

METHOD AND APPARATUS FOR PROCESSING DATA, AND DEVICE, STORAGE MEDIA AND PROGRAM PRODUCT THEREOF

TECHNICAL FIELD

[0001] The present disclosure relates to the technical field of edge computing, in particular related to a method and apparatus for processing data, and a device, a storage medium, and a program product thereof.

BACKGROUND

[0002] With the development of the Internet of things (IoT) technology, more and more devices are connected to the network, and massive data of IoT devices is uploaded to a cloud for processing, thereby increasing the burden of cloud data processing.

[0003] In the related art, an edge gateway may be deployed on an edge of IoT devices. The IoT devices upload their collected total data to the edge gateway, and the edge gateway processes the data collected by the IoT devices and then uploads the processed data to a cloud device.

[0004] However, each time an IoT device uploads data, the edge gateway triggers a data processing operation. Such frequent data processing imposes a stricter requirement on the hardware of the edge gateway. Consequently, in a case where the edge gateway has insufficient hardware capabilities, performance issues are likewise to arise.

SUMMARY

[0005] The present disclosure provides a method and apparatus for processing data, and a device, a storage medium, and a program product thereof. With the technical solutions, the frequency of computation of an edge gateway is reduced, without affecting the real-time performance of computation. The technical solution is as follows:

[0006] According to one aspect of the present disclosure, a method for processing data is provided. The method includes:

[0007] receiving a dataset from a first IoT device, wherein the dataset includes at least one piece of device data, each of the at least one piece of device data having a data identifier and a timestamp;

[0008] traversing the at least one piece of device data in the dataset in a chronological order of timestamps;

[0009] adding first device data in the dataset to the to-be-computed domain in a case where the first device data in the dataset is traversed and a to-be-computed domain of the edge gateway does not contain any device data having a same data identifier but a different timestamp as the first device data;

[0010] performing computation on each piece of device data in the to-be-computed domain in accordance with a predetermined computation rule in a case where second device data in the dataset is traversed and the to-be-computed domain of the edge gateway contains device data having a same data identifier but a different timestamp as the second device data.

[0011] According to one aspect of the present disclosure, an apparatus for processing data is provided. The apparatus includes:

[0012] a receiving module, configured to receive a dataset from a first IoT device, wherein the dataset includes at least one piece of device data, each of the at least one piece of device data having a data identifier and a timestamp;

[0013] a traversing module, configured to traverse the at least one piece of device data in the dataset in a chronological order of timestamps;

[0014] an adding module, configured to add first device data in the dataset to a to-be-computed domain in a case where the first device data in the dataset in the dataset is traversed and the to-be-computed domain of an edge gateway does not contain any device data having a same data identifier but a different timestamp as the first device data;

[0015] a computing module, configured to perform computation on each piece of device data in the to-be-computed domain in accordance with a predetermined computation rule in a case where second device data in the dataset is traversed and the to-be-computed domain of the edge gateway contains device data having a same data identifier but a different timestamp as the second device data.

[0016] In some embodiments, the computation rule includes data identifiers of at least two pieces of device data; wherein

[0017] the computing module is configured to traverse at least two pieces of device data in the to-be-computed domain in a reverse chronological order of timestamps in a case where second device data is traversed in the dataset and the to-be-computed domain of the edge gateway contains device data having a same data identifier but a different timestamp as the second device data;

[0018] the computing module is configured to associate a first computation rule with third device data in a case where the third device data in the to-be-computed domain is traversed, the third device data has a data identifier corresponding to the first computation rule, and the to-be-computed domain does not contain any device data associated with the first computation rule;

[0019] the computing module is configured to stop traversing the third device data in a case where the third device data is traversed, the third device data has the data identifier corresponding to the first computation rule, and the to-be-computed domain contains the device data associated with the first computation rule; and

[0020] the computing module is configured to perform computation on each piece of device data in the to-be-computed domain in accordance with a predetermined computation rule upon completion of traversing the at least two pieces of device data in the to-be-computed domain.

[0021] In some embodiments, the computing module is configured to traverse at least two pieces of device data in the to-be-computed domain in a chronological order of timestamps upon completion of traversing the at least two pieces of device data in the to-be-computed domain;

[0022] the computing module is configured to add fourth device data to a real-time library in a case where the fourth device data in the to-be-computed domain is traversed and the fourth device data is not associated with the computation rule; and

[0023] the computing module is configured to: in a case where fifth device data is traversed and the fifth device data is associated with a second computation rule, add the fifth device data to the real-time library, query at least two pieces of device data corresponding to the second computation rule from the real-time library, perform computation on the at least two pieces of device data corresponding to the second computation rule in accordance with the second computation rule to acquire a first computation data, set a timestamp of the fifth device data to a timestamp of the first computation data, and add the first computation data to the real-time library.

[0024] In some embodiments, the computing module is configured to query at least one piece of candidate device data corresponding to a first data identifier from the real-time library, wherein the first data identifier is any one of the data identifiers included in the second computation rule; and

[0025] the computing module is configured to acquire the candidate device data with a latest timestamp from at least one piece of candidate device data as the device data corresponding to the first data identifier from the at least two pieces of device data corresponding to the second computation rule.

[0026] In some embodiments, the apparatus further includes a clearing module, configured to clear the to-be-computed domain upon completion of traversing the at least two pieces of device data in the to-be-computed domain.

[0027] In some embodiments, the apparatus further includes an updating module, configured to move the second device data and other device data in the dataset with a timestamp subsequent to a timestamp of the second device data to the to-be-computed domain.

[0028] In some embodiments, a timer is provided in the edge gateway; and the apparatus further includes a timing module, configured to perform computation on each piece of device data in the to-be-computed domain in a case where the timer is expired;

[0029] where the timing module is further configured to reset the timer upon completion of performing the computation on the each piece of device data in the to-be-computed domain.

[0030] According to another aspect of the present disclosure, a computer device is provided. The computer device includes a processor and a memory, wherein the memory stores at least one computer instruction, wherein the processor, when loading and executing the at least one computer instruction, is caused to perform the method for processing data as described in the above aspect.

[0031] According to another aspect of the present disclosure, a computer-readable storage medium is provided. The readable storage medium stores at least one computer instruction, wherein the at least one computer instruction, when loaded and executed by a processor, causes the processor to perform the method for processing data as described in the above aspect.

[0032] According to another aspect of the present disclosure, a computer program product is provided. The computer program product includes at least one computer instruction, wherein the at least one computer instruction is stored in a computer-readable storage medium, and the at least one computer instruction, when read and executed by a processor from the computer-readable storage medium, causes the processor to perform the method for processing data as described in the above aspect.

[0033] The technical solutions according to the embodiments of the present disclosure may include the following beneficial effects:

[0034] The to-be-computed domain may be set up in the edge gateway. Subsequent to receiving the dataset from the IoT device, the edge gateway may compare the data identifier of the device data in the dataset with the data identifier of the device data in the to-be-computed domain. In a case where the data identifier of the second device data in the dataset is the same as the data identifier of certain device data in the to-be-computed domain, the device data with the timestamp earlier than the timestamp of the second device data in the dataset is added to the to-be-computed domain and computation is performed on each piece of device data in the to-be-computed domain. That is, the edge gateway may store the device data in the dataset from the IoT device in the to-be-computed domain in a chronological order of timestamps. Upon detection of the fact that the IoT device sends the data identifier corresponding to the data identifier of the device data in the to-be-computed domain, one computation is performed on the device data in the current to-be-computed domain, such that a situation in which two pieces of device data correspond to the same data identifier in the to-be-computed domain is avoided. Thus, under the premise of ensuring real-time and accurate computation, the present solution is able to reduce the computation frequency of the edge gateway, such that the resource consumption of the edge gateway is saved, thereby ensuring the device performance of the edge gateway.

BRIEF DESCRIPTION OF DRAWINGS

[0035] To describe the technical solutions in the embodiments of the present disclosure more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description illustrate merely some embodiments of the present disclosure,

and persons of ordinary skill in the art may still derive other drawings from these accompanying drawings without creative efforts.

