

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2008-505389

(P2008-505389A)

(43) 公表日 平成20年2月21日(2008.2.21)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 12/12 (2006.01)	G06F 12/12 553C	5B005
G06F 9/46 (2006.01)	G06F 9/46 350	
G06F 12/10 (2006.01)	G06F 12/10 553B	
G06F 12/08 (2006.01)	G06F 12/08 543B	
	G06F 12/08 565	
審査請求 未請求 予備審査請求 未請求 (全 14 頁)		

(21) 出願番号 特願2007-519201 (P2007-519201)
 (86) (22) 出願日 平成17年5月4日 (2005.5.4)
 (85) 翻訳文提出日 平成19年2月19日 (2007.2.19)
 (86) 国際出願番号 PCT/US2005/015493
 (87) 国際公開番号 W02006/007043
 (87) 国際公開日 平成18年1月19日 (2006.1.19)
 (31) 優先権主張番号 10/881, 508
 (32) 優先日 平成16年6月30日 (2004.6.30)
 (33) 優先権主張国 米国 (US)

(71) 出願人 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレーション
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 アメリカ合衆国10504 ニューヨーク州 アーモンク ニュー オーチャードロード
 (74) 代理人 100108501
 弁理士 上野 剛史
 (74) 代理人 100112690
 弁理士 太佐 種一
 (74) 代理人 100091568
 弁理士 市位 嘉宏

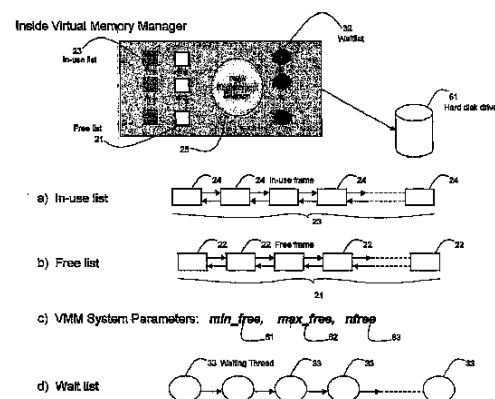
最終頁に続く

(54) 【発明の名称】 コンピュータ・オペレーティング・システムの仮想メモリのサブシステムを自動調整する方法、プログラム記憶デバイス、および装置

(57) 【要約】

【課題】 コンピュータ・システムのメイン・メモリをそのオペレーティング・システムのインスタンスで実行するアプリケーション間に効率的に配分する方法、情報処理システム、およびコンピュータ読み取り可能媒体を提供する。

【解決手段】 仮想メモリ・マネージャのページ置換アルゴリズムで使用する閾値を、コンピュータ・システムのメモリ負荷に応答して自動的に調整する。ある前記閾値は、メモリ負荷の関数として変更する空きメモリの下限閾値である。例えば、このような負荷を、所定の時間間隔の間に待ち行列に加えられたスレッドの数を、その間隔のクロック・ティックの数で割った値で表すことができるであろう。この値はスレッド待機率として知られる。さらにこの率を目標率と比較して、下限閾値の値を変更するべきかどうかを判断する。空きメモリ空間が下限閾値未満になったら、ページ置換デーモンを使ってメモリをページアウトして、利用できるメモリ空間を増やす。



【選択図】 図2

【特許請求の範囲】**【請求項 1】**

コンピュータ・システムのメモリの可用性を管理する方法で、前記方法が、
空きメモリ空間の下限閾値をメモリ負荷の関数として自動的に変更するステップと、
空きメモリ空間が前記下限閾値未満になったときに、利用できるメモリ空間を増やすステップとを有する、方法。

【請求項 2】

コンピュータ・システムのメモリの可用性を管理する方法で、前記方法が、
スレッド待機率が目標スレッド待機率と差が出てきたときに、空きメモリ空間の下限閾値を自動的に変更するステップで、前記スレッド待機率が単位時間あたりの空きメモリの待機リストに待機しているスレッドの平均数である、前記ステップと、
空きメモリ空間が前記下限閾値未満になったときに、利用できるメモリ空間を増やすステップとを有する、方法。

10

【請求項 3】

前記スレッド待機率が前記目標スレッド待機率よりも高くなったときに、前記下限閾値を上げる、請求項 2 に記載の方法。

【請求項 4】

前記スレッド待機率が前記目標スレッド待機率よりも低くなったときに、前記下限閾値を下げる、請求項 2 に記載の方法。

【請求項 5】

前記スレッド待機率が前記目標スレッド待機率よりも高くなったときに、上限閾値を上げ、前記上限閾値を使って、ページ置換デーモンを実行するとき、利用できるようにするメモリ空間の量を判断する、請求項 2 に記載の方法。

20

【請求項 6】

前記スレッド待機率が前記目標スレッド待機率よりも低くなったときに、上限閾値を下げ、前記上限閾値を使って、ページ置換デーモンを実行するとき、利用できるようにするメモリ空間の量を判断する、請求項 2 に記載の方法。

【請求項 7】

前記スレッド待機率を、空きメモリの待機リストに待機している全スレッドが費やすクロック・ティックの累積数をカウントして、前記累積数を前記ページ置換デーモンの連続 2 回の実行の間のクロック・ティックの総数で割って計算できる、請求項 5 に記載の方法。

