MEMORY RESOURCE OPTIMIZATION METHOD AND APPARATUS

TECHNICAL FIELD

[0001] The present invention relates to the computer field, and in particular, to a memory 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 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 memory 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.

[0003] Access to a main memory is used as an example. Usually, a single transistor is used as a storage unit, N*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 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

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 when sharing memory 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:

[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, 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, 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 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

as a low demand and intensive type.

[0011] With reference to the first 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 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:

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 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

the page coloring-based non-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.

[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

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 non-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.

[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 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 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.

[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 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 page coloring-based partitioning processing. In this way, a collaborative partitioning policy between the 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 memory 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 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.

[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 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;

corresponding page coloring-based partitioning processing.

[0033] According to the memory resource optimization method provided in this 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 page 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 memory resource can be reduced and even eliminated, thereby improving overall performance of a computer.

[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:

[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 program is decreased by one level in each adjustment, until the capacity of 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 in an adjustment process is monitored, and the variation is used as the performance data of each program.

[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 capacity of the LLC resource allocated to each program is decreased to 1/8 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 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 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.

[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 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 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

last level cache swing (LLC Swing, LLCS); the foregoing low demand type 033 may specifically include: core cache fitting (Core Cache Fitting, CCF) 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 non-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 14^{th} and 15^{th} 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, the 16^{th}, 17^{th} and 18^{th} index bits of the LLC and the 21^{th} and 22^{th} index bits 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 14^{th} and 15^{th} bits, the LLC may be partitioned into 4 equal portions,

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 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 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 page 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 memory 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:

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 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

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 non-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.

[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 page 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 memory 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:

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 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.

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 non-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 page 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 memory 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

CLAIMS

1. A memory resource optimization method, wherein the method comprises:

acquiring (S101) performance data of each program in a working set;

categorizing (S101) 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 an allocated last level cache LLC resource;

selecting (S102), 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 an LLC and a dynamic random access memory bank DRAM Bank; and

writing (S103) 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 step of acquiring (S101) performance data of each program in a working set comprises:

partitioning the LLC resource into N portions, taking 1/N of a maximum capacity of the LLC resource as one level, allocating the maximum 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, 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 step of categorizing (S101) 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. 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 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, 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 O-bits are not used, and the page coloring-based non-collaborative partitioning policy comprises:

a Cache-Only policy, in which page 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; and

the step of 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 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 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 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, wherein the apparatus comprises:

a data collecting unit (0011), configured to acquire performance data of each program in a working set;

a categorizing unit (0012), configured to 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 performance indicator of each program varies with a capacity 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, 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 the 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, wherein the operating system kernel performs corresponding page coloring-based partitioning processing.

9. The apparatus according to claim 8, wherein the data collecting unit (0011) is specifically configured to:

partition the LLC resource into N portions, 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 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, 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 the categorizing unit (0012) is 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 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;

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.

12. The apparatus according to claim 11, wherein the decision-making unit (002) is specifically 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 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 processor.

13. The apparatus according to claim 10, 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 comprises:

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; and

the decision-making unit 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.

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 decision-making unit (002) 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.

15. A computer program product, comprising a computer readable storage medium storing program code including executable instructions, which, when executed on a computer, causes the computer to perform the method according to any one of claims 1 to 7.