

## 審決

不服 2019-13265

(省略)

請求人 グーグルエルエルシー

(省略)

代理人弁理士 特許業務法人深見特許事務所

特願 2017-156543 「ハイブリッドメモリ管理」拒絶査定不服審判事件〔平成 30 年 2 月 15 日出願公開，特開 2018-26136〕について，次のとおり審決する。

### 結論

本件審判の請求は，成り立たない。

### 理由

#### 第 1 手続の経緯

本願は，平成 29 年 8 月 14 日（パリ条約による優先権主張外国庁受理 2016 年 8 月 12 日（以下，「優先日」という。）），アメリカ合衆国）を出願日とする外国語書面出願であって，平成 29 年 9 月 19 日付けで外国語書面の翻訳文が提出され，平成 30 年 8 月 29 日付けで拒絶理由が通知され，その指定期間内である同年 12 月 4 日に意見書及び手続補正書が提出されたが，令和 1 年 5 月 31 日付けで拒絶査定（以下，「原査定」という。）がなされ，これに対し，同年 10 月 3 日に拒絶査定不服審判が請求されたものである。

#### 第 2 本願発明について

本願の請求項に係る発明は，平成 30 年 12 月 4 日にされた手続補正により補正された特許請求の範囲の請求項 1 ないし 27 に記載された事項により特定されるものであるところ，その請求項 1 に係る発明（以下，「本願発明」という。）は，その請求項 1 に記載された以下のとおりのものである。

#### 「【請求項 1】

コンピュータによって実現される方法であって、

ページテーブルスキャナでのページテーブルのスキャンに基づいて、アクセスビットが前記ページテーブルの各ページテーブルエントリ毎に設定されているかどうかを、プロセッサで判断するステップを含み、前記アクセスビットは、前記ページテーブルエントリに関連付けられたページが最後のスキャン期間に

においてアクセスされたかどうかを示しており、前記方法はさらに、

前記アクセスビットが前記ページに関連付けられた前記ページテーブルエントリ毎に設定されていないと判断したことに応じて、各ページ毎にカウントをインクリメントするステップと、

前記アクセスビットが各ページテーブルエントリ毎に設定されているかどうかを前記プロセッサで判断した後、前記アクセスビットをリセットするステップと、

メインメモリに格納されているデータの第1のページにアクセスするようにとの要求を受取るステップと、

前記データの第1のページが前記メインメモリに格納されていないと判断したことに基づいて、ページフォルトを開始するステップと、

ダイレクトメモリアクセス（DMA）エンジンを用いて前記ページフォルトを処理するステップとを含み、

DMAエンジンを用いて前記ページフォルトを処理する前記ステップは、前記メインメモリがページ転送を受取ることができないと判断したことに応じて、各ページ毎の前記カウントに基づいて特定のページを決定し、二次メモリに前記特定のページをリリースし、前記リリースされた特定のページ的位置において、前記データの第1のページを前記二次メモリから前記メインメモリに転送するステップを含む、方法。」

### 第3 原査定の拒絶の理由

原査定の拒絶の理由は、この出願の請求項1ないし27に係る発明は、本願の優先日前に頒布された又は電気通信回線を通じて公衆に利用可能となった下記の引用文献1又は2に記載された発明、及び、引用文献2-6に記載された技術的事項に基づいて、その優先日前にその発明の属する技術の分野における通常の知識を有する者が容易に発明をすることができたものであるから、特許法第29条第2項の規定により特許を受けることができない、というものである。

引用文献1：特開平06-083713号公報

引用文献2：米国特許第6029224号明細書

引用文献3：特開平04-007653号公報

引用文献4：特開昭61-059557号公報

引用文献5：特開2011-165093号公報

引用文献6：特開2005-216053号公報

### 第4 引用文献の記載及び引用発明

#### 1 引用文献2

(1) 本願の優先日前に頒布された又は電気通信回線を通じて公衆に利用可能となった引用文献であって、原審の拒絶の査定の理由である上記平成30年8

月 29 日付けの拒絶理由通知において引用文献 2 として引用された米国特許第 6029224 号明細書（以下、「引用文献 2」という。）には、図面とともに次の記載がある。（当審注：下線は、参考のために当審で付与したものである。）

