

METHODS, APPARATUSES, AND ELECTRONIC DEVICES FOR EVICTING MEMORY BLOCK IN CACHE

TECHNICAL FIELD

[0001] This document pertains to the field of data storage technologies, and in particular,
5 relates to methods, apparatuses, and electronic devices for evicting a memory block in a cache.

BACKGROUND

[0002] Currently, all storage systems release cache space based on a specific policy. For example, value of a memory block in a cache is evaluated based on an access temperature (the memory block is a storage unit of the cache), so a lower-value memory block can be cleared.
10 However, in many storage systems, a memory block does not merely store one element (the element is cache data). Therefore, there is a possibility that a small part of elements in one memory block is frequently accessed, and the other part of the elements are rarely accessed. Consequently, the memory block has a cold temperature on the whole, and is finally evicted by the storage system. Clearly, performance and gain of the cache are reduced if a frequently accessed element is cleared
15 from the cache.

[0003] In view of this, a solution is urgently needed currently, to avoid evicting an element with a high access frequency when the storage system evicts a memory block.

SUMMARY

[0004] Embodiments of this specification aim to provide methods, apparatuses and electronic
20 devices for evicting a memory block in a cache, , so that when the memory block in the cache is to be evicted, a valuable element in a memory block that needs to be evicted is migrated to another memory block, to avoid affecting performance and gain of the cache.

[0005] To achieve the above-mentioned objective, the embodiments of this specification are implemented as follows: According to a first aspect, a method for evicting a memory block in a

cache is provided, including: determining a temperature index value of the memory block in the cache, where the memory block is a storage unit of the cache; determining, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache; migrating a target element in the to-be-evicted memory block to
5 a target memory block in the cache, where the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and deleting the to-be-evicted memory block from the cache.

[0006] According to a second aspect, a method for evicting a memory block in a cache is provided, applied to a storage system of a log structured merge (LSM) tree storage architecture, and including: determining a temperature index value of a memory block (MemBlock) in a
10 memory table (MemTable) of an LSM tree of the storage system, where the MemTable is configured to store cache data of the storage system; determining, based on the temperature index value of the MemBlock in the MemTable, a to-be-evicted MemBlock that needs to be cleared in the MemTable; migrating a target element in the to-be-evicted MemBlock to a target MemBlock
15 in the MemTable, where the target MemBlock is a MemBlock that is retained in the MemTable and whose available storage space is greater than storage space needed by the target element; and deleting the to-be-evicted MemBlock from the MemTable.

[0007] According to a third aspect, an apparatus for evicting a memory block in a cache is provided, including: a temperature determining module, configured to determine a temperature
20 index value of the memory block in the cache, where the memory block is a storage unit of the cache; an eviction analysis module, configured to determine, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache; an element migration module, configured to migrate a target element in the to-be-evicted memory block to a target memory block in the cache, where the target memory block is a memory
25 block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and an eviction execution module, configured to delete the to-be-evicted memory block from the cache.

[0008] According to a fourth aspect, an electronic device is provided, including a storage, a processor, and a computer program that is stored in the storage and that can run on the processor, where the computer program is executed by the processor, to implement the following steps:
30 determining a temperature index value of a memory block in a cache, where the memory block is

a storage unit of the cache; determining, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache; migrating a target element in the to-be-evicted memory block to a target memory block in the cache, where the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and deleting the to-be-evicted memory block from the cache.

[0009] According to a fifth aspect, a computer-readable storage medium is provided. A computer program is stored in the computer-readable storage medium, and when the computer program is executed by a processor, the following steps are implemented: determining a temperature index value of a memory block in a cache, where the memory block is a storage unit of the cache; determining, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache; migrating a target element in the to-be-evicted memory block to a target memory block in the cache, where the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and deleting the to-be-evicted memory block from the cache.

