

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6757289号  
(P6757289)

(45) 発行日 令和2年9月16日 (2020.9.16)

(24) 登録日 令和2年9月1日 (2020.9.1)

(51) Int. Cl.	F I
<b>G 0 6 F 12/02 (2006.01)</b>	G 0 6 F 12/02 5 7 O H
<b>G 0 6 F 12/1081 (2016.01)</b>	G 0 6 F 12/02 5 5 O B
<b>G 0 6 F 12/109 (2016.01)</b>	G 0 6 F 12/1081
	G 0 6 F 12/109 1 0 0

請求項の数 10 (全 10 頁)

(21) 出願番号	特願2017-81927 (P2017-81927)	(73) 特許権者	390019839
(22) 出願日	平成29年4月18日 (2017.4.18)		三星電子株式会社
(65) 公開番号	特開2017-194964 (P2017-194964A)		S a m s u n g E l e c t r o n i c s
(43) 公開日	平成29年10月26日 (2017.10.26)		C o . , L t d .
審査請求日	令和2年4月14日 (2020.4.14)		大韓民国京畿道水原市靈通区三星路129
(31) 優先権主張番号	62/326,537		129, S a m s u n g - r o , Y e o n
(32) 優先日	平成28年4月22日 (2016.4.22)		g t o n g - g u , S u w o n - s i , G
(33) 優先権主張国・地域又は機関	米国 (US)		y e o n g g i - d o , R e p u b l i c
			o f K o r e a
(31) 優先権主張番号	15/333,010	(74) 代理人	110000051
(32) 優先日	平成28年10月24日 (2016.10.24)		特許業務法人共生国際特許事務所
(33) 優先権主張国・地域又は機関	米国 (US)	(72) 発明者	グスーラ, マノイ ケイ
早期審査対象出願			アメリカ合衆国 カリフォルニア州 94
			022, ロス アルトス, ユニット 7,
			ティンダル ストリート 575
			最終頁に続く

(54) 【発明の名称】 メモリの事前割当と関連されたバッファマッピング方式

(57) 【特許請求の範囲】

【請求項 1】

第1のアプリケーション及び第2のアプリケーションと通信するオペレーティングシステムを利用したコンピュータ具現方法であって、

前記第1のアプリケーションから第1の物理的アドレスを受信する段階と、

前記第2のアプリケーションが、アプリケーションのオフセットを用いて前記第1のアプリケーションと通信する段階と、

データ伝達を実現するために前記第1の物理的アドレスに基づいてオペレーティングシステムレベルの仮想アドレスを決定する段階と、を含み、

前記第1のアプリケーションは、第1のページテーブル有し、前記第1のページテーブルを使用して前記第1のアプリケーションの仮想ページをメモリの物理的ページにマッピングし、前記第1のアプリケーションに対応する前記第1の物理的アドレスを決定するために第1の物理的ページフレームナンバー及び第1のオフセットを使用することによって、前記第1の物理的アドレスは決定され、

前記アプリケーションのオフセットは、前記第2のアプリケーションに対応する第2の物理的アドレスを決定するために前記第1の物理的アドレスに適用され、

前記オペレーティングシステムレベルの仮想アドレスは、前記第1の物理的アドレスのみに基づいたカーネル変換テーブルによって決定され、前記オペレーティングシステムは、前記第1の物理的アドレスのみを介して前記第1のアプリケーション及び前記第2のアプリケーションと通信することを特徴とする方法。

10

20

## 【請求項 2】

前記第 1 のアプリケーション又は前記第 2 のアプリケーションが使用者入力を受信する前にメモリの割当を遂行し、前記第 1 のアプリケーション及び前記第 2 のアプリケーションと前記割当を共有する段階をさらに含むことを特徴とする請求項 1 に記載の方法。

## 【請求項 3】

前記オペレーティングシステム内に多数のモジュールがあり、前記モジュールの各々は、自分のオペレーティングシステムレベルの仮想メモリを有し、

前記モジュールの全てが前記第 1 の物理的アドレスを使用して前記第 1 のアプリケーションと直接通信することを許容する段階をさらに含むことを特徴とする請求項 1 に記載の方法。