ア 「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.」

（第 1 欄第 5～10 行）

（当審訳：1. 発明の分野

この発明は、概して、コンピュータプロセッサのメモリストレージシステムに、より特には、プロセッサに対して拡張二次キャッシュの外観を与える増加された仮想メモリのストレージ階層を垂直的にスライスする自己管理メモリスシステムに関する。）

イ 「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 entry 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 has 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.」 （第 3 欄第 43～65 行）

（当審訳：TLBミスの場合、CPU12は、仮想アドレスを `get_pt_entry` リクエストとともに、そのプロセスのためのページテーブル64を有するメモリスシステム30に送る。そのプロセスをフィールドイングするメモリスシステム30は、対応するページテーブルエントリ66及び68をチェックし、ページがメインメモリにあるか決定し、ある場合、そのエントリがデータ

としてCPU 12に返される。ページテーブルエントリ68がページはメインメモリにないことを示す場合、データがディスク44から読み込まれる必要があるため、「待機」メッセージが、メモリ・ディスクコントロール76によってCPU 12に返される。この時点で、CPUは、他のタスクに切り替えることを決定してもよい。プログラム可能システムコントロール70は、メモリ・ディスクコントロール76にディスク44から読み出すことを指示し、ページテーブルエントリ68を更新し、「アクセス完了」メッセージをCPU 12に送る。CPU 12は、すると、TLBロケーション54を更新し、休止中のプロセスを再開することができる。プログラム可能システムコントロール70は、全てのページフォルト処理、ページテーブル64、キャッシュコントロール60及びディスクコントロール76の管理を行う。この即座に行われるディスクコントロール76のリード/ライト制御により、システムは、高レベルのメモリ資源の電力管理を行うことができる。ディスクコントロール76は、必要がある場合にのみディスク44を起動し、アプリケーションに対して起動時間を可能な限り透過的にすることができる。)

ウ 「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 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.」(第4欄第7~20行)

(当審訳：ページテーブル64における仮想から物理へのマッピングは、全く連想的である。各実行されたプログラムは、自身のページテーブル64と、使用されている現行のページテーブル54を指すページテーブルレジスタ56を有する。ページテーブルエントリ66の有効ビットは、ページが、プライマリメモリにあることを示す。参照されるページに関しこのビットがオフの状態の場合、ページフォルトが起こる。このビットがオンの状態の場合、エントリは物理ページ番号を含む。全ての仮想ページに対してエントリが存在するが、タグは必要とされない。例えば、ページテーブルエントリが1Mでページサイズが4KBであるとすると、各プロセスが20ビットの仮想ページ番号と12ビットのオフセットを扱うためには、全部で4Gバイトの仮想空間が必要となる。ページテーブル、プログラムカウンタ、レジスタ、及び、ページレジスタは、プログラムの状態を示し、したがって、異なるプロセスへのコンテキストスイッチの際には保存されなければならない。)

エ 「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. 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 in 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.」 (第5欄第39～57行)

(当審訳：ページフォルトの場合、メモリシステムは、参照されるページのディスク上の位置を見つけ、置き換える物理ページを選択するために、仮想アドレスを用いるページテーブルを検索しなければならない。メモリシステムは、CPUに、「待機」メッセージ(124)を返すことにより、ミスは実際にはページフォルトであることを知らせ、ディスクアクセスを要求する。物理ページがダーティである、すなわち、メモリが既に変更されている場合、まずディスクに書き出されなければならない。それから、メモリシステムは、ディスク(126)から選択した物理ページへの参照されるページの読み込みを開始する。)

