, the short link URL is https://m.alipay.com/t9Yv3MZ.

[0032] The corresponding long URL obtained by the server based on the short link URL can be: https://cmspromo.alipay.com/finance/fullyear.htm. For ease of description, the "long URL" is defined as a "long link URL" in this implementation of the present disclosure, and the definition is applicable to all the following implementations of the present disclosure. The following describes the implementations of the present disclosure in detail.

[0033] Referring to FIG. 2, FIG. 2 is a flowchart illustrating an implementation of a short link processing method according to the present disclosure. The method can be applied to any server in a distributed server cluster, and includes the steps below.

[0034] Step 210: Receive a short link application request from a requester, where the application request includes a first long link URL.

[0035] In this implementation of the present disclosure, the server can receive the short link application request from the requester. The purpose of the application request is to enable the requester to obtain a short link URL of the first long link URL.

In addition, the application request received by the server can include a plurality of short link application request parameters.

[0036] For example, the short link application request includes an actual URL to be converted (the first long link URL) and a validity period of the short link, as shown in Table 1.

Table 1

Parameter name	**Parameter description**	**Example**
targetUrl	Actual URL to be converted	https://cmspromo.alipay.com/finance/fullyear.htm
validDate	Validity period of the short link	30 days

[0037] Step 220: Calculate a corresponding first database table identifier based on the first long link URL.

[0038] In this implementation of the present disclosure, a database table identifier is used to represent a unique identifier of different physical tables corresponding to the same logical table in a database. For example, 00 represents a short link table 00, 01 represents a short link table 01, and so on.

[0039] The database table identifier can be a number ranging from 00 to 99 (100 numbers in total, which represent 100 sub-tables) from 00 to 99 or can be one or more characters, and is used to represent a sub-table name of a short link table.

[0040] The server can perform corresponding calculation on the first long link URL by using a predetermined algorithm, to obtain the corresponding first database table identifier. For example, the predetermined algorithm is a message digest algorithm 5 (MD5). In addition, if the application request includes the first database table identifier corresponding to the first long link URL, the server can directly obtain the first database table identifier from the application request.

[0041] Step 230: Determine whether a first database corresponding to the first database table identifier falls within an access range of the server.

[0042] For example, if the first database table identifier is 67, the corresponding first database is a database 67. In this case, whether the server that receives the

application request can access the database 67 needs to be determined. If the server can access the database 67, the server can directly obtain the corresponding short link URL from the database 67; or if the server cannot access the database 67, the server needs to forward the received application request to a server that can access the database 67, and the server that accesses the database 67 is responsible for processing the application request.

[0043] Step 240: If it is determined that the first database falls within the access range of the server, obtain a corresponding first short link URL from the first database, and send the first short link URL to the requester.

[0044] The first database can include a plurality of pieces of data of the same short link, such as a long link URL, a corresponding digest value, a corresponding validity period, a corresponding simplified short link, and a corresponding short link URL. Alternatively, the first database can include only a corresponding simplified short link, and the corresponding simplified short link forms a corresponding short link URL; as such, database storage space can be saved.

[0045] In addition, after step 230 of determining whether a first database corresponding to the first database table identifier falls within an access range of the server is performed, the method further includes: if it is determined that the first database does not fall within the access range of the server, forwarding the received application request to a server that can access the first database, so that the server obtains the corresponding first short link URL from the first database based on the application request, and sends the first short link URL to the requester.

[0046] It can be seen from the above implementation that, the short link application request is received from the requester, and the application request includes the first long link URL. The corresponding first database table identifier is calculated based on the first long link URL, which is used to determine whether the first database corresponding to the first database table identifier falls within the access range of the server. If it is determined that the first database falls within the access range of the server, the corresponding first short link URL is obtained from the first database, and the first short link URL is sent to the requester. As such, in a database and table sharding mode, the server converts a long link URL to a corresponding short link URL. In addition, because the corresponding short link URL is unique and short, horizontal scalability of short link storage space is improved, and a short link processing speed is further increased.

[0047] Referring to FIG. 3, FIG. 3 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure. The method can be applied to any server in a distributed server cluster, and is based on the method shown in FIG. 2. When step 220 of calculating a corresponding first database table identifier based on the first long link URL is performed, the method can include the steps below.

[0048] Step 310: Calculate a corresponding first digest value based on the first long link URL.

[0049] In this implementation of the present disclosure, different algorithms can be used to calculate the first digest value corresponding to the first long link URL. The present disclosure includes but is not limited to the following two algorithms.

