(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 4 386 536 A1**

(12) **EUROPEAN PATENT APPLICATION**
published in accordance with Art. 153(4) EPC

(43) Date of publication:
19.06.2024 Bulletin 2024/25

(21) Application number: **22863153.7**

(22) Date of filing: **17.08.2022**

(51) International Patent Classification (IPC):
G06F 3/06 *(2006.01)*

(52) Cooperative Patent Classification (CPC):
G06F 3/06; G06F 11/10; G06F 12/02

(86) International application number:
PCT/CN2022/113098

(87) International publication number:
WO 2023/030013 (09.03.2023 Gazette 2023/10)

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
KH MA MD TN

(30) Priority: **01.09.2021 CN 202111022655**

(71) Applicant: **Huawei Technologies Co., Ltd.**
Shenzhen, Guangdong 518129 (CN)

(72) Inventors:
- **ZHANG, Chenxu**
  Shenzhen, Guangdong 518129 (CN)
- **LUO, Xiaodong**
  Shenzhen, Guangdong 518129 (CN)

(74) Representative: **Gill Jennings & Every LLP**
The Broadgate Tower
20 Primrose Street
London EC2A 2ES (GB)

(54) DATA PROCESSING METHOD AND APPARATUS

(57) This application discloses a data processing method and apparatus. The method includes: A processor or a network interface card of a client generates a first copy and a second copy of data. The network interface card of the client writes the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral remote memory access RDMA manner. Alternatively, a network interface card of the storage device respectively writes the first copy and the second copy of the data into the first address and the second address in the memory according to a unilateral RDMA write request sent by the client. When detecting that the first copy read from the first address is inconsistent with the second copy read from the second address, the storage device performs a data overwrite operation on data stored in the first address or data stored in the second address, so that the data stored in the two memory addresses is the same. Implementing this application implements, through one-time IO consumption, atomicity of data writing, reduces a data writing delay, and improves performance of a storage system.

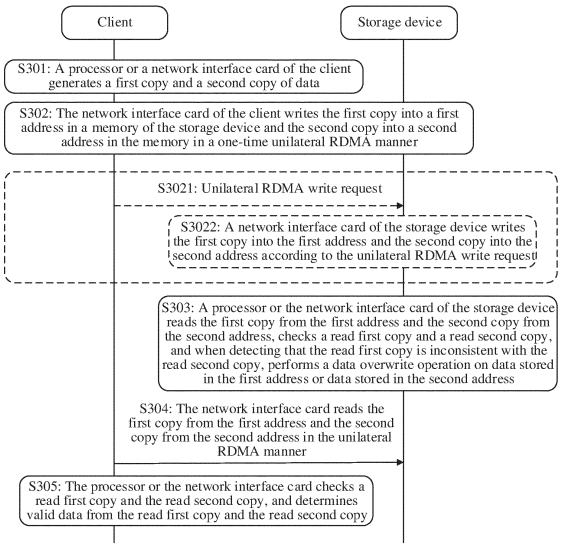

FIG. 12

EP 4 386 536 A1

Description

[0001] This application claims priority to Chinese Patent Application No. 202111022655.5, filed with the China National Intellectual Property Administration on September 1, 2021 and entitled "DATA PROCESSING METHOD AND APPARATUS", which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] This application relates to the field of data storage, and in particular, to a data processing method and apparatus.

BACKGROUND

[0003] To resolve a problem of data tearing due to a power failure or the like in a process of writing data into a storage medium, in other words, that some data is successfully written and some data fails to be written, a two-time submission manner, a three-time submission manner, and the like are proposed to ensure atomicity of data writing, so that all data carried in a write request of a current time is written or not written.

[0004] However, in both the two-time submission manner and the three-time submission manner, to-be-written data is persistently stored in a log-based manner. This results in large input/output (I/O) overheads and long network round-trip time, and reduces performance of a storage system.

SUMMARY

[0005] This application discloses a data processing method and apparatus, to implement atomicity of data writing, reduce I/O overheads, and improve performance of a storage system.

[0006] According to a first aspect, this application provides a data processing method, where the method includes: A processor or a network interface card of a client generates a first copy and a second copy of data, where the processor communicates with the network interface card. The network interface card writes the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral remote memory access RDMA manner.

[0007] The network interface card of the client writes the data into the memory of the storage device by using unilateral RDMA. This avoids access of operating systems of both parties in a data writing process, and this can reduce consumption of bandwidth and computing power of the memory in the data writing process, and improve data writing efficiency.

[0008] In the foregoing method, the network interface card of the client implements continuous dual write of the first copy and the second copy of the data into the memory of the storage device by using a one-time unilateral RDMA technology, in other words, writes the first copy into the first address in the memory of the storage device, and after the first copy is written into the first address, writes the second copy into the second address in the memory. This reduces I/O overheads during data writing, reduces a data writing delay, and effectively improves performance of a storage system.

[0009] Optionally, the memory is a persistent memory.

[0010] In the foregoing implementation, the persistent memory can improve a speed and efficiency of data reading and writing by the client, and implement persistence of stored data.

[0011] Optionally, the method further includes: The network interface card reads the first copy from the first address and the second copy from the second address in the unilateral RDMA manner. The processor or the network interface card checks the read first copy and the read second copy, and determines valid data from the read first copy and the read second copy.

[0012] In the foregoing implementation, the processor or the network interface card of the client may further verify validity or integrity of the first copy read from the first address and the second copy read from the second address, to obtain the valid data from the first copy and the second copy for use.

[0013] Optionally, that the processor or the network interface card checks the read first copy and the read second copy, and determines the valid data from the read first copy and the read second copy is specifically: When detecting that the read first copy is the same as the read second copy, the processor or the network interface card determines that the read first copy and the read second copy are the valid data.

[0014] That the read first copy is the same as the read second copy includes any one of the following two implementations.

[0015] In specific implementation, the read first copy is the same as the first copy of the data, and the read second copy is the same as the second copy of the data. In this case, it indicates that the first copy of the data has been successfully written into the first address in the memory, and the second copy of the data has been successfully written into the second address in the memory. In this way, the processor or the network interface card of the client determines that the read first copy and the read second copy are the valid data. This ensures atomicity of the data writing.

[0016] In another specific implementation, the read first copy is the same as initial data stored in the first address in the memory, and the read second copy is the same as initial data stored in the second address in the memory, where the initial data stored in the first address is data stored in the first address before the first copy of the data is written into the first address in the memory, the initial data stored in the second address is data stored in the second address before the second copy of the data is written into the second address in the memory, and

the initial data stored in the first address is the same as the initial data stored in the second address. In this way, the processor or the network interface card of the client determines that the read first copy and the read second copy are the valid data. This ensures the atomicity of the data writing.

[0017] According to a second aspect, this application provides a data processing method. The method includes: A network interface card of a storage device receives a unilateral remote memory access RDMA write request sent by a client, where the unilateral RDMA write request includes a first copy and a second copy of data, a first address of the first copy in a memory of the storage device, and a second address of the second copy in the memory. The network interface card writes the first copy into the first address and the second copy into the second address.

[0018] In the foregoing method, the network interface card of the storage device implements continuous dual write of the first copy of the data and the second copy of the data in the memory of the storage device in a one-time unilateral RDMA manner, in other words, after the first copy is written into the first address in the memory, the second copy is written into the second address in the memory. This reduces I/O consumption, reduces a data writing delay, and effectively improves storage system performance.

[0019] Optionally, the memory is a persistent memory.

[0020] In the foregoing implementation, the persistent memory can improve a speed and efficiency of data reading and writing by the client, and implement persistence of stored data.

[0021] Optionally, the method further includes: The network interface card or a processor of the storage device reads the first copy from the first address and the second copy from the second address. The network interface card or the processor of the storage device checks the read first copy and the read second copy, and when detecting that the read first copy is inconsistent with the read second copy, performs a data overwrite operation on data stored in the first address or data stored in the second address.

[0022] In the foregoing implementation, when detecting that the read first copy is inconsistent with the read second copy, the network interface card or the processor of the storage device may perform the data overwrite operation, so that the data stored in the first address is the same as the data stored in the second address. This implements atomicity of data writing.

[0023] Optionally, the read first copy includes first data and first check code, the read second copy includes second data and second check code, and the read first copy is inconsistent with the read second copy when at least one of the following conditions is met: the first data is different from the second data, or the first check code is different from the second check code.

[0024] That the first data is different from the second data means that content of the first data is different from content of the second data. That the first check code is different from the second check code means that content of the first check code is different from content of the second check code.

[0025] In specific implementation, a length of the first data is the same as a length of the second data, and a length of the first check code is the same as a length of the second check code.

[0026] Optionally, that the data overwrite operation is performed on the data stored in the first address or the data stored in the second address includes: When checking of the first data based on the first check code succeeds, the first data and the first check code overwrite the data stored in the second address.

[0027] In the foregoing implementation, that the checking of the first data based on the first check code succeeds indicates that the read first copy is complete and valid, that is, the data stored in the first address is complete and valid. Therefore, the first data and the first check code overwrite the data stored in the second address, so that the data stored in the first address is completely the same as the data stored in the second address. This implements the atomicity of the data writing.

[0028] Optionally, that the data overwrite operation is performed on the data stored in the first address or the data stored in the second address includes: When checking of the first data based on the first check code fails and checking of the second data based on the second check code succeeds, the second data and the second check code overwrite the data stored in the first address.

[0029] In the foregoing implementation, that the checking of the first data based on the first check code fails indicates that data tearing occurs in the read first copy, in other words, the data tearing occurs in the data stored in the first address. That the checking of the second data based on the second check code succeeds indicates that the read second copy is complete and valid, that is, the data stored in the second address is complete and valid. Therefore, the second data and the second check code overwrite the data stored in the first address, so that the data stored in the first address is completely the same as the data stored in the second address. This implements the atomicity of the data writing.

[0030] Optionally, the method further includes: When detecting that the read first copy is consistent with the read second copy, the network interface card or the processor of the storage device determines that the read first copy and the read second copy are valid data.

[0031] In the foregoing implementation, the processor or the network interface card of the storage device may further verify validity or integrity of the first copy read from the first address and the second copy read from the second address, to obtain the valid data from the first copy and the second copy for use.

[0032] According to a third aspect, this application provides a data processing apparatus, where the apparatus includes: a generation unit, configured to generate a first copy and a second copy of data; and a writing unit, con-

figured to write the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral remote memory access RDMA manner.

[0033] Optionally, the memory is a persistent memory.

[0034] Optionally, the apparatus further includes: a reading unit, configured to read the first copy from the first address and the second copy from the second address in the unilateral RDMA manner; and a checking unit, configured to: check the read first copy and the read second copy, and determine valid data from the read first copy and the read second copy.

[0035] Optionally, the checking unit is specifically configured to: when detecting that the read first copy is the same as the read second copy, determine that the read first copy and the read second copy are the valid data.

[0036] According to a fourth aspect, this application provides a data processing apparatus, where the apparatus includes: a receiving unit, configured to receive a unilateral remote memory access RDMA write request sent by a client, where the RDMA write request includes a first copy and a second copy of data, a first address of the first copy in a memory of a storage device, and a second address of the second copy in the memory; and a writing unit, configured to write the first copy into the first address and the second copy into the second address.

[0037] Optionally, the memory is a persistent memory.

[0038] Optionally, the apparatus further includes: a reading unit, configured to read the first copy from the first address and the second copy from the second address; and a processing unit, configured to: check the read first copy and the read second copy, and when detecting that the read first copy is inconsistent with the read second copy, perform a data overwrite operation on data stored in the first address or data stored in the second address.

[0039] Optionally, the read first copy includes first data and first check code, the read second copy includes second data and second check code, and the read first copy is inconsistent with the read second copy when at least one of the following conditions is met: the first data is different from the second data, or the first check code is different from the second check code.

[0040] Optionally, the processing unit is specifically configured to: when checking of the first data based on the first check code succeeds, overwrite, by using the first data and the first check code, the data stored in the second address.

[0041] Optionally, the processing unit is specifically configured to: when checking of the first data based on the first check code fails and checking of the second data based on the second check code succeeds, overwrite, by using the second data and the second check code, the data stored in the first address.

[0042] Optionally, the processing unit is specifically configured to: when detecting that the read first copy is consistent with the read second copy, determine that the read first copy and the read second copy are valid data.

[0043] According to a fifth aspect, this application provides an apparatus, where the apparatus includes at least one processor and a network interface card, and the processor communicates with the network interface card. The processor or the network interface card is configured to generate a first copy and a second copy of data. The network interface card is configured to write the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral remote memory access RDMA manner.

[0044] Optionally, the memory is a persistent memory.

[0045] Optionally, the network interface card is further configured to read the first copy from the first address and the second copy from the second address in the unilateral RDMA manner. The processor or the network interface card is further configured to: check the read first copy and the read second copy, and determine valid data from the read first copy and the read second copy.

