



- (51) **International Patent Classification:**
H04W 8/24 (2009.01)
- (21) **International Application Number:**
PCT/CN2013/090046
- (22) **International Filing Date:**
20 December 2013 (20.12.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
201210562756.6 21 December 2012 (21.12.2012) CN
- (71) **Applicant: TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED** [CN/CN]; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN).
- (72) **Inventors: ZHANG, Bifeng**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **MA, Huafeng**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **SHEN, Qiliang**; Room

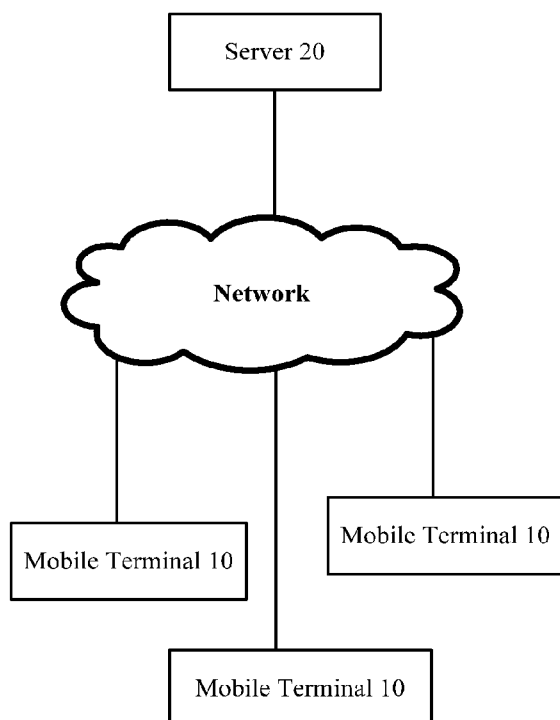
403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **LIN, Shili**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **XIAO, Xiao**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **GUO, Tongyi**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **ZHANG, Kun**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN).

(74) **Agent: DEQI INTELLECTUAL PROPERTY LAW CORPORATION**; 7/F, Xueyuan International Tower, No. 1 Zhichun Road, Haidian District, Beijing 100083 (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, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,

[Continued on next page]

(54) **Title:** METHODFOR MANAGING CACHE DATA OF A MOBILE TERMINAL, A MOBILE TERMINAL, AND A SERVER



(57) **Abstract:** The present disclosure discloses a method for managing cache data of a mobile terminal, comprising: determining whether first cache data corresponding to an application exists; and if it is determined that the first cache data does not exist, sending a data request to request for server data corresponding to the first cache data; and downloading the server data and forming the first cache data by using the downloaded server data. Accordingly, the present disclosure also discloses a mobile terminal and a server for managing cache data of the mobile terminal.

FIG. 1



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.

TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,

Published:

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

METHOD FOR MANAGING CACHE DATA OF A MOBILE TERMINAL, A MOBILE TERMINAL, AND A SERVER

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The application claims the benefit of Chinese Patent Application No. 201210562756.6, filed on December 21, 2012, the disclosure of which is incorporated herein in its entirety by reference.

TECHNICAL FIELD

[0002] The present disclosure relates to communications technologies, and particularly to a method for managing cache data of a mobile terminal, a mobile terminal, and a server.

BACKGROUND

[0003] Recently, technologies regarding mobile terminals (e.g., smart phones) have developed rapidly. However, when people enjoy convenience of a mobile terminal, they also meet with a problem that the cache of the mobile terminal is occupied too much, which influences the running speeds of system software and application software on the mobile terminal. In the traditional art, when a mobile terminal, such as a smart phone, is in an environment without network, the application software thereof often accesses to data cached locally. Nevertheless, the local cache data is not preferentially loaded according to the user's habit, thus it may be slow to find the cache data, which may influence the user experience. Also, the user has to frequently clean the cache to prevent the data in the cache from occupying too much system storage space. In an environment with network, application software may often download related data from a server. This process occupies the bandwidth heavily, and wastes network resources. Furthermore, the download speed may be affected by the network environment.

SUMMARY

[0004] To solve problems existing in the traditional art that in a mobile terminal (e.g., cell phone), application software accesses a cache over frequently, the speed of searching for a

cache file is slow, data downloading is too much, and network flow is wasted etc., the present disclosure provides the following schemes.

[0005] In an aspect, the present disclosure provides a method for managing cache data, including the following steps: determining whether first cache data corresponding to an application exists; and if it is determined that the first cache data does not exist, sending a data request to request for server data corresponding to the first cache data; and downloading the server data and forming the first cache data by using the downloaded server data.

[0006] In an aspect, the present disclosure provides a mobile terminal, including: memory; one or more processors; and one or more programs stored in the memory and configured for execution by the one or more processors, the one or more programs including instructions to: determine whether first cache data corresponding to an application exists; and if it is determined that the first cache data does not exist, send a data request to request for server data corresponding to the first cache data, and download the server data and form the first cache data by using the server data.

