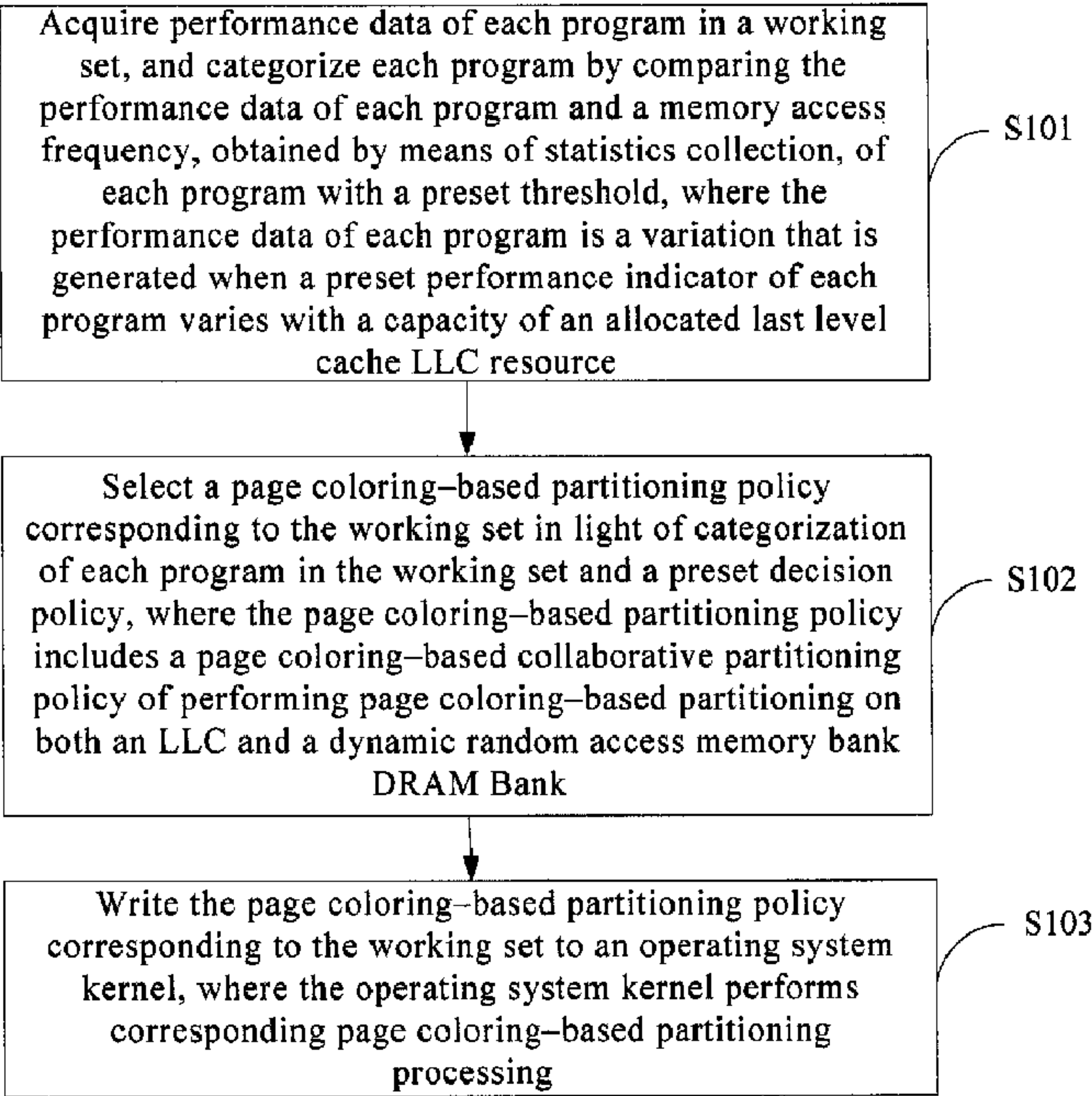




(86) Date de dépôt PCT/PCT Filing Date: 2014/10/22	(51) Cl.Int./Int.Cl. <i>G06F 12/08</i> (2016.01)
(87) Date publication PCT/PCT Publication Date: 2015/04/30	(72) Inventeurs/Inventors: LIU, LEI, CN; WU, CHENGYONG, CN; FENG, XIAOBING, CN
(45) Date de délivrance/Issue Date: 2018/02/13	(73) Propriétaire/Owner: HUAWEI TECHNOLOGIES CO., LTD, CN
(85) Entrée phase nationale/National Entry: 2016/04/13	(74) Agent: GOWLING WLG (CANADA) LLP
(86) N° demande PCT/PCT Application No.: CN 2014/089194	
(87) N° publication PCT/PCT Publication No.: 2015/058695	
(30) Priorité/Priority: 2013/10/23 (CN201310503238.1)	

(54) Titre : PROCEDE ET APPAREIL D'OPTIMISATION DE RESSOURCES DE MEMOIRE
(54) Title: MEMORY RESOURCE OPTIMIZATION METHOD AND APPARATUS



(57) **Abrégé/Abstract:**
Embodiments of the present invention relate to the field of computer. Provided are a memory resource optimization method and an apparatus, solving a mutual impact problem between existing multi-level memory resources and optimizing an existing unitary

(57) **Abrégé(suite)/Abstract(continued):**

division mechanism. A specific scheme is: obtaining performance data of each program in a working set through a page coloring technology, obtaining a category of the each program by incorporating memory access frequency, selecting a page coloring division policy corresponding to the working set based on the category of the each program, inputting the page coloring division policy into an operating system kernel, and completing corresponding coloring division processing. The present invention is used for incorporating features of the working set to eliminate or reduce mutual interference of processes and threads on memory resources, improving overall performance of a computer.

ABSTRACT

Embodiments of the present invention provide a memory resource optimization method and apparatus, relate to the computer field, solve a problem that existing multi-level memory resources affect each other, and optimize an existing single partitioning mechanism. A
5 specific solution is: obtaining performance data of each program in a working set by using a page coloring technology, obtaining a category of each program in light of a memory access frequency, selecting, according to the category of each program, a page coloring-based partitioning policy corresponding to the working set, and writing the page coloring-based partitioning policy to an operating system kernel, to complete corresponding coloring-based
10 partitioning processing. The present invention is used to eliminate or reduce mutual interference of processes or threads on a storage resource in light of a feature of the working set, thereby improving overall performance of a computer.

MEMORY RESOURCE OPTIMIZATION METHOD AND APPARATUS

TECHNICAL FIELD

[0001] The present invention relates to the computer field, and in particular, to a memory
5 resource optimization method and apparatus.

BACKGROUND

[0002] Currently, a multi-core high performance computer (Multi-core High Performance
Computer) is applied more widely. Moreover, as a quantity of calculating units (cores) on a
processor continues to increase, a memory access contention phenomenon among multiple
10 cores further complicates a problem. In any time period of concurrent execution, memory
access requests from different cores may contend for memory access to global storage
resources, further triggering a conflict over resources such as a memory controller (Memory
Controller, MC), bandwidth (Bandwidth), and a dynamic random access memory bank
(DRAM Bank); as a result, resource utilization is affected.

15 [0003] Access to a main memory is used as an example. Usually, a single transistor is
used as a storage unit, $N \times M$ storage units form a storage matrix, and several storage matrices
constitute a bank (Bank). Each Bank has a row-buffer (Row-Buffer); when data is to be
accessed, data in a target row can be read only after the data is moved to the row-buffer. A
modern DRAM system usually uses multiple Banks, and respective memory access requests
20 are processed independently at the same time. However, if two memory access requests that
are from different processes or threads access different rows of a same DRAM Bank, a
conflict (or referred to as a row-buffer conflict) on the DRAM Bank may be generated, and a
memory access delay is increased. A cache (Cache) is used to alleviate a gap between a
calculating unit and a main memory. Because the cache is closer to the calculating unit than

the main memory, the cache affects calculation performance more easily. Existing processors basically use a structure of sharing a last level cache (Last Level Cache, LLC) among multiple cores. However, the LLC is also intensely contended for among multiple concurrent programs or threads. If no proper management policy is used, it is very easy to cause serious performance deterioration.

5 [0004] In the prior art, the LLC or the DRAM Bank is partitioned by using page coloring (Page-Coloring). Regarding page coloring-based LLC partitioning, a Cache resource is partitioned into several independent parts in a manner of performing page coloring on index bits of a cache set (Cache Set) in a physical address, and the parts are assigned to different threads separately. Therefore, inter-thread contention due to sharing of a Cache disappears. Similar to the LLC partitioning, index bits of the DRAM Bank can also be reflected in a physical address, and the DRAM Bank can also be partitioned into several independent groups by performing coloring according to these address bits; therefore, inter-program contention on the DRAM Bank disappears. For some working sets, a relatively good performance improvement effect can be achieved.

15 [0005] However, when a conventional page coloring technology is used to partition a resource at a level, a negative impact is caused on utilization of a resource at another level. For example, work related to the LLC partitioning has a restriction on performance improvement of a DRAM Bank resource; similarly, work related to the DRAM Bank partitioning also affects performance of an LLC resource. In addition, because running of a modern computer system requires various working sets, an existing page coloring-based partitioning mechanism for a resource at a level is difficult to match up with features of different working sets; as a result, an optimum partitioning effect cannot be achieved, and overall performance improvement of the computer is limited.

25

SUMMARY

[0006] Embodiments of the present invention provide a memory resource optimization method and apparatus, provide a collaborative partitioning policy between an LLC and a DRAM Bank, where the policy is combined with a feature of a working set, and can solve a

problem that processes or threads mutually affect each other on storage resources, so that overall performance of a computer is improved.

[0007] To achieve the foregoing objectives, the following technical solutions are used in the embodiment of the present invention:

5 [0008] According to a first aspect, a memory resource optimization method is provided, where the method includes:

acquiring performance data of each program in a working set, and categorizing each program by comparing the performance data of each program and a memory access frequency, obtained by means of statistics collection, of each program with a preset threshold,
10 where the performance data of each program is a variation that is generated when a preset performance indicator of each program varies with a capacity of an allocated last level cache LLC resource;

selecting, in light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set,
15 where the page coloring-based partitioning policy includes a page coloring-based collaborative partitioning policy for performing page coloring-based partitioning on both the LLC and the dynamic random access memory bank DRAM Bank; and

writing the page coloring-based partitioning policy corresponding to the working set to an operating system kernel, where the operating system kernel performs corresponding
20 page coloring-based partitioning processing.