[0046] Optionally, that the processor or the network interface card is further configured to: check the read first copy and the read second copy, and determine valid data from the read first copy and the read second copy includes: When further detecting that the read first copy is the same as the read second copy, the processor or the network interface card determines that the read first copy and the read second copy are the valid data. The apparatus may be a chip or an integrated circuit, or may be the apparatus according to the third aspect.

[0047] According to a sixth aspect, this application provides an apparatus. The apparatus includes a network interface card and a memory. The network interface card is configured to receive a unilateral remote memory access RDMA write request sent by a client, where the unilateral RDMA write request includes a first copy and a second copy of data, a first address of the first copy in the memory, and a second address of the second copy in the memory. The network interface card is configured to write the first copy into the first address and the second copy into the second address.

[0048] Optionally, the memory is a persistent memory.

[0049] Optionally, the data processing apparatus further includes a processor. The processor communicates with the network interface card. The network interface card or the processor is configured to read the first copy from the first address and the second copy from the second address. The network interface card or the processor is further configured to: check the read first copy and the read second copy, and when detecting that the read first copy is inconsistent with the read second copy, perform a data overwrite operation on data stored in the first address or data stored in the second address.

[0050] Optionally, the read first copy includes first data and first check code, the read second copy includes second data and second check code, and the read first copy is inconsistent with the read second copy when at least one of the following conditions is met: the first data is

different from the second data, or the first check code is different from the second check code.

[0051] Optionally, that the data overwrite operation is performed on the data stored in the first address or the data stored in the second address includes: When checking of the first data based on the first check code succeeds, the first data and the first check code overwrite the data stored in the second address.

[0052] Optionally, that the data overwrite operation is performed on the data stored in the first address or the data stored in the second address includes: When checking of the first data based on the first check code fails and checking of the second data based on the second check code succeeds, the second data and the second check code overwrite the data stored in the first address.

[0053] Optionally, the network interface card or the processor is further configured to: when detecting that the read first copy is the same as the read second copy, determine that the read first copy and the read second copy are valid data. The apparatus may be a chip or an integrated circuit, or may be the apparatus according to the fourth aspect.

[0054] According to a seventh aspect, this application provides a computer-readable storage medium, including computer instructions. When the computer instructions are run by a processor, the method in any one of the first aspect or the possible implementations of the first aspect is implemented.

[0055] According to an eighth aspect, this application provides a computer-readable storage medium, including computer instructions. When the computer instructions are run by a processor, the method in any one of the second aspect or the possible implementations of the second aspect is implemented.

[0056] According to a ninth aspect, this application provides a computer program product. When the computer program product is executed by a processor, the method in any one of the first aspect or the possible implementations of the first aspect is implemented. For example, the computer program product may be a software installation package. If the method provided in any possible design of the first aspect needs to be used, the computer program product may be downloaded and executed on the processor, to implement the method in any one of the first aspect or the possible implementations of the first aspect.

[0057] According to a tenth aspect, this application provides a computer program product. When the computer program product is executed by a processor, the method in any one of the second aspect or the possible implementations of the second aspect is implemented. For example, the computer program product may be a software installation package. If the method provided in any possible design of the second aspect needs to be used, the computer program product may be downloaded and executed on the processor, to implement the method in any one of the second aspect or the possible implementations of the second aspect.

BRIEF DESCRIPTION OF DRAWINGS

[0058]

FIG. 1 is a schematic diagram of data tearing in a data writing process;

FIG. 2 is a schematic diagram of a system architecture according to an embodiment of this application;

FIG. 3 is a flowchart of a data processing method according to an embodiment of this application;

FIG. 4 is a schematic diagram of an interrupt event occurring before data is written into a first address according to an embodiment of this application;

FIG. 5 is a schematic diagram of an interrupt event occurring in a process of writing data into a first address according to an embodiment of this application;

FIG. 6 is a schematic diagram of an interrupt event occurring when data is about to be written into a second address according to an embodiment of this application;

FIG. 7 is a schematic diagram of an interrupt event occurring in a process of writing data into a second address according to an embodiment of this application;

FIG. 8 is a schematic diagram of an interrupt event occurring after data is written into a second address according to an embodiment of this application;

FIG. 9 is a schematic diagram of a unifying operation according to an embodiment of this application;

FIG. 10 is a schematic diagram of another unifying operation according to an embodiment of this application;

FIG. 11 is a flowchart of another data processing method according to an embodiment of this application;

FIG. 12 is a flowchart of still another data processing method according to an embodiment of this application;

FIG. 13 is a schematic diagram of a functional structure of an apparatus according to an embodiment of this application;

FIG. 14 is a schematic diagram of another functional structure of an apparatus according to an embodiment of this application; and

FIG. 15 is a schematic diagram of a structure of another device according to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

[0059] The terms used in embodiments of this application are merely for the purpose of illustrating specific embodiments, and are not intended to limit this application. In this specification and the claims in embodiments of this application, terms such as "first", "second", and the like are intended to distinguish between different objects but do not indicate a particular order of the objects.

[0060] When an event such as a power failure or a system breakdown occurs in a process of writing data into a storage medium (for example, a memory or a hard disk drive HDD), a part of the data is successfully written, and another part of the data fails to be written. As a result, the data in the storage medium is torn.

[0061] FIG. 1 is a schematic diagram of data tearing in a data writing process. As shown in FIG. 1, at a moment t1, data "1234" and check code "CRC0" are stored in a storage area 0. If a client intends to write data "4312" and check code "CRC1" into the storage area 0 at this moment, and a power failure occurs in a system after the check code "CRC1" and data "43" are successfully written at a moment t2, after the power failure, data stored in the storage area 0 changes to "4334", and stored check code is "CRC1". Therefore, it can be learned that only some data "43" in the data "4312" is successfully written. As a result, data tearing occurs.

[0062] After the client initiates a remote direct memory access (Remote Direct Memory Access, RDMA) write operation to the storage area 0 of a storage device to write the data "4312", if writing succeeds, the data "4312" and the check code "CRC1" are stored in the storage area 0, which is equivalent to that all data is written successfully. If the writing fails, it is expected that the storage area 0 still stores the data before the write operation, that is, the data "1234" and the check code "CRC0", which is equivalent to that all data fails to be written. If it may be ensured that all data is successfully written or not successfully written, that is, no data tearing occurs, atomicity of data writing is ensured.

[0063] In a possible embodiment, an atomicity problem of the data writing may be resolved in a two-time submission manner. For example, an application first writes to-be-written data/metadata into a log file instead of directly writing the data/metadata. In this process, the log file may be referred to as a log. The data/metadata in the log file is subsequently written for a second time to perform persistent writing. For example, writing of data/metadata in a target file is triggered once based on a preset periodicity. This process is referred to as a checkpoint or commit. If the power failure or the system breakdown occurs during this process, a file system may analyze the log after restart, and re-execute a commit operation that fails to be executed in the log until the commit is successfully executed.

[0064] It can be learned that, in the foregoing manner of resolving the atomicity problem of the data writing, I/O needs to be used for a plurality of times to execute writing of same data, and I/O overheads are large. In addition, an application in a distributed memory system causes a plurality of times of network round-trip consumption and reduces system performance.

[0065] For the foregoing problem, embodiments of this application provide a data writing method. Each time data needs to be written into memory space of the storage device, continuous dual writing is performed on the data through one-time I/O, and the atomicity of the data writing with less I/O consumption is ensured by using two pieces of memory space. This helps improve the system performance.

[0066] The following describes technical solutions of this application with reference to the accompanying drawings.

[0067] FIG. 2 shows an example of a schematic diagram of a system architecture. The system is used to implement atomic writing of data. As shown in FIG. 2, the system includes a client and a storage device. The client and the storage device are connected and communicate with each other by using a network. The network may be a wide area network, a local area network, or the like. This is not specifically limited in this embodiment of this application.

[0068] In specific implementation, the client is configured to send a unilateral RDMA write request to the storage device, where the unilateral RDMA write request indicates the storage device to respectively write to-be-written data into two addresses in a memory of the storage device continuously. The storage device is configured to: write the to-be-written data according to the unilateral RDMA write request, and perform processing such as checking on data in the two addresses, to ensure atomicity of writing the to-be-written data. It should be noted that the memory may be a persistent memory (Persistent Memory, PMEM).

[0069] In another specific implementation, the client may respectively write the to-be-written data in the two addresses into the memory (for example, the PMEM) of the storage device continuously in a unilateral RDMA manner, and the storage device is configured to perform the processing such as the checking on the data in the two addresses. This can reduce a data writing delay and improve system performance.

[0070] In some possible embodiments, the system shown in FIG. 2 may be a distributed persistent memory system. In this case, there are a plurality of clients shown in FIG. 2, and there are also a plurality of storage devices shown in FIG. 2. Any one of the plurality of clients may communicate with the plurality of storage devices, that is, the client may send a write instruction to at least one storage device.

[0071] It should be noted that FIG. 2 is merely an example of an architectural diagram, but a quantity of network elements included in the system shown in FIG. 2 is not limited. Although not shown in FIG. 2, another functional entity may further be included in FIG. 2 in addition to the functional entity shown in FIG. 2. In addition, the method provided in embodiments of this application may be applied to the communication system shown in FIG. 2. Certainly, the method provided in embodiments of this application may also be applied to another communication system. This is not limited in embodiments of this application.

[0072] When the client performs a data write operation on an address in the memory of the storage device, ensuring atomicity of data writing means: if the data write

operation is successfully performed, data in the address is target data; or if the data write operation fails to be performed, data stored in the address is still data stored before a moment at which the data write operation is performed. Based on this idea, a data writing method is proposed to implement the atomicity of the data writing based on as little I/O consumption as possible.

[0073] FIG. 3 is a flowchart of a data processing method according to an embodiment of this application. Atomicity of data writing can be implemented based on as few I/O overheads as possible, to reduce network round-trip overheads and improve system performance. The method includes but is not limited to the following steps.

[0074] S101: A client sends a unilateral RDMA write request to a storage device.

[0075] In this embodiment of this application, first target data is stored in a first address in a memory of the storage device, and the first target data is also stored in a second address. Data stored in the first address is the same as data stored in the second address. An address of the first address is different from an address of the second address.

[0076] In this embodiment of this application, the first target data includes data 1 and check code 1. The check code 1 corresponds to the data 1, the check code 1 uniquely indicates integrity and correctness of the data 1, and the data 1 may be checked based on the check code 1. For example, the first target data may be "1234 CRC1", where "1234" is referred to as the data 1, and "CRC1" is referred to as the check code 1.

[0077] In this embodiment of this application, the unilateral RDMA write request includes two copies of second target data (in other words, a first copy of the second target data and a second copy of the second target data), the first address of the first copy of the second target data in the memory of the storage device, and the second address of the second copy of the second target data in the memory. The unilateral RDMA write request indicates the storage device to write the first copy of the second target data into the first address and the second copy of the second target data into the second address. The two copies of the second target data are completely the same.

[0078] In this embodiment of this application, the second target data includes data 2 and check code 2. The data 2 in the second target data corresponds to the check code 2. In other words, the check code 2 uniquely indicates correctness and integrity of the data 2, and the data 2 may be checked based on the check code 2. For example, the second target data may be "4321 CRC2", where "4321" is referred to as the data 2, and "CRC2" is referred to as the check code 2.

[0079] In this embodiment of this application, a length of the second target data is the same as a length of the first target data, but content of the second target data is different from content of the first target data. Specifically, that the length of the second target data is the same as the length of the first target data means that a length of the data 2 is equal to a length of the data 1, and a length of the check code 2 is equal to a length of the check code 1. That the content of the second target data is different from the content of the first target data means that content of the data 2 is different from content of the data 1, and content of the check code 2 is different from content of the check code 1.

[0080] In this embodiment of this application, the storage device sequentially and respectively writes the two copies of the second target data into the first address and the second address in the memory of the storage device. Respectively writing means: writing the first copy of the second target data in the two copies of the second target data into the first address to overwrite the first target data in the first address, and writing the second copy of the second target data in the two copies of the second target data into the second address to overwrite the second target data in the second address. Sequentially writing means: first writing the first copy of the second target data into the first address, and after the first copy of the second target data is successfully written into the first address, writing the second copy of the second target data into the second address.

[0081] In this embodiment of this application, there may be two results of writing, by the storage device, the first copy of the second target data into the first address in the storage device. A first result is that the storage device successfully writes the first copy of the second target data into the first address. A second result is that the storage device fails to write the first copy of the second target data into the first address.

[0082] That the storage device successfully writes the first copy of the second target data into the first address means that the first copy of the second target data completely overwrites the first target data in the first address. Specifically, the data 2 in the first copy of the second target data completely overwrites the data 1 in the first target data in the first address, and the check code 2 in the first copy of the second target data completely overwrites the check code 1 in the first target data in the first address. Otherwise, the storage device fails to write the first piece of the second target data into the first address. Similarly, for a case in which the storage device successfully writes or fails to write the second copy of the second target data into the second address, respectively refer to related descriptions that the storage device successfully writes or fails to write the first copy of the second target data into the first address. Details are not described herein again.

