(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5583837号
(P5583837)
(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 9/50 (2006.01)　G06F 9/46 465C
G06F 9/46 465A

請求項の数 17 (全 17 頁)

(21)出願番号　特願2013-500348(P2013-500348)
(86)(22)出願日　平成22年11月8日(2010.11.8)
(65)公表番号　特表2013-524306(P2013-524306A)
(43)公表日　平成25年6月17日(2013.6.17)
(86)国際出願番号　PCT/EP2010/067044
(87)国際公開番号　WO2011/116842
(87)国際公開日　平成23年9月29日(2011.9.29)
審査請求日　平成25年8月2日(2013.8.2)
(31)優先権主張番号　12/731,426
(32)優先日　平成22年3月25日(2010.3.25)
(33)優先権主張国　米国(US)

(73)特許権者　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・システム内でタスクを開始するためのコンピュータ実装方法、システム及びコンピュータ・プログラム

(57)【特許請求の範囲】
【請求項１】
コンピュータ・システム内でタスクを開始するためのコンピュータ実装方法であって、
一のプロセッサによって実行すべき一のタスクを受け取るステップと、
一の目標サービス・レベルに対する前記タスクの状態を受け取るステップと、
前記プロセッサの現在の電力状態を決定するステップと、
前記タスクが前記目標サービス・レベルを満たしていることを示す前記状態及び前記現在の電力状態が低電力状態であるという決定に応答して、前記プロセッサ上で前記タスクの実行を開始するステップと、
前記タスクが前記目標サービス・レベルを満たしていないことを示す前記状態及び前記現在の電力状態が低電力状態であるという決定に応答して、前記プロセッサを高電力状態に移動させることができるか否かを決定するステップと、
前記プロセッサを高電力状態に移動させることができるという決定に応答して、前記プロセッサを前記高電力状態に移動させるステップと、
前記プロセッサを前記高電力状態に移動させることに応答して、前記プロセッサ上で前記タスクの実行を開始するステップと、
前記タスクが前記目標サービス・レベルを満たしていないことを示す前記状態、前記現在の電力状態が低電力状態であるという決定及び前記プロセッサを前記高電力状態に移動させることができないという決定に応答して、他のプロセッサ上で前記タスクの実行を開始するステップと、

を含む方法。
【請求項２】
前記状態は、前記タスクに関連する現在のサービス・レベルが前記目標サービス・レベルのしきい値内にあるときは、前記目標サービス・レベルが満たされていることを示し、前記タスクに関連する前記現在のサービス・レベルが前記目標サービス・レベルのしきい値内にないときは、前記目標サービス・レベルが満たされていないことを示す、請求項１に記載の方法。
【請求項３】
前記タスクが前記低電力状態で実行される場合には、前記タスクに関連する前記現在のサービス・レベルが前記目標サービス・レベルのしきい値内にないことを予測するステップと、
予測する前記ステップに応答して、前記タスクが前記目標サービス・レベルを満たしていないことを示すように前記状態をセットするステップをさらに含む、請求項２に記載の方法。
【請求項４】
前記目標サービス・レベルが、前記タスクに関連する一のサービス・レベル契約（ＳＬＡ）に含まれる、請求項１に記載の方法。
【請求項５】
前記他のプロセッサ上の前記タスクの実行の開始が、割り込みを使用して行われる、請求項１に記載の方法。
【請求項６】
前記他のプロセッサが前記高電力状態にあるか否かを決定するステップをさらに含み、
前記他のプロセッサ上で前記タスクの実行を開始する前記ステップが、前記他のプロセッサが前記高電力状態にあるという決定に応答して行われる、請求項１に記載の方法。
【請求項７】
前記タスクに関連する現在のサービス・レベルに関するデータを収集するステップと、
前記収集されたデータ及び前記目標サービス・レベルに応答して、前記状態をセットするステップをさらに含む、請求項１に記載の方法。
【請求項８】
前記タスクが前記目標サービス・レベルを満たしていることを示す前記状態及び前記現在の電力状態が前記高電力状態であるという決定に応答して、前記プロセッサを前記低電力状態に移動させるステップと、
前記プロセッサを前記低電力状態に移動させることに応答して、前記プロセッサ上で前記タスクの実行を開始するステップをさらに含む、請求項１に記載の方法。
【請求項９】
請求項１ないし請求項８の何れか１項に記載の方法の各ステップをコンピュータに実行させるためのコンピュータ・プログラム。
【請求項１０】
コンピュータ・システム内でタスクを開始するためのシステムであって、
一のコンピュータ・メモリと、
前記コンピュータ・メモリと通信する一の命令処理要素を備え、
前記命令処理要素は、メモリから命令を取り出すための一の命令取り出し要素及び取り出された命令を実行するための１つ以上の実行要素を有し、
前記命令処理要素は、
一のプロセッサによって実行すべき一のタスクを受け取るステップと、
一の目標サービス・レベルに対する前記タスクの状態を受け取るステップと、
前記プロセッサの現在の電力状態を決定するステップと、
前記タスクが前記目標サービス・レベルを満たしていることを示す前記状態及び前記現在の電力状態が低電力状態であるという決定に応答して、前記プロセッサ上で前記タスクの実行を開始するステップと、

前記タスクが前記目標サービス・レベルを満たしていないことを示す前記状態及び前記現在の電力状態が低電力状態であるという決定に応答して、前記プロセッサを高電力状態に移動させることができるか否かを決定するステップと、
前記プロセッサを高電力状態に移動させることができるという決定に応答して、前記プロセッサを前記高電力状態に移動させるステップと、
前記プロセッサを前記高電力状態に移動させることに応答して、前記プロセッサ上で前記タスクの実行を開始するステップと、
前記タスクが前記目標サービス・レベルを満たしていないことを示す前記状態、前記現在の電力状態が低電力状態であるという決定及び前記プロセッサを前記高電力状態に移動させることができないという決定に応答して、他のプロセッサ上で前記タスクの実行を開始するステップとを含む方法を実施するように構成される、システム。
【請求項１１】
前記プロセッサの前記現在の電力状態を格納するための電力状態レジスタをさらに備え、前記現在の電力状態を決定する前記ステップは、前記電力状態レジスタから前記現在の電力状態を読み取ることを含む、請求項１０に記載のシステム。
【請求項１２】
前記プロセッサを高電力状態に移動させることができるか否かを示すフラグを格納するための電力状態レジスタをさらに備え、
前記プロセッサを前記高電力状態に移動させることができるか否かを決定する前記ステップは、前記電力状態レジスタから前記フラグを読み取ることを含む、請求項１０に記載のシステム。
【請求項１３】
前記状態は、前記タスクに関連する現在のサービス・レベルが前記目標サービス・レベルのしきい値内にあるときは、前記目標サービス・レベルが満たされていることを示し、前記タスクに関連する前記現在のサービス・レベルが前記目標サービス・レベルのしきい値内にないときは、前記目標サービス・レベルが満たされていないことを示す、請求項１０に記載のシステム。
【請求項１４】
前記方法は、
前記タスクが前記低電力状態で実行される場合には、前記タスクに関連する前記現在のサービス・レベルが前記目標サービス・レベルのしきい値内にないことを予測するステップと、　予測する前記ステップに応答して、前記タスクが前記目標サービス・レベルを満たしていないことを示すように前記状態をセットするステップをさらに含む、請求項１３に記載のシステム。
【請求項１５】
前記目標サービス・レベルが、前記タスクに関連する一のサービス・レベル契約（ＳＬＡ）に含まれる、請求項１０に記載のシステム。
【請求項１６】
前記他のプロセッサ上の前記タスクの実行の開始が、割り込みを使用して行われる、請求項１０に記載のシステム。
【請求項１７】
前記方法は、
前記タスクが前記目標サービス・レベルを満たしていることを示す前記状態及び前記現在の電力状態が前記高電力状態であるという決定に応答して、前記プロセッサを前記低電力状態に移動させるステップと、
前記プロセッサを前記低電力状態に移動させることに応答して、前記プロセッサ上で前記タスクの実行を開始するステップをさらに含む、請求項１０に記載のシステム。
【発明の詳細な説明】
【技術分野】
【０００１】