30

【請求項 8】

前記スレッドが現在前記空きメモリの待機リストにある第 1 スレッドと、前記ページ置換デーモンの前記連続 2 回の実行の最初の実行の後に前記空きメモリの待機リストにあったが、もはや前記空きメモリの待機リストにない第 2 スレッドとを有する、請求項 7 に記載の方法。

【請求項 9】

前記スレッド待機率を、空きメモリの待機リストに現在あるスレッドの数をある数で割って計算できる、請求項 2 に記載の方法。

【請求項 10】

前記数が整数の 2 である、請求項 9 に記載の方法。

40

【請求項 11】

空きメモリ空間が前記下限閾値未満になったときにページ置換デーモンを実行し、前記ページ置換デーモンが利用できるメモリ空間を増やす、請求項 2 に記載の方法。

【請求項 12】

空きメモリ・フレームの数が下限閾値未満になったときに前記ページ置換デーモンを実行し、前記ページ置換デーモンが、フリー・フレームの数が上限閾値に達するようにフレームの数を解放するステップを有する、請求項 11 に記載の方法。

【請求項 13】

コンピュータ・システムのメモリの可用性を管理するためのプログラムであって、該プロ

50

グラムが、

空きメモリ空間の下限閾値をメモリ負荷の関数として自動的に変更するステップと、

空きメモリ空間が前記下限閾値未満になったときに、利用できるメモリ空間を増やすステップとをコンピュータに実行させるプログラム。

【請求項 14】

コンピュータ・システムのメモリの可用性を管理する装置で、前記装置が、

空きメモリ空間の下限閾値をメモリ負荷の関数として自動的に変更する手段と、

空きメモリ空間が下限閾値未満になったときに、利用できるメモリ空間を増やす手段とを有する、装置。

【発明の詳細な説明】

10

【技術分野】

【0001】

本発明は、コンピュータ・オペレーティング・システムの仮想メモリの管理コンポーネントに関する。さらに具体的には、本発明はオペレーティング・システムの仮想メモリ・マネージャのあらゆるページ置換アルゴリズムで使用する閾値の調整に関する。

【背景技術】

【0002】

マシン上で実行するオペレーティング・システム（OS）の仮想メモリ・マネージャ（VMM）コンポーネントは、マシンのメイン・メモリをそのOSインスタンスで実行するアプリケーション間に効率的に配分する役割をする。VMMの主な役割の1つが、十分に利用されていないメイン・メモリのブロックのコンテンツ（「フレーム」もしくは「ページ・フレーム」という）をディスク上のページング・スペースにページアウトすることと、そのフレームをメイン・メモリを必要とする別のアプリケーションに再割り当てすることである。これは一般的には、「ページ置換デーモン」（ほとんどのUNIXオペレーティング・システムでは「LRUデーモン」ともいう）と呼ばれるデーモン・プロセスの助けを借りて行う。

20

【0003】

フレームを解放するプロセス（すなわち、そのコンテンツをディスクに移動してフリー・フレームにする行為）はフリー・フレームを要求するアプリケーション（フリー・フレームの使用者）に割り当てるプロセスよりもかなり時間が長くなるため、ページ置換デーモンはOSのフリー・フレームが余分に必要になることを見越して、OSのフリー・フレームの数がゼロになる前に、フレームのページアウトを開始するのが一般的である。VMMはページ置換デーモンを開始する時期と各実行で解放すべきページ数を、2つの調整可能なパラメータmin__freeおよびmax__freeを使って決定できる。ページ置換デーモンはフリー・フレームの数がmin__free未満になるとすぐに開始し、各実行の最後にフリー・フレームの数がmax__freeに達するように十分なページを解放する。

30

【発明の開示】

【発明が解決しようとする課題】

【0004】

40

現在、これらのパラメータは、VMMのパフォーマンスを調整して、OSで実行するアプリケーションのニーズに合わせるために、システム・アドミニストレータが明確に入力しなければならない。この調整には人手による入力が必要なため、これらのパラメータが調整されることはほとんどなく、その結果VMMは、そのためOSも次善のパフォーマンスしか得られていない。この調整不足によりIT組織のコストがかかる。

【課題を解決するための手段】

【0005】

本発明の目的は、システム・アドミニストレータによるVMMの手作業による調整の必要をなくして、システムのパフォーマンスを改善することである。本発明の重要な利点は、OSがその作業負荷の変動にはるかに反応性/適応性があることである。さらに具体的

50

には、本発明はVMMシステム・パラメータの調整を、OSのメモリ負荷の変動に応答してその値を自動的に変えることによって自動化する。

【0006】

本発明は、OSのメモリ負荷に応答して、下限閾値として知られるパラメータを自動的に変更することによって、OSのメモリの可用性を高める方法を提供する。現在の空きメモリ空間が下限閾値未満になるとき、より多くの空きメモリ空間を作る。

【発明を実施するための最良の形態】

【0007】

本発明に関する主題は、本明細書の前にある請求項で具体的に指摘し、かつ個別に請求する。本発明の前述並びにその他の特徴および利点は、添付の図面と合わせて以下の詳細な説明を読めば明らかになるであろう。