ページフォルトの処理において、本発明は、ディスクからメインメモリに読み込むページをどこに配置するか決定しなければならない。これは、最も過去に使用された(LRU)スキームを用いて行われる。参照ビットがページテーブルエントリに追加される。このページがアクセスされると、参照ビットがセットされる。メモリシステムは、定期的に、これらのビットをチェックしてクリアし、セットされたビット、すなわち、使用されたページのカウンタ値を維持する。低いカウンタ値を有するページが、ディスク読み込みの対象となる。)

(2) 上記(1)の記載を踏まえると、引用文献2には、以下の発明(以下、「引用発明」という。)が記載されているものと認められる。

「コンピュータプロセッサのメモリストレージシステムに関する発明であって、ページテーブルエントリ68がページはメインメモリにないことを示す場合、データがディスク44から読み込まれる必要があるため、「待機」メッセージが、メモリ・ディスクコントロール76によってCPU12に返され、CPUは、他のタスクに切り替えることを決定し、プログラム可能システムコントロ

ール70は、メモリ・ディスクコントロール76にディスク44から読み出すことを指示し、ページテーブルエントリ68を更新し、

プログラム可能システムコントロール70は、全てのページフォルト処理、ページテーブル64、キャッシュコントロール60及びディスクコントロール76の管理を行い、

各実行されたプログラムは、自身のページテーブル64を有し、ページテーブルエントリ66の有効ビットは、ページが、プライマリメモリにあることを示し、参照されるページに関しこのビットがオフの状態の場合、ページフォルトが起こり、

ページフォルトの場合、メモリシステムは、参照されるページのディスク上の位置を見つけ、置き換える物理ページを選択するために、仮想アドレスを用いるページテーブルを検索しなければならず、物理ページがダーティである、すなわち、メモリが既に変更されている場合、まずディスクに書き出されなければならず、

ページフォルトの処理において、ディスクからメインメモリに読み込むページをどこに配置するか決定しなければならず、これは、最も過去に使用された(LRU)スキームを用いて行われ、参照ビットがページテーブルエントリに追加され、このページがアクセスされると、参照ビットがセットされ、メモリシステムは、定期的に、これらのビットをチェックしてクリアし、セットされたビット、すなわち、使用されたページのカウンタ値を維持し、低いカウンタ値を有するページが、ディスク読み込みの対象となる  
発明。」

## 2 参考文献1

(1) 本願の優先日前に頒布された又は電気通信回線を通じて公衆に利用可能となった米国特許出願公開第2011/0271070号明細書(以下、「参考文献1」という。)には、図面とともに次の記載がある。(当審注：下線は、参考のために当審で付与したものである。)

「[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 the pages were accessed over the time period of the scans, as well as other criteria.」

(当審訳：[0012] メモリスキャニングシステムは、メモリに格納されたアイテムの位置を管理するために用いられてよい。メモリスキャニングシステムは、メモリのアイテムに関するメタデータを定期的にスキャンすることにより、どのメモリページが最後のスキャン以降にアクセスされたかを決定するために、メモリ内のアイテムをスキャンしてよい。定期的なスキャニングは、スキャンの期間にわたってページがアクセスされる頻度、及び、その他の基準に基づい

てページを分類するために用いられてよい。)

「[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.」

(当審訳： [0050] スキャンエンジン148は、プロセスに関連付けられたページをスキャンすることにより、メモリページをスキャンしてよい。オペレーティングシステムの一部として、各プロセスは、メモリが割り当てられ、割り当てられたメモリのための仮想アドレスのセットが与えられてよい。プロセスは、仮想メモリアドレスを用いるメモリページをアドレスしてよく、ページテーブルは、これらのメモリページを物理アドレスにマップしてよい。各プロセスは、自身に割り当てられたメモリページの仮想アドレスを含むワーキングセットリストを有してよい。

[0051] スキャンエンジン148は、プロセスに関連付けられた各メモリページをスキャンするために、各プロセスのワーキングセットリストを使用してよい。スキャンの間、スキャンエンジンは、メモリページのための様々なビットマップ及びカウンタを更新し、それからメモリページがどれくらいの頻度でアクセスされるかを決定するためにビットマップ及びカウンタを使用してよい。)

「[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.

