(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5583274号
(P5583274)

(45)発行日　平成26年9月3日(2014.9.3)　(24)登録日　平成26年7月25日(2014.7.25)

(51)Int.Cl.　F I
G06F 12/10 (2006.01)　G06F 12/10 501Z
G06F 12/08 (2006.01)　G06F 12/08 507Z

請求項の数 10　(全 11 頁)

(21)出願番号　特願2013-523548 (P2013-523548)
(86)(22)出願日　平成23年7月7日 (2011.7.7)
(65)公表番号　特表2013-544380 (P2013-544380A)
(43)公表日　平成25年12月12日 (2013.12.12)
(86)国際出願番号　PCT/EP2011/061513
(87)国際公開番号　WO2012/016783
(87)国際公開日　平成24年2月9日 (2012.2.9)
審査請求日　平成25年3月21日 (2013.3.21)
(31)優先権主張番号　10305868.1
(32)優先日　平成22年8月6日 (2010.8.6)
(33)優先権主張国　欧州特許庁 (EP)

(73)特許権者　391030332
アルカテル－ルーセント
フランス国、９２１００・ブローニュ－ビヤンクール、ルート・ドゥ・ラ・レーヌ・１４８／１５２
(74)代理人　100094112
弁理士　岡部　讓
(74)代理人　100106183
弁理士　吉澤　弘司
(74)代理人　100128657
弁理士　三山　勝巳
(74)代理人　100160967
弁理士　▲濱▼口　岳久
(74)代理人　100170601
弁理士　川崎　孝

最終頁に続く

(54)【発明の名称】コンピュータ・メモリを管理する方法、対応するコンピュータ・プログラム製品、およびそのためのデータ・ストレージ・デバイス

(57)【特許請求の範囲】
【請求項１】
コンピュータ・メモリを管理する方法であって、
仮想アドレスを物理アドレスにマップするためのページ・テーブル・エントリー、および複数のデータ・ブロックを含むキャッシュを保持するステップ（１０１）と、
前記仮想アドレスへの参照に応答して、前記ページ・テーブル・エントリーを用いて前記仮想アドレスを前記物理アドレスへと変換するステップ（１０２）と、データを前記物理アドレスから前記キャッシュ内へフェッチするステップ（１０３）と
を含む方法において、前記ページ・テーブル・エントリーが、複数のインジケータを含み、それぞれのデータ・ブロックが、前記複数のインジケータのうちの１つに対応し、前記データを前記キャッシュ内へフェッチするステップ（１０３）が開始すると、当該方法が、
前記複数のインジケータから選択された１つのインジケータ（１１０）がセットされていることに応答して、前記対応するデータ・ブロックをゼロ化するさらなるステップ（１０４）を含むことを特徴とする方法。
【請求項２】
前記ページ・テーブル・エントリーがビットマスクを含み、前記インジケータが、前記ビットマスク内に含まれているビットであることを特徴とする、請求項１に記載の方法。
【請求項３】
前記ページ・テーブル・エントリーが、前記複数のデータ・ブロックを含むメモリ・ペ

ージに関連付けられ、前記方法が、
データ・ブロックを受け取る中間ステップと、
前記データ・ブロックを前記メモリ・ページ内に格納する中間ステップと、
前記データ・ブロックが対応する前記インジケータをクリアする中間ステップとを含むことを特徴とする、請求項１に記載の方法。
【請求項４】
前記データ・ブロックが対応していない、前記複数のインジケータのうちの残りのインジケータをセットするさらなるステップ
を含むことを特徴とする、請求項１に記載の方法。
【請求項５】
さらなるデータ・ブロックを上書きする後続ステップと、
前記さらなるデータ・ブロックが対応する前記インジケータをクリアする後続ステップと
を含むことを特徴とする、請求項３または４に記載の方法。
【請求項６】
コンピュータ・プログラムであって、プログラムがコンピュータ上で実行されたときに請求項１に記載の方法を実行するためのコンピュータ実行可能命令を含む、コンピュータ・プログラム。
【請求項７】
オペレーティング・システムを含むことを特徴とする、請求項６に記載のコンピュータ・プログラム。
【請求項８】
仮想アドレスを物理アドレスにマップするためのページ・テーブル・エントリー、および複数のデータ・ブロックを含むキャッシュを保持するステップ（１０１）を行うための手段と、
前記仮想アドレスへの参照に応答して、前記ページ・テーブル・エントリーを用いて前記仮想アドレスを前記物理アドレスへと変換するステップ（１０２）、およびデータを前記物理アドレスから前記キャッシュ内へフェッチするステップ（１０３）を行うための手段と
を含むデバイスにおいて、前記ページ・テーブル・エントリーが、複数のインジケータを含み、それぞれのデータ・ブロックが、前記複数のインジケータのうちの１つに対応し、当該デバイスが、前記データを前記キャッシュ内へフェッチするステップ（１０３）が開始されると、
前記複数のインジケータから選択された１つのインジケータ（１１０）がセットされていることに応答して、前記対応するデータ・ブロックをゼロ化するステップ（１０４）を行うためのさらなる手段を含むことを特徴とするデバイス。
【請求項９】
中央処理装置、
メモリ・マネージメント・ユニット、および
データ・キャッシュ
のうちの少なくとも１つを含むことを特徴とする、請求項８に記載のデバイス。
【請求項１０】
前記ページ・テーブル・エントリーを格納するように構成されている変換ルックアサイド・バッファをさらに含み、前記仮想アドレスが、前記変換ルックアサイド・バッファを用いて変換される（１０２）ことを特徴とする、請求項８または９に記載のデバイス。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、仮想アドレスを物理アドレスにマップするためのページ・テーブル・エントリー、および複数のデータ・ブロックを含むキャッシュを保持するステップと、仮想アド

