

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6668355号
(P6668355)

(45) 発行日 令和2年3月18日(2020.3.18)

(24) 登録日 令和2年2月28日(2020.2.28)

(51) Int. Cl.	F I	
G06F 1/3296 (2019.01)	G06F	1/3296
G06F 1/324 (2019.01)	G06F	1/324
G06F 1/3234 (2019.01)	G06F	1/3234
H02J 1/00 (2006.01)	H02J	1/00 307F

請求項の数 25 (全 26 頁)

(21) 出願番号	特願2017-533567 (P2017-533567)	(73) 特許権者	593096712 インテル コーポレーション アメリカ合衆国 95054 カリフォル ニア州 サンタ クララ ミッション カ レッジ ブールバード 2200
(86) (22) 出願日	平成27年10月13日 (2015.10.13)	(74) 代理人	100107766 弁理士 伊東 忠重
(65) 公表番号	特表2018-503184 (P2018-503184A)	(74) 代理人	100070150 弁理士 伊東 忠彦
(43) 公表日	平成30年2月1日 (2018.2.1)	(74) 代理人	100091214 弁理士 大貫 進介
(86) 国際出願番号	PCT/US2015/055353		
(87) 国際公開番号	W02016/105646		
(87) 国際公開日	平成28年6月30日 (2016.6.30)		
審査請求日	平成30年10月5日 (2018.10.5)		
(31) 優先権主張番号	14/581, 911		
(32) 優先日	平成26年12月23日 (2014.12.23)		
(33) 優先権主張国・地域又は機関	米国 (US)		

最終頁に続く

(54) 【発明の名称】 動的時間的電力ステアリングのためのシステム及び方法

(57) 【特許請求の範囲】

【請求項1】

電力マネージャを使用して、ノード上で実行するためのタスクを含むアプリケーションに対応する、フェーズのシーケンスを決定するステップであって、該シーケンスは電力上限に関連する時間間隔に対応する、ステップと、

各フェーズに対して、前記ノード内の複数の電力ドメインに分配される電力の増加又は減少のため、フェーズの実行時間が電力にどのようにスケールするかを測定して、電力スケールを決定するステップと

各フェーズに対する前記電力スケールに基づいて、前記シーケンスのフェーズ間での時間的電力分配を決定して、前記電力上限を満たし、前記ノードによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するステップと、を含み、

パフォーマンス指標と、目的関数と、前記パフォーマンス指標を評価するための測定間隔及び測定イベントのうちの少なくとも一つと、を選択するステップをさらに含み、

前記パフォーマンス指標は、前記ノードによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するように選択され、

前記目的関数は、前記電力上限内でのパフォーマンスを最大化すること、パフォーマンスを最大化すること、前記電力上限内でエネルギー効率を最大にすること、及び効率を最大にすることを含む群から選択される、方法。

【請求項2】

現在フェーズを検出するために前記アプリケーションをモニタするステップと、

前記現在フェーズにおいて、前記時間的電力分配が前記目的関数を満たさないと決定するステップと、

前記決定に応じて、前記シーケンスのフェーズ間で前記電力を時間的に再分配するステップと、をさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記シーケンスのフェーズ間で前記電力を時間的に再分配するステップは、前記シーケンスのフェーズ間で新しい時間的電力分配を選択するステップと、前記時間的電力分配に基づいて、前記現在フェーズが省電力又は加速のために指定されているかどうかを決定するステップと、

前記現在フェーズが省電力のために指定されている場合、前記新しい時間的電力分配に従って前記ノード内の複数の電力ドメインに対する電力を下げるステップと、

前記現在フェーズが加速のために指定されている場合、前記新しい時間的電力分配に従って前記ノード内の複数の電力ドメインに対する電力を上げるステップと、を含む、請求項 2 に記載の方法。

【請求項 4】

前記現在フェーズが省電力又は加速のために指定されているかどうかを示す、前記アプリケーションのコンピュータ実行可能コード内のアノテーションを検出するステップをさらに含む、請求項 3 に記載の方法。

【請求項 5】

前記ノード内の複数の電力ドメイン間の電力分配が前記パフォーマンス指標を満たさないと決定するステップと、

前記決定に応じて、前記新しい時間的電力分配に基づいて、前記現在フェーズにおける前記複数の電力ドメイン間の前記電力を再分配するステップと、をさらに含む、請求項 3 に記載の方法。

【請求項 6】

前記アプリケーション、前記ノード、前記電力上限及び前記ノードを含むシステムのうち少なくとも一つの変化を検出するステップと、

検出された前記変化に応じて、前記シーケンスのフェーズ間の時間的電力分配を再構成するステップと、をさらに含む、請求項 1 に記載の方法。

【請求項 7】

計算結合フェーズ、記憶結合フェーズ、通信結合フェーズ、アイドルフェーズ、アクティブフェーズ及びそれらのフェーズの任意のものの組み合わせのうち少なくとも 2 つを検出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】

前記シーケンスの各フェーズに対して、前記シーケンスの第 1 反復に基づいた前記時間的電力分配を決定するステップと、

前記シーケンスの各フェーズに対して、前記シーケンスの第 2 反復に前記時間的電力分配を適用するステップと、をさらに含む、請求項 1 に記載の方法。

【請求項 9】

少なくとも一部にハードウェアを含むロジックを含む電力マネージャであって、該電力マネージャは、

ノード上で実行するためのタスクを含むアプリケーションのフェーズのシーケンスを決定するフェーズモジュールであって、該シーケンスは電力上限に関連する時間間隔に対応し、各フェーズは、他のフェーズの動作挙動とは異なる動作挙動を呈するアプリケーションのセグメントを有する、フェーズモジュールと、

前記シーケンスのフェーズ間の時間的電力分配を決定して、前記電力上限を満たし、前記アプリケーションの実行時間パフォーマンスを改善又は最適化する動的時間的電力スケジューラモジュールと、を含み、

前記電力マネージャは、パフォーマンス指標と、目的関数と、前記パフォーマンス指標を評価するための測定間隔及び測定イベントのうちの少なくとも一つと、を選択するよう

10

20

30

40

50

に構成され、

前記目的関数は、前記アプリケーションの実行時間パフォーマンスを改善又は最適化するように選択される、装置。

【請求項 10】

前記フェーズモジュールは、さらに、現在フェーズを検出するために前記アプリケーションをモニタするように構成され、

前記動的時間的電力スケジューラモジュールは、さらに、

前記現在フェーズにおいて、前記時間的電力分配が前記目的関数を満たさないと決定し

、
前記決定に応じて、前記シーケンスのフェーズ間で電力を時間的に再分配する、ように構成されている、請求項 9 に記載の装置。 10

【請求項 11】

前記電力を時間的に再分配するために、前記動的時間的電力スケジューラモジュールは、さらに、

前記シーケンスのフェーズ間で新しい時間的電力分配を選択し、

前記時間的電力分配に基づいて、前記現在フェーズが省電力又は加速のために指定されているかどうかを決定し、

前記現在フェーズが省電力のために指定されている場合、前記新しい時間的電力分配に従って、前記ノード内の複数の電力ドメインに対する電力を下げ、

前記現在フェーズが加速のために指定されている場合、前記新しい時間的電力分配に従って前記ノード内の複数の電力ドメインに対する電力を上げる、ように構成されている、請求項 10 に記載の装置。 20

【請求項 12】

前記動的時間的電力スケジューラモジュールは、さらに、前記現在フェーズが省電力又は加速のために指定されているかどうかを示す、前記アプリケーションのコンピュータ実行可能コード内のアノテーションを検出するように構成されている、請求項 11 に記載の装置。