[0009] With reference to the first aspect, in a first implementation manner, the acquiring performance data of each program in a working set includes:

partitioning the LLC resource into N portions by using a page coloring technology, taking $1/N$ of a maximum capacity of the LLC resource as one level, allocating the maximum
25 capacity of the LLC resource to each program at the beginning, and decreasing the capacity of the LLC resource allocated to each program by one level in each adjustment, until the capacity is decreased to $1/N$ of the maximum capacity of the LLC resource; and

monitoring a variation that is generated when the preset performance indicator of each program varies with the capacity of the allocated LLC resource in an adjustment process,
30 and using the variation as the performance data of each program, where the preset

performance indicator is a speed-up ratio of each program.

[0010] With reference to the first aspect or the first possible implementation manner of the first aspect, in a second possible implementation manner, the categorizing each program by comparing the performance data of each program and a memory access frequency, obtained by means of statistics collection, of each program with a preset threshold includes:

counting a quantity of times that each program accesses a main memory in a preset stage of a running process, to obtain the memory access frequency of each program;

comparing the performance data of each program and the memory access frequency, obtained by means of statistics collection, of each program with the preset threshold, where the preset threshold includes a first threshold, a second threshold, and a third threshold, the first threshold and the second threshold are performance data thresholds, and the third threshold is a memory access frequency threshold; and

if performance data of a program is greater than the first threshold, categorizing the program as a high demand type;

if performance data of a program is less than the first threshold and greater than the second threshold, categorizing the program as a medium demand type; or

if performance data of a program is less than the second threshold and a memory access frequency is greater than the third threshold, categorizing the program as a low demand and intensive type.

[0011] With reference to the first aspect, in a third possible implementation manner, the selecting, in light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set includes:

the preset decision policy is a partitioning policy decision tree in the operating system kernel, and the partitioning policy decision tree is implemented in the operating system kernel in algorithm form; and

inputting the categorization of each program in the working set to the operating system kernel, and searching the partitioning policy decision tree in the operating system kernel for a corresponding node in light of a category of each program in the working set, to determine the page coloring-based partitioning policy corresponding to the working set.

[0012] With reference to the third possible implementation manner of the first aspect, in a

fourth possible implementation manner, the page coloring-based partitioning policy includes the page coloring-based collaborative partitioning policy and a page coloring-based non-collaborative partitioning policy;

the page coloring-based collaborative partitioning policy is a partitioning policy of
 5 using overlapped index address bits O-bits as page coloring-based partitioning index bits, and the O-bits are overlapped address bits of index bits of the LLC and index bits of the DRAM Bank in a physical page frame, and are used to index page coloring-based partitioning for both the LLC and the DRAM Bank; and

the page coloring-based collaborative partitioning policy includes:

10 a category A multi-level memory collaborative partitioning policy A-MMCP, in which the O-bits are used as partitioning index bits, and the LLC and the DRAM Bank are partitioned into a same quantity of equal portions;

a category B multi-level memory collaborative partitioning policy B-MMCP, in which the O-bits and an index bit of the DRAM Bank are used as partitioning index bits, the
 15 LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is greater than a quantity of partitioned equal portions of the LLC; and

a category C multi-level memory collaborative partitioning policy C-MMCP, in which the O-bits and an index bit of the LLC are used as partitioning index bits, the LLC and
 20 the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is less than a quantity of partitioned equal portions of the LLC; and

the page coloring-based collaborative partitioning policy is a partitioning policy in which the O-bits are not used and includes:

25 a Cache-Only policy, in which coloring-based partitioning is performed on the LLC by using the index bit of the LLC, and coloring-based partitioning is not performed on the DRAM Bank; and

a Bank-Only policy, in which coloring-based partitioning is performed on the DRAM Bank by using the index bit of the DRAM Bank, and coloring-based partitioning is
 30 not performed on the LLC.

[0013] With reference to the third possible implementation manner of the first aspect or the fourth possible implementation manner of the first aspect, in a fifth possible implementation manner, the searching the partitioning policy decision tree in the operating system kernel for a corresponding node in light of a category of each program in the working set, to determine the page coloring-based partitioning policy corresponding to the working set includes:

if each category to which each program in the working set belongs is the high demand type, selecting the Bank-Only policy;

if there is the low demand and intensive type in a category to which each program in the working set belongs, further determining a quantity of programs in the working set; and if the quantity of programs is less than or equal to N, selecting the A-MMCP; otherwise, selecting the C-MMCP, where N is a quantity of cores of a processor; or

if there is the medium demand type and there is no low demand and intensive type in a category to which each program in the working set belongs, further determining a quantity of programs in the working set; and if the quantity of programs is less than or equal to N, selecting the A-MMCP; otherwise, selecting the B-MMCP, where N is a quantity of cores of a processor.

[0014] According to a second aspect, a memory resource optimization apparatus is provided, where the apparatus includes:

a front end unit, configured to acquire performance data of each program in a working set, and categorize each program by comparing the performance data of each program and a memory access frequency, obtained by means of statistics collection, of each program with a preset threshold, where the performance data of each program is a variation that is generated when a preset performance indicator of each program varies with a capacity of an allocated last level cache LLC resource;

a decision-making unit, configured to select, in light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set, where the page coloring-based partitioning policy includes a page coloring-based collaborative partitioning policy for performing page coloring-based partitioning on both the LLC and the dynamic random access memory bank

DRAM Bank; and

a partitioning unit, configured to write the page coloring-based partitioning policy corresponding to the working set to an operating system kernel, where the operating system kernel performs corresponding page coloring-based partitioning processing.

5 [0015] With reference to the second aspect, in a first implementation manner, the front end unit includes a data collecting unit, where the data collecting unit is specifically configured to:

partition the LLC resource into N portions by using a page coloring technology, take 1/N of a maximum capacity of the LLC resource as one level, allocate the maximum
10 capacity of the LLC resource to each program at the beginning, and decrease the capacity of the LLC resource allocated to each program by one level in each adjustment, until the capacity is decreased to 1/N of the maximum capacity of the LLC resource; and

monitor a variation that is generated when the preset performance indicator of each program varies with the capacity of the allocated LLC resource in an adjustment process, and
15 use the variation as the performance data of each program, where the preset performance indicator is a speed-up ratio of each program.

[0016] With reference to the second aspect or the first possible implementation manner of the second aspect, in a second possible implementation manner, the front end unit further includes a categorizing unit, where the categorizing unit is specifically configured to:

20 count a quantity of times that each program accesses a main memory in a preset stage of a running process, to obtain the memory access frequency of each program;

compare the performance data of each program and the memory access frequency, obtained by means of statistics collection, of each program with the preset threshold, where the preset threshold includes a first threshold, a second threshold, and a third threshold, the
25 first threshold and the second threshold are performance data thresholds, and the third threshold is a memory access frequency threshold; and

if performance data of a program is greater than the first threshold, categorize the program as a high demand type;

if performance data of a program is less than the first threshold and greater than
30 the second threshold, categorize the program as a medium demand type; or

if performance data of a program is less than the second threshold and a memory access frequency is greater than the third threshold, categorize the program as a low demand and intensive type.

[0017] With reference to the second aspect, in a third possible implementation manner, the preset decision policy is a partitioning policy decision tree in the operating system kernel, and the partitioning policy decision tree is implemented in the operating system kernel in algorithm form, and the decision-making unit is specifically configured to:

input the categorization of each program in the working set to the operating system kernel, and search the partitioning policy decision tree in the operating system kernel for a corresponding node in light of a category of each program in the working set, to determine the page coloring-based partitioning policy corresponding to the working set.

[0018] With reference to the third possible implementation manner of the second aspect, in a fourth possible implementation manner, the page coloring-based partitioning policy includes the page coloring-based collaborative partitioning policy and a page coloring-based non-collaborative partitioning policy;

the page coloring-based collaborative partitioning policy is a partitioning policy of using overlapped index address bits O-bits as page coloring-based partitioning index bits, and the O-bits are overlapped address bits of index bits of the LLC and index bits of the DRAM Bank in a physical page frame, and are used to index page coloring-based partitioning for both the LLC and the DRAM Bank; and

the page coloring-based collaborative partitioning policy includes:

a category A multi-level memory collaborative partitioning policy A-MMCP, in which the O-bits are used as partitioning index bits, and the LLC and the DRAM Bank are partitioned into a same quantity of equal portions;

a category B multi-level memory collaborative partitioning policy B-MMCP, in which the O-bits and an index bit of the DRAM Bank are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is greater than a quantity of partitioned equal portions of the LLC; and

a category C multi-level memory collaborative partitioning policy C-MMCP, in

which the O-bits and an index bit of the LLC are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is less than a quantity of partitioned equal portions of the LLC; and

5 the page coloring-based collaborative partitioning policy is a partitioning policy in which the O-bits are not used and includes:

a Cache-Only policy, in which coloring-based partitioning is performed on the LLC by using the index bit of the LLC, and coloring-based partitioning is not performed on the DRAM Bank; and

10 a Bank-Only policy, in which coloring-based partitioning is performed on the DRAM Bank by using the index bit of the DRAM Bank, and coloring-based partitioning is not performed on the LLC.

[0019] With reference to the third possible implementation manner of the second aspect or the fourth possible implementation manner of the second aspect, in a fifth possible
15 implementation manner, the decision-making unit is further specifically configured to:

if each category to which each program in the working set belongs is the high demand type, select the Bank-Only policy;

if there is the low demand and intensive type in a category to which each program in the working set belongs, further determine a quantity of programs in the working set; and if
20 the quantity of programs is less than or equal to N, select the A-MMCP; otherwise, select the C-MMCP, where N is a quantity of cores of a processor; or

if there is the medium demand type and there is no low demand and intensive type in a category to which each program in the working set belongs, further determine a quantity of programs in the working set; and if the quantity of programs is less than or equal to N,
25 select the A-MMCP; otherwise, select the B-MMCP, where N is a quantity of cores of a processor.