[0007] In an aspect, the present disclosure provides a server, including: memory; one or more processors; and one or more programs stored in the memory and configured for execution by the one or more processors, the one or more programs including instructions to: if first cache data corresponding to an application does not exist; receive a data request that requests for server data corresponding to the application; and send the server data to form the first cache data.

[0008] According to the above schemes, in an environment where the network condition is good, the local cache can be used according to the cache situations thereof and the expiration time returned from the server; since in some conditions, there is no need to re-download the server data, which reduces waste of network traffic; the cache file stored locally is in the form of object stream instead of the original file, which avoids a second analysis, improves the loading speed, enhances the user's experience; the cache data of the mobile terminal takes precedence according to the frequency of usage by the user, which also improves the loading speed; the cache data is highly efficiently compressed, which reduces the storage space occupied by the cache data in the mobile terminal .

[0009] To make the above content of the present disclosure easily understood, the preferable embodiments of the present disclosure will be illustrated in conjunction with the accompanying figures, and will be described in detail as follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] For a better understanding of the present disclosure, reference should be made to the Detailed Description below, in conjunction with the following drawings in which like reference numerals refer to corresponding parts throughout the figures.

[0011] Fig.1 shows system architecture according to an example of the present disclosure;

[0012] Fig.2 shows a flow chart of a method for managing cache data of a mobile terminal according to an example of the present disclosure;

[0013] Fig.3 shows a detailed flow chart of determining whether the first cache data is expired according to an example of the present disclosure;

[0014] Fig.4 shows a detailed flow chart of downloading server data from a server to a cache of the mobile terminal to form the first cache data according to an example of the present disclosure;

[0015] Fig.5 shows a detailed flow chart of downloading server data from the server to the cache of the mobile terminal to form the second cache data according to an example of the present disclosure; and

[0016] Fig.6 shows a structural diagram of a system for managing cache data of the mobile terminal according to an example of the present disclosure.

DETAILED DESCRIPTION

[0017] Reference will now be made in detail to examples, which are illustrated in the accompanying drawings. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present disclosure. Also, the figures are illustrations of an example, in which modules or procedures shown in the figures are not necessarily essential for implementing the present disclosure. In other instances, well-

known methods, procedures, components, and circuits have not been described in detail so as not to unnecessarily obscure aspects of the examples.

[0018] Fig.1 shows system architecture according to an example of the present disclosure. As seen from Fig.1, the system includes some mobile terminals 10 and a server 20 which can communicate with each other via the network. Fig.1 is just an example, and the present disclosure is not limited to this.

[0019] The detailed technical scheme of the present disclosure is described as follows.

[0020] Fig.2 shows a flowchart of a method for managing cache data of a mobile terminal according to an example of the present disclosure.

[0021] As seen from Fig.2, at block 101, the mobile terminal runs an application (APP) on the mobile terminal. In an example, the application may be a third party APP.

[0022] At block 102, the mobile terminal checks a cache thereof and determines whether first cache data corresponding to the application exists in the cache. If it is determined that the first cache data does not exist in the cache of the mobile terminal, the flow goes to block 103, and if it is determined that the first cache data exists in the cache of the mobile terminal, then the flow goes to block 105.

[0023] At block 103, the mobile terminal sends a data request to the server via the network connection to request for downloading server data corresponding to the application. After the server receives the data request it will send the server data corresponding to the application to the mobile terminal. And at block 104, the mobile terminal downloads the server data corresponding to the application from the server and saves the downloaded server data to the cache of the mobile terminal to form the first cache data. The server data refers to data at the server corresponding to the cache data. Generally speaking, the server data is updated more frequently than the cache data at the mobile terminal. And under the condition that the first cache data is not in the cache, the mobile terminal will download the newest data from the server to the cache through the network connection. Establishing a network connection between an application (APP) of a mobile terminal, e.g., a smart phone, and a server is one of the basic functions of a mobile terminal (e.g., a cell phone), which will not be elaborated here.

[0024] At block 105, if the first cache data exists in the cache, the mobile terminal checks the first cache data to determine whether the first cache data is expired. For example, this is determined by checking expiration time of the first cache data. If the first cache data is not expired, the flow goes to block 106, where the first cache data still can be used for the application. And if the first cache data is expired, the flow goes to block 107, where the mobile terminal sends a data request corresponding to the first cache data to the server. The data request corresponding to the first cache data contains file description information which includes the expiration time of the first cache data. Thus the file description information containing the expiration time is uploaded to the server so that the server can record an address of the file description information to facilitate the file description information being read at any time. In addition, the file description information includes storage time of the first cache data and the storage time is sent to the server too. The storage time may be the time when the cache data is updated, i.e., update time. Then the flow goes to block 108.