[0083] In this embodiment of this application, an entity in the storage device that performs a data write operation may be specifically a network interface card of the storage device.

[0084] It should be noted that, in the foregoing example, a writing sequence of the data 2 and the check code 2 in the second target data is not limited. For example, the first copy of the second target data is written into the first address. In the first copy of the second target data, the data 2 may be written into the first address before

the check code 2, or the check code 2 may be written into the first address before the data 2, or the data 2 and the check code 2 may be written into the first address at the same time. This is not specifically limited in this embodiment of this application.

[0085] In this embodiment of this application, the client may perform a copy operation on the first copy of the second target data to obtain the second copy of the second target data.

[0086] S102: The storage device detects that an interrupt event occurs in a process of executing the unilateral RDMA write request.

[0087] In this embodiment of this application, the interrupt event may be a power failure event, a system breakdown event, a fault event, or the like.

[0088] In this embodiment of this application, when the interrupt event occurs in a process in which the storage device executes the unilateral RDMA write request, content stored in the first address and content stored in the second address include the following several cases.

[0089] Case 1: The interrupt event occurs before the first copy of the second target data is written into the first address. In this case, the content stored in the first address is the same as the content stored in the second address. Specifically, after the storage device receives the unilateral RDMA write request, and when the storage device is about to start to write the first copy of the second target data into the first address, the interrupt event occurs. In this case, both the content stored in the first address and the content stored in the second address remain the first target data.

[0090] For example, FIG. 4 is a schematic diagram of the interrupt event occurring before the data is written into the first address according to an embodiment of this application. In FIG. 4, before the unilateral RDMA write request is executed, the content stored in the first address is the first target data "1234 CRC1", and the content stored in the second address is also the first target data "1234 CRC1". If the interrupt event occurs when the first copy of the second target data "4321 CRC2" is written into the first address, after the interrupt event occurs, the content stored in the first address is still the first target data "1234 CRC1", and the content stored in the second address is also the first target data "1234 CRC1".

[0091] Case 2: The interrupt event occurs in a process of writing the first copy of the second target data into the first address. In this case, the content stored in the first address is different from the content stored in the second address. Specifically, when the storage device is writing a part of the first copy of the second target data into the first address, the interrupt event occurs. In this case, the first target data stored in the first address is partially changed, and the content stored in the second address remains the first target data.

[0092] If the storage device first writes the data 2 and then writes the check code 2 when writing the first copy of the second target data into the first address, that the first target data stored in the first address is partially changed may include the following cases: (1) The data 2 is partially written into the first address, and the check code 2 is not written into the first address. (2) All the data 2 is written into the first address, and the check code 2 is not written into the first address. (3) All the data 2 is written into the first address, and the check code 2 is partially written into the first address.

[0093] If the storage device first writes the check code 2 and then writes the data 2 when writing the first copy of the second target data into the first address, that the first target data stored in the first address is partially changed may include the following cases: (1) The check code 2 is partially written into the first address, and the data 2 is not written into the first address. (2) All the check code 2 is written into the first address, and the data 2 is not written into the first address. (3) All the check code 2 is written into the first address, and the data 2 is partially written into the first address.

[0094] If the writing sequence of the data 2 and the check code 2 in the first copy of the second target data is not limited when the storage device writes the first copy of the second target data into the first address, in addition to the foregoing cases, a case in which the data 2 is partially written into the first address and the check code 2 is partially written into the first address is further included.

[0095] For example, FIG. 5 is a schematic diagram of the interrupt event occurring in the process of writing the data into the first address according to an embodiment of this application. FIG. 5 is merely an example, and that there is only the form in FIG. 5 when the interrupt event occurs in the process of writing the data into the first address is not limited. In FIG. 5, before the unilateral RDMA write request is executed, the content stored in the first address is the first target data "1234 CRC1", and the content stored in the second address is also the first target data "1234 CRC1". If the interrupt event occurs in the process of writing the first copy of the second target data "4321 CRC2" into the first address, after the interrupt event occurs, the first target data "1234 CRC1" stored in the first address is partially changed to "4334 CRC1" (that is, the data 1 in the first address is partially changed), and the content stored in the second address remains the first target data "1234 CRC1".

[0096] Case 3: The interrupt event occurs after the first copy of the second target data is written into the first address and before the second copy of the target data is written into the second address. In this case, the content stored in the first address is different from the content stored in the second address. Specifically, after the storage device successfully writes the first copy of the second target data into the first address, and before the storage device is about to write the second copy of the second target data into the second address, the interrupt event occurs. In this case, the content stored in the first address is updated to the second target data. The second copy of the second target data is not written into the second address. Consequently, the content stored in the second

address is still the first target data.

[0097] For example, FIG. 6 is a schematic diagram of the interrupt event occurring when the data is about to be written into the second address according to an embodiment of this application. In FIG. 6, before the unilateral RDMA write request is executed, the content stored in the first address is the first target data "1234 CRC1", and the content stored in the second address is also the first target data "1234 CRC1". If the interrupt event occurs after the first copy of the second target data "4321 CRC2" is successfully written into the first address and before the second copy of the second target data is written into the second address, after the interrupt event occurs, the first target data "1234 CRC1" stored in the first address is updated to the second target data "4321 CRC2". Because the second copy of the second target data is not written into the second address, the content stored in the second address remains the first target data "1234 CRC1".

[0098] Case 4: The interrupt event occurs in a process of writing the second copy of the second target data into the second address. In this case, the content stored in the first address is different from the content stored in the second address. Specifically, when the storage device successfully writes the first copy of the second target data into the first address, and then is writing a part of the second copy of the second target data into the second address, the interrupt event occurs. In this case, the content stored in the first address is updated from the first target data to the second target data, and the first target data stored in the second address is partially changed.

[0099] If the storage device first writes the data 2 and then writes the check code 2 when writing the second copy of the second target data into the second address, that the second target data stored in the second address is partially changed may include the following cases: (1) The data 2 is partially written into the second address, and the check code 2 is not written into the second address. (2) All the data 2 is written into the second address, and the check code 2 is not written into the second address. (3) All the data 2 is written into the second address, and the check code 2 is partially written into the second address.

[0100] If the storage device first writes the check code 2 and then writes the data 2 when writing the second copy of the second target data into the second address, that the second target data stored in the second address is partially changed may include the following cases: (1) The check code 2 is partially written into the second address, and the data 2 is not written into the second address. (2) All the check code 2 is written into the second address, and the data 2 is not written into the second address. (3) All the check code 2 is written into the second address, and the data 2 is partially written into the second address.

[0101] If the writing sequence of the data 2 and the check code 2 in the second copy of the second target data is not limited when the storage device writes the second copy of the second target data into the second address, in addition to the foregoing cases, a case in which the data 2 is partially written into the second address and the check code 2 is partially written into the second address is further included.

[0102] For example, FIG. 7 is a schematic diagram of the interrupt event occurring in the process of writing the data into the second address according to an embodiment of this application. FIG. 7 is merely an example, and that there is only the form in FIG. 7 when the interrupt event occurs in the process of writing the data into the second address is not limited. In FIG. 7, before the unilateral RDMA write request is executed, the content stored in the first address is the first target data "1234 CRC1", and the content stored in the second address is also the first target data "1234 CRC1". If the interrupt event occurs in the process of writing the second copy of the second target data "4321 CRC2" into the second address, after the interrupt event occurs, the first target data "1234 CRC1" stored in the first address is updated to the second target data "4321 CRC2", and the content stored in the second address is partially changed to "4321 CRC1".

[0103] Case 5: The interrupt event occurs after the second copy of the second target data is written into the second address. In this case, the content stored in the first address is the same as the content stored in the second address. It may be understood that, in this case, both the two copies of the second target data are successfully written.

[0104] For example, FIG. 8 is a schematic diagram of the interrupt event occurring after the data is written into the second address according to an embodiment of this application. In FIG. 8, before the unilateral RDMA write request is executed, the content stored in the first address is the first target data "1234 CRC1", and the content stored in the second address is also the first target data "1234 CRC1". If the interrupt event occurs after the second copy of the second target data "4321 CRC2" is written into the second address, after the interrupt event occurs, the first target data "1234 CRC1" stored in the first address is updated to the second target data "4321 CRC2". The first target data "1234 CRC1" stored in the second address is also updated to the second target data "4321 CRC2". In other words, the unilateral RDMA write request is successfully executed.

[0105] S103: The storage device detects the content stored in the first address and the content stored in the second address.

[0106] In this embodiment of this application, a processor or the network interface card of the storage device detects the content stored in the first address and the content stored in the second address.

[0107] In this embodiment of this application, after the interrupt event occurs, the storage device detects whether the content in the first address is the same as the content in the second address. If it is detected that the content in the first address is the same as the content in

the second address, it indicates that a case in which the two copies of the second target data are written is the foregoing case 1 or 5, it is proved that the atomicity of the data writing is ensured, no additional processing is needed, and a procedure ends. If it is detected that the content stored in the first address is different from the content in the second address, it indicates that a case in which the two copies of the second target data are written is any one of the foregoing cases 2 to 4, it is proved that the atomicity of the data writing cannot be ensured, additional processing is needed, and step S104 is performed.

[0108] In this embodiment of this application, detecting whether the content in the first address is the same as the content in the second address is specifically: comparing whether data currently stored in the first address is the same as data currently stored in the second address, and whether check code currently stored in the first address is the same as check code currently stored in the second address.

[0109] That the content stored in the first address is the same as the content in the second address means that the data currently stored in the first address is the same as the data currently stored in the second address, and the check code currently stored in the first address is the same as the check code currently stored in the second address. That the content stored in the first address is different from the content in the second address means that the data currently stored in the first address is different from the data currently stored in the second address, and/or the check code currently stored in the first address is different from the check code stored in the second address.

[0110] For ease of description, the data currently stored in the first address is referred to as data 3, and the check code currently stored in the first address is referred to as check code 3. The data currently stored in the second address is referred to as data 4, and the check code currently stored in the second address is referred to as check code 4.

[0111] For the foregoing cases 2 to 4, that is, when it is detected that the content stored in the first address is different from the content stored in the second address, the content stored in the first address needs to be further checked to determine whether data tearing occurs in the content stored in the first address.

[0112] In the case 2 in which the first target data stored in the first address is partially changed but the content stored in the second address remains the first target data because the interrupt event occurs in the process of writing the first copy of the second target data into the first address in S102, the storage device checks the content stored in the first address, specifically, checks the data 3 based on the check code 3. Because the first target data stored in the first address is partially changed in the case 2, any one of the following cases may occur: (1) The data 3 is different from the data 1, but the check code 3 is the same as the check code 1. (2) The data 3 is the same as the data 1, but the check code 3 is different from the check code 1. (3) The data 3 is different from the data 1, and the check code 3 is different from the check code 1. Therefore, for any one of the foregoing cases, detecting of the data 3 based on the check code 3 fails, and it indicates that the data tearing occurs in the content stored in the first address. It may be understood that, because the content stored in the second address remains the first target data in the case 2, it means that the data 4 is the data 1, and the check code 4 is the check code 1. Therefore, detecting of the data 4 based on the check code 4 succeeds, and it indicates that no data tearing occurs in the content stored in the second address.

[0113] In the case 3 in which the content stored in the first address is updated to the second target data but the content in the second address remains the first target data because the interrupt event occurs after the first copy of the second target data is written into the first address and before the second copy of the second target data is written into the second address in S102, the storage device checks the content stored in the first address, that is, checks the data 3 based on the check code 3. Because the content stored in the first address has been updated to the second target data, the data 3 is the same as the data 2, and the check code 3 is the same as the check code 2. Therefore, checking of the data 3 based on the check code 3 succeeds, and it indicates that no data tearing occurs in the content stored in the first address.

[0114] In the case 4 in which content stored in the first address is updated to the second target data but the first target data stored in the second address is partially changed because the interrupt event occurs in the process of writing the second copy of the second target data into the second address in S102, the storage device checks the content stored in the first address, that is, checks the data 3 based on the check code 3. Because the content stored in the first address is updated to the second target data, the data 3 is the same as the data 2, and the check code 3 is the same as the check code 2. Therefore, checking of the data 3 based on the check code 3 succeeds, and it indicates that no data tearing occurs in the content stored in the first address.

[0115] In conclusion, if detecting that the content stored in the first address is different from the content stored in the second address, the storage device further detects whether the data tearing occurs in the content stored in the first address. A detection result shows that: For the case 2 in S102, the data tearing occurs in the content stored in the first address. For the case 3 and the case 4 in S102, no data tearing occurs in the content stored in the first address.

[0116] S104: When detecting that the content stored in the first address is different from the content in the second address, the storage device performs a unifying operation on the content stored in the first address and the content stored in the second address.

[0117] When the content stored in the first address is

different from the content stored in the second address, the storage device performs the unifying operation on the content stored in the first address and the content stored in the second address, so that the content stored in the first address is the same as the content stored in the second address. This ensures the atomicity of the data writing. It should be noted that in this embodiment of this application, the unifying operation may also be referred to as a data overwrite operation.