【請求項 13】

前記電力マネージャは、さらに、

前記ノード内の複数の電力ドメイン間の電力分配が前記目的関数を満たさないと決定し 30

、
前記決定に応じて、前記新しい時間的電力分配に基づいて、前記現在フェーズにおける前記複数の電力ドメイン間の前記電力を再分配する、ように構成されている請求項 11 に記載の装置。

【請求項 14】

前記電力マネージャは、さらに、

前記アプリケーション、前記ノード、前記電力上限及び前記装置を含むシステムのうちの少なくとも一つの変化を検出し、

検出された前記変化に応じて、前記シーケンスのフェーズ間の時間的電力分配を再構成する、ように構成されている、請求項 9 から 13 のいずれか一項に記載の装置。 40

【請求項 15】

システムであって、

少なくとも一つのコアドメインと、少なくとも一つのアノコアドメインとを含む複数の電力ドメインと、

一つ以上のプロセッサによって実行されると、該一つ以上のプロセッサに動作を実行させる命令を記憶した少なくとも一つのコピュータ読み取り可能記憶媒体と、を含み、

該動作は、

当該システムによって実行されるアプリケーションのフェーズのシーケンスを決定するステップであって、該シーケンスは、電力上限に関連する時間間隔に対応する、ステップと、

各フェーズに対して、当該システム内の前記複数の電力ドメインに分配される電力の増加又は減少のため、フェーズの実行時間が電力にどのようにスケールするかを測定して、電力スケールを決定するステップと、

各フェーズに対する前記電力スケールに基づいて、前記シーケンスのフェーズ間での時間的電力分配を決定して、前記電力上限を満たし、当該システムによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するステップと、を含み、

前記動作は、パフォーマンス指標と、目的関数と、前記パフォーマンス指標を評価するための測定間隔及び測定イベントのうち少なくとも一つと、を選択するステップをさらに含み、

前記パフォーマンス指標は、当該システムによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するように選択される、システム。

10

【請求項 16】

前記動作は、

現在フェーズを検出するために前記アプリケーションをモニタするステップと、

前記現在フェーズにおいて、前記時間的電力分配が前記目的関数を満たさないと決定するステップと、

前記決定に応じて、前記シーケンスのフェーズ間で前記電力を時間的に再分配するステップと、をさらに含む、請求項 15 に記載のシステム。

【請求項 17】

前記シーケンスのフェーズ間で前記電力を時間的に再分配するために、前記動作は、

前記シーケンスのフェーズ間で新しい時間的電力分配を選択するステップと、

前記時間的電力分配に基づいて、前記現在フェーズが省電力又は加速のために指定されているかどうかを決定するステップと、

前記現在フェーズが省電力のために指定されている場合、前記新しい時間的電力分配に従って当該システム内の複数の電力ドメインに対する電力を下げるステップと、

前記現在フェーズが加速のために指定されている場合、前記新しい時間的電力分配に従って当該システム内の複数の電力ドメインに対する電力を上げるステップと、をさらに含む、請求項 16 に記載のシステム。

20

【請求項 18】

前記動作は、前記現在フェーズが省電力又は加速のために指定されているかどうかを示す、前記アプリケーションのコンピュータ実行可能コード内のアノテーションを検出するステップをさらに含む、請求項 17 に記載のシステム。

30

【請求項 19】

前記動作は、

当該システム内の複数の電力ドメイン間の電力分配が前記パフォーマンス指標を満たさないと決定するステップと、

前記決定に応じて、前記新しい時間的電力分配に基づいて、前記現在フェーズにおける前記複数の電力ドメイン間の前記電力を再分配するステップと、をさらに含む、請求項 17 に記載のシステム。

【請求項 20】

前記フェーズのシーケンスを決定するために、前記動作は、

前記アプリケーション、前記電力上限及び当該システムのうち少なくとも一つの変化を検出するステップと、

検出された前記変化に応じて、前記シーケンスのフェーズ間の時間的電力分配を再構成するステップと、をさらに含む、請求項 15 から 19 のいずれか一項に記載のシステム。

40

【請求項 21】

前記フェーズのシーケンスを決定するために、前記動作は、計算結合フェーズ、記憶結合フェーズ、通信結合フェーズ、アイドルフェーズ、アクティブフェーズ及びそれらのフェーズの任意のもの組み合わせのうち少なくとも2つを検出するステップをさらに含む、請求項 15 から 19 のいずれか一項に記載のシステム。

50

【請求項 2 2】

前記動作は、

前記シーケンスの各フェーズに対して、前記シーケンスの第 1 反復に基づいた前記時間的電力分配を決定するステップと、

前記シーケンスの各フェーズに対して、前記シーケンスの第 2 反復に前記時間的電力分配を適用するステップと、をさらに含む、請求項 1 5 から 1 9 のいずれか一項に記載のシステム。

【請求項 2 3】

実行されると、請求項 1 から 8 のいずれか一項に記載の方法を実施するプログラム。

【請求項 2 4】

請求項 1 から 8 のいずれか一項に記載の方法を実行する手段を含む装置。

【請求項 2 5】

請求項 2 3 に記載のプログラムを記憶している機械読み取り可能媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、概して、サーバ及び他の計算装置のための電力管理に関する。

【背景技術】

【0002】

半導体処理及びロジック設計の進歩は、集積回路装置上に存在し得るロジックの量の増加を可能にしてきた。その結果、コンピュータシステム構成は、システム内に複数の集積回路があるものから、個々の集積回路上に複数のハードウェアスレッド、複数のコア、複数の装置及び/又は完全なシステムがあるものに発展してきた。集積回路の集積度が高まるにつれて、計算システム（組み込みシステムから高パフォーマンス計算（HPC）システム）への電力要件も高まっている。

【0003】

電力及び熱管理の問題は、コンピュータベースのシステムを設計する際の考慮事項である。HPCシステムでは、例えば、指定された熱設計電力（TDP）及び/又は電力コストが、低電力システムの必要性を促進し得る。モバイルシステムを含む他のシステムにおいても、バッテリー寿命と熱的制約により、それらの事項と関連がある。最低限の電力消費で最大のパフォーマンスを得るためのシステムの最適化は、通常では、オペレーティングシステム（OS）又はハードウェア要素を制御するシステムソフトウェアを使用してなされる。

【0004】

電力及び熱制御のためのいくつかのモデルは、比較的大きく、動的に調整可能な時間ウィンドウで、ハードウェア装置の実行平均電力をモニタし、所与の閾値以下にこの平均電力を維持することができる。このようなアプローチにより、装置は短期間（実行平均電力がリミット内に留まる限り）であれば上記のリミットを超えた電力を消費することができるが、装置の電源と冷却ソリューションはエクスカージョン（excursion）の大きさと長さに制限を設ける。

【図面の簡単な説明】

【0005】

アレンジメント及び実施形態は、次の図面を参照して詳細に説明されることができる。ここで、類似の参照符号は、類似の要素を示す。

【0006】

【図 1】図 1 は、本明細書で説明する、様々な実施形態を実装するのに適した例示的なコンピュータシステムの実施形態を示す。

【0007】

【図 2 A】図 2 A は、いくつかの実施形態による、例示的なフェーズのシーケンスの繰り返しを示す。

10

20

30

40

50

【図2B】図2Bは、いくつかの実施形態による、例示的なフェーズのシーケンスの繰り返しを示す。

【図2C】図2Cは、いくつかの実施形態による、例示的なフェーズのシーケンスの繰り返しを示す。

【0008】

【図3】図3は、いくつかの実施形態による、装置のブロック図である。

【0009】

【図4】図4は、一実施形態による、動的時間的電力ステアリングのための方法のフローチャートである。

【0010】

【図5】図5は、一実施形態による、時間的電力分配及び空間電力ステアリングの両方のためのフローチャートである。

【発明を実施するための形態】

【0011】

本明細書で開示される実施形態は、概して、コンピュータシステム上の作業負荷を管理するための装置、方法及び他の技術を対象とする。コンピュータシステムによって実行されるソフトウェアアプリケーションは、共通の目的（例えば、ある問題に対する解答を計算する）に向かって一緒に作業する一群のタスクを含むことができる。タスクは、一つ以上のノード上で実行することができる。ノードの内部には、一つ以上のタスクがあることができ、各タスクは一つ以上の処理スレッドを含むことができる。特定の実施形態は、装置又はコンピュータシステムによって実行されるアプリケーションのフェーズ間で動的時間的電力ステアリング（dynamic temporal power steering）を提供する。各フェーズに対して、電力は、コンピュータシステム内の複数の電力ドメイン間で分配される。