[0025] At block 108, after the server receives the data request containing the file description information corresponding to the first cache data from the mobile terminal, the server compares the expiration time included in the file description information with update time of the corresponding server data at the server to determine whether the first cache data of the application is out of date. The server may also compare the storage time with the update time of the corresponding server data at the server to determine whether the first cache data needs to be updated. And if the first cache data needs not to be updated, the flow goes to block 109 or if the first cache data needs to be updated, the flow goes to block 110.

[0026] At block 109, the server returns prompt information to the mobile terminal to prompt that the server data requested by the mobile terminal has existed in the cache of the mobile terminal. Specifically, when the first cache data is in the cache of the mobile terminal and the first cache data needs not to be updated immediately, the server will notify the mobile terminal to continue to use the first cache data and will not resend the server data to the mobile terminal. Thus there is no need to download server data from the server to the cache of the mobile terminal, which efficiently avoids wasting network resources and reduces network transmission traffic.

[0027] At block 110, if the first cache data is expired and needs to be updated, the server sends the server data corresponding to the application to the mobile terminal via the network

connection and the mobile terminal downloads the server data corresponding to the application from the server and updates the first cache data by using the server data to form second cache data.

[0028] The server data refers to data at the server corresponding to the cache data, and generally speaking, the server data is updated more frequently than the cache data at the mobile terminal. The server controls cache status and cache time of the cache at the mobile terminal so as to adjust and optimize the cache at any time.

[0029] Referring to Fig.3, it shows a detailed flow chart of determining whether the first cache data is expired and needs to be updated according to an example of the present invention. In Fig.3, the step of the server determining whether the first cache data is expired includes the following steps. At block 201, the server obtains the storage time of the first cache data, i.e., the time when the first cache data was updated at the mobile terminal. At block 202, the server obtains the update time of the server data; the server data is periodically updated by the server. At block 203, the server compares the storage time and the update time, obtains a comparison result, and forms the comparison result into response data. At block 204, the server sends the comparison result, i.e., the response data, to the mobile terminal. The application on the mobile terminal analyzes the response data so as to get a return code.

[0030] Referring to Fig.4, Fig.4 shows a detailed flow chart of downloading the server data from the server to the cache of the mobile terminal so as to form the first cache data according to an example of the present disclosure. At block 301, the mobile terminal downloads the server data from the server. At block 302, the mobile terminal writes the server data in the form of object stream to the cache thereof to form the first cache data. At block 303, the mobile terminal compresses the first cache data. And at block 304, the mobile terminal stores the compressed first cache data to the cache thereof.

[0031] Referring to Fig.5, it shows a detailed flowchart of downloading the server data from the server to the cache of the mobile terminal to form the second cache data according to an example of the present disclosure. At block 401, the mobile terminal downloads the server data from the server. At block 402, the mobile terminal replaces the first cache data by using the server data in the form of object stream so as to update the first cache data to the second cache data. At block 403, the mobile terminal compresses the second cache data. At

block 404, the mobile terminal stores the compressed second cache data to the cache of the mobile terminal.

[0032] The second cache data is in the form of object stream and is not the original file, which avoids a second analysis, increases the loading speed and enhances the user's experience.

[0033] The application running on the mobile terminal determines whether the return code returned from the server is smaller than an overflow value, and if the return code is larger than the overflow value, then information indicating that the data request of user is failed will be displayed on an interface of the application; and if the return code is smaller than or equal to the overflow value, then the application determines whether the return code is a standard code or not, and if the return code is a standard code, then it is determined that the first cache data is not expired; and if the return code is a non-standard code, it is determined that the first cache data is expired and the application sends a data request for downloading the data from the server.

[0034] As an example, if the server determines that the cache data is not expired, the server returns a status code, e.g., "304", instead of the whole data to the APP, thus reducing the amount of data transmission.

[0035] The application displays information relevant to the first cache data or to the second cache data on the interface of the application and notifies the information to the user. When it is determined that the cache data is the newest, the data downloading process all ends. And the local APP will load data according to the frequency of usage by the user to make the transition of the interface more smooth and fluent.

[0036] An example of the present disclosure also provides a system for managing cache data of a mobile terminal, so as to solve the technical problems of application software over-frequently accessing to the cache, searching cached files being slow, downloading too much data, and wasting network traffic.

[0037] The mobile terminal 10 includes memory; one or more processors; and one or more programs stored in the memory and configured for execution by the one or more processors, the one or more programs including instructions to: determine whether first cache

data corresponding to an application exists; and if it is determined that the first cache data does not exist, send a data request to request for server data corresponding to the first cache data, and download the server data and form the first cache data by using the server data.

[0038] In the mobile terminal 10, the one or more programs further include instructions to: if it is determined that the first cache data exists, and if the first cache data is not expired according to expiration time of the first cache data, load the first cache data for the application.

[0039] In the mobile terminal 10, the one or more programs further include instructions to: if it is determined that the first cache data exists, and if the first cache data is expired according to expiration time of the first cache data, send the data request containing storage time of the first cache data.