10

## 【請求項 4】

バッファを利用して互いに通信する単一ノードの第 1 のアプリケーション、第 2 のアプリケーション、オペレーティングシステム、及びハードウェアを利用したコンピュータ具現方法であって、

前記第 1 のアプリケーションが、第 1 のアプリケーションレベルの仮想アドレスを前記第 1 のアプリケーションに対応する第 1 の物理的アドレスに変換し、前記オペレーティングシステムに前記第 1 の物理的アドレスを通信する段階と、

前記第 2 のアプリケーションに対応する第 2 の物理的アドレスを決定するために、前記第 1 の物理的アドレスに適用されるアプリケーションのオフセットを用いて前記第 1 のアプリケーションと前記第 2 のアプリケーションとが互いに通信する段階と、

20

前記オペレーティングシステムが、前記第 1 の物理的アドレスを使用してオペレーティングシステムレベルの仮想アドレスを決定する段階と、を含み、

前記オペレーティングシステムレベルの仮想アドレスは、前記第 1 の物理的アドレスのみに基づいたカーネル変換テーブルによって決定され、前記オペレーティングシステムは、前記第 1 の物理的アドレスのみを介して前記第 1 のアプリケーション及び前記第 2 のアプリケーションと通信することを特徴とする方法。

## 【請求項 5】

前記第 1 のアプリケーションは、前記オペレーティングシステムの介入無しでハードウェアのオフセットを使用して計算された直接メモリアクセス (Direct Memory Access、以下、DMA) アドレスを使用して、前記ハードウェアと通信することを特徴とする請求項 4 に記載の方法。

30

## 【請求項 6】

前記オペレーティングシステムは、メモリバッファを事前割当し、使用者入力を受信する前に、物理的アドレスをアプリケーションにマッピングする方法を提供することを特徴とする請求項 4 に記載の方法。

## 【請求項 7】

前記第 1 のアプリケーションは、前記第 1 のアプリケーション、前記第 2 のアプリケーション、前記オペレーティングシステム、及び前記ハードウェアの間で前記バッファを使用してデータを伝達する前に、前記第 1 の物理的アドレス及びメモリの DMA アドレスを獲得することを特徴とする請求項 4 に記載の方法。

40

## 【請求項 8】

前記オペレーティングシステム内に複数のモジュールがあり、前記複数のモジュールの各々は、前記第 1 のアプリケーションと直接通信し、前記第 1 の物理的アドレスを利用して自分のオペレーティングシステムレベルの仮想アドレスを決定することを特徴とする請求項 4 に記載の方法。

## 【請求項 9】

データ伝達を制御する装置であって、

第 1 のアプリケーションが、第 1 のアプリケーションレベルの仮想アドレスを前記第 1 のアプリケーションに対応する第 1 の物理的アドレスに変換し、オペレーティングシステムに前記第 1 の物理的アドレスを通信することを許容し、第 2 のアプリケーションが、ア

50

アプリケーションのオフセットを用いて前記第 1 のアプリケーションと通信することを許容するメモリマッパーを備え、

前記アプリケーションのオフセットは、前記第 2 のアプリケーションに対応する第 2 の物理的アドレスを決定するために、前記第 1 の物理的アドレスに適用され、

カーネル変換テーブルは、前記第 1 の物理的アドレスのみに基づいたオペレーティングシステムレベルの仮想アドレスを決定するために使用され、

前記オペレーティングシステムは、前記第 1 の物理的アドレスのみを介して前記第 1 のアプリケーション及び前記第 2 のアプリケーションと通信することを特徴とする装置。

【請求項 10】

前記メモリマッパーは、前記第 1 のアプリケーションが DMA アドレス及びハードウェアのオフセットを使用して、ハードウェアと直接通信することを許容することを特徴とする請求項 9 に記載の装置。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は一般的にメモリバッファに係り、具体的にはメモリの事前割当と関連されたバッファマッピング方式に係る。