．．．中略．．．

[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.」

(当審訳： [0084] 実施例400は、リードアクセス又はライトアクセスに関しメモリページをスキャンし、リード操作及びライト操作に別個のカウンタを維持してよい方法の例である。 スキャンメカニズムは、メモリページがアクセスされるときはいつでもハードウェアまたはソフトウェアによって切り替えられる、ハードウェアビットまたはソフトウェアビットを使用してよい。ある実施例は、アクセスがあったことを示すが、リードアクセスとライトアクセスを区別しなくてよい単一のビットのみを含んでよい。

・・・中略・・・

[0086] 実施例400は、アクセスが起こった場合、スキャンごとにカウンタアップするカウンタを使用してよい。 そうした実施例の多くにおいては、何回かの完全スキャンが連続して行われてよく、そして、よく使用されるメモリページ、少しだけ使用されるメモリページ、及び、もしかすると、ある中間のレベルで使用されるメモリページを特定するために、カウンタが解析されてよい。解析の後、ほかの解析パスのためにカウンタはリセットされてよい。

[0087] 他の実施例は、異なる追跡及びカウンティングのメカニズムを用いてよい。例えば、ある実施例は、メモリページがアクセスされなかった場合にインクリメントされるカウンタを使用してよい。 ある事例では、カウンタは、初期値に設定され、それからメモリページが使用されない又は使用された場合に、連続するスキャンの各スキャンにおいてデクリメントされてよい。実施例400は、メモリページをスキャンするために用いられる単なる一メカニズムである。)

### 3 参考文献2

(1) 本願の優先日前に頒布された又は電気通信回線を通じて公衆に利用可能となった米国特許出願公開第2016/0055098号明細書(以下、「参考文献2」という。)には、図面とともに次の記載がある。(当審注：下線は、参考のために当審で付与したものである。)

「[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 is 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 is 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.]

（当審訳： [0005] メモリマネージャは、仮想アドレスに関連付けられたページを、必要とされる場合に、物理RAMに提供するために動作する。RAM内に現在存在している仮想アドレス空間のページのセットは、“ワーキングセット”として知られる。特定の仮想ページの内容が必要とされるが、そのページがRAMに格納されていない場合、メモリマネージャは、そのデータを不揮発性メモリから抽出し、RAM内において利用可能としてよい。反対に、プロセスがRAM内において、空いている以上のメモリを要求する場合、メモリマネージャは、RAM内の一つまたはそれ以上のページの内容を除去し、その内容が不揮発性ストレージに格納されることを確実にすることにより、ワーキング

セットからページを“トリム”してよい。ページをトリミングすることにより、他のページがRAMに読み込まれるためにRAM内の空間が解放される。

[0006] ページング操作を行うために、仮想アドレスと物理アドレスの間のマッピングを記憶するためにページテーブルが使用される。マッピングに加えて、ページテーブルエントリ（PTE）は、ページオフセット、及び、ページの使用に関する情報をしばしば含む。この情報は、

アクセスビットがリセットされて以降の1回のインターバルの間にページがアクセスされたかどうかを示すアクセスビットの形態であってよい。アクセスビットに加えて、年齢カウンタが、ページの使用に関する情報を提供する。年齢カウンタは、しばしば、ワーキングセットリストエントリ（WSLE）に格納され、ページがアクセスされなかった、連続するインターバルの数を示す。この情報は、利用可能なRAMの量が制限されているため、最も頻繁にアクセスされるページがRAM内に維持され、アクセス頻度がより少ないページは、より多くのRAMを利用可能とするためにページアウトされる場合に、どのページをトリムすべきか決定するために、メモリマネージャによって使用される。年齢カウンタは、“エイジング”と呼ばれる操作によりメモリマネージャによってセットされる。エイジングは、ページへのアクセスが最後にチェックされてから1回のインターバルの間に各ページがアクセスされたかどうかを定期的にチェックすることにより、ワーキングセット内のページのアクセス履歴を追跡する。そのようなチェックは、そのページにアクセスビットがセットされているかどうかを定期的にチェックすることによって行われてよい。アクセスビットがセットされていれば、ページはアクセスされており、そのページの年齢カウンタはリセットされる。アクセスビットもリセットされる。アクセスビットがセットされていなければ、年齢カウンタを保持するために割り当てられたWSLE（またはPTE）内のビット数によって定まるある最大値まで、そのページの年齢カウンタは増加される。年齢カウンタにより、メモリマネージャは、活発に使用されているページをワーキングセットから除去することによりコンピュータシステムのパフォーマンスが影響を受ける可能性を小さくしたまま、トリム可能なページを選択することが可能となる。）