本発明は、コンピュータ環境内の処理に係り、さらに詳細に説明すれば、コンピュータ環境内でサービス・レベルの要件を満たしつつ、電力消費量を最小化することに係る。

【背景技術】

【０００２】

コンピュータ・システム全体の性能は、プロセッサの性能／構造、キャッシュ・メモリ、入力／出力（Ｉ／Ｏ）サブシステム、メモリ管理機能の効率、主メモリ装置、並びに相互接続インタフェースのタイプ及び構造を含む、コンピュータ構造の重要な要素の各々によって影響を受ける。

【０００３】

産業界では、システム／サブシステムの設計及び／又は構造を改良することを通してコンピュータ・システム全体の性能及び密度を最大化するための改良された及び／又は革新的な解決策を作成することを目的として、広範囲な研究開発努力に対する投資が継続的に行われている。高可用性システムは、システム全体の信頼性に関係するさらなる挑戦を提示する。というのは、顧客が期待するのは、新しいコンピュータ・システムが平均故障間隔（ＭＴＢＦ）の点で既存のシステムよりも著しく優れているというだけではなく、追加の機能、増大した性能、増大したストレージ、より低い運転コスト等を提供するというものであるからである。コンピュータ・システム設計の挑戦をさらに困難にする顧客の他の要件には、アップグレードの容易さ、システム環境（例えば、スペース、電力、冷却等）への影響を低減すること等がある。さらに、サービス（例えば、配信時間、性能）の必要なレベルを定義するために、サービス・レベル契約（ＳＬＡ）を採用することがある。

【０００４】

プロセッサの性能が継続的に増大するにつれて、プロセッサが消費する電力量も継続的に増大する傾向がある。所与のプロセッサが消費し得る電力量は、プロセッサの動作環境のような種々の要因によってしばしば制限される。従って、プロセッサの性能上の改良は、これらの改良を実装するためにプロセッサが必要とする追加の電力によって制限されることがある。

【０００５】

米国特許第７４６１２７５号明細書は、諸プロセッサ・コアを動的に交換するための技術に向けられている。第１のコアは、第１の命令セットを有する。第１のコアは、第１の性能レベルでプログラムを実行する。第１のコアは、トリガ・イベントが生じるときに、プログラムの実行を停止する。第２のコアは、第１の命令セットと互換性のある第２の命令セットを有し、第１の性能レベルとは異なる第２の性能レベルを有する。第１のコアがプログラムを実行しているとき、第２のコアはパワーダウン状態にある。第１のコアがプログラムの実行を停止した後、第２のコアが第２の性能レベルでプログラムの実行を継続するように、一の回路が第２のコアをパワーアップする。

【０００６】

米国特許第７０９３１４７号明細書は、少なくともそれぞれの動作電力要件及び処理能力が互いに異なる複数のコンピュータ・ハードウェア・プロセッサ・コアを含むコンピュータ・システムにおいて、その動作電力を節約するための技術に向けられている。一のモニタは、コンピュータ・ハードウェア・プロセッサ・コアの各々から、そのときに実行しているアプリケーション・ソフトウェアの特定のランに特有の性能メトリック情報を収集する。ワークロード転送機構は、低減された動作電力を検索するために、実行中のアプリケーション・ソフトウェアを第２のコンピュータ・ハードウェア・プロセッサ・コアに転送する。もし、システムの動作電力が或る遅延によって節約されるならば、実行中のアプリケーション・ソフトウェアのその後の転送を遅らせるために転送遅延構造が接続される。

【０００７】

米国特許第７５２６６６１号明細書は、実行すべきスレッドを選択し且つこのスレッドに基づいてプロセッサ・コアの目標性能状態を識別するための、スレッドを管理するシステム及び方法に向けられている。目標性能状態の識別は、選択されたスレッドの優先順位

をマッピング・ポリシに適用して、目標性能状態を得ることを含み得る。一実施形態では、選択されたコアの目標性能状態への遷移を開始することができ、そして選択されたスレッドを選択されたプロセッサ・コアによる実行のためにスケジュールすることができる。

【先行技術文献】

【特許文献】

【０００８】

【特許文献１】米国特許第７４６１２７５号明細書

【特許文献２】米国特許第７０９３１４７号明細書

【特許文献３】米国特許第７５２６６６１号明細書

【発明の概要】

【発明が解決しようとする課題】

【０００９】

しかしながら、従来技術のシステムは、複数のプロセッサにわたってバイナリ互換のアプリケーション・コードを実行する間に、１ワット当たりの最良のスループットを提供するように最適化されていないという課題がある。

【課題を解決するための手段】

【００１０】

前記の課題を解決するために、本発明の第１の側面では、コンピュータ・システム内でタスクを開始するためのコンピュータ実装方法が提供される。前記方法は、一のプロセッサによって実行すべき一のタスク及び一の目標サービス・レベルに対する前記タスクの状態を受け取るステップを含む。前記プロセッサの現在の電力状態が決定される。前記タスクが前記目標サービス・レベルを満たしていることを示す前記状態及び前記現在の電力状態が低電力状態（low power state）であるという決定に応答して、前記プロセッサ上で前記タスクの実行が開始される。前記方法は、前記タスクが前記目標サービス・レベルを満たしていないことを示す前記状態及び前記現在の電力状態が低電力状態であるという決定に応答して、前記プロセッサを高電力状態（high power state）に移動させることができるか否かを決定するステップと、前記プロセッサを高電力状態に移動させることができるという決定に応答して、前記プロセッサを前記高電力状態に移動させるステップと、前記プロセッサを前記高電力状態に移動させることに応答して、前記プロセッサ上で前記タスクの実行を開始するステップをさらに含む。

【００１１】

本発明の第２の側面では、コンピュータ・システム内でタスクを開始するためのシステムが提供される。前記システムは、一のコンピュータ・メモリと、前記コンピュータ・メモリと通信する一の命令処理要素を備える。前記命令処理要素は、メモリから命令を取り出すための一の命令取り出し要素及び取り出された命令を実行するための１つ以上の実行要素を有する。前記コンピュータ・システムは、一のプロセッサによって実行すべき一のタスク及び一の目標サービス・レベルに対する前記タスクの状態を受け取るステップを含む方法を実施するように構成される。前記プロセッサの現在の電力状態が決定される。前記タスクが前記目標サービス・レベルを満たしていることを示す前記状態及び前記現在の電力状態が低電力状態であるという決定に応答して、前記プロセッサ上で前記タスクの実行が開始される。前記方法は、前記タスクが前記目標サービス・レベルを満たしていないことを示す前記状態及び前記現在の電力状態が低電力状態であるという決定に応答して、前記プロセッサを高電力状態に移動させることができるか否かを決定するステップと、前記プロセッサを高電力状態に移動させることができるという決定に応答して、前記プロセッサを前記高電力状態に移動させるステップと、前記プロセッサを前記高電力状態に移動させることに応答して、前記プロセッサ上で前記タスクの実行を開始するステップをさらに含む。