[0010] In the solutions in the embodiments of this specification, the temperature index value of the memory block in the cache can be updated, and when cache space needs to be cleared, the to-be-evicted memory block that needs to be cleared is determined based on the updated temperature index value of the memory block. Before clearing, a valuable element in the to-be-evicted memory block is migrated to another memory block, to avoid a case in which a high-value element in the cache is lost due to clearing of the memory block, and consequently, performance and gain are reduced.

BRIEF DESCRIPTION OF DRAWINGS

[0011] To describe the technical solutions in embodiments of this specification or in an existing technology more clearly, the following briefly describes the accompanying drawings needed for describing the embodiments or the existing technology. Clearly, the accompanying drawings in the following descriptions merely show some embodiments recorded in the embodiments of this specification, and a person of ordinary skill in the art can still derive other

drawings from these accompanying drawings without creative efforts.

[0012] FIG. 1 is a schematic flowchart illustrating a method for evicting a memory block in a cache, according to one or more embodiments of this specification;

[0013] FIG. 2 is a schematic structural diagram illustrating a storage system, according to one
5 or more embodiments of this specification;

[0014] FIG. 3 is a schematic structural diagram illustrating an apparatus for evicting a memory block in a cache, according to one or more embodiments of this specification; and

[0015] FIG. 4 is a schematic structural diagram illustrating an electronic device, according to one or more embodiments of this specification.

10

DESCRIPTION OF EMBODIMENTS

[0016] To make a person skilled in the art better understand the technical solutions in this specification, the following clearly and comprehensively describes the technical solutions in the embodiments of this specification with reference to the accompanying drawings in the embodiments of this specification. Clearly, the described embodiments are merely some but not
15 all of the embodiments of this specification. All other embodiments obtained by a person of ordinary skill in the art based on the embodiments of this specification without creative efforts shall fall within the protection scope of this specification.

[0017] As described above, currently, all storage systems release cache space based on a specific policy. Here, a storage unit of a cache is referred to as a memory block. Usually, value of
20 the memory block is evaluated based on an access temperature of the memory block, and a low-value memory block is evicted when cache space needs to be released. However, in many storage systems, a memory block does not merely store one element (the element is cache data). Therefore, there is a possibility that a small part of elements in one memory block is frequently accessed, and the other part of the elements is rarely accessed. Consequently, the memory block has a cold
25 temperature on the whole, and is finally evicted by the storage system. Clearly, performance and gain of the cache are reduced if a frequently accessed element is cleared from the cache.

[0018] In view of this, this specification aims to provide a solution for evicting a memory block in a cache, so that when the memory block in the cache is to be evicted, a valuable element in a memory block that needs to be evicted is migrated to another memory block, to avoid affecting

performance and gain of the cache.

[0019] FIG. 1 is a flowchart illustrating a method for evicting a memory block in a cache, according to one or more embodiments of this specification. The method shown in FIG. 1 can be performed by the following corresponding hardware. The method includes the following steps.

5 S102: Determine a temperature index value of a memory block in a cache, where the memory block is a storage unit of the cache.

[0020] It should be understood that a method of calculating the temperature index value of the memory block is not unique, and is not specifically limited in this specification.

[0021] For example, the temperature index value of the memory block in the one or more
10 embodiments of this specification can be determined based on a quantity of access times of an element in the memory block.

[0022] For example, the element is used as a granularity. A temperature index value of the element is calculated based on the quantity of access times corresponding to the element and an element weight coefficient (the element weight coefficient can reflect importance of the element),
15 and then the temperature index value of the memory block is calculated based on a temperature index value of each element in the memory block.

[0023] In addition, timeliness of a temperature can be further introduced based on the above. For example, the temperature index value of the element is calculated based on the quantity of access times corresponding to the element, a previous-period temperature index value, a
20 temperature temporal attenuation coefficient, and the element weight coefficient, and then the temperature index value of the memory block is calculated based on the temperature index value of each element in the memory block. Clearly, when a current-period temperature index value of the element is calculated based on the previous-period temperature index value of the element and the temperature temporal attenuation coefficient, it can be ensured that a temperature of the
25 element does not lose impact in a short period of time. However, the impact is weakened over time. This reflects timeliness of the temperature to a certain extent.