## 第5 対比

本願発明と引用発明とを対比する。

1 引用発明は、「コンピュータプロセッサのメモリストレージシステムに関する」ものであり、当該コンピュータプロセッサのメモリストレージシステムにおいて行われる処理、すなわち、方法を特定するものであるから、本願発明と引用発明は、後記する点で相違するものの、「コンピュータによって実現される方法」である点において一致する。

2 本願発明の「アクセスビット」と引用発明の「参照ビット」とは、「ビット」である点において共通する。

また、引用発明は、「参照ビットがページテーブルエントリに追加され、こ

のページがアクセスされると、参照ビットがセットされ、メモリシステムは、定期的に、これらのビットをチェックしてクリアし、セットされたビット、すなわち、使用されたページのカウント値を維持」するものであるところ、引用発明の「メモリシステム」は、ページテーブルエントリに追加された参照ビットがセットされているかどうかについてページテーブルに対する定期的なチェックを行っているといえ、また、当該ページテーブルに対する定期的なチェックを行うためのチェック手段を当然に備えていると認められ、当該「チェック手段」及び「ページテーブルに対する定期的なチェック」が、本願発明の「ページテーブルスキャナ」及び「ページテーブルのスキャン」に相当する。

さらに、引用発明において、上記定期的なチェックに基づいて、ページエントリに追加された参照ビットがセットされているかどうかを判断していることは明らかであり、引用発明に係るコンピュータプロセッサのメモリストレージシステムは当該判断を行うための判断手段を備えているものと認められ、当該判断手段が、本願発明の「プロセッサ」に相当する。

してみると、引用発明は、上記チェック手段によるページテーブルに対する定期的なチェックに基づいて、ページテーブルエントリに追加された参照ビットがセットされているかどうかを、上記判断手段で判断するものであるといえるから、本願発明と引用発明は、後記する点で相違するものの、「ページテーブルスキャナでのページテーブルのスキャンに基づいて、ビットが前記ページテーブルの各ページテーブルエントリ毎に設定されているかどうかを、プロセッサで判断するステップを含」むものである点において一致する。

3 引用発明は、「参照ビットがページテーブルエントリに追加され、このページがアクセスされると、参照ビットがセットされ」、また、「定期的に、これらのビットをチェックしてクリア」するものであるところ、引用発明の「参照ビット」は、自身が追加されたページテーブルエントリに対応するページがアクセスされたかどうかを示しているものであるといえ、また、定期的なチェックの際にクリアされるものであることから、当該ページテーブルエントリに対応するページへのアクセスは、最後の定期的なチェックが行われた後についてのものであることが明らかである。

したがって、引用発明の「参照ビット」は、上記2での検討も踏まえると、本願発明の「前記ページテーブルエントリに関連付けられたページが最後のスキャン期間においてアクセスされたかどうかを示して」いるビットである「アクセスビット」に相当する。また、本願発明と引用発明は、上記2での検討も踏まえると、後記する点で相違するものの、「ページテーブルスキャナでのページテーブルのスキャンに基づいて、アクセスビットが前記ページテーブルの各ページテーブルエントリ毎に設定されているかどうかを、プロセッサで判断するステップを含」むものである点において一致する。