[0020] According to the memory resource optimization method and apparatus provided in the embodiments of the present invention, an LLC resource is partitioned by using a page coloring technology, performance data of each program in a working set is acquired, a
30 category of each program is obtained in light of a memory access frequency, a page

coloring-based partitioning policy corresponding to the working set is selected according to the category of each program, and the page coloring-based partitioning policy is written to an operating system kernel, to complete corresponding coloring-based partitioning processing. In this way, a collaborative partitioning policy between the LLC and a DRAM Bank is
 5 implemented in light of a feature of the working set, and mutual interference of processes or threads on a storage resource can be reduced and even eliminated, thereby improving overall performance of a computer.

BRIEF DESCRIPTION OF DRAWINGS

[0021] To describe the technical solutions in the embodiments of the present invention or
 10 in the prior art more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments or the prior art. Apparently, the accompanying drawings in the following description show merely some embodiments of the present invention, and a person of ordinary skill in the art may still derive other drawings from these accompanying drawings without creative efforts.

15 [0022] FIG. 1 is a schematic flowchart of a memory resource optimization method according to an embodiment of the present invention;

[0023] FIG. 2 is a schematic flowchart of another memory resource optimization method according to an embodiment of the present invention;

[0024] FIG. 3 is a schematic diagram of a program categorization effect according to an
 20 embodiment of the present invention;

[0025] FIG. 4 is schematic structural diagram 1 of a memory resource optimization apparatus according to an embodiment of the present invention;

[0026] FIG. 5 is schematic structural diagram 2 of a memory resource optimization apparatus according to an embodiment of the present invention; and

25 [0027] FIG. 6 is schematic structural diagram 3 of a memory resource optimization apparatus according to an embodiment of the present invention.

DESCRIPTION OF EMBODIMENTS

[0028] The following clearly and completely describes the technical solutions in the embodiments of the present invention with reference to the accompanying drawings in the embodiments of the present invention. Apparently, the described embodiments are merely
 5 some but not all of the embodiments of the present invention. All other embodiments obtained by a person of ordinary skill in the art based on the embodiments of the present invention without creative efforts shall fall within the protection scope of the present invention.

[0029] An embodiment of the present invention provides a memory resource optimization method. As shown in FIG. 1, the method includes:

10 [0030] S101: Acquire performance data of each program in a working set, and categorize each program by comparing the performance data of each program and a memory access frequency, obtained by means of statistics collection, of each program with a preset threshold, where the performance data of each program is a variation that is generated when a preset performance indicator of each program varies with a capacity of an allocated last level cache
 15 LLC resource.

[0031] S102: Select, in light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set, where the page coloring-based partitioning policy includes a page coloring-based collaborative partitioning policy for performing page coloring-based partitioning on both an
 20 LLC and a dynamic random access memory bank DRAM Bank.

[0032] S103: Write the page coloring-based partitioning policy corresponding to the working set to an operating system kernel, where the operating system kernel performs corresponding page coloring-based partitioning processing.

[0033] According to the memory resource optimization method provided in this
 25 embodiment of the present invention, performance data of each program in a working set is acquired, a category of each program is obtained in light of a memory access frequency, a page coloring-based partitioning policy of the working set is selected according to the category of each program, and the page coloring-based partitioning policy is written to an operating system kernel, to complete corresponding coloring-based partitioning processing. In

this way, a collaborative partitioning policy between an LLC and a DRAM Bank is implemented in light of a feature of the working set, and mutual interference of processes or threads on a storage resource can be reduced and even eliminated, thereby improving overall performance of a computer.

5 [0034] In order to enable a person skilled in the art to understand the technical solutions provided by the embodiments of the present invention more clearly, the following describes another memory resource optimization method provided by an embodiment of the present invention in detail by using a specific example. As shown in FIG. 2, the method includes:

10 [0035] S201: Acquire performance data of each program in a working set.

[0036] Specifically, an LLC resource is partitioned by using a page coloring technology, the LLC resource is partitioned into N portions, and $1/N$ of a maximum capacity of the LLC resource is taken as one level. The maximum capacity of the LLC resource is allocated to each program at the beginning, and the capacity of the LLC resource allocated to each
15 program is decreased by one level in each adjustment, until the LLC resource allocated to each program is decreased to $1/N$ of the maximum capacity of the LLC resource; a variation that is generated when a preset performance indicator of each program varies with the capacity of the allocated LLC resource is monitored in an adjustment process, and the variation is used as the performance data of each program.

20 [0037] Exemplarily, in a quad-core processor, N may be 8, $1/8$ of a maximum capacity of an LLC resource is taken as one level, a capacity of the LLC resource allocated to each program in a working set is adjusted, and the capacity of the LLC resource allocated to each program in the working set is decreased by one level in each adjustment, until the LLC resource allocated to each program is decreased to $1/8$ of the maximum capacity of the LLC
25 resource; a variation that is generated when a preset performance indicator of each program varies with the capacity of the allocated LLC resource in the foregoing adjustment process is monitored.

[0038] The preset performance indicator of each program may be a normalization speed-up ratio, the variation that is generated when the normalization speed-up ratio of each
30 program varies with the capacity of the allocated LLC resource in the entire adjustment

process is acquired, and the variation is used as the performance data of each program.

[0039] S202: Determine a category to which each program belongs in light of a memory access frequency, obtained by means of statistics collection, of each program in the working set.

5 [0040] Specifically, a hardware performance counter is used to count a quantity of times that each program accesses a main memory in a preset stage or time period of a running process, to obtain the memory access frequency of each program.

[0041] Then, the category to which each program belongs is determined in light of the performance data of each program and the memory access frequency of each program and
10 according to a preset threshold.

[0042] The preset threshold includes a first threshold, a second threshold, and a third threshold, where the first threshold and the second threshold are performance data thresholds, and the third threshold is a memory access frequency threshold.

[0043] Exemplarily, the performance data is a variation that is generated when the
15 normalization speed-up ratio of each program varies with the capacity of the LLC resource allocated to each program. In a process in which the capacity of the LLC allocated to each program is adjusted and decreased from the maximum capacity of the LLC to 1/N of the maximum capacity of the LLC, taking a quad-core computer as an example, where N is 8, as shown in FIG. 3, a curve in which the normalization speed-up ratio of each program varies
20 with the capacity of the LLC resource allocated to each program is obtained. Therefore, it may be learned that a main categorization manner is as follows:

if a normalization speed-up ratio of a program is greater than the first threshold, it can be known that deterioration in performance of such a program is very considerable when the capacity of the LLC resource decreases, that is, the performance of such a program is
25 greatly affected by the capacity of the allocated LLC resource, and such a program is categorized as a high demand type 031;

if a normalization speed-up ratio of a program is less than the first threshold and greater than the second threshold, it can be known that deterioration in performance of such a program is moderate when the capacity of the LLC resource decreases, that is, the
30 performance of such a program is moderately affected by the capacity of the allocated LLC

resource, and such a program is categorized as a medium demand type 032; or

if a normalization speed-up ratio of a program is less than the second threshold and a memory access frequency is greater than the third threshold, it can be known that a variation range of performance of such a program is very small when the capacity of the LLC resource decreases, and that the memory access frequency of the program is high, that is, the performance of such a program is relatively slightly affected by the capacity of the allocated LLC resource, and the program is of a low demand type 033; however, because the performance of such a program is greatly affected by a capacity of the main memory, the program is of a memory access intensive type; therefore, such a program is categorized as a low demand and intensive type.

[0044] Exemplarily, the foregoing high demand type 031 may further specifically include last level cache fitting (LLC Fitting, LLCF) and last level cache friendly (LLC Friendly, LLCFR); the foregoing medium demand type 032 may specifically include last level cache swing (LLC Swing, LLCS); the foregoing low demand type 033 may specifically include: core cache fitting (Core Cache Fitting, CCS) and LLC thrashing (LLC Thrashing, LLCT).

[0045] S203: Select a page coloring-based partitioning policy corresponding to the working set according to the category to which each program in the working set belongs.

[0046] Specifically, the category to which each program in the working set belongs is written to an operating system kernel, and then the page coloring-based partitioning policy corresponding to the working set is selected according to a preset decision policy.

[0047] The category to which each program in the working set belongs may be written to the operating system kernel by using a /proc mechanism. The preset decision policy may be a partitioning policy decision tree in the operating system kernel, and the partitioning policy decision tree is implemented in the operating system kernel in algorithm form.

[0048] Specifically, categorization of each program in the working set is written to the operating system kernel, the partitioning policy decision tree in the operating system kernel is searched for a corresponding node in light of the category of each program in the working set, so as to determine the page coloring-based partitioning policy corresponding to the working set. The page coloring-based partitioning policy includes a page coloring-based collaborative partitioning policy and a page coloring-based non-collaborative partitioning policy.

[0049] The page coloring-based collaborative partitioning policy is a partitioning policy of using overlapped index address bits O-bits as page coloring-based partitioning index bits, and the O-bits are overlapped address bits of index bits of the LLC and index bits of a DRAM Bank in a physical page frame, and are used to index page coloring-based partitioning for both the LLC and the DRAM Bank.

[0050] The page coloring-based collaborative partitioning policy includes:

a category A multi-level memory collaborative partitioning policy A-MMCP, in which the O-bits are used as partitioning index bits, and the LLC and the DRAM Bank are partitioned into a same quantity of equal portions;

a category B multi-level memory collaborative partitioning policy B-MMCP, in which the O-bits and an index bit of the DRAM Bank are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is greater than a quantity of partitioned equal portions of the LLC; and

a category C multi-level memory collaborative partitioning policy C-MMCP, in which the O-bits and an index bit of the LLC are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is less than a quantity of partitioned equal portions of the LLC.

[0051] The page coloring-based collaborative partitioning policy is a partitioning policy in which the O-bits are not used and includes:

a Cache-Only policy, in which coloring-based partitioning is performed on the LLC by using the index bit of the LLC, and coloring-based partitioning is not performed on the DRAM Bank; and

a Bank-Only policy, in which coloring-based partitioning is performed on the DRAM Bank by using the index bit of the DRAM Bank, and coloring-based partitioning is not performed on the LLC.