[0036] FIG. 1 is a schematic diagram of an implementation environment according to an exemplary embodiment of the present disclosure;

[0037] FIG. 2 is a flowchart of a method for processing data according to an exemplary embodiment of the present disclosure;

[0038] FIG. 3 is a flowchart of a method for processing data according to another exemplary embodiment of the present disclosure;

[0039] FIG. 4 is a flowchart of an implementation of a method for processing data according to an exemplary embodiment of the present disclosure;

[0040] FIG. 5 is a schematic diagram of a method for processing data according to an exemplary embodiment of the present disclosure;

[0041] FIG. 6 is a schematic diagram of traversing device data according to a first embodiment of the present disclosure;

[0042] FIG. 7 is a schematic diagram of traversing device data according to a second embodiment of the present disclosure;

[0043] FIG. 8 is a block diagram of an apparatus for processing data illustrated in an exemplary embodiment of the present disclosure; and

[0044] FIG. 9 is a block diagram of a structure of a computer device according to an exemplary embodiment of the present disclosure.

[0045] The accompanying drawings herein, which are incorporated into and form a part of the specification, illustrate embodiments consistent with the present disclosure and, together with the description, serve to explain the principles of the present disclosure.

DESCRIPTION OF THE EMBODIMENTS

[0046] In order to make the purpose, technical solutions and advantages of the present disclosure clearer, the embodiments of the present disclosure are described in further detail below in conjunction with the accompanying drawings.

[0047] Exemplary embodiments are described herein in detail, examples of which are represented in the accompanying drawings. The following description refers to the accompanying drawings in which the same numerals in the different accompanying drawings represent the same or similar elements unless otherwise indicated. The

implementations described in the following exemplary embodiments do not represent all implementations consistent with the present disclosure. Instead, the implementations are merely examples of apparatuses and methods consistent with some aspects of the present disclosure as detailed in the appended claims.

[0048] The terms used in the present disclosure are used merely for the purpose of describing particular embodiments and are not intended to limit the present disclosure. The singular forms of "a," "said," and "the" used in the present disclosure and the appended claims are also intended to include the plural forms, unless the context clearly indicates other meanings.

[0049] It should be understood that while the terms "first," "second," and the like may be used in the present disclosure to describe various types of information, the information should not be limited to the terms. These terms are only used to distinguish the same type of information from one another. For example, without departing from the scope of the present disclosure, a first parameter may also be referred to as a second parameter, and similarly, a second parameter may be referred to as a first parameter. Depending on the context, as used herein, the phrase "if" may be interpreted as "at" or "when" or "in response to determining".

[0050] The following describes the interpretation of the terms according to the present disclosure.

[0051] IoT: The IoT is a network based on traditional communication networks, the Internet and other information carriers, such that all ordinary objects with independent functions to implement the interconnection and intercommunication. The IoT implements the interconnection of everything by combining sensors, smart devices, cloud computation, big data, artificial intelligence, and other information technologies.

[0052] An IoT system includes four main parts: a sensing layer, a network layer, a platform layer (or a data layer), and an application layer. The sensing layer is the foundation of the IoT system and is responsible for data acquisition. The network layer is responsible for transmitting data acquired in the sensing layer to the platform layer or the application layer. The platform layer (or the data layer) is the core of the IoT system and is responsible for storing, processing, and analyzing the data. The application layer is a user interface of the IoT system, and is responsible for presenting the data processed by the platform layer to a user in an easy-to-understand and use manner.

[0053] (2) Edge computing: On an edge of the network close to a data source, a distributed open platform that integrates core capability of network, computing, storage, and application is used to provide edge intelligent services nearby to satisfy the key needs of industry digitization in terms of agile connectivity, real-time business, data optimization, application intelligence, security and privacy protection, and the like.

[0054] Referring to FIG. 1, FIG. 1 illustrates a schematic diagram of an implementation environment according to an exemplary embodiment of the present disclosure. The implementation environment may involve an IoT device 10, an edge gateway 20, and a cloud device 30.

[0055] With the rapid development of the IoT technology, digital transformation is prevailing in all industries, such that more and more devices are connected to the network. This means that the data of the massive IoT devices 10 is uploaded to the cloud, which consequently increasing the processing load of the cloud device 30. The edge computing addresses the difficulty in data cloud processing of the massive IoT devices 10.

[0056] Specifically, the edge gateway 20 operates on an edge of the IoT device 10, and receives data uploaded by the IoT device 10 to the edge gateway 20. The edge gateway 20 computes and processes the data, and then uploads the processed data to the cloud device 30.

[0057] The IoT device 10 is an IoT underlying device capable of data acquisition and reporting, disposed in the sensing layer of the IoT system, and may transmit the acquired device data to the edge gateway 20. The IoT device 10 includes various sensors and smart terminal devices, such as a temperature and humidity sensor, a light sensor, an accelerometer, a particle counter, an RFID reader, an RFID tag, a wearable device, a Bluetooth device, and other types of wearable devices, which is not limited in the present disclosure.

[0058] Different IoT devices 10 may be applied in different application scenarios. For example, in an electric power IoT application scenario, the IoT device 10 may include a transformer, a meter, a charging pile, a temperature controller, a low-voltage master switch, a fan, or the like. As another example, under a smart home application scenario, the IoT device 10 may include a monitor, a smart phone, a smart TV, a smart doorbell, a smart light, or the like. As another example, in a transportation application

scenario, the IoT device 10 may include a traffic signal, a power distribution facility, a smart vehicle, a road camera, or the like.

[0059] The edge gateway 20 is an IoT gateway capable of performing edge computing, which is disposed in the network layer of the IoT system. The edge gateway 20 connects the IoT device 10 to the cloud device 30, which implements the analysis and processing of massive IoT device data and uploads the analyzed and processed data to the cloud device 30. The edge gateway 20 supports a variety of network protocols, and may flexibly access a variety of IoT devices 10. For example, the network protocol is the RS-485 protocol, the radio frequency identification (RFID) protocol, the near-field communication (NFC) protocol, or other network protocols, which is not limited in the present disclosure.

[0060] In some embodiments, data or information is transferred between the cloud device 30 and the edge gateway 20 using the following network protocols: the Message Queuing Telemetry Transport (MQTT) Protocol, the Hypertext Transfer Protocol (HTTP), the Constrained Application Protocol (CoAP), the Constrained Application Protocol (CoAP), or other network protocols, which is not limited in the present disclosure.

[0061] The cloud device 30 is a cloud computing platform with the capabilities to store, process, and analyze data, which is disposed in the platform layer of the IoT system. The cloud device 30 may extract valuable information through data mining, machine learning and other technologies, such that the application layer implements visualization and management.

[0062] One edge gateway 20 may receive device data from at least one IoT device 10.

[0063] In a data processing solution according to the present disclosure, each time the IoT device 10 uploads data, the edge gateway 20 triggers data processing, which is prone to causing performance problems in an edge computing scenario with insufficient hardware capabilities. Accordingly, some embodiments of the present disclosure provide a method for processing data, which is capable of reducing the computation frequency of the edge gateway 20.

[0064] Referring to FIG. 2, FIG. 2 illustrates a flowchart of a method for processing data according to an exemplary embodiment of the present disclosure. The method is

performed by the edge gateway 20 illustrated in FIG. 1. As illustrated in FIG. 2, the method may include step 210, step 220, step 230, and step 240.

[0065] In step 210, a dataset is received from a first IoT device, wherein the dataset includes at least one piece of device data having a data identifier and a timestamp.

[0066] In the IoT system, one edge gateway 20 corresponds to at least one IoT device 10. That is, the edge gateway 20 may receive a dataset from the at least one IoT device 10. In a case where one edge gateway 20 is connected to at least two IoT devices 10, a first IoT device is one of the at least two IoT devices 10.

[0067] In the embodiments of the present disclosure, the IoT device 10 (including the first IoT device as described above) in the IoT system includes one or more data acquisition points. Each of the data acquisition points periodically or irregularly acquires respective device data. The IoT device 10 periodically or irregularly reports the device data acquired by each of the data acquisition points to the edge gateway 20. In some embodiments, the data identifier of the device data is configured to indicate the data acquisition point corresponding to the device data. That is, for a same data acquisition point, the device data acquired at different moments has a same data identifier but different timestamps.

[0068] The device data is a sensor reading in the first IoT device or a data value generated by another device. The data identifier is an identifier configured to identify the device data. For example, the data identifier is a device ID, number, or other identifier. The timestamp is configured to record the time when the device data is generated. For example, the timestamp is a value accurate to milliseconds.