[0024] S104: Determine, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache.

[0025] Specifically, in the method in the one or more embodiments of the specification, a to-
30 be-evicted memory block to be cleared can be determined when the cache has clearing needs. For example, the to-be-evicted memory block can start to be selected when occupation of storage space

of the cache reaches an occupation upper limit, when a quantity of memory blocks in the cache reaches a quantity upper limit, and when a memory block that is not accessed for predetermined time exists in the cache.

5 [0026] It should be understood that, in this step, a relatively lower-value memory block is selected as a to-be-evicted memory block. In other words, a temperature index value of the to-be-evicted memory block is lower than that of another memory block that needs to be retained in the cache.

10 [0027] For example, a predetermined quantity of memory blocks can be selected, in ascending order of temperature index values, as a target memory block that needs to be cleared; or a memory block whose temperature index value does not reach a memory block temperature threshold in the cache can be determined as a target memory block that needs to be cleared. Because a policy of selecting the to-be-evicted memory block is not unique, examples of details are omitted here.

15 [0028] S106: Migrate a target element in the to-be-evicted memory block to a target memory block in the cache, where the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element.

[0029] It should be understood that the memory block retained in the cache is another memory block other than the to-be-evicted memory block in the cache. The memory block retained in this step may be retained or evicted when a memory block is to be evicted next time.

20 [0030] In actual applications, in the one or more embodiments of this specification, the target element in the to-be-evicted memory block can be flexibly selected based on occupation of space of the cache. That is, the target element in the to-be-evicted memory block is not unique, and can be all elements in the to-be-evicted memory block, or can be some elements whose temperature index values reach an element temperature threshold (an element whose temperature index value reaches the element temperature threshold can be considered as an element that satisfies a value
25 need) in the to-be-evicted memory block.

[0031] For example, if a memory block that can store all the elements in the to-be-evicted memory block is retained in the cache, the target element is all the elements in the to-be-evicted memory block. In other words, when there is a capability of migrating all the elements in the to-be-evicted memory block, all the elements in the to-be-evicted memory block can be migrated, to
30 clear the to-be-evicted memory block when it is ensured that all elements in the cache are not lost.

[0032] Certainly, if no memory block that can store all the elements in the to-be-evicted

memory block is retained in the cache, the target element is the element whose temperature index value reaches the element temperature threshold in the to-be-evicted memory block. In other words, the element worth migrating in the to-be-evicted memory block is selected as the target element based on the element temperature threshold.

5 **[0033]** In a migration process, if the target element in the to-be-evicted memory block is previously selected based on the element temperature threshold, but still no another memory block that can store the target element is retained in the cache, migration can be implemented in the following method:

10 **[0034]** An element in another memory block retained in the cache is deleted, to clear storage space used to store the target element, and the another memory block whose storage space is cleared is determined as the target memory block. A temperature index value of the deleted element in the another memory block is less than the temperature index value of the target element.

15 **[0035]** For ease of understanding, it is assumed here that a memory block 1 is a to-be-evicted memory block, and the memory block 1 has three high-value elements: an element A, an element B, and an element C. Before the memory block 1 is evicted, the element A, the element B, and the element C in the memory block 1 need to be migrated. Here, assuming that all other memory blocks that do not need to be evicted have no space for storing the element A, the element B, and the element C, the memory blocks that do not need to be evicted in the cache need to be traversed, to identify a target memory block that stores at least three elements whose temperature index values are lower than those of the element A, the element B, and the element C, and the three elements whose temperature index values in the target memory block are lower than those of the element A, the element B, and the element C are cleared, to spare space for storing the three high-value elements A, B, and C in the memory block 1.

