

## **DATA PROCESSING METHOD AND APPARATUS, AND EDGE DEVICE**

### **Technical Field**

The present disclosure relates to the technical field of edge computation, and in particular, to a data processing method and apparatus, and an edge device.

### **Background**

Due to the continuous extension of Internet of Things service and the explosive growth of Internet of Things terminal data, in order to implement real-time processing of data, some data is distributed to an edge device close to a data source for processing, and some other important data still needs to be transmitted back to a cloud. In order to ensure that this part of data can be reliably transmitted back to the cloud, a data processing manner in an abnormal situation such as network disconnection and a power failure need to be improved.

### **Summary**

The present disclosure provides a data processing method and apparatus, and an edge device.

Specifically, the present disclosure is implemented by means of the following technical solutions.

According to a first aspect of embodiments of the present disclosure, a data processing method is provided, and the method includes:

writing a currently received message to a first area of a local disk;

after writing the currently received message to the first area of the local disk, buffering the currently received message to a cache; and

after a device comprising the local disk and the cache is restarted, restoring data in the cache according to data stored in the first area of the local disk.

In some embodiments, the writing a currently received message to a first area of a local disk includes:

writing the currently received message to a WAL log.

In some embodiments, the method further includes:

reading data of a message numbered  $k+1$  from the cache, where  $k$  is the number of the last message successfully sent to a cloud device, the number of a message is determined according to

the order in which the message is received, and  $k$  is a positive integer; and

sending the read data to the cloud device.

In some embodiments, the method further includes:

when a preset condition is satisfied, saving the data in the cache to a second area of the local disk and emptying the cache.

In some embodiments, the saving the data in the cache to a second area of the local disk includes:

compressing the data in the cache, and saving the compressed data to the second area of the local disk.

In some embodiments, the preset condition includes at least one of the following conditions:

the occupied space of the cache reaches a set value of the cache space; and

a time interval between the current time and the time when the cache was emptied last time reaches a preset time threshold.

In some embodiments, after emptying the cache, the method further includes:

writing a message received after emptying the cache this time to the first area of the local disk to overwrite the data stored in the first area of the local disk before emptying the cache this time; and

after writing the message received after emptying the cache this time to the first area of the local disk, buffering the message received after emptying the cache this time to the cache.

In some embodiments, after saving the data in the cache to a second area of the local disk and emptying the cache, the method further includes:

when a message numbered  $m+1$  is stored in the cache, reading data of the message numbered  $m+1$  from the cache, where  $m$  is the number of the last message successfully sent to the cloud device, the number of a message is determined according to the order in which the message is received, and  $m$  is a positive integer; and

sending the read data to the cloud device.

In some embodiments, the method further includes:

when no message numbered  $m+1$  is stored in the cache, reading the data of the message numbered  $m+1$  from the second area of the local disk; and

sending the read data to the cloud device.

According to a second aspect of the embodiments of the present disclosure, a data processing apparatus is provided, and the apparatus includes:

a first writing module, configured to write a currently received message to a first area of a local disk;

a first buffering module, configured to, after writing the currently received message to the first area of the local disk, buffer the currently received message to a cache; and

a data restoring module, configured to, after a device comprising the local disk and the cache is restarted, restore data in the cache according to data stored in the first area of the local disk.

In some embodiments, the first writing module is configured to:

write the currently received message to a WAL log.

In some embodiments, the apparatus further includes:

a first reading module, configured to read data of a message numbered  $k+1$  from the cache, where  $k$  is the number of the last message successfully sent to a cloud device, the number of a message is determined according to the order in which the message is received, and  $k$  is a positive integer; and

a first sending module, configured to send the read data to the cloud device.

In some embodiments, the apparatus further includes:

a saving module, configured to, when a preset condition is satisfied, save the data in the cache to a second area of the local disk and empty the cache.

In some embodiments, the saving module is configured to:

compress the data in the cache, and save the compressed data to the second area of the local disk.

In some embodiments, the preset condition includes at least one of the following conditions:

the occupied space of the cache reaches a set value of the cache space; and

a time interval between the current time and the time when the cache was emptied last time reaches a preset time threshold.

