Appeal decision

Appeal No. 2019-13265

Appellant	Google LLC
Patent Attorney	FUKAMI PATENT OFFICE, P. C.

The case of appeal against the examiner's decision of refusal of Japanese Patent Application No. 2017-156543, entitled "HYBRID MEMORY MANAGEMENT" [the application published on February 15, 2018, Japanese Unexamined Patent Application Publication No. 2018-26136] has resulted in the following appeal decision.

Conclusion

The appeal of the case was groundless.

Reason

No. 1 History of the procedures

The present application was filed on August 14, 2017 (priority claim under the Paris Convention received by the foreign receiving office on August 12, 2016 (hereinafter referred to as "Priority date", United States)) as an application in foreign language. A translation of the document in foreign language was submitted on September 19, 2017. A notice of reasons for refusal was issued on August 29, 2018. A written opinion and a written amendment were submitted on December 4, 2018, which is within the designated period thereof. An examiner's decision of refusal (hereinafter referred to as "Examiner's decision") was issued on May 31, 2019. Against this, an appeal against the examiner's decision of refusal was made on October 3, 2019.

No. 2 Regarding the Invention

The inventions according to claims of the present application are as specified by the matters recited in Claims 1 to 27 of the scope of claims amended by the written amendment submitted on December 4, 2018. The invention according to Claim 1 (hereinafter referred to as "the Invention") is as follows and is recited in Claim 1.

"[Claim 1]

A computer-implemented method comprising:

determining, with a processor, whether an access bit is set for each page table entry of a page table based on a scan of the page table with a page table scanner, the access bit indicating whether a page associated with the page table entry was accessed in a last scan period;

incrementing a count for each page in response to determining that the access bit is not set for the page table entry associated with the page;

after determining, with the processor, whether the access bit is set for each page table entry, resetting the access bit;

receiving a request to access a first page of data stored in a main memory;

initiating a page fault based on determining that the first page of data is not stored in the main memory; and

servicing the page fault using a direct memory access (DMA) engine,

wherein servicing the page fault using a DMA engine comprises: in response to determining the main memory cannot receive a page transfer, determining a particular page based on the count for each page and releasing the particular page into the secondary memory, and transferring the first page of data from the secondary memory to the main memory at the location of the released particular page."

No. 3 Reasons for refusal stated in the examiner's decision

The reasons for refusal stated in the Examiner's decision are as follows: The inventions according to Claims 1 to 27 of this application could have been easily made by a person ordinarily skilled in the art of the invention before the priority date on the basis of inventions described in the following Cited Document 1 or 2 which were distributed or made publicly available through an electric telecommunication line prior to the priority date of the application, or technical matters described in Cited Documents 2 to 6. Thus, the Appellant should not be granted a patent under the provisions of Article 29(2) of the Patent Act.

Cited Document 1: Japanese Unexamined Patent Application Publication No. H06-083713

Cited Document 2: Specification of U.S. Patent No. 6029224

Cited Document 3: Japanese Unexamined Patent Application Publication No.

H04-007653

Cited Document 4: Japanese Unexamined Patent Application Publication No. S61-059557

Cited Document 5: Japanese Unexamined Patent Application Publication No. 2011-165093

Cited Document 6: Japanese Unexamined Patent Application Publication No. 2005-216053

No. 4 Description in the Cited Documents and Cited Invention

1 Cited Document 2

(1) The specification of U.S. Patent No. 6029224 (hereinafter referred to as "Cited Document 2") which is a cited document distributed or made publicly available through an electric telecommunication line prior to the priority date of the application and cited as Cited Document 2 in the notice of reasons for refusal dated August 29, 2018, which are reasons for refusal stated in the Examiner's decision, includes the following description with drawings. (Note by the body: The underlines were added by the body for reference.)

A "1. Field of the Invention

This invention relates generally to computer processor memory storage systems and more particularly to self-managing memory systems that vertically slice the storage hierarchy for increased virtual memory giving the appearance of an extended secondary cache to the processor." (Column 1, lines 5 to 10)

B "In the case of a TLB miss, the CPU 12 sends a virtual address with a get_pt_entry request to the memory system 30 that has a page table 64 for that process. The memory system 30 fielding that process checks the corresponding page table entries 66 and 68 to determine if the page is in main memory, and if so, the entry is returned to the CPU 12 as data. If the page table entry 68 indicates that the page is not in main memory, a 'wait' message is returned to the CPU 12 by the memory and disk control 76, since data have to be read from the disk 44. At this point the CPU may decide to switch to another task. The programmable system control 70 instructs the memory and disk control 76 to read from the disk 44, updates the page table entry 68, and sends an 'access done' message to the CPU 12. The CPU 12 can then update the TLB location 54 and resume the paused process. The programmable system control 70 handles all the page fault handling, management of the page table 64, cache control 60, and disk control 76. This immediate read/write control of the disk control 76 allows the system to perform a high level of power management of the memory resources. The disk control 76 will only start the disk 44 when needed and be able to keep the start-up time as transparent as possible to the application." (Column 3, lines 43 to 65)