[0040] In the mobile terminal 10, the one or more programs further include instructions to: if it is determined that the first cache data needs to be updated, download the server data; replace the first cache data with the server data in object stream to update the first cache data into the second cache data; compress the second cache data; and save the compressed second cache data.

[0041] In the mobile terminal 10, the one or more programs further include instructions to: if it is determined that the first cache data needs not to be updated, load the first cache data for the application.

[0042] The server 20 includes memory; one or more processors; and one or more programs stored in the memory and configured for execution by the one or more processors, the one or more programs including instructions to: if first cache data corresponding to an application does not exist; receive a data request that requests for server data corresponding to the application; and send the server data to form the first cache data.

[0043] In the server 20, the one or more programs further include instructions to: if the first cache data exists, and if the first cache data is expired according to expiration time of the first cache data, receive the data request containing storage time of the first cache data.

[0044] In the server 20, the one or more programs further include instructions to: compare update time of the server data and the storage time contained in the data request to determine whether the first cache data needs to be updated.

[0045] In the server 20, the one or more programs further include instructions to: if it is determined that the first cache data needs to be updated, send the server data to update the first cache data.

[0046] In the server 20, the one or more programs further include instructions to: if it is determined that the first cache data needs not to be updated, send a notification.

[0047] Fig.6 shows a structural diagram of a system for managing cache data of the mobile terminal according to an example of the present disclosure.

[0048] As shown in Fig.6, the mobile terminal 10 includes: a network connection module 1, to establish a network connection between the server and the mobile terminal; a starting module 2, to start an application on the mobile terminal; a checking module 3, to check whether the first cache data corresponding to the application exists in the cache of the mobile terminal; a first data requesting module 4, to send a data request to the server for downloading data; a data downloading module 5, to download the server data corresponding to the application from the server to the cache of the mobile terminal; a second data request module 6, to send the data request corresponding to the first cache data to the server; and a data updating module 7, to download the server data corresponding to the application from the server to the cache of the mobile terminal so as to update the first cache data into the second cache data.

[0049] In the above, if the first cache data is not in the cache of the mobile terminal, the mobile terminal 10 downloads the server data to the cache thereof to form the first cache data; and if the first cache data has existed in the cache of the mobile terminal 10 and the first cache data has not expired, the server 20 returns prompt information to the mobile terminal 10 to prompt that the requested data has existed in the cache of the mobile terminal 10; and if the first cache data exists in the cache of the mobile terminal 10 and the first cache data has expired, the mobile terminal 10 downloads the server data from the server 20 to the cache of the mobile terminal 10 to update the first cache data into the second cache data.

[0050] The device for managing the cache data of the mobile terminal also includes a determination module 8 at the server for determining whether the first cache data is expired.

[0051] In the above, the determination module 8 includes: a comparison module 81 to compare the storage time of the first cache data and the update time of the server data to get a comparison result; and a sending module 82 to send the comparison result to the mobile terminal.

[0052] In the above, the data downloading module 5 includes: a first downloading module 51 to download the server data from the server; a writing module 52 to write the server data in the form of object stream into the cache of the mobile terminal so as to form the first cache data; and a first compression module 53 to compress the first cache data; and a first storage module 54 to save the compressed first cache data to the cache of the mobile terminal.

[0053] In the above, the data updating module 7 includes: a second downloading module 71 to download the server data from the server; an update module 72 to cover the first cache data by using the server data in the form of object stream to update the first cache data into the second cache data; a second compression module 73 to compress the second cache data; and a second storage module 74 to save the compressed second cache data to the cache of the mobile terminal.

[0054] The advantages of the present disclosure are that, in an environment where the network condition is good, the local cache can be used according to the cache situations thereof and the expiration time returned from the server; since in some conditions, there is no need to re-download the server data, which reduces waste of network traffic; the cache file stored locally is in the form of object stream instead of the original file, which avoids a second analysis, improves the loading speed, enhances the user's experience; the cache data of the mobile terminal takes precedence according to the frequency of usage by the user, which also improves the loading speed; the cache data is highly efficiently compressed, which reduces the storage space occupied by the cache data in the mobile terminal .

[0055] The foregoing description, for purpose of explanation, has been described with reference to specific examples. However, the illustrative discussions above are not intended

to be exhaustive or to limit the present disclosure to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The examples were chosen and described in order to best explain the principles of the present disclosure and its practical applications, to thereby enable others skilled in the art to best utilize the present disclosure and various examples with various modifications as are suited to the particular use contemplated.