【００１２】

本発明の第３の側面では、前記第１の側面に係るコンピュータ実装方法の各ステップをコンピュータに実行させるためのコンピュータ・プログラムが提供される。

【００１３】
追加の特徴及び利点は、本発明の技術を通して実現される。本発明の他の実施形態及び側面は、本明細書において詳述され、請求項に記載の発明の一部と見なされる。利点及び特徴を有する本発明をより良く理解するためには、以下の説明及び図面を参照されたい。
【発明の効果】
【００１４】
本発明は、ＳＬＡ内で定義されるような一のサービス・レベルを満たすのに必要な電力量を提供することにより、電力効率を改良することができるという効果を奏する。また、本発明は、ＳＬＡ要件を依然として満たしつつ、電力消費量を低減させるか又はシステムによって実行することができる処理の量を増加させることができるという効果を奏する。
【図面の簡単な説明】
【００１５】
【図１】本発明の実施形態によって実装することができる、データ処理システムのブロック図である。
【図２】本発明の実施形態によって実装することができる、集積化回路のブロック図である。
【図３】本発明の実施形態によって実装することができる、ディスパッチ・アルゴリズムのフローチャートである。
【図４】本発明の実施形態によって実装することができる、コンピュータ・プログラムを示す図である。
【発明を実施するための形態】
【００１６】
本発明の実施形態は、一のタスクを実行するのに割り振るべき電力量を決定するための入力として、当該タスクに関連するサービス・レベル契約（ＳＬＡ）を利用する。実施形態では、ディスパッチ・プロセスは、実行すべき一のタスクを受け取るとともに、当該タスクがそれに関連するＳＬＡを現在満たしているか否かという情報を受け取る。もし、当該タスクが現在ＳＬＡを満たしていなければ、プロセッサが高電力状態にあるか又は当該プロセッサを高電力状態に移動させることができることを条件として、当該プロセッサ上で当該タスクが実行されるであろう。もし、当該プロセッサを高電力状態に移動させることができなければ、当該タスクは、実行のために他のプロセッサ（例えば、高電力状態で動作することができるプロセッサ）に送信されるであろう。このように、ＳＬＡを満たすためにより高速の実行を必要とするタスクは、高電力状態にあるプロセッサ上で開始されるであろう。
【００１７】
実施形態では、システム全体について最良の電力／性能を得るために、異なるシリコン最適化を有する複数のプロセッサ・チップが利用される。実施形態では、ディスパッチ・プロセスは、ＳＬＡによって必要とされる一の目標サービス・レベルを検査することを含む。もし、現在のサービス・レベルが特定のプロセス・スレッド（すなわち、タスク）のためのＳＬＡを満たさなければ、ディスパッチ・プロセスは、現在のコアの電力状態を検査した後、現在のコア上で処理を継続すべきであるか又は当該処理を他のコアに移動させるべきであるかを決定する。
【００１８】
本明細書では、「電力／性能」という用語は、マルチプロセッサ・システム又はプロセッサのようなコンピュータ・システム内の電力消費量の効率の測定量を指す。
【００１９】
「タスク」という用語は、スレッド又は１群のスレッドを指す。ＳＬＡは、単一のタスク又は１群のタスク（例えば、アプリケーション又はコンピュータ・システム）に関連することがある。
【００２０】
「サービス・レベル契約」又は「ＳＬＡ」という用語は、顧客とサービス・プロバイダ

との間の交渉による合意であって、サービス、優先順位、責任、保証等に関する共通の理解を記述したものを指す。ＳＬＡは、可用性、保守性、性能、サービスの運営等のサービスの属性を指定することができる。幾つかのケースでは、一のサービス・レベルが目標として指定され、他のケースでは、一のサービス・レベルが最小として指定される。実施形態では、ＳＬＡは、一の要求を受け取ってから当該要求をサービスするまでの経過時間として定義される、応答時間を指定する。その場合、当該ＳＬＡが指定する応答時間は、当分野で公知の任意の方法を使用して追跡される。実施形態では、かかる応答時間（又は当該ＳＬＡに関連する他の任意のサービス属性）を追跡するために、IBM（登録商標）社のEnterprise Workload Manager（商標）が使用される。

【００２１】

既存のソフトウェア及びハードウェアを使用する場合、１ワット当たりのピーク性能について大規模マルチプロセッサ・システムを最適化することは、困難であることが多い。システムは、複数の処理要素を含むことがあり、これらの処理要素は、単一のパッケージ内にパッケージ化された種々の電力／性能動作点（本明細書では「電力状態」とも称する）を有するか、又は複数のパッケージ間のキャッシュ・コヒーレント・リンクを介して接続される。現在のディスパッチ・アルゴリズムは、ピーク性能を達成するというゴールに基づいて、諸プロセスを諸処理要素にディスパッチするために使用される。現在のマルチプロセッサ・コンピュータ・システムは、所与の量の電力について必ずしも最良のスループットを供給するとは限らない。幾つかの既存のコンピュータ・システムは、単一のマイクロプロセッサ・コア設計を使用し、アプリケーションを実行している種々の期間中に、１ワット当たりの性能を変更するために、動的な電圧スケーリングを使用する。他の既存のシステムは、特定のアプリケーションについて性能優位性を提供するために、異なる命令セット・アーキテクチャを有する複数のコアを含む。これらのシステムは、多くのマイクロプロセッサ・コア（本明細書では、「プロセッサ」とも称する）にわたってバイナリ互換のアプリケーション・コードを実行する間に、１ワット当たりの最良のスループットを提供するように最適化されていない。１つの理由は、ディスパッチャが、システム内のコア相互間の性能差を利用せず、それらの使用率を最適化しないということである。

【００２２】

本発明の実施形態は、種々のワークロードを実行する間に、システム全体の電力／性能を最適化するように作用する。実施形態では、１ワット当たりのシステム全体のスループットを最適化するコンピュータ・システムを提供するために、複数のマルチプロセッサ・システムがともに使用される。実施形態は、２つ以上のチップ（本明細書では、「集積回路」又は「マイクロプロセッサ・チップ」とも称する）にまたがる複数のプロセッサから構成されたマルチプロセッサ・システムを含む。これらのチップは、異なるタイプのもので、同じバイナリ・ソフトウェアを実行することができるが、１ワット当たりの性能特性において互いに異なる。非常に広範なコンピュータ・システムについて、Intel（登録商標）社のIA-32（IntelArchitecture、３２ビット）及びIBM（登録商標）社のPower Architecture（登録商標）のようなアーキテクチャが使用される。これらのアーキテクチャについては、既に多くのバイナリ互換の処理コア設計が利用可能である。これらのコア設計の各々は、異なる電力／性能特性を有する。実施形態は、単一のマルチプロセッサ・サーバ上で複数のプロセスを実行するとき、より良い全体的な電力／性能を可能にするであろう。異なる性能及び電力最適化を有する複数のチップを使用することによって、１ワット当たりのシステム・スループットを増大させることができる。既存のマイクロプロセッサは、最大で４．５～５×の電力／性能を犠牲にして、最大で２×の範囲の性能を示す。この範囲は、追加のシリコン・チューニング、装置チューニング及び回路の選択でさらに拡大することができる。複数のマルチプロセッサ・チップ及び当該チップの複数の電力／性能点を使用する機会を与えられると、実施形態は、システムの電力／性能能力を最大化するように、諸コアへの諸スレッドのディスパッチを最適化する。実施形態では、複数のチップは、潜在的には一のノード・コントローラを通る、諸相互接続バスを介して接続される。これらのチップのサブセットは、高電力を犠牲にして高速化のために最適化されたシ