[0069] Exemplarily, the first IoT device acquires at least one piece of device data, and adds a corresponding data identifier and timestamp to the device data. Then, the acquired device data is encapsulated into a dataset conforming to a network protocol format, and then the dataset is sent to the edge gateway 20 using a network protocol.

[0070] For example, the first IoT device is a wind turbine for wind power generation. The dataset from the wind turbine includes the following device data: wind speed, noise intensity, vibration intensity, and the like. The wind speed corresponds to the measured value of the wind speed sensor. The data identifier of the wind speed is the device number of the wind speed sensor (e.g., FIG. 1). The timestamp of the wind speed is the measurement moment corresponding to the measured value of the wind speed sensor. The noise intensity corresponds to the measured value of the sound level meter. The

data identifier of the noise intensity is the device number of the sound level meter (e.g., SJ1). The timestamp of the noise intensity is the measurement moment corresponding to the measured value of the sound level meter. The vibration intensity corresponds to the measured value of the vibration sensor. The data identifier of the vibration intensity is the device number of the vibration sensor (e.g., ZG1). The timestamp of the vibration intensity is the measurement moment corresponding to the measured value of the vibration sensor.

[0071] In the embodiments of the present disclosure, after receiving the dataset from the first IoT device, the edge gateway 20 parses the dataset to acquire the device data, the data identifier and the timestamp corresponding to the device data.

[0072] In some embodiments, the edge gateway 20, corresponding to the at least one IoT device 10, sends to the edge gateway 20, at different moments, a dataset including at least one piece of device data. The data identifier of each piece of device data in the dataset 2 sent at the next moment is the same, or different, or partially the same as the data identifier of each piece of device data in the dataset 1 sent at the previous moment.

[0073] In step 220, at least one piece of device data in the dataset is traversed in a chronological order of timestamps.

[0074] In the embodiments of the present disclosure, the edge gateway 20 traverses the data identifier corresponding to the at least one piece of device data in the dataset according to the timestamp corresponding to the device data, and manages at least one piece of device data in the dataset in an orderly manner, thereby ensuring the temporal order and consistency of the device data for subsequent analysis and processing.

[0075] Exemplarily, an input domain is provided in the edge gateway 20 for storing at least one piece of device data from the first IoT device received at the current moment. Accordingly, the edge gateway 20 stores the at least one piece of device data from the first IoT device to the input domain in a chronological order of timestamps. Thereafter, the edge gateway 20 traverses the device data starting from the device data with an earliest timestamp.

[0076] In step 230, first device data in the dataset is added to a to-be-computed domain in a case where the first device data in the dataset is traversed and the to-be-

computed domain of the edge gateway 20 does not contain any device data having a same data identifier but a different timestamp as the first device data.

[0077] That is, the device data is added to the to-be-computed domain in a case where the edge gateway 20 traverses to the first device data in the dataset that satisfies a predetermined condition. The predetermined condition is that the data identifier of the first device data is the same as the data identifier of certain device data in the to-be-computed domain.

[0078] The to-be-computed domain of the edge gateway 20 is configured by the edge gateway 20 to store at least one piece of device data in the dataset transmitted from the first IoT device or other IoT device 10 and received prior to the current moment.

[0079] In step 240, computation is performed on each piece of device data in the to-be-computed domain in accordance with a predetermined computation rule in a case where second device data in the dataset is traversed and the to-be-computed domain of the edge gateway 20 contains the device data having a same data identifier but a different timestamp as the second device data.

[0080] That is, in a case where the edge gateway 20 traverses to the device data in the dataset that meets the unspecified conditions, the traversal is stopped and the computation on each piece of device data in the to-be-computed domain is started.

[0081] Taking the data identifier of the second device data as number n as an example, the to-be-computed domain of the edge gateway 20 containing the device data having a same data identifier but a different timestamp as the second device data indicates that the device data with the data identifier of number n is stored in the to-be-computed domain. That is, before the dataset including the second device data is sent, the first IoT device sends the dataset including the number n, and the previously transmitted device data of number n is not computed. At this time, in a case where the newly received device data of number n is also directly placed into the to-be-computed domain, two or more pieces of device data of the same number appear in subsequent computation, thereby causing computation errors. In order to ensure that the computation is real-time and to avoid errors in the subsequent computation, the computation on each piece of device data in the to-be-computed domain is performed before the newly received device data of number n is put into the to-be-computed domain.

[0082] In some embodiments, prior to step 220, the edge gateway 20 may query whether the device data is stored in the current to-be-computed domain. In a case where at least one piece of device data is stored in the current to-be-computed domain, step 220 to step 240 are performed. In a case where any one piece of device data is not stored in the current to-be-computed domain, the edge gateway 20 adds the at least one piece of device data in the dataset received at step 210 to the to-be-computed domain.

[0083] In summary, in the solution shown in the embodiments of the present disclosure, the to-be-computed domain may be set up in the edge gateway 20. In a case where the dataset from the IoT device 10 is received, the edge gateway 20 may compare the data identifier of the device data in the dataset to the data identifier of the device data in the to-be-computed domain. In a case where the data identifier of the second device data in the dataset is the same as the data identifier of certain device data in the to-be-computed domain, the device data with the timestamp earlier than the timestamp of the second device data in the dataset is added to the to-be-computed domain and computation is performed on each piece of device data in the to-be-computed domain. That is, the edge gateway 20 may store the device data in the dataset from the IoT device 10 in the to-be-computed domain in a chronological order of timestamps. In response to detecting that the IoT device 10 transmits the data identifier corresponding to the data identifier of the device data in the to-be-computed domain, computation is performed on the device data in the current to-be-computed domain, such that a situation where two pieces of device data correspond to the same data identifier in the to-be-computed domain is avoided. Thus, under the premise of ensuring real-time and accurate computation, the solution is capable of reducing the computation frequency of the edge gateway 20, such that the resource consumption of the edge gateway 20 is saved, thereby ensuring the performance of the edge gateway 20.

[0084] Based on the solution in the embodiment illustrated in FIG. 2, in one possible implementation, the computation rule includes the data identifiers of at least two pieces of device data.

[0085] The edge gateway 20 is pre-stored with a computation rule associated with the corresponding at least one IoT device 10.

[0086] For example, based on the example of the first IoT device being the wind turbine for wind power generation, the computation rule is: an output value $a = FG1 + SJ1$, or: an output value $b = SJ1 \times ZG1 - FG1$. At the time when the edge gateway 20 performs the computation, the device data corresponding to the data identifier is substituted into the computation rule to perform the computation.

[0087] In some embodiments, after the computation rule is established by the cloud device 30, the computation rule related to the edge gateway 20 is filtered and the computation rule is sent to the edge gateway 20. Accordingly, the edge gateway 20 receives the computation rule from the cloud device 30, and stores the computation rule.

[0088] Step 240 may be implemented as step 240a, step 240b, step 240c, and step 240d.

[0089] In step 240a, in a case where the second device data in the dataset is traversed and the to-be-computed domain of the edge gateway 20 contains the device data having a same data identifier but a different timestamp as the second device data, the at least two pieces of device data in the to-be-computed domain are traversed in a reverse chronological order of timestamps.

[0090] In the embodiments of the present disclosure, the edge gateway 20 traverses the data identifiers corresponding to at least two pieces of device data in the to-be-computed domain according to the timestamp corresponding to the device data, such that the at least two pieces of device data in the to-be-computed domain are managed in an orderly manner, thereby ensuring the temporal order and consistency of the device data to facilitate subsequent analysis and processing.

[0091] Exemplarily, the edge gateway 20 traverses the device data starting from the device data with the latest timestamp to query whether the computation rule corresponding to the data identifier of the current device data is stored.

[0092] In step 240b, in a case where the third device data in the to-be-computed domain is traversed, the data identifier of the third device data corresponds to the first computation rule and the to-be-computed domain does not contain any device data associated with the first computation rule, the first computation rule is associated with the third device data.

[0093] The data identifier of the third device data corresponding to the first computation rule refers to the fact that the first computation rule includes the data identifier of the third device data. The to-be-computed domain not containing any device data associated with the first computation rule means that the first computation rule is not associated with device data in the to-be-computed domain. At the time, the edge gateway 20 associates the first computation rule with the third device data for subsequent computation.