レスへの参照に応答して、ページ・テーブル・エントリーを用いて仮想アドレスを物理アドレスへと変換するステップと、データを物理アドレスからキャッシュ内へフェッチするステップとを含む、コンピュータ・メモリを管理する方法に関する。本発明はさらに、コンピュータ・プログラム製品であって、プログラムがコンピュータ上で実行されるときに前記方法を実行するためのコンピュータ実行可能命令を含む、コンピュータ・プログラム製品に関し、また、前記方法を実行するようにプログラムまたは構成されているデバイスに関する。

【背景技術】

【０００２】

コンピューティングにおいては、メモリ・マネージメントは、コンピュータ・メモリを管理する行為である。これは、そのよりシンプルな形態においては、プログラムの要求に応じてそれらのプログラムにメモリの一部分を割り当てるための方法を提供することと、もはや必要なくなったときに再利用のためにメモリを解放することとを含む。メモリのマネージメントは、いかなるコンピュータ・システムにとっても重要である。

【０００３】

仮想メモリ・システムは、プロセスによって使用されるメモリ・アドレスを物理アドレスから分離し、プロセスの分離を可能にし、メモリのうちの有効に利用できる量を増やす。仮想メモリ・マネージャーの質は、全体的なシステム・パフォーマンスに著しい影響を与える。

【０００４】

Ｌｕｄｍｉｌａ ＣｈｅｒｋａｓｏｖａおよびＲｏｂ Ｇａｒｄｎｅｒは、「Ｍｅａｓｕｒｉｎｇ ＣＰＵ Ｏｖｅｒｈｅａｄ ｆｏｒ Ｉ／Ｏ Ｐｒｏｃｅｓｓｉｎｇ ｉｎ ｔｈｅ Ｘｅｎ Ｖｉｒｔｕａｌ Ｍａｃｈｉｎｅ Ｍｏｎｉｔｏｒ」、Ｐｒｏｃｅｅｄｉｎｇｓ ｏｆ ｔｈｅ ＵＳＥＮＩＸ Ａｎｎｕａｌ Ｔｅｃｈｎｉｃａｌ Ｃｏｎｆｅｒｅｎｃｅ ２００５、アメリカ合衆国カリフォルニア州アナハイムにおいて、仮想メモリ・システムの概要を示している。オペレーティング・システム（ＯＳ）が、メイン・メモリにおいて使用するために二次ストレージからデータを取り出せるようにするために、この従来のシステムは、ページングとして知られているメモリ・マネージメント・スキームを利用し、このページングにおいては、データは、メモリ・ページと呼ばれる固定サイズのブロックで取り出される。本明細書においては、二次ストレージ（外部メモリと呼ばれる場合もある）とは、ハード・ディスク・ドライブ、光ストレージ、フラッシュ・メモリ、フロッピー・ディスク、磁気テープ、紙テープ、せん孔カード、またはＺｉｐドライブなど、中央処理装置（ＣＰＵ）によって直接アクセス可能ではない任意のストレージを意味する。