[0056] The above examples may be implemented by hardware, software, firmware, or a combination thereof. For example the various methods, processes and functional modules described herein may be implemented by a processor (the term processor is to be interpreted broadly to include a CPU, processing unit/module, ASIC, logic module, or programmable gate array, etc.). The processes, methods and functional modules may all be performed by a single processor or split between several processors; reference in this disclosure or the claims to a 'processor' should thus be interpreted to mean 'one or more processors'. The processes, methods and functional modules are implemented as machine readable instructions executable by one or more processors, hardware logic circuitry of the one or more processors or a combination thereof. The modules, if mentioned in the aforesaid examples, may be combined into one module or further divided into a plurality of sub-modules. Further, the examples disclosed herein may be implemented in the form of a software product. The computer software product is stored in a non-transitory storage medium and includes a plurality of instructions for making an electronic device implement the method recited in the examples of the present disclosure.

WHAT IS CLAIMED IS:

1. A method for managing cache data, comprising:
determining whether first cache data corresponding to an application exists; and
if it is determined that the first cache data does not exist,
 sending a data request to request for server data corresponding to the
 first cache data; and
 downloading the server data and forming the first cache data by using
 the downloaded server data.
2. The method according to claim 1, further comprising:
if it is determined that the first cache data exists and if the first cache data is not
 expired according to expiration time of the first cache data, loading the first
 cache data for the application.
3. The method according to claim 1, further comprising:
if it is determined that the first cache data exists, and if the first cache data is expired
 according to expiration time of the first cache data,
 sending the data request containing storage time of the first cache data;
 determining whether the first cache data needs to be updated,
 if it is determined that the first cache data needs to be updated,
 updating the first cache data into second cache data, or
 if it is determined that the first cache data needs not to be updated,
 loading the first cache data for the application.
4. The method according to claim 3, wherein determining whether the first cache data
needs to be updated comprises:
 comparing update time of the server data and the storage time contained in the data
 request.
5. The method according to claim 3 or 4, wherein updating the first cache data into the
second cache data comprises,
 downloading the server data;

replacing the first cache data with the server data in object stream to update the first cache data into the second cache data;
compressing the second cache data; and
saving the compressed second cache data.

6. The method according to claim 5, comprising if it is determined that the first cache data needs not to be updated,
providing the determination result.

7. A mobile terminal, comprising
memory;
one or more processors; and
one or more programs stored in the memory and configured for execution by the one or more processors, the one or more programs including instructions to:
determine whether first cache data corresponding to an application exists; and
if it is determined that the first cache data does not exist,
send a data request to request for server data corresponding to the first cache data, and
download the server data and form the first cache data by using the server data.

8. The mobile terminal according to claim 7, wherein the one or more programs include instructions to:

if it is determined that the first cache data exists, and if the first cache data is not expired according to expiration time of the first cache data, load the first cache data for the application.

9. The mobile terminal according to claim 7, wherein the one or more programs include instructions to:

if it is determined that the first cache data exists, and if the first cache data is expired according to expiration time of the first cache data,
send the data request containing storage time of the first cache data.

10. The mobile terminal according to claim 9, wherein the one or more programs include instructions to: if it is determined that the first cache data needs to be updated, download the server data;
replace the first cache data with the server data in object stream to update the first cache data into the second cache data;
compress the second cache data; and
save the compressed second cache data.

11. The mobile terminal according to claim 8, wherein the one or more programs includes instructions to: if it is determined that the first cache data needs not to be updated, load the first cache data for the application.

12. A server, comprising
memory;
one or more processors; and
one or more programs stored in the memory and configured for execution by the one or more processors, the one or more programs including instructions to:
if first cache data corresponding to an application does not exist;
receive a data request that requests for server data corresponding to the application; and
send the server data to form the first cache data.

13. The server according to claim 12, wherein the one or more programs include instructions to:

if the first cache data exists, and if the first cache data is expired according to expiration time of the first cache data, receive the data request containing storage time of the first cache data.

14. The server according to claim 12, wherein the one or more programs include instructions to:

compare update time of the server data and the storage time contained in the data request to determine whether the first cache data needs to be updated.

15. The server according to claim 14, wherein the one or more programs include instructions to:

if it is determined that the first cache data needs to be updated, send the server data to update the first cache data.

16. The server according to claim 14, wherein the one or more programs include instructions to:

if it is determined that the first cache data needs not to be updated, send a notification.