[0050] (1) The first digest value corresponding to the first long link URL is calculated by using a message digest algorithm, and the first digest value is a digest value of the message digest algorithm. The message digest algorithm can be MD5.

[0051] For example, the first long link URL is: https://cmspromo.alipay.com/finance/fullyear.htm. An MD5 digest value calculated based on the MD5 algorithm is 429C92076FC4DF64F20EBE0D392B9F27.

[0052] (2) The first digest value corresponding to the first long link URL is calculated by using a secure hash algorithm, and the first digest value is a digest value of the secure hash algorithm. The secure hash algorithm can be a secure hash algorithm 1 (SHA1).

[0053] Step 320: Calculate the corresponding first database table identifier based on the first digest value.

[0054] In this implementation of the present disclosure, different algorithms can be used to calculate the first database table identifier corresponding to the first digest value, and a basis for algorithm selection is that calculated first database table identifiers can be evenly distributed. For example, the first digest value is an MD5 digest value and is 429C92076FC4DF64F20EBE0D392B9F27.

[0055] First, an American Standard Code for Information Interchange (ASCII) code of each character of the MD5 digest value is obtained. Then the sum of the obtained ASCII codes is calculated. Finally, as shown in Equation (1), a REM operation is performed on 100 by using the calculated sum value to obtain a residual value, and the residual value is the first database table identifier ID.

[0056] ID = [ASCII(′4′)+ ASCII(′2′)+···+ ASCII(′7′)]%100=67...(1)

[0057] 67 is the calculated first database table identifier, and the first database table identifier is expressed as a number. In addition, the first database table identifier is expressed as any one or more characters, and is used to represent a sub-table name of a short link table.

[0058] It can be seen from the above implementation that, the message digest algorithm or the secure hash algorithm is used to calculate the first digest value corresponding to the first long link URL and the corresponding first database table identifier, so as to ensure that the short link records corresponding to the same long link URL are unique and valid, and avoid repeated short link creation, thereby increasing a short link processing speed.

[0059] Referring to FIG. 4, FIG. 4 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure. The method can be applied to any server in a distributed server cluster, and is based on the method shown in FIG. 3. When step 240 of obtaining a corresponding first short link URL from the first database, and sending the first short link URL to the requester is performed, the method can include the steps below.

[0060] Step 400: Search the first database for a first simplified short link corresponding to the first digest value.

[0061] In this implementation of the present disclosure, a simplified short link is a random character string of a pre-agreed number of characters in a short link.

[0062] Step 410: Determine whether the first simplified short link is found; and perform step 420 if the first simplified short link is found and a second long link URL corresponding to the found first simplified short link is the same as the first long link URL included in the application request, or perform step 440 if the first simplified short link is not found or a second long link URL corresponding to the found first simplified short link is different from the first long link URL included in the application request.

[0063] In this implementation of the present disclosure, the first database can include a plurality of pieces of data of the same short link, such as a long link URL, a corresponding digest value, a corresponding validity period, and a corresponding simplified short link.

[0064] Step 420: Form the first short link URL based on the found first simplified short link, and send the first short link URL to the requester.

[0065] Step 430: When the application request further includes a validity period

of the short link, update a validity period of the first simplified short link with the validity period included in the application request, and end the procedure.

[0066] A sequence of performing steps 420 and 430 is not limited in the present disclosure, and the steps can be performed simultaneously or in sequence.

[0067] Step 440: Generate a corresponding second simplified short link based on the first database table identifier. The second simplified short link is a simplified identifier of a fixed number of characters, for example, 7 characters.

[0068] In this implementation of the present disclosure, different rules can be used to generate the corresponding second simplified short link. The generation rule can be that the first two characters of the simplified identifier are calculated based on the first database table identifier and a pre-agreed algorithm, and the subsequent five characters of the simplified identifier, Y, v, 3, M, and Z, are random characters.

[0069] For example, the first database table identifier is 67. Based on an algorithm through which a result 6 is obtained by using ASCII(′X′)%100 and a result 7 is obtained by using ASCII(′Y′)%100, a value of X can be a random character in [8|j|t|B|L|V], and a value of Y can be a random character in [9|a|k|u|C|M|W]. As such, t and 9 can be randomly selected as the first two characters of the simplified identifier, and Y, v, 3, M, and Z can be randomly selected as the subsequent five characters of the simplified identifier, so that the obtained second simplified short link is t9Yv3MZ.