20 **[0036]** Certainly, in addition to the above-mentioned method, a plurality of target memory blocks can be selected. Here, each target memory block stores at least another element whose temperature index value is lower than those of the elements A, B, and C, and three elements whose temperature index values are lower than those of the elements A, B, and C are cleared from the plurality of target memory blocks, to spare space for storing the high-value elements A, B, and C in the memory block 1. For example, memory blocks 2, 3, and 4 that do not need to be evicted in the cache are selected as target memory blocks, and one element whose temperature index value is lower than those of the elements A, B, and C is selected from each of the memory blocks 2, 3,

and 4 for clearing, so that the memory block 2 stores the element A in the memory block 1, the memory block 3 stores the element B in the memory block 1, and the memory block 4 stores the element C in the memory block 1.

[0037] Clearly, based on the above-mentioned design, if all memory blocks that do not need to be evicted in the cache have no space for storing the target element that needs to be migrated in the to-be-evicted memory block, an element can be selected based on a temperature index value, to ensure a maximum gain of the cache.

[0038] S108: Delete the to-be-evicted memory block from the cache.

[0039] In the method in the one or more embodiments of this specification, the temperature index value of the memory block in the cache can be updated, and when space of the cache needs to be cleared, the to-be-evicted memory block that needs to be cleared can be determined based on the updated temperature index value of the memory block. Before clearing, a valuable element in the to-be-evicted memory block is migrated to another memory block, to avoid a case in which a high-value element in the cache is lost due to clearing of the memory block, and consequently, performance and gain are reduced.

[0040] The following describes the method in the one or more embodiments of this specification in detail with reference to an actual application scenario.

[0041] As shown in FIG. 2, this application scenario provides a storage system of an LSM tree architecture, including a memory table (MemTable) storage structure and a sorted string table (SSTable) storage structure. A MemTable is configured to store cache data of the storage system. All data in the storage system is updated in a memory table (MemTable). The background persistently stores, in the SSTable through dump, a cache element in the MemTable. A compression tool (compactor) in the SSTable is responsible for collecting and compressing data.

[0042] To help the compactor perform data compression and clearing, and to facilitate management of a plurality of types of caches, the cache element in the MemTable needs to have a variable length in the storage system in this application scenario. To manage the cache element with a variable length, a memory block (MemBlock) with a fixed length is set in the MemTable as a minimum storage unit. Although there are many different types of cache elements with a variable length in the storage system, a unified cache framework is designed. Different types of caches of different tenants are managed by the framework together, and different types of cache elements are distinguished based on different priorities.

[0043] Further, the storage system calculates a score of an access temperature of each element in the MemBlock, and a more frequently accessed element has a higher score. In addition, for each MemBlock, a total score is determined based on scores of all elements in the MemBlock.

[0044] In the one or more embodiments of this specification, a cache is evicted in a unit of a
5 MemBlock that is a minimum unit of managing memory. A cache clearing thread in the background of the storage system periodically calculates scores of all the elements in the MemTable and scores of all the MemBlocks, sorts all the current MemBlocks based on the scores, and evicts a MemBlock with a low score.

[0045] In specific applications, a user writes data (a to-be-cached element) through IO.

10 [0046] The storage system allocates a memory to the to-be-cached element: Because the to-be-cached element has a variable length, an actual needed memory size is first calculated. If the to-be-cached element can be accommodated by a current MemBlock, the to-be-cached element is allocated to the current MemBlock; or if the to-be-cached element cannot be accommodated by a current MemBlock, a new MemBlock is allocated for storage.

15 [0047] In a process of allocating the new MemBlock, if the MemTable does not have enough memory or a quantity of MemBlocks reaches an upper limit, a MemBlock eviction process is executed: All MemBlocks in the cache are traversed, and a score of a temperature of the current MemBlock is updated. A range of a MemBlock that needs to be evicted in a current cache eviction process is calculated based on information about a specified upper/lower limit of the MemBlock.

20 All the MemBlocks in the cache are traversed, the score of the temperature of the MemBlock is sorted, and a MemBlock with a lowest temperature is selected for eviction.

[0048] In the eviction process, the background of the storage system adjusts an element with a high temperature in the to-be-evicted MemBlock to another MemBlock that has a high temperature and that does not need to be evicted, to improve a hit rate of the cache, thereby
25 reflecting higher performance and a higher gain.