【0012】

用語「電力ドメイン」又は単にドメインは、本明細書では、電力消費に関してモニタ及び制御されることができるコンポーネント又はコンポーネントのセットを示すために使用される。特定の実施形態は、アプリケーション又はシステムの挙動の実行時間を、安定した又は予測可能な動作挙動を呈するセグメントに自動的に分割する。安定した又は予測可能なセグメントは、本明細書では、アプリケーションの「フェーズ」と呼ばれる。各フェーズは、例えば、他のフェーズの動作挙動と異なる動作挙動を呈するアプリケーションのセグメントを含む。フェーズは、例えば、計算結合フェーズ（compute bound phase）、記憶結合フェーズ、通信結合フェーズ、計算/記憶/通信の混合フェーズ、アイドルフェーズ、アクティブフェーズ、任意の動作挙動に結合されるその他のフェーズを含むことができる。アプリケーションの異なるフェーズは、ノード（例えば、サーバ）内のリソース又はドメイン（例えば、コア、アンコア、メモリなど）が異なることがある。

【0013】

時間次元においてフェーズ境界を越えたより効率的な電力使用のための機会を特定することによって、アプリケーションのパフォーマンスは最適化又は改善されることができる。例えば、動的時間的電力ステアリングは、記憶結合フェーズのパフォーマンスを潜在的に低下させながら、記憶結合フェーズ中、コンピュータシステムの電力リミットを下げることを特定することができ、その電力を計算結合フェーズに与えることで（すなわち、計算結合フェーズ中、コンピュータシステムの電力リミットを比例的に上げる）アプリケーション全体のパフォーマンスを向上させることができる。

【0014】

動的時間的電力ステアリングの実施形態は、全体としてアプリケーションの挙動を決定又は学習し、電力をあるフェーズから別のフェーズに移す（すなわち、コンピュータシステムの電力リミットを上げる又は下げる）ことによってパフォーマンスを改善する機会を特定する。以下に詳細に説明するように、例えば、ある実施形態では、電力にあまりスケールしないパフォーマンス（例えば、パフォーマンスの向上がパフォーマンスを上げるために必要とされる電力増加に比べて不相応に低い）のフェーズ中は電力を意図的に確

10

20

30

40

50

保しておき (reserve)、電力に良好にスケーリングするパフォーマンスのフェーズのパフォーマンスを引き上げるために電力をさらに使用することで、エネルギーバジェットを全体的に維持しながら、アプリケーションのより高いパフォーマンスをもたらす。

【0015】

このような実施形態は、他の電力管理アプローチと比べて、アプリケーションパフォーマンス又は電力管理の他の目的関数 (例えば、エネルギー遅延プロダクト (EDP: energy-delay product) 又はエネルギー遅延スクエアプロダクト (ED²P: energy-delay-square product) のようなシステム効率指標を改善すること) における明らかな向上を提供することができる。他のアプローチは、電力管理の目的関数 (例えば、パフォーマンス、効率性、EDP、ED²P等) に決して害を及ぼさないというパラダイムの下で動作する一方で、本明細書で開示される特定の実施形態は、いくつかのフェーズでは電力管理の目的関数に意図的に「ダメージ」を与え、他のフェーズでより大きな改善を生み出すことができ、アプリケーション全体としての電力管理の目的関数を改善するようにする。

10

【0016】

本明細書での説明の目的のため、特定の例を高パフォーマンス計算 (HPC) システムに関して説明することがある。しかし、本開示の範囲はHPCの例に限定されず、実施形態は、任意の大規模なクラウド、ビッグデータ、HPC、他のシステム等の高電力コンピュータシステムから、ラップトップ、UltraBook (登録商標)、タブレットコンピュータ、スマートフォン、他のポータブル装置等の低電力コンピュータシステムに至るまで多くの様々なタイプのコンピュータシステムに対する解決手段を可能とするように高度に拡張可能である。実施形態は、デスクトップコンピュータなどの高電力システムと低電力システムとの間にある電力要件を有するシステムにも等しく適用される。

20

【0017】

他の解決策は、アプリケーション実行全体にわたる平均電力のような特性に基づいてドメインに電力を静的に割り当てる、又は電力をモニタして動的に割り当てるが、直近のk時間ウィンドウでの挙動履歴にわたるアプリケーション特性に基づいて行う。電力がドメインに静的に割り当てられる場合、特定のドメインが全ての電力割り当てを使用しているときや、より多くの電力が与えられた場合にドメインがより高速に実行できるときや、ドメインが全電力バジェットを使用しないとき (すなわち、電力が取り残される) があり得る。時間ウィンドウでの挙動履歴に基づいて電力を動的に割り当てる解決手段は、全体を通してずっと安定した状態の挙動 (単一フェーズ) を備えたアプリケーションの電力の取り残しを改善することができるが、異なる特性を備えた複数のフェーズを有するアプリケーションではほとんど、又は全く改善されない可能性がある。このようなソリューションは、インターバルの間に (部分的に又は完全に) 実行する任意のフェーズの挙動をブレンドする時間ウィンドウ (固定又は可変) で挙動をモニタする。このようなソリューションは、過去のウィンドウでブレンドされた挙動が将来の挙動の良い予測値になることを想定しており、各時間ウィンドウが異なる動作挙動及び異なる最適な電力割り当てで完全に異なるアプリケーションフェーズを含むことがある。

30

【0018】

HPCソフトウェアアプリケーションは、異なる動作挙動を備えた (したがって、ドメイン内で異なるアクティビティを備えた) 複数のフェーズを有するので、ドメインへの電力の静的な割り当てでは最適ではない可能性がある。例えば、コンピュータが2つのドメインを含み、一方がプロセッサであり、他方がメインメモリ (例えば、ダイナミックランダムアクセスメモリ (DRAM)、不揮発性メモリ又は他のメモリ装置) であると仮定する。コンピュータが平均電力100Wを超えることができず、アプリケーションのフェーズ全体にわたるアプリケーションの特性を通じて、最適な静的割り当てはプロセッサに対しては50Wの平均電力リミットであり、メインメモリに対しては50Wの平均電力リミットであると決定されたと仮定する。この例では、アプリケーション実行時間 (runtime) を大幅に増加させることなく記憶結合フェーズで平均電力は50W未満に下げられないので、50Wが最適であると決定される。しかし、アプリケーションフェーズのうち計算

40

50

結合フェーズ及び通信結合フェーズ中、主メモリドメインに予約されている50Wをフルに利用する十分な記憶アクティビティがない。このように、静的割り当ては、取り残された電力（現在のドメインが必要とするよりも必要とするドメインのために確保された電力）及び準最適なパフォーマンスをもたらす。

【0019】

HPCクラスタ又はスーパーコンピュータは、アプリケーションが最高パフォーマンスに達するために必要とする電力よりも電力上限が低くなるように「電力制約」をかけて実行することができる。電力制約されたシナリオでは、可能な限り効率的に電力を使用することでパフォーマンスを向上させる。電力上限は、定義された期間（time period）にわたってシステムに使用されることができる最大平均電力である。しかし、所定の期間にわたる平均使用電力（average power usage）が電力上限を超えない限り、電力上限は便宜的に超過されることができる。