[0070] In addition, another algorithm can be used to calculate the first two characters of the simplified identifier, and a basis for algorithm selection is that characters can be evenly distributed, and the first database table identifier can be deduced based on calculated characters. Certainly, the first database table identifier is unnecessarily represented by the first two characters of the simplified identifier, and can be a pre-agreed simplified identifier character at any location.

[0071] Step 450: If it is determined that the second simplified short link is not used, form a second short link URL based on the second simplified short link, and send the second short link URL to the requester.

[0072] In this implementation of the present disclosure, after the corresponding second simplified short link is generated based on the first database table identifier, whether the generated second simplified short link is used further needs to be determined. If the second simplified short link is used, a second simplified short link needs to be regenerated, until a newly generated second simplified short link is not used.

[0073] Step 460: When the application request further includes a validity period of the short link, set a validity period of the second simplified short link to the validity period included in the application request.

[0074] Step 470: Bind the second simplified short link, the corresponding first long link URL, the corresponding first digest value, and a corresponding validity period, and store them in the first database.

[0075] A sequence of performing steps 450, 460, and 470 is not limited in the present disclosure, and the steps can be performed simultaneously or in sequence.

[0076] It can be seen from the above implementation that, when the first database is searched for the first simplified short link corresponding to the first digest value, different processing methods can be used based on different results obtained through searching. Especially when the first simplified short link is not found or the second long link URL corresponding to the found first simplified short link is different from the first long link URL included in the application request, the corresponding second simplified short link further needs to be generated based on the first database table identifier, and the generated second simplified short link has a fixed number of characters. As such, the server can generate a simplified short link as short as possible while ensuring that the short link records corresponding to the same long link URL are unique and valid, thereby saving content space for transmitting the short link.

[0077] Referring to FIG. 5, FIG. 5 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure. The method can be applied to any server in a distributed server cluster, and is based on the method shown in any one of FIG. 2 to FIG. 4. The method can further include the steps below.

[0078] Step 510: Detect a validity period of each short link in the first database.

[0079] In this implementation of the present disclosure, the validity period of each short link in the first database can be detected periodically.

[0080] Step 520: Delete corresponding storage information when the validity period expires.

[0081] For example, the server detects the validity period of each short link in the first database once every five minutes, and directly deletes a record of an expired short link, so as to improve effective utilization of storage space, and keep a healthy expansion of short link storage space.

[0082] It can be seen from the above implementation that, the validity period of

each short link in the first database is detected, and the corresponding storage information is deleted when the validity period expires, thereby improving effective utilization of storage space, and keeping a healthy expansion of short link storage space.

[0083] Referring to FIG. 6, FIG. 6 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure. The method can be applied to any server in a distributed server cluster, and is based on the method shown in any one of FIG. 2 to FIG. 4. The method can further include the steps below.

[0084] Step 610: Receive a short link access request from the requester, where the access request includes a third short link URL.

[0085] In this implementation of the present disclosure, the server can receive the short link access request from the requester. The purpose of the access request is to enable the requester to obtain a long link URL corresponding to the third short link URL.

[0086] Step 620: Calculate a corresponding second database table identifier based on the third short link URL.

[0087] In this implementation of the present disclosure, the server can perform corresponding calculation on the third short link URL by using a predetermined algorithm, to obtain the corresponding second database table identifier. In addition, if the third short link URL includes the corresponding second database table identifier, the server can directly obtain the second database table identifier from the third short link URL.

[0088] Step 630: Determine whether a second database corresponding to the second database table identifier falls within the access range of the server.

[0089] Step 640: If it is determined that the second database falls within the access range of the server, obtain a corresponding third long link URL from the second database, and jump to the third long link URL.

[0090] In this implementation of the present disclosure, the second database can include a plurality of pieces of data of the same short link, such as a long link URL, a corresponding digest value, a corresponding validity period, a corresponding simplified short link, and a corresponding short link URL. Alternatively, the second database can include only a corresponding simplified short link, and the corresponding simplified short link forms a corresponding short link URL; as such,

database storage space can be saved.

[0091] It can be seen from the above implementation that, the short link access request is received from the requester, and the access request includes the third short link URL. The corresponding second database table identifier is calculated based on the third short link URL, which is used to determine whether the second database corresponding to the second database table identifier falls within the access range of the server. If it is determined that the second database falls within the access range of the server, the corresponding third long link URL is obtained from the second database, and the third long link URL is jumped to. As such, in a database and table sharding mode, the server jumps from a short link URL to a corresponding long link URL. In addition, because the corresponding long link URL is unique, accuracy of short link processing is improved.