【0008】

本発明のより具体的な好適な実施例は、空きメモリ空間の下限閾値を初期値に設定し、現在の「スレッド待機率」が目標の「スレッド待機率」と異なるときに、この下限閾値を自動的に変更することによって、OSのメモリ・マネージャを自動的に調整する方法を提供する。「スレッド待機率」とは、規定の時間間隔における単位時間あたりのスレッド待ちの数である。さらに、メモリ・マネージャは、空きメモリ空間が下限閾値未満になるときに利用できるメモリ空間を増やす操作を開始する。

【0009】

図1を参照すると、代表的なオペレーティング・システム・カーネル10は、仮想メモリ・マネージャ(VMM)20、プロセス・マネージャ30、ファイルシステム40、ネットワーク・サブシステム50等の複数のコンポーネントからなる。ファイルシステム40はハードディスク・ドライブへの構造的なアクセスを提供する。プロセス・マネージャはプロセスのスケジュールを管理して、プロセッサにスレッドを割り当て、一方ネットワーク・サブシステムはアプリケーションが他のマシンまたはコンピュータと通信するのを可能にする。本発明は、オペレーティング・システム(OS)の仮想メモリ・マネージャ・コンポーネントに関係しており、他のコンポーネントは技術上よく知られているため本出願では説明しない。図2は、VMM20がシステムのメモリの管理を維持するデータ構造の一部を示す。VMM20は、フリー・ページ・フレーム22のトラックを実メモリに保持する一又は複数のフリー・リスト21を維持する。VMM20はフリー・フレームの総数も変数`nfree63`に維持する。フリー・ページの要求がきたら、VMM20は`nfree`が >0 かどうかをチェックし、もしそうなら、そのフリー・リスト21を検索してフリー・ページを確保する。`nfree63`の値が下限閾値`min__free61`未満の場合、VMM20はページ置換デーモン(LRUデーモンとしても知られる)25を呼び出して、ページアウトを開始し、使用中のページ・フレーム23のコンテンツをディスク51に退避させることによってフリー・ページ・フレーム22を作成する。ここで説明する実施例では、このページアウトは、ページ置換デーモンと呼ばれる個別のプロセスで行う。ページ置換デーモンは、その実行の最後に、`nfree`が`max__free`以上になるような十分な数のページ・フレームをページアウトする。フリー・ページ・フレームの要求が満たされないものはすべて、図2に図示する待機リスト32に入れられる。フリー・ページ・フレームが後で利用できるようになったら、要求するスレッド33がウェイクアップされる。

【0010】

ページ置換デーモンが使用する高次のアルゴリズムを図3に示す。ブロック101は、`nfree < min__free`のときのページ置換デーモンの呼び出しを表す。102で、ページ置換デーモンは、`max__free61`から`nfree63`を引いて、ページアウトするページ数を計算する。103で、使用中のページ・フレーム・リスト23をスキャンして、退避させるのに適切な候補を見つける。ページ置換デーモンが退避させるページの候補を選択するのにメモリのスキャンを開始したら、どのページをページアウトするかを決めるにはいくつかのポリシーがある。本発明は退避させるページの候補選びに特に

10

20

30

40

50

決まったポリシーに依存しない。102で計算したページ数を退避させたら、104でページ置換デモンは`nfree`がまだ`max__free`未満かどうかをもう一度チェックする。ページ・フレームを解放した直後に消費される場合には、そうなる可能性があるからである。104の述語が真なら、分岐線106に進んで、ステップ102を再開する。そうでなければ、分岐線105に進んで、ページ置換デモンは107でスリープに戻る。

【0011】

上記説明によると、VMMのページ置換デモンの最終目標は、待機リスト32に過多のフリー・ページを要求するスレッド33をもつコストと、過多の使用中のページ24を早期に退避させるコストとのバランスをとることである。本発明は、パラメータ`min__free61`および`max__free62`の最適値を判断して、これらパラメータをフリー・フレームの変動を待っているスレッドの数に応じて継続的に調整するメカニズムを提供することによって、この問題を解決する。

10

【0012】

図4は、本発明の実装に必要な好適な実施例のデータ構造を示す。要求するスレッドの数と、待機リストで各要求するスレッドがフリー・フレームを待つ時間数を継続的にモニターする必要がある。これは変数`thrd__wait35`を維持することによって行う。