10

【0020】

特定の実施形態では、様々な電力ドメインが、実行平均電力リミット（RAPL）に基づいてモニタされ、制御される。電力マネージャは、利用可能な電力ドメインの各々に対してRAPLを指定し、電力制御及び/又は個別プロセッサパフォーマンス状態（P-state）などの低レベルパフォーマンスパラメータを変調することによってRAPL電力制限を実施することができる。特定のこのような実施形態では、動的空間電力ステアリングは、アプリケーションフェーズのそれぞれについて様々なRAPL電力ドメイン間の電力の最適な割り当てを学習することを含む。HPCアプリケーション及び他のアプリケーションは、同じフェーズが何度も起きる繰り返し挙動を示すことがある。特定の実施形態は、この繰り返し挙動を活用する。特定のフェーズについて最適又は改善された電力割り当てが学習又は決定されると、アプリケーションが将来的に同じフェーズに入るときにそれが（例えば、即座に）再利用されることができる。

20

【0021】

様々な実施形態はまた、本明細書で開示される動作を実行するための装置又はシステムに関する。この装置は、必要とされる目的のために特別に構築されることができる、あるいは、コンピュータに記憶されたコンピュータプログラムによって選択的に起動又は再構成される汎用コンピュータを含むこともできる。本明細書に提示される手順は、特定のコンピュータ又は他の装置に固有のものではない。様々な汎用機械が、本明細書の教示に従って書かれたプログラムとともに使用されることができ、あるいは開示された方法を実行するのにより特化した装置を構成することが好都合であることを示すこともできる。様々なこれらの機械のために必要とされる構造は、与えられる説明から明らかとなるものである。

30

【0022】

ここで、図面を参照する。類似する参照符号は、全体を通して類似の要素を示すのに使用される。以下の説明では、説明のため、その完全な理解を提供するために、多くの具体的な詳細を述べる。しかし、実施形態は、これらの具体的な詳細なしに実施することができることは明らかである。他の例では、周知の構造及び装置は、その説明を容易にするために、ブロック図で示される。これは、特許請求の範囲に記載される主題と一致するすべての修正物、均等物及び代替物をカバーすることを意図している。

40

【0023】

図1は、本明細書で説明される様々な実施形態を実装するのに適した例示的なコンピュータシステム105の実施形態を示す。様々な実施形態において、コンピュータシステム105は、処理装置110と、コントローラ120と、メモリ装置130と、入出力（I/O）装置140とを含む。処理装置110は、（例えば、サーバラック内のベイ内の）サーバ、マイクロプロセッサ、プロセッサ、中央処理装置、デジタル信号処理ユニット、デュアルコアプロセッサ、モバイルデバイスプロセッサ、デスクトッププロセッサ、シングルコアプロセッサ、システムオンチップ（SoC）デバイス、複合命令セットコンピュータ（CISC）マイクロプロセッサ、縮小命令セットコンピュータ（RISC）マイク

50

ロプロセッサ、超長命令語（VLIW）マイクロプロセッサ、若しくは任意の他のタイプのプロセッサ又は単一チップ若しくは集積回路上の処理回路等の任意のタイプの計算要素であり得るが、これらに限定されない。処理装置110は、相互接続122、132、142を介してコントローラ120、メモリ装置130及びI/O装置140にそれぞれ接続され、それらと通信することができる。さらに、コントローラ120、メモリ装置130及びI/O装置140は、処理装置110の全てのコンポーネントと通信することができる。

【0024】

この点に関して限定されるものではないが、処理装置110は、電力管理装置115と、一つ以上の処理コア112と、グラフィックス処理ユニット114と、I/O装置116と、メモリ装置118と、他のアンコア（uncore）（非コア（non-core））コンポーネント（図示せず）とを含むことができる。アンコアコンポーネントは、他のメモリ、キャッシュ、I/O装置及びコントローラを含む。いくつかの実施形態では、処理装置110は、例えば、3つ以上の処理コアを含むことができる。一つ以上の処理コア112は、2つ以上の処理ユニットを有することができる。一つ以上の処理コア112は、命令をフェッチするためのプリフェッチロジック、命令をデコードするためのデコードロジック、命令を実行するための実行ロジック及び他のロジックを含むが、これらに限定されない。

10

【0025】

特定の実施形態では、処理装置110は、グラフィックス処理ユニット114を含む。グラフィックス処理ユニット114は、グラフィックス及び/又はビデオの動作及び/又は処理を管理する補助プロセッサとして機能することができる。

20

【0026】

いくつかの実施形態では、処理装置110は、I/O装置116及びメモリ装置118を含む。I/O装置116は、コンピュータシステム105内の他の装置への、処理装置110のための外部インタフェースを提供する。追加的に、メモリ装置118は、処理装置110及び特に、一つ以上の処理コア112の命令及び/又はデータを記憶する。メモリ装置118はまた、処理装置110及び一つ以上の処理コア112が命令を実行している間、一時変数又は他の中間情報を記憶することができる。他の実施形態では、メモリ装置118は、レベル1（L1）、レベル2（L2）及びレベル3（L3）キャッシュメモリ、又は処理装置110内にキャッシュメモリの任意の他の構成を含むが、これらに限定

30

【0027】

コントローラ120は、マイクロコントローラ又は他のタイプの処理回路、メモリ及びインタフェースロジックを含むことができる。いくつかの実施形態では、コントローラ120は、メモリ装置130及びI/O装置140の動作のモニタ及び制御すること（これに限定されない）を含め、コンピュータシステム105の動作をモニタ及び制御することができる。コントローラ120はまた、コンピュータシステム105の様々なコンポーネントと処理装置110との間の通信（これに限定されない）を含め、処理装置110の様々な側面をモニタ及び制御することができる。様々な実施形態において、コントローラは、相互接続122を介して処理装置110に結合されることができる。

40

【0028】

メモリ装置130は、コンピュータシステム105の命令及び/又はデータを記憶する。メモリ装置130はまた、コンピュータシステム105が命令を実行している間、一時変数又は他の中間情報を記憶することができる。I/O装置140は、コンピュータシステム105と外部システム又は装置との間のインタフェースである。I/O装置140は、接続144を介して外部システム又は装置と通信することができる。接続144は、有線若しくは無線又は当業者に知られている他の任意の手段であることができる。一実施形態では、I/O装置140は、I/O装置116を通じて相互接続142を介して処理装置110と通信することができる。電力管理装置115、メモリ装置130、及びI/O

50

装置 140 はまた、相互接続（図示せず）を介して互いに通信することができる。

【0029】

コンピュータシステム 105 に図示されているコンポーネントは、限定を意味するものではなく、説明のために提示されている。他の実施形態では、コンピュータシステム 105 の構成は、図 1 に示されている以外の他のコンポーネントを（多かれ少なかれ）含む。当業者であれば、本明細書に記載される実施形態の作業に影響を及ぼすことなく、コンピュータシステム 105 の他の構成を使用できることを理解するであろう。

【0030】

特定の実施形態では、電力管理装置 115 は、全体の電力リミットなどの全体のシステム制約を遵守しながら、特定のアプリケーションの複数の異なるフェーズ間で電力を再分配する機械を特定する動的時間的電力ステアリングを提供するように構成されている。電力ドメインは、例えば、一つ以上の処理コア 112 を含むコアドメインと、グラフィックス処理ユニット 114、I/O 装置 116、メモリ装置 118 及び他の任意のアンコアダメイン（図示せず）を含むアンコアダメインとを含むことができる。追加的に、又は他の実施形態では、一つ以上の処理コア 112 は、複数の電力ドメインを含むことができ、各電力ドメインがコア（必ずしも均一なサイズではない）に対応する。さらに、図 1 に示す任意の要素（すなわち、コントローラ 120、メモリ装置 130、I/O 装置 140、処理装置 110、電力管理装置 115、一つ以上の処理コア 112、グラフィックス処理ユニット 114、I/O 装置 116 及びメモリ装置 118）が別個の電力ドメインであってよく、別々の電力ドメイン内の任意の他の要素と組み合わせられてもよく、及び/又は複数の異なる電力ドメインに分割されてもよい。