[092] Referring to FIG. 7, FIG. 7 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure. The method can be applied to any server in a distributed server cluster, and is based on the method shown in FIG. 6. When step 620 of calculating a corresponding second database table identifier based on the third short link URL is performed, the method can include the steps below.

[093] Step 710: Extract a third simplified short link from the third short link URL.

[094] For example, the third short link URL is https://m.alipay.com/t9Yv3MZ, and t9Yv3MZ is the third simplified short link.

[095] Step 720: Generate the corresponding second database table identifier based on the third simplified short link.

[096] In this implementation of the present disclosure, an algorithm for generating a corresponding database table identifier based on a simplified short link corresponds to an algorithm for generating a corresponding simplified short link based on a database table identifier.

[097] For example, t9Yv3MZ is the third simplified short link, t and 9 are calculated based on a pre-agreed algorithm, and the subsequent five characters of the simplified identifier, Y, v, 3, M, and Z, are random characters. The server can calculate the corresponding second database table identifier based on t and 9. That is, a REM operation is performed on 10 by using ASCII codes of t and 9, to obtain the corresponding second database table identifier 76.

[098] It can be seen from the above implementation that, the third simplified short link is extracted from the third short link URL, and the corresponding second database table identifier is generated based on the third simplified short link. In addition, an algorithm for generating a corresponding database table identifier based on a simplified short link corresponds to an algorithm for generating a corresponding simplified short link based on a database table identifier, thereby ensuring uniqueness between the simplified short link and the database table identifier.

[099] Referring to FIG. 8, FIG. 8 is a flowchart illustrating another implementation of a short link processing method according to the present disclosure. The method can be applied to any server in a distributed server cluster, and is based on the method shown in FIG. 6. When step 640 of obtaining a corresponding third long link URL from the second database, and jumping to the third long link URL is performed, the method can include the steps below.

[0100] Step 800: Search the second database for the third simplified short link.

[0101] Step 810: Determine whether the third simplified short link is found and whether a validity period of the third simplified short link does not expire, and perform step 820 if the third long link URL is found and the validity period of the third simplified short link does not expire, or perform step 830 if the third simplified short link is not found or the validity period of the third simplified short link expires.

[0102] Step 820: Jump to the third long link URL corresponding to the third simplified short link.

[0103] For example, the third long link URL is: https://cmspromo.alipay.com/finance/fullyear.htm.

[0104] Step 830: Jump to a specified URL.

[0105] In this implementation of the present disclosure, the specified URL can be a common default URL. For example, the common default URL is https://d.alipay.com.

[0106] It can be seen from the above implementation that, when the second database is searched for the third simplified short link, different URLs can jump to an URL based on different results obtained through searching, thereby satisfying different access requirements from the requester, and increasing a short link processing speed.

[0107] Corresponding to the implementations of the short link processing method in the present disclosure, the present disclosure further provides implementations of a

short link processing apparatus.

[0108] The implementations of the short link processing apparatus can be applied to any server in a distributed server cluster. The apparatus implementation can be implemented by using software, hardware, or both. Software implementation is used as an example. As a logical apparatus, the apparatus is formed by reading a corresponding computer program instruction in a non-volatile memory to a memory by a processor of a device where the apparatus is located. From a perspective of hardware, as shown in FIG. 9, FIG. 9 is a structural diagram of hardware of a device where a short link processing apparatus is located. In addition to a processor, a memory, a network interface, and a non-volatile memory shown in FIG. 9, the device where the apparatus in this implementation is located can include other hardware based on actual functions of the device. For example, for a terminal, the apparatus can include a camera, a touchscreen, a communications component, etc.; and for a server, the apparatus can include a forwarding chip, etc., that is responsible for processing packets.

[0109] Referring to FIG. 10, FIG. 10 is a block diagram illustrating an implementation of a short link processing apparatus according to the present disclosure. The short link processing apparatus can be applied to a short link processing system, and can be configured to perform the short link processing method in the above implementation. The apparatus includes a first receiving unit 101, a first calculation unit 102, a first determining unit 103, and a first processing unit 104.

[0110] The first receiving unit 101 is configured to receive a short link application request from a requester, where the application request includes a first long link URL.

[0111] The first calculation unit 102 is configured to obtain a corresponding first database table identifier based on the first long link URL.

[0112] The first determining unit 103 is configured to determine whether a first database corresponding to the first database table identifier falls within an access range of system.