[0049] It can be seen from the above-mentioned content that, to facilitate management of a plurality of types of caches, in the storage system in this application scenario, the cache element with a variable length is designed, and the MemBlock with a fixed length is used for storage, to implement cache clearing at a granularity of MemBlock. In addition, the storage system calculates
30 a score of a temperature of each cache element. When a certain MemBlock is evicted, a cache element with a high temperature score can be migrated to another MemBlock, to avoid losing an

element with a high temperature when space of the cache is cleared.

[0050] The above-mentioned application scenario is an example description of the method in the embodiments of this invention. It should be understood that the storage system in the application scenario can perform steps in the eviction method shown in FIG. 1, including:
5 determining, based on the temperature index value of the MemBlock in the MemTable, a to-be-evicted MemBlock that needs to be cleared in the MemTable; migrating a target element in the to-be-evicted MemBlock to a target MemBlock in the MemTable, where the target MemBlock is a MemBlock that is retained in the MemTable and whose available storage space is greater than storage space needed by the target element; and deleting the to-be-evicted MemBlock from the
10 MemTable.

[0051] In addition, corresponding to the method for evicting a memory block in a cache shown in FIG. 1, one or more embodiments of this invention further provide an apparatus for evicting a memory block in a cache. FIG. 3 is a schematic structural diagram illustrating an eviction apparatus 300, according to one or more embodiments of this invention. The eviction apparatus 300 includes:
15 a temperature determining module 310, configured to determine a temperature index value of a memory block in a cache, where the memory block is a storage unit of the cache; an eviction analysis module 320, configured to determine, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache;
an element migration module 330, configured to migrate a target element in the to-be-evicted
20 memory block to a target memory block in the cache, where the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and an eviction execution module 340, configured to delete the to-be-evicted memory block from the cache.

[0052] The apparatus in the one or more embodiments of this specification can update the
25 temperature index value of the memory block in the cache, and when space of the cache needs to be cleared, determine the to-be-evicted memory block that needs to be cleared based on the updated temperature index value of the memory block. Before clearing, a valuable element in the to-be-evicted memory block is migrated to another memory block, to avoid a case in which a high-value element is lost in the cache due to clearing of the memory block, and consequently,
30 performance and gain are reduced.

[0053] Optionally, the target element is all elements in the to-be-evicted memory block, or the

target element is an element whose temperature index value reaches an element temperature threshold in the to-be-evicted memory block.

[0054] If a memory block that can store all the elements in the to-be-evicted memory block is retained in the cache, the target element is all the elements in the to-be-evicted memory block. In other words, when there is a capability of migrating all the elements in the to-be-evicted memory block, all the elements in the to-be-evicted memory block can be migrated, to clear the to-be-evicted memory block when it is ensured that no element is lost.

[0055] If no memory block that can store all the elements in the to-be-evicted memory block is retained in the cache, the target element is the element whose temperature index value reaches the element temperature threshold in the to-be-evicted memory block. In other words, the element worth migrating in the to-be-evicted memory block is selected as the target element based on the element temperature threshold.

[0056] Optionally, before migrating the target element in the to-be-evicted memory block to the target memory block in the cache, the element migration module 330 is further configured to select, from the cache as the target memory block, another memory block that is retained and that can store the target element. If no another memory block that can store the target element is retained in the cache, the element migration module 330 can delete an element in another memory block retained in the cache, to clear storage space used to store the target element, and determine, as the target memory block, the another memory block whose storage space is cleared. A temperature index value of the deleted element in the another memory block is less than a temperature index value of the target element.

[0057] Based on a deletion method of the element migration module 330, if all memory blocks that do not need to be evicted in the cache have no space for storing the target element that needs to be migrated in the to-be-evicted memory block, an element can be selected based on a temperature index value. If an element whose temperature index value is lower than that of the target element exists in the memory block that does not need to be evicted, the element can be deleted, to spare storage space for the target element.

