METHODS, SYSTEMS, AND STORAGE MEDIA FOR DATA STORAGE

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims priority of Chinese Patent Application No. 202410233433.5 filed on February 29, 2024, the content of which is entirely incorporated herein by reference.

TECHNICAL FIELD

[0002] The present disclosure relates to a field of data storage, and in particular, to methods, systems and storage media for data storage.

BACKGROUND

[0003] In a field of data storage, data is written to storage hardware (e.g., a hard disk) via data nodes. Often, after data is divided into blocks, and a plurality of data blocks are written to corresponding storage hardware according to the data nodes. In this process, usually, the plurality of data blocks are centrally written to the corresponding storage hardware under one or more data nodes, and when a data node with a great count of written data blocks fails (e.g., a storage hardware damage, a node breakdown, a communication failure, etc.), the written data blocks are unreadable, which greatly reduces a data storage reliability. Accordingly, it is desired to provide methods, systems, and storage media for data storage to improve the data storage reliability.

SUMMARY

[0004] One of the embodiments of the present disclosure provides a method performed by a computing device with at least one processor and at least one storage device. The method includes generating a plurality of data blocks based on data to be stored and determining a target storage layer for the plurality of data blocks in a storage pool of the at least one storage device. The storage pool has a multi-layer architecture including a plurality of storage layers. Each of the plurality of storage layers in the storage pool corresponds to a plurality of storage units, and each storage unit in each storage layer contains a plurality of storage units in a next storage layer of the storage layer. The target storage layer is a storage layer in the storage pool that satisfies a first preset condition. The first preset condition is related to a count of storage units in the storage pool and a count of the data blocks. The method further includes determining a plurality of target storage units from the storage pool based on the target storage layer and storing the plurality of data blocks based on the plurality of target storage units.

[0005] In some embodiments, in the multi-layer architecture of the storage pool, the plurality of storage layers from top to bottom include a first layer, a second layer, a third layer, and a fourth layer.

[0006] In some embodiments, the storage pool may be one of a plurality of candidate storage pools of a resource pool, and the resource pool is configured to store data of a same type.

[0007] In some embodiments, the determining a target storage layer for the plurality of data blocks in a storage pool may include: determining whether there is at least one candidate storage pool that satisfies a second preset condition in the plurality of candidate storage pools; in response to that there is the at least one candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools, determining the storage pool from the at least one candidate storage pool that satisfies the second preset condition; and taking the first layer of the storage pool as the target storage layer.

[0008] In some embodiments, the storage pool is a candidate storage pool with the lowest load among the at least one candidate storage pool that satisfies the second preset condition.

[0009] In some embodiments, the determining a target storage layer for the plurality of data blocks in a storage

pool further includes: in response to that there is no candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools; determining at least one available storage pool from the plurality of candidate storage pools; determining the target storage layer from the second layer and layers below the second layer of the storage pool.

[0010] In some embodiments, the storage pool is an available storage pool with the lowest load among the at least one available storage pool.

[0011] In some embodiments, the second preset condition is related to a count of storage units in the first layer of the storage pool and the count of the data blocks.

[0012] In some embodiments, the determining the target storage layer from the second layer and layers below the second layer of the storage pool may include: taking the second layer as a current layer; determining whether the current layer satisfies a third preset condition; in response to that the current layer satisfies the third preset condition, designating the current layer as the target storage layer; and in response to that the current layer does not satisfy the third preset condition, determining a next layer of the current layer as a new current layer, and re-determining whether the new current layer satisfies the third preset condition until the target storage layer is determined.

[0013] In some embodiments, the third preset condition is related to a count of storage units in the current layer and the count of the data blocks.

[0014] In some embodiments, the determining a plurality of target storage units from the storage pool based on the target storage layer may include: selecting a plurality of storage units from the storage pool as the plurality of target storage units based on the target storage layer using a first selection strategy. The first selection strategy includes at least one of a first sub-strategy or a second sub-strategy. The first sub-strategy includes: in a situation where an upper layer of the target storage layer does not satisfy a fourth preset condition, the plurality of target storage units are distributed in all the storage units corresponding to the upper layer of the target storage layer; and in a situation where the upper layer of the target storage layer satisfies the fourth preset condition, the plurality of target storage units are distributed in a first count of storage units corresponding to the upper layer of the target storage layer, the first count being equal to the count of the data blocks. The second sub-strategy includes: a difference of a count of target storage units between each two of a plurality of distributed storage units is less than a preset count, and the plurality of distributed storage units are storage units of the upper layer of the target storage layer and contain the plurality of target storage units.

[0015] In some embodiments, the fourth preset condition may be related to a count of storage units in the upper layer of the target storage layer and the count of the data blocks.

[0016] In some embodiments, the storing the plurality of data blocks based on the plurality of target storage units includes: in response to that the target storage layer is not a last layer in the multi-layer architecture of the storage pool, for each of the plurality of target storage units, taking the target storage unit as a write storage unit; determining whether a layer corresponding to the write storage unit is the last layer; in response to that the layer corresponding to the write storage unit is not the last layer, selecting, based on a second selection strategy, a new write storage unit from storage units in a next layer contained in the write storage unit; determining whether a layer corresponding to the new write storage unit is the last layer until the layer corresponding to the new write storage unit is determined to be the last layer; and respectively storing the plurality of data blocks in write storage units corresponding to the plurality of target storage units.

[0017] In some embodiments, the selecting, based on a second selection strategy, a new write storage unit from storage units in a next layer contained in the write storage unit includes: determining a plurality of candidate queues from the storage units in the next layer contained in the write storage unit, the plurality of candidate queues being consist of a plurality of updatable storage units; determining a target candidate queue from the plurality of candidate queues; and determining the new write storage unit from a plurality of storage units of the target candidate queue.

[0018] In some embodiments, the determining a target candidate queue from the plurality of candidate queues includes: determining, based on load information of each of the plurality of candidate queues, a queue priority of at least one of the plurality of candidate queues, the load information at least including a count of free storage units in the corresponding candidate queue; and determining the target candidate queue based on the queue priority of the at least one candidate queue.

[0019] One of the embodiments of the present disclosure provides a system including at least one processor and at least one storage device storing computer instructions. When the computer instructions are executed by the at least one processor, the at least one processor is configured to perform the method in any one embodiment of the present disclosure.

[0020] One of the embodiments of the present disclosure provides a system including a data block generating module, a storage layer determining module, a storage unit determining module, and a data block storage module. The data block generation module is configured to generate a plurality of data blocks based on the data to be stored. The storage layer determination module is configured to determine a target storage layer for the plurality of data blocks in a storage pool of at least one storage device. The storage pool has a multi-layer architecture including a plurality of storage layers. Each of the plurality of storage layers in the storage pool corresponds to a plurality of storage units, and each storage unit in each storage layer contains a plurality of storage units in the next storage layer of the storage layer. The target storage layer is a storage layer in the storage pool that satisfies a first preset condition. The first preset condition is related to a count of storage units of the storage pool and a count of the data blocks. The storage unit determining module is configured to determine a plurality of target storage units from the storage pool based on the target storage layer. The data block storage module is configured to store the plurality of data in blocks based on the plurality of target storage units.

[0021] One of the embodiments of the present disclosure provides a computer-readable storage medium, the storage medium storing computer instructions, and when the computer reads the computer instructions in the storage medium, the computer executes the method in any one embodiment of the present disclosure.

[0022] Additional features are set forth in part in the description which follows, and the other part may become apparent to those skilled in the art upon examination of the following and the accompanying drawings or may be learned by production or operation of the examples. The features of the present disclosure may be realized and attained by practice or use of various aspects of the methodologies, instrumentalities, and combinations set forth in the detailed examples discussed below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] The present disclosure is further described in terms of exemplary embodiments. These exemplary embodiments are described in detail with reference to the drawings. The drawings are not to scale. These embodiments are non-limiting exemplary embodiments, in which like reference numerals represent similar

structures throughout the several views of the drawings, and wherein:

[0024] FIG. 1 is a schematic diagram illustrating an exemplary data storage system according to some embodiments of the present disclosure;

[0025] FIG. 2 is a block diagram illustrating an exemplary data storage system according to some embodiments of the present disclosure;

[0026] FIG. 3 is a flowchart illustrating an exemplary process for data storage according to some embodiments of the present disclosure;

[0027] FIG. 4 is a schematic diagram illustrating an exemplary storage device according to some embodiments of the present disclosure;

[0028] FIG. 5 is a flowchart illustrating an exemplary process for determining a target storage layer according to some embodiments of the present disclosure;

[0029] FIG. 6 is a flowchart illustrating an exemplary process for determining a target storage layer according to some embodiments of the present disclosure;

[0030] FIG. 7 is a schematic diagram illustrating an exemplary first selection strategy according to some embodiments of the present disclosure;

[0031] FIG. 8 is a flowchart illustrating an exemplary process for storing a plurality of data blocks according to some embodiments of the present disclosure;

[0032] FIG. 9 is a flowchart illustrating an exemplary process for selecting a new write storage unit according to some embodiments of the present disclosure; and

[0033] FIG. 10 is a schematic diagram illustrating an exemplary process for data storage according to some embodiments of the present disclosure.

DETAILED DESCRIPTION

[0034] In the following detailed description, numerous specific details are set forth by way of examples in order to provide a thorough understanding of the relevant disclosure. However, it should be apparent to those skilled in the art that the present disclosure may be practiced without such details. In other instances, well-known methods, procedures, systems, components, and/or circuitry have been described at a relatively high level, without detail, in order to avoid unnecessarily obscuring aspects of the present disclosure. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present disclosure. Thus, the present disclosure is not limited to the embodiments shown, but is to be accorded the widest scope consistent with the claims.