C "The mapping in the page table 64 from virtual to physical is fully associative. Each program that is run has its own page table 64 and page table register 56 that points to the current page table 54 in use. A valid bit in the page table entry 66 indicates that the page is in primary memory. If the bit is off for a referenced page, a page fault occurs. If the bit is on, the entry contains the physical page number. No tags are required, since there is an entry for every virtual page. For example, 1M page table entries and a 4 KB page size would be required for a total of 4 GBytes of virtual space for each process to handle a 20 bit virtual paged number and a 12 bit offset. The page table, program counter, register, and page register specify the program state and therefore must be saved upon a context switch to a different process." (Column 4, lines 7 to 20)

D "In a page fault case, the memory system must now look up the page table using the virtual address to find the location of the referenced page on the disk and choose a physical page to replace. It informs the CPU via a return 'wait' message (124) that the miss is in fact a page fault and will require a disk access. If the physical page is dirty; i.e., if the memory has already been modified, it must be written out to disk first. Then the memory system starts a read for the referenced page from the disk (126) in to the chosen physical page.
In processing the page fault, the present invention must decide where to put the page it reads from disk into main memory. This is done using a Least Recently Used (LRU) scheme. A reference bit is added to the page table entry. When this page gets accessed, the reference bit gets set. The memory system periodically checks and clears these bits, maintaining a count of the bits set; i.e., pages being used. The pages with the low counts would then be subject to the disk reads." (Column 5, lines 39 to 57)

(2) According to the description in (1), it is recognized that Cited Document 2 describes the following invention (hereinafter referred to as "Cited Invention").

"An invention relating to computer processor memory storage systems, wherein

if the page table entry 68 indicates that the page is not in main memory, a 'wait' message is returned to the CPU 12 by the memory and disk control 76, since data have to be read from the disk 44, the CPU may decide to switch to another task, the programmable system control 70 instructs the memory and disk control 76 to read from the disk 44, and updates the page table entry 68,

the programmable system control 70 handles all the page fault handling, management of the page table 64, cache control 60, and disk control 76,

each program that is run has its own page table 64, a valid bit in the page table entry 66 indicates that the page is in primary memory, if the bit is off for a referenced page, a page fault occurs,

in a page fault case, the memory system must now look up the page table using the virtual address to find the location of the referenced page on the disk and choose a physical page to replace, if the physical page is dirty; i.e., if the memory has already been modified, it must be written out to disk first,

in processing the page fault, the present invention must decide where to put the page it reads from disk into main memory, this is done using a Least Recently Used (LRU) scheme, a reference bit is added to the page table entry, when this page gets accessed, the reference bit gets set, the memory system periodically checks and clears these bits, maintaining a count of the bits set; i.e., pages being used, the pages with the low counts would then be subject to the disk reads."

2 Reference 1

(1) The specification of U. S. Patent Application Publication No. 2011/0271070 (hereinafter referred to as "Reference 1"), which was distributed or made publicly available through an electric telecommunication line prior to the priority date of the application, includes the following description with drawings. (Note by the body: The underlines were added by the body for reference.)

"[0012] A memory scanning system may be used to manage the location of items stored in memory. The memory scanning system may scan items in memory by periodically scanning metadata about the memory items to determine which memory pages have been accessed since the last scan. The periodic scanning may be used to classify the pages based on the frequency at which the pages were accessed over the time period of the scans, as well as other criteria."

"[0050] The scan engine 148 may scan memory pages by scanning those pages associated with processes. As part of an operating system, each process may be allocated memory and given a set of virtual addresses for the allocated memory. The process may address the memory pages using the virtual memory addresses, while a page table may map those memory pages to a physical address. Each process may have a working set list that contains the virtual addresses of the memory pages assigned to the process.

[0051] The scan engine 148 may use the working set list for each process to scan each memory page associated with the process. During the scan, the scan engine may update various bitmaps and counters for the memory pages, and then use the bitmaps and counters to determine how frequently the memory page is accessed."

"[0084] Embodiment 400 is an example of a method that may scan a memory page for either read or write access and may keep a separate counter for read and write operations. The scanning mechanism may use a hardware or software bit or bits that may be toggled by hardware or software whenever a memory page is accessed. Some embodiments may include only a single bit that may indicate that an access occurred, but not differentiate between read and write access.