[0113] The first processing unit 104 is configured to obtain a corresponding first short link URL from the first database, and send the first short link URL to the requester, if it is determined that the first database falls within the access range of the system.

[0114] In an optional implementation, the first calculation unit 102 can include a first calculation subunit and a second calculation subunit (not shown in FIG. 10).

[0115] The first calculation subunit is configured to calculate a corresponding first digest value based on the first long link URL.

[0116] The second calculation subunit is configured to calculate the corresponding first database table identifier based on the first digest value.

[0117] In addition, the first calculation subunit is specifically configured to calculate the first digest value corresponding to the first long link URL by using a message digest algorithm, where the first digest value is a digest value of the message digest algorithm; or calculate the first digest value corresponding to the first long link URL by using a secure hash algorithm, where the first digest value is a digest value of the secure hash algorithm.

[0118] In addition, the first database table identifier is expressed as a number or a character.

[0119] In another optional implementation, the first processing unit 103 can include a first search subunit and a first processing subunit (not shown in FIG. 10).

[0120] The first search subunit is configured to search the first database for a first simplified short link corresponding to the first digest value.

[0121] The first processing subunit is configured to form the first short link URL based on the found first simplified short link, and send the first short link URL to the requester, if the first simplified short link is found and a second long link URL corresponding to the found first simplified short link is the same as the first long link URL included in the application request

[0122] In another optional implementation, the application request further includes a validity period of the short link, and the first processing unit 103 can further include an updating subunit (not shown in FIG. 10).

[0123] The updating subunit is configured to update a validity period of the first simplified short link with the validity period included in the application request.

[0124] In another optional implementation, the first processing unit 103 can further include a generation subunit and a second processing subunit (not shown in FIG. 10).

[0125] The generation subunit is configured to generate a corresponding second simplified short link based on the first database table identifier if the first simplified short link is not found or the second long link URL corresponding to the found first simplified short link is different from the first long link URL included in the application request.

[0126] The second processing subunit is configured to form a second short link URL based on the second simplified short link, and send the second short link URL to the requester, if it is determined that the second simplified short link is not used.

[0127] In another optional implementation, the application request further includes a validity period of the short link, and the first processing unit 103 can further include a setting subunit (not shown in FIG. 10).

[0128] The setting subunit is configured to set a validity period of the second simplified short link to the validity period included in the application request.

[0129] In another optional implementation, the first processing unit 103 can further include a storage subunit (not shown in FIG. 10).

[0130] The storage subunit is configured to bind the second simplified short link, the corresponding first long link URL, the corresponding first digest value, and a corresponding validity period, and store them in the first database.

[0131] In another optional implementation, the short link processing apparatus further includes a second processing unit (not shown in FIG. 10).

[0132] The second processing unit is configured to forward the application request to a server that can access the first database, so that the server obtains the corresponding first short link URL from the first database based on the application request, and sends the first short link URL to the requester, if it is determined that the first database does not fall within the access range of the system.

[0133] Referring to FIG. 11, FIG. 11 is a block diagram illustrating another implementation of a short link processing apparatus according to the present disclosure. The short link processing apparatus can be applied to a short link processing system, and is based on the apparatus shown in FIG. 10. The apparatus can further include a detection unit 111 and a deletion unit 112.

[0134] The detection unit 111 is configured to detect a validity period of each short link in the first database.

[0135] The deletion unit 112 is configured to delete corresponding storage information when the validity period expires.

[0136] Referring to FIG. 12, FIG. 12 is a block diagram illustrating another implementation of a short link processing apparatus according to the present disclosure. The short link processing apparatus can be applied to a short link processing system, and can be based on the apparatus shown in FIG. 10. The apparatus can further include a second receiving unit 121, a second calculation unit

122, a second determining unit 123, and a second processing unit 124.

[0137] The second receiving unit 121 is configured to receive a short link access request from the requester, where the access request includes a third short link URL.

[0138] The second calculation unit 122 is configured to calculate a corresponding second database table identifier based on the third short link URL.

[0139] The second determining unit 123 is configured to determine whether a second database corresponding to the second database table identifier falls within an access range of the system.

[0140] The third processing unit 124 is configured to obtain a corresponding third long link URL from the second database and jump to the third long link URL if it is determined that the second database falls within the access range of the system.

[0141] In an optional implementation, the second calculation unit 122 can include an extraction subunit and a generation subunit (not shown in FIG. 12).