[0094] In step 240c, in a case where the third device data is traversed, the data identifier of the third device data corresponds to the first computation rule and the to-be-computed domain contains the device data associated with the first computation rule, traversing the third device data is completed.

[0095] The to-be-computed domain containing the device data associated with the first computation rule refers to the fact that the first computation rule is associated with certain device data in the to-be-computed domain. At the time, the edge gateway 20 stops traversing the third device data and continues traversing the next device data.

[0096] In step 240d, upon completion of traversing at least two pieces of device data in the to-be-computed domain, the computation is performed on each piece of device data in the to-be-computed domain in accordance with a predetermined computation rule.

[0097] In a case where the computation rules corresponding to at least two pieces of device data in the to-be-computed are acquired by querying, the edge gateway 20 performs computation on the at least two pieces of device data in the to-be-computed domain in accordance with the computation rules, such as the first computation rule, and acquires an output result corresponding to each of the computation rules.

[0098] As the computation rules include data identifiers of at least two pieces of device data, in the embodiments of the present disclosure, by the orderly management of at least two pieces of device data in the to-be-computed domain, duplicate computation rules may be filtered out before the computation is performed on each piece of device data in the to-be-computed domain, such that duplicate computations in accordance with the same computation rules are avoided. In this way, the resource consumption of the edge gateway 20 is saved, and the computation efficiency of the edge gateway 20 is improved.

[0099] Based on the solutions shown in the various embodiments of the present disclosure, in one possible implementation, the step 240d may be implemented as step 240d1, step 240d2, and step 240d3.

[00100] In step 240d1, upon completion of traversing at least two pieces of device data in the to-be-computed domain, the at least two pieces of device data in the to-be-computed domain are traversed in a chronological order of timestamps.

[00101] In the embodiments of the present disclosure, the edge gateway 20 traverses the computation rules corresponding to the at least two pieces of device data in the to-be-computed domain according to the timestamp corresponding to the device data, such that the at least two pieces of device data in the to-be-computed domain are managed in an orderly manner, thereby ensuring the time sequence and consistency of the device data to facilitate subsequent analysis and processing.

[00102] In step 240d2, in a case where the fourth device data in the to-be-computed domain is traversed and the fourth device data is not associated with the computation rule, the fourth device data is added to the real-time library.

[00103] The real-time library is a memory opened by the edge gateway 20 in the process for storing the device data in the dataset from the IoT device 10 and computation data of the edge gateway 20.

[00104] In some embodiments, the edge gateway 20 sends the device data and the computation data in the real-time library to the cloud device 30 in real-time, or the edge gateway 20 sends the device data and the computation data in the real-time library to the cloud device 30 according to a specified period for processing and application of the device data and the computation data by the cloud device 30.

[00105] That the fourth device data is not associated with the computation rule may mean that no computation rule is designated for the fourth device data or that the computation rule designated for the fourth device data is associated with the device data with a later timestamp. In order to avoid repeated computation, at this time, the edge gateway 20 does not perform computation on the fourth device data, but adds the fourth device data to the real-time library.

[00106] In step 240d3, in a case where the fifth device data is traversed and the fifth device data is associated with the second computation rule, the fifth device data is added to the real-time library, at least two pieces of device data corresponding to the second computation rule is queried from the real-time library, the computation on the at

least two pieces of device data corresponding to the second computation rule is performed in accordance with the second computation rule to acquire the first computation data, the timestamp of the fifth device data is set to the timestamp of the first computation data, and the first computation data is added to the real-time library.

[00107] That the fifth device data is associated with the second computation rule may mean that the fifth device data corresponds to the second computation rule. As the second computation rule also includes the data identifier of at least one piece of other device data, the edge gateway 20 may query the device data corresponding to the other device identifier in the second computation rule before performing the computation in accordance with the second computation rule.

[00108] In the process of associating the computation rule, i.e., in the step 240a and 240b, the edge gateway 20 traverses the at least two pieces of device data in the to-be-computed domain in a reverse chronological order of timestamps. In the process of executing the computation, the edge gateway 20 traverses at least two pieces of device data in the to-be-computed domain in a reverse chronological order of timestamps. Therefore, in the second computation rule, the timestamp of the device data corresponding to the other device identifier is earlier than the timestamp of the fifth device data, and the corresponding device data is already stored in the real-time library. In other words, the timestamp of the fifth device data is the latest timestamp among the timestamps of the at least two pieces of device data corresponding to the second computation rule, and the edge gateway 20 can query the device data corresponding to the other device identifier in the second computation rule in the real-time library.

[00109] In a case where the device data corresponding to the other device identifier in the second computation rule is acquired by querying, the edge gateway 20 substitutes the fifth device data and the device data corresponding to the other device identifier into the second computation rule to acquire the first computation data. Afterwards, the edge gateway 20 determines the timestamp of the fifth device data as the timestamp of the first computation data, and adds the first computation data to the real-time library.

[00110] In the embodiments of the present disclosure, after the relevant computation rule is queried in a reverse chronological order of timestamps, the edge gateway 20 can perform computation on each piece of device data in the to-be-computed domain

in accordance with the queried computation rule in a reverse chronological order of timestamps. In this way, not only can the repetitive computation of the same computation rule be avoided, but also the device data and the computation data can be added to the real-time library according to the timestamp. On the premise of ensuring the time sequence of the data in the real-time library, simultaneous computation and storage are implemented, such that the computation efficiency of the edge gateway 20 is improved.

[00111] Based on the solutions illustrated in the various embodiments of the present disclosure, in one possible implementation, querying at least two pieces of device data corresponding to the second computation rule from the real-time library in the step 240d3 may be implemented as follows.

[00112] At least one piece of candidate device data corresponding to the first data identifier is queried in the real-time library. The first data identifier is any one of the data identifiers included in the second computation rule.

[00113] From the at least one piece of candidate device data, the candidate device data with the latest timestamp is acquired as the device data corresponding to the first data identifier in the at least two pieces of device data corresponding to the second computation rule.

[00114] In the embodiments of the present disclosure, the edge gateway 20 acquires, based on the second computation rule, the at least one data identifier included in the second computation rule. Subsequently, the edge gateway 20 queries the device data corresponding to the at least one data identifier in the real-time library.

[00115] Since the device data in the datasets previously transmitted from the first IoT device and other IoT devices 10 is stored in the real-time library, one data identifier may correspond to unique candidate device data or may correspond to at least two pieces of candidate device data with different timestamps in the real-time library.

[00116] Accordingly, in a case where a certain data identifier in the second computation rule corresponds to at least two pieces of candidate device data, the edge gateway 20 identifies the one piece of candidate device data with the latest timestamp, as the device data corresponding to the data identifier.

[00117] The candidate device data with the latest timestamp refers to that the candidate device data is the latest device data in at least two pieces of candidate device data corresponding to the data identifier. In the embodiments of the present

disclosure, by querying the latest device data in the real-time library, the computation of historical device data of the same data identifier is avoided, such that the accuracy of the edge gateway 20 in selecting the device data participating in the computation of the rule is improved. As a result, the computation efficiency of the edge gateway 20 is improved, thereby avoiding unnecessary computation processes and saving the resource consumption of the edge gateway 20.

[00118] Based on the solutions shown in the various embodiments, in one possible implementation, the method further includes the following:

[00119] The to-be-computed domain is cleared upon completion of traversing at least two pieces of device data in the to-be-computed domain data.

[00120] The second device data and any other device data in the dataset with the timestamp later than the timestamp of the second device data are moved to the to-be-computed domain.

[00121] Completion of traversing the at least two pieces of device data in the to-be-computed domain means that each piece of device data in the to-be-computed domain is added to the real-time library. At this time, in order to save the storage space of the to-be-computed domain, and at the same time, in the process of adding new device data to the to-be-computed domain subsequently, which leads to different device data in the to-be-computed domain with the same data identifier, the edge gateway 20 clears each piece of device data in the to-be-computed domain.

[00122] The second device data described and any other device data in the dataset with the timestamp later than the timestamp of the second device data, refers to the device data that is not added to the to-be-computed domain in step 220 and step 230.

[00123] In the embodiments of the present disclosure, the edge gateway 20 clears the to-be-computed domain, and moves the second device data and other device data in the dataset with the timestamp later than the timestamp of the second device data to the to-be-computed domain. In this way, orderly management of the device data from the IoT device 10 is implemented, and thus computation is performed on the device data under the premise of ensuring the time order of the device data.