...Omitted...

[0086] Embodiment 400 may use counters that count up for each scan where an access has occurred. In many such embodiments, several full scans may be performed in succession, and then the counters may be analyzed to identify those memory pages that are highly used, those that are lightly used, and possibly those memory pages that are used at some intermediate level. After the analysis, the counters may be reset for another analysis pass.

[0087] Other embodiments may use different tracking and counting mechanisms. For example, some embodiments may use counters that increment when a memory page has not been accessed. In some cases, a counter may be set to an initial value then decremented on each successive scan when a memory page is unused or used. The embodiment 400 is merely one mechanism that may be used to scan memory pages."

3 Reference 2

(1) The specification of U. S. Patent Application Publication No. 2016/0055098 (hereinafter referred to as "Reference 2"), which was distributed or made publicly available through an electric telecommunication line prior to the priority date of the application, includes the following description with drawings. (Note by the body: The underlines were added by the body for reference.)

"[0005] The memory manager operates to provide pages associated with a virtual address in physical RAM when those pages are needed. The set of pages in the virtual address space that currently reside in RAM is known as the "working set." When contents of a particular virtual page are needed, but that page is not stored in the RAM, the memory manager may retrieve that data from non-volatile memory and make it available in the RAM. Conversely, when a process requires more memory than is free in the RAM, the memory manager may 'trim' pages from the working set by removing the contents of one or more pages in RAM and ensuring that the contents are stored in the non-volatile storage. Trimming pages frees up space in the RAM for other pages to be brought into RAM.

[0006] To perform paging operations, page tables are used to store a mapping between virtual and physical addresses. In addition to the mapping, page table entries (PTEs) often include a page offset, and information which details use of the page. This information may be in the form of an access bit, which indicates whether the page has been accessed during an interval since the access bit was reset. In addition to an access bit, an age count provides information that details the use of the page. The age count is often stored in a working set list entry (WSLE), and indicates the number of successive intervals during which the page was not accessed. This information is used by the memory manager to determine which pages to trim when the amount of available RAM becomes so limited that the most frequently accessed pages are kept in RAM and the less frequently accessed pages are paged out to make more RAM available. The age count is set by the memory manager through an operation called 'aging.' Aging tracks access history of pages in a working set by periodically checking whether each page has been accessed during an interval since access to the page was last checked. Such a check may be performed by periodically checking whether the access bit for the page is set. If so, the page has been accessed and the age count for the page is reset. The access bit is also reset. If not, the age count for the page may be increased, up to some maximum count established by the number of bits in the WSLE (or PTE) allocated for holding the age count. The age count allows the memory manager to select pages that can be trimmed with low likelihood that performance of the computer system will be impacted by removing from the working set a page that is actively being used."

No. 5 Comparison

The Invention and the Cited Invention are compared below.

1 The Cited Invention, which "relates to computer processor memory storage systems",

specifies processing to be executed in the memory storage systems of the computer processor; i.e., a method. Thus, the Invention and the Cited Invention are identical in "a computer-implemented method" although they are different in the point described later.

2 The "access bit" in the Invention and the "reference bit" in the Cited Invention are both "bits".

The Cited Invention is configured so that "a reference bit is added to the page table entry, when this page gets accessed, the reference bit gets set, the memory system periodically checks and clears these bits, maintaining a count of the bits set; i.e., pages being used". The "memory system" in the Cited Invention is considered to periodically check a page table as to whether a reference bit added to the page table entry has been set or not, and the memory system is also considered to have check means for periodically checking the page table. The "check means" and "periodically checking the page table" correspond to the "page table scanner" and the "scan of the page table" in the Invention, respectively.

In the Cited Invention, it is obvious that it is determined whether the reference bit added to the page entry has been set or not on the basis of a periodic check. The memory storage system of the computer processor relating to the Cited Invention is considered to include determination means for the above determination. Thus, the determination means corresponds to the "processor" in the Invention.

Accordingly, it can be said that the Cited Invention is to determine, with the determination means, whether the reference bit added to the page table entry has been set or not on the basis of a periodic check of the page table with the above check means. Thus, the Invention and the Cited Invention are identical in "including a step of determining, with a processor, whether a bit is set for each page table entry of a page table based on a scan of the page table with a page table scanner", although they are different in the point described later.

3 The Cited Invention is configured so that "a reference bit is added to the page table entry, when this page gets accessed, the reference bit gets set" and configured to "periodically check and clear these bits". The "reference bit" in the Cited Invention is considered to indicate whether a page associated with the page table entry to which the reference bit has been added was accessed or not, and the reference bit is cleared at a periodic check. Thus, it is obvious that the page associated with the page table entry is accessed after the last periodic check.