【0031】

一つ以上の処理コア 112 は、同じフェーズが何度も何度も起きる挙動を示すアプリケーションを実行することができる。各フェーズは、異なるリソースを必要とし、処理装置 110 上のリソースにおいて異なる量のアクティビティを生成することができるので、処理装置 110 上のリソース又はドメイン間の電力の最適な割り当ては、各フェーズで異なることがある。最適又は改善されたパフォーマンスを達成するために（又は、効率、EDP、ED2P 等の他の電力管理目的関数を最大化するために）、電力管理装置 115 は、ある実施形態によれば、異なる電力ドメイン間で分配された電力を増加させ又は減少させ、アプリケーションの異なるフェーズ間で電力を再分配するようにする。

【0032】

特定のアプリケーションは、アプリケーションがその作業を完了するために繰り返すときに、フェーズのシーケンスが数回発生する挙動を示すことがある。例えば、図 2A、図 2B 及び図 2C は、特定の実施形態による例示的なフェーズのシーケンスの繰り返しを示す。図 2A は、フェーズ「A」に、フェーズ「B」が続き、フェーズ「C」が続き、フェーズ「A」が続き、フェーズ「B」が続くものを含むフェーズのシーケンス 200 を示す。このシーケンスは、「反復 1」に続いて、「反復 2」が続き、「反復 3」が続くもので繰り返される。そのような繰り返しは、特定の動的時間的電力ステアリングの実施形態を通じたパフォーマンスの最適化に適している。例えば、電力管理装置 115 は、フェーズ A、フェーズ B 及びフェーズ C が繰り返しており、多くの反復で繰り返す可能性が高いと認識するように構成することができる。それに応じて、電力管理装置 115 は、一つ以上の反復にわたって、フェーズ A、フェーズ B 及びフェーズ C に対する電力ドメイン間での最適又は改善された時間的電力割り当てを決定する（例えば、あるフェーズでは電力を節約又は制限し、別のフェーズでは電力リミットを適用する又は上げる）。電力管理装置 115 は、その後の各反復において、フェーズ A、フェーズ B 及び/又はフェーズ C の間での最適又は改善された電力分配を適用する。

【0033】

アプリケーションは、電力をより効率的に割り当てるために観察され、活用され得る、明確な動作挙動及びリソースアクティビティパターンを備えたフェーズを有することができる。例えば、図 2B は、記憶フェーズ 212 に、計算フェーズ 214 が続き、通信フェ

10

20

30

40

50

ーズ 2 1 6 が続き、記憶フェーズ 2 1 2 が続き、計算フェーズ 2 1 4 が続くものを含むフェーズのシーケンス 2 1 0 を示す。記憶フェーズ 2 1 2 は、コンピュータメモリリソース内で高いアクティビティを呈し、計算及び通信リソース内で低いアクティビティを呈する。同様に、計算フェーズ 2 1 4 及び通信フェーズ 2 1 6 は、それぞれ、偏った計算及び通信アクティビティを呈する。図 2 A に示す例のように、シーケンス 2 1 0 は、多くの反復で繰り返すことができる。他の実施形態では、記憶フェーズ 2 1 2、計算フェーズ 2 1 4、通信フェーズ 2 1 6、記憶 / 計算 / 通信フェーズの混合フェーズ及び / 又は他のフェーズは、同じシーケンス内で又は予測可能なシーケンス内で起きるとは限らない。そのような実施形態では、電力管理装置 1 1 5 は、依然として、電力パジェットに対応する時間間隔内での各フェーズの発生を認識し、シーケンス 2 1 0 内の時間的電力分配を適用する。

10

【 0 0 3 4 】

(限定ではなく) 例として、電力管理装置 1 1 5 は、第 1 記憶フェーズ 2 1 2 が、コンピュータシステム 1 0 5 の処理装置 1 1 0 とメモリ装置 1 3 0 との間の相互作用によって支配されるアプリケーション挙動と関連し、かつ、第 1 計算フェーズ 2 1 4 は、(例えば、メモリアクセス及び / 又は他の装置との通信ではなく) 一つ以上のコア 1 1 2 での計算によって支配されるアプリケーション挙動と関連することを決定することができる。システムの電力上限が 1 0 0 W であり、これは、ある期間 t にわたってコンピュータシステム 1 0 5 によって使用される平均電力として評価されると仮定する。また、第 1 記憶フェーズ 2 1 2 の制約なしの使用電力 (すなわち、フェーズがフルのパフォーマンスで実行するのに必要な電力) は、1 0 0 W の電力上限よりも 1 0 W 低い 9 0 W であり、実行に 5 秒か

20

【 0 0 3 5 】

この例では、電力管理装置が、フルのパフォーマンスに達するためにどれだけの電力を必要とするかにかかわらず、電力管理装置が各フェーズに対して 1 0 0 W の電力を確保する場合、第 1 記憶フェーズ 2 1 2 の持続時間に対して 1 0 W の電力が無駄になる。さらに、以下の式によって示すように、第 1 記憶フェーズ 2 1 2 及び第 1 計算フェーズ 2 1 4 にわたる平均使用電力 (average power usage) は 9 6 . 6 7 W であり、これは 1 0 0 W

30

【 数 1 】

Average Power

$$= \frac{\text{PWR}(\text{Mem}) \times \text{Runtime}(\text{Mem}) + \text{PWR}(\text{Comp}) \times \text{Runtime}(\text{Comp})}{\text{Runtime}_{\text{Mem}} + \text{Runtime}_{\text{Comp}}}$$

$$\text{Average Power} = \frac{90\text{W} \times 5 \text{ secs} + 100\text{W} \times 10 \text{ secs}}{5 \text{ secs} + 10 \text{ secs}}$$

$$\text{Average Power} = \frac{1450\text{Joules}}{15\text{secs}} = 96.67\text{W} < \text{Power Cap} = 100\text{W}$$

40

【 0 0 3 6 】

上記の式において、 $\text{PWR}(\text{Mem})$ は、第 1 期間 $\text{Runtime}(\text{Mem})$ 中に第 1 記憶フェーズ 2 1 2 に割り当てられた電力であり、 $\text{PWR}(\text{Comp})$ は、第 2 期間 $\text{Runtime}(\text{Comp})$ 中に第 1 計算フェーズ 2 1 4 に割り当てられた電力である。この例における上記の式は、平均電力上限に等しい各フェーズについて確保する電力がどのようにして電力の準最適な使用につながり得るかを示す。

【 0 0 3 7 】

しかし、電力管理装置 1 1 5 が 1 0 0 W を超える第 1 計算フェーズ 2 1 4 を割り当てる場合、そのフェーズのパフォーマンスが改善し、アプリケーション全体のパフォーマンス

50

が改善される可能性がある。例えば、105.6Wの電力制限だと、第1計算フェーズ214は、100Wの電力制限での10秒と比較して、9秒の実行時間を有すると仮定する。電力管理装置115が計算制限フェーズ105.6Wを割り当てる場合、平均使用電力は、

【数2】

$$\text{Average Power} = \frac{90\text{W} \times 5 \text{ secs} + 105.6\text{W} \times 9 \text{ secs}}{5 \text{ secs} + 9 \text{ secs}}$$

$$\text{Average Power} = \frac{1400 \text{ Joules}}{14 \text{ secs}} = 100\text{W} \leq \text{Power Cap} = 100\text{W}$$