[0118] In specific implementation, if the content stored in the first address is different from the content stored in the second address, and the data tearing occurs in the content stored in the first address and no data tearing occurs in the content stored in the second address, performing the unifying operation on the content stored in the first address and the content stored in the second address is specifically: If the content stored in the second address overwrites the content stored in the first address, and the content stored in the second address remains the first target data, the content stored in the first address is restored to the first target data. In other words, both the content stored in the first address and the content stored in the second address are the data 1 and the check code 1.

[0119] For example, FIG. 9 is a schematic diagram of the unifying operation according to an embodiment of this application. In FIG. 9, it is assumed that after the interrupt event occurs, the content stored in the first address is "4334 CRC1", in other words, the first target data stored in the first address is partially changed, and the content stored in the second address remains the first target data "1234 CRC1". Because the content stored in the first address is different from the content stored in the second address, and it is detected that the data tearing occurs in the content stored in the first address, the unifying operation is performed. The content stored in the second address is used to overwrite the content stored in the first address. In this case, both the content stored in the first address and the content stored in the second address are the first target data "1234 CRC1".

[0120] In specific implementation, if the content stored in the first address is different from the content stored in the second address, and no data tearing occurs in the content stored in the first address, performing the unifying operation on the content stored in the first address and the content stored in the second address is specifically: If the content stored in the first address overwrites the content stored in the second address, and the content stored in the first address is the second target data, the content stored in the second address is updated to the second target data. In other words, both the content stored in the first address and the content stored in the second address are the data 2 and the check code 2.

[0121] For example, FIG. 10 is a schematic diagram of the unifying operation according to an embodiment of this application. In FIG. 10, it is assumed that after the interrupt event occurs, the content stored in the first address is "4321 CRC2", that is, the content stored in the first address is updated to the second target data, and the content stored in the second address is "4321 CRC1", that is, the first target data stored in the second address is partially changed. Because the content stored in the first address is different from the content stored in the second address, and it is detected that no data tearing occurs in the content stored in the first address, the unifying operation is performed. The content stored in the first address is used to overwrite the content stored in the second address. In this case, both the content stored in the first address and the content stored in the second address are the second target data "4321 CRC2".

[0122] It can be learned that, in this embodiment of this application, continuous dual writing is performed on to-be-written data based on one-time I/O in two addresses that are in the memory of the storage device and that store same content. When a data writing process is interrupted, the unifying operation is performed based on a status of writing the data into the two addresses, so that the content stored in the two addresses is the same. This implements the atomicity of the data writing, reduces I/O consumption, and improves the system performance.

[0123] In conclusion, it can be learned that the atomicity of the data writing may be implemented by sequentially writing the to-be-written data into the two addresses in the memory of the storage device in serial. To reduce a data writing delay and further improve data writing efficiency, the client may further sequentially write the to-be-written data into the two addresses in the memory of the storage device by using a unilateral RDMA technology.

[0124] It should be noted that data in one computer may be transmitted to another computer based on the RDMA technology by using a network without intervention of operating systems of both parties, to eliminate overheads of copying and moving a data packet in user space and kernel space and context switching, and reduce consumption of bandwidth and computing power of the memory.

[0125] FIG. 11 is a flowchart of another data processing method according to an embodiment of this application. Compared with the embodiment in FIG. 3, in FIG. 11, a data write operation is performed by a client instead of the storage device in the embodiment in FIG. 3. FIG. 11 may be independent of the embodiment of FIG. 3. The method includes but is not limited to the following steps.

[0126] S201: The client sequentially writes two copies of second target data into a first address and a second address in a memory of the storage device based on a unilateral RDMA technology.

[0127] In this embodiment of this application, before the client writes the two copies of the second target data into the storage device, first target data is stored in the first address in the memory of the storage device, and the first target data is also stored in the second address. In other words, content stored in the first address is completely the same as content stored in the second address.

It should be noted that for details of the first target data and the second target data, refer to related descriptions of the first target data and the second target data in S101 in the embodiment in FIG. 3. Details are not described herein again.

[0128] In this embodiment of this application, the first address is different from the second address.

[0129] In this embodiment of this application, the client sequentially and respectively writes the two copies of the second target data into the first address and the second address in the memory of the storage device by using the unilateral RDMA technology. Respectively writing means that the client writes a first copy of the second target data in the two copies of the second target data into the first address in the memory of the storage device to overwrite the first target data in the first address and a second copy of the second target data in the two copies of second target data into the second address in the memory of the storage device to overwrite the first target data in the second address. Sequentially writing means that the client first writes the first copy of the second target data into the first address, and after the first copy of the second target data is successfully written into the first address, the client writes the second copy of the second target data into the second address. It should be noted that, for specific descriptions about a data writing success or failure, refer to related descriptions of S101 in the embodiment in FIG. 3. Details are not described herein again.

[0130] In this embodiment of this application, an entity in the client that performs the data write operation may be specifically a network interface card of the client.

[0131] It may be understood that, that the client uses the unilateral RDMA technology means that a data write/read operation is performed by the client. In an actual communication process, the client writes data into a target address in the memory of the storage device, and in the data writing process, the storage device does not need to perform any operation. It should be noted that, before using the unilateral RDMA technology, the client pre-obtains the two addresses (for example, the first address and the second address) in the memory of the storage device and permission to use the two addresses.

[0132] In specific implementation, the second copy of the second target data in the two copies of the second target data may be obtained by the client by copying the first copy of the second target data. The two copies of the second target data are written into the first address and the second address by the client based on one-time I/O consumption.

[0133] S202: If an interrupt event occurs in the data writing process, the storage device detects the content stored in the first address and the content stored in the second address.

[0134] In this embodiment of this application, the interrupt event occurs in the data writing process performed by the client. In this case, for details of cases of writing the two copies of the second target data into the first address and the second address in the memory of the storage device, refer to related descriptions of the cases 1 to 5 in S102 in the embodiment in FIG. 3. Details are not described herein again. It should be noted that, a difference from S102 lies in that an execution body of data writing is changed from the storage device in S102 to the client.

[0135] In this embodiment of this application, for a specific process in which the storage device detects the content in the first address and the content in the second address, refer to related descriptions of S103 in FIG. 3. For brevity of this specification, details are not described herein again.

[0136] S203: When detecting that the content stored in the first address is different from the content stored in the second address, the storage device performs a unifying operation on the content stored in the first address and the content stored in the second address. For details of this step, refer to related descriptions of S 104 in FIG. 3. For brevity of the specification, details are not described herein again.

[0137] It can be learned that, in this embodiment of this application, to-be-written data is continuously dual written into the two addresses in the memory of the storage device based on one-time I/O by using the unilateral RDMA technology. This can effectively reduce a data writing delay, and improve data writing efficiency. When the data writing process is interrupted, the unifying operation is performed based on a status of writing data into the two addresses, so that the content stored in the two addresses is the same. This implements atomicity of data writing, reduces I/O consumption, and helps improve system performance.

[0138] FIG. 12 is a flowchart of still another data processing method according to an embodiment of this application. The embodiment described in FIG. 12 may be independent of embodiments in FIG. 3 and FIG. 11, or may be a supplement to embodiments in FIG. 3 and FIG. 11. The method includes but is not limited to the following steps.

[0139] S301: A processor or a network interface card of a client generates a first copy and a second copy of data.

[0140] In this embodiment of this application, the first copy is completely the same as the second copy. It should be noted that data is equivalent to the second target data in the embodiment in FIG. 3 or FIG. 11, where the first copy is the first copy of the second target data in the embodiment in FIG. 3 or FIG. 11, and the second copy is the second copy of the second target data in the embodiment in FIG. 3 or FIG. 11.

[0141] S302: The network interface card of the client writes the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral RDMA manner.

[0142] In specific implementation, the storage device is independent of the client.

[0143] In specific implementation, the memory is a persistent memory.

[0144] In this embodiment of this application, the network interface card of the client writes the first copy into the first address in the memory of the storage device and the second copy into the second address in the memory in the unilateral RDMA manner. In other words, an execution body of data writing is the client. For details of this step, refer to related descriptions of S201 in the embodiment in FIG. 11. Details are not described herein again.

[0145] It should be noted that, for details of a status of writing the first copy into the first address and a status of writing the second copy into the second address, refer to related descriptions of the cases 1 to 6 in S102 in the embodiment in FIG. 3. Details are not described herein again.

[0146] In some possible embodiments, S302 may not be performed, and S3021 and S3022 are performed to replace S302.

[0147] S3021: The client sends a unilateral RDMA write request to the storage device.

[0148] Optionally, after S301, S3021 may be performed.

[0149] In this embodiment of this application, the client sends the unilateral RDMA write request to the storage device. Correspondingly, the network interface card of the storage device receives the unilateral RDMA write request.

[0150] The unilateral RDMA write request includes the first copy and the second copy of the data, the first address of the first copy in the memory of the storage device, and the second address of the second copy in the memory. It may be understood that the unilateral RDMA write request indicates to write the first copy into the first address in the memory of the storage device and the second copy into the second address in the memory.

[0151] S3022: The network interface card of the storage device writes the first copy into the first address and the second copy into the second address according to the unilateral RDMA write request.

[0152] It may be understood that S3021 and S3022 also implement writing the first copy of the data into the first address in the memory and the second copy of the data into the second address in the memory. Different from S302, in S3021 and S3022, the execution body of the data writing is the storage device. For details of this embodiment, refer to related descriptions of S101 in the embodiment in FIG. 3. Details are not described herein again.

[0153] S303: The processor or the network interface card of the storage device reads the first copy from the first address and the second copy from the second address, checks the read first copy and the read second copy, and when detecting that the read first copy is inconsistent with the read second copy, performs a data overwrite operation on data stored in the first address or data stored in the second address.

[0154] In this embodiment of this application, the read first copy includes first data and first check code, the read second copy includes second data and second check code, and the read first copy is inconsistent with the read second copy when at least one of the following conditions is met: the first data is different from the second data, or the first check code is different from the second check code. For a reason why the read first copy is inconsistent with the read second copy, refer to any one of the cases 2 to 4 in S 102 in the embodiment in FIG. 3.

[0155] In specific implementation, that when it is detected that the read first copy is inconsistent with the read second copy, the data overwrite operation is performed on the data stored in the first address or the data stored in the second address may be: When it is detected that the read first copy is inconsistent with the read second copy, and checking of the first data based on the first check code succeeds, the first data and the first check code overwrite the data stored in the second address. This embodiment corresponds to that the content stored in the first address overwrites the content in the second address in S104 in the embodiment in FIG. 3.

[0156] In specific implementation, that when it is detected that the read first copy is inconsistent with the read second copy, the data overwrite operation is performed on the data stored in the first address or the data stored in the second address may be: When it is detected that the read first copy is inconsistent with the read second copy, and checking of the first data based on the first check code fails and checking of the second data based on the second check code succeeds, the second data and the second check code overwrite the data stored in the first address. This embodiment corresponds to that the content stored in the second address overwrites the content stored in the first address in S 104 in the embodiment in FIG. 3.

[0157] It should be noted that the data overwrite operation is the unifying operation in the embodiment in FIG. 3 or FIG. 11, and an objective of the data overwrite operation is to make the data stored in the first address completely the same as the data stored in the second address.

[0158] It should be noted that, a quantity of copies of the data generated by the processor or the network interface card of the client is not limited in this embodiment of this application.

[0159] In specific implementation, the processor or the network interface card of the client may also generate three copies of the data, which are respectively the first copy, the second copy, and a third copy. Correspondingly, the network interface card of the client may write the first copy into the first address in the memory of the storage device, the second copy into the second address in the memory, and the third copy into a third address in the memory in the one-time unilateral RDMA manner. Alternatively, the client may send the unilateral RDMA write request to the storage device to indicate the storage device to write the first copy into the first address in the memory, the second copy into the first address in the

memory, and the third copy into the third address in the memory according to the RDMA write request. This is not specifically limited in this embodiment of this application. The processor or the network interface card of the storage device reads the first copy from the first address, the second copy from the second address, and the third copy from the third address, and when detecting that two copies in the read first copy, the read second copy, and the read third copy are inconsistent, performs the data overwrite operation on at least one of the data stored in the first address, the data stored in the second address, or data stored in the third address, so that the data stored in the three addresses is the same.

[0160] S304: The network interface card of the client reads the first copy from the first address and the second copy from the second address in the unilateral RDMA manner.

[0161] S305: The processor or the network interface card of the client checks the read first copy and the read second copy, and determines valid data from the read first copy and the read second copy.

[0162] In this embodiment of this application, when detecting that the read first copy is the same as the read second copy, the processor or the network interface card of the client determines that the read first copy and the read second copy are the valid data.

[0163] In specific implementation, the read first copy is the same as the first copy of the data, and the read second copy is the same as the second copy of the data. In this case, it indicates that the first copy of the data has been successfully written into the first address in the memory, and the second copy of the data has been successfully written into the second address in the memory. In this way, the processor or the network interface card of the client determines that the read first copy and the read second copy are the valid data. This ensures atomicity of the data writing. It should be noted that, for details of this embodiment, refer to the case 5 in S102 in the embodiment in FIG. 3.