[0052] Exemplarily, the overlapped index address bits O-bits of the LLC and the DRAM Bank may be the 14th and 15th bits in the physical page frame, and four colors, that is, 00, 01, 10, and 11, may be obtained by means of partitioning by using the two index bits. Further,

index bits {16, 17, 18} of the LLC and index bits {21, 22} of the DRAM Bank also exist in the physical page frame.

[0053] In the A-MMCP, coloring-based partitioning is performed only by using the O-bits, that is, the 14th and 15th bits, the LLC may be partitioned into 4 equal portions, and the
5 DRAM Bank may be partitioned into 4 equal portions.

[0054] In the B-MMCP, coloring-based partitioning is performed by using the O-bits and one of the index bits of the DRAM Bank, that is, the 14th, 15th, and 21st bits, the LLC may be partitioned into 4 equal portions, and the DRAM Bank may be partitioned into 8 equal portions.

10 **[0055]** In the C-MMCP, coloring-based partitioning is performed by using the O-bits and one of the index bits of the LLC, that is, the 14th, 15th, and 16th bits, the LLC may be partitioned into 8 equal portions, and the DRAM Bank may be partitioned into 4 equal portions.

[0056] Specifically, that the partitioning policy decision tree is implemented in the
15 operating system kernel in algorithm form includes:

if each category to which each program in the working set belongs is the high demand type 031, the Bank-Only policy is selected;

if there is the low demand and intensive type in a category to which each program in the working set belongs, a quantity of programs in the working set is further determined;
20 and if the quantity of programs is less than or equal to N, the A-MMCP is selected; otherwise, the C-MMCP is selected, where N is a quantity of cores of a processor; or

if there is the medium demand type 032 and there is no low demand and intensive type in a category to which each program in the working set belongs, a quantity of programs in the working set is further determined; and if the quantity of programs is less than or equal
25 to N, the A-MMCP is selected; otherwise, the B-MMCP is selected, where N is a quantity of cores of a processor.

[0057] S204: Write the page coloring-based partitioning policy corresponding to the working set to an operating system kernel, where the operating system kernel performs corresponding page coloring-based partitioning processing.

30 **[0058]** Specifically, the selected page coloring-based partitioning policy may be written to

the operating system kernel by using the /proc mechanism, and a buddy system (Buddy System) in an operating system adjusts a memory allocation mechanism, to complete a coloring-based partitioning operation.

[0059] Exemplarily, a management mechanism and a retrieval mechanism in the buddy
 5 system may complete management and retrieval on a page required for coloring; the buddy system may adjust the memory allocation mechanism, to enable the memory allocation mechanism to switch among various partitioning policies.

[0060] According to the memory resource optimization method provided in this
 embodiment of the present invention, an LLC resource is partitioned by using a page coloring
 10 technology, performance data of each program in a working set is acquired, a category of each program is obtained in light of a memory access frequency, a page coloring-based partitioning policy of the working set is selected according to the category of each program, and the page coloring-based partitioning policy is written to an operating system kernel, to complete
 corresponding coloring-based partitioning processing. In this way, a collaborative partitioning
 15 policy between an LLC and a DRAM Bank is implemented in light of a feature of the working set, and mutual interference of processes or threads on a storage resource can be reduced and even eliminated, thereby improving overall performance of a computer.

[0061] An embodiment of the present invention provides a memory resource optimization
 apparatus 00. As shown in FIG. 4, the apparatus includes:

20 a front end unit 001, configured to acquire performance data of each program in a working set, and categorize each program by comparing the performance data of each program and a memory access frequency, obtained by means of statistics collection, of each program with a preset threshold, where the performance data of each program is a variation that is generated when a preset performance indicator of each program varies with a capacity
 25 of an allocated last level cache LLC resource;

a decision-making unit 002, configured to select, in light of categorization of each
 program in the working set and a preset decision policy, a page coloring-based partitioning
 policy corresponding to the working set, where the page coloring-based partitioning policy
 includes a page coloring-based collaborative partitioning policy for performing page
 30 coloring-based partitioning on both an LLC and a dynamic random access memory bank

DRAM Bank; and

a partitioning unit 003, configured to write the page coloring-based partitioning policy corresponding to the working set to an operating system kernel, where the operating system kernel performs corresponding page coloring-based partitioning processing.

5 [0062] As shown in FIG. 5, the front end unit 001 may include a data collecting unit 0011 and a categorizing unit 0012.

[0063] Optionally, the data collecting unit 0011 is configured to partition the LLC resource into N portions by using a page coloring technology, take 1/N of a maximum capacity of the LLC resource as one level, allocate the maximum capacity of the LLC
10 resource to each program at the beginning, and decrease the capacity of the LLC resource allocated to each program by one level in each adjustment, until the capacity is decreased to 1/N of the maximum capacity of the LLC resource; and

monitor a variation that is generated when the preset performance indicator of each program varies with the capacity of the allocated LLC resource in an adjustment process, and
15 use the variation as the performance data of each program, where the preset performance indicator is a speed-up ratio of each program.

[0064] Optionally, the categorizing unit 0012 is configured to count a quantity of times that each program accesses a main memory in a preset stage of a running process, to obtain the memory access frequency of each program;

20 compare the performance data of each program and the memory access frequency, obtained by means of statistics collection, of each program with the preset threshold, where the preset threshold includes a first threshold, a second threshold, and a third threshold, the first threshold and the second threshold are performance data thresholds, and the third threshold is a memory access frequency threshold; and

25 if performance data of a program is greater than the first threshold, categorize the program as a high demand type;

if performance data of a program is less than the first threshold and greater than the second threshold, categorize the program as a medium demand type; or

if performance data of a program is less than the second threshold and a memory
30 access frequency is greater than the third threshold, categorize the program as a low demand

and intensive type.

[0065] Optionally, the preset decision policy is a partitioning policy decision tree in the operating system kernel, and the partitioning policy decision tree is implemented in the operating system kernel in algorithm form; the decision-making unit 002 is specifically
5 configured to:

write the categorization of each program in the working set to the operating system kernel, and search the partitioning policy decision tree in the operating system kernel for a corresponding node in light of a category of each program in the working set, to determine the page coloring-based partitioning policy corresponding to the working set.

10 [0066] Optionally, the page coloring-based partitioning policy includes the page coloring-based collaborative partitioning policy and a page coloring-based non-collaborative partitioning policy, where, specifically:

the page coloring-based collaborative partitioning policy is a partitioning policy of using overlapped index address bits O-bits as page coloring-based partitioning index bits, and
15 the O-bits are overlapped address bits of index bits of the LLC and index bits of the DRAM Bank in a physical page frame, and are used to index page coloring-based partitioning for both the LLC and the DRAM Bank, where

the page coloring-based collaborative partitioning policy includes:

a category A multi-level memory collaborative partitioning policy A-MMCP, in
20 which the O-bits are used as partitioning index bits, and the LLC and the DRAM Bank are partitioned into a same quantity of equal portions;

a category B multi-level memory collaborative partitioning policy B-MMCP, in which the O-bits and an index bit of the DRAM Bank are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a
25 quantity of partitioned equal portions of the DRAM Bank is greater than a quantity of partitioned equal portions of the LLC; and

a category C multi-level memory collaborative partitioning policy C-MMCP, in which the O-bits and an index bit of the LLC are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of
30 partitioned equal portions of the DRAM Bank is less than a quantity of partitioned equal

portions of the LLC; and

the page coloring-based collaborative partitioning policy is a partitioning policy in which the O-bits are not used and includes:

a Cache-Only policy, in which coloring-based partitioning is performed on the LLC by using the index bit of the LLC, and coloring-based partitioning is not performed on the DRAM Bank; and

a Bank-Only policy, in which coloring-based partitioning is performed on the DRAM Bank by using the index bit of the DRAM Bank, and coloring-based partitioning is not performed on the LLC.

10 [0067] Optionally, the decision-making unit 002 is further specifically configured to:

if each category to which each program in the working set belongs is the high demand type, select the Bank-Only policy;

if there is the low demand and intensive type in a category to which each program in the working set belongs, further determine a quantity of programs in the working set; and if the quantity of programs is less than or equal to N, select the A-MMCP; otherwise, select the C-MMCP, where N is a quantity of cores of a processor; or

if there is the medium demand type and there is no low demand and intensive type in a category to which each program in the working set belongs, further determine a quantity of programs in the working set; and if the quantity of programs is less than or equal to N, select the A-MMCP; otherwise, select the B-MMCP, where N is a quantity of cores of a processor.

[0068] According to the memory resource optimization apparatus provided in this embodiment of the present invention, an LLC resource is partitioned by using a page coloring technology, performance data of each program in a working set is acquired, a category of each program is obtained in light of a memory access frequency, a page coloring-based partitioning policy of the working set is selected according to the category of each program, and the page coloring-based partitioning policy is written to an operating system kernel, to complete corresponding coloring-based partitioning processing. In this way, a collaborative partitioning policy between an LLC and a DRAM Bank is implemented in light of a feature of the working set, and mutual interference of processes or threads on a storage resource can be

reduced and even eliminated, thereby improving overall performance of a computer.

[0069] An embodiment of the present invention further provides a memory resource optimization apparatus 01. As shown in FIG. 6, the memory resource optimization apparatus 01 includes:

5 a bus 011, and a processor 012, a memory 013, and an interface 014 connected to the bus 011, where the interface 014 is configured to communicate with an external device.

[0070] The memory 013 is configured to store an instruction. The processor 012 is configured to execute the instruction, and is configured to acquire performance data of each program in a working set, and categorize each program by comparing the performance data of
10 each program and a memory access frequency, obtained by means of statistics collection, of each program with a preset threshold, where the performance data of each program is a variation that is generated when a preset performance indicator of each program varies with a capacity of an allocated last level cache LLC resource.

[0071] The processor 012 executes the instruction, and is further configured to select, in
15 light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set, where the page coloring-based partitioning policy includes a page coloring-based collaborative partitioning policy for performing page coloring-based partitioning on both an LLC and a dynamic random access memory bank DRAM Bank.

20 **[0072]** The processor 012 executes the instruction, and is further configured to write the page coloring-based partitioning policy corresponding to the working set to an operating system kernel, where the operating system kernel performs corresponding page coloring-based partitioning processing.