リコン・プロセスで実装され、一方、これらのチップの大部分は、電力のために最適化されたシリコン・プロセスで実装される。これらのチップ・タイプは、両方とも同一のバイナリを実行することができる。実施形態は、これらの異種のプロセッサを対称マルチプロセッシング（ＳＭＰ）システムとして組み合わせるとともに、電力／性能を最適化するように調整されたディスパッチ・アルゴリズムを使用する。もし、高性能で且つ高電力のプロセッサを使用することなく、これらのプロセスのサービス・レベル契約（ＳＬＡ）上の要件（又は他のシステム要件）を満たすことができれば、実施形態によって提供される電力遮断（power gating）制御は、当該プロセッサの有効電力及び漏洩電力の両方を最小化するために使用される。

【００２３】

必要とされる場合、高性能で且つ高電力の一のコアが電力遮断制御を通してパワーオンされ、そして高優先順位の諸プロセスが（高電力状態にある）このプロセッサにディスパッチされる。実施形態では、このことは、ＳＬＡ及び／又はマルチプロセッサ・システムの現在の状態のテーブルを使用する、新規なハイパーバイザ又はオペレーティング・システムによって行われる。このテーブルは、どのコアが高性能対電力最適化であり且つどのコアが現在パワーオンされているかというマッピングを含む。最適な電力／性能のケースでは、全ての高電力のコアがパワーオフされ、これらのコアは、ＳＬＡ（又は他のシステム要件）を満たすために必要とされる場合にのみパワーオンされる。

【００２４】

１例を挙げると、電力が最適化された一のコア上では、一のスレッドは、タイマ割り込みによって示されるその現在のタイムスライスを終了し、そしてディスパッチャが呼び出される。ディスパッチャ・コードは、サービス品質又は現在のサービス・レベルを定期的に検査することを含む。この例では、潜在的なサービス品質の問題が発生したことが決定される。この情報は、ディスパッチすべき次のスレッドがどのスレッドであるかを決定するために使用される。高性能のコアを使用すべきであると決定されることがある。ディスパッチャは、現在実行中の諸スレッド及び潜在的なサービス品質の問題を発生したこの新しいスレッドの組み合わせについて、十分な高性能ハードウェア・コアが利用可能である（そしてパワーオンされている）か否かを決定する。幾つかのケースでは、ディスパッチャは、システムの制御コードに対し、システム内の全てのスレッドのＳＬＡを満たすために利用可能なハードウェアを増大させるために高性能コアをパワーオンすべきであることを通知する。利用可能であれば、ディスパッチャは、待機中の他のスレッドを選択することができる。実施形態では、このスレッドは、現在それぞれのサービス品質レベルを満たしている複数のスレッドのうち最高優先順位のスレッドになるであろう。代替的な実施形態では、発熱オーバーヘッドを解放してシステムが追加の高電力コアをパワーオンすることを可能にするため、現在実行中の低優先順位の諸スレッドを多数のコア上で中断することができる。

【００２５】

実施形態は、ハイパーバイザ又はオペレーティング・システムによって制御される、１セットのソフトウェア・テーブルを使用する。これらのテーブルは、どのスレッドが高電力／高性能コア上にディスパッチされるのを待機しているかという情報を含む。他の情報は、システム内の全てのコアの現在の状態（特に、どのコアが高性能であり且つどの位置にあるかという情報や、どのコアが現在パワーオンされているかという情報）を含むことができる。結果的なシステムは、可変数のスレッドをディスパッチャに利用可能にするであろう。この数は、ＳＬＡ要件を満たすために、ハイパーバイザ及び／又はオペレーティング・システムによって動的に変更されるであろう。図１は、本発明の実施形態を実装することができる、データ処理システム１００のブロック図である。実施形態では、システム１００は、対称マルチプロセッシング（ＳＭＰ）サーバ・コンピュータ・システムである。図１のＳＭＰサーバ・コンピュータ・システム１００は、一のユーザ・アプリケーションを実行するために当該アプリケーションにマップ（すなわち、当該アプリケーションが一時的に所有）することができる、複数の物理ハードウェア装置を含む。

【００２６】
ＳＭＰサーバ・コンピュータ・システム１００は、物理ＳＭＰサーバ１０２を含み、物理ＳＭＰサーバ１０２は、プロセッサ１０４、メモリ１０６及びＩ／Ｏアダプタ１０８のような物理ハードウェア装置を含む。これらの物理装置は、物理ＳＭＰサーバ・コンピュータ・システム１００上に常駐するハイパーバイザ１１０によって管理される。図１に示すように、物理ＳＭＰサーバ１０２は、１つ以上のマイクロプロセッサ・チップ１２６によって実装される。図１では、同様の構成を有するものとして示されているが、物理ＳＭＰサーバ１０２内の複数のマイクロプロセッサ・チップ１２６は、異なる構成（例えば、異なる数のプロセッサ１０４、異なるタイプのプロセッサ１０４）を有することができる。

【００２７】
本明細書では、「マイクロプロセッサ・チップ」という用語は、シリコンの単一ピースを使用して製造される装置を指す。マイクロプロセッサ・チップ１２６をチップ又は集積回路と称することもある。１つ以上のハードウェア要素が単一のマイクロプロセッサ・チップ上に製造される。一般に、ハードウェア要素は、プロセッサ１０４（又は処理コア）、メモリ１０６（例えば、キャッシュ・メモリ）、Ｉ／Ｏアダプタ１０８を含むが、これに加えて、圧縮エンジン、暗号化エンジン、プロトコル処理エンジン、アーキテクチャ・エミュレーション・エンジン、データ・ストリーム処理エンジン等の特殊機能要素を含むことがある。

【００２８】
仮想サーバとは、同じ能力、インタフェース及び状態を有する物理サーバのためのプロキシである。仮想サーバは、ハイパーバイザ１００によって作成され且つ管理される。各仮想サーバは、物理ＳＭＰサーバのユーザ（例えば、その上で稼働するオペレーティング・システム、ミドルウェア及びアプリケーション・ソフトウェア）に対し、物理ＳＭＰサーバであるように見える。図１のＳＭＰサーバ・コンピュータ・システム１００は、仮想サーバ１１２及び１１２ａのような１つ以上の仮想サーバを含む。各仮想サーバ１１２は、そのソフトウェアに対し、当該仮想サーバ１１２の排他的使用に利用可能なそれ自体のプロセッサ、メモリ及びＩ／Ｏアダプタを含むように見える。例えば、仮想サーバ１１２は、仮想プロセッサ１２０、仮想メモリ１２２及び仮想Ｉ／Ｏアダプタ１２４を含む。仮想サーバ１１２ａは、仮想プロセッサ１２０ａ、仮想メモリ１２２ａ及び仮想Ｉ／Ｏアダプタ１２４ａを含む。

【００２９】
各仮想サーバ１１２は、オペレーティング・システム、ミドルウェア及びアプリケーションを含む、それ自体のソフトウェア環境をサポートする。各仮想サーバ１１２のソフトウェア環境は、他の仮想サーバのソフトウェア環境とは異なることがある。実施形態では、各仮想サーバによって実行されるオペレーティング・システムは、互いに異なっていてもよい。例えば、仮想サーバ１１２は、オペレーティング・システム１１４、ミドルウェア１１６及びアプリケーション１１８をサポートする。仮想サーバ１１２ａは、オペレーティング・システム１１４ａ、ミドルウェア１１６ａ及びアプリケーション１１８ａをサポートする。オペレーティング・システム１１４及び１１４ａは、互いに同じであってもよいし、互いに異なっていてもよい。