[0035] The terminology used herein is for the purpose of describing particular example embodiments only and is not intended to be limiting. As used herein, the singular forms “a,” “an,” and “the,” may be intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “include,” “includes,” and/or “comprising,” “include,” “includes,” and/or “including,” when used in the present disclosure, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0036] It will be understood that the terms “system,” “engine,” “unit,” “module,” and/or “block” used herein

are one method to distinguish different components, elements, parts, sections, or assemblies of different levels in ascending order. However, the terms may be displaced by other expressions if they achieve the same purpose.

[0037] It may be understood that when a unit, engine, module, or block is referred to as being "on," "connected to," or "coupled to," another unit, engine, module, or block, it may be directly on, connected or coupled to, or communicate with the other unit, engine, module, or block, or an intervening unit, engine, module, or block may be present unless the context clearly indicates otherwise. As used herein, the term "and/or" includes any or all combinations of one or more of the associated listed items.

[0038] These and other features, and characteristics of the present disclosure, as well as the methods of operation and functions of the related elements of structure and the combination of parts and economies of manufacture, may become more apparent upon consideration of the following description with reference to the accompanying drawings, all of which form a part of this disclosure. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only and are not intended to limit the scope of the present disclosure. It is understood that the drawings are not to scale.

[0039] In a distributed storage system, to improve a data storage reliability, a multicopy technique or an erasure coding (EC) technique is often used to store data. The multicopy technique has a low space utilization due to a need to store a plurality of copies on a data node. The erasure coding technique is widely used because of its high space utilization. The erasure coding technique means dividing user's data into groups to obtain N count of data blocks and M count of verification blocks and store them in storage hardware of the data node, thus enhancing a reliability and an availability of the data. However, in an actual situation, both the multicopy technique and the erasure coding technique are likely to make the data unavailable due to problems such as a hard disk corruption, a node breakdown, and a network inaccessible, etc. For example, for a system that employs the erasure coding technique, if there are more than M count of data blocks in the system that are stored on the same data node, the data in the system will be unavailable after the data node breaks down. As another example, for a system that employs the multicopy technique, if an original data block and a copy of that data block exist on the same data node, the data in the system is unusable after the data node breaks down. Currently, to solve the above problem, a combination of erasure coding technique and multicopy technique may be used. First, the erasure coding technique is used to obtain the data blocks and the verification blocks, and then the data blocks are stored in a multicopy form on the data node, however, this approach still has the problem of low space utilization.

[0040] Embodiments of the present disclosure provide a method and a system for data storage. The method includes generating a plurality of data blocks based on data to be stored and determining a target storage layer for the plurality of data blocks in a storage pool of at least one storage device. The storage pool has a multi-layer architecture including a plurality of storage layers. Each of the plurality of storage layers in the storage pool corresponds to a plurality of storage units, and each storage unit in each storage layer contains a plurality of storage units in a next storage layer of the storage layer. The target storage layer is a storage layer in the storage pool that satisfies a first preset condition. The first preset condition is related to a count of storage units in the storage pool and a count of the data blocks. The method further includes determining a plurality of target storage units from the storage pool based on the target storage layer and storing the plurality of data blocks based on the plurality of target storage units.

[0041] According to the embodiments of the present disclosure, it is possible to avoid storing the plurality of

data blocks centrally in some storage units. Furthermore, as a higher storage layer corresponds to a greater storage space of the storage unit, and less likely for the plurality of the storage units in the higher storage layer to fail, therefore the higher storage layer has a higher failure tolerance. By using the plurality of storage units to store the plurality of data blocks in a distributed manner, in an event of failure, a possibility of the stored data being unrecoverable may be minimized, thus improving the failure tolerance of the data storage, and with higher space utilization, the data storage reliability and a data availability are improved.

[0042] FIG. 1 is a schematic diagram illustrating an exemplary data storage system 100 according to some embodiments of the present disclosure. As shown in FIG. 1, the data storage system 100 may include a storage device 110, a processing device 120, a terminal 130, and a network 140.

[0043] The storage device 110 may store data and/or instructions. The data and/or instructions may be obtained from, for example, the processing device 120, and/or any other component of the data storage system 100. In some embodiments, the storage device 110 may store the data and/or instructions that the processing device 120 may execute or use to perform the exemplary method described in the present disclosure. In some embodiments, the storage device 110 may include a hard disk device, a mass storage device, a removable storage device, a volatile read-and-write memory, a read-only memory (ROM), etc., or any combination thereof. In some embodiments, the storage device 110 may be implemented on a cloud platform. In some embodiments, the storage device 110 may be a portion of the processing device 120. In some embodiments, the storage device 110 may include a plurality of storage hardware devices (e.g., hard disks, etc.). These storage hardware devices may form a multi-layer architecture. The multi-layer architecture may include, from top to bottom, a plurality of resource pools, a plurality of storage pools, and a plurality of storage layers (e.g., a first layer, a second layer, a third layer, and a fourth layer). Each resource pool is configured to store the same type of data. Multiple storage pools are located one layer below each resource pool. Multiple storage layers are located below each storage pool. Each storage layer corresponds to a plurality of storage units, and each storage unit in one storage layer includes a plurality of storage units in a next storage layer of the storage layer. More descriptions of the multi-layer architecture may be found elsewhere in the present disclosure (e.g., FIG. 2 and the descriptions thereof).

[0044] The processing device 120 may process information and/or data related to the data storage system 100 to perform one or more functions described in the present disclosure. For example, the processing device 120 may generate a plurality of data blocks based on data to be stored, and determine a target storage layer for the plurality of data blocks in the storage pool of the storage device 110. The target storage layer refers to a storage layer in the storage pool that satisfies a first preset condition. The first preset condition is related to a count of the storage units in the storage pool and a count of the data blocks. Further, the processing device 120 may determine a plurality of target storage units from the storage pool based on the target storage layer, and store the plurality of data blocks based on the plurality of target storage units.

[0045] The processing device 120 may be a single server or a server group. The server group may be centralized, or distributed (e.g., the processing device 120 may be a distributed system). In some embodiments, the processing device 120 may be local or remote. For example, the processing device 120 may access the information and/or data stored in the storage device 110 via the network 140. As another example, the processing device 120 may be directly connected to the storage device 110 to access the stored information and/or data. In some embodiments, the processing device 120 may be implemented on a cloud platform.

Merely by way of example, the cloud platform may include a private cloud, a public cloud, a hybrid cloud, a community cloud, a distributed cloud, an inter-cloud, a multi-cloud, or the like, or any combination thereof.

[0046] In some embodiments, the processing device 120 may include one or more processors (e.g., single-core processor(s) or multi-core processor(s)). Merely by way of example, the processing device 120 may include a central processing unit (CPU), an application-specific integrated circuit (ASIC), an application-specific instruction-set processor (ASIP), a graphics processing unit (GPU), a physics processing unit (PPU), a digital signal processor (DSP), a field-programmable gate array (FPGA), a programmable logic device (PLD), a controller, a microcontroller unit, a reduced instruction-set computer (RISC), a microprocessor, or the like, or any combination thereof.

[0047] The terminal 130 may interact with a user. The user may give an operation instruction to the processing device 120 through the terminal 130 to make the processing device 120 complete a specified operation, e.g., to read or write data from the storage device 110, etc. In some embodiments, the terminal 130 may make the processing device 120 perform the exemplary methods described in the present disclosure. In some embodiments, the terminal 130 may be a mobile device 130-1, a tablet computer 130-2, a laptop computer 130-3, a desktop computer, and other devices with input and/or output capabilities or any combination thereof.

[0048] The network 140 may include any suitable network that facilitates an exchange of the information and/or data for the data storage system 100. In some embodiments, one or more components (e.g., the processing device 120) of the data storage system 100 may communicate the information and/or data with one or more other components of the data storage system 100 via the network 140. For example, the processing device 120 may write data to the storage device 110 or read data from the storage device 110 via the network 150. In some embodiments, the network 150 may be or may include a wired network, a wireless network (e.g., an 802.11 network, a Wi-Fi network), etc.

[0049] It should be noted that the above description is merely provided for the purposes of illustration, and is not intended to limit the scope of the present disclosure. For those skilled in the art, multiple variations and modifications may be made under the teachings of the present disclosure. In some embodiments, the data storage system 100 may include one or more additional components, and/or one or more components of the data storage system 100 described above may be omitted. Additionally or alternatively, two or more components of the data storage system 100 may be integrated into a single component. A component of the image processing system 100 may be implemented on two or more sub-components. However, those variations and modifications do not depart from the scope of the present disclosure.

[0050] FIG. 2 is a block diagram illustrating an exemplary data storage system 200 according to some embodiments of the present disclosure. As shown in FIG. 2, the data storage system 200 includes a data block generation module 210, a storage layer determination module 220, a storage unit determination module 230, and a data block storage module 240. In some embodiments, one or more modules of the data storage system 200 may be implemented by the processing device 120.

[0051] The data block generation module 210 may be configured to generate a plurality of data blocks based on data to be stored. More descriptions of the generation of the plurality of data blocks may be found elsewhere in the present disclosure (e.g., operation 310 and the descriptions thereof).

[0052] The storage layer determination module 220 may be configured to determine a target storage layer for the plurality of data blocks in a storage pool of at least one storage device. The storage pool has a multi-layer

architecture including a plurality of storage layers. Each of the plurality of storage layers in the storage pool corresponds to a plurality of storage units, and each storage layer contains a plurality of storage units in the next storage layer of the storage layer. The target storage layer is a storage layer in the storage pool that satisfies a first preset condition. The first preset condition is related to a count of storage units of the storage pool and a count of the data blocks. More descriptions of the determination of the target storage layer may be found elsewhere in the present disclosure (e.g., operations 320 and the descriptions thereof).