1/4

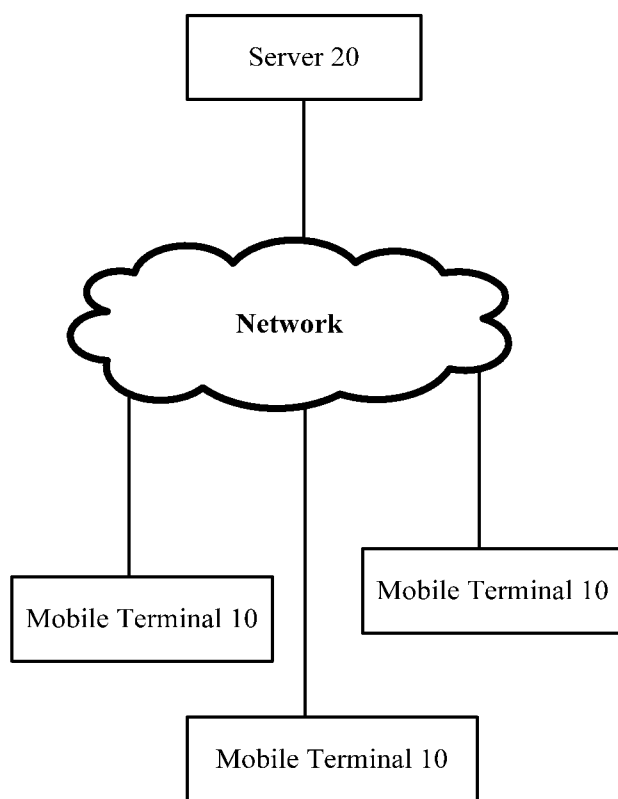


FIG. 1

2/4

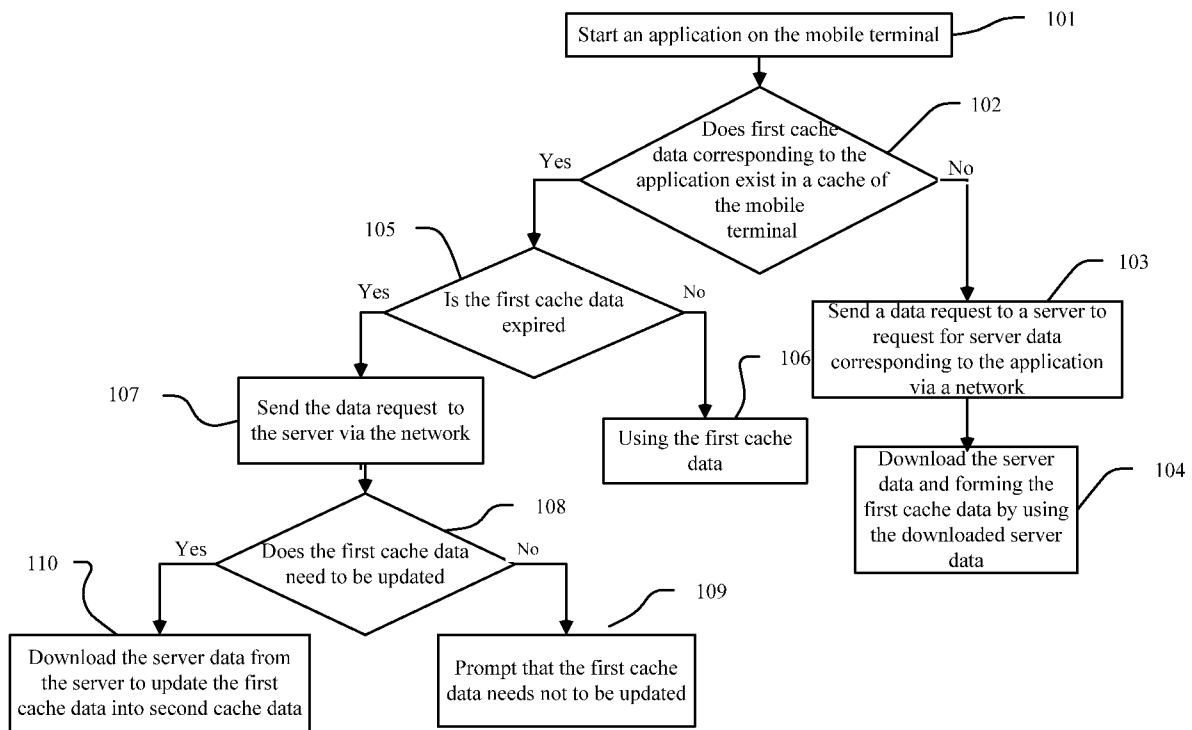


FIG. 2

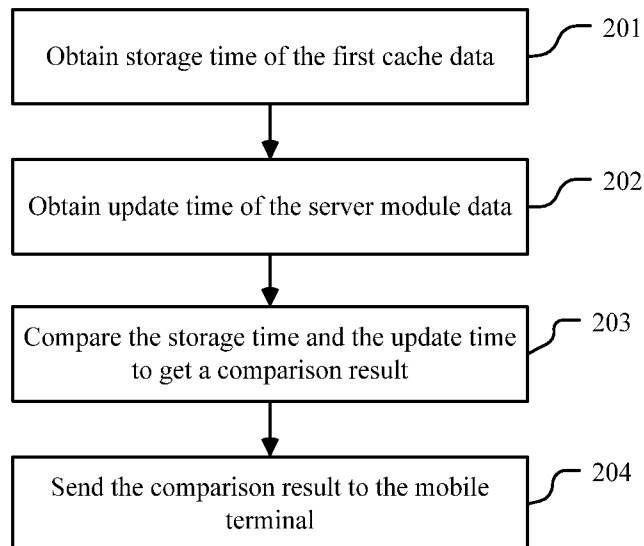


FIG. 3

3/4

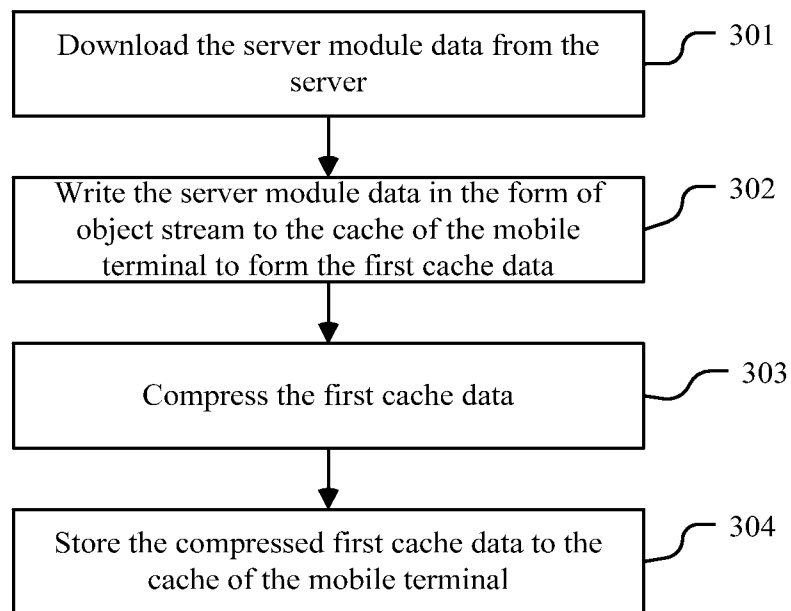


FIG. 4

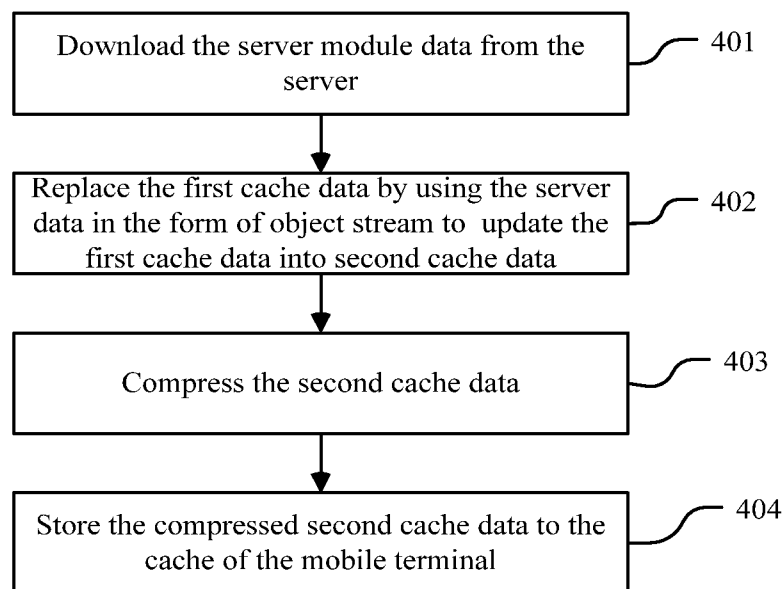


FIG. 5

4/4

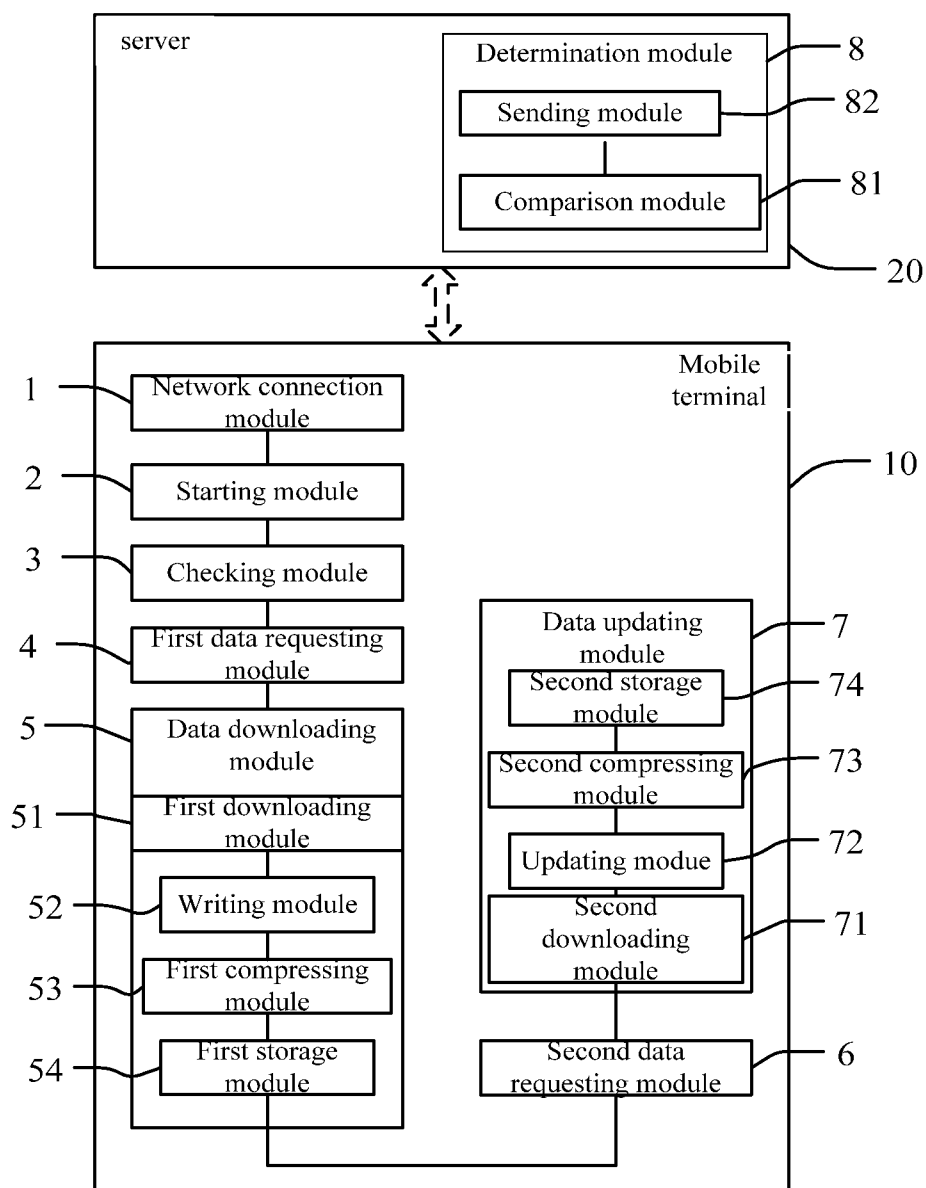


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2013/090046

A. CLASSIFICATION OF SUBJECT MATTER

H04W 8/24 (2009.01) i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04W, H04Q, H04L, G06F

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)

CNABS, CNTXT, CJFD, CNKI, VEN, USTXT, JPTXT, EPTXT, WOTXT, 3GPP FTP, IEEE: cache, buffer?, application?, software, program??. data, server?, download???, updat???, expir+, exist?

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 101605186 A (ZTE CORPORATION) 16 December 2009 (16.12.2009) description, page 3, line 14 to page 4, line 4, claims 1-4, figure 2 and the abstract	1-16