【背景技術】

【0002】

UNIX（登録商標）基盤のストレージ/サーバシステム内には各々が特定作業を遂行する多様なアプリケーション及び装置ドライバがある。アプリケーション、オペレーティングシステム/カーネル（operating system（OS）/kernel）、及びハードウェアは、効果的に通信するために、しばしばメモリバッファに順にアクセス（pass\_around）する。通常、このような通信をする間に、アプリケーションはそのアプリケーションレベル仮想アドレス（application-level virtual address）をオペレーティングシステム/カーネルに伝達する。メモリバッファはアプリケーションレベル仮想アドレスを使用してドライバを呼出し、ドライバはアプリケーションレベル仮想アドレスをオペレーティングシステム/カーネルレベル仮想アドレス（operating system/kernel-level virtual address）にマッピング（mapping）する。

20

30

【0003】

このような変換を容易にするために、仮想及び物理的メモリはページと称される扱い易いサイズのチャンク（chunk）に分割される。このページドモデル（paged model）において、仮想アドレスはオフセット（offset）及び仮想ページフレームナンバー（virtual page frame number）から構成される。プロセッサが仮想アドレスに出会う毎に、プロセッサは仮想アドレスからオフセット及び仮想ページフレームナンバーを抽出する。プロセッサはその後、仮想ページフレームナンバーを物理的ページフレームナンバーに変換してその物理的ページ内の正しいオフセットの位置にアクセスする。

40

【0004】

仮想アドレスを物理的アドレスに変換するために、プロセッサは第 1 に仮想アドレスページフレームナンバーとその仮想ページ内のオフセットを算出する。プロセッサは仮想ページフレームナンバーをプロセスページテーブルへのインデックスとして使用してそのページテーブルエントリを検索する。そのオフセットにおけるページテーブルエントリが有効であれば、プロセッサはこのエントリから物理的ページフレームナンバーを持って来る。プロセッサが仮想ページフレームナンバーを物理的フレームナンバーに変換するのに使用するテーブルはページテーブル（page table）と称される。

【0005】

仮想アドレスはオフセットを仮想ページナンバーに加算することによって計算される。

50

保護機能を強化するために、使用者領域（空間、スペース）アプリケーション及びカーネルのために別途分離されたページテーブルがある。使用者領域仮想アドレスにアクセスするために、カーネルレベルソフトウェアは使用者領域アドレスをカーネルアドレス領域にマッピングする。このプロセスは使用者領域アドレスのためにカーネルページテーブルエントリを生成することを含む。

#### 【 0 0 0 6 】

ハードウェアに対しては、オペレーティングシステム／カーネルとハードウェアとの間のコネクションは、直接メモリアクセス（direct memory access、以下、DMA）として発生する。DMAを使用して、ハードウェア装置はCPUが介入しなくとも、コンピュータのメインメモリから、又はコンピュータのメインメモリへ、データを伝達できる。DMAが作動するために、メモリバッファはハードウェア装置に認識されるアドレス範囲に頻繁にマッピングされる。このアドレス範囲はIO仮想アドレスと称される。このマッピングはアーキテクチャにも依存するが、IO仮想アドレスとコンピュータメインメモリの物理的アドレスとの間の変換を設定（set\_\_up）することを含む。一般的に、これは入出力メモリ管理装置（Input / Output Memory Management Unit、IOMMU）ハードウェアを使用して発生する。幾つかのアーキテクチャにおいて、IO仮想アドレスはコンピュータメインメモリの物理的アドレスと同一である。

#### 【 0 0 0 7 】

上述したマッピング方式は、ページテーブルエントリを設定することによってアプリケーションレベル仮想アドレスをオペレーティングシステムレベル仮想アドレスに最初に変換することを要求するので、オペレーティングシステム／カーネルに重い負担を与える。同様に、DMAマッピングはすべてのDMA伝送のためにその都度設定されなければならない。そこで、OS、アプリケーション、及びハードウェア間の通信がさらに効率的にできる方法が要求される。