【0013】

`thrd__wait35`の値は次のとおり更新する。スレッド33が待機リスト32に入る度に、オペレーティング・システムはスレッドがスレッドあたりの変数`waitlist__enqueue__time36`で待ち行列に入っている時間を表す。フリー・ページが利用できるようになってそのスレッドが最終的に待ち行列から出されたら、OSは現在時刻から`waitlist__enqueue__time36`を引いて、スレッドが待機リストで費やした合計時間を出す。次に、OSはこの合計時間をクロック・ティックに変換して、それに`thrd__wait35`カウンタに追加すると、カウンタは待機リストですべてのスレッドが合わせて消費したクロック・ティックの合計時間数に相当する。このカウンタは、図5で説明するようにページ置換デモンでリセットされる。そのため、`thrd__wait35`カウンタは、待ち行列に入れられて、後で、ページ置換デモンの2つの実行の間のある時点で待機リストから出されたすべてのスレッドの合計待ち時間に相当する。ページ置換デモンは別の変数`thrd__wait__rate37`を維持するが、これはクロック・ティックあたりの待機スレッドの平均数である。これはクロック・ティックあたりのカウント単位で維持されるため、この値はCPUの周波数が異なる様々なシステム全体で統一できる。ページ置換デモンが始動する度に、すべてのスレッドが最後の実行以降待機に費やした合計時間数の累計と、2つの実行の間に含まれるクロック・ティック数との比として、`thrd__wait__rate37`の値を計算する。ページ置換デモンは別の変数`strt__time39`も維持し、それを使って前述した2つの実行の間に含まれるクロック・ティック数を計算する。最後に、ページ置換デモンは図4に示す`thrd__wait__rate__tgt38`の値を使って、`min__free`および`max__free`をどれくらい変更しなければならないかを計算する。

20

30

【0014】

図5に示すフローチャートは、ページ置換デモンの新しい機能をまとめている。ブロック201は、`nfree < min__free`の場合のページ置換デモンの呼び出しを表す。202で、ページ置換デモンは、その最後の実行以降の経過時間と、すべてのスレッドが最後の実行以降待機リストで費やしたクロック・ティックの合計時間数を計算する。経過時間は、システムの現在時刻を読みとって、以前に記録された`strt__time38`の値を引いて計算する。すべてのスレッドが最後の実行から待機している合計時間数は、2つの部分からなる。最初の部分は、現在待機リストに入っていないスレッドの合計待ち時間である。これはOSにより`thrd__wait`カウンタに維持される。2つめの部分は、すでに待機リストにあって待ち状態で費やしているスレッドの合計時間である。ページ置換デモン25は、待機リスト32をウォークスルーし、現在時刻から各スレ

40

50

ットのwaitlist_enqueue_time36を引いて、すべての待ち時間を合計することによって、この2つめの部分を計算する。待機リストに入るまたは出るすべてのスレッドの合計待ち時間は、上記値をthrd_waitカウンタ35に足して計算する。

【0015】

すべてのスレッドの合計待ち時間を計算する代替法は、クロック・ティックごとに1回待機リストにポーリングして待機リストのスレッドの数をカウントし、それをthrd_waitカウンタに加える。この場合、ページ置換デーモンは各実行の始めに待機リストをウォークスルーする必要はない。欠点は、OSがクロック・ティックの度に追加作業をする必要があるため、かなりのオーバーヘッドになりうることである。

10

【0016】

ページ置換デーモンが202で経過時間と合計スレッド待ち時間を計算するとき、タイムスタンプ値strt_timeとスレッドごとのwaitlist_enqueue_timeも、これら変数を読み込んだ直後のシステムの現在時刻にリセットする。thrd_wait_timeも0にリセットして、このカウンタが今から待機リストに行く全スレッドの待ち時間に対応させるようにする。203で、ページ置換デーモンは、合計スレッド待ち時間を経過時間で割って、thrd_wait_rate36を計算する。204で、以下に示すように、上記計算したthrd_wait_rateと予め設定した目標値thrd_wait_rate_tgtとの差に基づいて、min_free61およびmax_free62の値を再計算する。

20

【0017】

望ましいmin_free値は、thrd_wait_rateがthrd_wait_rate_tgtよりも高い場合に増やし、thrd_wait_rateがthrd_wait_rate_tgtよりも低い場合に減らす。実施例では、望ましいmin_freeは以下に示すようにメモリ負荷の関数として計算する。

【0018】

望ましいmin_free = min_free × thrd_wait_rate / thrd_wait_rate_tgt

【0019】

min_freeパラメータは、望ましいmin_free値と現在値の平均として更新する。この平均化は、作業負荷の突発的な急増による振幅に対するある程度の減幅を提供する。スラッシングを避けるために、ページ置換デーモンによって更新できるmin_free値に上限を課すこともできる。

30

【0020】

max_freeパラメータは、このmin_freeの更新前と同じmax_freeとmin_freeとの差を維持するように更新する。

【0021】

例：thrd_wait_rate_tgtを1に設定し、thrd_wait_rateの計算値が1.5で、min_freeおよびmax_freeがそれぞれ120および128と仮定する。望ましいmin_freeは、 $120 \times 1.5 / 1 = 180$ となる。ページ置換デーモンはmin_freeを、望ましいmin_freeと現在のmin_freeとの平均に変更するので、 $120 + 180 / 2 = 150$ である。max_freeの新たな値は、 $150 + (128 - 120) = 158$ となる。

40

【0022】

前述した単純な線形近似の代わりに、thrd_wait_rateの偏差から望ましいmin_free値を計算するいくつかの代替法も使用できる。望ましいmin_free値を計算するのに使う方法は、min_freeを、thrd_wait_rate > thrd_wait_rate_tgtの場合に増やし、thrd_wait_rate < thrd_wait_rate_tgtの場合に減らすという一般原則を守るべきである。