[0073] In this embodiment of the present invention, optionally, the processor 012 executes
25 the instruction, and may be specifically configured to: partition the LLC resource into N portions by using a page coloring technology, take $1/N$ of a maximum capacity of the LLC resource as one level, allocate the maximum capacity of the LLC resource to each program at the beginning, and decrease the capacity of the LLC resource allocated to each program by one level in each adjustment, until the capacity is decreased to $1/N$ of the maximum capacity
30 of the LLC resource; and

monitor a variation that is generated when the preset performance indicator of each program varies with the capacity of the allocated LLC resource in an adjustment process, and use the variation as the performance data of each program, where the preset performance indicator is a speed-up ratio of each program.

5 [0074] In this embodiment of the present invention, optionally, the processor 012 executes the instruction, and may be specifically configured to: count a quantity of times that each program accesses a main memory in a preset stage of a running process, to obtain the memory access frequency of each program;

compare the performance data of each program and the memory access frequency,
10 obtained by means of statistics collection, of each program with the preset threshold, where the preset threshold includes a first threshold, a second threshold, and a third threshold, the first threshold and the second threshold are performance data thresholds, and the third threshold is a memory access frequency threshold; and

if performance data of a program is greater than the first threshold, categorize the
15 program as a high demand type;

if performance data of a program is less than the first threshold and greater than the second threshold, categorize the program as a medium demand type; or

if performance data of a program is less than the second threshold and a memory access frequency is greater than the third threshold, categorize the program as a low demand
20 and intensive type.

[0075] In this embodiment of the present invention, optionally, the preset decision policy is a partitioning policy decision tree in the operating system kernel, and the partitioning policy decision tree is implemented in the operating system kernel in algorithm form. The processor 012 executes the instruction, and may be specifically configured to:

25 write the categorization of each program in the working set to the operating system kernel, and search the partitioning policy decision tree in the operating system kernel for a corresponding node in light of a category of each program in the working set, to determine the page coloring-based partitioning policy corresponding to the working set.

[0076] In this embodiment of the present invention, optionally, the page coloring-based
30 partitioning policy includes the page coloring-based collaborative partitioning policy and a

page coloring-based non-collaborative partitioning policy, where, specifically:

the page coloring-based collaborative partitioning policy is a partitioning policy of using overlapped index address bits O-bits as page coloring-based partitioning index bits, and the O-bits are overlapped address bits of index bits of the LLC and index bits of the DRAM Bank in a physical page frame, and are used to index page coloring-based partitioning for both the LLC and the DRAM Bank, where

the page coloring-based collaborative partitioning policy includes:

a category A multi-level memory collaborative partitioning policy A-MMCP, in which the O-bits are used as partitioning index bits, and the LLC and the DRAM Bank are partitioned into a same quantity of equal portions;

a category B multi-level memory collaborative partitioning policy B-MMCP, in which the O-bits and an index bit of the DRAM Bank are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is greater than a quantity of partitioned equal portions of the LLC; and

a category C multi-level memory collaborative partitioning policy C-MMCP, in which the O-bits and an index bit of the LLC are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is less than a quantity of partitioned equal portions of the LLC; and

the page coloring-based collaborative partitioning policy is a partitioning policy in which the O-bits are not used and includes:

a Cache-Only policy, in which coloring-based partitioning is performed on the LLC by using the index bit of the LLC, and coloring-based partitioning is not performed on the DRAM Bank; and

a Bank-Only policy, in which coloring-based partitioning is performed on the DRAM Bank by using the index bit of the DRAM Bank, and coloring-based partitioning is not performed on the LLC.

[0077] In this embodiment of the present invention, optionally, the processor 012 executes the instruction, and may be specifically configured to:

if each category to which each program in the working set belongs is the high demand type, select the Bank-Only policy;

if there is the low demand and intensive type in a category to which each program in the working set belongs, further determine a quantity of programs in the working set; and if the quantity of programs is less than or equal to N, select the A-MMCP; otherwise, select the C-MMCP, where N is a quantity of cores of a processor; or

if there is the medium demand type and there is no low demand and intensive type in a category to which each program in the working set belongs, further determine a quantity of programs in the working set; and if the quantity of programs is less than or equal to N, select the A-MMCP; otherwise, select the B-MMCP, where N is a quantity of cores of a processor.

[0078] According to the memory resource optimization apparatus provided in this embodiment of the present invention, an LLC resource is partitioned by using a page coloring technology, performance data of each program in a working set is acquired, a category of each program is obtained in light of a memory access frequency, a page coloring-based partitioning policy of the working set is selected according to the category of each program, and the page coloring-based partitioning policy is written to an operating system kernel, to complete corresponding coloring-based partitioning processing. In this way, a collaborative partitioning policy between an LLC and a DRAM Bank is implemented in light of a feature of the working set, and mutual interference of processes or threads on a storage resource can be reduced and even eliminated, thereby improving overall performance of a computer.

[0079] In the several embodiments provided in the present application, it should be understood that the disclosed apparatus and method may be implemented in other manners. For example, the described apparatus embodiments are merely exemplary. For example, the module division is merely logical function division, and there may be other division manners in actual implementation. In addition, the displayed or discussed connections between modules may be implemented by using some interfaces, and may be implemented in electronic, mechanical, or other forms.

[0080] The modules may or may not be physically separate, and may or may not be physical units. Some or all of the modules may be selected according to actual requirements

to achieve the objectives of the solutions of the embodiments.

[0081] In addition, functional modules in the embodiments of the present invention may be integrated into one processing module, or each of the modules may exist alone physically, or two or more modules are integrated into one module. The integrated module may be
5 implemented in a form of hardware, or may be implemented in a form of hardware in addition to a software functional module.

[0082] When the foregoing integrated module is implemented in a form of a software functional unit, the integrated unit may be stored in a computer-readable storage medium. The software functional module is stored in a storage medium and includes several instructions for
10 instructing a computer device (which may be a personal computer, a server, or a network device) to perform some of the steps of the methods described in the embodiments of the present invention. The foregoing storage medium includes: any medium that can store program code, such as a USB flash drive, a removable hard disk, a read-only memory (Read-Only Memory, ROM), a random access memory (Random Access Memory, RAM), a
15 magnetic disk, or an optical disc.

[0083] The foregoing descriptions are merely specific implementation manners of the present invention, but are not intended to limit the protection scope of the present invention. Any variation or replacement readily figured out by a person skilled in the art within the technical scope disclosed in the present invention shall fall within the protection scope of the
20 present invention. Therefore, the protection scope of the present invention shall be subject to the protection scope of the claims.

CLAIMS

What is claimed is:

1. A memory resource optimization method, wherein the method comprises:
acquiring performance data of each program in a working set;
5 categorizing each program according to the performance data of each program and a
memory access frequency, obtained by means of statistics collection, of each program,
wherein the performance data of each program is a variation that is generated when a preset
performance indicator of each program varies with a capacity of resource of an allocated last
level cache LLC;
10 selecting, in light of categorization of each program in the working set and a preset
decision policy, a page coloring-based partitioning policy corresponding to the working set,
wherein the page coloring-based partitioning policy comprises a page coloring-based
collaborative partitioning policy for performing page coloring-based partitioning on both the
LLC and a dynamic random access memory bank DRAM Bank; and
15 writing the page coloring-based partitioning policy corresponding to the working set to
an operating system kernel, wherein the operating system kernel performs corresponding page
coloring-based partitioning processing.
2. The method according to claim 1, wherein the acquiring performance data of each
program in a working set comprises:
20 partitioning the resource of the LLC into N portions, taking $1/N$ of a maximum capacity
of resource of the LLC as one level, allocating the maximum capacity of resource of the LLC
to each program at the beginning, and decreasing the capacity of resource of the LLC
allocated to each program by one level in each adjustment, until the capacity is decreased to
 $1/N$ of the maximum capacity of resource of the LLC; and
25 monitoring a variation that is generated when the preset performance indicator of each
program varies with the capacity of resource of the allocated LLC in an adjustment process,
and using the variation as the performance data of each program, wherein the preset
performance indicator is a speed-up ratio of each program.

3. The method according to claim 1 or 2, wherein the page coloring-based collaborative partitioning policy is a partitioning policy of using overlapped index address bits O-bits as page coloring-based partitioning index bits, and the O-bits are overlapped address bits of index bits of the LLC and index bits of the DRAM Bank in a physical page frame, and are used to index page coloring-based partitioning for both the LLC and the DRAM Bank; and
- the page coloring-based collaborative partitioning policy comprises:
- a category A multi-level memory collaborative partitioning policy A-MMCP, in which the O-bits are used as partitioning index bits, and the LLC and the DRAM Bank are partitioned into a same quantity of equal portions;
- a category B multi-level memory collaborative partitioning policy B-MMCP, in which the O-bits and an index bit of the DRAM Bank are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is greater than a quantity of partitioned equal portions of the LLC; and
- a category C multi-level memory collaborative partitioning policy C-MMCP, in which the O-bits and an index bit of the LLC are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is less than a quantity of partitioned equal portions of the LLC.
4. The method according to claim 3, wherein the categorizing each program according to the performance data of each program and a memory access frequency, obtained by means of statistics collection, of each program comprises:
- counting a quantity of times that each program accesses a main memory in a preset stage of a running process, to obtain the memory access frequency of each program;
- comparing the performance data of each program with a preset performance data threshold, wherein the performance data threshold comprises a first threshold and a second threshold, and the first threshold is greater than the second threshold; and
- if performance data of a program is greater than the first threshold, categorizing the program as a high demand type;
- if performance data of a program is less than the first threshold and greater than the

second threshold, categorizing the program as a medium demand type; or

if performance data of a program is less than the second threshold and a memory access frequency of the program is greater than a preset memory access frequency threshold, categorizing the program as a low demand and intensive type.

5 5. The method according to claim 4, wherein the selecting, in light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set comprises:

if there is the low demand and intensive type in a category to which each program in the working set belongs, further determining a quantity of programs in the working set; and if the
10 quantity of programs is less than or equal to N, selecting the A-MMCP; otherwise, selecting the C-MMCP, wherein N is a quantity of cores of a processor; or

if there is the medium demand type and there is no low demand and intensive type in a category to which each program in the working set belongs, further determining a quantity of programs in the working set; and if the quantity of programs is less than or equal to N,
15 selecting the A-MMCP; otherwise, selecting the B-MMCP, wherein N is a quantity of cores of a processor.