[0058] Optionally, an execution condition in which the eviction analysis module 320 determines, based on the temperature index value of the memory block in the cache, a target memory block that needs to be cleared in the cache includes at least one of the following: occupation of storage space of the cache reaches an occupation upper limit; a quantity of memory

blocks in the cache reaches a quantity upper limit; and a memory block that is not accessed for predetermined time exists in the cache.

[0059] Optionally, a temperature index value of an element in the memory block is calculated based on a quantity of access times corresponding to the element, a previous-period temperature index value, and a temperature temporal attenuation coefficient. The temperature index value of the memory block is determined based on temperature index values of all elements in the memory block.

[0060] Optionally, the eviction analysis module 320 can select, in ascending order of temperature index values, a predetermined quantity of memory blocks as the target memory block that needs to be cleared; or can determine that a memory block whose temperature index value does not reach a memory block temperature threshold in the cache is the target memory block that needs to be cleared.

[0061] Optionally, the cache is a MemTable of an LSM tree, the MemTable is an element in cache with a variable length, and space occupied by data of the element is not greater than storage space provided by the memory block.

[0062] Clearly, the eviction apparatus in the one or more embodiments of this specification can be used as an execution body of the eviction method shown in FIG. 1, so that a function implemented in the eviction method in FIG. 1 can be implemented. Because principles are the same, details are omitted in this specification.

[0063] FIG. 4 is a schematic structural diagram illustrating an electronic device, according to one or more embodiments of this specification. Referring to FIG. 4, at a hardware level, the electronic device includes a processor, and optionally, further includes an internal bus, a network interface, and a storage. The storage may include a memory, for example, a random access memory (RAM), and may further include a non-volatile memory, for example, at least one disk memory. Certainly, the electronic device may further include hardware needed by another service.

[0064] The processor, the network interface, and the storage can be connected to each other through an internal bus. The internal bus can be an industry standard architecture (ISA) bus, a peripheral component interconnect (PCI) bus, an extended industry standard architecture (EISA) bus, etc. The buses can be classified into an address bus, a data bus, a control bus, etc. For ease of representation, in FIG. 4, only one bidirectional arrow is used to represent the bus, but it does not indicate that there is only one bus or only one type of bus.

[0065] The storage is configured to store a program. Specifically, the program can include program code, and the program code includes computer operation instructions. The storage can include a memory and a non-volatile memory, and provide instructions and data for the processor.

[0066] The processor reads a corresponding computer program from the non-volatile memory, and runs the computer program in the memory, to implement, at a logical level, an apparatus for evicting a memory block in a cache. The eviction apparatus can be a storage system, or can be a component of the storage system. The processor executes the program stored in the storage, and is specifically configured to perform the following operations: determining a temperature index value of the memory block in the cache, where the memory block is a storage unit of the cache; determining, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache; migrating a target element in the to-be-evicted memory block to a target memory block in the cache, where the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and deleting the to-be-evicted memory block from the cache.

[0067] The electronic device in the one or more embodiments of this specification can update the temperature index value of the memory block in the cache, and when cache space needs to be cleared, determine, based on the updated temperature index value of the memory block, a to-be-evicted memory block that needs to be cleared. Before clearing, a valuable element in the to-be-evicted memory block is migrated to another memory block, to avoid a case in which a high-value element in the cache is lost due to clearing of the memory block, and consequently, performance and gain are reduced.