[0053] The storage unit determination module 230 may be configured to determine a plurality of target storage units from the storage pool based on the target storage layer. More descriptions of the determination of the plurality of target storage units may be found elsewhere in the present disclosure (e.g., operations 330 and the descriptions thereof).

[0054] The data block storage module 240 may be configured to store the plurality of data in blocks based on the plurality of target storage units. More descriptions of the determination of the plurality of data blocks may be found elsewhere in the present disclosure (e.g., operations 340 and the descriptions thereof).

[0055] It should be noted that the above description is merely provided for the purposes of illustration, and is not intended to limit the scope of the present disclosure. For those skilled in the art, multiple variations and modifications may be made under the teachings of the present disclosure. However, those variations and modifications do not depart from the scope of the present disclosure. In some embodiments, the processing device 120 may include one or more additional modules, such as a storage module (not shown) for storing data.

[0056] FIG. 3 is a flowchart illustrating an exemplary process 300 for data storage according to some embodiments of the present disclosure. In some embodiments, the process 300 may be implemented by the data storage system 100 and/or the data storage system 200. For example, the process 300 may be implemented as a set of instructions (e.g., an application) stored in a storage device (e.g., the storage device 110 illustrated in FIG. 1). The processing device 120 of the data storage system 100 and/or one or more modules of the data storage system 200 may perform the set of instructions and accordingly be directed to perform the process 300.

[0057] In 310, a plurality of data blocks are generated based on data to be stored. In some embodiments, operation 310 may be performed by the data block generation module 210.

[0058] The data to be stored refers to data that needs to be stored, which includes any type of data. For example, the data to be stored may include one of a text, an image, a multimedia file, a binary file, etc., or any combination thereof. The data block is a storage unit of the data to be stored.

[0059] In some embodiments, the processing device 120 may divide the data to be stored into blocks in a variety of ways. For example, the processing device 120 may divide the data to be stored according to a size of the data to be stored, a type of the data to be stored, etc. In some embodiments, the processing device 120 may divide the data to be stored into first data blocks and second data blocks (also referred to as data verification blocks). The first data blocks are original data contained in the data to be stored, and the second data blocks are verification information generated based on the original data contained in the data to be stored. Specifically, the processing device 120 may process the data to be stored by an erasure coding technique to obtain a plurality of first data blocks and second data blocks. By dividing the data to be stored into the plurality of first data blocks and second data blocks, when a portion of the data blocks of the data to be stored is lost, the data to be stored may be restored through the data blocks whose data is not lost, thereby reducing a possibility of the loss of the data to be stored, and improving an availability of the data and a reliability of the data storage.

[0060] In 320, a target storage layer for the plurality of data blocks is determined in a storage pool of the at least one storage device (e.g., the storage device 110 in FIG. 1). In some embodiments, operation 320 may be performed by the storage layer determination module 220.

[0061] The storage pool is a logical unit used to store data. The storage pool has a multi-layer architecture including a plurality of storage layers, each of the plurality of storage layers in the storage pool corresponding to a plurality of storage units, and each storage unit in each storage layer contains a plurality of storage units in a next storage layer of the storage layer. Different layers in the multi-layer architecture correspond to different storage capacities (also referred to as storage spaces), and the storage capacity of each storage unit in each storage layer is equal to a sum of the storage capacities of the multiple storage units, contained in the storage unit, in the next storage layer of the storage layer. As a result, the higher the layer, the greater the storage capacity of the storage units corresponding to the layer, which makes it less likely for the storage units of a higher layer to fail, i.e., the higher the fault tolerance.

[0062] In some embodiments, in the multi-layer architecture of the storage pool, the plurality of storage layers from top to bottom include a first layer, a second layer, a third layer, and a fourth layer. The first layer, the second layer, the third layer, and the fourth layer may be referred to as a rack layer, a node layer, a hard disk pool layer, and a hard disk layer, respectively, and the storage units corresponding to these layers may be sequentially referred to as racks, nodes, storage hardware pools, and storage hardware. A rack is a storage unit that consists of a plurality of nodes based on their logical relationship or physical position relationship. A node is a storage unit that consists of a plurality of storage hardware pools based on their logical relationship or physical position relationship. A storage hardware pool is a storage unit that consists of a plurality of pieces of storage hardware based on their logical relationship or a physical position relationship. The storage hardware may include at least one of a mechanical hard disk, a solid state drive, a magnetic tape, a removable storage (e.g., a USB flash drive), etc., or any combination thereof. Through the multi-layer architecture of the storage pool, the data may be stored in a layer that satisfies a failure tolerance requirement, thereby reducing the possibility that the plurality of storage units all fail and cause the data to be stored to be unrecoverable, which improves the data storage reliability.

[0063] In some embodiments, the storage pool may be one of a plurality of candidate storage pools of a resource pool. The candidate storage pool is a storage pool with a free storage space that is able to satisfy the requirements for the data block storing the data to be stored. For example, the candidate storage pool has a count of available storage units that is greater than or equal to a count of the data blocks of the data to be stored. The available storage units refer to storage units whose free storage space is capable of storing at least one data block. The resource pool is configured to store data of a same type. For example, a resource pool A is configured to store image data. A resource pool B is configured to store video data. By selecting the storage pool from the plurality of candidate storage pools of the resource pool, the obtained storage pools may be more capable of satisfying the requirement of the data failure tolerance, which further improves the data storage reliability. By configuring the resource pool to store the same type of data, a storage density of the data storage may be increased, and a read and write performance may be improved by maintaining a data consistency. For example, a certain degree of lossless compression may be performed for this type of data, thereby increasing the storage density of the data storage. As another example, when the data is all of the same type, a read and write optimization may be specifically performed on this type.

[0064] FIG. 4 is a schematic diagram illustrating an exemplary storage device 400 according to some embodiments of the present disclosure. As shown in FIG. 4, the storage device 400 has a multi-layer architecture. The multi-layer architecture may include, from top to bottom, six layers. The six layers are, from top to bottom, a resource pool layer, a storage pool layer, a rack layer, a node layer, a hard disk pool layer, and a hard disk layer. For example, as shown in FIG. 4, the resource pool layer includes a resource pool. The storage pool layer includes storage pool 1 and storage pool 2, the rack layer includes rack 1 to rack N1 respectively corresponding to the storage pool 1 and the storage pool 2. The node layer includes node 1 to node N1 respectively corresponding to the rack 1 to rack N1. The hard disk pool layer includes hard disk pool 1 to hard disk pool N3 respectively corresponding to the node 1 to node N2. The hard disk layer includes hard disk 1 to hard disk N4 respectively corresponding to the hard disk pool 1 to the hard disk pool N4. It is noted that the rack layer, the node layer, the hard disk pool layer, and the hard disk layer are referred to as the storage layers.

[0065] Each storage unit in each layer contains a plurality of storage units in the next layer of the layer. For example, as shown in FIG. 4, the resource pool includes the storage pool 1 and the storage pool 2, the storage pool 1 includes the rack 1 to the rack N1, the rack 1 includes the node 1 to the node N2, the node 1 includes the hard disk pool 1 to the hard disk pool N3, and the hard disk pool 1 includes the hard disk 1 to the hard disk N4. It should be noted that N1, N2, N3, and N4 are integers greater than 1.

[0066] The target storage layer is a storage layer in a storage pool that satisfies requirements for a data failure tolerance, also known as a failure tolerance layer. In some embodiments, the target storage layer may be a storage layer in the storage pool that satisfies a first preset condition. The first preset condition is related to a count of the storage units in the storage pool and a count of the data blocks. For example, the first preset condition may be that the count of the available storage units in the target storage layer is greater than or equal to the count of the data blocks. It is understood that the higher the layer of the target storage layer, the more dispersed storage units for storing the data blocks may be determined from the storage units corresponding to the layer.

[0067] In some embodiments, the processing device 120 may determine whether there is at least one candidate storage pool that satisfies a second preset condition in the plurality of candidate storage pools (e.g., the storage pool 1 or the storage pool 2 in FIG. 4). In response to that there is the at least one candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools, the processing device 120 may determine the storage pool from the at least one candidate storage pool that satisfies the second preset condition and take the first layer (e.g., the rack layer) of the storage pool as the target storage layer. In response to that there is no candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools, the processing device 120 may determine at least one available storage pool from the plurality of candidate storage pools; determine the storage pool from the at least one available storage pool; and determine the target storage layer from the second layer and layers below the second layer of the storage pool. For more contents on how to determine the target storage layer, please refer to FIG. 5 and FIG. 6.

[0068] In 330, a plurality of target storage units are determined from the storage pool based on the target storage layer. In some embodiments, operation 330 may be performed by the storage unit determination module 230.

[0069] In some embodiments, the processing device 120 may determine a plurality of storage units from the storage pool based on the target storage layer using a preset selection strategy (e.g., a first selection strategy),

and take the plurality of storage units as the target storage units for storing the data to be stored. For more contents on how to determine the target storage unit, please refer to FIG. 7.

[0070] In 340, the plurality of data blocks are stored based on the plurality of target storage units. In some embodiments, the operation 340 may be performed by the data block storage module 240.

[0071] In some embodiments, in response to that the target storage layer is not a last layer in the multi-layer architecture of the storage pool, for each of the plurality of target storage units, the processing device 120 may take the target storage unit as a write storage unit and determine whether a layer corresponding to the write storage unit is the last layer. In response to that the layer corresponding to the write storage unit is not the last layer, the processing device 120 may select, based on a second selection strategy, a new write storage unit from storage units in a next layer contained in the write storage unit; and determine whether a layer corresponding to the new write storage unit is the last layer until the layer corresponding to the new write storage unit is determined to be the last layer. Further, the processing device 120 may respectively store the plurality of data blocks in write storage units corresponding to the plurality of target storage units. For more contents on how to store the data blocks, please refer to FIG. 8 and FIG. 9.