#### 【 先行技術文献 】

#### 【 特許文献 】

#### 【 0 0 0 8 】

【 特許文献 1 】 米国特許第 8 , 9 0 9 , 7 2 7 号公報

【 特許文献 2 】 米国特許第 6 , 4 0 5 , 2 3 7 号公報

【 特許文献 3 】 米国特許第 8 , 5 0 4 , 7 4 4 号公報

【 特許文献 4 】 米国特許第 8 , 1 6 6 , 1 9 4 号公報

【 特許文献 5 】 米国特許第 7 , 5 9 0 , 7 7 7 号公報

【 特許文献 6 】 米国特許第 8 , 8 6 8 , 8 6 7 号公報

【 特許文献 7 】 米国特許第 9 , 0 9 2 , 4 2 6 号公報

【 特許文献 8 】 米国特許公開第 2 0 0 7 / 0 2 5 5 8 6 6 号明細書

#### 【 発明の概要 】

#### 【 発明が解決しようとする課題 】

#### 【 0 0 0 9 】

上記の要求に応じるためになされた本発明は、アプリケーションレベル仮想アドレスをオペレーティングシステムレベル仮想アドレスに最初に変換する必要が無く、オペレーティングシステム／カーネルに重い負担を与えない、同様に、すべてのDMA伝送のためにDMAマッピングをその都度設定する必要が無い、ような、アプリケーション、オペレーティングシステム、及びハードウェア間の効率的な通信を可能にするコンピュータ具現方法を提供する。

#### 【 課題を解決するための手段 】

#### 【 0 0 1 0 】

本発明の一実施形態による発明の概念は、オペレーティングシステムがアプリケーションと通信するために利用するコンピュータ具現方法に係る。この方法では、オペレーティングシステムがアプリケーションから直接物理的アドレスを受信し、データ伝達を実現す

10

20

30

40

50

るためにメモリの物理的アドレスに基づいてオペレーティングシステムレベル仮想アドレスを決定する。

【 0 0 1 1 】

本発明の他の一実施形態による発明の概念は、アプリケーション、オペレーティングシステム、及びハードウェアが、バッファを利用して互いに単一ノードにおいて通信するために利用するコンピュータ具現方法に係る。この方法では、アプリケーションが、アプリケーションレベル仮想アドレスを物理的アドレスに変換し、その物理的アドレスをオペレーティングシステムに対して通信する。そうすると、オペレーティングシステムは、その物理的アドレスを使用してオペレーティングシステムレベル仮想アドレスを決定する。

【 0 0 1 2 】

本発明のその他の一実施形態による発明の概念は、データ伝達制御装置に係り、前記装置は、アプリケーションが自分のアプリケーションレベル仮想アドレスを物理的アドレスに変換し、その物理的アドレスをオペレーティングシステムに通信することを可能にするメモリマッパーを含む。

【発明の効果】

【 0 0 1 3 】

本発明の実施形態によれば、オペレーティングシステムがアプリケーションから直接物理的アドレスを受信し、データ伝達を実現するためにメモリの物理的アドレスに基づいてオペレーティングシステムレベル仮想アドレスを決定するので、例えば、アプリケーションは単純にオフセットをバッファの開始点のDMAアドレスに加算してオフセットのDMAアドレスを決定できる。このように、アプリケーションレベル仮想アドレスをオペレーティングシステムレベル仮想アドレスに最初に変換する必要が無く、オペレーティングシステム/カーネルに重い負担を与えない、同様に、すべてのDMA伝達のためにDMAマッピングをその都度設定する必要が無い。よって、アプリケーションはカーネルの介入無しでバッファアドレスをハードウェア装置に直接伝達でき、アプリケーション、オペレーティングシステム、及びハードウェア間の効率的な通信が可能になる。

【図面の簡単な説明】

【 0 0 1 4 】

【図 1】本発明の一実施形態に係る 1 つの可能な環境を提供する、単一ノードにおける使用者領域、カーネル領域、及びハードウェアを概念的に示した図である。

【図 2】本発明の一実施形態に係るアプリケーション、オペレーティングシステム、及びハードウェアの間の通信方法を示した概略図である。

【図 3】本発明の一実施形態に係る使用者領域における多様なアプリケーション、及び同一の物理的アドレスをポインティングし、バッファ共有を可能にするカーネル内の相異なる仮想アドレスを示した概略図である。

【図 4】本発明の一実施形態に係るアプリケーション、オペレーティングシステム、及びハードウェアの間の通信方法を示した概略図である。

【発明を実施するための形態】

【 0 0 1 5 】

本発明に係るシステムでは、オペレーティングシステムが、メモリバッファが通過される毎にカーネルレベルページテーブル基盤の変換を設定する必要がない。本開示では、アプリケーションは物理的アドレスをカーネルに伝達する。本発明の一実施形態によれば、カーネルはこのようなバッファのために要求されるマッピングを有している。従って、カーネルは仮想アドレスを計算でき、その度にマッピング動作を遂行しなくともよい。すべてのカーネルモジュールが同一の仮想アドレス領域を共有するので、どのオペレーティングシステムモジュール（メモリに割当されたオペレーティングシステムモジュールのみならず）も、物理的アドレスを使用して仮想アドレスを獲得し、バッファ上で動作できる。