【０００５】

一般に入力／出力またはＩ／Ｏと呼ばれる、他のシステムとの効率のよい通信を可能にするために、現況技術の仮想メモリ・システムは、ページ・フリッピングとして知られているメカニズムを実装している。この技術によれば、アプリケーションが、入力データを受け取るための１つまたは複数のメモリ・ページを指定し、それによってＯＳは、その内蔵されたページング・スキームを用いて、そのデータを供給することができる。このために、ＯＳは、アプリケーションのワーキング・メモリ（一般には、そのアドレス空間と呼ばれる）内の指定されたメモリ・ページを、要求された入力と「交換」する。

【０００６】

この知られているシステムの主なマイナス面は、供給される入力データがターゲットのメモリ・ページに正確にフィットすることがめったにないという事実にある。そのようにして交換されるそれぞれのメモリ・ページの残りは、メモリの初期化に依存するアプリケーションに対応するために、ならびに要求元のアプリケーションが外部の潜在的に機密性のデータへの無許可のアクセスを得ることを防止するために、ゼロ化する必要がある。ソフトウェア開発において、ゼロ化するとは、固定された無意味な値、たとえばゼロでデータを上書きして、そのデータの不測の開示を防止することを意味し、そのような開示は、

要求元のアプリケーションによるセキュリティー侵害を潜在的に可能にする。
【０００７】
米国特許第５９２０８９５Ａ号によれば、マップされたファイルＩ／Ｏを使用してキャッシュされるファイルを書き込む効率は、ファイルのキャッシュされたページ内の初期化されていないデータをゼロ化することを、そのファイルがユーザ・モード・スレッドによってマップされるまで控えることによって、改善される。ページング・オペレーションが仮想メモリ・マネージャーによってコントロールされ、マップされたファイルＩ／Ｏを使用したメモリ・ベースのキャッシングがキャッシュ・マネージャーによって管理されるオペレーティング・システムにおいては、書き込みの際に、マップされたファイルをゼロ化するのを控えることは、仮想メモリ・マネージャーとキャッシュ・マネージャーの間における通信のための一式の内部オペレーティング・システム・インターフェースによって実施される。キャッシュされているファイルが、ユーザ・モード・スレッドによってまだマップされていないときには、キャッシュ・マネージャーは、そのファイルのキャッシュ・ページがどの程度まで書き込まれているかを追跡把握し、それによって、そのキャッシュ・ページ内の初期化されていないデータはすべて、後でそのファイルがユーザ・モード・スレッドによってマップされたときにゼロ化することができる。
【０００８】
米国特許出願第２００８／２２９１１７Ａ１号による、コンピューティング環境におけるデジタル上の著作権侵害を防止するための方法は、アプリケーションをコンピューティング環境内にロードするステップであって、前記アプリケーションが、暗号化キーを使用して暗号化される、ステップと、仮想アドレス空間を前記アプリケーションに割り当てるステップと、前記アプリケーションのための前記暗号化キーを、中央処理装置によってのみアクセス可能であるレジスタ内にロードするステップと、前記キーのためのインデックス値を、前記アプリケーションのための前記仮想アドレス空間に対応するページ・テーブル・エントリー内の前記レジスタ内に格納し、それによって、前記仮想アドレス空間を前記アプリケーションのための前記キーにリンクさせるステップとを含む。
【０００９】
ＭＥＮＯＮ Ａ．らによる「Ｏｐｔｉｍｉｚｉｎｇ Ｎｅｔｗｏｒｋ Ｖｉｒｔｕａｌｉｚａｔｉｏｎ ｉｎ Ｘｅｎ」、ＰＲＯＣＥＥＤＩＮＧＳ ＯＦ ＴＨＥ ２００６ ＵＳＥＮＩＸ ＡＮＮＵＡＬ ＴＥＣＨＮＩＣＡＬ ＣＯＮＦＥＲＥＮＣＥ、２００６年５月２９日（２００６－０５－２９）、ＸＰ００２６２３６９９において、筆者らは、Ｘｅｎ仮想化環境においてネットワーク・パフォーマンスを最適化するための３つの技術を提案および評価している。彼らの技術は、デバイス・ドライバを、Ｉ／Ｏデバイスへのアクセスを有する特権のある「ドライバ」ドメイン内に配置し、仮想化されたネットワーク・インターフェースを通じて、特権のない「ゲスト」ドメインにネットワーク・アクセスを提供するという基本的なＸｅｎアーキテクチャーを保持している。
【先行技術文献】
【特許文献】
【００１０】
【特許文献１】米国特許第５９２０８９５Ａ号
【特許文献２】米国特許出願第２００８／２２９１１７Ａ１号
【非特許文献】
【００１１】
【非特許文献１】Ｌｕｄｍｉｌａ ＣｈｅｒｋａｓｏｖａおよびＲｏｂ Ｇａｒｄｎｅｒ、「Ｍｅａｓｕｒｉｎｇ ＣＰＵ Ｏｖｅｒｈｅａｄ ｆｏｒ Ｉ／Ｏ Ｐｒｏｃｅｓｓｉｎｇ ｉｎ ｔｈｅ Ｘｅｎ Ｖｉｒｔｕａｌ Ｍａｃｈｉｎｅ Ｍｏｎｉｔｏｒ」、Ｐｒｏｃｅｅｄｉｎｇｓ ｏｆ ｔｈｅ ＵＳＥＮＩＸ Ａｎｎｕａｌ Ｔｅｃｈｎｉｃａｌ Ｃｏｎｆｅｒｅｎｃｅ ２００５、アメリカ合衆国カリフォルニア州アナハイム
【非特許文献２】ＭＥＮＯＮ Ａ．らによる「Ｏｐｔｉｍｉｚｉｎｇ Ｎｅｔｗｏｒｋ Ｖｉｒｔｕａｌｉｚａｔｉｏｎ ｉｎ Ｘｅｎ」、ＰＲＯＣＥＥＤＩＮＧＳ ＯＦ ＴＨ