[0072] In the embodiments of the present disclosure, a plurality of target storage units are determined in the multi-layer architecture in the storage pool, and the plurality of data blocks are stored based on the plurality of target storage units, e.g., each data block is stored, based on each target storage unit, in the storage unit on a final layer in the multi-layer architecture. Thereby, when a relatively great count of storage units storing the data blocks all fail, as one storage unit correspondingly stores only one data block, which results in an inability to read one data block only, and does not lead to the data to be stored to be unable to restore, and the data storage reliability is further improved.

[0073] FIG. 5 is a flowchart illustrating an exemplary process 500 for determining a target storage layer according to some embodiments of the present disclosure. In some embodiments, at least part of the process 500 may be performed to achieve at least part of the operation 320 as described in connection with FIG. 3. For example, the processing device 120 or the storage layer determination module 220 may determine a target storage layer for a plurality of data blocks in a storage pool by performing at least part of the process 500.

[0074] In 510, whether there is at least one candidate storage pool that satisfies a second preset condition is determined in a plurality of candidate storage pools.

[0075] The second preset condition may be related to a count of storage units in a first layer of a storage pool and a count of data blocks. For example, the second preset condition may be that the count of storage units in the first layer (e.g., a rack layer) of the storage pool is greater than or equal to the count of data blocks. The processing device 120 may obtain a count of available storage units (the storage units whose free space is able to store at least one data block) in the first layer of each candidate storage pool of the plurality of candidate storage pools, and compare the count to the count of data blocks to determine whether the candidate storage pool satisfies the second preset condition. By using the second preset condition related to the count of the storage units and the count of the data blocks in the first layer of the storage pool, the target storage layer may be obtained starting from the highest layer of the storage pool, and the higher the layer of the target storage, the more failure-tolerant the data storage.

[0076] In some embodiments, in response to that there is the at least one candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools, the processing device 120 may determine

the target storage layer by performing operations 520-530. In response to that there is no candidate storage pool that satisfies the second preset condition, the processing device 120 may determine the target storage layer by performing operations 540-560.

[0077] In 520, the storage pool is determined from the at least one candidate storage pool that satisfies the second preset condition.

[0078] In some embodiments, the processing device 120 may determine the storage pool from the candidate storage pool that satisfies the second preset condition based on at least one of a load of the candidate storage pool, a communication quality, a free storage space, etc.

[0079] The load refers to a ratio of the available storage units of the first layer in the candidate storage pool to all the storage units in the at least one candidate storage pool. Merely by way of example, if the first layer of storage pool A includes 100 storage units, in which a count of available storage units is 50, the load of the storage pool A is 50%. In some embodiments, the storage pool determined by the processing device 120 is a candidate storage pool with the lowest load among the at least one candidate storage pool that satisfies the second preset condition. For example, assuming that the at least one candidate storage pool that satisfies the second preset condition includes the storage pool A and storage pool B, and that the load of the storage pool A is 50% and the load of the storage pool B is 30%, the processing device 120 may determine that the storage pool B is a selected storage pool. By determining the candidate storage pool with the lowest load as the storage pool, a load balance degree of the plurality of candidate storage pools may be improved, thereby reducing a possibility of a node breakdown due to a load imbalance, and improving a storage reliability.

[0080] In some embodiments, the storage pool determined by the processing device 120 may be a candidate storage pool with the best communication quality among the at least one candidate storage pool that satisfies the second preset condition. The communication quality may be determined based on a network available bandwidth, a network stability, etc. For example, the higher the network available bandwidth, the better the network stability, the better the communication quality.

[0081] In some embodiments, the storage pool determined by the processing device 120 may be a candidate storage pool with the greatest free storage space among the at least one candidate storage pool that satisfies the second preset condition. The free storage space may refer to a count of available storage units on the first layer of the storage pool. For example, assuming that the at least one candidate storage pool that satisfies the second preset condition includes a storage pool C and a storage pool D, the first layer of the storage pool C includes 80 available storage units, and the first layer of the storage pool D includes 40 available storage units, the processing device 120 may determine the storage pool C as the selected storage pool.

[0082] In some embodiments, the processing device 120 may determine an evaluated value for each candidate storage pool based on at least two of the load, the communication quality, and the free storage space, and determine the candidate storage pool that satisfies the second preset condition with the highest evaluated value of as the storage pool. Specifically, the processing device 120 may score the candidate storage pool based on each of the load, the communication quality, and the free storage space, and perform a weighted summation on the plurality of scores to obtain the evaluation value. Through determining the storage pool by combining a plurality of conditions, a more balanced performance quality of the plurality of candidate storage pools may be achieved, and the plurality of candidate storage pools may have better reliabilities.

[0083] In 530, the first layer of the storage pool is taken as the target storage layer.

[0084] In some embodiments, the processing device 120 may take the first layer of the storage pool determined in operation 520 as the target storage layer. For example, the processing device 120 may take the rack layer of the storage pool B or the storage pool C as the target storage layer.

[0085] In the embodiments of the present disclosure, by determining the storage pool based on the load, the communication quality, and/or the free storage space from the at least one candidate storage pool that satisfies the second preset condition, and taking the highest layer (the first layer) of the storage pool as the target storage layer, the data storage may be failure-tolerant and the load balance degree of the plurality of candidate storage pools may be improved, and the data reliability and the data availability of the storage are improved.

[0086] In 540, at least one available storage pool is determined from the plurality of candidate storage pools.

[0087] The available storage pool may be determined based on at least one of the load, the communication quality, and the free storage space. For example, the processing device 120 may select, from the plurality of candidate storage pools, a candidate storage pool with a current load satisfying a load requirement (e.g., the current load is less than a preset load threshold, etc.) or communication quality satisfying a communication requirement, or in which a count of free storage units in the first layer of the candidate storage pool is greater than a preset threshold, as the available storage pool.

[0088] In 550, the storage pool is determined from the at least one available storage pool.

[0089] Specifically, the processing device 120 may determine the storage pool based on at least one of the load, the communication quality, or the free storage space of the at least one available storage pool.

[0090] In some embodiments, the storage pool determined by the processing device 120 may be an available storage pool with the lowest load in the at least one available storage pool. By determining the available storage pool with the lowest load as the storage pool, load balances of the plurality of candidate storage pools may be improved, thereby reducing the possibility of a node breakdown due to a load imbalance, and improving the storage reliability. In some embodiments, a manner of determining the storage pool from the at least one available storage pool may be similar to the manner of determining the storage pool from the at least one candidate storage pool that satisfies the second preset condition in operation 520, which is not repeated here.

[0091] In 560, the target storage layer is determined from the second layer and layers below the second layer of the storage pool.

[0092] In some embodiments, the processing device 120 may determine, from the second layer and layers below the second layer of the storage pool determined in operation 550, a layer that satisfies a preset condition (e.g., a third preset condition) as the target storage layer. For more contents on how to determine the target storage layer from the second layer and the layers below the second layer of the storage pool, please refer to FIG. 6.

[0093] In the embodiments of the present disclosure, by determining the available storage pool from the at least one candidate storage pool based on the load, the communication quality, and/or the free storage space, determining the storage pool from the at least one available storage pool, and determining the target storage layer from the second layer and layers below the second layer of the storage pool, the determined target storage layer may satisfy a failure tolerance requirement, thereby improving the data storage reliability and the data availability.

[0094] FIG. 6 is a flowchart illustrating an exemplary process 600 for determining a target storage layer according to some embodiments of the present disclosure. In some embodiments, at least part of the process

600 may be performed to achieve at least part of operation 560 as described in connection with FIG. 5. For example, the processing device 120 or the storage layer determination module 220 may determine the target storage layer from a second layer and layers below the second layer of the storage pool by performing at least part of the process 600.

[0095] In 610, the second level is taken as a current level.

[0096] In some embodiments, after step 550, the processing device 120 may take the second layer (e.g., a node layer) of the storage pool as the current layer.

[0097] In 620, whether the current layer satisfies a third preset condition is determined.

[0098] The third preset condition is related to a count of storage units in the current layer and a count of data blocks. For example, the third preset condition may be that the count of the available storage units in the current layer is greater than or equal to the count of the data blocks. The processing device 120 may obtain the count of the available storage units in the current layer, and compare the count to the count of the data blocks to determine whether the current layer satisfies the third preset condition. By using the third preset condition related to the count of the storage units in the current layer and the count of the data blocks, the target storage layer that satisfies a failure tolerance requirement may be obtained, thereby ensuring a data storage reliability.

[0099] In some embodiments, in response to that the current layer satisfies the third preset condition, the processing device 120 may determine the target storage layer by performing operation 630; in response to that the current layer does not satisfy the third preset condition, the processing device 120 may determine the target storage layer by performing operation 640.

[0100] In 630, the current layer is designated as the target storage layer.

[0101] In 640, a next layer of the current layer is determined as a new current layer.

[0102] In some embodiments, in response to that the current layer does not satisfy the third preset condition, the processing device 120 may determine the next layer of the current layer as a new current layer, return to reperform operation 620, and determine whether the new current layer satisfies the third preset condition until the target storage layer is determined. It may be appreciated that the target storage layer determined by operation 640 may be a third layer (e.g., a hard disk pool layer) or a fourth layer (e.g., a hard disk layer).

[0103] In the embodiments of the present disclosure, by starting from the second layer of the storage pool, determining whether the current layer satisfies the third preset condition layer by layer downward, and determining the current layer that satisfies the third preset condition as the target storage layer, so that the highest possible storage layer may be obtained and taken as the target storage layer, and the higher the layer of the target storage layer, the stronger the failure tolerance of the data storage, thus maximizing the reliability of the data storage and the availability of the data.