[0142] The extraction subunit is configured to extract a third simplified short link from the third short link URL.

[0143] The generation subunit is configured to generate the corresponding second database table identifier based on the third simplified short link.

[0144] In another optional implementation, the second processing unit 123 can include a search subunit and a third processing subunit (not shown in FIG. 12).

[0145] The search subunit is configured to search the second database for the third simplified short link.

[0146] The third processing subunit is configured to jump to the third long link URL corresponding to the third simplified short link if the third simplified short link is found and a validity period of the third simplified short link does not expire.

[0147] In another optional implementation, the third processing unit 123 can further include a fourth processing subunit (not shown in FIG. 12).

[0148] The fourth processing subunit is configured to jump to a specified URL if the third simplified short link is not found or the validity period of the third simplified short link expires.

[0149] In another optional implementation, the short link processing apparatus further includes a fourth processing unit (not shown in FIG. 12).

[0150] The fourth processing unit is configured to forward the access request to a server that can access the first database, so that the server obtains the corresponding third short link URL from the second database based on the access request, and jumps

to the third long link URL, if it is determined that the second database does not fall within the access range of the system.

[0151] For a detailed implementation process of functions and roles of each unit in the apparatus, refer to an implementation process of a corresponding step in the previous method. Details are omitted here.

[0152] An apparatus implementation basically corresponds to a method implementation. As such, for related parts, refer to related descriptions in the method implementation. The described apparatus implementation is merely an example. The units described as separate parts can be physically separate or not, and parts displayed as units can be physical units or not, can be located in one position, or can be distributed on a plurality of network units. Some or all of the modules can be selected based on actual requirements to achieve the objectives of the solutions of the present disclosure. A person of ordinary skill in the art can understand and implement the implementations of the present disclosure without creative efforts.

[0153] An implementation of the present disclosure further provides a server, and the server includes: a processor, and a memory configured to store an instruction that can be executed by the processor.

[0154] The processor is configured to: receive a short link application request from a requester, where the application request includes a first long link URL; calculate a corresponding first database table identifier based on the first long link URL; determine whether a first database corresponding to the first database table identifier falls within an access range of the server; and obtain a corresponding first short link URL from the first database, and send the first short link URL to the requester, if it is determined that the first database falls within the access range of the server.

[0155] It can be seen from the above implementation that, the short link application request is received from the requester, and the application request includes the first long link URL. The corresponding first database table identifier is calculated based on the first long link URL, which is used to determine whether the first database corresponding to the first database table identifier falls within the access range of the server. If it is determined that the first database falls within the access range of the server, the corresponding first short link URL is obtained from the first database, and the first short link URL is sent to the requester. As such, in a database and table sharding mode, the server converts a long link URL to a corresponding short

link URL. In addition, because the corresponding short link URL is unique and short, horizontal scalability of short link storage space is improved, and a short link processing speed is further increased.

[0156] A person skilled in the art can easily figure out another implementation solution of the present disclosure after considering the specification and practicing the present disclosure. The present disclosure is intended to cover any variations, functions, or adaptive changes of the present disclosure. These variations, functions, or adaptive changes comply with general principles of the present disclosure, and include common knowledge or a commonly used technical means in the technical field that is not disclosed in the present disclosure. The specification and the implementations are merely considered as examples, and the actual scope and the spirit of the present disclosure are described by the following claims.

[0157] It should be understood that the present disclosure is not limited to the accurate structures that are described in the foregoing and that are shown in the accompanying drawings, and modifications and changes can be made without departing from the scope of the present disclosure. The scope of the present disclosure is limited only by the appended claims.

CLAIMS

What is claimed is:

1. A short link processing method, wherein the method is applied to any server in a distributed server cluster and comprises:

receiving a short link application request from a requester, wherein the application request comprises a first long link URL;

calculating a corresponding first database table identifier based on the first long link URL;

determining whether a first database corresponding to the first database table identifier falls within an access range of the server; and

if it is determined that the first database falls within the access range of the server, obtaining a corresponding first short link URL from the first database, and sending the first short link URL to the requester.

2. The method according to claim 1, wherein the calculating a corresponding first database table identifier based on the first long link URL comprises:

calculating a corresponding first digest value based on the first long link URL; and

calculating the corresponding first database table identifier based on the first digest value.

3. The method according to claim 2, wherein the calculating a corresponding first digest value based on the first long link URL comprises:

calculating the first digest value corresponding to the first long link URL by using a message digest algorithm, wherein the first digest value is a digest value of the message digest algorithm; or

calculating the first digest value corresponding to the first long link URL by using a secure hash algorithm, wherein the first digest value is a digest value of the secure hash algorithm.