50

【0023】

min__freeおよびmax__freeの値を再計算した後、ページ置換デモンのステップの残り、205、206、207、208、209、210は、それぞれ図3の102、103、104、105、106、107のステップと同様である。より具体的には、205で、ページ置換デモンは、max__free61からnfree63を引いて、ページアウトするべきページ数を計算する。206で、使用中のページ・フレーム・リストをスキャンして、退避させるのに適切な候補を見つける。ページ置換デモンが退避させるページの候補を選択するためにメモリをスキャンし始めたら、どのページをページアウトするかを決めるポリシーはいくつかある。本発明は、退避させるページの候補選に特に決まったポリシーに依存しない。205で計算したページ数を退避させたら、ページ置換デモンは、207でまだnfreeがmax__free未満かどうかを再びチェックする。ページ・フレームを解放した直後に消費されたら、そうなる可能性があるためである。207の述語が真の場合、分岐線209に進んで、ステップ205を再び始める。そうでなければ、分岐線208に進んで、210でページ置換デモンはスリープに戻る。

10

【0024】

図6は、時間線図を使って、thrd__wait__rateをどのように計算するかを例証する。

【0025】

矢印381は壁時計時刻の基準線を表す。381上で、3個のタイムスタンプts₁、ts₂、ts₃は、ページ置換デモンの3回の実行の始まりを表す。時間軸に示される双頭の矢印は、ページ置換デモン25の各実行が完了するのにかかる時間数を表す。この図から、ページ置換デモンの各実行は完了するのにかかる時間数が異なることが分かる。また、ページ置換デモンの連続する2つのインスタンスの間の経過時間は定まっていない。各タイムスタンプのTW_n値は、ページ置換デモンの最後の実行から全スレッドが待ち行列で費やす合計時間数を表す。TW_n値は、前の段落で説明したように、待機リストにある各スレッドのthrd__waitカウンタとwaitlist__enqueue__timeを使って計算する。

20

【0026】

実施例では、様々なパラメータは以下に示すようにシステム初期化のときに初期設定しなければならない。

30

- ・min__free61およびmax__free62はあるデフォルト値に設定する。
- ・thrd__waitカウンタ35およびthrd__wait__rate37は0に初期設定する。
- ・thrd__wait__rate__tgt38は特定の値に初期設定する。
- ・strt__time39は現在時刻に初期設定する。

【0027】

上記の実施例で説明するように、たとえthrd__wait__rate値を正確に維持しなくても、本発明を実施できることに注目すべきである。本出願で説明する好適な実施例では、ページ置換デモンは、呼び出される度に待機リスト全体をウォークスルーすることによって、thrd__wait__rateの正確な値を計算する。thrd__wait__rateの概算値を計算することによっても本発明を実施することはでき、それによってメモリの可用性に対する影響を著しく減じることなく、実装の複雑さを減らすことができる。以下の段落で、thrd__wait__rateを計算する代替法を2つ説明する。

40

【0028】

1) システムはthrd__waitカウンタに加えて、2つの変数nthrds__wait__edおよびnthrds__waitingを維持できる。nthrds__wait__edは、thrd__waitの値に寄与したスレッドの数に相当する。nthrds__waitingは、現在待機リストにあるスレッドの数に相当する。これら変数は共に、1つ

50

のスレッドが待機リストからなくなる度に更新される。`nthrds_waiting`は、1つのスレッドが待機リストに入れられるときにも更新される。これらの変数があれば、ページ置換デモンは以下のように`thrd_wait_rate`を計算できる。

【0029】

```
thrd_wait_rate = (thrd_wait + ((thrd_wait / nthrds_waited) * nthrds_waiting)) / (current_time - strt_time)
```

【0030】

この方法を使うと、各スレッドの`waitlist_enqueue_time`を維持する必要はなくなる。

10

【0031】

2) 待機リストから外されたスレッドを無視すれば、予測はさらに簡略化できる。現在待機リストにないスレッドを無視して、待機リストにあるスレッドが均一な時間間隔で待ち行列に入れられたと仮定すると、`thrd_wait_rate`は単純に`nthrds_waiting / 2`として計算できる。これは以下のように導くことができる。

【0032】

待機リストにある最初のスレッドが時間 T_1 で待ち行列に入れられ、現在時刻が T_2 だと仮定する。スレッドは均一な時間間隔で待機リストに入れられたと仮定しているため、平均して各スレッドは $(T_2 - T_1) / 2$ の時間数分待機している。

【0033】

現在待機リストにある全スレッドの合計待ち時間 = $(nthrds_waiting \times (T_2 - T_1) / 2)$

20

【0034】

経過時間 = $(T_2 - T_1)$

【0035】

```
thrd_wait_rate = (nthrds_waiting * (T2 - T1) / 2) / (T2 - T1) = nthrds_waiting / 2
```