[0104] FIG. 7 is a schematic diagram illustrating an exemplary first selection strategy 700 according to some embodiments of the present disclosure. In some embodiments, after operation 320, i.e., after determining the target storage layer, the processing device 120 may, based on the target storage layer, use the first selection strategy to select a plurality of storage units from the storage pool as the plurality of target storage units. The first selection strategy may include at least one of a first sub-strategy or a second sub-strategy. For example, as shown in FIG. 7, a first selection strategy 700 may include a first sub-strategy 710 and a second sub-strategy 720.

[0105] The first sub-strategy 710 includes that in a situation where an upper layer of the target storage layer

does not satisfy a fourth preset condition, the plurality of target storage units are distributed in all the storage units corresponding to the upper layer of the target storage layer. The fourth preset condition is related to a count of storage units in the upper layer of the target storage layer and the count of the data blocks. For example, the fourth preset condition may be that a count of available storage units in the upper layer of the target storage layer is greater than or equal to the count of data blocks. In some embodiments, the first sub-strategy 710 may further include that in a situation where the upper layer of the target storage layer satisfies the fourth preset condition, the plurality of target storage units are distributed in a first count of storage units corresponding to the upper layer of the target storage layer. For example, when the fourth preset condition is that the count of the available storage units in the upper layer of the target storage layer is greater than or equal to the count of the data blocks, the first count may be the count of the data blocks, i.e., the count of the target storage units is equal to the count of the data blocks.

[0106] For example, when the target storage layer is a second layer (e.g., a node layer), the layer above the target storage layer is a first layer (e.g., a rack layer). In a situation where the upper layer of the target storage layer does not satisfy the fourth preset condition, the plurality of target storage units are distributed in all the storage units corresponding to the upper layer of the target storage layer. In a situation where the upper layer of the target storage layer satisfies the fourth preset condition, the target storage units are distributed on a portion of the storage units corresponding to the upper layer of the target storage layer and a count of the portion of the storage units is equal to the count of data blocks. Merely by way of example, assuming that the count of data blocks is 12, the target storage layer is a node layer, the layer above the target storage layer is a rack layer, and the count of racks (e.g., 9, 10, etc.) is less than or equal to 12, the target storage units are distributed in all racks in the rack layer, and at least one target storage unit is distributed in each rack. If the count of racks (e.g., 15, 16, etc.) is greater than 12, 12 racks are selected from all the racks in the rack layer, and the target storage units are distributed in the selected 12 racks, and one target storage unit is distributed in each rack. By using the fourth preset condition related to the count of the storage units corresponding to the upper layer of the target storage layer and the count of the data blocks, as many count of target storage units as possible may be obtained, thereby maximizing a satisfaction of the failure tolerance requirements, which in turn improves the reliability of the data storage.

[0107] The second sub-strategy 720 includes that a difference of the count of the target storage units between each two of the plurality of distributed storage units is less than a preset count. The preset count may be determined in various ways. For example, the preset count may be set manually, determined by a preset algorithm or a machine learning model, etc. The plurality of distributed storage units are storage units of the upper layer of the target storage layer and contain the plurality of target storage units. For example, if the target storage layer is the node layer, the distributed storage units are storage units included in the rack layer (e.g., rack 1 to rack N1 shown in FIG. 4) and the target storage units are included in the plurality of distributed storage units. For example, if the count of data blocks is 15 (i.e., 15 target storage units) and the count of racks in the rack layer is 5 (e.g., N1=5), and the preset count is 3, 15 target storage units may be distributed in the 5 racks with a difference of less than 3 in the count of target storage units distributed in each two racks. By limiting the difference of the count of target storage units between each two of the plurality of distributed storage units to be less than the preset count, the difference in the count of target storage units in different distributed storage units may not be too great, so as to avoid an unbalanced load on the distributed storage units, thus improving a

load balancing and improving the data storage reliability.

[0108] In some embodiments, the preset count is related to an access safety degree of at least one of the plurality of distributed storage units. For example, the preset count = k*fluctuation/fluctuation threshold, wherein the fluctuation may be related to a fluctuation of the access safety degree of all of the distributed storage units, and the fluctuation = standard deviation of the fluctuation of the access safety degrees of all distributed storage units / mean value of the fluctuation of the access safety degrees of all distributed storage units; k denotes a constant indicating an order of magnitude; and the fluctuation threshold is a preset value indicating an acceptable fluctuation of the access security degree. The access safety degree refers to a numerical value used to measure an access safety of the storage unit, the higher the value, the safer. The access safety degree is related to an access probability and an access type of at least one distributed storage unit. For example, the access safety degree of a distributed storage unit = access type component + w*access probability, where w denotes a weight. The access type includes read-only, write-only, and readable-writable, and the access type component is 1 when the access type is read-only, and the access type component is 0 when the access type is write-only and readable-writable. The processing device 120 may calculate the access probability for each storage unit of the distributed storage unit, and take an average value of the access probabilities of all storage units as the access probability of the distributed storage unit. The access probability of the distributed storage unit = a count of times the data of that storage unit is accessed/a total count of accesses of all distributed storage units. By determining the preset count based on the access safety degree of the distributed storage unit, the access safety degree of the data of different distributed storage units may be located at the same level, thereby improving the load balancing of the storage units and improving the data storage reliability.

[0109] In some embodiments, the first sub-strategy and the second sub-strategy may be used in combination or separately. Merely by way of example, the count of data blocks is 12, the target storage layer is the node layer, and the count of racks corresponding to the rack layer is 4. If the first selection strategy includes only the second sub-strategy, i.e., without considering the target storage units to be dispersed as much as possible, any count of racks corresponding to the rack layer (which needs to be less than the count of the data blocks 12) may be used as the distributed storage units. For example, the processing device 120 may take 2 racks as the distribution storage units, namely distribution storage unit 1 and distribution storage unit 2, respectively, and correspondingly, 6 nodes in each distribution storage unit may be taken as the target storage unit, or 5 nodes in the distribution storage unit 1 and 7 nodes in the distribution storage unit 2 may be taken as the target storage unit. If the first selection strategy includes the first sub-strategy and the second sub-strategy, i.e., it is necessary to consider that the target storage units are dispersed as much as possible, all 4 racks corresponding to the rack layer may be used as the distribution storage units, i.e., the count of distribution storage units is 4, namely distribution storage unit 1, distribution storage unit 2, distribution storage unit 3, and distribution storage unit 4, respectively. Correspondingly, the processing device 120 may take 3 nodes in each distribution storage unit as the target storage units, or 2 nodes in the distributed storage unit 1, 4 nodes in the distributed storage unit 2, 4 nodes in the distributed storage unit 3, and 2 nodes in the distributed storage unit 4 as the target storage units.

[0110] In some embodiments, the load balancing is ensured by a combination of the first sub-strategy and the second sub-strategy, so as to satisfy the data failure tolerance requirement while improving the data storage reliability.

[0111] FIG. 8 is a flowchart illustrating an exemplary process 800 for storing a plurality of data blocks

according to some embodiments of the present disclosure. In some embodiments, at least part of the process 800 may be performed to achieve at least part of operation 340 as described in connection with FIG. 3. For example, the processing device 120 or the data block storage module 240 may store the plurality of data blocks based on the plurality of target storage units by performing at least part of process 800.

[0112] In some embodiments, in response to that a target storage layer is not a last layer in a multi-layer architecture of a storage pool, for each of the plurality of target storage units, the processing device 120 may determine, by performing operations 810-830, a write storage unit corresponding to the target storage unit. The write storage unit is an actual storage unit for the data block. As shown in FIG. 8, the target storage unit includes target storage units 1-n (n>1), and operations 810-830 in process 800 are described using the target storage unit 1 as an example.

[0113] In 810, the target storage unit 1 is taken as a write storage unit.

[0114] In 820, whether a layer corresponding to the write storage unit is the last layer is determined.

[0115] The last layer is a bottom layer in the storage pool, i.e., the fourth layer. For example, if the write storage unit is a hard disk, the layer corresponding to the write storage unit is a storage hardware layer, which is the last layer. As another example, if the write storage unit is a node, the layer corresponding to the write storage unit is a node layer, which is not the last layer.

[0116] In some embodiments, when the layer corresponding to the write storage unit is not the last layer, the processing device 120 may select a new write storage unit by performing operation 830. When the layer corresponding to the write storage unit is the last layer, the processing device 120 may store the data block to the write storage unit by performing operation 840.

[0117] In 830, the new write storage unit is selected from storage units in a next layer contained in the write storage unit based on a second selection strategy.

[0118] The second selection strategy may include determining a plurality of candidate queues from storage units of the next layer contained in the write storage unit, determining a target candidate queue from the plurality of candidate queues, and determining the new write storage unit from the plurality of storage units for the target candidate queue. For more contents on how to determine the new write storage unit based on the second selection strategy, please refer to FIG. 9.

[0119] In some embodiments, after determining the new write storage unit, the processing device 120 may return to reperform operation 820 to determine whether the layer corresponding to the new write storage unit is the last layer until the layer corresponding to the new write storage unit is determined to be the last layer.

[0120] In 840, the plurality of data blocks are respectively stored in write storage units corresponding to the plurality of target storage units.

[0121] In some embodiments, after determining the write storage units corresponding to all the target storage units, the processing device 120 may store the plurality of data blocks in the write storage units corresponding to the plurality of target storage units, respectively. Specifically, the processing device 120 may store each of the plurality of data blocks into a single write storage unit, i.e., each write storage unit stores only one data block.