[00124] Based on the solutions shown in the various embodiments, a timer is provided in the edge gateway 20.

[00125] The timer is a hardware component or a software component for triggering computation on each piece of device data in the to-be-computed domain once after a predetermined time interval.

[00126] Exemplarily, in a case where device data is added to the to-be-computed domain, the edge gateway 20 starts the timer, and the timer may record the storage time of the device data in the to-be-computed domain.

[00127] Referring to FIG. 3, FIG. 3 illustrates a flowchart of a method for processing data according to another exemplary embodiment of the present disclosure. As illustrated in FIG. 3, the method further includes step 250 and step 260.

[00128] In step 250, in a case where a timer is expired, computation is performed on each piece of device data in the to-be-computed domain.

[00129] Based on step 240, the embodiments of the present disclosure illustrate another trigger condition that the edge gateway 20 performs the computation on each piece of device data in the to-be-computed domain, i.e., the timer is expired.

[00130] Expiration of the timer means that the timer reaching a predetermined time interval.

[00131] In some embodiments, the timer is predetermined at a time interval of 100 ms, 150 ms, 200 ms, 300 ms, or other time intervals, which is not limited in the embodiments of the present disclosure.

[00132] The process of performing the computation on each piece of device data in the to-be-computed domain is the same as the process of performing computation on each piece of device data in the to-be-computed domain in step 240, which is not limited in the present disclosure.

[00133] In step 260, the timer is reset each time the computation is performed on each piece of device data in the to-be-computed domain.

[00134] The resetting the timer refers to that the edge gateway 20 resets the timer interval, such that the timer starts timing again.

[00135] Since the edge gateway 20 clears the to-be-computed domain upon computation on each piece of device data in the to-be-computed domain, the edge gateway 20 resets the timer in order to re-record a duration of computation on each piece of device data in the to-be-computed domain.

[00136] By setting the timer, in the embodiments of the present disclosure, the duration of performing the computation on each piece of device data in the to-be-

computed domain is controlled within a reasonable range to avoid affecting the real-time performance of the computation.

[00137] Based on the solutions shown in the various embodiments of the present disclosure, referring to FIG. 4, FIG. 4 illustrates a flowchart of an implementation of a method for processing data according to an exemplary embodiment of the present disclosure. As illustrated in FIG. 4, an edge gateway 20 is provided with a to-be-computed domain and a timer. The to-be-computed domain stores at least one piece of device data in a dataset from the IoT device 10 to the edge gateway 20. The steps in FIG. 4 are performed by the edge gateway 20 and specifically include the following steps.

[00138] In step 401, whether the timer is expired is determined.

[00139] In a case where the timer is expired, step 406 is performed. Otherwise, step 402 is performed.

[00140] In step 402, the dataset is received from a first IoT device and then stored to an input domain.

[00141] The dataset includes at least one piece of device data. The device data has a data identifier and a timestamp.

[00142] In step 403, the at least one piece of device data in the input domain is traversed in a chronological order of timestamps.

[00143] In step 404, whether the traversed device data has a same data identifier and a different timestamp as the device data in the to-be-computed domain is determined.

[00144] In a case where the traversed device data has a same data identifier but a different timestamp as the device data in the to-be-computed domain, step 406 is performed. Otherwise, step 405 is performed.

[00145] In step 405, the traversed device data is added to the to-be-computed domain. Then, next device data continues to be traversed.

[00146] In step 406, in a case where the timer is expired, at least two pieces of device data in the to-be-computed domain are traversed in a reverse chronological order of timestamps.

[00147] In step 407, whether the traversed device data corresponds to a first computation rule is determined.

[00148] In a case where the traversed device data corresponds to a first computation rule, step 408 is performed. Otherwise, next device data continues to be traversed.

[00149] In step 408: whether the to-be-computed domain contains the device data associated with the first computation rule is determined. In a case where the to-be-computed domain contains the device data associated with the first computation rule, next device data continues to be traversed. Otherwise, step 409 is performed.

[00150] In step 409, the first computation rule is associated with the traversed device data.

[00151] In step 410, whether the traversal is completed is determined.

[00152] In a case where the traversal is complete, step 411 is performed. Otherwise, the traversal is continued.

[00153] In step 411, at least two pieces of device data in the to-be-computed domain are traversed in a chronological order of timestamps.

[00154] In step 412, the traversed device data is added to a real-time library.

[00155] In step 413, whether the traversed device data is associated with a computation rule is determined.

[00156] In a case where the traversed device data is associated with a computation rule, step 414 is performed. Otherwise, step 412 is performed to continue the traversal.

[00157] In step 414, first computation data is acquired and a timestamp of the traversed device data is set to a timestamp of the first computation data.

[00158] In step 415, the first computation data is added to the real-time library.

[00159] Step 401 to step 415 are performed in a same manner as that in the various embodiments of the present disclosure, which is not repeated herein.

[00160] Referring to FIG. 5, FIG. 5 illustrates a schematic diagram of a method for processing data according to an exemplary embodiment of the present disclosure. As illustrated in FIG. 5, IoT devices 10 connected to an edge gateway 20 are an IoT device A, an IoT device B, an IoT device C, and other IoT devices 10 (not illustrated in the drawing).

[00161] The edge gateway 20 establishes an MQTT connection with a configuration center in a cloud. After the configuration center in the cloud establishes a computation rule, all configuration information related to the edge gateway 20 is packaged and sent to the edge gateway 20 through MQTT connection. Accordingly, the edge gateway 20

receives a resource packet from the cloud through the MQTT connection, analyzes the resource packet and then takes out the computation rule.

[00162] The IoT device A, the IoT device B, and the IoT device C may each send the dataset including at least one piece of device data to the edge gateway 20 at different moments. Exemplarily, the dataset from the IoT device A includes three pieces of device data: a_in1 (X1ms), a_in2 (X2ms) and a_in3 (X3ms). a_in1, a_in2, and a_in3 correspond to data identifiers of the three pieces of device data. X1ms, X(2)ms, and X3ms respectively correspond to the timestamps of the three pieces of device data. The dataset from the IoT device B includes three pieces of device data: b_in1 (Y1ms), b_in2 (Y2ms), b_in3 (Y3ms). The dataset from the IoT device C includes three pieces of device data: c_in1(Z1ms), c_in2(Z2ms), c_in3(Z3ms).

[00163] In the embodiments of the present disclosure, after receiving the dataset from the IoT device A, the IoT device B, the IoT device C, or the other IoT device 10, the edge gateway 20 first performs packet aggregation.

[00164] Specifically, the edge gateway 20 stores at least one piece of device data in the received dataset to the input domain. Since different IoT devices 10 may transmit the datasets to the edge gateway 20 at different moments, the device data received by the edge gateway 20 is usually disordered. Accordingly, the edge gateway 20 traverses each piece of device data in the input domain in a chronological order of timestamps in the input domain.

[00165] Second, in the packet aggregation process, in a case where the edge gateway 20 discovers that the input domain and the to-be-computed domain contain device data (i.e., duplicate points) having the same data identifier but different timestamps, such as the device data 1 (i.e., the second device data described above), the computation on each piece of device data in the to-be-computed domain is triggered by the edge gateway 20, and the packet aggregation timer is reset.

[00166] Specifically, the edge gateway 20, from the input domain, extracts the device data whose timestamp is earlier than the device data 1, and moves the device data in an orderly manner to the to-be-computed domain. Thereafter, the edge gateway 20 performs computation on each piece of device data in to-be-computed domain in accordance with the computation rule issued by the cloud.

[00167] In addition, in the packet aggregation process, in a case where the packet aggregation timer is expired, the computation on each piece of device data in the to-be-computed domain is triggered by the edge gateway 20.

[00168] In addition, in a case where no duplicate points are found during one packet aggregation cycle, but the packet aggregation timer is expired, the computation on each piece of device data in the to-be-computed domain is triggered by the edge gateway 20.

[00169] In a process of computing each piece of device data in the to-be-computed domain by the edge gateway 20, in order to avoid repeated computations, each piece of device data in the to-be-computed domain is first combined into a total ordered device data by merging and sorting, and then traversed in a reverse order according to the timestamp of the device data, to find out all required computation rules.