[0164] In another specific implementation, the read first copy is the same as initial data stored in the first address in the memory, and the read second copy is the same as initial data stored in the second address in the memory, where the initial data stored in the first address is data stored in the first address before the first copy of the data is written into the first address in the memory, the initial data stored in the second address is data stored in the second address before the second copy of the data is written into the second address in the memory, and the initial data stored in the first address is the same as the initial data stored in the second address. In this way, the processor or the network interface card of the client determines that the read first copy and the read second copy are the valid data. This ensures the atomicity of the data writing. It should be noted that the initial data stored in the first address may be the first target data in the embodiment in FIG. 3, and the initial data stored in the second address is also the first target data in the embodiment in FIG. 3. It should be noted that, for details of this embodiment, refer to the case 1 in S102 in the embodiment in FIG. 3.

[0165] It can be learned that, in this embodiment of this application, the two copies of the data are written into the two addresses in the memory of the storage device in the one-time RDMA manner. When it is detected that the data stored in the two memory addresses is inconsistent, the data overwrite operation is performed on the data in a corresponding memory address, so that the data stored in the two memory addresses is the same. This ensures the atomicity of the data writing based on as little I/O consumption as possible, reduces a data writing delay, and improves performance of a storage system.

[0166] FIG. 13 is a schematic diagram of a functional structure of an apparatus according to an embodiment of this application. The apparatus 30 includes a generation unit 310 and a writing unit 312. The apparatus 30 may be implemented by hardware, software, or a combination of software and hardware.

[0167] The generation unit 310 is configured to generate a first copy and a second copy of data. The writing unit 312 is configured to write the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral remote memory access RDMA manner.

[0168] In some possible embodiments, the apparatus 30 further includes a checking unit 314 and a reading unit 316. The reading unit 316 is configured to read the first copy from the first address and the second copy from the second address in the unilateral RDMA manner. The checking unit 314 is configured to: check the read first copy and the read second copy, and determine valid data from the read first copy and the read second copy.

[0169] Functional modules of the apparatus 30 may be configured to implement the method described in the embodiment in FIG. 12. In the embodiment in FIG. 12, the generation unit 310 may be configured to perform S301, and the writing unit 312 may be configured to perform S302. The reading unit 316 may be configured to perform S304, and the checking unit 314 may be configured to perform S305. The functional modules of the apparatus 30 may be further configured to implement the methods described in embodiments in FIG. 3 and FIG. 11. For brevity of the specification, details are not described herein again.

[0170] FIG. 14 is a schematic diagram of a functional structure of an apparatus according to an embodiment of this application. The apparatus 40 includes a receiving unit 410 and a writing unit 412. The apparatus 40 may be implemented by hardware, software, or a combination of software and hardware.

[0171] The receiving unit 410 is configured to receive a unilateral remote memory access RDMA write request sent by a client, where the unilateral RDMA write request includes a first copy and a second copy of data, a first address of the first copy in a memory of a storage device, and a second address of the second copy in the memory.

The writing unit 412 is configured to write the first copy into the first address and the second copy into the second address.

[0172] In some possible embodiments, the apparatus 40 further includes a processing unit 414 and a reading unit 416. The reading unit 416 is configured to read the first copy from the first address and the second copy from the second address. The processing unit 414 is configured to: check the read first copy and the read second copy, and when detecting that the read first copy is inconsistent with the read second copy, perform a data overwrite operation on data stored in the first address or data stored in the second address.

[0173] Functional modules of the apparatus 40 may be configured to implement the method described in the embodiment in FIG. 12. In the embodiment in FIG. 12, the receiving unit 410 and the writing unit 412 may be configured to perform S3022, and the reading unit 416 and the processing unit 414 may be configured to perform S303. The functional modules of the apparatus 40 may be configured to implement the methods described in embodiments in FIG. 3 and FIG. 11. For brevity of the specification, details are not described herein again.

[0174] An embodiment of this application further provides a device. As shown in FIG. 15, the device 50 includes at least a processor 501, a network interface card 502, and a bus 503. The processor 501 and the network interface card 502 communicate with each other through the bus 503. The device 50 may be the foregoing client or storage device. It should be understood that a quantity of processors in the device 50 is not limited in this application.

[0175] The bus 503 may be a peripheral component interconnect (peripheral component interconnect, PCI) bus, an extended industry standard architecture (extended industry standard architecture, EISA) bus, or the like. The bus may be classified in an address bus, a data bus, a control bus, and the like. For ease of representation, only one line is used to represent the bus in FIG. 15, but this does not mean that there is only one bus or only one type of bus. The bus 503 may include a path for transmitting information between components (for example, the network interface card 502 and the processor 501) of the device 50.

[0176] The processor 501 may include any one or more of processors such as a central processing unit (central processing unit, CPU), a microprocessor (microprocessor, MP), or a digital signal processor (digital signal processor, DSP).

[0177] When the device 50 is the foregoing client, the network interface card 502 may be configured to generate a first copy and a second copy of data, and is further configured to write the first copy into a first address in a memory of the storage device and the second copy into a second address in the memory in a one-time unilateral remote memory access RDMA manner. When the device 50 is the foregoing storage device, the network interface card 502 may be configured to receive a unilateral remote memory access RDMA write request sent by the client, where the unilateral RDMA write request includes the first copy and the second copy of the data, the first address of the first copy in the memory of the storage device, and the second address of the second copy in the memory. The network interface card 502 is further configured to write the first copy into the first address and the second copy into the second address.

[0178] In specific implementation, when the device 50 is the foregoing client, modules of the device 50 are configured to perform the method on a client side described in the embodiment in FIG. 3, FIG. 11, or FIG. 12.

[0179] In a possible design manner, for example, the device 50 may be one or more modules in the client that performs the method shown in FIG. 12, and the device 50 is configured to perform the following operations:

generating the first copy and the second copy of the data by using the processor 501 or the network interface card 502; and

writing the first copy into the first address in the memory of the storage device and the second copy into the second address in the memory by using the network interface card 502 in the one-time unilateral remote memory access RDMA manner.

[0180] In another specific implementation, when the device 50 is the foregoing storage device, the device 50 further includes a memory 504, and the memory 504 may be a persistent memory (Persistent Memory, PMEM) or the like. The modules of the device 50 may be configured to perform the method on a storage device side described in the embodiment in FIG. 3, FIG. 11, or FIG. 12.

[0181] In a possible design manner, for example, the device 50 may be one or more modules in the storage device that performs the method shown in FIG. 12, and the device 50 is configured to perform the following operations:

receiving, by using the network interface card 502, the unilateral remote memory access RDMA write request sent by the client, where the unilateral RDMA write request includes the first copy and the second copy of the data, the first address of the first copy in the memory 504 of the storage device, and the second address of the second copy in the memory 504; and

writing the first copy into the first address and the second copy into the second address by using the network interface card 502.

[0182] In the foregoing embodiments in this specification, the descriptions of each embodiment have respective focuses. For a part that is not described in detail in an embodiment, refer to related descriptions in another embodiment.

[0183] It should be noted that a person of ordinary skill in the art may see that, all or a part of the steps in each

method of the foregoing embodiments may be implemented by a program instructing related hardware. The program may be stored in a computer-readable storage medium. The storage medium includes a read-only memory (Read-Only Memory, ROM), a random access memory (Random Access Memory, RAM), a programmable read-only memory (Programmable Read-only Memory, PROM), an erasable programmable read-only memory (Erasable Programmable Read-Only Memory, EPROM), a one-time programmable read-only memory (One-time Programmable Read-Only Memory, OTPROM), an electrically-erasable programmable read-only memory, (Electrically-Erasable Programmable Read-Only Memory, EEPROM), a compact disc read-only memory (Compact Disc Read-Only Memory, CD-ROM), or another optical disk memory, magnetic disk memory, magnetic tape memory, or any other computer-readable medium that can be configured to carry or store data.

[0184] The technical solutions of this application essentially, or the part contributing to the conventional technology, or all or some of the technical solutions may be implemented in a form of a software product. A computer program product is stored in the storage medium and includes several instructions for instructing a device (which may be a personal computer, a server, or a network device, a robot, a single-chip microcomputer, a chip, a robot, or the like) to perform all or some of the steps of the methods described in embodiments of this application.

Claims

1. A data processing method, wherein the method comprises:

   generating, by a processor or a network interface card of a client, a first copy and a second copy of data, wherein the processor communicates with the network interface card; and
   writing, by the network interface card, the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral remote direct memory access RDMA manner.

2. The method according to claim 1, wherein the memory is a persistent memory.

3. The method according to claim 1 or 2, wherein the method further comprises:

   reading, by the network interface card, the first copy from the first address and the second copy from the second address in the unilateral RDMA manner; and
   checking, by the processor or the network interface card, the read first copy and the read second copy, and determining valid data from the read first copy and the read second copy.

4. The method according to claim 3, wherein the checking, by the processor or the network interface card, the read first copy and the read second copy, and determining valid data from the read first copy and the read second copy comprises:
   when detecting that the read first copy is the same as the read second copy, determining, by the processor or the network interface card, that the read first copy and the read second copy are the valid data.

5. A data processing method, wherein the method comprises:

   receiving, by a network interface card of a storage device, a unilateral remote memory access RDMA write request sent by a client, wherein the unilateral RDMA write request comprises a first copy and a second copy of data, a first address of the first copy in a memory of the storage device, and a second address of the second copy in the memory; and
   writing, by the network interface card, the first copy into the first address and the second copy into the second address.

6. The method according to claim 5, wherein the memory is a persistent memory.

7. The method according to claim 5 or 6, wherein the method further comprises:

   reading, by the network interface card or a processor of the storage device, the first copy from the first address and the second copy from the second address; and
   checking, by the network interface card or the processor of the storage device, the read first copy and the read second copy, and when detecting that the read first copy is inconsistent with the read second copy, performing a data overwrite operation on data stored in the first address or data stored in the second address.

8. The method according to claim 7, wherein the read first copy comprises first data and first check code, the read second copy comprises second data and second check code, and the read first copy is inconsistent with the read second copy when at least one of the following conditions is met:

   the first data is different from the second data; or
   the first check code is different from the second check code.

9. The method according to claim 8, wherein the per-

forming a data overwrite operation on data stored in the first address or data stored in the second address comprises:
when checking of the first data based on the first check code succeeds, overwriting, by using the first data and the first check code, the data stored in the second address.

10. The method according to claim 8, wherein the performing a data overwrite operation on data stored in the first address or data stored in the second address comprises:
when checking of the first data based on the first check code fails and checking of the second data based on the second check code succeeds, overwriting, by using the second data and the second check code, the data stored in the first address.

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

a generation unit, configured to generate a first copy and a second copy of data; and
a writing unit, configured to write the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral remote memory access RDMA manner.

12. The apparatus according to claim 11, wherein the memory is a persistent memory.

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

a reading unit, configured to read the first copy from the first address and the second copy from the second address in the unilateral RDMA manner; and
a checking unit, configured to: check the read first copy and the read second copy, and determine valid data from the read first copy and the read second copy.

14. The apparatus according to claim 13, wherein the checking unit is specifically configured to:
when detecting that the read first copy is the same as the read second copy, determine that the read first copy and the read second copy are the valid data.

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

a receiving unit, configured to receive a unilateral remote memory access RDMA write request sent by a client, wherein the unilateral RDMA write request comprises a first copy and a second copy of data, a first address of the first copy in a memory of a storage device, and a second address of the second copy in the memory; and
a writing unit, configured to write the first copy into the first address and the second copy into the second address.

16. The apparatus according to claim 15, wherein the memory is a persistent memory.

17. The apparatus according to claim 15 or 16, wherein the apparatus further comprises:

a reading unit, configured to read the first copy from the first address and the second copy from the second address; and
a processing unit, configured to: check the read first copy and the read second copy, and when detecting that the read first copy is inconsistent with the read second copy, perform a data overwrite operation on data stored in the first address or data stored in the second address.

18. The apparatus according to claim 17, wherein the read first copy comprises first data and first check code, the read second copy comprises second data and second check code, and the read first copy is inconsistent with the read second copy when at least one of the following conditions is met:

the first data is different from the second data; or
the first check code is different from the second check code.

19. The apparatus according to claim 18, wherein the processing unit is specifically configured to:
when checking of the first data based on the first check code succeeds, overwrite, by using the first data and the first check code, the data stored in the second address.

20. The apparatus according to claim 18, wherein the processing unit is specifically configured to:
when checking of the first data based on the first check code fails and checking of the second data based on the second check code succeeds, overwrite, by using the second data and the second check code, the data stored in the first address.

21. A data processing apparatus, wherein the apparatus comprises at least one processor and a network interface card, and the processor communicates with the network interface card;

the processor or the network interface card is configured to generate a first copy and a second copy of data; and

the network interface card is configured to write the first copy into a first address in a memory of a storage device and the second copy into a second address in the memory in a one-time unilateral remote memory access RDMA manner.