[0122] In the embodiments of the present disclosure, by starting from the target storage layer to determine the write storage units layer by layer downward, so that only one data block is stored in one write storage unit of each layer, thereby storing the data blocks as dispersed as possible, when any one or more of the storage units fail, a possibility that the data to be stored is unable to be restored may be reduced, thereby improving a data

storage reliability.

[0123] FIG. 9 is a flowchart illustrating an exemplary process 900 for selecting a new write storage unit according to some embodiments of the present disclosure. In some embodiments, at least part of the process 900 may be performed to achieve at least part of operation 830 as described in connection with FIG. 8. For example, the processing device 120 or the data block storage module 240 may select, based on a second selection strategy, the new write storage unit from a next layer of storage units contained in a write storage unit by performing at least part of process 900.

[0124] In 910, a plurality of candidate queues are determined from the storage units in the next layer contained in the write storage unit.

[0125] In some embodiments, after operation 820, in response to that the layer corresponding to the write storage unit is not the last layer, the processing device 120 may determine the plurality of candidate queues from the next layer of storage units contained in the write storage unit. The plurality of candidate queues include a plurality of updateable storage units. Specifically, the processing device 120 may determine a plurality of candidate storage units from the next layer of available storage units contained in the write storage unit, and randomly arrange these candidate storage units in a different order to obtain the plurality of candidate queues. The updatable refers to that each of the storage units in the candidate queue may be replaced by other available storage units in the next layer contained in the write storage unit.

[0126] It may be appreciated that each of the available storage units has different amounts of free spaces and different load capacities. In some embodiments, the processing device 120 may determine the available storage units that satisfy at least one of a relatively great free space or a relatively low load as the candidate storage units. A count of the candidate storage units should be less than a count of the available storage units. For example, the count of the candidate storage units may be 80% of the count of available storage units, with the remaining 20% forming an alternative queue as spare storage units. In some embodiments, the processing device 120 may regularly or irregularly (e.g., each time when a data storage is performed) swap a candidate storage unit with the highest load in the candidate queue and a spare storage unit with the greatest free space in the candidate queue, thereby ensuring that a used space and the load is balanced across all of the storage units of the system.

[0127] In some embodiments, when the plurality of candidate storage units are randomly arranged, a count of times of random arrangement may be the same as the count of the candidate queues.

[0128] In some embodiments, the plurality of candidate queues may be determined based on a current storage task and a historical storage task. For example, if a similarity degree between the current storage task and the historical storage task exceeds a similarity degree threshold, the candidate queues for the current storage task may be the same or similar to the candidate queues for the historical storage task. The processing device 120 may construct vectors based on the current storage task and the historical storage task, respectively, and calculate a vector similarity degree and take the vector similarity as the similarity degree between the current storage task and the historical storage task. The similarity degree threshold may be a preset value, which is set based on experience, etc.

[0129] In 920, a target candidate queue is determined from the plurality of candidate queues.

[0130] In some embodiments, the processing device 120 may determine a queue priority for at least one of the plurality of candidate queues based on load information of each of the plurality of candidate queues. The load information of a candidate queue may at least include a count of free storage units in the candidate queue.

The queue priority is a numerical value used to measure how high or how low the priority of the queue is, the higher the value, the higher the priority. Merely by way of example, assuming that there are n (n>1) count of candidate queues, the processing device 120 may obtain the load information for each candidate queue, obtain, from the load information, a count of free storage units for the each candidate queue; and sort the n count of candidate queues in ascending order according to the count of free storage units for the each candidate queue. The queue priority of the each candidate queue is a sequence number of the candidate queue obtained by the sorting.

[0131] In some embodiments, the processing device 120 may determine, based on the load information, a load balance degree of the at least one candidate queue. The load balance degree refers to a numerical value used to measure whether the load in the queue is even and reasonable, and the greater the value, the more balanced and reasonable the load. For example, if there are a total of 20 storage units in a candidate queue, in which 18 storage units have been used for storage, and a space utilization of the storage units that have been used for storage is greater than 80%, the load balance degree may be considered as balanced and reasonable. As another example, there are a total of 20 storage units in a candidate queue, in which 18 storage units have been used for storage, and the space utilization of the storage units that have been used for storage is less than 10%, the load may be considered as imbalance and unreasonable. As another example, if there are a total of 20 storage units in a candidate queue, in which 2 storage units have been used for storage, the space utilization of the storage units that have been used for storage is greater than 80%), the load may be considered as imbalance and unreasonable. Merely by way of example, for a certain candidate queue, a load balance degree = w1*a percentage of a count of storage units with a 1st space utilization + w2*a percentage of a count of storage units with a 2nd space utilization + w3*a percentage of a count of storage units with a 3rd space utilization + ..., + wn*a percentage of a count of storage units in nth space utilization, n>1, the 1st to nth space utilizations are 0 to (100*1/n) %, (100*1/n) % to (100*2/n) %, ..., (100*(n-1)/n) % to 100%. w1, w2, w3, ..., wn are all preset values and w1 <w2<w3<...<wn.

[0132] Further, the processing device 120 may determine the queue priority of the at least one candidate queue based on the load balance degree of the at least one candidate queue. For example, the processing device 120 may sort the at least one candidate queue based on the load balance degree from smallest to greatest, and use the sequence number of each candidate queue obtained by the sorting as the queue priority for the candidate queue.

[0133] In some embodiments, the processing device 120 may determine a target candidate queue based on the queue priority of the at least one candidate queue. For example, the processing device 120 may take the candidate queue with the highest queue priority as the target candidate queue. By determining the target candidate queue based on the queue priority determined based on the load information, the data to be stored may be stored in the storage unit with the greatest free space, which results in a more balanced utilization of the storage space of each storage unit, and an improved load balancing. At the same time, a possibility of a node breakdown due to insufficient free space is reduced, thus improving the storage reliability.

[0134] In some embodiments, for a plurality of candidate queues, the processing device 120 may cyclically select one of the candidate queues as the target candidate queue according to a certain order. For example, the processing device 120 may cyclically select one of the candidate queues as the target candidate queue in a sorted order according to the queue priority. Merely by way of example, assuming that there are three candidate queues sorted as candidate queue 1, candidate queue 2, and candidate queue 3, then when selecting the target

candidate queue for the first time, the candidate queue 1 may be selected; and when selecting the target candidate queue for a second time and a third time, the candidate queue 2 and the candidate queue 3 may be selected, and when selecting the target candidate queue for a fourth time, the candidate queue 1 may be selected again, and so on. The above operations prevent some storage units from being unselected for a long time, which makes the access to each storage unit more balanced.

[0135] In 930, the new write storage unit is determined from the plurality of storage units of the target candidate queue.

[0136] In some embodiments, the processing device 120 may determine the new write storage unit from the plurality of storage units of the target candidate queue in a variety of ways. For example, the processing device 120 may determine the new write storage unit based on a size of free space, an access frequency, etc. of each storage unit.

[0137] In some embodiments, the processing device 120 may obtain load information of each of the plurality of storage units of the target candidate queue. The load information may include the size of a free space of the corresponding storage unit. The processing device 120 may determine a unit priority of at least one of the plurality of storage units based on the load information. For example, the processing device 120 may sort the plurality of storage units of the target candidate queue based on an order of the free spaces of the storage units from smallest to greatest, and use a sequence number of each storage unit obtained by the sorting as a unit priority of the storage unit. The processing device 120 may determine the new write storage unit based on the unit priority of the at least one storage unit. For example, the processing device 120 may determine a storage unit with the highest unit priority as the new write storage unit. By measuring the priority of each storage unit based on the size of free space in the storage unit, the storage unit in which the data is subsequently written to is able to satisfy a storage requirement, thereby improving the storage space utilization.

[0138] In some embodiments, the load information may include an access frequency of the corresponding storage unit. The processing device 120 may determine the unit priority for the at least one of the plurality of storage units based on the access frequency. For example, the processing device 120 may sort the plurality of storage units of the target candidate queue based on the access frequencies of the storage units in descending order, and take the sequence number of each storage unit obtained by the sorting as the unit priority of the storage unit. By measuring the priority of each storage unit based on the access frequency of the each storage unit, the access frequency of each storage unit may be able to converge in the subsequent data reading and writing process, thereby improving the load balance.

[0139] In the embodiments of the present disclosure, by determining the plurality of candidate queues, determining the target candidate queue from these candidate queues, and determining the write storage unit from the plurality of storage units of the target candidate queue, the storage unit may be able to be selected as evenly as possible, and the storage units is accessed in a more balanced manner, which improves the load balance and, in turn, improves the storage reliability.

[0140] FIG. 10 is a schematic diagram illustrating an exemplary process 1000 for data storage according to some embodiments of the present disclosure. In some embodiments, the process 1000 may be implemented by the data storage system 100 and/or the data storage system 200. For example, the process 1000 may be implemented as a set of instructions (e.g., an application) stored in a storage device (e.g., the storage device 110 illustrated in FIG. 1). In some embodiments, the processing device 120 of the data storage system 100 and/or

one or more modules of the data storage system 200 may execute the set of instructions and may accordingly be directed to perform the process 1000.

[0141] As shown in FIG. 10, at the start, the processing device 120 may determine, by operation 1010, whether there is a storage pool with N+M count of racks. N may be a count of first data blocks of data to be stored, M may be a count of second blocks of the data to be stored, and N+M is a count of the data blocks of the data to be stored. When there is a storage pool with N+M counts of racks, the processing device 120 may continue to perform operation 1020; when there is no storage pool with N+M counts of racks, the processing device 120 may perform operation 1060 to degrade a target storage layer (a failure tolerant layer).

[0142] The processing device 120 may select a storage pool from available storage pools and determine the storage pool as a rack layer failure tolerant (i.e., the target storage layer is the first layer) by performing operation 1020. The available storage pools are storage pools with N+M counts of racks.