[00170] In order to ensure that the device data is output to the real-time library in an orderly manner, the device data is traversed in a chronological order of timestamps. Simultaneous computation and storage are adopted to ensure that the device data and the computation data acquired by the computation rule are entered into the library in a chronological order of timestamps.

[00171] First embodiment

[00172] The IoT device A sends a first dataset 1 to the edge gateway 20 at a time instant 110 ms. The dataset 1 includes device data a_in1 (100 ms), device data a_in2 (105 ms), and device data a_in3 (110 ms). At this time, no other device data is stored in the to-be-computed domain, such that each piece of device data in the dataset 1 is directly placed in the to-be-computed domain. In a case where the packet aggregation timer is expired and the edge gateway 20 does not receive any other dataset, the edge gateway 20 arranges all the device data in the to-be-computed domain into a queue from small to large according to the timestamps, and traverses forward from the tail of the queue to acquire the corresponding computation rule. The computation rule 1 is expressed as: $a_{out1}=a_{in1}+a_{in2}$.

[00173] Referring to FIG. 6, FIG. 6 illustrates a schematic diagram of traversing device data according to a first embodiment of the present disclosure. As illustrated in FIG. 6, first, the edge gateway 20 traverses a_in3 (110 ms) and finds no corresponding computation rule, and is thus ignored. The edge gateway 20 traverses to a_in2 (105 ms) and finds a corresponding computation rule 1, and hangs the computation rule 1

after the point a_in2 (105 ms). The edge gateway 20 traverses to a_in1 (100 ms) and finds the corresponding computation rule 1, but the corresponding computation rule 1 is already present at this point and is thus ignored.

[00174] Second, subsequent to acquiring the computation rule, the edge gateway 20 traverses all the device data from the head of the queue, and performs computation. The whole computation process includes the following operations:

[00175] a_in1 (100 ms) is input into a real-time library.

[00176] a_in2 (105 ms) is input into the real-time library.

[00177] In accordance with the computation rule 1, the latest device data snapshots a_in1 (100 ms) and a_in2 (105 ms) are fetched from the real-time library to acquire the computation data a_out1, the timestamp is 105 ms, and then a_out1 (105 ms) is entered into the real-time library.

[00178] a_in3 (110ms) is input into the real-time library.

[00179] Upon completion of the computation process, the edge gateway 20 clears each piece of device data in the to-be-computed domain and resets a packet aggregation timer.

[00180] At this point, the device data a_in1 (100 ms), a_in2 (105 ms), a_out1 (105 ms), and a_in3 (110 ms) is already present in the real-time library.

[00181] Subsequently, the IoT device A sends a first dataset 2 to the edge gateway 20 at a time instant 500 ms. The dataset 2 includes device data a_in1 (500 ms), device data a_in2 (505 ms), and device data a_in3 (510 ms). At this time, no other device data is stored in the to-be-computed domain, such that each piece of device data in dataset 2 is directly placed in the to-be-computed domain. In a case where the packet aggregation timer is expired and the edge gateway 20 does not receive any other dataset, the edge gateway 20 arranges all the device data in the to-be-computed domain into a queue from small to large according to the timestamps, and traverses forward from the tail of the queue to acquire the corresponding computation rule. The computation rule 1 is expressed as: $a_out1 = a_in1 + a_in2$.

[00182] First, the edge gateway 20 traverses a_in3 (510 ms) and finds no corresponding computation rule, and is thus ignored. The edge gateway 20 traverses to a_in2 (505 ms) and finds the corresponding computation rule 1, and hangs the computation rule 1 after the point a_in2 (505 ms). The edge gateway 20 traverses to

a_in1 (500 ms) and finds the corresponding computation rule 1, but the computation rule 1 is already present at this point and is thus ignored.

[00183] Second, subsequent to acquiring the computation rule, the edge gateway 20 traverses all the device data from the head of the queue, and performs computation. The whole computation process includes the following operations:

[00184] a_in1 (500 ms) is input into the real-time library.

[00185] a_in2 (505 ms) is input into the real-time library.

[00186] In accordance with the computation rule 1, the latest device data snapshots a_in1 (500ms) and a_in2(505ms) are fetched from the real-time library to acquire the computation data a_out1, the timestamp is 505 ms, and then a_out1 (505 ms) is entered into the real-time library.

[00187] a_in3 (510 ms) is input into the real-time library.

[00188] Upon completion of the above computation process, the edge gateway 20 clears each piece of device data in the to-be-computed domain, and resets a packet aggregation timer.

[00189] At this time, the device data a_in1 (100 ms), the device data a_in2 (105 ms), the device data a_out1 (105 ms), the device data a_in3 (110 ms), the device data a_in1 (500 ms), the device data a_in2 (505 ms), the device data a_out1 (505 ms), and the device data a_in3 (510 ms) is already present in the real-time library.

[00190] Second embodiment

[00191] The device data c_in3 (100 ms) is already present in the real-time library. At this time, the IoT device A acquires and transmits one dataset 3 including device data a_in2 (105 ms), device data a_in3 (115 ms), device data a_in1 (200 ms), device data a_in2 (205 ms), and device data a_in3 (210 ms). Meanwhile, a_in1 (100 ms) of the IoT device A, b_in1 (110 ms), b_in2 (120 ms), and b_in3 (130 ms) of the IoT device B, and c_in1 (125ms) and c_in2 (135 ms) of the IoT device C are already present in the to-be-computed domain.

[00192] Referring to FIG. 7, FIG. 7 illustrates a schematic diagram of traversing device data according to a second embodiment of the present disclosure. As illustrated in FIG. 7, first, the device data traverses a_in2 (105 ms) and a_in3 (115 ms) and finds no corresponding computation rule, and is thus ignored. In a case where the edge gateway 20 traverses to a_in1 (200 ms) and finds the device data a_in1 (100ms) with the same data identifier in the to-be-computed domain, the device data a_in2(105ms)

and the device data a_{in3} (100ms) before the device data a_{in1} (200ms) are put into the to-be-computed domain and one computation is performed. The device data a_{in1} (200 ms) and the subsequent device data a_{in2} (205 ms) and a_{in3} (210 ms) are suspended and stored in the input domain to wait for the next traversal.

[00193] Second, all the device data in the to-be-computed domain are arranged in a queue from small to large according to the timestamps, and the time complexity of the step is $O(n \log(k))$, wherein k represents the number of devices participating in the computation. The edge gateway 20 traverses forward from the tail of the queue to acquire the corresponding computation rule. The computation rule 2 is expressed as: $b_{out1} = a_{in3} + b_{in1}$, the computation rule 3 is expressed as: $b_{out2} = a_{in1} * b_{in2}$, the computation rule 4 is expressed as: $c_{out1} = a_{in2} + b_{in3} + c_{in2} - c_{in3}$, and the computation rule 5 is expressed as: $c_{out2} = c_{in1} / c_{in2}$.

[00194] Then, the edge gateway 20 traverses to c_{in2} (135 ms) and finds the corresponding computation rule 4 and computation rule 5, and hangs the computation rule 4 and the computation rule 5 after point c_{in2} (135 ms). The edge gateway 20 traverses to b_{in3} (130 ms) and finds the corresponding computation rule 4, but the computation rule 4 is already present at this point and is thus ignored. The edge gateway 20 traverses to c_{in1} (125 ms) and finds the corresponding computation rule 5, but the computation rule 5 is already present at this time and is thus ignored. The edge gateway 20 traverses to b_{in2} (120 ms) and finds the corresponding computation rule 3, and hangs the computation rule 3 subsequent to point b_{in2} (120 ms). The edge gateway 20 traverses to a_{in3} (115 ms) and finds the corresponding computation rule 2, and hangs the computation rule 2 after point a_{in3} (115 ms). The edge gateway 20 traverses to b_{in1} (110 ms) and finds the corresponding computation rule 2, but the computation rule 2 is already present at this time and is thus ignored. The edge gateway 20 traverses to a_{in2} (105 ms) and finds the corresponding computation rule 4, but the computation rule 4 is already present at this time and is thus ignored. The edge gateway 20 traverses to a_{in1} (100 ms) and finds the corresponding computation rule 3, but the computation rule 3 is already present at this time and is thus ignored.

[00195] After the edge gateway 20 acquires the computation rule, all the device data are traversed from the head of the queue and executes the computation. The whole computation process includes the following:

[00196] a_in1 (100 ms) is input into the real-time library.

[00197] a_in2 (105 ms) is input into the real-time library.