10

【0038】

上記の例に示すように、特定の動的時間的電力ステアリングの実施形態は、フェーズにわたる電力の準最適使用を特定し、システムに課された電力上限を一時的に違反することによってそれを訂正しながら、評価期間においては平均電力は、システムの電力上限に違反しないことを保証している。その結果、アプリケーション全体のパフォーマンスが改善される（例えば、上記の例では2つのフェーズにわたる実行時間が15秒から14秒に短縮されている）。

【0039】

特定の実施形態では、電力管理装置115は、フェーズの特性をモニタ及び学習する動的時間的電力スケジューラ（図示せず）を含む。動的時間的電力スケジューラは、各アプリケーションフェーズについて、どのようにパフォーマンスが電力にスケールするかを学習する。動的時間的電力スケジューラはまた、電力を制限することを通じて、パフォーマンス-電力のスケール（scaling）が悪い、アプリケーションの一つ以上のフェーズの速度を落とし、パフォーマンス-電力のスケールが良いフェーズに電力を転送することによって改善されたアプリケーションパフォーマンスが得られる場合を認識する。

20

【0040】

（限定ではなく）上記の例を続けると、80Wの電力制限だと、第1記憶フェーズ212はいくらかのパフォーマンスを失い、完了までに5秒ではなく6秒かかり、117.1Wの電力制限だと、第1計算フェーズ214は、100Wで10秒ではなく7秒で終了することができることを学習したと想定する。すると、平均電力（average power）は、

30

【数3】

$$\text{Average Power} = \frac{80\text{W} \times 6\text{secs} + 117.1\text{W} \times 7\text{secs}}{6\text{secs} + 7\text{secs}}$$

$$\text{Average Power} = \frac{1300 \text{ Joules}}{13 \text{ secs}} = 100\text{W} \leq \text{Power Cap} = 100\text{W}$$

【0041】

上記の例示的な電力割り当ては、アプリケーションによって使用される平均電力がシステム電力上限（100W）を超えないという要件を満たす。しかし、上記の例では、2つのフェーズにわたる実行時間が15秒から13秒に短縮されている。この例は、フェーズ実行時間がそのフェーズ中に割り当てられた電力量にどのように関連しているかを学習することにより、どのように動的時間的電力ステアリングが時間ドメイン（すなわち、フェーズにわたる）において全体的に最適又は改善された電力決定を行うことができるのかを示す。そのようにすることで、動的時間的電力スケジューラは、特定のフェーズの電力を制約するなどの直観的でない決定を特定することができ、全体の電力リミットなどの全体のシステム制約を遵守しながら、節約されたエネルギーがいくつかの目的関数（例えば、効率、パフォーマンス等）を改善するように使用されることができる。

40

50

【 0 0 4 2 】

第1記憶フェーズ212及び第1計算フェーズ214に関する上記の例は、例示として提供されているが、多くのアプリケーションが、作業の大部分がメモリコントローラにメモリ要求を送信している、長くメモリ支配的なフェーズを有することに留意されたい。メモリアクセスには時間がかかるため、アプリケーションはメモリアクセスが戻ってくるのを待つのにかなりの時間を費やす可能性がある。逆に、HPCやその他のシステムは電力が制約されているため、ベクトルユニットのように電力を消費するリソースを大量に使用する計算結合フェーズでは、電力の割り当てにおけるわずかな増加だけでも大きなメリットを得ることができる。このように、多くの場合、電力の割り当てにおけるわずかな減少によりパフォーマンス - 電力のスケールリングが悪いフェーズで失われるパフォーマンスの絶対値は、電力の割り当てにおけるわずかな増加によりパフォーマンス - 電力のスケールリングがよいフェーズで得られるパフォーマンスの絶対値よりも小さい。結果として、アプリケーション全体のパフォーマンスが向上し、電力がより効率的に使用され、そして、アプリケーションによって使用される平均電力がシステムの電力リミットを超えない。上記の説明は例としてパフォーマンスを使用するが、この実施形態は他の目的関数（例えば、効率、パフォーマンスなど）に広げることができる。

10

【 0 0 4 3 】

パフォーマンス - 電力スケールリングが悪いフェーズの電力を下げることは、いつまでも利益をもたらさない可能性があることにも留意されたい。例えば、ある地点の後では、記憶制限フェーズの電力を切ることは、もはやパフォーマンスにわずかな影響も与えない可能性がある。電力がある変曲点を過ぎて制限されていると、フェーズのパフォーマンスが急激に低下し、追加の電力を受ける他のフェーズへの利益にもかかわらずアプリケーション全体のパフォーマンスに実質的に害を及ぼす可能性がある。したがって、動的時間的電力スケジューラは、特定の実施形態によれば、アプリケーション全体のパフォーマンスが最大化又は改善されるように、フェーズ間でステアリングする（steer）電力量を見つけるように構成されている。

20

【 0 0 4 4 】

特定の実施形態では、電力管理装置115は、シーケンス210の現在フェーズに基づいて、処理装置110及び/又はコンピュータシステム105内の電力を空間的に再分配するようにさらに構成されている。第1記憶フェーズ212では、電力管理装置115は、一つ以上の処理コア112に関連づけられた電力ドメインが必要とする電力がより少なくてもよいと決定することができる。したがって、電力管理装置115は、記憶フェーズ212中は（例えば、エネルギーを節約するために）一つ以上の処理コア112に割り当てられた電力を下げて、及び/又は電力の少なくとも一部を（例えば、記憶フェーズ212に關与するリソースのパフォーマンスを向上させるために）他の電力ドメインに再分配する。

30

【 0 0 4 5 】

他の例として、図2Cは、アイドルフェーズ222とアクティブフェーズ224とが交互に繰り返されるシーケンス220を示す。電力管理装置115は、例えば、（強制又は非強制の）アイドル期間及びアクティブ期間を含む処理のデューティサイクルを制御するように構成することができる。いくつかの実施形態では、アイドル期間中、処理装置110は状態期間（state period）に置かれ、ハードウェア及びソフトウェア動作を能動的には処理しない。処理装置110は、一つ以上の処理コア112、グラフィックス処理ユニット114、I/O装置116、メモリ装置118及び他のアンコアコンポーネントを含め、装置の一部又は全部にわたってアイドル状態に置かれてもよい。他の実施形態では、処理装置110は、コンピュータシステム105を正しく機能させるために不可欠な動作のみを処理することができる。他の実施形態では、処理装置110は、スロットリング閾値（throttling threshold）に基づいて、一部ではあるが全てではないソフトウェア及びハードウェア動作を処理することができる。電力管理装置115は、特定のアプリケーションに対してアイドルフェーズ222とアクティブフェーズ224との間の最適又は

40

50

改善された電力分配を決定し、実行中、それに応じて電力を再分配するように構成されている。

【0046】

図3は、複数の電力ドメイン（ドメイン310a、310b、...、310nとして示されている）含むノード305と、電力供給システム312と、電力マネージャ314と、実行平均電力コントローラ316とを含む装置300のブロック図である。ノード305は、例えば、図1に示すコンピュータシステム105などのコンピュータシステムを含むことができる。このような実施形態では、各ドメイン310a、310b、...、310nは、別個の処理装置110を含むことができる。他の例では、ノード305は、サーバ若しくは、マイクロプロセッサ、マルチコアマイクロプロセッサの個々のコア、メモリコントローラハブ（MCH）、入出力コントローラハブ（IOH）、メモリ装置、ネットワークインタフェース等のコンピュータシステムコンポーネント、又は電力消費についてモニタされ、制御されることができる一つ以上の電力ドメインを有する任意の他のタイプの装置とすることができる。図示されている電力供給システム312は、ノード305にある範囲の電圧を供給することができるスイッチング又は線形電圧レギュレータ（VR、図示せず）を含むことができる。