4. The method according to claim 2, wherein the first database table identifier is expressed as a number or a character.

5. The method according to claim 2, wherein the obtaining a corresponding first short link URL from the first database, and sending the first short link URL to the requester comprises:

searching the first database for a first simplified short link corresponding to the first digest value; and

if the first simplified short link is found and a second long link URL corresponding to the found first simplified short link is the same as the first long link URL comprised in the application request, forming the first short link URL based on the found first simplified short link, and sending the first short link URL to the requester.

6. The method according to claim 5, wherein the application request further comprises a validity period of the short link, and the method further comprises:

updating a validity period of the first simplified short link with the validity period comprised in the application request.

7. The method according to claim 5, wherein the obtaining a corresponding first short link URL from the first database, and sending the first short link URL to the requester further comprises:

if the first simplified short link is not found or the second long link URL corresponding to the found first simplified short link is different from the first long link URL comprised in the application request, generating a corresponding second simplified short link based on the first database table identifier; and

if it is determined that the second simplified short link is not used, forming a second short link URL based on the second simplified short link, and sending the second short link URL to the requester.

8. The method according to claim 7, wherein the application request further comprises a validity period of the short link, and the method further comprises:

setting a validity period of the second simplified short link to the validity period comprised in the application request.

9. The method according to claim 7, wherein the method further comprises:

binding and storing, in the first database, the second simplified short link, the corresponding first long link URL, the corresponding first digest value, and a corresponding validity period.

10. The method according to claim 1, wherein after the determining whether a first database corresponding to the first database table identifier falls within an access range of the server, the method further comprises:

if it is determined that the first database does not fall within the access range of the server, forwarding the application request to a server that can access the first

database, so that the server obtains the corresponding first short link URL from the first database based on the application request, and sends the first short link URL to the requester.

11. The method according to any one of claims 1 to 10, wherein the method further comprises:

detecting a validity period of each short link in the first database; and

deleting corresponding storage information when the validity period expires.

12. The method according to any one of claims 1 to 10, wherein the method further comprises:

receiving a short link access request from the requester, wherein the access request comprises a third short link URL;

calculating a corresponding second database table identifier based on the third short link URL;

determining whether a second database corresponding to the second database table identifier falls within the access range of the server; and

if it is determined that the second database falls within the access range of the server, obtaining a corresponding third long link URL from the second database, and jumping to the third long link URL.

13. The method according to claim 12, wherein the calculating a corresponding second database table identifier based on the third short link URL comprises:

extracting a third simplified short link from the third short link URL; and

generating the corresponding second database table identifier based on the third simplified short link.

14. The method according to claim 13, wherein the obtaining a corresponding third long link URL from the second database, and jumping to the third long link URL comprises:

searching the second database for the third simplified short link; and

if the third simplified short link is found and a validity period of the third simplified short link does not expire, jumping to the third long link URL corresponding to the third simplified short link.

15. The method according to claim 14, wherein the obtaining a corresponding third long link URL from the second database corresponding to the second database table identifier, and jumping to the third long link URL further comprises:

jumping to a specified URL if the third simplified short link is not found or the

validity period of the found third simplified short link expires.

16. The method according to claim 12, wherein after the determining whether a second database corresponding to the second database table identifier falls within the access range of the server, the method further comprises:

if it is determined that the second database does not fall within the access range of the server, forwarding the access request to a server that can access the first database, so that the server obtains the corresponding third short link URL from the second database based on the access request, and jumps to the third long link URL.

17. A short link processing apparatus, wherein the apparatus is applied to any server in a distributed server cluster and comprises:

a first receiving unit, configured to receive a short link application request from a requester, wherein the application request comprises a first long link URL;

a first calculation unit, configured to obtain a corresponding first database table identifier based on the first long link URL;

a first determining unit, configured to determine whether a first database corresponding to the first database table identifier falls within an access range of the server; and

a first processing unit, configured to: if it is determined that the first database falls within the access range of the server, obtain a corresponding first short link URL from the first database, and send the first short link URL to the requester.

18. The apparatus according to claim 17, wherein the first calculation unit comprises:

a first calculation subunit, configured to calculate a corresponding first digest value based on the first long link URL; and

a second calculation subunit, configured to calculate the corresponding first database table identifier based on the first digest value.