[0068] The eviction method disclosed in the embodiment shown in FIG. 1 in this specification can be applied to the processor, or implemented by the processor. The processor may be an integrated circuit chip, and has a signal processing capability. In an implementation process, the steps of the method can be completed by using an integrated logic circuit of hardware of the processor or instructions in the form of software. The processor can be a general-purpose processor, including a central processing unit (CPU), a network processor (NP), etc., or can be a digital signal processor (DSP), an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA) or another programmable logic device, a discrete gate or transistor logic device, or a discrete hardware component. The methods, steps, and logical block diagrams disclosed in the

embodiments of this specification can be implemented or executed. The general-purpose processor can be a microprocessor, or the processor can be any conventional processor, etc. The steps of the method disclosed with reference to the embodiments of the specification can be directly completed by a hardware decoding processor, or can be completed by a combination of hardware and software modules in the decoding processor. The software module can be located in a mature storage medium in the art, for example, a random access memory, a flash memory, a read-only memory, a programmable read-only memory, an electrically erasable programmable memory, or a register. The storage medium is located in the storage, and the processor reads information in the storage and completes steps of the method in combination with hardware of the processor.

10 [0069] It should be understood that, the electronic device in the embodiments of this specification can implement functions of the consensus method in the embodiments shown in FIG. 1 and FIG. 2. Because principles are the same, details are omitted in this specification.

[0070] Certainly, in addition to software implementations, the electronic device in this specification does not preclude other implementations, for example, a logic device or a combination of software and hardware. In other words, an execution body of the following processing procedure is not limited to each logical unit, and can be hardware or a logic device.

15 [0071] In addition, one or more embodiments of this specification further provide a computer-readable storage medium. The computer-readable storage medium stores one or more programs. The one or more programs include instructions. When the instructions are executed by a portable electronic device including a plurality of applications, the portable electronic device can perform the method in the embodiment shown in FIG. 1, and is specifically configured to perform the following steps: determining a temperature index value of a memory block in a cache, where the memory block is a storage unit of the cache; determining, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache; 20 migrating a target element in the to-be-evicted memory block to a target memory block in the cache, where the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and deleting the to-be-evicted memory block from the cache.

[0072] A person skilled in the art should understand that the embodiments of this specification can be provided as a method, a system, or a computer program product. Therefore, this specification can use a form of hardware only embodiments, software only embodiments, or

embodiments with a combination of software and hardware. Moreover, this specification can use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk memory, a CD-ROM, an optical memory, etc.) that include computer-usable program code.

5 **[0073]** Some specific embodiments of this specification are described above. Other embodiments fall within the scope of the appended claims. In some situations, the actions or steps described in the claims can be performed in an order different from the order in the embodiments and the desired results can still be achieved. In addition, the process depicted in the accompanying drawings does not necessarily need a particular execution order to achieve the desired results. In
10 some implementations, multi-tasking and concurrent processing is feasible or may be advantageous.

[0074] The above-mentioned descriptions are merely some embodiments of this specification and are not intended to limit this specification. A person skilled in the art can make various changes and variations to this specification. Any modification, equivalent replacement, or improvement
15 made without departing from the spirit and principle of this specification shall fall within the scope of the claims in this specification. In addition, all other embodiments obtained by a person of ordinary skill in the art based on one or more embodiments of this specification without creative efforts shall fall within the protection scope of this specification.

CLAIMS

What is claimed is:

1. A method for evicting a memory block in a cache, comprising:
determining a temperature index value of the memory block in the cache, wherein the memory
5 block is a storage unit of the cache;
determining, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache;
migrating a target element in the to-be-evicted memory block to a target memory block in the
cache, wherein the target memory block is a memory block that is retained in the cache and whose
10 available storage space is greater than storage space needed by the target element; and
deleting the to-be-evicted memory block from the cache.
2. The method according to claim 1, wherein
the target element is all elements in the to-be-evicted memory block, or the target element is
an element whose temperature index value reaches an element temperature threshold in the to-be-
15 evicted memory block.
3. The method according to claim 2, wherein
if a memory block that can store all the elements in the to-be-evicted memory block is retained
in the cache, the target element is all the elements in the to-be-evicted memory block.
4. The method according to claim 2, wherein
20 if no memory block that can store all the elements in the to-be-evicted memory block is
retained in the cache, the target element is the element whose temperature index value reaches the
element temperature threshold in the to-be-evicted memory block.
5. The method according to claim 1, wherein
before the migrating a target element in the to-be-evicted memory block to a target memory
25 block in the cache, the method further comprises:
selecting, from the cache as the target memory block, another memory block that is retained
and that can store the target element.
6. The method according to claim 5, wherein
if no another memory block that can store the target element is retained in the cache, the
30 selecting, from the cache as the target memory block, another memory block that is retained and

that can store the target element further comprises:

deleting an element in another memory block retained in the cache, to clear storage space used to store the target element, and determining, as the target memory block, the another memory block whose storage space is cleared, wherein a temperature index value of the deleted element in the another memory block is less than a temperature index value of the target element.