【００３０】
仮想サーバ１１２は、サーバ環境を定義するサーバの論理記述であって、ユーザに対しあたかも物理サーバであるかのように作用し、物理サーバと同じ方法でアクセスされ、情報を提供するものである。各仮想サーバのために定義される仮想プロセッサ、仮想メモリ及び仮想Ｉ／Ｏアダプタは、物理プロセッサ、メモリ及びＩ／Ｏアダプタの論理的な代替物である。

【００３１】
ハイパーバイザ１１０は、仮想プロセッサ、仮想メモリ及び仮想Ｉ／Ｏアダプタを有する仮想サーバとこれらの仮想装置を実装するために選択される物理ハードウェア装置との

間のマッピングを管理する。例えば、一の仮想プロセッサがディスパッチされる場合、ハイパーバイザ１１０は、当該仮想プロセッサを実行し且つ実装するために使用すべき１つの物理プロセッサ１０４を選択する。ハイパーバイザ１１０は、物理装置の選択及び仮想装置へのそれらの一時的割り当てを管理する。

【００３２】

ハイパーバイザ１１０は、諸仮想ＳＭＰサーバを動的に作成し、管理し、破壊することに責任を有する。ハイパーバイザ１１０は、諸仮想プロセッサ、諸仮想Ｉ／Ｏアダプタ及び諸仮想メモリ・ブロックを、全体として除去又は追加することができる。さらに、ハイパーバイザ１１０は、資源を動的に割り振り、物理資源のタイム・シェアリングを管理し、オペレーティング・システムを介在させることなく、一のプロセッサにマップされた物理資源を変更することにも責任を有する。さらに、ハイパーバイザ１１０は、シェアリングが望まれない状況では、諸物理資源を諸仮想資源に専用化することができる。ハイパーバイザ１１０は、諸物理資源の追加又は除去を管理する責任を有する。ハイパーバイザ１１０は、これらの追加及び除去を上部レベルのアプリケーションに対し透明な態様で実行する。

【００３３】

図２は、実施形態によって実装することができる、マイクロプロセッサ・チップ１２６上に位置するプロセッサ１０４（図２では処理コア２０２と表記）のブロック図である。図２の処理コア２０２は、電力状態レジスタ２０４を含む。実施形態では、電力状態レジスタ２０４は、当該プロセッサの現在の電力状態（例えば、低電力状態、高電力状態）を格納するとともに、当該プロセッサを高電力状態に移動させることができるか否かを示すフラグ（又は他の標識）を格納する。代替的な実施形態では、フラグ及び現在の電力状態は、別個のレジスタ内に格納される。フラグは、プロセッサ１０４の物理能力及び動作環境によって課される制限に基づいて、セットすることができる。電力事態及びフラグの両者は、処理コア２０２又は外部の制御プログラム（例えば、マイクロプロセッサ・チップ１２６上に位置する電力制御装置モジュール、ハイパーバイザ１１０等）によってセットすることができる。

【００３４】

図２に示すように、処理コア２０２は、複数のサブ要素（例えば、１つ以上の浮動小数点ユニット、１つ以上のロード／ストア・ユニット、命令順序付けユニット、固定小数点実行ユニット、命令取り出し／分岐実行ユニット等）を含むことができる。

【００３５】

本明細書では、「処理コア」及び「プロセッサ」という用語は、同じ装置を指すために交換可能に使用される。「物理プロセッサ」という用語は、処理コアを指し、当該コアに専用化されているか又は複数のコアによって共用される他のハードウェア要素を含むことがある。従って、物理プロセッサは、処理コア及び当該処理コアに専用化されているか又は当該処理コアによって共用されるハードウェア要素である。

【００３６】

「現在の電力状態」という用語は、一のプロセッサの現在の電力状態を指す。実施形態では、現在の電力状態は、高電力状態又は低電力状態である。「高電力状態」という用語は、高クロック周波数モードに置かれた処理コア２０２を指し、電力状態レジスタ２０４は、このモードを反映するように更新される。「高電力コア」という用語は、高電力状態で実行中の処理コア２０２を指す。「低電力状態」という用語は、低クロック周波数モードに置かれた処理コア２０２を指し、電力状態レジスタ２０４は、このモードを反映するように更新される。実施形態は、システム内の種々の処理コアの電力／性能を調整するための手段として、コア・クロック周波数の変化を使用する。このクロック周波数制御に代えて、又はこれに加えて、コアの電力／性能を調整するための他の手段を使用することもできる。本明細書では、コアの電力／性能の調整は、プロセッサを特定の電力状態に「移動させる」ことを指す。プロセッサを低電力状態から高電力状態に移動させることは、クロック制御論理における動作周波数設定値をより高い周波数設定値に変更することを含む

。適切な任意のクロック制御機構を使用することができる。実施形態は、２つの設定値として、高い設定値（例えば、４ ＧＨｚ）及び低い設定値（例えば、３ ＧＨｚ）を利用する。代替実施形態は、周波数スケール単位で増加されるステップの数を決定するために、幾つかの可能な周波数設定値及び追加の論理を含む。同様に、プロセッサを高電力状態から低電力状態に移動させることは、クロック制御論理における動作周波数設定値をより低い周波数設定値に変更することを含む。

【００３７】

電力状態レジスタ２０４は、処理コア２０２の電力／性能状態を示し、処理コア２０２の現在の電力状態を決定するためにソフトウェア命令（例えば、ディスパッチャ）によって読み取ることができる。実施形態では、電力状態レジスタ２０４は、システム内の処理コアの電力／動作パラメータを制御している省電力ソフトウェアによって書き込まれる。処理コア２０２の電力モードが変更される場合には、電力状態レジスタ２０４が更新される。実施形態では、電力状態レジスタ２０４の実装は、特殊目的レジスタと同様であり、診断命令によって書き込まれる。代替的な実施形態では、電力状態レジスタ２０４は、処理コア２０２の外部にあるレジスタ内に置かれるか又はメモリ内の１つ以上のビットとして格納される。

【００３８】

図３は、実施形態によって実装することができる、ディスパッチ・アルゴリズムのフローチャートを示す。実施形態では、ディスパッチ・アルゴリズムは、ハイパーバイザ１１０によって実行される。前述のように、ハイパーバイザ１１０は、データ処理システム内で諸仮想サーバを実装する責任を有し、例えば、多数の異なる仮想プロセッサ間で物理プロセッサのタイム・シェアリングを管理することを含む。実施形態では、ハイパーバイザ１１０内のディスパッチ・アルゴリズムは、諸タスクの実行を開始する。

【００３９】

ブロック３０２では、一のプロセッサ（例えば、プロセッサ・コア２０２）上で実行すべき一のタスクが選択される。実施形態では、一の作動可能キューは、実行する準備ができている１つ以上のタスクを含み、ハイパーバイザ１１０は、この作動可能キューから当該プロセッサ上で実行すべき１つのタスク（例えば、次のタスク）を選択する。選択されたタスクは、ディスパッチ・プロセスによって受け取られる。ブロック３０４では、一の目標サービス・レベル（すなわち、ＳＬＡ）に対する当該タスクの状態が受け取られる。ブロック３０６では、当該プロセッサの電力状態が決定される。実施形態では、この電力状態は、処理コア２０２上に位置する電力状態レジスタ２０４を読み取ることにより決定される。他の実施形態では、ハイパーバイザ１１０は、当該プロセッサの現在の電力状態を追跡し、これをメモリ１０６内に格納する。