In some embodiments, the apparatus further includes:

a second writing module, configured to write a message received after emptying the cache this

time to the first area of the local disk to overwrite the data stored in the first area of the local disk before emptying the cache this time; and

a second buffering module, configured to, after writing the message received after emptying the cache this time to the first area of the local disk, buffer the message received after emptying the cache this time to the cache.

In some embodiments, the apparatus further includes:

a second reading module, configured to, when a message numbered  $m+1$  is stored in the cache, read data of the message numbered  $m+1$  from the cache, where  $m$  is the number of the last message successfully sent to the cloud device, the number of a message is determined according to the order in which the message is received, and  $m$  is a positive integer; and

a second sending module, configured to send the read data to the cloud device.

In some embodiments, the apparatus further includes:

a third reading module, configured to, when no message numbered  $m+1$  is stored in the cache, read the data of the message numbered  $m+1$  from the second area of the local disk; and

a third sending module, configured to send the read data to the cloud device.

According to a third aspect of the embodiments of the present disclosure, a computer readable storage medium is provided, having a computer program stored thereon, where when the program is executed by a processor, the method according to any one of the embodiments is implemented.

According to a fourth aspect of the embodiments of the present disclosure, an edge device is provided, and includes a memory, a processor, and a computer program stored on the memory and runnable on the processor, where when the processor executes the program, the method according to any one of the embodiments is implemented.

According to the embodiments of the present disclosure, a currently received message is written to a first area of a local disk, and after the currently received message is written to the first area of the local disk, the currently received message is buffered to a cache. When an edge device crashes abnormally and a service is restarted, data in the cache may be restored according to data stored in the first area of the local disk, thereby ensuring that data of a client received by the edge device is not lost.

It should be understood that the above general description and the following detailed description are merely exemplary and explanatory, and are not intended to limit the present disclosure.

### **Brief Description of the Drawings**

The drawings here incorporated in the description and constituting a part of the description describe embodiments of the present disclosure and are intended to explain the technical solutions of the present disclosure together with the description.

FIG. 1 is a schematic diagram of an application scenario of the embodiments of the present disclosure.

FIG. 2 is a flow chart of a data processing method of the embodiments of the present disclosure.

FIG. 3 is a schematic diagram of a data transmission process between an edge device and a cloud device of the embodiments of the present disclosure.

FIG. 4 is a block diagram of a data processing apparatus of the embodiments of present disclosure.

FIG. 5 is a block diagram of a computer device for implementing a method of the present disclosure according to the embodiments of the present disclosure.

### **Detailed Descriptions**

Exemplary embodiments are explained in detail here, examples of which are illustrated in the accompanying drawings. When the following description refers to the accompanying drawings, the same numerals in different drawings represent the same or similar elements unless otherwise indicated. The implementation modes described in the following exemplary embodiments do not represent all implementation modes consistent with the present application. On the contrary, they are merely examples of the apparatus and method consistent with some aspects of the present application as detailed in the appended claims.

The terms used in the present application is for the purpose of describing particular embodiments only and is not intended to limit present application. The singular forms "a", "said", and "the" as used in the present application and the appended claims are intended to include the plural forms as well, unless the context clearly indicates something else. It should also be understood that the term "and/or" as used herein refers to and includes any or all possible combinations of one or more of the associated listed items. In addition, the term "at least one" herein means any one of a plurality or any combination of at least two of the plurality.

It should also be understood that although the terms such as first, second, and third may be used in the present application to describe various information, such information should not be

limited to these terms. These terms are only used to distinguish the same type of information from each other. For example, without departing from the scope of the present application, the first information may also be referred to as the second information, and similarly, the second information may also be referred to as the first information. Depending on the context, the word "if" as used herein may be interpreted as "at the time of ..." or "when ..." or "in response to determining".

In order to enable a person skilled in the art to better understand the technical solutions in the embodiments of the present disclosure, and to make the above-mentioned purposes, features, and advantages of the embodiments of the present disclosure more obvious and understandable, the technical solutions in the embodiments of the present disclosure are further explained in detail below in combination with the accompanying drawings.