22. The data processing apparatus according to claim 21, wherein the memory is a persistent memory.

23. The data processing apparatus according to claim 21 or 22, wherein

the network interface card is further configured to read the first copy from the first address and the second copy from the second address in the unilateral RDMA manner; and
the processor or the network interface card is further configured to: check the read first copy and the read second copy, and determine valid data from the read first copy and the read second copy.

24. The data processing apparatus according to claim 23, wherein that the processor or the network interface card is further configured to: check the read first copy and the read second copy, and determine valid data from the read first copy and the read second copy comprises:
when further detecting that the read first copy is the same as the read second copy, the processor or the network interface card determines that the read first copy and the read second copy are the valid data.

25. A data processing apparatus, wherein the apparatus comprises a network interface card and a memory, wherein

the network interface card is configured to receive a unilateral remote memory access RDMA write request sent by a client, wherein the unilateral RDMA write request comprises a first copy and a second copy of data, a first address of the first copy in the memory, and a second address of the second copy in the memory; and
the network interface card is configured to write the first copy into the first address and the second copy into the second address.

26. The data processing apparatus according to claim 25, wherein the memory is a persistent memory.

27. The data processing apparatus according to claim 25 or 26, wherein the data processing apparatus further comprises a processor, and the processor communicates with the network interface card;

the network interface card or the processor is configured to read the first copy from the first address and the second copy from the second address; and
the network interface card or the processor is further configured to: check the read first copy and the read second copy, and when detecting that the read first copy is inconsistent with the read second copy, perform a data overwrite operation on data stored in the first address or data stored in the second address.

28. The data processing apparatus according to claim 27, wherein the read first copy comprises first data and first check code, the read second copy comprises second data and second check code, and the read first copy is inconsistent with the read second copy when at least one of the following conditions is met:

the first data is different from the second data; or
the first check code is different from the second check code.

29. The data processing apparatus according to claim 28, wherein performing the data overwrite operation on the data stored in the first address or the data stored in the second address comprises:
when checking of the first data based on the first check code succeeds, overwriting, by using the first data and the first check code, the data stored in the second address.

30. The data processing apparatus according to claim 28, wherein performing the data overwrite operation on the data stored in the first address or the data stored in the second address comprises:
when checking of the first data based on the first check code fails and checking of the second data based on the second check code succeeds, overwriting, by using the second data and the second check code, the data stored in the first address.

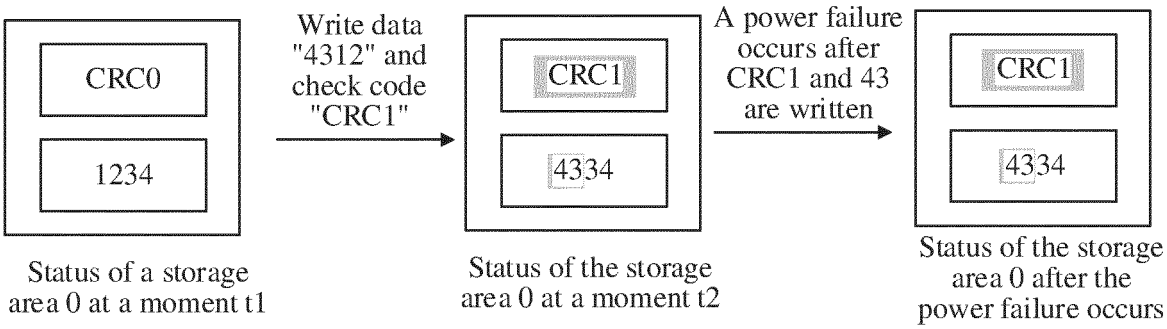

FIG. 1

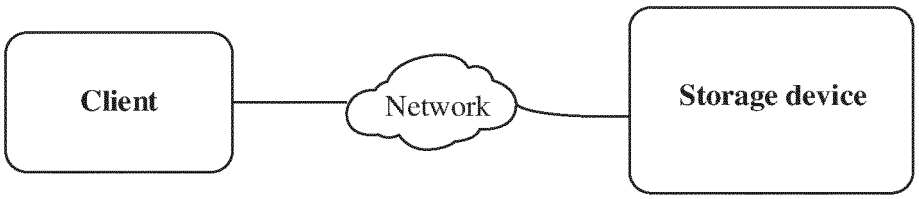

FIG. 2

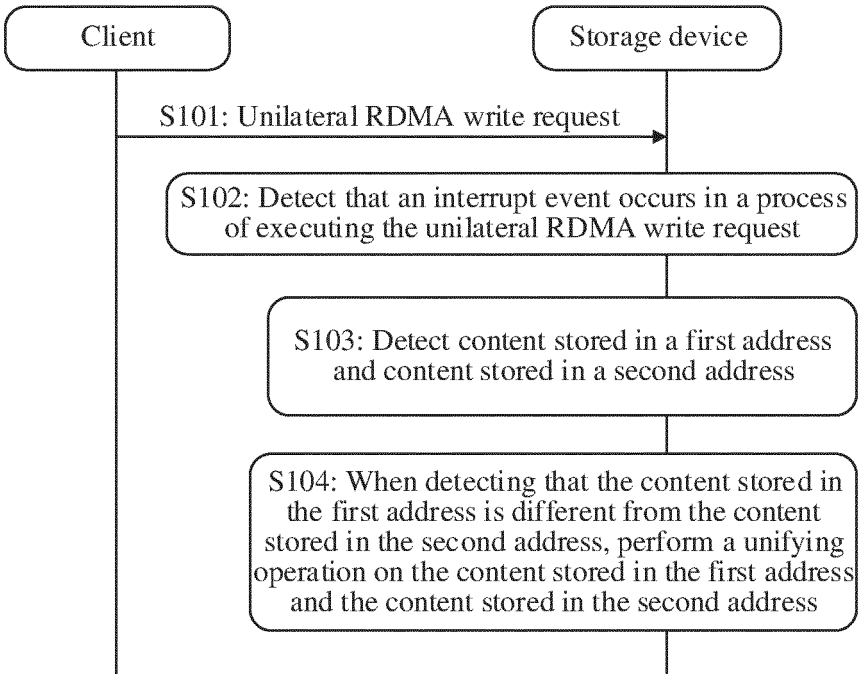

FIG. 3

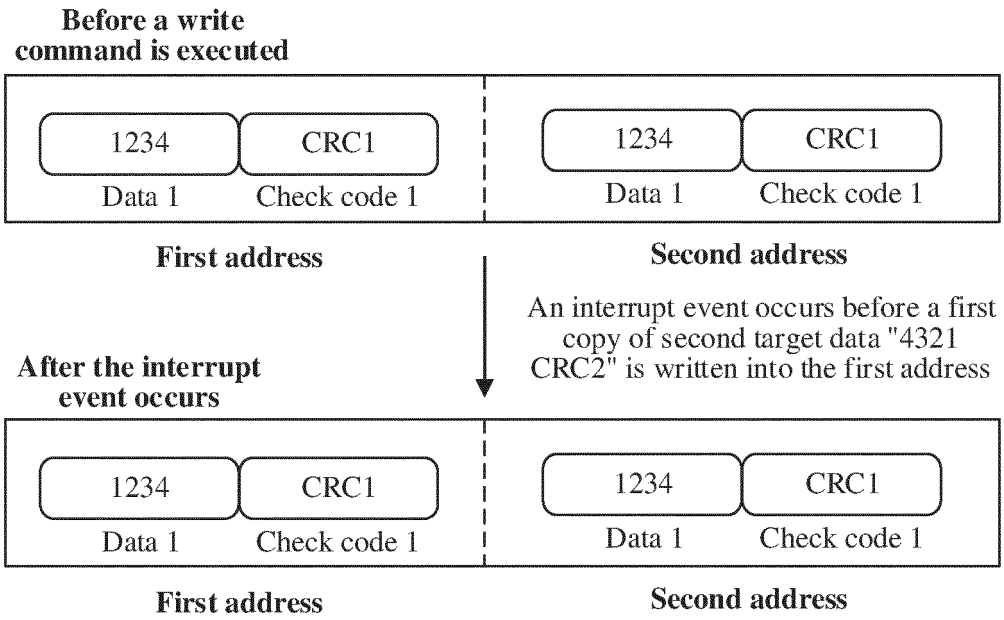
Before a write command is executed
1234
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address
An interrupt event occurs before a first copy of second target data "4321 CRC2" is written into the first address
After the interrupt event occurs
1234
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address

FIG. 4

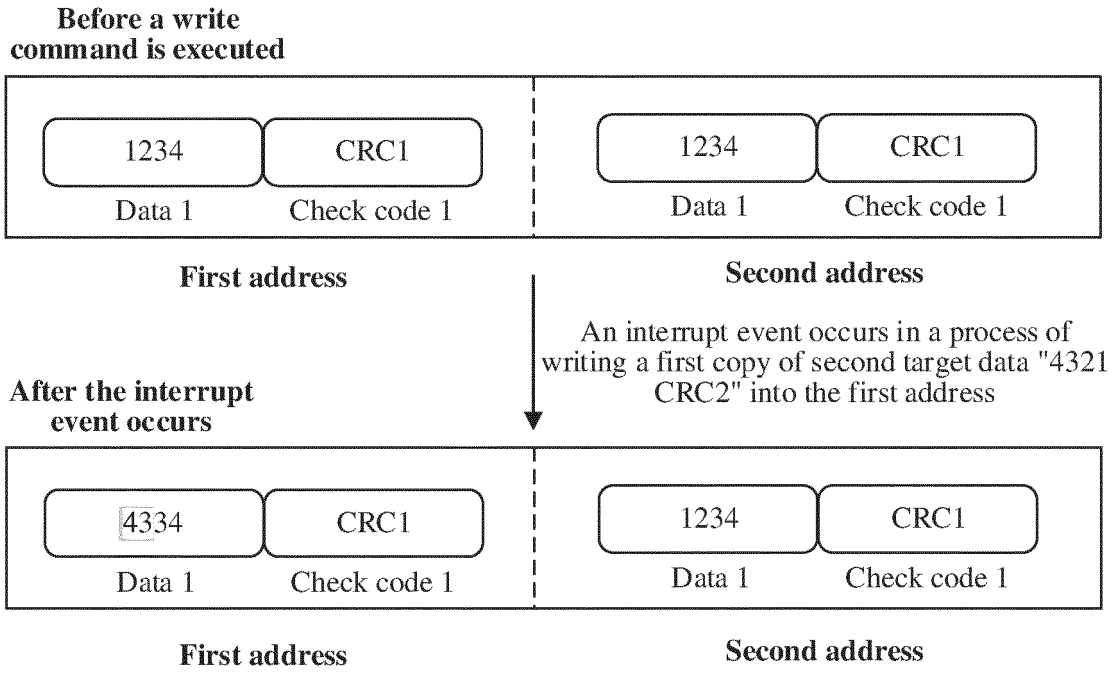
Before a write command is executed
1234
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address
An interrupt event occurs in a process of writing a first copy of second target data "4321 CRC2" into the first address
After the interrupt event occurs
4334
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address

FIG. 5

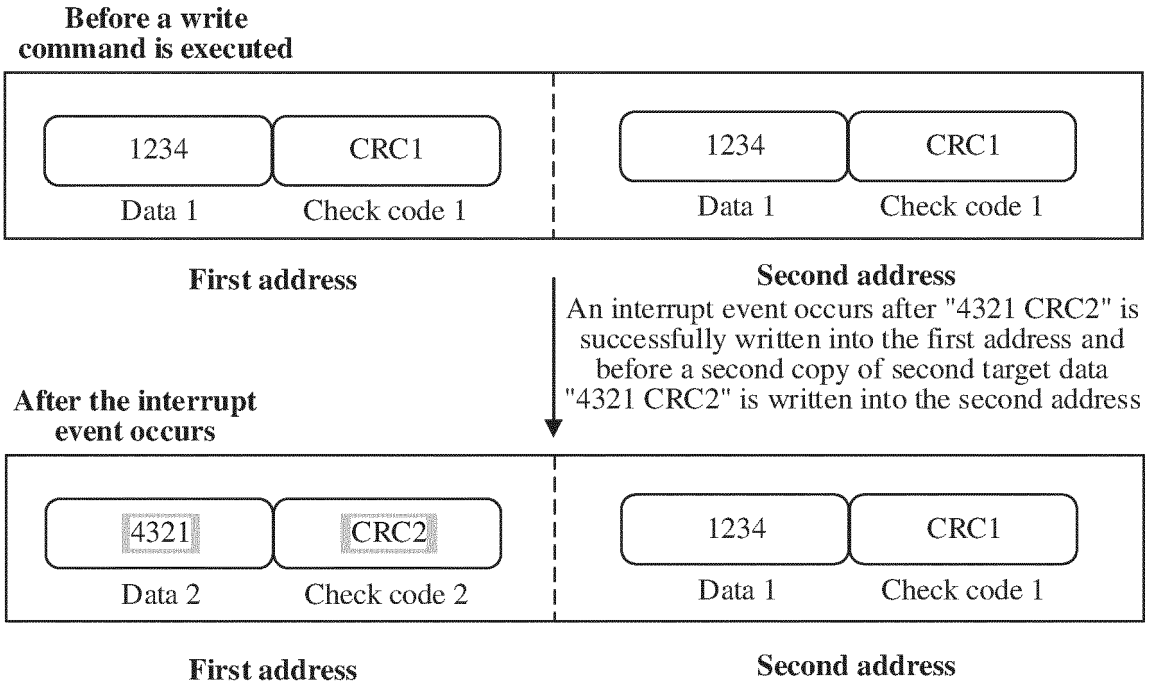
Before a write command is executed
1234
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address
An interrupt event occurs after "4321 CRC2" is successfully written into the first address and before a second copy of second target data "4321 CRC2" is written into the second address
After the interrupt event occurs
4321
CRC2
Data 2
Check code 2
1234
CRC1
Data 1
Check code 1
First address
Second address