【 0 0 1 6 】

相異なるアプリケーションの間で、通信がバッファオフセットを使用して発生する。アプリケーションは仮想アドレスを使用してバッファ上で動作する。アプリケーションは、

10

20

30

40

50

単純にオフセットをバッファの開始点の仮想アドレスに加算して自分の仮想アドレスを計算できる。

【0017】

アプリケーションは、単純にオフセットをバッファの開始点のDMAアドレスに加算してオフセットのDMAアドレスを決定できる。アプリケーションはカーネル介入無しでバッファアドレスをハードウェア装置に直接伝達できる。

【0018】

本開示は単一ノードの状況に対して開示しているが、本発明はこれに制限されない。

【0019】

図1は本発明の一実施形態に係る1つの可能な環境を提供する、単一ノード(single node)における使用者領域(user space)、カーネル領域(kernel space)、及びハードウェアを概念的に示す。図示されたように、使用者領域を形成するアプリケーション100、オペレーティングシステム/カーネル200、及びハードウェア300は互いに通信して、使用者の要請を受信し、実行する。ハードウェア300は多様な装置、中央処理装置、及びシステムメモリを含む。

オペレーティングシステム200は、使用者領域を形成するアプリケーション100及びハードウェア300の間のインターフェイスを担い、且つ、中でもアプリケーション100にシステムメモリへのアクセスを可能にする。装置ドライバは一般的にオペレーティングシステム200の一部である。メモリマッパーはイメージ及びデータファイルを使用者領域のアプリケーション内にマッピングする。メモリマッピングにおいて、ファイルのコンテンツは仮想アドレスと連結(link)される。

【0020】

図2は本発明の一実施形態に係るアプリケーション、オペレーティングシステム、及びハードウェアの間の通信方法を示した概略図である。アプリケーション100に関しては、2つのアプリケーション、アプリケーションX及びアプリケーションYが例示される。2つのアプリケーションの各々は自分の仮想アドレスのセットと共に自分の仮想メモリVMを有し、これは図2のVM-X及びVM-Yで示されている。

また、各アプリケーションは自分の各々の仮想ページをメモリの物理的ページにマッピングする自分のページテーブル110を有する。例えば、図示されたように、アプリケーションXの仮想ページフレームナンバー0(VPFN0)は物理的ページフレームナンバー1(PFN1)のメモリにマッピングされ、アプリケーションYの仮想ページフレームナンバー1(VPFN1)は物理的ページフレームナンバー4(PFN4)のメモリにマッピングされる。

【0021】

ページテーブル110は仮想ページフレームナンバーをオフセットとして使用してアクセスされる。仮想アドレスを物理的アドレスに変換するために、仮想アドレスページフレームナンバー及びその仮想ページ内のオフセットが先に決定される。仮に仮想メモリアドレスが有効であり、テーブルエントリが有効であれば、プロセッサは物理的ページフレームナンバーを取り上げ、これにページサイズを乗じて、物理的メモリのページのベース(base)アドレスを獲得する。その後、オフセットが加えられる。

【0022】

例えば、図2に示されたように、ページサイズが0x2000であると仮定する。VM-Yの0x2194のアドレスに対して、プロセッサは前記アドレスを仮想ページフレームナンバー1内の0x194のオフセットに変換する。この仮想ページフレームナンバー1は物理的ページフレームナンバー4にマッピングされ、これは0x8000(4x2000)から開始する。0x194のオフセットを物理的ページフレームナンバーに加えて、最終物理的アドレス0x8194を生成する。

従来のアプリケーションは仮想アドレスと仮想アドレスのベースアドレスに対するオフセットと、のみを使用して互いに通信するが、本システムは、アプリケーションが物理的アドレスを使用してカーネルと通信できるようにする。図示されたように、カーネル変換