[0143] The processing device 120 may select N+M counts of racks, each of which corresponds to one data block, from the racks of the selected storage pool by performing operation 1021.

[0144] The processing device 120 may select one node (i.e., the target storage unit) for each of the N+M counts of racks based on loads and capacities of nodes corresponding to each rack by performing operation 1022 to obtain N+M counts of nodes. For more contents on how to select the target storage unit, please refer to operation 330 and FIG. 7.

[0145] After operation 1022, the processing device 120 may select one hard disk pool for each of the N+M counts of nodes based on loads and capacities of hard disk pools corresponding to each node to obtain the N+M counts of hard disk pools by performing step 1030.

[0146] The processing device 120 may select one hard disk for each of the N+M counts of hard disk pools based on loads and capacities of hard disks corresponding to each hard disk pool to obtain the N+M counts of hard disks by performing operation 1030.

[0147] After operation 1030, the processing device 120 may write data to the N+M counts of hard disks by performing operation 1040, i.e., storing the data blocks of the data to be stored to the hard disks and end the process 1000. For more contents on how to store the data blocks, please refer to operation 340, FIG. 8, and FIG. 9.

[0148] After operation 1010, when there is no storage pool with N+M counts of racks, the processing device 120 may select a storage pool from the available storage pools by performing operation 1060.

[0149] By performing operation 1061, the processing device 120 may degrade to a node layer failure tolerance (i.e., the target storage layer is a second layer), and select a rack that includes as many nodes as possible from the available racks.

[0150] By performing operation 1062, the processing device 120 may determine whether a count of nodes corresponding to the selected rack is greater than or equal to N+M. When the count of nodes corresponding to the selected rack is greater than or equal to N+M, the processing device 120 may continue to perform operation 1030; when the count of nodes corresponding to the selected rack is smaller than N+M, the processing device 120 may perform operation 1063 to degrade the target storage layer again.

[0151] The processing device 120 may select a node that includes as many hard disk pools as possible from the available nodes (i.e., available storage units) by performing operation 1063, and degrade to a hard disk pool layer failure tolerance (i.e., the target storage layer is a third layer).

[0152] By performing operation 1064, the processing device 120 may determine whether a count of the hard disk pools corresponding to the selected node is greater than or equal to N+M. When the count of the hard disk pools corresponding to the selected node is greater than or equal to N+M, the processing device 120 may continue to perform operation 1050; when the count of the hard disk pools corresponding to the selected node is smaller than N+M, the processing device 120 may perform operation 1065 to degrade the target storage layer again.

[0153] By performing operation 1065, the processing device 120 may degrade to a hard disk layer failure tolerance (i.e., the target storage layer is the fourth layer), select N+M counts of hard disks from available hard disks, and then perform operation 1050.

[0154] The operations of the illustrated processes 300, 500, 600, 800, 900 and 1000 presented above are intended to be illustrative. In some embodiments, a process may be accomplished with one or more additional operations not described, and/or without one or more of the operations discussed. Additionally, the order in which the operations of a process described above is not intended to be limiting.

[0155] Having thus described the basic concepts, it may be rather apparent to those skilled in the art after reading this detailed disclosure that the foregoing detailed disclosure is intended to be presented by way of example only and is not limiting. Various alterations, improvements, and modifications may occur and are intended to those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested by this disclosure, and are within the spirit and scope of the exemplary embodiments of the present disclosure.

[0156] Moreover, certain terminology has been used to describe embodiments of the present disclosure. For example, the terms "one embodiment," "an embodiment," and/or "some embodiments" may mean that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present disclosure. Therefore, it is emphasized and should be appreciated that two or more references to "an embodiment" or "one embodiment" or "an alternative embodiment" in various portions of the present disclosure are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures or characteristics may be combined as suitable in one or more embodiments of the present disclosure.

[0157] Further, it will be appreciated by those skilled in the art that aspects of the present disclosure may be illustrated and described herein in any of a count of patentable classes or context including any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof. Accordingly, the aspects of the present disclosure may be entirely implemented on hardware, entirely implemented on software (including firmware, resident software, micro-code, etc.) or a combination of the software and the hardware. Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable media having computer readable program code embodied thereon.

[0158] A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as a portion of carrier wave. Such a propagated signal may take any of a variety of forms, including electro-magnetic, optical, etc., or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that is able to communicate, propagate, or transport a program for use by or in

connection with an instruction execution system, apparatus, or a device. Program code embodied on a computer readable signal medium may be transmitted using any appropriate medium, including wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

[0159] Computer program code for carrying out operations for aspects of the present disclosure may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Scala, Smalltalk, Eiffel, JADE, Emerald, C++, C#, VB. NET, Python etc., conventional procedural programming languages, such as the "C" programming language, Visual Basic, Fortran 2103, Perl, COBOL 2102, PHP, ABAP, dynamic programming languages such as Python, Ruby and Groovy, or other programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer, or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider) or in a cloud computing environment or offered as a service such as a Software as a Service (SaaS).

[0160] Furthermore, the recited order of processing elements or sequences, or the use of counts, letters, or other designations therefore, is not intended to limit the claimed processes and methods to any order except as may be specified in the claims. Although the above disclosure discusses through various examples what is currently considered to be a variety of useful embodiments of the present disclosure, it is to be understood that such detail is solely for that purpose, and that the appended claims are not limited to the disclosed embodiments, but, on the contrary, are intended to cover modifications and equivalent arrangements that are within the spirit and scope of the disclosed embodiments. For example, although the implementation of various components described above may be embodied in a hardware device, it may also be implemented as a software only solution, for example, an installation on an existing server or mobile device.

[0161] Similarly, it should be appreciated that in the foregoing description of embodiments of the present disclosure, various features are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure aiding in the understanding of one or more of the various inventive embodiments. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed object matter requires more features than are expressly recited in each claim. Rather, inventive embodiments lie in less than all features of a single foregoing disclosed embodiment.

[0162] In some embodiments, the counts expressing quantities or properties used to describe and claim certain embodiments of the present disclosure are to be understood as being modified in some instances by the term "about," "approximate," or "substantially." For example, the "about," "approximate," or "substantially" may indicate ±1%, ±5%, ±10%, or ±20% variation of the value it describes, unless otherwise stated. Accordingly, in some embodiments, the numerical parameters set forth in the written description and attached claims are approximations that varies depending upon the desired properties sought to be obtained by a particular embodiment. In some embodiments, the numerical parameters should be construed in light of the count of reported significant digits and by applying ordinary rounding techniques. Notwithstanding that the numerical ranges and parameters setting forth the broad scope of some embodiments of the application are approximations, the numerical values set forth in the specific examples are reported as precisely as practicable.

[0163] Each of the patents, patent applications, publications of patent applications, and other material, such as articles, books, specifications, publications, documents, things, and/etc., referenced herein is hereby incorporated herein by this reference in its entirety for all purposes, excepting any prosecution file history associated with same, any of same that is inconsistent with or in conflict with the present document, or any of same that may have a limiting effect as to the broadest scope of the claims now or later associated with the present document. By way of example, should there be any inconsistency or conflict between the description, definition, and/or the use of a term associated with any of the incorporated material and that associated with the present document, the description, definition, and/or the use of the term in the present document shall prevail.

[0164] In closing, it is to be understood that the embodiments of the application disclosed herein are illustrative of the principles of the embodiments of the application. Other modifications that may be employed may be within the scope of the present disclosure. Thus, by way of example, but not of limitation, alternative configurations of the embodiments of the application may be utilized in accordance with the teachings herein. Accordingly, embodiments of the present disclosure are not limited to that precisely as shown and described.

WHAT IS CLAIMED IS:

1. A method performed by a computing device with at least one processor and at least one storage device, the method comprising:

generating a plurality of data blocks based on data to be stored;

determining a target storage layer for the plurality of data blocks in a storage pool of the at least one storage device, wherein the storage pool has a multi-layer architecture including a plurality of storage layers, each of the plurality of storage layers in the storage pool corresponds to a plurality of storage units, each storage unit in each storage layer contains a plurality of storage units in a next storage layer of the storage layer, the target storage layer is a storage layer in the storage pool that satisfies a first preset condition, and the first preset condition is related to a count of storage units in the storage pool and a count of the data blocks;

determining a plurality of target storage units from the storage pool based on the target storage layer; and

storing the plurality of data blocks based on the plurality of target storage units.

2. The method of claim 1, wherein in the multi-layer architecture of the storage pool, the plurality of storage layers from top to bottom include a first layer, a second layer, a third layer, and a fourth layer.

3. The method of claim 2, wherein the storage pool is one of a plurality of candidate storage pools of a resource pool, the resource pool being configured to store data of a same type.

4. The method of claim 3, wherein the determining a target storage layer for the plurality of data blocks in a storage pool includes:

determining whether there is at least one candidate storage pool that satisfies a second preset condition in the plurality of candidate storage pools;

in response to that there is the at least one candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools,

determining the storage pool from the at least one candidate storage pool that satisfies the second preset condition; and

taking the first layer of the storage pool as the target storage layer.

5. The method of claim 4, wherein the storage pool is a candidate storage pool with a lowest load among the at least one candidate storage pool that satisfies the second preset condition.

6. The method of claim 4 or 5, wherein the determining a target storage layer for the plurality of data blocks in a storage pool further includes:

in response to that there is no candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools,

determining at least one available storage pool from the plurality of candidate storage pools;

determining the storage pool from the at least one available storage pool; and

determining the target storage layer from the second layer and layers below the second layer of the storage pool.

7. The method of claim 6, wherein the storage pool is an available storage pool with a lowest load among the at least one available storage pool.