Edge computation refers to that a nearest-end service is provided nearby on the side close to an object or a data source by using an open platform with core capabilities of a network, computing, storage, and application. Its application program is initiated at an edge side, thereby producing a faster network service response, and satisfying the basic demands in the industry in real-time service, application intelligence, security, privacy protection, and the like. Internet of Things is one of the important applications of edge computation technology. In an application scenario of Internet of Things, in order to implement real-time processing of data, some data is distributed to an edge device close to a data source for processing, and some other important data still needs to be transmitted back to a cloud.

FIG. 1 is a schematic diagram of an application scenario of the embodiments of the present disclosure. It can be understood that the application scenario is merely one possible application scenario of the embodiments of the present disclosure, and is not intended to limit the present disclosure. An edge device may be a device such as a vehicle-mounted server, a monitoring system, a notebook computer, and a mobile phone, and the edge device may be connected to the cloud by means of a network. For example, a vehicle-mounted server may process road information and vehicle information acquired by various vehicle sensors, so that assistance driving and automatic driving are implemented. For another example, a monitoring system may obtain a monitoring object acquired by a monitoring camera, so as to track a monitoring target. Because these data requires high real-time, an edge device close to the data source may be used for processing these data. For some more complicated and important data integration, data analysis, and other processes, data needs to be sent from the edge device to the cloud device, and then processed by the cloud device.

However, the network environment is more complicated in an Internet of Things scenario, network disconnection often occurs, and sometimes, the network disconnection lasts for a long time, as a result, some important data may not be reliably transmitted back to the cloud. In this regard, the embodiments of the present disclosure provide a data processing method. As shown in FIG. 2, the method may include the following steps.

Step S201: a currently received message is written to a first area of a local disk.

Step S202: after writing the currently received message to the first area of the local disk, the currently received message is buffered to a cache.

Step S203: after a device comprising the local disk and the cache is restarted, data in the cache is restored according to data stored in the first area of the local disk.

In step S201, the currently received message may be various types of messages such as a control message, a request message, or a response message. Each time an edge device receives a message from a client, the edge device assigns a number to the message, and the number is strictly and progressively increased according to the time when the message is received. That is to say, each message has a number, i.e., seqID (Sequence Identifier) for uniquely identifying the message. The seqID is strictly and progressively increased. For example, if a message of which the seqID is  $n$  is received at time  $T1$ , the seqID of the next message received is  $n+1$ , the seqID of the still next message received is  $n+2$ , and so on. Each time a message is received, the message may be written to the first area of the local disk to implement persistence processing of the message.

Persistence is a mechanism that converts program data between a persistent state and a transient state. Generally, transient data that cannot be saved permanently (such as data in the cache) is persisted into persistent data that can be saved for a long time (such as data in local disk). By means of persistence, data would not be lost in an abnormal situation such as a power failure and network disconnection.

In some embodiments, the currently received message may be written into a WAL log. The WAL refers to Write-Ahead Logging, i.e., a write-ahead logging system. Writing the currently received messages to the WAL log is for backing up the received message. After the edge device is restarted due to a power failure or other reasons, the currently received message may be read from the WAL, and then the read currently received message is buffered to the cache, so as to ensure that data is not lost.

In step S202, because the currently received message is written into the first area of the local disk, and then the currently received message is buffered to the cache, the currently received

message in the cache may be found in the local disk. When an abnormal situation such as a power failure or restarting occurs, the corresponding message may be found from the first area of the local disk, and the data in the cache is restored, thereby ensuring that the data in the cache is not lost. When the edge device works normally, the data may be read directly from the cache.

In some embodiments, data of a message numbered  $k+1$  may be read from the cache, and the read data is sent to a cloud device, where  $k$  is the number of the last message successfully sent to the cloud device, the number of a message is determined according to the order in which the message is received, and  $k$  is a positive integer. When an abnormal situation such as network disconnection occurs between the edge device and the cloud device in the process of sending the message numbered  $k+1$ , after the network is recovered, data is read from the cache and re-sent to the cloud device, so as to ensure that messages, which need to be sent to the cloud device, in the received data are not lost.