【0036】

図7は、本発明の実施例を実装するのに使える情報処理システムを示す高次のブロック図である。コンピュータシステムは、プロセッサ704などの一又は複数のプロセッサを含む。プロセッサ704は通信インフラストラクチャ702(例、通信バス、クロスオーバー・バー、またはネットワーク)に接続する。様々なソフトウェア実施例がこの例示的なコンピュータ・システムに関して考えられる。本明細書を読めば、関連技術の当業者には、他のコンピュータ・システムおよびコンピュータ・アーキテクチャもしくは他のコンピュータ・システムまたはコンピュータ・アーキテクチャを使って本発明を実施する方法が明らかになるであろう。

30

【0037】

コンピュータ・システムは、ディスプレイ装置710に表示するために、通信インフラストラクチャ702(もしくは図示していないフレーム・バッファ)から画像、テキスト、および他のデータを送るディスプレイ・インターフェース708を含むことができる。コンピュータ・システムはメイン・メモリ706、好ましくはランダム・アクセス・メモリ(RAM)も含み、また二次メモリ712を含んでもよい。二次メモリ712は、例えば、ハードディスク・ドライブ714および取り外し可能ストレージ・ドライブ716もしくはハードディスク・ドライブ714または取り外し可能ストレージ・ドライブ716を含んでもよく、取り外し可能ドライブの代表的なものとしてフレキシブルディスク・ドライブ、磁気テープ・ドライブ、光ディスク・ドライブ等がある。取り外し可能ストレージ・ドライブ716は、当業者に周知の方法で、取り外し可能記憶装置718から読み取りかつ取り外し可能記憶装置718に書き込み、もしくは取り外し可能記憶装置718から読み取りまたは取り外し可能記憶装置718に書き込む。取り外し可能記憶装置718

40

50

の代表的なものとしてフレキシブルディスク、コンパクト・ディスク、磁気テープ、光ディスク等があり、取り外し可能ストレージ・ドライブ 716 によって読み取り、書き込まれる。言うまでもなく、取り外し可能記憶装置 718 は、内部にコンピュータ・ソフトウェアおよびデータもしくはコンピュータ・ソフトウェアまたはデータを記憶したコンピュータ読み取り可能媒体を含む。

【0038】

代替実施例では、二次メモリ 712 は、コンピュータ・プログラムまたは他の命令をコンピュータ・システムにロードさせる他の同様な手段を含んでもよい。前記手段には、例えば、取り外し可能記憶装置 722 およびインターフェース 720 がある。前記例は、プログラム・カートリッジおよびカートリッジ・インターフェース（ビデオゲーム機にあるようなものなど）、取り外し可能メモリチップ（EPROM または PROM など）および付属ソケット、ならびにその他取り外し可能記憶装置 722 およびソフトウェアとデータを取り外し可能記憶装置 722 からコンピュータ・システムに転送できるインターフェース 720 を含んでもよい。

【0039】

さらに、コンピュータ・システムは通信インターフェース 724 を含んでもよい。通信インターフェース 724 は、ソフトウェアおよびデータをコンピュータ・システムと外部デバイスとの間で転送できる。通信インターフェース 724 の例には、モデム、ネットワーク・インターフェース（イーサネット・カードなど）、通信ポート、PCMCIA スロットおよびカード等がある。通信インターフェース 724 経由で転送されるソフトウェアおよびデータは、例えば、電子信号、電磁信号、光信号、または通信インターフェース 724 が受信できるその他の信号のような信号の形態である。これら信号は通信路（すなわち、チャネル）726 経由で通信インターフェース 724 に供給される。このチャネル 726 は信号を伝送するが、ワイヤまたはケーブル、光ファイバー、電話回線、携帯電話リンク、RF リンク、およびその他の通信チャネル、もしくはワイヤまたはケーブル、光ファイバー、電話回線、携帯電話リンク、RF リンク、またはその他通信チャネルを使って実施できよう。

【0040】

本明細書で、「コンピュータ・プログラム媒体」、「コンピュータ使用可能媒体」、および「コンピュータ読み取り可能媒体」という表現は、一般的にメイン・メモリ 706 および二次メモリ 712、取り外し可能ストレージ・ドライブ 716、ハードディスク・ドライブ 714 にインストールされたハードディスク、および信号などの媒体を言うために使う。これらコンピュータ・プログラムは、ソフトウェアをコンピュータ・システムに提供する手段である。コンピュータ読み取り可能媒体は、コンピュータ・システムにデータ、命令、メッセージまたはメッセージ・パケット、並びにその他コンピュータ読み取り可能な情報をコンピュータ読み取り可能媒体から読み取らせることができる。コンピュータ読み取り可能媒体には、例えば、フレキシブルディスク、ROM、フラッシュメモリ、ディスク・ドライブ・メモリ、CD-ROM、およびその他永久記憶装置などの不揮発性メモリがある。例えば、データおよびコンピュータ命令などの情報を、コンピュータ・システム間で転送するのに便利である。さらに、コンピュータ読み取り可能媒体は、コンピュータにコンピュータ読み取り可能な情報を読み取らせることのできる、有線ネットワークまたは無線ネットワークなど、ネットワーク・リンクおよびネットワーク・インターフェースもしくはネットワーク・リンクまたはネットワーク・インターフェースなどの一時的な状態の媒体のコンピュータ読み取り可能情報を含んでもよい。