4 本願発明の「カウント」と引用発明の「カウント値」とは、後記する点で相違するものの、数を数えるものである点において共通する。

また、引用発明は、「定期的に、これらのビットをチェックしてクリアし、

セットされたビット，すなわち，使用されたページのカウンタ値を維持」するものであるところ，引用発明においては，定期的なチェックの結果，参照ビットがページテーブルエントリにセットされている場合，対応する使用されたページのカウンタ値を増やす，すなわち，インクリメントしているといえる。

よって，本願発明と引用発明とは，上記3での検討も踏まえると，後記する点で相違するものの，「各ページ毎にカウンタをインクリメントするステップ」を含むものである点において，一致する。

5 引用発明は，「定期的に，これらのビットをチェックしてクリア」するものであるから，上記2及び3での検討も踏まえると，「前記アクセスビットが各ページテーブルエントリ毎に設定されているかどうかを前記プロセッサで判断した後，前記アクセスビットをリセットするステップ」を含むものである点において，本願発明に一致する。

6 引用発明の「プライマリメモリ」は，本願発明の「メインメモリ」に相当する。

また，引用発明は，「各実行されたプログラムは，自身のページテーブル64を有し，ページテーブルエントリ66の有効ビットは，ページが，プライマリメモリにあることを示し，参照されるページに関しこのビットがオフの状態の場合，ページフォルトが起こるものであるところ，引用発明の「参照されるページ」は，プライマリメモリにあり得るものであるのに対し，本願発明は「メインメモリに格納されているデータの第1のページにアクセスするようにとの要求を受取るステップと、

前記データの第1のページが前記メインメモリに格納されていないと判断したことに基づいて，ページフォルトを開始するステップと、」を含むものであるところ，本願発明の「データの第1のページ」は，メインメモリに格納されてい得るものであるから，引用発明の「参照されるページ」は，本願発明の「データの第1のページ」に相当する。

そして，引用発明において，上記した処理が起こる前提として，実行されたプログラムからプライマリメモリにあるであろう上記参照されるページへのアクセス要求が行われることが明らかであることから，引用発明は，プライマリメモリに格納されている参照されるページへのアクセス要求を受け取るものであるといえるから，本願発明と引用発明は，「メインメモリに格納されているデータの第1のページにアクセスするようにとの要求を受取るステップ」を含むものである点において一致する。

さらに，引用発明の「ページテーブルエントリ66の有効ビット」は，「ページが，プライマリメモリにあることを示すものであることから，「参照されるページに関しこのビットがオフの状態の場合」とは，当該参照されるページが，プライマリメモリにない場合であるといえる。すなわち，引用発明は，参照されるページがプライマリメモリにない場合，ページフォルトが起こるものであるから，本願発明と引用発明は，後記する点で相違するものの，「前記データの第1のページが前記メインメモリに格納されていないと判断したこと

に基づいて、「ページフォルトを開始するステップ」を含むものである点において一致する。

7 ダイレクトメモリアクセス（DMA）は、CPUによる処理を介在させず、メモリ間またはメモリとI/Oデバイスの間で直接データを転送する機構であることが、本願の優先日における技術常識である。

ここで、引用発明は、「ページテーブルエントリ68がページはメインメモリにないことを示す場合、データがディスク44から読み込まれる必要があるため、「待機」メッセージが、メモリ・ディスクコントロール76によってCPU12に返され、CPUは、他のタスクに切り替えることを決定し、プログラム可能システムコントロール70は、メモリ・ディスクコントロール76にディスク44から読み出すことを指示し、ページテーブルエントリ68を更新するものであるところ、プログラム可能システムコントロール70及びメモリ・ディスクコントロール76は、ディスク44からのデータの読み出しを、CPUによる処理を介在させずに行っているといえるから、引用発明の「プログラム可能システムコントロール70」と「メモリ・ディスクコントロール76」をあわせたものは、上記技術常識に照らせば、本願発明の「ダイレクトメモリアクセス（DMA）エンジン」に相当するものであることが明らかである。