FIG. 6

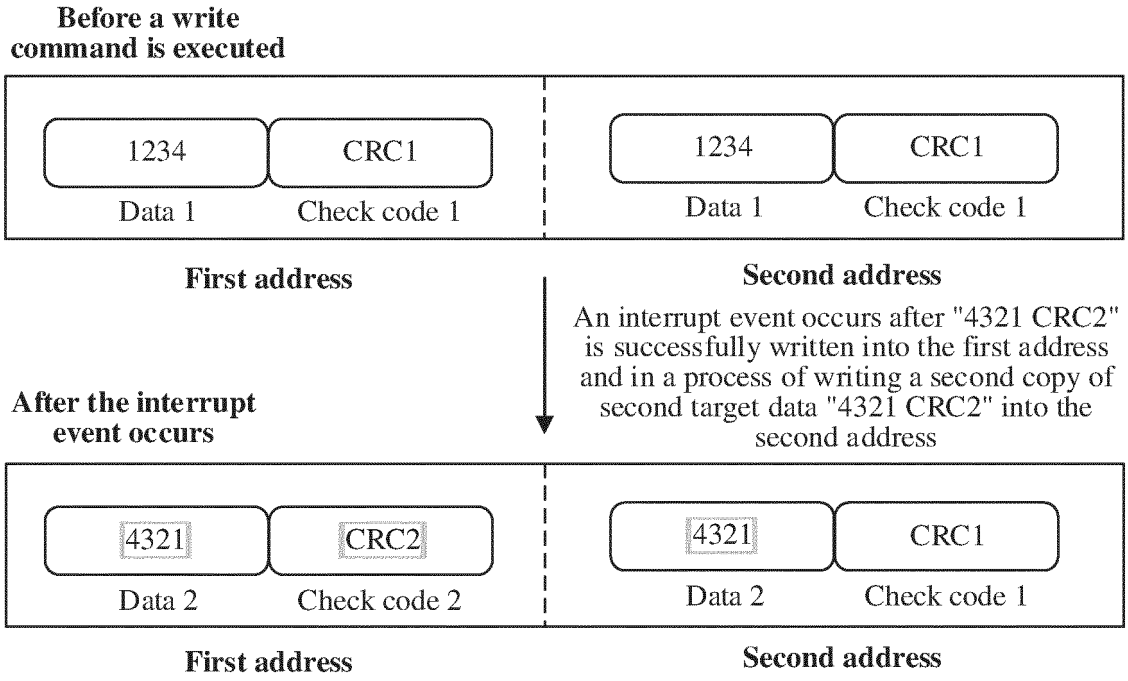
Before a write command is executed
1234
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address
An interrupt event occurs after "4321 CRC2" is successfully written into the first address and in a process of writing a second copy of second target data "4321 CRC2" into the second address
After the interrupt event occurs
4321
CRC2
Data 2
Check code 2
4321
CRC1
Data 2
Check code 1
First address
Second address

FIG. 7

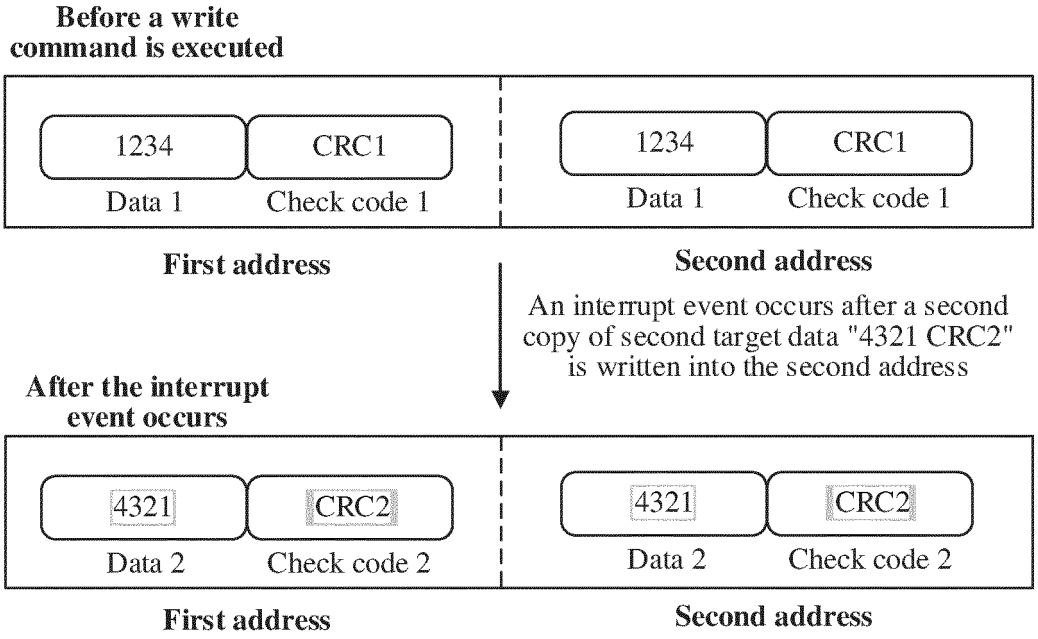
Before a write command is executed
1234
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address
An interrupt event occurs after a second copy of second target data "4321 CRC2" is written into the second address
After the interrupt event occurs
4321
CRC2
Data 2
Check code 2
4321
CRC2
Data 2
Check code 2
First address
Second address

FIG. 8

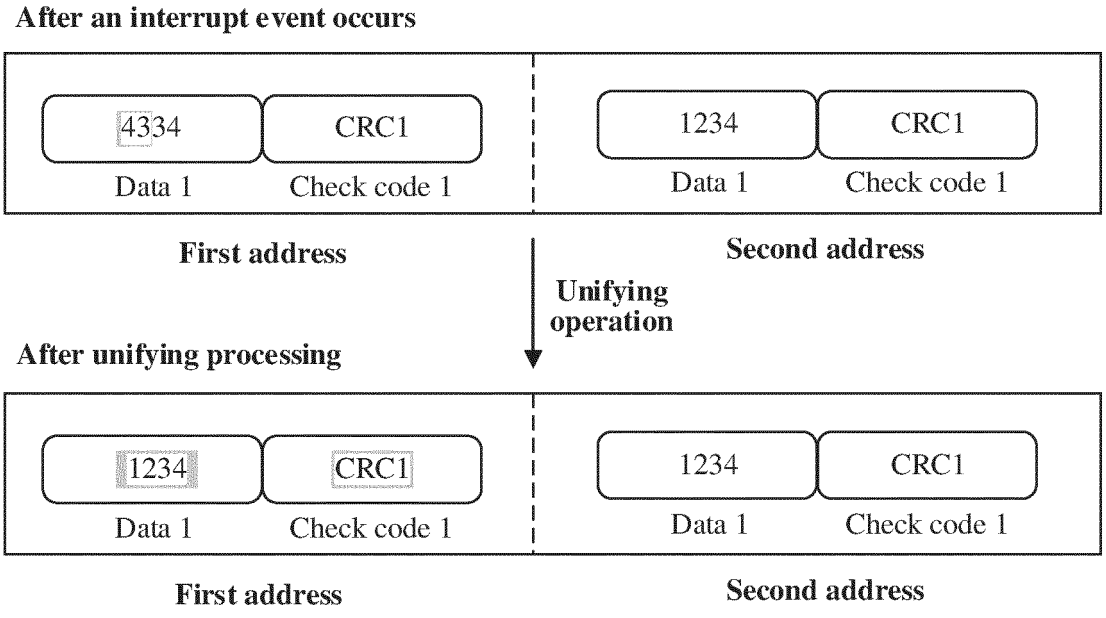
After an interrupt event occurs
4334
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address
Unifying operation
After unifying processing
1234
CRC1
Data 1
Check code 1
1234
CRC1
Data 1
Check code 1
First address
Second address

FIG. 9

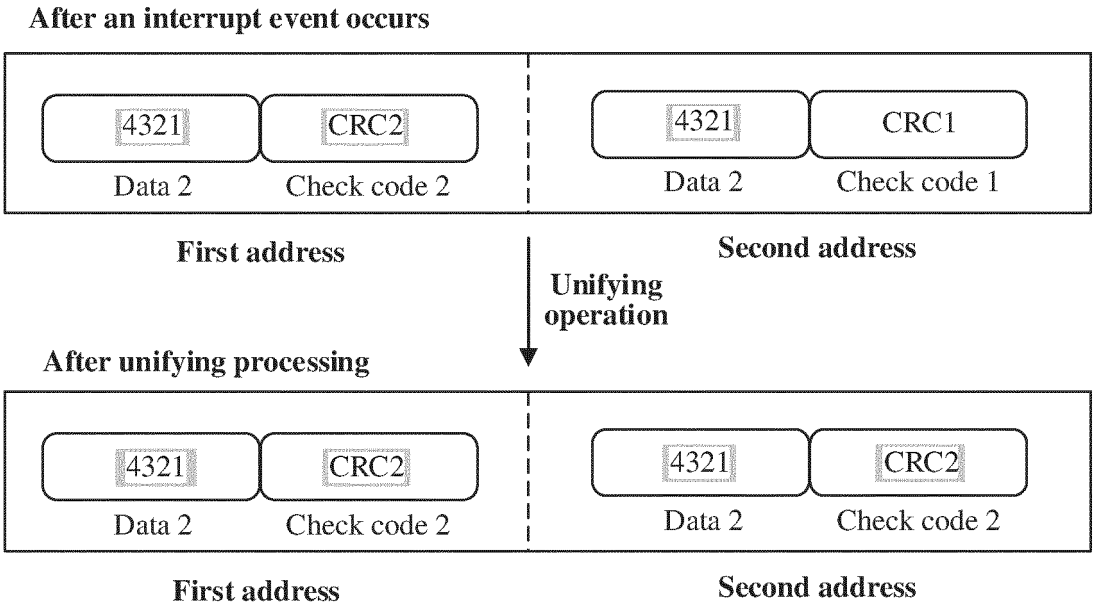
After an interrupt event occurs
4321
CRC2
Data 2
Check code 2
4321
CRC1
Data 2
Check code 1
First address
Second address
Unifying operation
After unifying processing
4321
CRC2
Data 2
Check code 2
4321
CRC2
Data 2
Check code 2
First address
Second address

FIG. 10

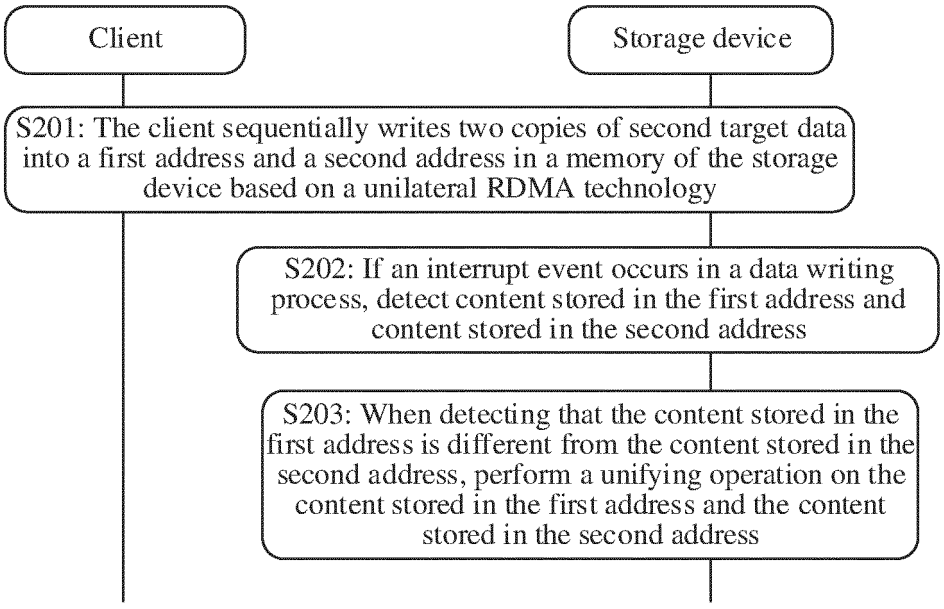
Client
Storage device
S201: The client sequentially writes two copies of second target data into a first address and a second address in a memory of the storage device based on a unilateral RDMA technology
S202: If an interrupt event occurs in a data writing process, detect content stored in the first address and content stored in the second address
S203: When detecting that the content stored in the first address is different from the content stored in the second address, perform a unifying operation on the content stored in the first address and the content stored in the second address