Accordingly, based on the examination in 2, the "reference bit" in the Cited Invention corresponds to an "access bit" which is a bit that "indicates whether a page associated with the page table entry was accessed in a last scan period" in the Invention. Considering the examination in 2, the Invention and the Cited Invention are identical in "including a step of determining, with a processor, whether an access bit is set for each page table entry of a page table based on a scan of the page table with a page table scanner", although they are different in the point described later.

4 The "count" in the Invention and the "count" in the Cited Invention are identical in counting the number although they are different in the point described later.

The Cited Invention is configured to "periodically check and clear these bits, maintain a count of the bits set; i.e., pages being used". In the Cited Invention, when a reference bit is set in the page table entry as a result of a periodic check, a count of an associated and used page is increased; i.e., incremented, accordingly.

Thus, considering the examination in 3, the Invention and the Cited Invention are identical in including "a step of incrementing a count for each page" although they are different in the point described later.

5 The Cited Invention is configured to "periodically check and clear these bits". Considering the examinations in 2 and 3, the Cited Invention is identical with the Invention in including "a step of after determining, with the processor, whether the access bit is set for each page table entry, resetting the access bit".

6 The "primary memory" in the Cited Invention corresponds to the "main memory" in the Invention.

The Cited Invention is configured so that "each program that is run has its own page table 64, a valid bit in the page table entry 66 indicates that the page is in primary memory, if the bit is off for a referenced page, a page fault occurs". The "referenced page" in the Cited Invention can be in a primary memory. The Invention includes the steps of "receiving a request to access a first page of data stored in a main memory, and

initiating a page fault based on determining that the first page of data is not stored in the main memory". The "first page of data" in the Invention can be stored in a main memory. Thus, the "referenced page" in the Cited Invention corresponds to the "first page of data" in the Invention.

In the Cited Invention, as premises for the above processing, it is obvious that an access request is made to access the referenced page which is considered to be located in the primary memory from an executed program. Thus, it can be said that the Cited Invention is configured to receive an access request to access a referenced page stored in a primary memory. The Invention and the Cited Invention are identical in including "a step of receiving a request to access a first page of data stored in a main memory".

The "valid bit in the page table entry 66" in the Cited Invention "indicates that the page is in primary memory". Thus, it can be said that the description, "if the bit is off for a referenced page", indicates that the referenced page is not in a primary memory. In the Cited Invention, when the reference page is not in the primary memory, a page fault occurs. Therefore, the Invention and the Cited Invention are identical in including "a step of initiating a page fault based on determining that the first page of data is not stored in the main memory" although they are different in the point described later.

7 The direct memory access (DMA) is a mechanism, as a matter of common general technical knowledge on the priority date of the application, for directly transferring data between memories or between a memory and an I/O device without using processing of a CPU.

The Cited Invention is configured so that "if the page table entry 68 indicates that the page is not in main memory, a 'wait' message is returned to the CPU 12 by the memory and disk control 76, since data have to be read from the disk 44, the CPU may decide to switch to another task, the programmable system control 70 instructs the memory and disk control 76 to read from the disk 44, and updates the page table entry 68". It can be said that the programmable system control 70 and the memory and disk control 76 read data from the disk 44 without using CPU processing. Thus, the combination of the "programmable system control 70" and the "memory and disk control 76" in the Cited Invention corresponds, according to the above common general technical knowledge, to the "direct memory access (DMA)" in the Invention, obviously.

The "programmable system control 70" in the Cited Invention is to "handle all the page fault handling, management of the page table 64, cache control 60, and disk control 76". It can be said that the Cited Invention is configured to perform page fault processing by using the programmable system control 70 and disk control. Thus, the Invention and the Cited Invention are identical in including "the step of servicing the page fault using a direct memory access (DMA) engine" although they are different in the point described later.

8 The "disk" in the Cited Invention corresponds to the "secondary memory" in the

Invention.

In the Cited Invention, "in processing the page fault, the present invention must decide where to put the page it reads from disk into main memory, this is done using a Least Recently Used (LRU) scheme, a reference bit is added to the page table entry, when this page gets accessed, the reference bit gets set, the memory system periodically checks and clears these bits, maintaining a count of the bits set; i.e., pages being used, the pages with the low counts would then be subject to the disk reads". The "pages with the low counts" in the Cited Invention are obviously determined based on a count for each page. Thus, the "particular page" in the Invention and the "pages with the low counts" in the Cited Invention are identical in being "determined" "based on the count for each page".