10

20

30

40

50

テーブル 2 1 0 は物理的アドレスをオペレーティングシステムレベル仮想アドレスに変換するのに使用される。カーネル変換テーブル 2 1 0 は物理的アドレスの仮想アドレスへの変換を可能にし、各オペレーティングシステムに特有 ( O S - s p e c i f i c ) であり得る。

#### 【 0 0 2 3 】

本発明の一実施形態によれば、メモリは、アプリケーション 1 0 0 と事前割当 ( P r e - a l l o c a t i o n ) 及び共有され、従って、アプリケーション 1 0 0 及びオペレーティングシステム 2 0 0 は共に物理的アドレステーブルにアクセスできる。ここで使用された “ 事前割当 ( P r e - a l l o c a t i o n ) ” は、アプリケーション / カーネル / ハードウェアの各ドメイン間のデータを伝達するためにバッファが使用される前に割当されることを意味する。また、オペレーティングシステム 2 0 0 内の相異なるモジュールは、すべてのモジュールが同一の仮想アドレス領域を共有する場合、物理的アドレスを自分のオペレーティングシステムレベル仮想アドレスに変換する。

10

各オペレーティングシステムが物理的アドレスを仮想アドレスに変換するのに使用する方法は、各オペレーティングシステムのアーキテクチャに依存する。例えば、リナックス ( 登録商標 ) オペレーティングシステムは、アドレスの或る特定範囲に対する単純な算術演算を使用して物理的アドレスを仮想アドレスに変換する。リナックス ( 登録商標 ) で具現される場合、本システムの事前割当されたバッファは、物理的アドレスに到達するために単純な算術が使用されたこのようなアドレス範囲に属する。一部の他のオペレーティングシステムは相異なるメカニズムを通じてこの作業を遂行される。

20

#### 【 0 0 2 4 】

この場合アプリケーションは、オフセットの D M A アドレスを単純にバッファの開始点の D M A アドレスにオフセットを加えて計算できる。このように、アプリケーションはオペレーティングシステム 2 0 0 の介入無しでハードウェア 3 0 0 装置にバッファアドレスを直接伝達できる。

#### 【 0 0 2 5 】

図 3 は本発明の一実施形態に係る、使用者領域における多重アプリケーション、及び、同一の物理的アドレスをポインティングし、バッファ共有を可能にするカーネル内の相異なる仮想アドレスを示した概略図である。図 3 ではアプリケーション使用者領域 1 0 0 のアプリケーション X 及びアプリケーション Y が例示されている。アプリケーション X において “ バッファ - 1 ” とラベルされたデータはアプリケーション X のアドレス領域 0 x 3 0 0 0 に格納される。このデータは、例えば図 2 に示されたプロセスを使用してカーネルアドレス領域 0 x 5 0 0 0 に変換される。

30

同一のデータ ( バッファ - 1 ) は、アプリケーション Y のアドレス領域アドレス 0 x 1 0 0 0 のデータに対応するが、2つのアプリケーション X 及び Y は物理的アドレスを使用して同一のデータを指し示すことができる。 “ バッファ - 2 ” とラベルされたデータはアプリケーション Y の仮想アドレス 0 x 4 0 0 0 に格納され、これはカーネルアドレス領域 0 x 7 0 0 0 に対応する。カーネルモジュールが同一の仮想アドレス領域を共有するという事実のため、如何なるオペレーティングシステムモジュールも図 3 に示されたカーネルアドレスを使用して仮想アドレスを得る。

40

#### 【 0 0 2 6 】

図 4 は本発明の一実施形態に係るアプリケーション、オペレーティングシステム、及びハードウェアの間の通信方法を示した概略図である。具体的には、図 4 の実施形態は使用者領域 ( アプリケーション 1 0 0 ) のアプリケーション - 1 ( 1 0 2 ) 及びアプリケーション - 2 ( 1 0 4 ) がオフセット 1 0 6 を使用して互いに通信し、アプリケーション - 2 ( 1 0 4 ) は物理的アドレス 2 0 4 を使用してカーネルモジュール 2 0 2 と通信することを示す。アプリケーション ( 例えば、アプリケーション - 2 ( 1 0 4 ) ) はオフセットを利用して生成された D M A アドレスを使用してハードウェア装置 3 0 2 と直接的に通信できる。