FIG. 11

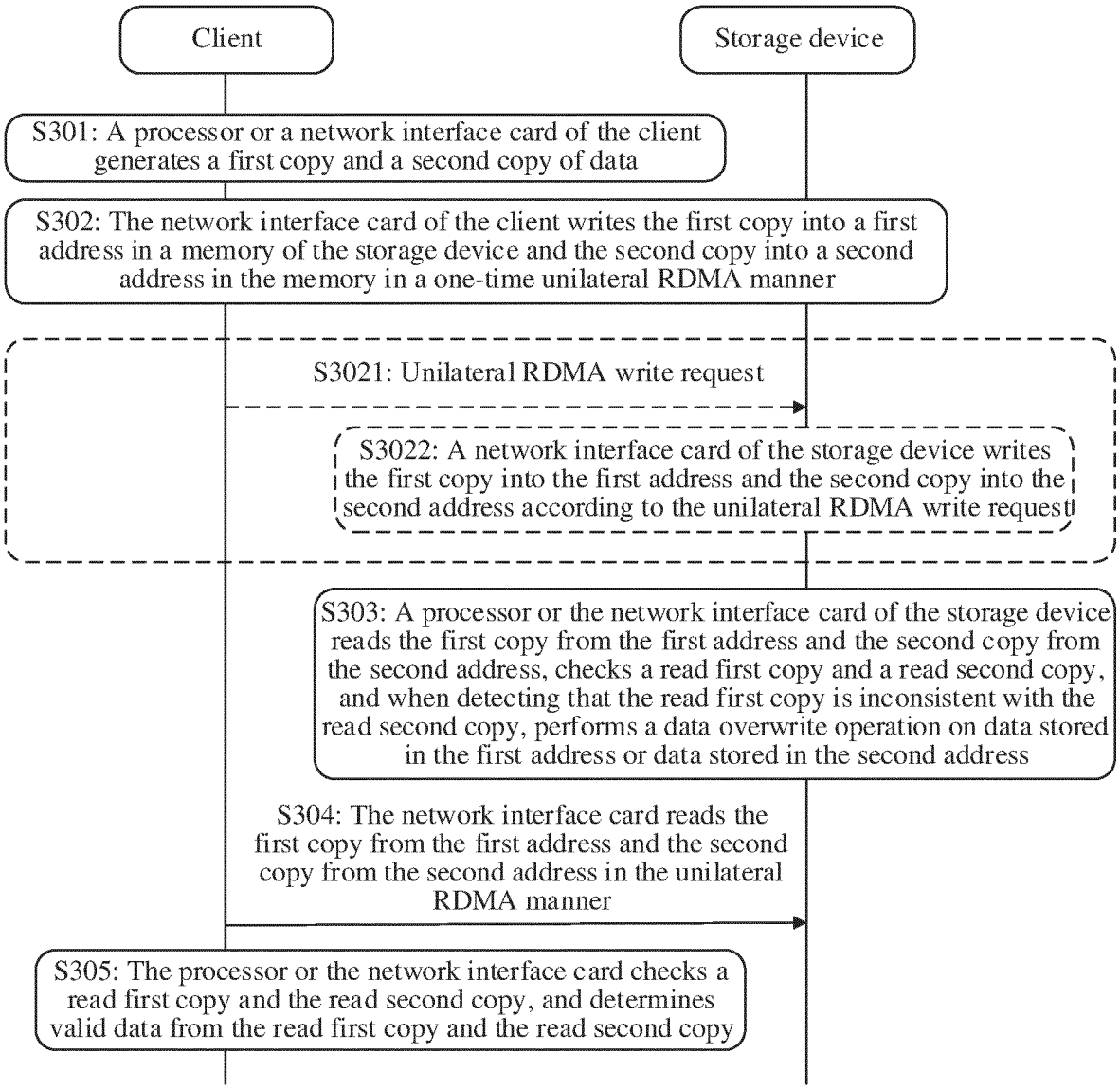
Client
Storage device
S301: A processor or a network interface card of the client generates a first copy and a second copy of data
S302: The network interface card of the client writes the first copy into a first address in a memory of the storage device and the second copy into a second address in the memory in a one-time unilateral RDMA manner
S3021: Unilateral RDMA write request
S3022: A network interface card of the storage device writes the first copy into the first address and the second copy into the second address according to the unilateral RDMA write request
S303: A processor or the network interface card of the storage device reads the first copy from the first address and the second copy from the second address, checks a read first copy and a read second copy, and when detecting that the read first copy is inconsistent with the read second copy, performs a data overwrite operation on data stored in the first address or data stored in the second address
S304: The network interface card reads the first copy from the first address and the second copy from the second address in the unilateral RDMA manner
S305: The processor or the network interface card checks a read first copy and the read second copy, and determines valid data from the read first copy and the read second copy

FIG. 12

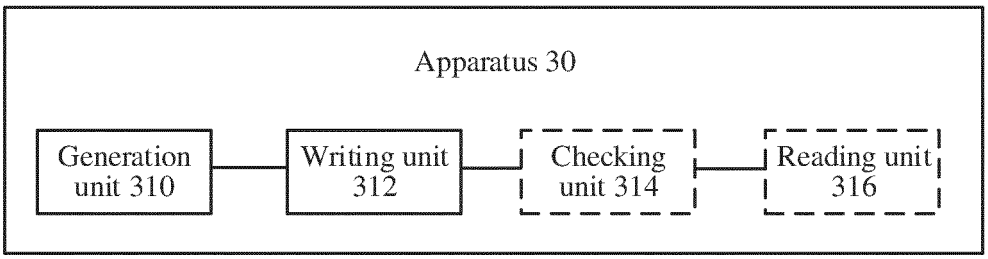
Apparatus 30
Generation unit 310
Writing unit 312
Checking unit 314
Reading unit 316

FIG. 13

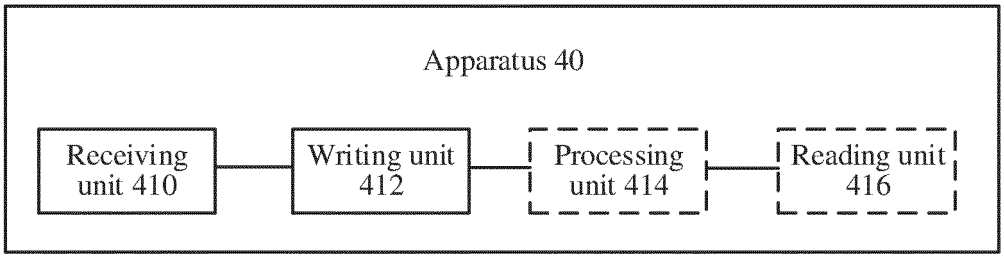
Apparatus 40
Receiving unit 410
Writing unit 412
Processing unit 414
Reading unit 416

FIG. 14

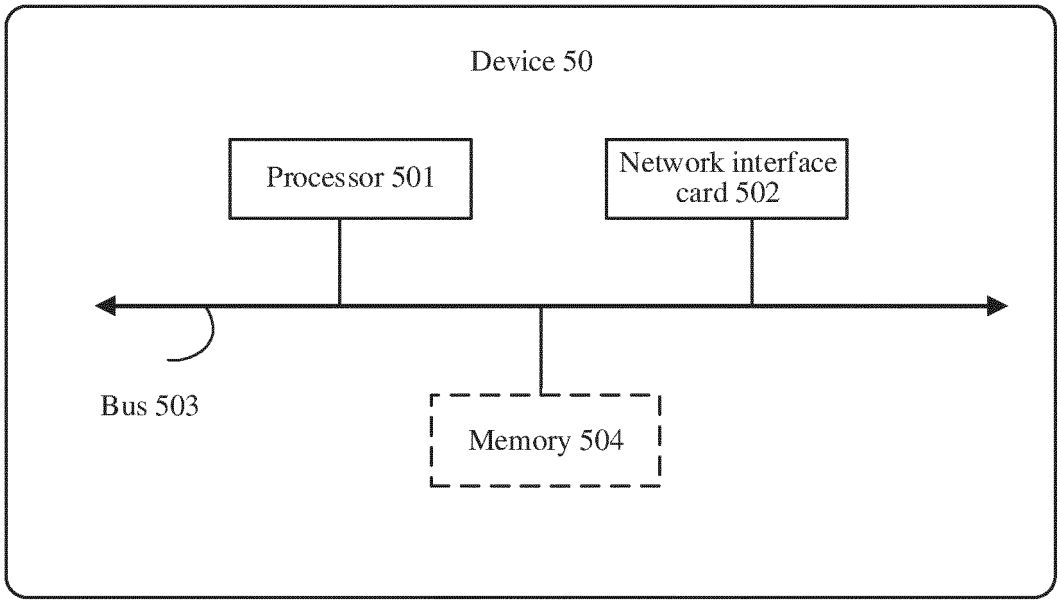
Device 50
Processor 501
Network interface card 502
Bus 503
Memory 504

FIG. 15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2022/113098

A. CLASSIFICATION OF SUBJECT MATTER

G06F 3/06(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F,H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI; CNPAT; WPI; EPODOC; IEEE: RDMA, 远程, 内存访问, DMA, 写入, 存储, 拷贝二, 拷贝2, 副本二, 第二, 副本, 拷贝, 副本2, 第二拷贝, 地址, 指针, RDMA, 请求, 指令, 多, 两个, remote, write, copy, duplicate, address, pointer, request, instruction, second, command, multipl+, several, two, dual

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 114817121 A (EMC IP HOLDING COMPANY LLC) 29 July 2022 (2022-07-29) claims 1-24, description, paragraphs 2-3 and 25-84, and figures 1-8	1-30
A	CN 112788079 A (HUAWEI TECHNOLOGIES CO., LTD.) 11 May 2021 (2021-05-11) claims 1-10, description, paragraphs 2-3, 30-38, and 51-91, and figures 1-4	1-30
A	US 2018341429 A1 (WESTERN DIGITAL TECHNOLOGIES, INC.) 29 November 2018 (2018-11-29) entire document	1-30
A	CN 112955876 A (ARM LTD.) 11 June 2021 (2021-06-11) entire document	1-30
A	US 2019155524 A1 (ULTRATA, LLC) 23 May 2019 (2019-05-23) entire document	1-30
A	WO 2021103036 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 03 June 2021 (2021-06-03) entire document	1-30
A	CN 104571956 A (CHENGDU ZHIYUN TECHNOLOGY CO., LTD.) 29 April 2015 (2015-04-29) entire document	1-30

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 October 2022	**17 November 2022**
Name and mailing address of the ISA/CN **China National Intellectual Property Administration (ISA/CN)** **No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088, China** Facsimile No. **(86-10)62019451**	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2022/113098

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2017131934 A1 (NEC CORP.) 11 May 2017 (2017-05-11) entire document	1-30
A	US 5742843 A (FUJITSU LTD.) 21 April 1998 (1998-04-21) entire document	1-30
A	US 9547598 B1 (EMULEX CORP.) 17 January 2017 (2017-01-17) entire document	1-30

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2022/113098

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 114817121 A	29 July 2022	None	
CN 112788079 A	11 May 2021	None	
US 2018341429 A1	29 November 2018	WO 2018217272 A1	29 November 2018
CN 112955876 A	11 June 2021	US 2020125518 A1	23 April 2020
		WO 2020083720 A1	30 April 2020
		EP 3644190 A1	29 April 2020
		KR 20210076103 A	23 June 2021
US 2019155524 A1	23 May 2019	WO 2017100292 A1	15 June 2017
		WO 2017100288 A1	15 June 2017
		CA 3006776 A1	15 June 2017
		CN 108885607 A	23 November 2018
		US 2019171373 A1	06 June 2019
		EP 3387548 A1	17 October 2018
		US 2020379656 A1	03 December 2020
WO 2021103036 A1	03 June 2021	CN 114730314 A	08 July 2022
CN 104571956 A	29 April 2015	None	
US 2017131934 A1	11 May 2017	JP 2017521762 A	03 August 2017
		WO 2015198591 A1	30 December 2015
		EP 3161609 A1	03 May 2017
		CN 106662981 A	10 May 2017
		CA 2953657 A1	30 December 2015
US 5742843 A	21 April 1998	JP H0830566 A	02 February 1996
US 9547598 B1	17 January 2017	None	

Form PCT/ISA/210 (patent family annex) (January 2015)

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- CN 202111022655 **[0001]**