Ｅ ２００６ ＵＳＥＮＩＸ ＡＮＮＵＡＬ ＴＥＣＨＮＩＣＡＬ ＣＯＮＦＥＲＥＮＣＥ、２００６年５月２９日（２００６－０５－２９）、ＸＰ００２６２３６９９
【非特許文献３】ＩＥＥＥ８０２．３－２００８標準

【発明の概要】
【発明が解決しようとする課題】
【００１２】
本発明の目的は、ページ・フリッピングの際にゼロ化を行う必要性をなくすかまたは少なくする、仮想メモリ・マネージメントに対する改善されたアプローチを提示することである。さらなる目的は、限られている帯域幅のメモリに適した方法を提供することであり、すなわち、その速度でデータがＣＰＵによってメモリから読み出され、またはメモリに格納され得る。他のさらなる目的は、メモリ・バス、すなわち、メイン・メモリを、そのメイン・メモリに入るデータおよびそのメイン・メモリから出るデータの流れを管理するメモリ・コントローラに接続するコンピュータ・サブシステムに課される負荷を少なくすることにある。本明細書においては、メイン・メモリ（プライマリー・ストレージまたは内部メモリとも呼ばれる）とは、ランダムアクセス・メモリ（ＲＡＭ）、読み取り専用メモリ（ＲＯＭ）、プロセッサ・レジスタ、またはプロセッサ・キャッシュなど、ＣＰＵによって直接アクセス可能な任意のストレージを意味する。
【課題を解決するための手段】
【００１３】
この目的は、コンピュータ・メモリを管理する方法によって達成され、本方法は、仮想アドレスを物理アドレスにマップするためのページ・テーブル・エントリー、および複数のデータ・ブロックを含むキャッシュを保持するステップと、仮想アドレスへの参照に応答して、ページ・テーブル・エントリーを用いて仮想アドレスを物理アドレスへと変換するステップと、データを物理アドレスからキャッシュ内へフェッチするステップとを含み、ページ・テーブル・エントリーは、複数のインジケータを含み、それぞれのデータ・ブロックは、複数のインジケータのうちの１つに対応し、データをキャッシュ内へフェッチするステップが開始すると、本方法は、前記複数のインジケータから選択された１つのインジケータがセットされていることに応答して、対応するデータ・ブロックをゼロ化するさらなるステップを含む。この目的はさらに、コンピュータ・プログラム製品であって、プログラムがコンピュータ上で実行されたときに前記方法を実行するためのコンピュータ実行可能命令を含む、コンピュータ・プログラム製品によって、または前記方法を実行するようにプログラムもしくは構成されているデバイスによって達成される。
【００１４】
本発明の主要なアイディアは、仮想アドレス、すなわち、アプリケーションのワーキング・メモリ内のあるロケーションを表すインデックスと、対応する物理アドレスとの間におけるマッピングを格納するためにＯＳの仮想メモリ・システムによって使用されるデータ構造であるページ・テーブルを増強することである。仮想メモリのコンテキストにおいては、仮想アドレスは、アクセス・プロセスにとってのみ一意であり、その一方で物理アドレスは、メイン・メモリの実際のストレージ・セルを指す。
【００１５】
本発明のさらなる発展形態は、従属請求項および以降の説明から得ることができる。
【００１６】
以降では、添付の図面を参照して、本発明についてさらに説明する。
【００１７】
本発明の一実施形態に従ってコンピュータ・メモリを管理するために、仮想アドレスを物理アドレスにマップするためのページ・テーブル・エントリーが保持される。さらに、複数のデータ・ブロックを含むキャッシュが保持される。ページ・テーブル・エントリーは、複数のインジケータを含み、それぞれのデータ・ブロックは、１つのインジケータに対応する。仮想アドレスへの参照に応答して、その仮想アドレスは、ページ・テーブル・エントリーを用いて物理アドレスへと変換され、データが、その物理アドレスからキャッ