#### 【 0 0 2 7 】

50

本発明の一実施形態によれば本システムは、機械によって実行可能な少なくとも１つのコードセクションを有するコンピュータプログラムが格納されたコンピュータ読出し可能な格納媒体を含み、これによって上述したような段階を機械に遂行させる。

#### 【 0 0 2 8 】

本発明の一実施形態によれば、本システムはハードウェア、ソフトウェア、又は、ハードウェアとソフトウェアの組合せにより具現できる。本開示は単一のコンピュータシステムを含む単一ノード具現に中心に説明しているが、これは互いに異なる要素が複数の相互連結されたコンピュータシステムに亘って散在している分散方式において使用するように適応できよう。任意の種類 of コンピュータシステム又は装置は、本願に記載された方法を遂行するように適応するのに適合できる。典型的なハードウェアとソフトウェアの組合せは、ローディングされて実行される時、ここに記述された方法を遂行するようにコンピュータシステムを制御するコンピュータプログラムを有する汎用コンピュータシステムである。

10

#### 【 0 0 2 9 】

本システムは、上述した方法の具現を可能にするすべての機能を含み、且つコンピュータシステムにローディングされる時、この方法を遂行できる、コンピュータプログラム製品に組み込み可能である。本文脈で“コンピュータプログラム”とは、情報処理能力を有するシステムが特定の機能を、(a)直接遂行するように意図された、若しくは、(b) 1) 別の言語、コード、又は表記法に変換、2) 相異なる材質形態に再生、の中でいずれか 1 つ、或いは双方を行った後に遂行するように意図された命令のセットの、何らかの言語、コード、又は表記法による何らかの表現を意味する。

20

#### 【 0 0 3 0 】

本発明が特定な実施形態を参照して説明されたが、当業者ならば本発明の範囲を逸脱しない範囲内で多様な変更をなし、等価物で置換できることが理解できよう。また、本発明の範囲を逸脱せずに本発明の教示に特定状況又は特定材料を適応させる多くの変形が可能であろう。従って、本発明の開示は、本開示が開示された特定実施形態に限定されず、本発明が添付された請求の範囲内に属するすべての実施形態を含むことを意図している。

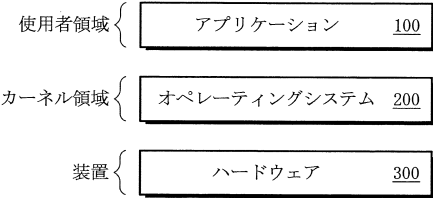
#### 【 符号の説明 】

#### 【 0 0 3 1 】

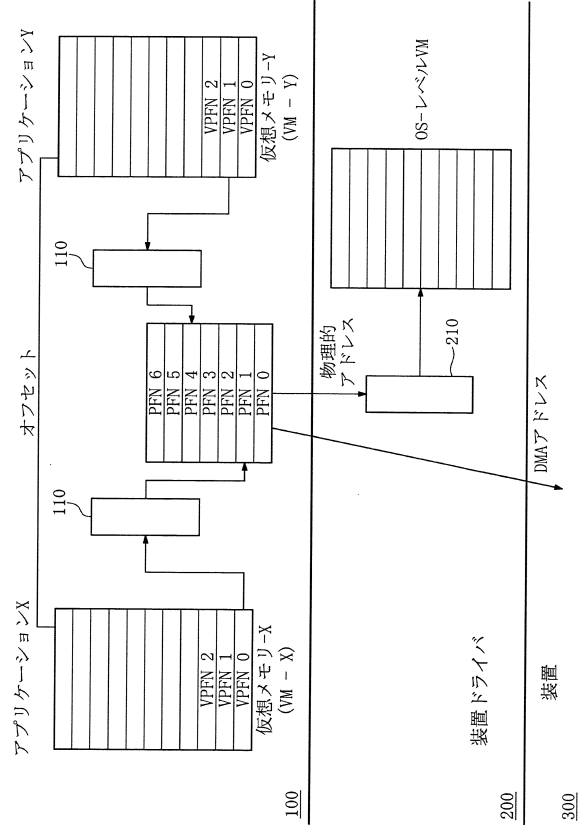
1 0 0	アプリケーション	30
1 0 2	アプリケーション - 1	
1 0 4	アプリケーション - 2	
1 0 6	オフセット	
1 1 0	ページテーブル	
2 0 0	オペレーティングシステム	
2 0 2	カーネルモジュール	
2 0 4	物理的アドレス	
2 1 0	カーネル変換テーブル	
3 0 0	ハードウェア	
3 0 2	ハードウェア装置	40
P F N n	物理的ページフレームナンバー n	
V M	仮想メモリ	
V M - X、V M - Y	仮想メモリ - X、仮想メモリ - Y	
V P F N n	仮想ページフレームナンバー n	