7. The method according to claim 1, wherein

an execution condition of determining, based on the temperature index value of the memory block in the cache, a target memory block that needs to be cleared in the cache comprises at least one of the following:

10 occupation of storage space of the cache reaches an occupation upper limit;
a quantity of memory blocks in the cache reaches a quantity upper limit; and
a memory block that is not accessed for predetermined time exists in the cache.

8. The method according to claim 1, wherein

15 a temperature index value of an element in the memory block is calculated based on a quantity of access times corresponding to the element.

9. The method according to claim 8, wherein

the temperature index value of the element in the memory block is alternatively calculated based on a previous-period temperature index value, a temperature temporal attenuation coefficient, and/or an element weight coefficient that correspond/corresponds to the element.

20 10. The method according to claim 8 or 9, wherein

the temperature index value of the memory block is determined based on temperature index values of all elements in the memory block.

11. The method according to claim 1, wherein

25 the determining, based on the temperature index value of the memory block in the cache, a target memory block that needs to be cleared in the cache comprises:

selecting, in ascending order of temperature index values, a predetermined quantity of memory blocks as the target memory block that needs to be cleared; or

determining that a memory block whose temperature index value does not reach a memory block temperature threshold in the cache is the target memory block that needs to be cleared.

30 12. A method for evicting a memory block in a cache, applied to a storage system of a log structured merge (LSM) tree storage architecture, and comprising:

determining a temperature index value of a memory block (MemBlock) in a memory table (MemTable) of an LSM tree of the storage system, wherein the MemTable is configured to store cache data of the storage system;

5 determining, based on the temperature index value of the MemBlock in the MemTable, a to-be-evicted MemBlock that needs to be cleared in the MemTable;

migrating a target element in the to-be-evicted MemBlock to a target MemBlock in the MemTable, wherein the target MemBlock is a MemBlock that is retained in the MemTable and whose available storage space is greater than storage space needed by the target element; and

deleting the to-be-evicted MemBlock from the MemTable.

10 13. An apparatus for evicting a memory block in a cache, comprising:

a temperature determining module, configured to determine a temperature index value of the memory block in the cache, wherein the memory block is a storage unit of the cache;

an eviction analysis module, configured to determine, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the
15 cache;

an element migration module, configured to migrate a target element in the to-be-evicted memory block to a target memory block in the cache, wherein the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and

20 an eviction execution module, configured to delete the to-be-evicted memory block from the cache.

14. An electronic device, comprising a storage, a processor, and a computer program that is stored in the storage and that can run on the processor, wherein the computer program is executed by the processor to implement the following steps:

25 determining a temperature index value of a memory block in a cache, wherein the memory block is a storage unit of the cache;

determining, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache;

30 migrating a target element in the to-be-evicted memory block to a target memory block in the cache, wherein the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and

deleting the to-be-evicted memory block from the cache.

15. A computer-readable storage medium, wherein a computer program is stored in the computer-readable storage medium, and when the computer program is executed by a processor, the following steps are implemented:

5 determining a temperature index value of a memory block in a cache, wherein the memory block is a storage unit of the cache;

 determining, based on the temperature index value of the memory block in the cache, a to-be-evicted memory block that needs to be cleared in the cache;

 migrating a target element in the to-be-evicted memory block to a target memory block in the
10 cache, wherein the target memory block is a memory block that is retained in the cache and whose available storage space is greater than storage space needed by the target element; and

 deleting the to-be-evicted memory block from the cache.