【００４０】

ブロック３０８では、当該プロセッサの現在の電力状態が低電力状態であるか否かが決定される。もし、現在の電力状態が低電力状態であれば、ブロック３１２に進み、そこで当該タスクに関連する一の目標サービス・レベルに対する当該タスクの状態が決定される。実施形態では、目標サービス・レベルは、当該タスクに関連するＳＬＡ内で定義される。実施形態では、ハイパーバイザ１１０は、ＳＬＡに対する当該タスクの状態を決定する。実施形態では、ＳＬＡに対する当該タスクの状態は、ディスパッチャと並列に実行中のIBM（登録商標）社のEnterprise Workload Manager（商標）のようなツール使用して追跡される。このツールからのデータは、ブロック３１２で使用するために、ハイパーバイザ１１０に利用可能になる。

【００４１】

実施形態では、当該タスクに関連する現在のサービス・レベルが目標サービス・レベルのしきい値内にある場合、当該タスクのための目標サービス・レベルが満たされている。例えば、もし、ＳＬＡが１秒当たり１００トランザクションのサービス・レベルを指定し、しきい値が３であり、そして現在のサービス・レベルが１秒当たり９８トランザクションであれば、現在のサービス・レベル（１秒当たり９８トランザクション）が目標サービ

ス・レベルの３トランザクション以内にあるから、目標サービス・レベルが満たされている。この例では、目標サービス・レベルに対する当該タスクの状態は、当該タスクが目標サービス・レベルを満たしていることを示すであろう。

【００４２】

実施形態では、ハイパーバイザ１１０は、当該タスクが低電力状態にある当該プロセッサ上で実行される場合、現在のサービス・レベルに対する影響の予測に基づいて、ＳＬＡに対する当該タスクの状態を決定する。もし、現在のサービス・レベルがもはや目標サービス・レベルを満たさないであろうと予測されるならば、当該タスクが目標サービス・レベルを満たしていないことを示すように状態がセットされる。この能力は、ディスパッチャが、当該タスクに関連するＳＬＡを満たすことに関して事前の対策を講じることを可能にする。

【００４３】

ブロック３１２で、目標サービス・レベルが満たされていることが決定される場合、ブロック３１４に進み、そこで当該プロセッサ上で当該タスクの実行が開始される。一方、ブロック３１２で、目標サービス・レベルが満たされていないことが決定される場合は、ブロック３１６に進み、そこで当該プロセッサを高電力状態に移動させることができるか否かという決定が行われる。実施形態では、この決定は、電力状態レジスタ２０４内のフラグを読み取ることにより行われる。他の実施形態では、ハイパーバイザ１１０は、当該プロセッサを高電力状態に移動させることができるか否かを追跡し、フラグをメモリ１０６内に格納する。

【００４４】

当該プロセッサを高電力状態に移動させることができる場合は、ブロック３１８に進み、そこで当該プロセッサを高電力状態に移動させる。次のブロック３２０では、当該プロセッサ上で当該タスクの実行が開始される。ブロック３１６で決定されるように、当該プロセッサを高電力状態に移動させることができなければ、ブロック３２２に進み、そこで他のプロセッサ上で当該タスクの実行が開始される。実施形態では、ディスパッチャは、当該他のプロセッサ上で当該タスクの実行を開始する前に、当該他のプロセッサが高電力状態にあることを確認する。実施形態では、当該他のプロセッサが低電力状態にある場合、ディスパッチャは、当該他のプロセッサ上で当該タスクの実行を開始する前に、当該他のプロセッサを高電力状態に移動させることができることを確認する。実施形態では、当該タスクの実行を開始することは、高電力状態にある一のプロセッサ上で当該タスクを実行すべきであることを指定する標識を、ハイパーバイザ１１０によってアクセス可能なメモリ内にマーク（又は格納）することにより行われる。実施形態では、高電力状態にあるプロセッサが割り込みされた後、当該タスクの実行を開始するためにハイパーバイザ１１０内のディスパッチ・コードが呼び出される。

【００４５】

ブロック３０８で、当該プロセッサが低電力状態にないことが決定される場合、ブロック３１０に進み、そこで当該タスクが高電力状態のタスクであるか否かが決定される。実施形態では、ブロック３１０への入力は、目標サービス・レベルが満たされているか否かというものである。もし、目標サービス・レベルが現在満たされていなければ、当該タスクは高電力状態のタスクである。一方、目標サービス・レベルが現在満たされているか又はそれを超えていれば、当該タスクは高電力状態のタスクではない。もし、当該タスクが高電力状態のタスクであれば、ブロック３２６に進み、そこで当該プロセッサ上で当該タスクの実行が開始される。一方、ブロック３１０で、当該タスクが高電力状態のタスクではないことが決定されるならば、ブロック３２４に進み、そこで当該プロセッサを低電力状態に移動させる。このように、ディスパッチャは、ＳＬＡを満たすのに余分な電力が必要とされない場合、当該タスクを低電力状態で実行することにより、システムによって使用される電力量を制限することができる。

【００４６】

他の実施形態では、３つ以上の電力状態パスがサポートされる。例えば、高電力状態、

中間の電力状態及び低電力状態に対応する、３つの異なるパスが存在し得る。技術的な効果及び利点は、ＳＬＡ内で定義されるような一のサービス・レベルを満たすのに必要な電力量を提供することにより、電力効率を改良するという能力を含む。ＳＬＡ内で定義されるサービス・レベルに基づいて、一のタスクによって利用される電力量を目標とすることができる。もし、ＳＬＡが満たされているか又はこれを超えていれば、当該タスクは、低電力状態にあるプロセッサ上で実行することができる。一方、ＳＬＡが満たされていないか又は（しきい値によって定義されるように）ぎりぎりのところで満たされていなければ、当該タスクは、高電力状態にあるプロセッサ上で実行することができる。当該タスクを高電力状態及び低電力状態のうちどちらの電力状態にあるプロセッサ上で実行すべきかを決定するための入力として、予測データも使用することができる。このようにすると、ＳＬＡ要件を依然として満たしつつ、電力消費量を低減させるか又はシステムによって実行することができる処理の量を増加させることができる。本明細書で使用される用語は、特定の実施形態を説明するためだけのものであるに過ぎず、本発明を制限することは意図されていない。

【００４７】

請求項に記載された全ての手段又はステップ＋機能要素に対応する構造、材料、行為及びそれらの均等物は、請求項に明示的に記載された他の要素と組み合わせてその機能を実施するための任意の構造、材料又は行為を含むことが意図される。本発明に関する記述は、例示及び説明を目的として与えられたものであり、網羅的であること及び開示された形態に本発明を限定することを意図するものではない。当業者にとって、本発明の範囲及び精神から逸脱することなく、多くの修正及び変形を施し得ることが明らかであろう。実施形態は、本発明の原理及び実際の応用を最もよく説明し、考えられる特定の用途に適するような種々の修正を伴う種々の実施形態に関して当業者が本発明を理解することを可能にするために、選択され説明されたものである。