In some embodiments, after the currently received message is buffered to the cache, data compression processing may also be performed on the currently received message in the cache. By means of data compression, the amount of data in the cache may be reduced and the utilization of the cache may be improved.

In some embodiments, if a preset condition is satisfied, the data in the cache may be saved to a second area of the local disk, and the cache is emptied. In this embodiment, when a preset condition is satisfied, all messages in the cache may be persistently processed to a local disk, and the cache is emptied. If the current cache includes message 1, message 2, ..., and message  $n$ , message 1, message 2, ..., and message  $n$  may all be saved to the second area of the local disk, and the cache is emptied.

The preset condition may include at least one of the following: the occupied space of the cache reaches a set value of the cache space; and a time interval between the current time and the time when the cache was emptied last time reaches a preset time threshold. Because the cache capacity is limited, emptying the cache according to the preset conditions may avoid data loss due to cache overflow. Moreover, the messages in the cache are persisted to the second area of the local disk, which may back up the data in the cache and prevent data loss.

Moreover, in order to reduce the amount of data in the local disk and improve the utilization of the local disk, the data in the cache may be compressed, and the compressed data is saved to the second area of the local disk.

Each time a message in the cache is saved to the second area of the local disk, a file may be

generated in the second area of the local disk, and the message in the cache is stored in the file. The file may be named according to the seqID of the first message in the cache. For example, at time T1, messages with the seqIDs from 000000 to 000199 are buffered in the cache, then the 100 messages may be stored in a file in the local disk, and the file is named 000000. At time T2 (time T2 is after time T1), messages with the seqIDs from 000200 to 000399 are buffered in the cache, then the 100 messages may be stored in another file in the local disk, and the file is named 000200.

The currently received message may be searched for according to the file name of the file in the local disk used for persisting the cache messages. During search, the file names of files may be sorted, and then according to the seqID of the currently received message and the starting seqID corresponding to each file name, the file where the seqID of the currently received message is located is searched for by means of dichotomy.

In some embodiments, after the cache is emptied, a message received after emptying the cache this time may also be written to the first area of the local disk to overwrite the data stored in the first area of the local disk before emptying the cache this time; and after the message received after emptying the cache this time is written to the first area of the local disk, the message received after emptying the cache this time is buffered to the cache. In this way, the messages in the first area of the local disk may be kept consistent with the messages in the cache. Because the function of the first area of the local disk is to back up the cache, the first area of the local disk only needs to save the latest message.

By using a new message received after emptying the cache to overwrite the original data in the first area of the local disk, the amount of data may be reduced. In some embodiments, a WAL log may be newly created in the first area of the local disk, and the file name of the newly created WAL log may be the same as the file name of the original WAL log, so as to overwrite and delete the original WAL file.

After the data in the cache is saved to the second area of the local disk and the cache is emptied, and when a message numbered  $m+1$  is stored in the cache, data of the message numbered  $m+1$  may be read from the cache, and the read data is sent to the cloud device, where  $m$  is the number of the last message successfully sent to the cloud device, the number of a message is determined according to the order in which the message is received, and  $m$  is a positive integer.

Furthermore, when no message numbered  $m+1$  is stored in the cache, the data of the message numbered  $m+1$  is read from the second area of the local disk; and the read data is sent to the cloud device.

In practical application, currently received messages are continuously sent to the cache. In a normal (the edge device is online) situation, the currently received message may be directly read from the cache and sent to the cloud device. If network disconnection occurs, but the network disconnection time is relatively short, messages generated during the network disconnection may be stored in the cache without overflowing. Therefore, after the network is recovered, the next message of the last message successfully sent to the cloud device may still be read from the cache and sent to the cloud device, so as to ensure that messages, which need to be sent to the cloud device, in the received data are not lost.

If the network disconnection lasts for a long time, not all messages generated during the network disconnection can be stored in the cache. In order to avoid data loss, the messages in cache are maintained in the second area of the local disk. In this way, after the network is recovered, when the next message of the last message successfully sent to the cloud device cannot be read from the cache, the next message of the last message successfully sent to the cloud device may be read from the second area on the local disk and sent to the cloud device, so as to ensure that messages, which need to be sent to the cloud device, in the received data are not lost.