8. The method of any one of claims 4-7, wherein the second preset condition is related to a count of storage units in the first layer of the storage pool and the count of the data blocks.

9. The method of claim 6 or 7, wherein the determining the target storage layer from the second layer and layers below the second layer of the storage pool includes:

taking the second layer as a current layer;

determining whether the current layer satisfies a third preset condition;

in response to that the current layer satisfies the third preset condition, designating the current layer as the target storage layer; and

in response to that the current layer does not satisfy the third preset condition, determining a next layer of the current layer as a new current layer, and re-determining whether the new current layer satisfies the third preset condition until the target storage layer is determined.

10. The method of claim 9, wherein the third preset condition is related to a count of storage units in the current layer and the count of the data blocks.

11. The method of any one of claims 1-10, wherein the determining a plurality of target storage units from the storage pool based on the target storage layer includes:

selecting a plurality of storage units from the storage pool as the plurality of target storage units based on the target storage layer using a first selection strategy, wherein

the first selection strategy includes at least one of a first sub-strategy or a second sub-strategy, and

the first sub-strategy includes that:

in a situation where an upper layer of the target storage layer does not satisfy a fourth preset condition, the plurality of target storage units are distributed in all the storage units corresponding to the upper layer of the target storage layer; and

in a situation where the upper layer of the target storage layer satisfies the fourth preset condition, the plurality of target storage units are distributed in a first count of storage units corresponding to the upper layer of the target storage layer, the first count being equal to the count of the data blocks, and

the second sub-strategy includes that:

a difference of a count of target storage units between each two of a plurality of distributed storage units is less than a preset count, and the plurality of distributed storage units are storage units of the upper layer of the target storage layer and contain the plurality of target storage units.

12. The method of claim 11, wherein the fourth preset condition is related to a count of storage units in the upper layer of the target storage layer and the count of the data blocks.

13. The method of any one of claims 1-12, wherein the storing the plurality of data blocks based on the plurality of target storage units includes:

in response to that the target storage layer is not a last layer in the multi-layer architecture of the storage pool, for each of the plurality of target storage units,

taking the target storage unit as a write storage unit;

determining whether a layer corresponding to the write storage unit is the last layer;

in response to that the layer corresponding to the write storage unit is not the last layer, selecting, based on a second selection strategy, a new write storage unit from storage units in a next layer contained in the write storage unit;

determining whether a layer corresponding to the new write storage unit is the last layer until the layer corresponding to the new write storage unit is determined to be the last layer; and

respectively storing the plurality of data blocks in write storage units corresponding to the plurality of target storage units.

14. The method of claim 13, wherein the selecting, based on a second selection strategy, a new write storage unit from storage units in a next layer contained in the write storage unit includes:

determining a plurality of candidate queues from the storage units in the next layer contained in the write storage unit, the plurality of candidate queues being consist of a plurality of updatable storage units;

determining a target candidate queue from the plurality of candidate queues; and

determining the new write storage unit from a plurality of storage units of the target candidate queue.

15. The method of claim 14, wherein the determining a target candidate queue from the plurality of candidate queues includes:

determining, based on load information of each of the plurality of candidate queues, a queue priority of at least one of the plurality of candidate queues, the load information at least including a count of free storage units in the corresponding candidate queue; and

determining the target candidate queue based on the queue priority of the at least one candidate queue.

16. A system including at least one processor and at least one storage device storing computer instructions, when the computer instructions are executed by the at least one processor, the at least one processor is configured to perform operations including:

generating a plurality of data blocks based on data to be stored;

determining a target storage layer for the plurality of data blocks in a storage pool of the at least one storage device, wherein the storage pool has a multi-layer architecture including a plurality of storage layers, each of the plurality of storage layers in the storage pool corresponds to a plurality of storage units, each storage unit in each storage layer contains a plurality of storage units in a next storage layer of the storage layer, the target storage layer is a storage layer in the storage pool that satisfies a first preset condition, and the first preset condition is related to a count of storage units in the storage pool and a count of the data blocks;

determining a plurality of target storage units from the storage pool based on the target storage layer; and

storing the plurality of data in blocks based on the plurality of target storage units.

17. The system of claim 16, wherein in the multi-layer architecture of the storage pool, the plurality of storage layers from top to bottom include a first layer, a second layer, a third layer, and a fourth layer.

18. The system of claim 17, wherein the storage pool is one of a plurality of candidate storage pools of a resource pool, the resource pool being configured to store data of a same type.

19. The system of claim 18, wherein the determining a target storage layer for the plurality of data blocks in a storage pool includes:

determining whether there is at least one candidate storage pool that satisfies a second preset condition in the plurality of candidate storage pools;

in response to that there is the at least one candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools,

determining the storage pool from the at least one candidate storage pool that satisfies the second preset condition; and

taking the first layer of the storage pool as the target storage layer.

20. The system of claim 19, wherein the storage pool is a candidate storage pool with a lowest load among the at least one candidate storage pool that satisfies the second preset condition.

21. The system of claim 19 or 20, wherein the determining a target storage layer for the plurality of data blocks in a storage pool further includes:

in response to that there is no candidate storage pool that satisfies the second preset condition in the plurality of candidate storage pools,

determining at least one available storage pool from the plurality of candidate storage pools;

determining the storage pool from the at least one available storage pool; and

determining the target storage layer from the second layer and layers below the second layer of the storage pool.

22. The system of claim 21, wherein the storage pool is an available storage pool with a lowest load among the at least one available storage pool.

23. The system of any one of claims 19-22, wherein the second preset condition is related to a count of storage units in the first layer of the storage pool and the count of the data blocks.

24. The system of claim 21 or 22, wherein the determining the target storage layer from the second layer and layers below the second layer of the storage pool includes:

taking the second layer as a current layer;

determining whether the current layer satisfies a third preset condition;

in response to that the current layer satisfies the third preset condition, designating the current layer as the target storage layer; and

in response to that the current layer does not satisfy the third preset condition, determining a next layer of the current layer as a new current layer, and re-determining whether the new current layer satisfies the third preset condition until the target storage layer is determined.

25. The system of claim 24, wherein the third preset condition is related to a count of storage units in the current layer and the count of the data blocks.

26. The system of any one of claims 16-25, wherein the determining a plurality of target storage units from the storage pool based on the target storage layer includes:

selecting a plurality of storage units from the storage pool as the plurality of target storage units based on the target storage layer using a first selection strategy, wherein

the first selection strategy includes at least one of a first sub-strategy or a second sub-strategy, and

the first sub-strategy includes:

in a situation where an upper layer of the target storage layer does not satisfy a fourth preset condition, the plurality of target storage units are distributed in all the storage units corresponding to the upper layer of the target storage layer; and

in a situation where the upper layer of the target storage layer satisfies the fourth preset condition, the plurality of target storage units are distributed in a first count of storage units corresponding to the upper layer of the target storage layer, the first count being equal to the count of the data blocks, and

the second sub-strategy includes:

a difference of a count of target storage units between each two of a plurality of distributed storage units is less than a preset count, and the plurality of distributed storage units are storage units of the upper layer of the target storage layer and contain the plurality of target storage units.

27. The system of claim 26, wherein the fourth preset condition is related to a count of storage units in the upper layer of the target storage layer and the count of the data blocks.

28. The system of any one of claims 16-27, wherein the storing the plurality of data blocks based on the plurality of target storage units includes:

in response to that the target storage layer is not a last layer in the multi-layer architecture of the storage pool, for each of the plurality of target storage units,

taking the target storage unit as a write storage unit;

determining whether a layer corresponding to the write storage unit is the last layer;

in response to that the layer corresponding to the write storage unit is not the last layer, selecting, based on a second selection strategy, a new write storage unit from storage units in a next layer contained in the write storage unit;

determining whether a layer corresponding to the new write storage unit is the last layer until the layer corresponding to the new write storage unit is determined to be the last layer; and

respectively storing the plurality of data blocks in write storage units corresponding to the plurality of target storage units.

29. The system of claim 28, wherein the selecting, based on a second selection strategy, a new write storage unit from storage units in a next layer contained in the write storage unit includes:

determining a plurality of candidate queues from the storage units in the next layer contained in the write storage unit, the plurality of candidate queues being consist of a plurality of updatable storage units;

determining a target candidate queue from the plurality of candidate queues; and

determining the new write storage unit from a plurality of storage units of the target candidate queue.

30. The system of claim 29, wherein the determining a target candidate queue from the plurality of candidate queues includes:

determining, based on load information of each of the plurality of candidate queues, a queue priority of at least one of the plurality of candidate queues, the load information at least including a count of free storage units in the corresponding candidate queue; and

determining the target candidate queue based on the queue priority of the at least one candidate queue.

31. A system, including a data block generation module, a storage layer determination module, a storage unit determination module, and a data block storage module, wherein

the data block generation module is configured to generate a plurality of data blocks based on data to be stored;

the storage layer determination module is configured to determine a target storage layer for the plurality of data blocks in a storage pool of at least one storage device, wherein the storage pool has a multi-layer architecture including a plurality of storage layers, each of the plurality of storage layers in the storage pool corresponds to a plurality of storage units, and each storage unit in each storage layer contains a plurality of storage units in a next storage layer of the storage layer, the target storage layer is a storage layer in the storage pool that satisfies a first preset condition, and the first preset condition is related to a count of storage units in the storage pool and a count of the data blocks;

the storage unit determining module is configured to determine a plurality of target storage units from the storage pool based on the target storage layer;

the data block storage module is configured to store the plurality of data blocks based on the plurality of target storage units.

32. A computer-readable storage medium, the storage medium storing computer instructions, wherein when reading the computer instructions in the storage medium, a computer performs the method of any one of claims 1-15.