10

【0047】

図示されている電力マネージャ314は、フェーズモジュール318及び動的時間的電力スケジューラ320を含み、第三者要素であっても、又は電力供給システム312に含まれてもよく、固定機能ハードウェア、マイクロコード、ファームウェア、ソフトウェア又はそれらの任意の組み合わせで実装されることができる。他の実施形態では、電力マネージャ314は、ノード305に含まれても、ノード305によって実行されてもよい。例えば、ノード305は、電力マネージャ314を実装するように構成された一つ以上の処理コアを含むことができる。フェーズモジュール318は、ノード305によって実行されるアプリケーション又は処理のフェーズを検出するように構成されている。特定の実施形態は、例えば、ランタイム開発者、ライブラリ開発者、アプリケーションプログラマ等のユーザがソフトウェアコードにアノテーションをつけて再コンパイルすることができるソフトウェアアプリケーションプログラムインタフェース（API）を含み、コードがフェーズ情報を電力マネージャ314に通信するようにする。フェーズモジュール318は、コードと共にコンパイルされたフェーズ情報を使用して、実行中にフェーズに入っている（being entered）が又は終了している（being exited）のかを決定する。APIは、コードに挿入されるオーバーヘッドを最小限に抑えるために軽量にすることができる。APIは、例えば、各フェーズのコードに一つ以上のAPI呼び出しを挿入することができる。

20

30

【0048】

APIの特定の実施形態では、共有ライブラリが使用されることができ、オペレーティングシステム（例えば、Linux（登録商標））は、アプリケーションと共有ライブラリとの間に計測レイヤを挿入することができる。ユーザが計測レイヤを書き込む必要があるかもしれないが、このような実施形態であれば、アプリケーション又は共有ライブラリの修正又は再コンパイルを必要としない。さらに、ライブラリはアプリケーションによって使用される標準APIを有するので、一つの計測レイヤがAPIを実装する任意の共有ライブラリをサポートすることができる。さらに、再リンクが可能な場合、この例示的な実施形態は、静的リンクライブラリも同様に使用することができる。

40

【0049】

追加的に、又は他の実施形態では、フェーズモジュール318は、フェーズを自動的に推定するモードをサポートするように構成されている。そのような実施形態は、アプリケーションを計測しない（instrument）。むしろ、装置300は、ハードウェアパフォーマンスモニタ能力を介してアプリケーションアクティビティを測定する。例えば、装置300は、ハードウェアパフォーマンスカウンタの痕跡（trace）をサンプリングし、分析する。実行時間では、フェーズモジュール318は、痕跡に時系列信号処理分析を適用して

50

、アプリケーション実行の痕跡を安定した挙動のセグメントに分割する。次いで、フェーズモジュール318は、クラスタ化技術を適用して、反復セグメントをアプリケーションフェーズとして認識する。例えば、フェーズモジュール318は、再帰最小二乗を伴うARRIMAX自己回帰を使用して、次の時間ステップにおいてパフォーマンスカウンタの痕跡が何であるかを学習し予測することができる。予測が間違っている場合、フェーズモジュール318は、アプリケーションがフェーズを変更している可能性があるとして決定する。フェーズモジュール318は、安定化ループを適用して、偽の(mispredictions)誤予測をフィルタリングする。しかし、誤予測が偽でない場合、フェーズモジュール318は、フェーズが変化した時点特定する。

【0050】

特定の実施形態では、フェーズモジュール318は、アプリケーションがフェーズに入っている、又はフェーズを出ているのかを判断するステートマシンを含む。各フェーズについて、フェーズモジュール318は、アプリケーションアクティビティシグネチャを追跡する。シグネチャは、例えば、モニタされているパフォーマンスカウンタアクティビティから導出された、計算、通信又は記憶の強度を含む特性のベクトルである。どのフェーズが現在実行されているかを決定するために、フェーズモジュール318は、アクティビティシグネチャを使用し、空間クラスタリング技術を適用する。フェーズモジュール318は、可能性のあるシグネチャの空間を等しいサイズの領域に分割し、各領域に、対応するフェーズ番号を与える。限定ではなく例として、100以上の異なる領域/フェーズがあることができる。アプリケーションがフェーズに入ると、フェーズモジュール318は、フェーズのシグネチャが空間のどの領域内に含まれるか調べ、そのフェーズを、その領域に対して指定された番号又は識別子でラベル付けする。動的時間的電力スケジューラ320は、フェーズ番号又は識別子を受け、各フェーズに対する最適又は改善された電力割り当てを決定する。

【0051】

動的時間的電力スケジューラ320は、どのフェーズ間で電力をステアリングするかを決定するように構成される。一実施形態は、上記に参照されたソフトウェアAPIを使用する。このAPIにより、開発者などのユーザは、省電力や加速のためにフェーズは最適化されるべきかどうかをアノテーションを付けることができる。動的時間的電力スケジューラ320は、省電力のために指定されたフェーズからどれだけの電力を取り除くべきか、及び最大又は向上したパフォーマンスのため、加速のために指定されたフェーズにどれだけの電力を加えるべきかを学習する。例えば、動的時間的電力スケジューラ320は、すべての可能性にわたる徹底的な探索を使用する、又は、勾配降下法若しくは機械学習を介して知的に探索を削ることができる。探索は、各フェーズに割り当てられた電力が、電力上限によって示唆されるエネルギーバジェットと一致する(フェーズにわたって足された)合計エネルギーをもたらすという制限によって結合されている。言い換えると、電力上限は、フェーズよりも長い時間間隔にわたる平均電力であり、電力上限と時間間隔との積はエネルギーバジェットに等しい。

【0052】

制約を満たす割り当てを見つけるために、特定の実施形態によれば、動的時間的電力スケジューラ320は、(各フェーズについて)フェーズの実行時間が電力にどのようにスケールするかを測定し、この電力スケール情報をテーブルに記録する。割り当てを試すときは、電力スケール情報を備えたテーブルを参照して、不整合な(inconsistent)割り当てを除外する。他の実施形態では、動的時間的電力スケジューラ320は、フェーズの実行時間が電力にどのようにスケールするかを測定し、全体の制約と不整合な割り当てを早急に訂正する。

【0053】

特定の実施形態では、動的時間的電力スケジューラ320は、省電力のために指定されたフェーズと次の加速フェーズとの間にどのフェーズが入るかを知らるためにフェーズのシーケンスを予測する。装置300は、例えば、時間の経過とともにフェーズの痕跡を保持

10

20

30

40

50

することができる。特定の実施形態は、動的時間的電力スケジューラ320は、現在フェーズ及び最後にこのフェーズにいたアプリケーションでどのシーケンスが起きたかのみに基づいて次のフェーズが予測できるように、省電力フェーズと加速フェーズとの間のフェーズは常に同じであると想定している。多くのバルク同期HPCアプリケーションは、この想定に適合する。追加的に、又は他の実施形態では、機械学習方法を介して次のフェーズを予測することによって、分岐（又は、フェーズが省電力フェーズと加速フェーズとの間に入る可変性）が対応されることができる。

【0054】

他の実施形態は、API又はソフトウェアによって提供されるアノテーションからの情報を使用せずに、どのフェーズが電力を節約し、どのフェーズを加速すべきかを指定する。特定のような実施形態では、動的時間的電力スケジューラ320は、上述した自動フェーズ検出及びアクティビティシグネチャを使用する。シグネチャを介して、動的時間的電力スケジューラ320は、例えば、どのフェーズが記憶集中型であり、通信集中型であり、計算集中型であるかを決定する。動的時間的電力スケジューラ320は、例えば、電力が節約されるべきフェーズとして記憶集中型及び通信集中型のフェーズを指定し、加速されるべきフェーズとして計算集中型のフェーズを指定することができる。次に、上述したように、動的時間的電力スケジューラ320は、平均電力上限によって示唆されるエネルギーバジェット要件に適合させることによって制約されるフェーズ間での電力の最適又は改善された割り当てを探索する。