シュ内へフェッチされる。データをフェッチすると、インジケータがセットされていることに応答して、対応するデータ・ブロックがゼロ化される。

【図面の簡単な説明】

【００１８】

【図１】本発明の一実施形態による方法を示すフローチャートである。

【発明を実施するための形態】

【００１９】

以降では、図１を参照して、本発明による方法を例として説明する。

【００２０】

図１のフローチャート１００は、第１の処理ステップ１０１、第２の処理ステップ１０２、第３の処理ステップ１０３、および第４の処理ステップ１０４を含む。一連の矢印は、処理ステップ１０１から１０３を通って、内部ストレージ１１０と合流し、最終的に第４の処理ステップ１０４へとわたるコントロールのフローを表している。このフロー内では、第１のシンボルから始まって第２のシンボルで終わる矢印は、コントロールが第１のシンボルから第２のシンボルへわたることを示している。

【００２１】

目下の実施形態においては、本方法は、専用のメモリ・マネージメント・ユニット（ＭＭＵ）によって適用される。マイクロプロセッサの設計においては、ＭＭＵとは、ＣＰＵによって要求されるメモリへのアクセスを処理することを担当する任意のコンピュータ・ハードウェア・コンポーネントを意味する。仮想メモリ・マネージメント、すなわち、仮想アドレスを物理アドレスへ変換することのほかに、ＭＭＵは、メモリ保護、バス・アービトレーション、および、８ビット・システムなどのよりシンプルなコンピュータ・アーキテクチャーにおけるバンク切り替えなどの技術をサポートすることができる。

【００２２】

メモリは物理的に断片化されることがあり、二次ストレージ上へオーバーフローすることさえあるという事実の一方で、プロセスが、切れ目のないワーキング・メモリ（一般にはアドレス空間と呼ばれる）の概念に基づくことを可能にするために、ＭＭＵは、ページングをサポートし、ひいては、当技術分野においてページドＭＭＵ（ＰＭＭＵ）として知られている形式を取る。

【００２３】

ＣＰＵがメモリにアクセスするのに必要とする平均時間を少なくするために、ＰＭＭＵはさらに、ＣＰＵキャッシュ、すなわち、最も頻繁に使用されるメイン・メモリ・ロケーションからのデータのコピーを格納するように構成されている、より小さな、より高速なメモリをコントロールする。ＣＰＵは、メイン・メモリ内のあるロケーションからの読み取り、またはそのロケーションへの書き込みを行う必要がある場合には、まず、そのデータのコピーがキャッシュ内にあるかどうかをチェックする。そのデータのコピーがキャッシュ内にある場合には、ＣＰＵはすぐに、キャッシュからの読み取り、またはキャッシュへの書き込みを行い、これは、メイン・メモリからの読み取り、またはメイン・メモリへの書き込みよりも、本質的に高速である。キャッシングのコンテキストにおいては、メイン・メモリは、キャッシュのバッキング・ストアと呼ばれる場合がある。

【００２４】

より具体的には、ＰＭＭＵによってコントロールされるＣＰＵキャッシュは、データのフェッチおよび格納をスピード・アップするためのデータ・キャッシュ、ならびに、実行可能な命令およびデータの両方に関する仮想メモリ・マネージメントをスピード・アップするための変換ルックアサイド・バッファ（ＴＬＢ）を含む。ＴＬＢは、仮想アドレスを物理アドレスにマップするページ・テーブル・エントリー（ＰＴＥ）を含む固定された数のスロットを含む。典型的な一実施態様においては、それぞれのＰＴＥは、３２から６４の間のビットを含む。