そして、引用発明の「プログラム可能システムコントロール70」は、「全てのページフォルト処理、ページテーブル64、キャッシュコントロール60及びディスクコントロール76の管理を行う」ものであるところ、引用発明は、プログラム可能システムコントロール70とディスクコントロールを用いて、ページフォルト処理を行うものであるといえるから、本願発明と引用発明は、後記する点で相違するものの、「ダイレクトメモリアクセス（DMA）エンジンを用いて前記ページフォルトを処理する前記ステップ」を含むものである点において一致する。

8 引用発明の「ディスク」は、本願発明の「二次メモリ」に相当する。

また、引用発明は、「ページフォルトの処理において、ディスクからメインメモリに読み込むページをどこに配置するか決定しなければならず、これは、最も過去に使用された（LRU）スキームを用いて行われ、参照ビットがページテーブルエントリに追加され、このページがアクセスされると、参照ビットがセットされ、メモリシステムは、定期的に、これらのビットをチェックしてクリアし、セットされたビット、すなわち、使用されたページのカウンタ値を維持し、低いカウンタ値を有するページが、ディスク読み込みの対象となる」ものであるところ、引用発明の「低いカウンタ値を有するページ」は、各ページのカウンタ値に基づいて決定されるものであることが明らかであるから、本願発明の「特定のページ」と引用発明の「低いカウンタ値を有するページ」とは、「各ページ毎の前記カウンタに基づいて」「決定」されるものである点において一致する。

加えて、引用発明は、「物理ページがダーティである、すなわち、メモリが既に変更されている場合、まずディスクに書き出されなければなら」ないもの

であるところ、引用発明において、上記低いカウント値を有するページがダーティである場合には、当然に、上記低いカウント値を有するページがディスクに書き出されているものと認められる。

さらに、引用発明のページフォルトの処理において、「ディスクからメインメモリに読み込むページをどこに配置するか決定」する際に、「最も過去に使用された（LRU）スキームを用いて行」う必要があるのは、ページテーブルエントリに空きがなく、いずれかのページテーブルエントリに対応するページを解放しない限りは、メインメモリにディスクからページを読み込むことができない場合のみであることが、技術常識から明らかである。

以上を総合すると、引用発明の「ページフォルトの処理」においては、いずれかのページテーブルエントリに対応するページを解放しない限りは、メインメモリにディスクからページを読み込むことができないと判断される場合に、各ページ毎の前記カウントに基づいて低いカウント値を有するページを決定し、当該低いカウント値を有するページの内容をディスクに書き出して解放し、当該書き出され解放された低いカウント値を有するページの位置に、ディスクからメインメモリに読み込むページを配置しているといえるから、本願発明と引用発明とは、後記する点で相違するものの、

「DMAエンジンを用いて前記ページフォルトを処理する前記ステップは、

前記メインメモリがページ転送を受取ることができないと判断したことに応じて、各ページ毎の前記カウントに基づいて特定のページを決定し、二次メモリに前記特定のページをリリースし、前記リリースされた特定のページの位置において、前記データの第1のページを前記二次メモリから前記メインメモリに転送するステップを含む」

ものである点において一致する。

9 以上から、本願発明と引用発明とは、以下の点で一致し、また、相違する。

（一致点）

「コンピュータによって実現される方法であって、

ページテーブルスキャナでのページテーブルのスキャンに基づいて、アクセスビットが前記ページテーブルの各ページテーブルエントリ毎に設定されているかどうかを、プロセッサで判断するステップを含み、前記アクセスビットは、前記ページテーブルエントリに関連付けられたページが最後のスキャン期間においてアクセスされたかどうかを示しており、前記方法はさらに、

各ページ毎にカウントをインクリメントするステップと、

前記アクセスビットが各ページテーブルエントリ毎に設定されているかどうかを前記プロセッサで判断した後、前記アクセスビットをリセットするステップと、

メインメモリに格納されているデータの第1のページにアクセスするようにとの要求を受取るステップと、

前記データの第1のページが前記メインメモリに格納されていないと判断したことに基づいて、ページフォルトを開始するステップと、

ダイレクトメモリアクセス（DMA）エンジンを用いて前記ページフォルトを処理するステップとを含み、

DMAエンジンを用いて前記ページフォルトを処理する前記ステップは、前記メインメモリがページ転送を受取ることができないと判断したことに応じて、各ページ毎の前記カウントに基づいて特定のページを決定し、二次メモリに前記特定のページをリリースし、前記リリースされた特定のページ的位置において、前記データの第1のページを前記二次メモリから前記メインメモリに転送するステップを含む、方法。」