19. The apparatus according to claim 18, wherein the first calculation subunit is specifically configured to: calculate the first digest value corresponding to the first long link URL by using a message digest algorithm, wherein the first digest value is a digest value of the message digest algorithm; or calculate the first digest value corresponding to the first long link URL by using a secure hash algorithm, wherein the first digest value is a digest value of the secure hash algorithm.

20. The apparatus according to claim 18, wherein the first database table identifier is expressed as a number or a character.

21. The apparatus according to claim 18, wherein the first processing unit comprises:

a first search subunit, configured to search the first database for a first simplified short link corresponding to the first digest value; and

a first processing subunit, configured to: if the first simplified short link is found and a second long link URL corresponding to the found first simplified short link is the same as the first long link URL comprised in the application request, form the first short link URL based on the found first simplified short link, and send the first short link URL to the requester.

22. The apparatus according to claim 21, wherein the application request further comprises a validity period of the short link, and the first processing unit further comprises:

an updating subunit, configured to update a validity period of the first simplified short link with the validity period comprised in the application request.

23. The apparatus according to claim 21, wherein the first processing unit further comprises:

a generation subunit, configured to: if the first simplified short link is not found or the second long link URL corresponding to the found first simplified short link is different from the first long link URL comprised in the application request, generate a corresponding second simplified short link based on the first database table identifier; and

a second processing subunit, configured to: if it is determined that the second simplified short link is not used, form a second short link URL based on the second simplified short link, and send the second short link URL to the requester.

24. The apparatus according to claim 23, wherein the application request further comprises a validity period of the short link, and the first processing unit further comprises:

a setting subunit, configured to set a validity period of the second simplified short link to the validity period comprised in the application request.

25. The apparatus according to claim 23, wherein the first processing unit further comprises:

a storage subunit, configured to bind the second simplified short link, the corresponding first long link URL, the corresponding first digest value, and a corresponding validity period, and store them in the first database.

26. The apparatus according to claim 17, wherein the apparatus further comprises:

a second processing unit, configured to: if it is determined that the first database does not fall within the access range of the server, forward the application request to a server that can access the first database, so that the server obtains the corresponding first short link URL from the first database based on the application request, and sends the first short link URL to the requester.

27. The apparatus according to any one of claims 17 to 26, wherein the apparatus further comprises:

a detection unit, configured to detect a validity period of each short link in the first database; and

a deletion unit, configured to delete corresponding storage information when the validity period expires.

28. The apparatus according to any one of claims 17 to 26, wherein the apparatus further comprises:

a second receiving unit, configured to receive a short link access request from the requester, wherein the access request comprises a third short link URL;

a second calculation unit, configured to calculate a corresponding second database table identifier based on the third short link URL;

a second determining unit, configured to determine whether a second database corresponding to the second database table identifier falls within the access range of the server; and

a third processing unit, configured to: if it is determined that the second database falls within the access range of the server, obtain a corresponding third long link URL from the second database, and jump to the third long link URL.

29. The apparatus according to claim 28, wherein the second calculation unit comprises:

an extraction subunit, configured to extract a third simplified short link from the third short link URL; and

a generation subunit, configured to generate the corresponding second database table identifier based on the third simplified short link.

30. The apparatus according to claim 29, wherein the third processing unit comprises:

a search subunit, configured to search the second database for the third

simplified short link; and

a third processing subunit, configured to: if the third simplified short link is found and a validity period of the third simplified short link does not expire, jump to the third long link URL corresponding to the third simplified short link.

31. The apparatus according to claim 30, wherein the third processing unit further comprises:

a fourth processing subunit, configured to jump to a specified URL if the third simplified short link is not found or the validity period of the found third simplified short link expires.

32. The apparatus according to claim 28, wherein the apparatus further comprises:

a fourth processing unit, configured to: if it is determined that the second database does not fall within the access range of the server, forward the access request to a server that can access the first database, so that the server obtains the corresponding third short link URL from the second database based on the access request, and jumps to the third long link URL.

33. A server, wherein the server is any server in a distributed server cluster, and comprises:

a processor, and a memory configured to store an instruction that can be executed by the processor, wherein

the processor is configured to:

receive a short link application request from a requester, wherein the application request comprises a first long link URL;

calculate a corresponding first database table identifier based on the first long link URL;

determine whether a first database corresponding to the first database table identifier falls within an access range of the server; and

if it is determined that the first database falls within the access range of the server, obtain a corresponding first short link URL from the first database, and send the first short link URL to the requester.