[00198] b_in1 (110 ms) is input into the real-time library.

[00199] a_in3 (115 ms) is input into the real-time library.

[00200] In accordance with the computation rule 2, the latest input data snapshots b_in1 (110ms) and a_in3 (115ms) are fished from the real-time library to acquire the computation data b_out1, the timestamp is 115ms, and then b_out1 (115ms) is entered into the real-time library.

[00201] b_in2 (120 ms) is input into the real-time library.

[00202] In accordance with the computation rule 3, the latest input data snapshots a_in1(100ms) and b_in2(120ms) are fished from the real-time library to acquire the computation data b_out2, the timestamp is 120 ms, and then b_out2 (120 ms) is input into the real-time library.

[00203] c_in1 (125 ms) is input into the real-time library.

[00204] b_in3 (130 ms) is input into the real-time library.

[00205] c_in2 (135 ms) is input into the real-time library.

[00206] In accordance with the computation rule 4, the latest input data snapshots a_in2 (105 ms), b_in3 (130 ms), c_in2 (135 ms) and c_in3 (100 ms) are fished from the real-time library to acquire the computation data c_out1, the timestamp is at 135ms, and then c_out1 (135 ms) is entered into the real-time library.

[00207] In accordance with computation rule 5, the latest input data snapshots c_in1(125ms) and c_in2(135ms) is fished from the real-time library to acquire the computation data c_out2, the timestamp is 135ms, and then c_out2 (135 ms) is entered into the real-time library.

[00208] Upon completion of the computation process, the edge gateway 20 clears each piece of device data in the to-be-computed domain and resets a packet aggregation timer.

[00209] At this time, the device data c_in3 (100 ms), a_in1 (100 ms), a_in2 (105 ms), b_in1 (110 ms), a_in3 (115 ms), b_out1 (115 ms), b_in2 (120 ms), b_out2 (120 ms), c_in1 (125 ms), b_in3 (130 ms), c_in2 (135 ms), c_out1 (135 ms), and c_out2 (135 ms) is already present in the real-time library.

[00210] It is noted that the edge gateway 20 may transmit the ordered device data and the computation data, in the real-time library in real-time to the cloud, such that the cloud may process and apply the device data and the computation data.

[00211] In summary, the method may be widely applied to scenarios requiring fast response and instant insight, such as common power combination of electric meters branches, which requires the power of a plurality of branches to be aggregated into a total power to be reported after the branch collection point is reported. For another example, in the scenario of integrating light storage and charging, the working state and the total active power are read to determine whether the energy storage system is charging or discharging currently. For another example, after the particle counter is installed in a laboratory with dust-free requirements, a periodic average operation needs to be performed on the reported value of the counter to monitor the number of particles in real time. For another example, when the fan generates electricity, the real-time generated power of the fan is calculated through the wind speed, the radius of the blade and the power coefficient.

[00212] Referring to FIG. 8, FIG. 8 illustrates a block diagram of an apparatus for processing data illustrated in an exemplary embodiment of the present disclosure. The apparatus may be configured to perform all or some of the steps performed by a computer device as in the method illustrated in FIG. 2 or FIG. 3. As illustrated in FIG. 8, the apparatus includes:

[00213] a receiving module 801, configured to receive a dataset from a first IoT device, wherein the dataset includes at least one piece of device data. The device data has a data identifier and a timestamp;

[00214] a traversing module 802, configured to traverse the at least one piece of device data in the dataset in a chronological order of timestamps;

[00215] an adding module 803, configured to add first device data to the to-be-computed domain in a case where the first device data in the dataset is traversed and the to-be-computed domain of the edge gateway 20 does not contain any device data having a same data identifier but a different timestamp as the first device data; and

[00216] a computing module 804, configured to perform a computation on each piece of device data in the to-be-computed domain in accordance with a predetermined computation rule in a case where second device data in the dataset is traversed and

the to-be-computed domain of the edge gateway 20 contains the device data having a same data identifier but a different timestamp as the second device data.

[00217] In some embodiments, the computation rule includes data identifiers of at least two pieces of device data.

[00218] The computing module 804 is configured to traverse at least two pieces of device data in the to-be-computed domain in a reverse chronological order of timestamps in a case where the second device data in the dataset is traversed and the to-be-computed domain of the edge gateway 20 contains the device data having a same data identifier but a different timestamp as the second device data.

[00219] The computing module 804 is configured to associate a first computation rule with third device data after the third device data in the to-be-computed domain is traversed and data identifier of the third device data corresponds to the first computation rule and the to-be-computed domain does not contain any device data associated with the first computation rule.

[00220] The computing module 804 is configured to stop traversing the third device data in a case where the third device data is traversed and the data identifier of the third device data corresponds to the first computation rule and the to-be-computed domain contains the device data associated with the first computation rule.

[00221] The computing module 804 is configured to perform computation on each piece of device data in the to-be-computed domain in accordance with the predetermined computation rule upon completion of traversing the at least two pieces of device data in the to-be-computed domain.

[00222] In some embodiments, the computing module 804 is configured to traverse at least two pieces of device data in the to-be-computed domain in a chronological order of timestamps upon completion of traversing the at least two pieces of device data in the to-be-computed domain.

[00223] The computation module 804 is configured to add fourth device data to the real-time library in a case where the fourth device data in the to-be-computed domain is traversed and the fourth device data is not associated with the computation rule;

[00224] The computing module 804 is configured to add fifth device data to the real-time library in a case where the fifth device data is traversed and the fifth device data is associated with the second computation rule. At least two pieces of device data corresponding to the second computation rule is queried from the real-time library. In

accordance with the second computation rule, the computation on the at least two pieces of device data corresponding to the second computation rule is performed to acquire the first computation data. A timestamp of the fifth device data is set to the timestamp of the first computation data. The first computation data is added to the real-time library.

[00225] In some embodiments, the computing module 804 is configured to query at least one piece of candidate device data corresponding to the first data identifier in the real-time library. The first data identifier is any one of the data identifiers included in the second computation rule.

[00226] The computation module 804 is configured to acquire, from the at least one piece of candidate device data, one piece of candidate device data corresponding to a latest timestamp, as the device data corresponding to the first data identifier in the at least two pieces of device data corresponding to the second computation rule.

[00227] In some embodiments, the apparatus further includes a clearing module, configured to clear the to-be-computed domain upon completion of traversing the at least two pieces of device data in the to-be-computed domain.

[00228] The apparatus further includes an updating module, configured to move the second device data and other device data in the dataset whose timestamp are after the timestamp of the second device data to the to-be-computed domain.

[00229] In some embodiments, a timer is provided in the edge gateway 20. The apparatus further includes: a timing module configured to perform the computation on each piece of device data in the to-be-computed domain in a case where the timer is expired.

[00230] The timing module is configured to reset the timer each time upon completion of computation on each piece of device data in the to-be-computed domain.

[00231] It should be noted that, the apparatus according to the embodiments implements the functions thereof, only the division of the foregoing functional modules is illustrated. In actual application, the functions can be assigned to be completed by different functional modules according to the actual needs, i.e., the content structure of the device can be divided into different functional modules in order to complete all or part of the functions.

[00232] With respect to the apparatus in the embodiments, the specific manner in which each module performs the operation is described in detail in the embodiment

relating to the method. The technical effect achieved by the execution of the operation by each module is the same as that in the embodiment relating to the method, which is not described in detail herein.

[00233] Referring to FIG. 9, FIG. 9 illustrates a structural block diagram of a computer device 900 according to an exemplary embodiment of the present disclosure. The computer device may be implemented as a server in the embodiments of the present disclosure. The computer device 900 includes a central processing unit (CPU) 901, a system memory 904 including a random access memory (RAM) 902 and a read-only memory (ROM) 903, and a system bus 905 connected to the system memory 904 and the central processing unit 901. The computer device 900 also includes a mass storage device 906 for storing an operating system 909, an application program 910, and other program module 911.

[00234] The mass storage device 906 is connected to the central processing unit 901 via a mass storage controller (not illustrated) connected to the system bus 905. The mass storage device 906 and an associated computer-readable media thereof provide non-volatile storage for the computer device 900. That is, the mass storage device 906 may include the computer-readable media, such as a hard disk or a compact disc read-only memory (CD-ROM) drive (not illustrated).