（相違点1）

本願発明は、「カウント」を、アクセスビットがページに関連付けられたページテーブルエントリ毎に設定されていないと判断したことに応じて、インクリメントするのに対し、引用発明は、「カウント値」を、定期的なチェックの結果、参照ビットがページテーブルエントリにセットされている場合、インクリメントする点。

## 第6 判断

### 1 相違点1について

例えば、参考文献1及び参考文献2にも記載されるように、メモリシステムにおいて、各ページへのアクセス頻度を把握する手法として、各ページへのアクセスがあったことを示すアクセスビットを定期的に確認することにより、各ページへのアクセスがあった場合、すなわち、アクセスビットがセットされている場合にカウントをインクリメントする方法、及び、各ページへのアクセスがなかった場合、すなわち、アクセスビットがセットされていない場合にカウントをインクリメントする方法のいずれもが、本願の優先日時点において当業者の常套的手法であったことが認められる。

引用発明は、カウント値に基づいてアクセスの頻度が少ないページを特定するものであり、当該アクセスの頻度を把握するために、定期的なチェックの結果、参照ビットがページテーブルエントリにセットされている場合にカウント値をインクリメントするものであるところ、上記当業者の常套的手法に鑑みれば、引用発明において、カウント値に基づいてアクセスの頻度を把握するために、上記の構成に代えて、定期的なチェックの結果、参照ビットがページテーブルエントリにセットされていない場合にカウント値をインクリメントする構成とすることは、通常の創作能力を有する当業者であれば適宜選択し得た設計的事項にすぎないものと認められる。

したがって、本願発明は、本願の優先日における技術水準に鑑み、引用発明に基づいて当業者が容易に発明をすることができたものである。

### 2 効果・請求人の主張について

請求人は、審判請求書の「第2 4.」において、本願発明は、上記相違点に係る構成（審判請求書の「第2 1.」に記載の「構成C」）により、

「アクセスのないページのカウンートをインクリメントすることで、仮に複数のページのアクセスが0であった場合においても、どちらがより長い期間アクセスされていないかを判別し得」との効果を奏する点において、進歩性を有する旨主張する。

しかしながら、アクセスのないページのカウンートをインクリメントすることにより、複数のページについてどちらがより長い期間アクセスされていないかを判別し得るとの効果がもたらされることは、自明な事項である。そして、アクセスのないページのカウンートをインクリメントすること自体は、当業者の常套的手法にすぎないものであり、引用発明において当該構成を採用することが当業者にとって容易であることは、上記1で検討したとおりである。

したがって、上記効果は当業者が予測できた範囲内のものであり、格別のものとするすることはできず、請求人の上記主張は採用することができない。

## 第7 むすび

以上のとおり、本願発明は、特許法第29条第2項の規定により特許を受けることができないから、他の請求項に係る発明について検討するまでもなく、本願は拒絶されるべきものである。

よって、結論のとおり審決する。

令和 2年 7月 29日

審判長 特許庁審判官 田中 秀人  
特許庁審判官 ▲はま▼中 信行  
特許庁審判官 小林 秀和

(行政事件訴訟法第46条に基づく教示)

この審決に対する訴えは、この審決の謄本の送達があった日から30日(附加期間がある場合は、その日数を附加します。)以内に、特許庁長官を被告として、提起することができます。

審判長 田中 秀人

出訴期間として在外者に対し90日を附加する。

〔審決分類〕 P 1 8 . 1 2 1 - Z ( G 0 6 F )

審判長 特許庁審判官 田中 秀人 9066

特許庁審判官 小林 秀和 3449

特許庁審判官 ▲はま▼中 信行 3240