The embodiments of the present disclosure are applicable to an Internet of Things scenario, and may improve the reliability of transmission of important data. When the resources of the cache and the local disk are limited, messages are persisted to the local disk to ensure long-term data retention. According to the embodiments of the present disclosure, not only can the reliability of data transmission be ensured when offline for a long time, but also data can be directly read from the cache when offline for a short time, without losing the data transmission performance.

A schematic diagram of a data transmission process between an edge device and a cloud device of the embodiments of the present disclosure is as shown in FIG. 3. At the edge device, a message generated by a data source may be stored in the WAL log file to be persisted to the local disk; after the persistence is successful, the message is stored in the cache; and when the cache is full, the data in the cache is compressed and stored in the local disk, and the cache is emptied. A sender in the edge device may search for a message from the cache and send same to the cloud device. When the edge device is normal or offline for a short time, the corresponding message may be found in the cache, but when the edge device is offline for a long time, the corresponding message cannot be found in the cache. Therefore, the sender searches for the corresponding message in the local disk and sends same to the cloud device.

The method of the embodiments of the present disclosure may be implemented by an edge device, where the edge device may be a device such as a vehicle-mounted server, a monitoring

system, a notebook computer, and a mobile phone in the application scenario shown in FIG. 1. By means of the embodiments of the present disclosure, data loss may be prevented when the edge device is abnormally powered off or the network connection between the edge device and the cloud device is interrupted.

A person skilled in the art can understand that, in the foregoing methods of the specific implementation modes, the order in which the steps are written does not imply a strict execution order which constitutes any limitation to the implementation process, and the specific order of executing the steps should be determined by functions and possible internal logics thereof.

As shown in FIG. 4, the present disclosure further provides a data processing apparatus, and the apparatus includes:

- a first writing module 401, configured to write a currently received message to a first area of a local disk;

- a first buffering module 402, configured to, after writing the currently received message to the first area of the local disk, buffer the currently received message to a cache; and

- a data restoring module 403, configured to, after a device comprising the local disk and the cache is restarted, restore data in the cache according to data stored in the first area of the local disk.

In some embodiments, the first writing module is configured to:

- write the currently received message to a WAL log.

In some embodiments, the apparatus further includes:

- a first reading module, configured to read data of a message numbered  $k+1$  from the cache, where  $k$  is the number of the last message successfully sent to a cloud device, the number of a message is determined according to the order in which the message is received, and  $k$  is a positive integer; and

- a first sending module, configured to send the read data to the cloud device.

In some embodiments, the apparatus further includes:

- a saving module, configured to, when a preset condition is satisfied, save the data in the cache to a second area of the local disk and empty the cache.

In some embodiments, the saving module is configured to:

- compress the data in the cache, and save the compressed data to the second area of the local

disk.

In some embodiments, the preset condition includes at least one of the following conditions:

the occupied space of the cache reaches a set value of the cache space; and

a time interval between the current time and the time when the cache was emptied last time reaches a preset time threshold.

In some embodiments, the apparatus further includes:

a second writing module, configured to write a message received after emptying the cache this time to the first area of the local disk to overwrite the data stored in the first area of the local disk before emptying the cache this time; and

a second buffering module, configured to, after writing the message received after emptying the cache this time to the first area of the local disk, buffer the message received after emptying the cache this time to the cache.

In some embodiments, the apparatus further includes:

a second reading module, configured to, when a message numbered  $m+1$  is stored in the cache, read data of the message numbered  $m+1$  from the cache, where  $m$  is the number of the last message successfully sent to the cloud device, the number of a message is determined according to the order in which the message is received, and  $m$  is a positive integer; and

a second sending module, configured to send the read data to the cloud device.

In some embodiments, the apparatus further includes:

a third reading module, configured to, when no message numbered  $m+1$  is stored in the cache, read the data of the message numbered  $m+1$  from the second area of the local disk; and

a third sending module, configured to send the read data to the cloud device.

In some embodiments, the functions provided by or the modules included in the apparatus provided by the embodiments of the present disclosure may be used for implementing the method described in the foregoing method embodiments. For specific implementations, reference may be made to the description in the method embodiments above. For the purpose of brevity, details are not described herein again.