X	CN 101358851 A (BEIJING LINGTU SOFTWARE CO LTD) 04 February 2009 (04.02.2009) description, page 6, line 8 to page 8, line 9, claims 1-10, figures 1-4 and the abstract	1-16
X	CN 1525318 A (FOXCONN HONGFUJIN PRECISION IND SHENZHEN et al.) 01 September 2004 (01.09.2004) description, page 5, line 15 to page 7, line 21, claims 1-10, figures 5-6 and the abstract	1-16
X	CN 101719073 A (LANGCHAO GROUP SHANDONG GENERSOFT CO LTD) 02 June 2010 (02.06.2010) description, paragraphs [0027]-[0030], [0118]-[0128], claims 1-7, figure 10 and the abstract	1-16
A	US 2005154759 A1 (MICROSOFT CORPORATION) 14 July 2005 (14.07.2005) the whole document	1-16

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 13 March 2014 (13.03.2014)	Date of mailing of the international search report 27 Mar. 2014 (27.03.2014)
---	--

Name and mailing address of the ISA/CN
The State Intellectual Property Office, the P.R.China
6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China
100088
Facsimile No. 86-10-62019451

Authorized officer

LV, Sihua

Telephone No. (86-10)62089473

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2013/090046

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 101605186 A	16.12.2009	CN 101605186 B	26.10.2011
CN 101358851 A	04.02.2009	None	
CN 1525318 A	01.09.2004	TW 200411559 A	01.07.2004
		TW I317503 B	21.11.2009
		CN 1525318 B	05.05.2010
CN 101719073 A	02.06.2010	CN 101719073 B	03.10.2012
US 2005154759 A1	14.07.2005	EP 1553492 A3	31.10.2007
		EP 1553492 A2	13.07.2005
		KR 100999267 B1	07.12.2010
		JP 4404781 B2	27.01.2010
		JP 2005251167 A	15.09.2005
		CN 1649300 B	12.06.2013
		CN 1649300 A	03.08.2005
		KR 20050073552 A	14.07.2005
		IN 200500045 I1	08.12.2006
		US 7257583 B2	14.08.2007