【００２５】

第１の処理ステップ１０１において、ＰＭＭＵは、仮想アドレスを物理アドレスにマッ

プするためのＰＴＥを保持する。さらにＰＭＭＵは、複数のデータ・ブロックを含むキャッシュを保持し、それらのデータ・ブロックは一般に、ＣＰＵキャッシングのコンテキストにおいては、キャッシュ・ラインまたはキャッシュ・ブロックと呼ばれる。典型的に、それぞれのデータ・ブロックは、８から５１２バイトのサイズにわたる。ハードウェア・アクセラレーションを最大にするために、目下のＰＭＭＵは、現況技術の３２ビットおよび６４ビットのＣＰＵアーキテクチャーにおいては１バイトから１６バイトにわたることが典型的である単一のＣＰＵ命令によって要求される可能性があるデータの量よりも大きいデータ・ブロックを採用する。

【００２６】

第１の処理ステップ１０１において保持されているＰＴＥは、複数のインジケータを含み、それぞれのデータ・ブロックは、１つのインジケータに対応し、このインジケータは、その対応するデータ・ブロックを要求元のアプリケーションに提供する前にゼロ化する必要があるかどうかを示す。したがって、このインジケータは、フラグ、すなわち「真」または「偽」の値のいずれかを有するブール変数の形態を取る。ＰＴＥによって必要とされるストレージ容量を最小限に抑えるために、それぞれのインジケータは、ビットの形態を取り、そのバイナリー値のそれぞれは、割り当てられた意味を有する。ＣＰＵが複数のインジケータを単一のビット単位のオペレーション内にセットすることをさらに可能にするために、それらのビットは、コンピュータ計算においてはビットマスクとして知られている適切にサイズ設定されたベクトルへとグループ化される。代替実施形態においては、平均の書き込みスピードを犠牲にしてＰＴＥのサイズをさらに減らすために、それぞれのビットを、複数のデータ・ブロックに対応するように解釈して、ビットマスクを構成するビットの数を有効に減らすことができる。

【００２７】

中間のステップ（図示せず）において、ＯＳは、アプリケーションによる要求に応じて、ネットワーク・インターフェース・カード、ネットワーク・アダプタ、またはローカル・エリア・ネットワーク（ＬＡＮ）アダプタとしても知られている関連付けられたネットワーク・インターフェース・コントローラ（ＮＩＣ）を用いて、コンピュータ・ネットワークを介してデータ・ブロックを受け取る。スループットを最大にするために、データ・ブロックは、ギガビット・イーサネット（ＧｂＥ、１ＧｉｇＥ）接続を用いて、すなわち、ＩＥＥＥ８０２．３－２００８標準によって定義されているように１ギガビット／秒の速度で伝送される。ＣＰＵを用いてＩ／Ｏデータをコピーするというオーバーヘッドを避けることによってオペレーションをさらにスピード・アップするために、ＯＳは、ページ・フリッピングを採用して、アプリケーションによって指定されたメモリ・ページ内にそのＩ／Ｏデータを格納する。メモリ・ページを構成するデータ・ブロックのうちの一部のみが、Ｉ／Ｏオペレーションによって修正され、影響を受けない残りのデータ・ブロックは、すぐにはゼロ化されないと想定すると、ＯＳは、ゼロ化されるデータ・ブロックに対応する、ＰＴＥのビットマスク内のビットをセットし、その一方で、修正されたデータ・ブロックに対応するビットをクリアする。

【００２８】

イーサネットはパケット交換プロトコルであるため、Ｉ／Ｏデータは、パケット、またはこのコンテキストにおいてはイーサネット・フレームと呼ばれるデータ伝送単位の形態を取る。このフレームは、単なる確認応答に必要とされる場合などの６４バイトの下限と、コンピュータ・ネットワーキングの技術分野においては一般に最大伝送単位（ＭＴＵ）と呼ばれる９０００バイト以上の上限との間でさまざまなサイズとすることができる。４から６４キロバイトの典型的なメモリ・ページ・サイズを考慮に入れると、従来のメモリ・マネージメント・システムならば、このシナリオにおいて、それぞれのパケットを受け取った際に消去する必要があるメモリ・ページの多大な部分に起因して多大なゼロ化のオーバーヘッドをＣＰＵおよびメモリに課すことになる。したがって、本発明のさらなる利点は、小規模から中規模のパケット・データを受け取ることに対するその特有の適性にある。