【0041】

コンピュータ・プログラム（コンピュータ制御ロジックともいう）は、メイン・メモリ 706 および二次メモリ 712 もしくはメイン・メモリ 706 または二次メモリ 712 に格納される。コンピュータ・プログラムは通信インターフェース 724 経由で受信してもよい。前記コンピュータ・プログラムは、実行時、コンピュータ・システムに本明細書で説明した本発明の特徴を実施させることができる。具体的には、コンピュータ・プログラ

10

20

30

40

50

ムは、実行時、プロセッサ 704 にコンピュータ・システムの特徴を実施させることができる。従って、前記コンピュータ・プログラムはコンピュータ・システムのコントローラに相当する。

【0042】

本発明の特定の実施例を開示してきたが、当業者は本発明の精神および範囲を逸脱することなくこの特定の実施例に変更を行えることは理解するであろう。従って、本発明の範囲はこの特定の実施例に制限するつもりはない。

【0043】

さらに、添付の請求項は本発明の範囲内にあるあらゆる前記応用、変更、および実施例を含むことが意図される。

【図面の簡単な説明】

【0044】

【図1】代表的なコンピュータ・オペレーティング・システムの主要なサブシステムを示す図。

【図2】現在のページ置換アルゴリズムを実装するのに使用するデータ構造を示す図。

【図3】現在のページ置換アルゴリズムのフローチャートを示す図。

【図4】ページ置換デモンの動的な自動調整を実装するために、好適な実施例で使用するデータ構造を示す図。

【図5】本発明による新規のページ置換アルゴリズムのフローチャートを示す図。

【図6】LRUデモンを使ってどのようにmin__freeおよびmax__freeを時間について変更するかを示す図。

【図7】本発明の一又は複数のコンポーネント／方法論を実装できるコンピューティング・システムの例示的なハードウェアの実装図。

【符号の説明】

【0045】

- 10 オペレーティング・システム
- 20 仮想メモリ・マネージャ
- 21 フリー・リスト
- 22 フリー・フレーム
- 23 使用中のリスト
- 24 使用中のフレーム
- 25 ページ置換デモン
- 30 プロセス・マネージャ
- 32 待機リスト
- 33 待機中のスレッド
- 40 ファイルシステム
- 50 ネットワーク・サブシステム
- 51 ハードディスク・ドライブ
- 702 通信インフラストラクチャ（バス）
- 704 プロセッサ
- 706 メイン・メモリ
- 708 ディスプレイ・インターフェース
- 710 ディスプレイ装置
- 712 二次メモリ
- 714 ハードディスク・ドライブ
- 716 取り外し可能ストレージ・ドライブ
- 718 取り外し可能記憶装置
- 720 インターフェース
- 724 通信インターフェース
- 726 通信路