In addition, the Cited Invention is configured so that "if the physical page is dirty; i.e., if the memory has already been modified, it must be written out to disk first". In the Cited Invention, when the pages with the low counts are dirty, it is considered that the pages with the low counts have been written out to disk, naturally.

In processing the page fault in the Cited Invention, in "deciding where to put the page it reads from disk into main memory", it is obvious from common general technical knowledge that it is required to "do using a Least Recently Used (LRU) scheme" only when there is no available page table entry and pages cannot be read from disk into main memory unless a page corresponding to one page table entry is released.

In light of the above, in "processing the page fault" in the Cited Invention, it can be said that when pages cannot be read from disk into main memory unless a page corresponding to one page table entry is released, pages with the low counts are determined based on the count for each page, the contents of the pages with the low counts are written out to the disk and released, and the pages to be read from the disk to the main memory are arranged in the location of the pages with the low counts which have been written out and released. Thus, the Invention and the Cited Invention are identical in the following point:

"the step of servicing the page fault using a DMA engine comprises:

in response to determining that the main memory cannot receive a page transfer, determining a particular page based on the count for each page and releasing the particular page into the secondary memory, and transferring the first page of data from the secondary memory to the main memory at the location of the released particular page",

although they are different in the point described later.

9 As above, the Invention and the Cited Invention have the following corresponding feature and different feature.

(Corresponding Feature)

"A computer-implemented method comprising:

determining, with a processor, whether an access bit is set for each page table entry of a page table based on a scan of the page table with a page table scanner, the access bit indicating whether a page associated with the page table entry was accessed in a last scan period;

incrementing a count for each page;

after determining, with the processor, whether the access bit is set for each page table entry, resetting the access bit;

receiving a request to access a first page of data stored in a main memory;

initiating a page fault based on determining that the first page of data is not stored in the main memory; and

servicing the page fault using a direct memory access (DMA) engine,

wherein servicing the page fault using a DMA engine comprises: in response to determining the main memory cannot receive a page transfer, determining a particular page based on the count for each page and releasing the particular page into the secondary memory, and transferring the first page of data from the secondary memory to the main memory at the location of the released particular page."

(Different Feature 1)

The Invention is configured to increment a "count" in response to determining that the access bit is not set for the page table entry associated with the page. The Cited Invention is configured to increment a "count" when a reference bit is set in the page table entry as a result of a periodic check.

No. 6 Judgment

1 Regarding Different Feature 1

For example, as described in References 1 and 2, in a memory system, as a method of grasping access frequency to each page, a method of incrementing a count when there is an access to each page; i.e., when the access bit is set, and a method of incrementing a count when there is no access to each page; i.e., when the access bit is not set by periodically checking the access bit indicating that each page has been accessed, had been a conventional technique for a person skilled in the art at the priority date of the application.

The Cited Invention is configured to specify a page which is not frequently accessed based on a count and to increment the count when a reference bit is set in a page table entry as a result of a periodic check for grasping the access frequency. According to the above conventional technique of a person skilled in the art, a configuration of incrementing a count when a reference bit is not set in a page table entry as a result of a periodic check for grasping access frequency based on the count, instead of the above configuration; in the Cited Invention, is only a design matter which could be selected by a person skilled in the art having ordinary creativity when appropriate.

Therefore, considering the technical level on the priority date of the application, the Invention could be easily made by a person skilled in the art on the basis of the Cited Invention.

2 Effect and Appellant's allegation

The Appellant alleges in the written appeal "No. 2 4." that the Invention has inventive step in producing an effect of "determining a page which has not been accessed for a longer time even when there is no access to a plurality of pages, by incrementing a count of the pages not accessed", with the configuration (Written appeal "No. 2 1." "Configuration C") relating to the different feature.

However, it is obvious that the effect of determining a page which has not been accessed for a longer time out of a plurality of pages is produced by incrementing a count of the pages not accessed. Incrementing a count of pages not accessed is a conventional technique for a person skilled in the art. As examined in 1, a person skilled in the art could have easily employed the configuration in the Cited Invention.

Therefore, the above effect is within a scope which can be predicted by a person skilled in the art and it is not remarkable. The Appellant's allegation cannot be accepted.

No. 7 Closing

As described above, the Appellant should not be granted a patent for the Invention under the provisions of Article 29(2) of the Patent Act. The present application should be rejected without examining inventions according to other claims.

Therefore, the appeal decision shall be made as described in the conclusion.

July 29, 2020

Chief administrative judge:	TANAKA, Hideto
Administrative judge:	HAMANAKA, Nobuyuki
Administrative judge:	KOBAYASHI, Hidekazu