[00235] Without loss of generality, the computer-readable medium may include a computer storage medium and a communication medium. The computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technique for storing information such as computer-readable instructions, data structures, program modules, or other data. The computer storage media includes a RAM, a ROM, an erasable programmable read only memory (EPROM), an electrically-erasable programmable read-only memory (EEPROM) flash memory or any other solid state storage device, a CD-ROM, a digital versatile disc (DVD) or any other optical storage device, a tape cartridge, a magnetic tape, a disk storage device, or any other magnetic storage device. A person skilled in the art may know that the computer storage medium is not limited to the foregoing. The system memory 904 and the mass storage device 906 may be collectively referred to as a memory.

[00236] According to various embodiments of the present disclosure, the computer device 900 may also be operated by a remote computer connected to a network via a

network, such as the Internet. That is, the computer device 900 may be connected to a network 908 via a network interface unit 907 connected to the system bus 905, or alternatively, the network interface unit 907 may also be configured to connect to other types of networks or remote computer systems (not illustrated).

[00237] The memory further includes at least one computer instruction, wherein the at least one computer instruction is stored in the memory. The central processing unit 901 implements all or part of the steps in the method in each of the embodiments by executing the at least one computer instruction.

[00238] In an exemplary embodiment, a chip is also provided. The chip includes a programmable logic circuitry and a program instruction. The chip is configured to implement the method for processing data above when the chip is operated on a computer device.

[00239] In an exemplary embodiment, a computer program product is also provided. The computer program product includes a computer instruction stored in a computer-readable storage medium. A processor of the computer device reads the computer instruction from the computer-readable storage medium. The processor reads and executes the computer instruction from the computer-readable storage medium to implement the method for processing data according to each of the method embodiments.

[00240] In an exemplary embodiment, a computer-readable storage medium is also provided. The computer-readable storage medium stores a computer instruction. The computer instruction is loaded and executed by a processor to implement the method for processing data according to each of the method embodiments.

[00241] A person of ordinary skill in the art may understand that all or part of the steps for implementing the embodiments may be accomplished by a hardware, or may be accomplished by instructing related hardware through a program. The program may be stored in a computer-readable storage medium. The storage medium may be a read-only memory, a disk or a CD-ROM, or the like.

[00242] It should be appreciated by those skilled in the art that, in one or more of the embodiments, the functions described in the embodiments of the present disclosure may be implemented by hardware, software, firmware, or any combination thereof. When implemented by software, the functions may be stored in a computer-readable medium or transmitted as one or more instructions or code on a computer-readable

medium. The computer-readable media include computer storage media and communication media, wherein the communication media include any medium that facilitates transfer of a computer program from one location to another. The storage medium may be any available medium that can be accessed by a general-purpose or special-purpose computer.

[00243] The foregoing are only optional embodiments of the present disclosure and are not intended to limit the present disclosure, and any modifications, equivalent substitutions, improvements, etc., made within the spirit and principles of the present disclosure shall be included in the scope of protection of the present disclosure.

CLAIMS

1. A method for processing data, applicable to an edge gateway in an Internet of things (IoT) system, the method comprising:

receiving a dataset from a first IoT device, wherein the dataset comprises at least one piece of device data, each of the at least one piece of device data having a data identifier and a timestamp;

traversing the at least one piece of device data in the dataset in a chronological order of timestamps;

adding first device data in the dataset to a to-be-computed domain in a case where the first data in the dataset is traversed and the to-be-computed domain of the edge gateway does not contain any data having a same data identifier but a different timestamp as the first device data;

traversing at least two pieces of device data in the to-be-computed domain in a reverse chronological order of timestamps in a case where second device data in the dataset is traversed and the to-be-computed domain of the edge gateway contains device data having a same data identifier but a different timestamp as the second device data;

associating the first computation rule with third device data in a case where third device data in the to-be-computed domain is traversed, a data identifier of the third device data corresponds to a first computation rule, and the to-be-computed domain does not contain any device data associated with the first computation rule;

stopping traversing the third device data in a case where the third device data is traversed, the data identifier of the third device data corresponds to the first computation rule, and the to-be-computed domain contains the device data associated with the first computation rule; and

performing computation on each piece of device data in the to-be-computed domain in accordance with a predetermined computation rule upon completion of

traversing the at least two pieces of device data in the to-be-computed domain, wherein the computation rule comprises data identifiers of the at least two pieces of device data.

2. The method according to claim 1, wherein performing the computation on the each piece of device data in the to-be-computed domain in accordance with the predetermined computation rule upon completion of traversing the at least two pieces of device data in the to-be-computed domain comprises:

traversing the at least two pieces of device data in the to-be-computed domain in a chronological order of timestamps upon completion of traversing the at least two pieces of device data in the to-be-computed domain;

adding fourth device data to a real-time library in a case where the fourth device data in the to-be-computed domain is traversed and the fourth device data is not associated with the predetermined computation rule; and

in a case where fifth device data is traversed and the fifth device data is associated with a second computation rule, adding the fifth device data to the real-time library, querying at least two pieces of device data corresponding to the second computation rule from the real-time library, performing computation on the at least two pieces of device data corresponding to the second computation rule in accordance with the second computation rule to acquire first computation data, setting a timestamp of the fifth device data to a timestamp of the first computation data, and adding the first computation data to the real-time library.

3. The method according to claim 2, wherein querying the at least two pieces of device data corresponding to the second computation rule from the real-time library comprises:

querying, from the real-time library, at least one piece of candidate device data corresponding to a first data identifier, wherein the first data identifier is any one of data identifiers included in the second computation rule; and

acquiring candidate device data with a latest timestamp from the at least one piece of candidate device data as the device data corresponding to the first data

identifier from the at least two pieces of device data corresponding to the second computation rule.

4. The method according to claim 2, further comprising:

clearing the to-be-computed domain upon completion of traversing the at least two pieces of device data in the to-be-computed domain; and

moving the second device data and other device data in the dataset with a timestamp subsequent to a timestamp of the second device data to the to-be-computed domain.

5. The method according to any one of claims 1 to 4, wherein a timer is arranged in the edge gateway; and the method further comprises:

performing computation on each piece of device data in the to-be-computed domain in a case where the timer is expired; and

resetting the timer upon completion of performing the computation on the each piece of device data in the to-be-computed domain.

6. An apparatus for processing data, comprising:

a receiving module, configured to receive a dataset from a first Internet of things (IoT) device, wherein the dataset comprises at least one piece of device data, each of the at least one piece of device data having a data identifier and a timestamp;

a traversing module, configured to traverse the at least one piece of device data in the dataset in a chronological order of timestamps;

an adding module, configured to add first device data in the dataset to a to-be-computed domain in a case where the first device data in the dataset is traversed and the to-be-computed domain of an edge gateway does not contain any device data having a same data identifier but a different timestamp as the first device data; and

a computing module, configured to: traverse at least two pieces of device data in the to-be-computed domain in a reverse chronological order of timestamps in a case

where second device data is traversed in the dataset and the to-be-computed domain of the edge gateway contains device data having a same data identifier but a different timestamp as the second device data; associate a first computation rule with third device data in a case where the third device data in the to-be-computed domain is traversed, the third device data has a data identifier corresponding to the first computation rule, and the to-be-computed domain does not contain any device data associated with the first computation rule; stop traversing the third device data in a case where the third device data is traversed, the third device data has the data identifier corresponding to the first computation rule, and the to-be-computed domain contains the device data associated with the first computation rule; and perform computation on each piece of device data in the to-be-computed domain in accordance with a predetermined computation rule upon completion of traversing the at least two pieces of device data in the to-be-computed domain, wherein the computation rule comprises data identifiers of the at least two pieces of device data.

7. A computer device, comprising: a processor and a memory, wherein the memory stores at least one computer instruction, wherein the processor, when loading and executing the at least one computer instruction, is caused to perform the method for processing data as defined in any one of claims 1 to 5.

8. A computer-readable storage medium, storing at least one computer instruction, wherein the at least one computer instruction, when loaded and executed by a processor, causes the processor to perform the method for processing data as defined in any one of claims 1 to 5.

9. A computer program product, comprising at least one computer instruction, wherein the at least one computer instruction is stored in a computer-readable storage medium, and the at least one computer instruction, when read and executed by a processor of a computer device, causes the processor to perform the method for processing data as defined in any one of claims 1 to 5.