10

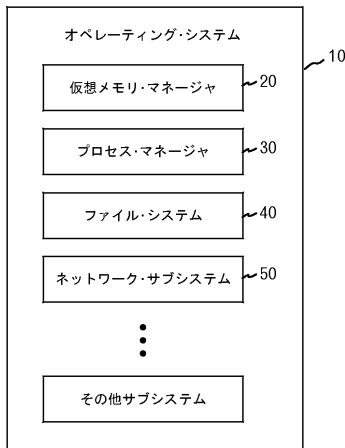
20

30

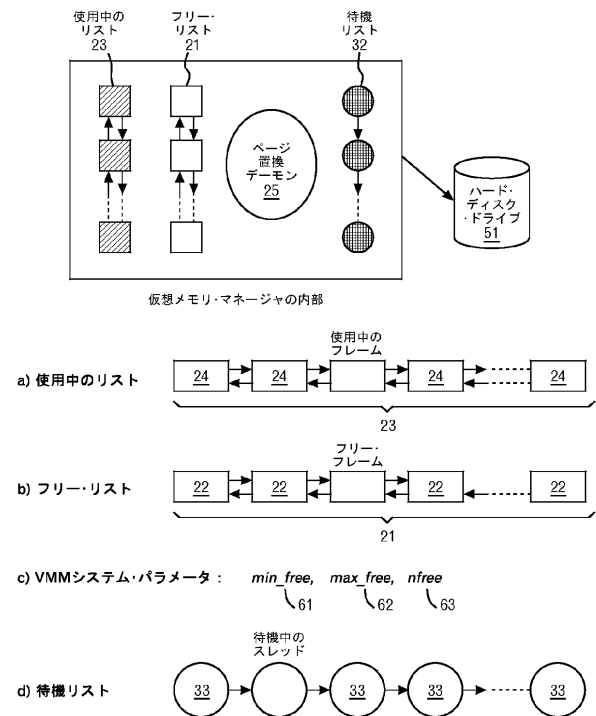
40

50

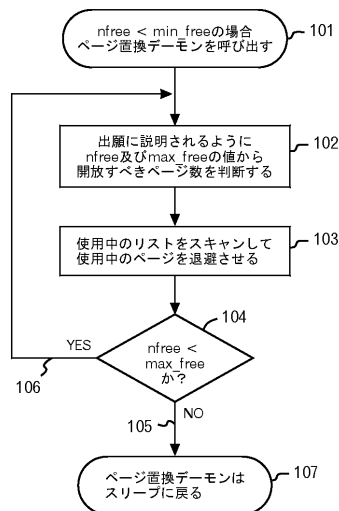
【図 1】



【図 2】



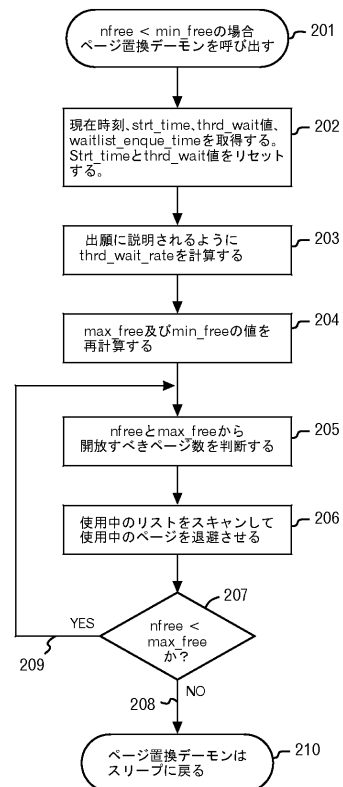
【図 3】



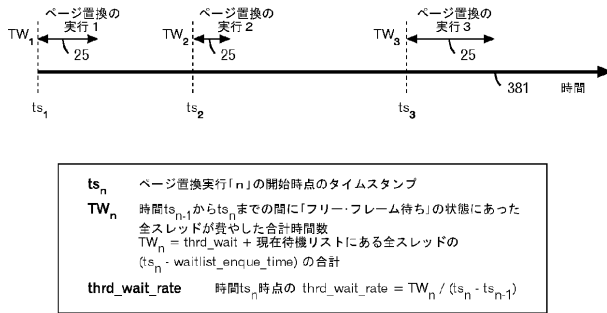
【図 4】

- a) $thrd_wait$ 35
- b) 各スレッドの $waitlist_enqueue_time$ 36
- c) $thrd_wait_rate$ 37
- d) $thrd_wait_rate_tgt$ 38
- e) $strt_time$ 39

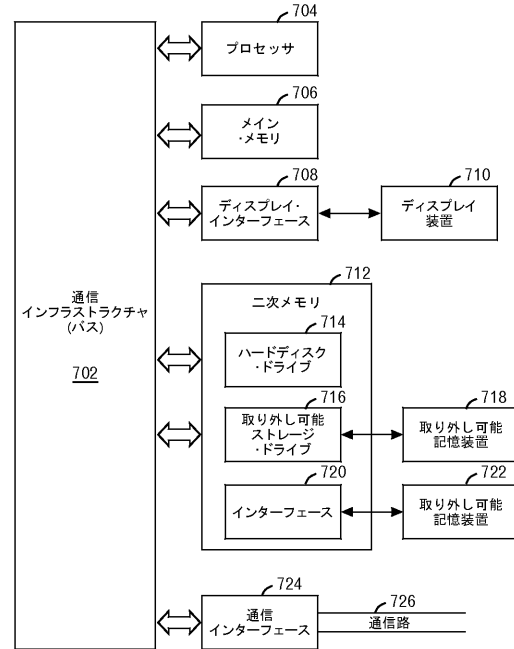
【図 5】



【図 6】



【図 7】



【国際調査報告】

60700320016



11

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US05/15493

A. CLASSIFICATION OF SUBJECT MATTER

IPC: G06F 12/00(2006.01), 9/00(2006.01)

USPC: 711/137, 170, 204; 713/1, 2, 100

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S.: 711/137, 170, 204; 713/1, 2, 100

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EAST (US-PQFUB, USPAT, USOCR, EPO, JPO, DERWENT, IBM, TDB, FPRS)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6,112,301 (Johnson) 29 August 2000 (29.08.2000), col. 1, line 48-64 - col. 4, line 14-67.	1, 13 and 14

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"B" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T"

later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X"

document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y"

document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"A"

document member of the same patent family

Date of the actual completion of the international search

25 January 2007 (25.01.2007)

Date of mailing of the international search report

02 MAR 2007

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Facsimile No. (571) 273-3201

Authorized officer

Reginald Bragdon

Telephone No. (571) 272-0703

Form PCT/ISA/210 (second sheet) (April 2005)

17. 5. 2007

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

(特許庁注：以下のものは登録商標)

1 . U N I X

(74)代理人 100086243

弁理士 坂口 博

(72)発明者 ジャン、ジョーフォン

アメリカ合衆国 1 0 5 6 2 ニューヨーク州 オッシニング バーンズ・ストリート 2 1 3

(72)発明者 バトナイク、ブラタップ、チャンドラ

アメリカ合衆国 1 0 5 6 2 ニューヨーク州 オッシニング バーンズ・ストリート 2 1 3

(72)発明者 ブルグラ、ラマンジャネヤ、サルマ

アメリカ合衆国 1 0 5 2 0 ニューヨーク州 クロトン・オン・ハドソン シーニック・ドライブ 2 4 アpartment U

Fターム(参考) 5B005 JJ11 MM36 QQ02 UU43 VV02