【００４８】

当業者には明らかなように、本発明の諸側面は、システム、方法又はコンピュータ・プログラムとして具体化することができる。従って、本発明の諸側面は、完全にハードウェアの実施形態、（ファームウェア、常駐ソフトウェア、マイクロコード等を含む）完全にソフトウェアの実施形態、又はソフトウェア及びハードウェア側面を組み合わせた実施形態の形式を取ることができ、これらの全てを一般に「回路」、「モジュール」又は「システム」と称することができる。さらに、本発明の諸側面は、コンピュータ可読プログラム・コードを１つ以上のコンピュータ可読媒体上に具体化したコンピュータ・プログラムの形式を取ることができる。１つ以上のコンピュータ可読媒体の任意の組み合わせを利用することができる。コンピュータ可読媒体は、コンピュータ可読信号媒体又はコンピュータ可読ストレージ媒体とすることができる。例えば、コンピュータ可読ストレージ媒体は、電子、磁気、光学、電磁気、赤外線、半導体システム、装置又はこれらの任意の適切な組み合わせとすることができる。コンピュータ可読ストレージ媒体の特定の例は、１つ以上の線を有する電気接続、ポータブル・コンピュータ用のフレキシブル・ディスク、ハード・ディスク、ランダム・アクセス・メモリ（ＲＡＭ）、読み取り専用メモリ（ＲＯＭ）、消去可能プログラマブル読み取り専用メモリ（ＥＰＲＯＭ又はフラッシュ・メモリ）、光ファイバ、ポータブルのコンパクト・ディスクを使った読み出し専用メモリ（ＣＤ－ＲＯＭ）、光ストレージ装置、磁気ストレージ装置又はこれらの任意の適切な組み合わせとすることができる。本明細書の文脈では、コンピュータ可読ストレージ媒体は、命令実行システム等に関連して又はこれらによって使用するためのプログラムを保持するか又は格納することができる、任意の媒体とすることができる。

【００４９】

コンピュータ可読信号媒体は、伝搬されるデータ信号の形式を有することもできるが、その場合には、ベースバンド内に又は搬送波の一部として、コンピュータ使用可能プログラム・コードを具体化することができる。そのような伝搬信号は、電磁気、光学又はその任意の適切な組み合わせ等を含む、種々の形式のうち任意の形式を取ることができる。コ

ンピュータ可読信号媒体は、コンピュータ可読ストレージ媒体ではない任意のコンピュータ可読媒体であって、命令実行システム等に関連して又はこれらによって使用するためのプログラムを通信し、伝搬し又は移送することができる。コンピュータ可読媒体上に具体化されたプログラム信号は、無線、有線、光ファイバ・ケーブル、ＲＦ又はこれらの任意の適切な組み合わせを含む、適切な任意の媒体を使用して伝送することができる。

【００５０】

本発明の諸側面に係る動作を実施するためのコンピュータ・プログラム・コードは、Ｊａｖａ、Ｓｍａｌｌｔａｌｋ、Ｃ＋＋等のようなオブジェクト指向プログラミング言語及び「Ｃ」プログラミング言語又は同様のプログラミング言語のような通常の手続き的プログラミング言語を含む、１つ以上のプログラミング言語の任意の組み合わせで書くことができる。かかるプログラム・コードは、完全にユーザのコンピュータ上で、部分的にはユーザのコンピュータ上で、独立のソフトウェア・パッケージとしてユーザ・コンピュータ上で完全に実行することができ、その一部をユーザ・コンピュータ上で且つ他の一部を遠隔コンピュータ上で実行することができ、或いは遠隔コンピュータ又はサーバ上で完全に実行することができる。後者のシナリオでは、遠隔コンピュータは、ローカル・エリア・ネットワーク（ＬＡＮ）又は広域ネットワーク（ＷＡＮ）を含む任意のタイプのネットワークを通してユーザ・コンピュータに接続することができ、或いはその接続を（例えば、インターネット・サービス・プロバイダを利用するインターネットを通して）外部コンピュータに行うことができる。

【００５１】

以上では、本発明の実施形態に従った方法、装置（システム）及びコンピュータ・プログラムのフローチャート及び／又はブロック図を参照して、本発明の諸側面を説明した。この点に関し、フローチャート及び／又はブロック図の各ブロック、並びにフローチャート及び／又はブロック図の諸ブロックの組み合わせは、複数のコンピュータ・プログラム命令によって実装することができることを理解されたい。これらのコンピュータ・プログラム命令を、汎用コンピュータ、専用コンピュータ又は他のプログラム可能なデータ処理装置のプロセッサに提供すると、前記コンピュータ又は他のプログラム可能なデータ処理装置のプロセッサ上で実行される諸命令が、前記フローチャート又はブロック図の諸ブロックで指定された機能／行為を実装するための手段を作成することを目的として、一のマシンを生産することができる。

【００５２】

また、これらのコンピュータ・プログラム命令をコンピュータ可読媒体内に格納すると、前記コンピュータ可読媒体内に格納された諸命令が、前記フローチャート及び／又はブロック図の諸ブロックで指定された機能／行為を実装する命令手段を含む一の製品を生産することを目的として、コンピュータ又は他のプログラム可能なデータ処理装置に対し特定の態様で機能するように指示することができる。

【００５３】

また、これらのコンピュータ・プログラム命令を、コンピュータ、他のプログラム可能なデータ処理装置又は他の装置にロードすると、前記コンピュータ、他のプログラム可能なデータ処理装置又は他の装置上で実行される諸命令が、前記フローチャート及び／又はブロック図の諸ブロックで指定された機能／行為を実装するためのプロセスを提供することを目的として、一のコンピュータ実装方法を生成するように前記コンピュータ、他のプログラム可能なデータ処理装置又は他の装置上で一連の動作ステップを実行させることができる。

【００５４】

前述のように、諸実施形態は、コンピュータ実装プロセス及びそれらのプロセスを実施するための装置の形式で具体化することができる。実施形態では、本発明は、１つ以上のネットワーク要素によって実行されるコンピュータ・プログラム・コード内で具体化される。諸実施形態は、図４に示すように、製品として有形的媒体内に記録された諸命令を保持する、コンピュータ・プログラム信号論理４０４を備えたコンピュータ使用可能媒体４

０２上のコンピュータ・プログラム４００を含む。コンピュータ使用可能媒体４０２のための推奨製品は、フレキシブル・ディスク、ＣＤ－ＲＯＭ、ハード・ドライブ、ユニバーサル・シリアル・バス（ＵＳＢ）フラッシュ・ドライブ、又は他の任意のコンピュータ可読ストレージ媒体を含むことができる。コンピュータ・プログラム信号論理４０４が一のコンピュータにロードされ且つ当該コンピュータによって実行される場合、当該コンピュータは、本発明を実施するための装置になる。コンピュータ・プログラム・コード論理４０４は、例えば、ストレージ媒体内に格納され、一のコンピュータにロードされ且つ当該コンピュータによって実行され、或いは電気的ケーブル、光ファイバ、電磁放射等の特定の伝送媒体を介して伝送される。その場合、コンピュータ・プログラム・コード論理４０４が一のコンピュータにロードされ且つ当該コンピュータによって実行される場合、当該コンピュータは、本発明を実施するための装置になる。汎用のマイクロプロセッサ上で実装される場合、コンピュータ・プログラム・コード論理４０４のセグメントは、特定の論理回路を作成するようにマイクロプロセッサを構成する。