6. The method according to claim 3, wherein the page coloring-based partitioning policy further comprises a page coloring-based non-collaborative partitioning policy, the page coloring-based non-collaborative partitioning policy is a partitioning policy in which the
20 O-bits are not used, and the page coloring-based non-collaborative partitioning policy comprises:

a Cache-Only policy, in which the page coloring-based partitioning is performed on the LLC by using the index bit of the LLC, and the page coloring-based partitioning is not performed on the DRAM Bank; and

25 a Bank-Only policy, in which the page coloring-based partitioning is performed on the DRAM Bank by using the index bit of the DRAM Bank, and the page coloring-based partitioning is not performed on the LLC; and

the selecting, in light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set
30 further comprises:

if each category to which each program in the working set belongs is the high demand type, selecting the Bank-Only policy.

7. The method according to any one of claims 1 to 6, wherein the preset decision policy is a partitioning policy decision tree in the operating system kernel, and the partitioning policy
5 decision tree is implemented in the operating system kernel in algorithm form; and

the selecting, in light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set comprises:

inputting the categorization of each program in the working set to the operating system
10 kernel, and searching the partitioning policy decision tree in the operating system kernel for a corresponding node in light of a category of each program in the working set, to determine the page coloring-based partitioning policy corresponding to the working set.

8. A memory resource optimization apparatus, comprising a memory and a processor coupled to the memory, wherein the memory is configured to store an instruction, and
15 wherein the processor is configured to execute the instruction and is configured to :

acquire performance data of each program in a working set;

categorize each program according to the performance data of each program and a memory access frequency, obtained by means of statistics collection, of each program, wherein the performance data of each program is a variation that is generated when a preset
20 performance indicator of each program varies with a capacity of resource of an allocated last level cache LLC;

select, in light of categorization of each program in the working set and a preset decision policy, a page coloring-based partitioning policy corresponding to the working set, wherein the page coloring-based partitioning policy comprises a page coloring-based collaborative
25 partitioning policy for performing page coloring-based partitioning on both the LLC and a dynamic random access memory bank DRAM Bank; and

write the page coloring-based partitioning policy corresponding to the working set to an operating system kernel, wherein the operating system kernel performs corresponding page coloring-based partitioning processing.

30 9. The apparatus according to claim 8, wherein in the step of acquiring, the processor is

configured to:

partition the resource of LLC into N portions, take 1/N of a maximum capacity of resource of the LLC as one level, allocate the maximum capacity of resource of the LLC to each program at the beginning, and decrease the capacity of resource of the LLC allocated to each program by one level in each adjustment, until the capacity is decreased to 1/N of the maximum capacity of resource of the LLC; and

monitor a variation that is generated when the preset performance indicator of each program varies with the capacity of resource of the allocated LLC in an adjustment process, and use the variation as the performance data of each program, wherein the preset performance indicator is a speed-up ratio of each program.

10. The apparatus according to claim 8 or 9, wherein the page coloring-based collaborative partitioning policy is a partitioning policy of using overlapped index address bits O-bits as page coloring-based partitioning index bits, and the O-bits are overlapped address bits of index bits of the LLC and index bits of the DRAM Bank in a physical page frame, and are used to index page coloring-based partitioning for both the LLC and the DRAM Bank; and

the page coloring-based collaborative partitioning policy comprises:

a category A multi-level memory collaborative partitioning policy A-MMCP, in which the O-bits are used as partitioning index bits, and the LLC and the DRAM Bank are partitioned into a same quantity of equal portions;

a category B multi-level memory collaborative partitioning policy B-MMCP, in which the O-bits and an index bit of the DRAM Bank are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is greater than a quantity of partitioned equal portions of the LLC; and

a category C multi-level memory collaborative partitioning policy C-MMCP, in which the O-bits and an index bit of the LLC are used as partitioning index bits, the LLC and the DRAM Bank are partitioned into different quantities of equal portions, and a quantity of partitioned equal portions of the DRAM Bank is less than a quantity of partitioned equal portions of the LLC.

11. The apparatus according to claim 10, wherein in the step of categorizing, the processor is configured to:

count a quantity of times that each program accesses a main memory in a preset stage of a running process, to obtain the memory access frequency of each program;

5 compare the performance data of each program with a preset performance data threshold, wherein the performance data threshold comprises a first threshold and a second threshold, and the first threshold is greater than the second threshold; and

if performance data of a program is greater than the first threshold, categorize the program as a high demand type;

10 if performance data of a program is less than the first threshold and greater than the second threshold, categorize the program as a medium demand type; or

if performance data of a program is less than the second threshold and a memory access frequency of the program is greater than a preset memory access frequency threshold, categorize the program as a low demand and intensive type.

15 12. The apparatus according to claim 11, wherein in the step of selecting, the processor is configured to:

if there is the low demand and intensive type in a category to which each program in the working set belongs, further determine a quantity of programs in the working set; and if the quantity of programs is less than or equal to N, select the A-MMCP; otherwise, select the
20 C-MMCP, wherein N is a quantity of cores of a processor; or

if there is the medium demand type and there is no low demand and intensive type in a category to which each program in the working set belongs, further determine a quantity of programs in the working set; and if the quantity of programs is less than or equal to N, select the A-MMCP; otherwise, select the B-MMCP, wherein N is a quantity of cores of a
25 processor.

13. The apparatus according to claim 12, wherein the page coloring-based partitioning policy further comprises a page coloring-based non-collaborative partitioning policy, the page coloring-based non-collaborative partitioning policy is a partitioning policy in which the O-bits are not used, and the page coloring-based non-collaborative partitioning policy
30 comprises:

a Cache-Only policy, in which the page coloring-based partitioning is performed on the LLC by using the index bit of the LLC, and the page coloring-based partitioning is not performed on the DRAM Bank; and

5 a Bank-Only policy, in which the page coloring-based partitioning is performed on the DRAM Bank by using the index bit of the DRAM Bank, and the page coloring-based partitioning is not performed on the LLC; and

the processor is further configured to: if each category to which each program in the working set belongs is the high demand type, select the Bank-Only policy.

10 14. The apparatus according to any one of claims 8 to 13, wherein the preset decision policy is a partitioning policy decision tree in the operating system kernel, and the partitioning policy decision tree is implemented in the operating system kernel in algorithm form; and

the processor is configured to:

15 input the categorization of each program in the working set to the operating system kernel, and search the partitioning policy decision tree in the operating system kernel for a corresponding node in light of a category of each program in the working set, to determine the page coloring-based partitioning policy corresponding to the working set.