The foregoing described apparatus embodiments are merely schematic. The modules described as separate parts may or may not be physically separate, and the parts displayed as modules may or may not be physical modules, may be located in one position, or may be

distributed on a plurality of network modules. Some or all of the modules may be selected according to actual needs to implement the purpose of the solutions in the present description. A person of ordinary skill in the art may understand and implement same without involving an inventive effort.

The apparatus embodiments of the present description may be applied to a computer device, such as a server or a terminal device. The apparatus embodiments may be implemented by software, or implemented by hardware or a combination of software and hardware. Taking software implementation as an example, as an apparatus in a logical sense, it is formed by reading a corresponding computer program instruction from a non-volatile memory to a cache and running same by means of the processor for file processing of the apparatus. For hardware, FIG. 5 shows a hardware structural diagram of a computer device in which the apparatus is located according to the present description. In addition to a processor 501, a cache 502, a network interface 503, and a non-volatile memory 504 shown in FIG. 5, the server or electronic device in which the apparatus of the embodiments is located may further include other hardware generally according to the actual functions of the computer device. Details are not described herein again.

Accordingly, the embodiments of the present disclosure further provide a computer storage medium having a computer program stored thereon. When the program is executed by a processor, the method according to any one of the embodiments is implemented.

Accordingly, the embodiments of the present disclosure further provide an edge device, including a memory, a processor, and a computer program stored on the memory and capable of running on the processor, where when the processor executes the program, the method according to any one of the embodiments is implemented.

In the present disclosure, a form of a computer program product implemented on one or more storage media (including, but not limited to, a disk memory, a Compact Disc Read-Only Memory (CD-ROM), an optical memory, etc.) including program codes may be used. A computer available storage medium includes permanent and non-permanent, movable and non-movable media, and information storage can be implemented by any method or technology. Information may be a computer readable command, a data structure, a module of a program or other data. Examples of the computer storage medium include, but are not limited to, a Phase Change Random Access Memory (PRAM), a Static Random Access Memory (SRAM), a Dynamic Random Access Memory (DRAM), other types of Random Access Memories (RAMs), a Read-Only Memory (ROM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a flash memory or other cache technologies, a CD-ROM, a Digital Video Disk (DVD) or other optical storage, and

a magnetic tape cartridge. Magnetic tape disk storage or other magnetic storage devices or any other non-transmission medium can be used for storing information that can be accessed by a computing device.

After considering the description and practicing the description disclosed herein, a person skilled in the art would easily conceive of other implementation solutions of the present disclosure. The present disclosure aims at including any variations, uses or adaptive changes of the present disclosure. These variations, uses or adaptive changes follow the general principles of the present disclosure and include common general knowledge or common technical means in the technical field that are not disclosed in the present disclosure. The description and embodiments are merely viewed as exemplars, and the true scope and spirit of the present disclosure are pointed out in the following claims.

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

The above descriptions are merely preferred embodiments of the present disclosure and are not intended to limit the present disclosure. Any modification, equivalent replacement, or improvement made within the spirit and principle of the present disclosure shall fall within the scope of protection of the present disclosure.

The descriptions of the embodiments above tend to focus on differences between the embodiments, and for same or similar parts in the embodiments, refer to these embodiments. For the purpose of brevity, details are not described again.

## CLAIMS

1. A data processing method, wherein the method comprises:
  - writing a currently received message to a first area of a local disk;
  - after writing the currently received message to the first area of the local disk, buffering the currently received message to a cache; and
  - after a device comprising the local disk and the cache is restarted, restoring data in the cache according to data stored in the first area of the local disk.
2. The method according to claim 1, wherein the writing a currently received message to a first area of a local disk comprises:
  - writing the currently received message to a WAL log.
3. The method according to claim 1 or 2, wherein the method further comprises:
  - reading data of a message numbered  $k+1$  from the cache, wherein  $k$  is the number of the last message successfully sent to a cloud device, the number of a message is determined according to the order in which the message is received, and  $k$  is a positive integer; and
  - sending the read data to the cloud device.
4. The method according to any one of claims 1 to 3, wherein the method further comprises:
  - when a preset condition is satisfied, saving the data in the cache to a second area of the local disk and emptying the cache.
5. The method according to claim 4, wherein the saving the data in the cache to a second area of the local disk comprises:
  - compressing the data in the cache, and saving the compressed data to the second area of the local disk.
6. The method according to claim 4 or 5, wherein the preset condition comprises at least one

of the following conditions:

the occupied space of the cache reaches a set value of the cache space; and

a time interval between the current time and the time when the cache was emptied last time reaches a preset time threshold.