【００２９】
ほとんどの場合において、ＮＩＣを通じて受け取られるデータの量は、ＰＴＥによって定義されたデータ・ブロックの固定サイズの倍数と一致しない可能性がある。そのような場合においては、Ｉ／Ｏによって影響を受ける最初のおよび／または最後のデータ・ブロックの少なくとも一部が、定義されないままとなる。その中に含まれているデータを要求元のアプリケーションに開示することを避けるために、ＯＳは、ページ・フリッピングの前に、データ・ブロックのうちのその残りの部分をすぐにゼロ化する必要がある。
【００３０】
代替実施形態においては、ＮＩＣを通じてデータを受け取る代わりに、プログラムをＣＰＵによって実行するためにメイン・メモリ内へロードするようＯＳに要求することができる。この場合には、実行ファイル全体の物理的な転送を避けるために、プログラム・ローダは、ページ・テーブルおよびＴＬＢを更新し、実行ファイル用に指定された仮想アドレスを、関連付けられたオブジェクト・ファイルのコンテンツにマップすることを有効に宣言する。このようにしてマップされた仮想メモリのセグメントは、メモリマップされたファイルと一般に呼ばれ、使用されていないプログラム・コードをメイン・メモリ内へロードする必要性をいっさいなくすことができるという利点を有する。
【００３１】
場合によっては、ロードされるプログラムは、ＢＳＳ（ｂｌｏｃｋ　ｓｔａｒｔｅｄ　ｂｙ　ｓｙｍｂｏｌ）として知られているセクションを含み、これは、ＢＳＳセグメントと呼ばれる場合が多く、はじめに、すなわち、プログラムの実行が開始するときに、ゼロの値のデータで満たされることを予期されている静的に割り当てられた変数を含む。この場合には、実行の前にＢＳＳセクション全体をゼロ化することを避けるために、ＯＳは、ゼロ化されるデータ・ブロックに対応する、ＰＴＥのビットマスク内のビットをセットし、その一方で、プログラムのその他のデータ・セグメントに対応するビットをクリアすることによって、同様に上述のメカニズムを採用する。
【００３２】
第２の処理ステップ１０２において、仮想アドレスがＣＰＵ命令によって参照されたことに応答して、ＰＭＭＵは、仮想アドレスを物理アドレスへと変換することによって、仮想メモリ・マネージメントというその主要な機能を果たす。このために、ＰＭＭＵは、第１の処理ステップ１０１のＴＬＢ内に含まれているＰＴＥを採用する。より詳細には、ＰＭＭＵは最初に、目下の仮想アドレスに対応するＰＴＥを探してＴＬＢを検索する。マッチするものが見つかると、すなわち、仮想メモリ・マネージメントの技術分野においてＴＬＢヒットとして知られている状況になると、そのマッチするＰＴＥから直接、物理アドレスを取り出すことができる。マッチするものが見つからないと、すなわち、一般にＴＬＢミスと呼ばれる状況になると、ＰＭＭＵには、ＯＳによって保持されているページ・テーブルを調べることが要求される。このページ・テーブルが、マッチするＰＴＥを含む場合には、ＰＭＭＵは、そのＰＴＥをＴＬＢ内にキャッシュし、失敗しているＣＰＵ命令を再実行する。
【００３３】
第３の処理ステップ１０３において、ＰＭＭＵは、第２の処理ステップ１０２において得られた物理アドレスにアクセスし、その中に含まれているデータ・ブロックをＣＰＵのデータ・キャッシュ内へフェッチし始める。データ・キャッシュがこの時点で既に満杯である場合には、占有されているキャッシュ・ラインを上書きする必要が生じることがあり、新たにフェッチされたデータ・ブロックを収容するために、その前のコンテンツを破棄する必要が生じることがある。どのキャッシュ・ラインの占有を解消するかを選択するためのさまざまなアルゴリズムが、当技術分野において知られている。
【００３４】
データ・キャッシュ上のいかなる書き込みオペレーションもスピード・アップするために、データ・キャッシュは、ライトバック・キャッシュ（ライトビハインド・キャッシュと呼ばれる場合もある）の形態を取る。このポリシーによれば、書き込みは、すぐにはバ

ッキング・ストアに反映されない。代わりに、ＰＭＭＵは、どのキャッシュ・ラインがＣＰＵによって上書きされているかを追跡把握し、それに応じて、それらのロケーションを「ダーティー」としてマークする。ダーティーなキャッシュ・ラインの占有を解消する前に、それぞれのデータ・ブロックがバッキング・ストアに書き戻されて、その更新されたコンテンツがその後の読み取りアクセスのために保存される。このアプローチは、コンピュータ・サイエンスにおいては「レイジー・ライト」として知られている。データ・ブロックを書き戻したら、ＰＭＭＵは、そのデータ・ブロックがいつか取り出された場合にゼロ化されることを防止するためにＰＴＥのビットマスク内の対応するビットをクリアする必要がある。