1/4

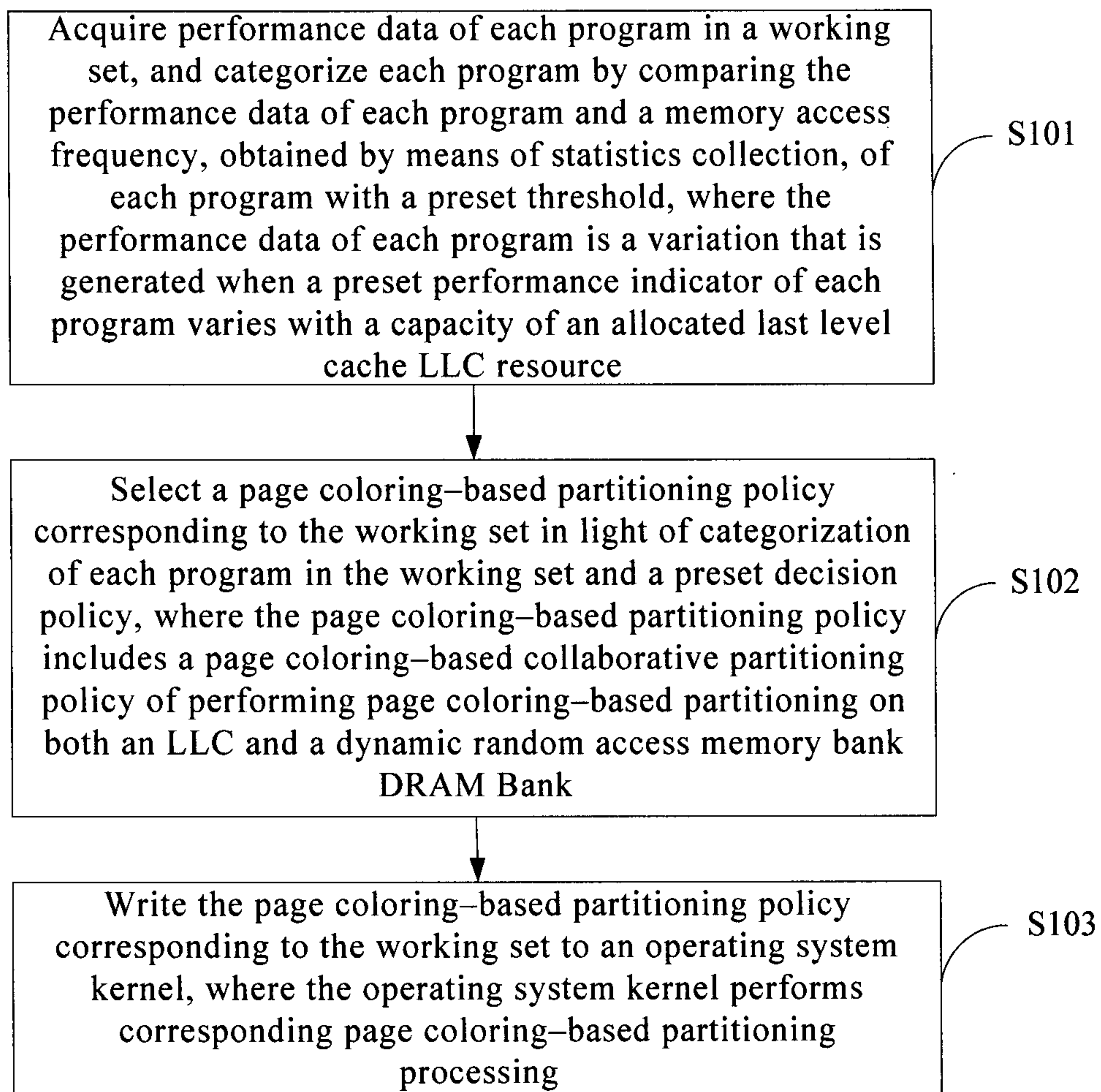


FIG. 1

2/4

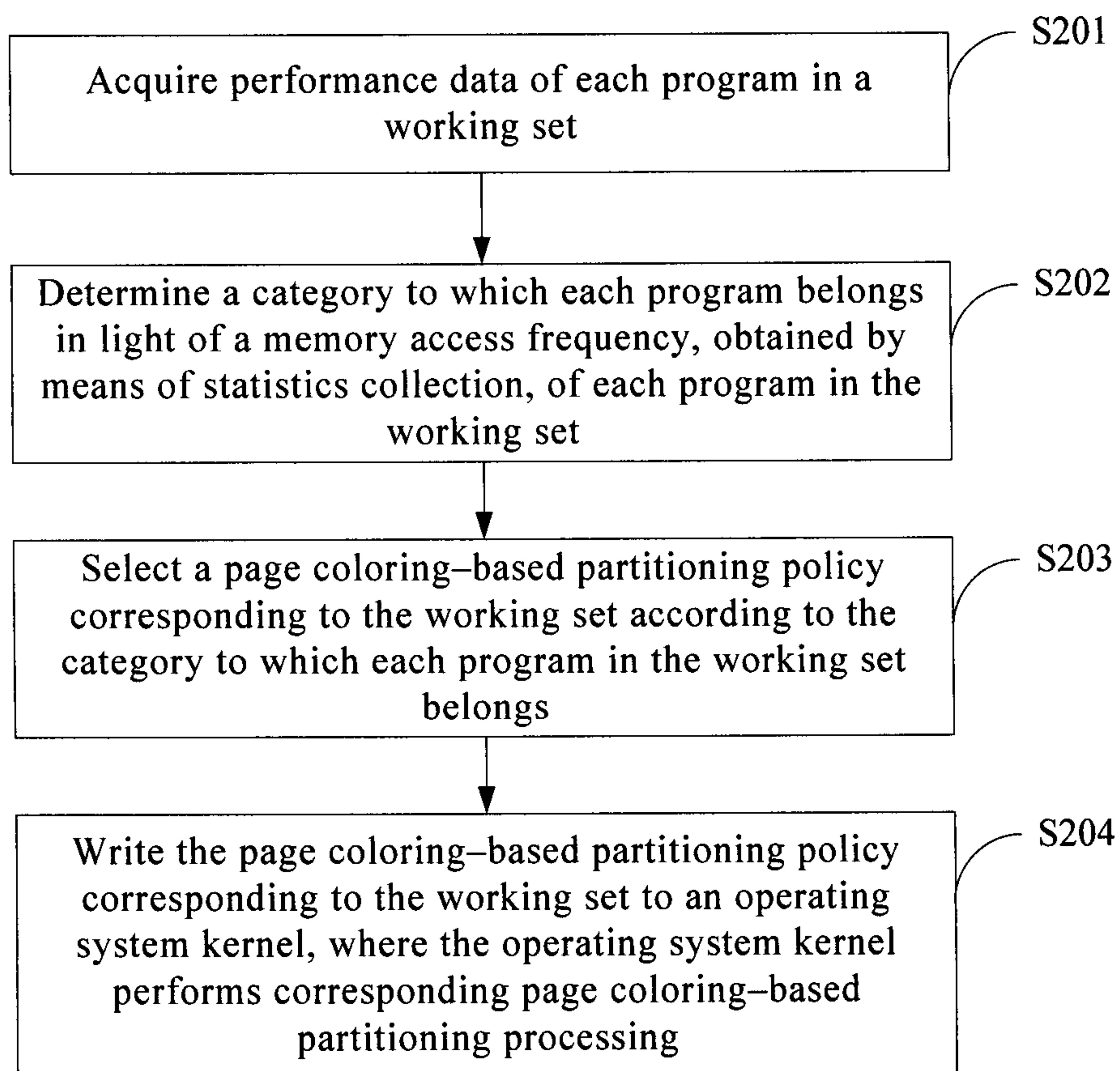


FIG. 2

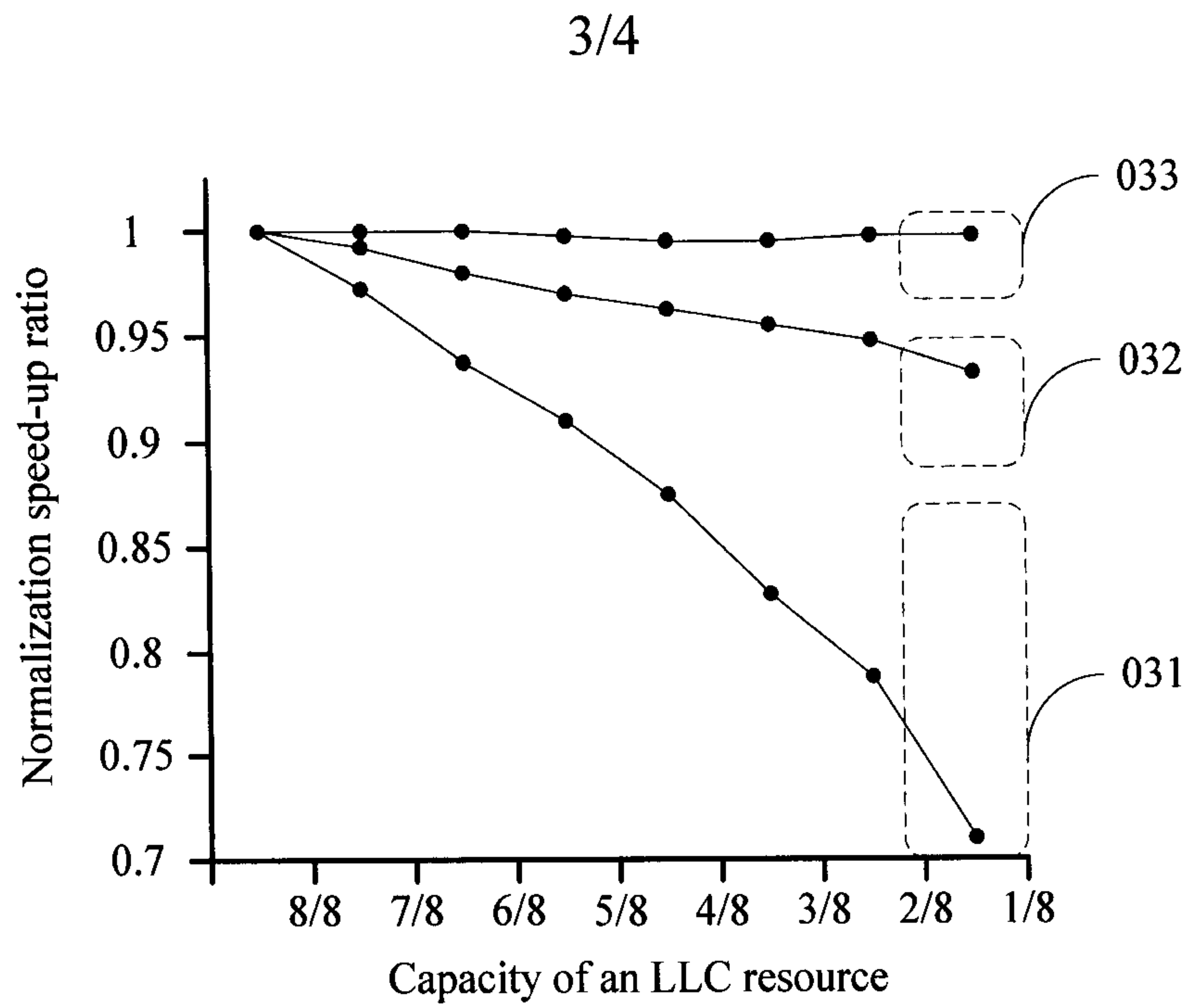


FIG. 3

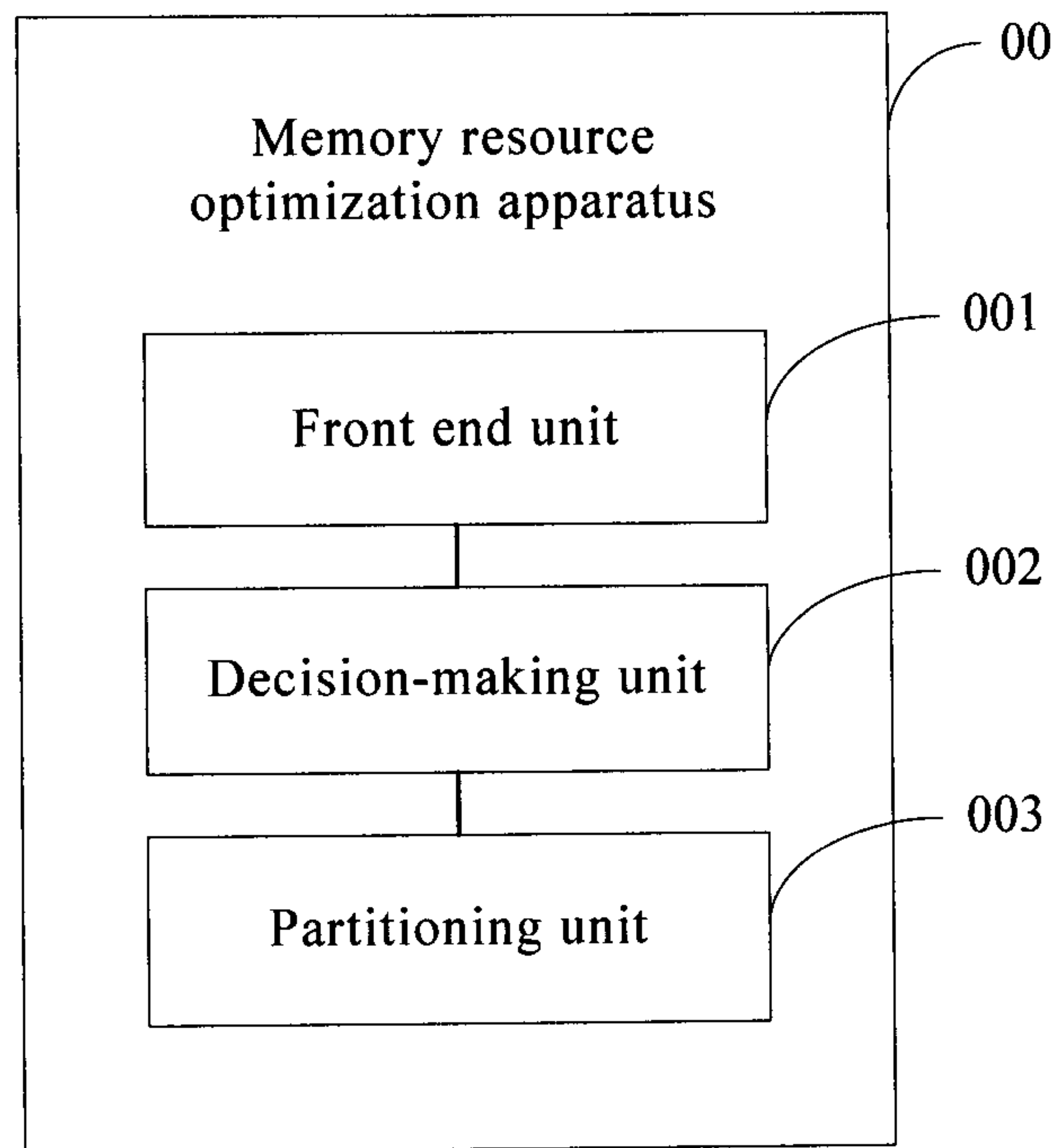


FIG. 4

4/4

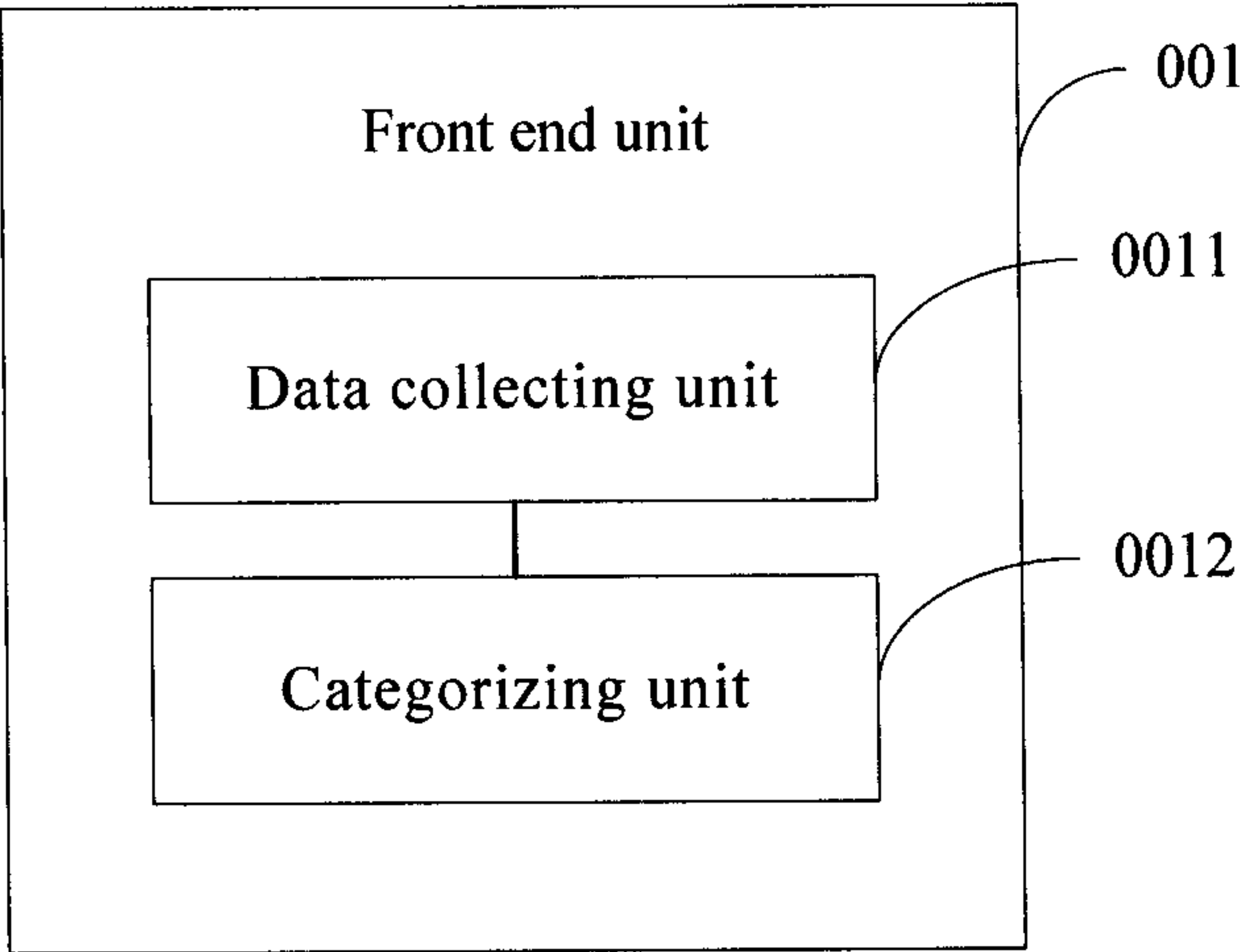


FIG. 5

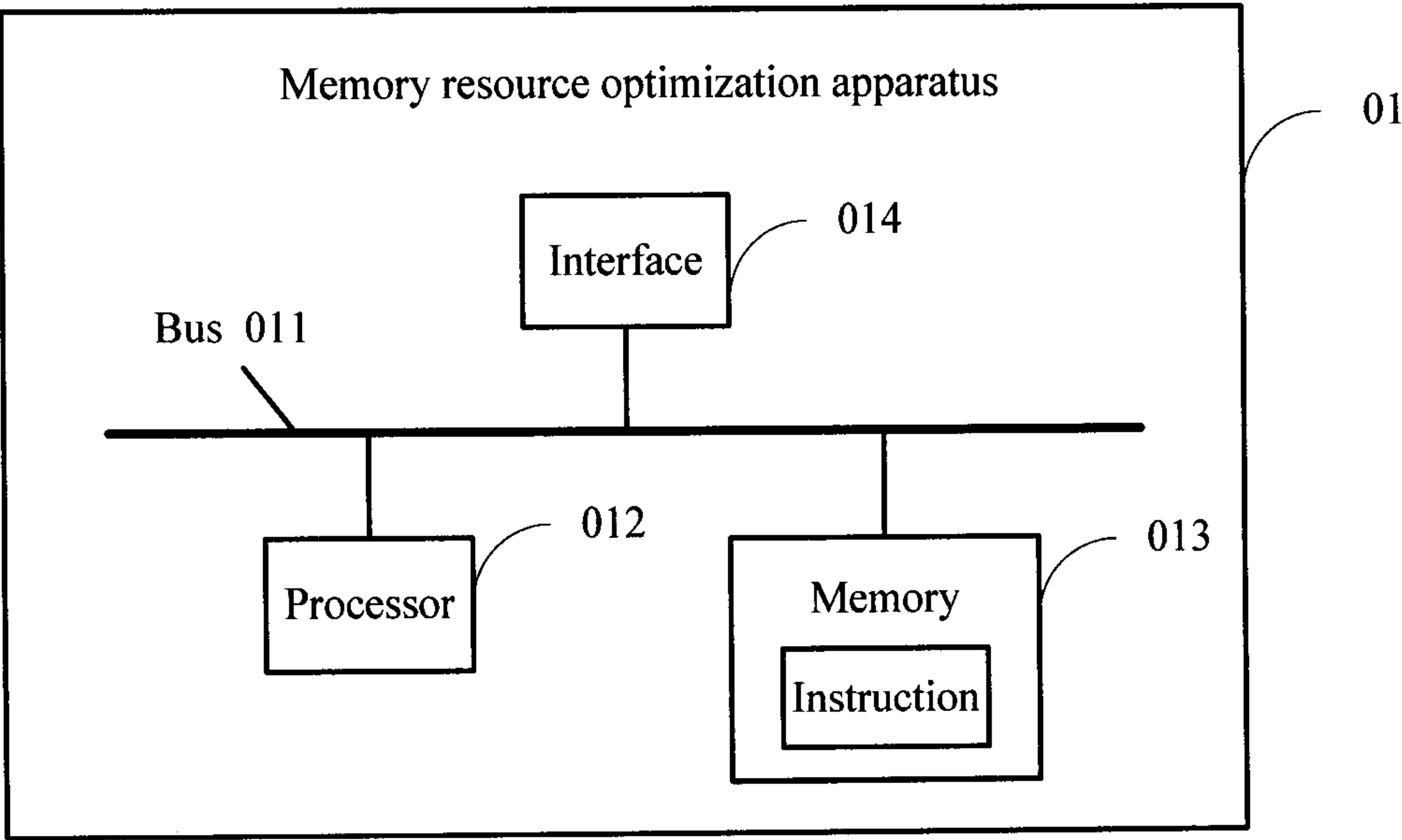


FIG. 6

Acquire performance data of each program in a working set, and categorize each program by comparing the performance data of each program and a memory access frequency, obtained by means of statistics collection, of each program with a preset threshold, where the performance data of each program is a variation that is generated when a preset performance indicator of each program varies with a capacity of an allocated last level cache LLC resource

S101

Select a page coloring-based partitioning policy corresponding to the working set in light of categorization of each program in the working set and a preset decision policy, where the page coloring-based partitioning policy includes a page coloring-based collaborative partitioning policy of performing page coloring-based partitioning on both an LLC and a dynamic random access memory bank DRAM Bank

S102

Write the page coloring-based partitioning policy corresponding to the working set to an operating system kernel, where the operating system kernel performs corresponding page coloring-based partitioning processing

S103