7. The method according to claim 4 or 5, wherein after emptying the cache, the method further comprises:

writing a message received after emptying the cache this time to the first area of the local disk to overwrite the data stored in the first area of the local disk before emptying the cache this time; and

after writing the message received after emptying the cache this time to the first area of the local disk, buffering the message received after emptying the cache this time to the cache.

8. The method according to claim 7, wherein after saving the data in the cache to a second area of the local disk and emptying the cache, the method further comprises:

when a message numbered  $m+1$  is stored in the cache, reading data of the message numbered  $m+1$  from the cache, wherein  $m$  is the number of the last message successfully sent to the cloud device, the number of a message is determined according to the order in which the message is received, and  $m$  is a positive integer; and

sending the read data to the cloud device.

9. The method according to claim 8, wherein the method further comprises:

when no message numbered  $m+1$  is stored in the cache, reading the data of the message numbered  $m+1$  from the second area of the local disk; and

sending the read data to the cloud device.

10. A data processing apparatus, wherein the apparatus comprises:

a first writing module, configured to write a currently received message to a first area of a local disk;

a first buffering module, configured to, after writing the currently received message to the first area of the local disk, buffer the currently received message to a cache; and

a data restoring module, configured to, after a device comprising the local disk and the cache is restarted, restore data in the cache according to data stored in the first area of the local disk.

11. The apparatus according to claim 10, wherein the first writing module is configured to: write the currently received message to a WAL log.

12. The apparatus according to claim 10 or 11, wherein the apparatus further comprises:

a first reading module, configured to read data of a message numbered  $k+1$  from the cache, wherein  $k$  is the number of the last message successfully sent to a cloud device, the number of a message is determined according to the order in which the message is received, and  $k$  is a positive integer; and

a first sending module, configured to send the read data to the cloud device.

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

a saving module, configured to, when a preset condition is satisfied, save the data in the cache to a second area of the local disk and empty the cache.

14. The apparatus according to claim 13, wherein the saving module is configured to:

compress the data in the cache, and save the compressed data to the second area of the local disk.

15. The apparatus according to claim 13 or 14, wherein the preset condition comprises at least one of the following conditions:

the occupied space of the cache reaches a set value of the cache space; and

a time interval between the current time and the time when the cache was emptied last time reaches a preset time threshold.

16. The apparatus according to claim 13 or 14, wherein the apparatus further comprises:

a second writing module, configured to write a message received after emptying the cache this time to the first area of the local disk to overwrite the data stored in the first area of the local disk before emptying the cache this time; and

a second buffering module, configured to, after writing the message received after emptying the cache this time to the first area of the local disk, buffer the message received after emptying the cache this time to the cache.

17. The apparatus according to claim 16, wherein the apparatus further comprises:

a second reading module, configured to, when a message numbered  $m+1$  is stored in the cache, read data of the message numbered  $m+1$  from the cache, wherein  $m$  is the number of the last message successfully sent to the cloud device, the number of a message is determined according to the order in which the message is received, and  $m$  is a positive integer; and

a second sending module, configured to send the read data to the cloud device.

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

a third reading module, configured to, when no message numbered  $m+1$  is stored in the cache, read the data of the message numbered  $m+1$  from the second area of the local disk; and

a third sending module, configured to send the read data to the cloud device.

19. A computer readable storage medium, having a computer program stored thereon, wherein when the program is executed by a processor, the method according to any one of claims 1 to 9 is implemented.

20. An edge device, comprising a memory, a processor, and a computer program stored on the memory and runnable on the processor, wherein when the processor executes the program, the method according to any one of claims 1 to 9 is implemented.