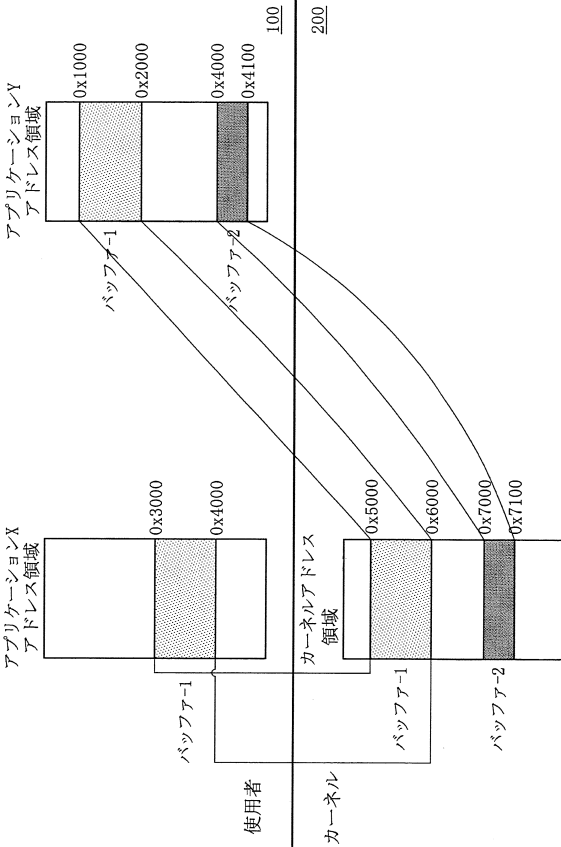
【図 1】



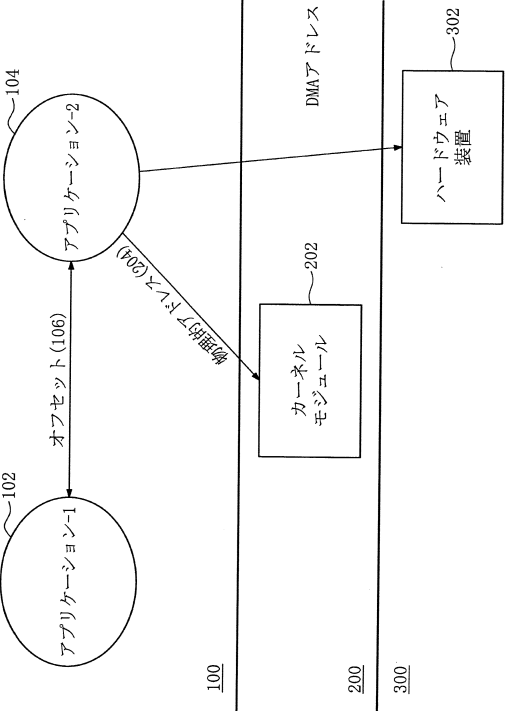
【図 2】



【図 3】



【図 4】



---

フロントページの続き

(72)発明者 ゴルラブディ, ヴェンカタ バース ブラカシュ  
アメリカ合衆国 カリフォルニア州 95051, サンタ クララ, #60, ホームステッド ロ  
ード 3455

審査官 後藤 彰

(56)参考文献 特開2003-248626(JP, A)  
特開2001-22640(JP, A)  
米国特許第9092426(US, B1)  
米国特許出願公開第2015/0095610(US, A1)

(58)調査した分野(Int.Cl., DB名)  
G06F 12/02  
G06F 12/1081  
G06F 12/109