【００３５】

それぞれのデータ・ブロックをフェッチするために、空いているキャッシュ・ラインが識別されると、または占有されているキャッシュ・ラインが解放されると、ＰＭＭＵは、自分の内部ストレージ１１０内で見つかったＰＴＥ内に含まれているビットマスクを調べる。その対応するビットがセットされている場合には、ＰＭＭＵは、メイン・メモリからデータ・ブロックを取り出さずに、第４の処理ステップ１０４においてそれぞれのキャッシュ・ラインをゼロ化してから、そのデータ・ブロックを要求元のアプリケーションが利用できるようにする。

【００３６】

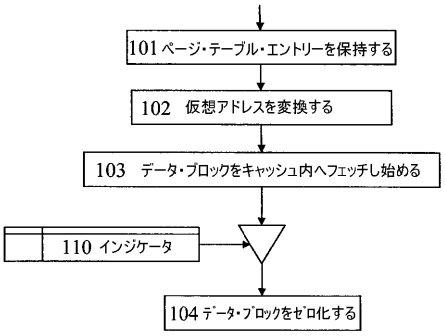
上述のさまざまな方法のステップは、プログラムされたコンピュータによって実行することができるということを当業者なら容易に認識するであろう。本明細書においては、いくつかの実施形態は、プログラム・ストレージ・デバイス、たとえばデジタル・データ・ストレージ・メディアをカバーするように意図されており、それらのプログラム・ストレージ・デバイスは、マシンまたはコンピュータによって読み取ることができ、マシンによって実行可能なまたはコンピュータによって実行可能な命令のプログラムをエンコードし、前記命令は、本明細書に記載の方法のステップのうちのいくつかまたはすべてを実行する。それらのプログラム・ストレージ・デバイスは、たとえば、デジタル・メモリ、磁気ディスクもしくは磁気テープなどの磁気ストレージ・メディア、ハード・ドライブ、または光学的に読み取り可能なデジタル・データ・ストレージ・メディアとすることができる。これらの実施形態はまた、本明細書に記載の方法の前記ステップを実行するようにプログラムされたコンピュータをカバーするように意図されている。

【００３７】

本発明は、その他の特定の装置および／または方法で具体化することもできる。記載した実施形態は、すべての点で例示的なものにすぎず、限定的なものではないとみなすべきである。とりわけ、本発明の範囲は、本明細書に記載の説明および図ではなく、添付の特許請求の範囲によって示される。特許請求の範囲の均等物の意味および範囲内に収まるすべての変更は、特許請求の範囲内に包含される。

【図１】

100


101 ページ・テーブル・エントリーを保持する
102 仮想アドレスを変換する
103 データ・ブロックをキャッシュ内へフェッチし始める
110 インジケータ
104 データ・ブロックをゼロ化する

フロントページの続き

(72)発明者 マレンダー, セイプ
ベルギー ベー－２６００ アントワープ, アンジャベルシュトラート １５
(72)発明者 マッキー, ジェームス バルマー
アメリカ合衆国 ０７９７４ ニュージャーシィ, マレー ヒル, チェスナットヒル ドライヴ １５
(72)発明者 ピアネーゼ, ファビオ
ベルギー ベー－１０００ ブリュッセル, ル ドゥ ラ マドレーヌ ６１
(72)発明者 エヴァンス, ノア
ベルギー ベー－２０００ アントワープ, ズワールドシュトラート １４

審査官 野田 佳邦

(56)参考文献 特開平０６－２０８５１１（ＪＰ，Ａ）
米国特許第０５９２０８９５（ＵＳ，Ａ）
特開平０２－１７６９５１（ＪＰ，Ａ）
特開２００８－２７６７７８（ＪＰ，Ａ）
Xiaolan Zhang, Suzanne McIntosh, Pankaj Rohatgi, and John Linwood Griffin , XenSocket: A High-Throughput Interdomain Transport for Virtual Machines, Middleware '07 Proceedings of the ACM/IFIP/USENIX 2007 International Conference on Middleware, ２００７年, p. 184-203

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ １２／０８－１２／１２
Ｇ０６Ｆ １２／１４