【００５５】

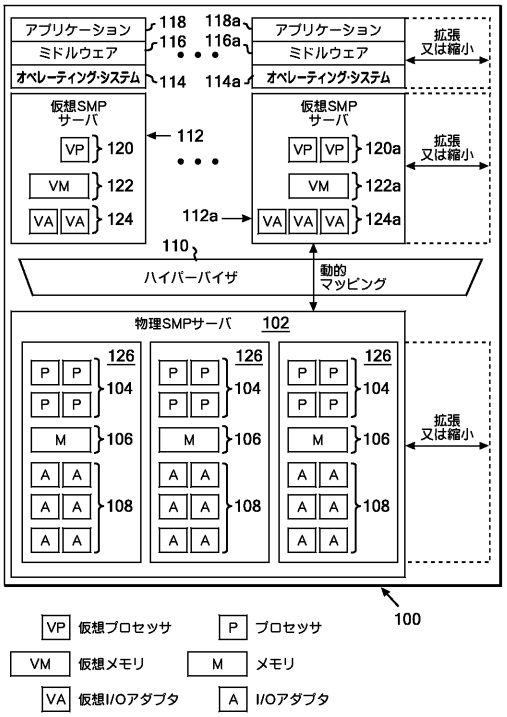
諸図面のうちフローチャート及びブロック図は、本発明の種々の実施形態に従った、システム、方法及びコンピュータ・プログラムの可能な実装のアーキテクチャ、機能性及び動作を示す。この点に関連して、フローチャート又はブロック図内の各ブロックは、指定された論理機能を実装するための１つ以上の実行可能命令から成る、モジュール、セグメント又はコード部分を表すことがある点に留意されたい。また、幾つかの代替的実装では、ブロック内に表記された機能を図面に示した順序とは異なる順序で実施することができる点にも留意されたい。例えば、特定の機能性に依存して、連続的に示した２つのブロックを実質的に並列に実施したり、これらのブロックを反対の順序で実施することができる。さらに、ブロック図又はフローチャートの各ブロック及び複数ブロックの組み合わせは、指定された機能又は行為を実行する専用のハードウェア・ベースのシステム又は専用ハードウェア及びコンピュータ命令の組み合わせによって実装することができる点にも留意されたい。

【符号の説明】

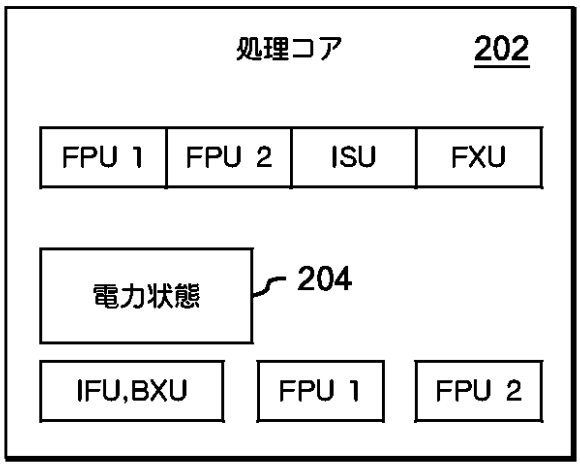
【００５６】

１００・・・ＳＭＰサーバ・コンピュータ・システム
１０２・・・物理ＳＭＰサーバ
１０４・・・プロセッサ
１０６・・・メモリ
１０８・・・Ｉ／Ｏアダプタ
１１０・・・ハイパーバイザ
１１２、１１２ａ・・・仮想ＳＭＰサーバ
１１４、１１４ａ・・・オペレーティング・システム
１１６、１１６ａ・・・ミドルウェア
１１８、１１８ａ・・・アプリケーション
１２０、１２０ａ・・・仮想プロセッサ
１２２、１２２ａ・・・仮想メモリ
１２４、１２４ａ・・・仮想Ｉ／Ｏアダプタ
１２６・・・マイクロプロセッサ・チップ
２０２・・・処理コア（プロセッサ）
２０４・・・電力状態レジスタ

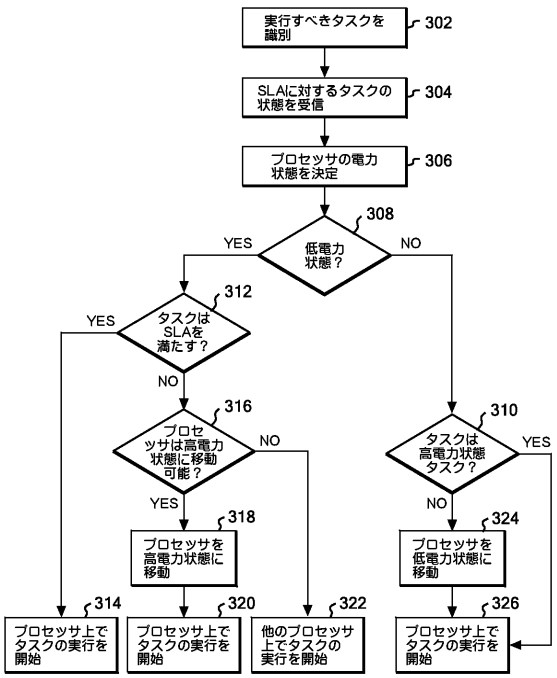
【図１】


アプリケーション
ミドルウェア
オペレーティング・システム
118
116
114
118a
116a
114a
拡張
又は縮小
仮想SMP
サーバ
VP
VM
VA
120
122
124
112
120a
122a
124a
112a
110
ハイパーバイザ
動的
マッピング
物理SMPサーバ
102
126
104
106
108
P
M
A
100
仮想プロセッサ
プロセッサ
仮想メモリ
メモリ
仮想I/Oアダプタ
I/Oアダプタ

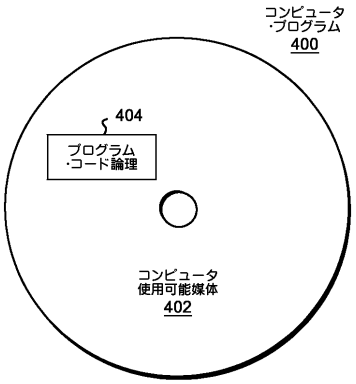
【図２】


処理コア
202
FPU 1
FPU 2
ISU
FXU
電力状態
204
IFU,BXU
FPU 1
FPU 2

【図３】


実行すべきタスクを識別
302
SLAに対するタスクの状態を受信
304
プロセッサの電力状態を決定
306
低電力状態？
308
YES
NO
タスクはSLAを満たす？
312
YES
NO
プロセッサは高電力状態に移動可能？
316
NO
YES
タスクは高電力状態タスク？
310
YES
NO
プロセッサを高電力状態に移動
318
プロセッサを低電力状態に移動
324
プロセッサ上でタスクの実行を開始
314
プロセッサ上でタスクの実行を開始
320
他のプロセッサ上でタスクの実行を開始
322
プロセッサ上でタスクの実行を開始
326

【図４】


コンピュータ・プログラム
400
404
プログラム・コード論理
コンピュータ使用可能媒体
402

フロントページの続き

(72)発明者　ボイド、ウイリアム　トッド
アメリカ合衆国１２６０１－５４００　ニューヨーク州　ポキプシー　サウス・ロード　２４５５　エムディー３１５
(72)発明者　ヘラー　ジュニア、トマス
アメリカ合衆国１２６０１－５４００　ニューヨーク州　ポキプシー　サウス・ロード　２４５５　エムディー３１５

審査官　大塚　俊範

(56)参考文献　特開２００４－３２６３６４（ＪＰ，Ａ）
米国特許出願公開第２００６／０１４９９８５（ＵＳ，Ａ１）
国際公開第２００９／０６１４３２（WO，Ａ１）
米国特許出願公開第２００６／０１２３２５１（ＵＳ，Ａ１）
米国特許出願公開第２００４／０２１５９８７（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４６　９／５４