【0055】

追加的に、又は他の実施形態では、動的時間的電力スケジューラ320は、システム構成、実行時間イベント及び他の挙動変化の変更に適応する。例えば、アプリケーションの挙動が変化する（例えば、新しいアプリケーションが実行を開始する）、システム管理者が電力マネージャ314の所望の目的関数を変更し得る（例えば、エネルギー効率から最大パフォーマンスへの切り替え）、及び/又は基本的なシステム制約が変化し得る（例えば、電力上限又は利用可能な総電力）ということが起き得る。動的時間的電力スケジューラ320は、ノード305がアプリケーションを実行するときにフェーズ間の電力の最適な又は改善された割り当てを再度、決定又は学習することによって、実行時間における動作挙動の変化に適応するように構成されている。動的時間的電力スケジューラ320は、例えば、ユーザ入力、定期的な所定の時間間隔によって指示された場合に、及び/又はシステム内の変化の検出に応じて、再学習手順が自動的に起動されるモードにおいて、再学習手順を行うことができる。

【0056】

実行平均電力コントローラ316は、電力モタモジュール322と、ドメインパフォーマンス制御モジュール324と、平均電力エンフォーサモジュール325と、ドメイン電力バジェットマネージャモジュール326とを含む。実行平均電力コントローラ316は、別個の装置又はモジュールとして示されているが、実行平均電力コントローラ316は、電力マネージャ314又はノード305の一部である、又はそれらによって実行されてよい。実行平均電力コントローラ316は、電力マネージャ314から受けた（例えば、ノード305の全体平均電力を示す）平均電力信号328及び時間的ステアリングインタフェース330に基づいて、ノード305の様々な電力ドメイン310a、310b、...、310n内の電力をモニタ及び制御するように構成されている。実装によるが、時間的ステアリングインタフェース330は、各フェーズに対する、各ドメイン又はドメインのグループの電力制限、ドメイン電力バジェットマネージャモジュール326に内部ヒューリスティックを通じた自発的な電力分配を決定させることを任せるヌル値、及び/又は各ドメインの相対的な重要性又は影響を示すもの（例えば、ドメイン電力バジェットマネージャモジュール326に相対的な重要度又は影響に基づいて電力分配を決定させることを任せる）を含むことができる。

【0057】

電力モタモジュール322は、様々な電力ドメイン310a、310b、...、310

10

20

30

40

50

nの電力消費を測定し、上述したように、対応する電力モニタデータ321をフェーズモジュール318及び/又は動的時間的電力スケジューラ320に供給するように構成されている。電力モニタモジュール322はまた、電力モニタデータを平均電力エンフォースモジュール325及びドメイン電力バジェットマネージャモジュール326に提供する。
【0058】

ドメインパフォーマンス制御モジュール324は、例えば、周波数、制限帯域(throttling bandwidth)、及び/又は他のパフォーマンスパラメータを調整することを通じて、各ドメインのパフォーマンス及び電力を調整する制御を含む。特定の実施形態では、パフォーマンスレベルは、多数の異なる方法で定義され、選択されることができる。一つのアプローチとしては、パフォーマンス状態又は、P0状態の装置が最大パフォーマンス能力を使用し、最大電力を消費することができ、P1状態の装置が、その最大値よりも低く制限されたパフォーマンス能力を使用し、最大電力未満を消費する等のPx状態を選択することである。平均電力エンフォースモジュール325は、ドメインパフォーマンス制御モジュール324を設定し、プログラブル時間ウィンドウにわたってノード305についての平均電力信号328によって示される全体平均電力を維持するようにその設定を時間とともに適応させる。平均電力エンフォースモジュール325は、標準コントローラ技術で実装され得る制御システムを含む。ドメイン電力バジェットマネージャモジュール326は、電力バジェットを平均電力エンフォースモジュール325に出力するロジックを含む。ドメイン電力バジェットマネージャモジュール326は、動的時間的電力スケジューラ320にいくつかの異なるインタフェースを提供することができ、それにより、動的時間的電力スケジューラ320が、各ドメインに対する特定の電力バジェット(例えば、ワットに比例する単位)を直接的に指定する、又は各ドメインの相対的重要度又は影響に関するヒントを(抽象単位で)指定することによって、ドメイン電力バジェットマネージャモジュール326が内部ヒューリスティックをヒントに適用し、それらをドメインについての特定のバジェットにマッピングすることによりバジェットを間接的に指定することができる。

【0059】

図4は、一実施形態による動的時間的電力ステアリングのための方法400のフローチャートである。方法400は、アプリケーションのフェーズのシーケンスを決定するステップ410を含み、複数のフェーズの各々は、他のフェーズの動作挙動とは異なる動作挙動を呈するアプリケーションのセグメントを含む。このシーケンスは、エネルギーバジェットに関連する時間間隔に対応する。方法400はまた、パフォーマンス指標、目的関数、及び測定間隔及び/又はイベント(例えば、フェーズ変化又は反復の終了)を選択するステップ412を含む。パフォーマンスには様々な定義が使用されることができ、パフォーマンスは様々な異なる粒度で測定されることができ、これにより、パフォーマンスは、リタイアされる命令のレート、1秒当たりの浮動小数点演算(FLOPS)、進行度、実行時間等の指標によって定義されることができ、このようなレート、実行時間又は他のパフォーマンス指標は、様々な粒度で計算されることができ、例えば、選択されたパフォーマンス指標は、各フェーズ内の短い時間間隔で測定される、各フェーズの終了などのイベントがあったら測定される、フェーズよりも長い時間間隔で測定される、(反復アプリケーションに対しては)反復の終了などのイベントがあったら測定される等とすることができる。方法400は、さらに、現在フェーズを検出するためにアプリケーションをモニタするステップ414を含む。

【0060】

特定の実施形態では、方法400は、さらに、選択されたパフォーマンス指標ごとに最適な時間的電力割り当てが達成されたかどうかを問い合わせるステップ418を含む。言い換えると、方法400は、フェーズ間での準最適な時間的割り当てで一つ以上の反復を実行し、各フェーズ中及び複数のフェーズ境界にわたってパフォーマンスを測定して、改善又は最適な割り当てを探索することができる。最適な時間的電力割り当てが達成されたかどうかを決定するために、いくつかの方法が使用されることができ、例えば、方法4

10

20

30

40

50

00は、多数又はすべての順列（すなわち、可能な電力割り当て）を試し、結果としての測定されたパフォーマンスを比較して、最適分配を決定することができる。他の例として、方法400は、勾配探索法（すなわち、電力割り当て変数に関してパフォーマンス指標の勾配を推定し、勾配方向にステップし、勾配経路に沿って順列を試みるだけである）を用いることによって最適解の近似を求めることができる。さらに他の例では、順列空間が大きい場合、順列空間を徹底的に探索する代わりに、方法400は、様々なサンプリング方法を適用し、試行された割り振りの中で最適の割り当てを採用することができる。

【0061】

測定が、フェーズのシーケンスにわたる最適な時間的電力割り当てが達成されていない（又は、まだ改善されることができると決定する場合、方法400はシーケンスのフェーズ間で電力を時間的に再割り当てするステップ420を含む。しかし、測定が、フェーズのシーケンスにわたる最適な時間的電力割り当てが達成された決定する場合、方法400は、現在フェーズに対応する時間的電力割り当てを複数の電力ドメインに適用するステップ421を含む。特定の実施形態では、例えば、図3に示す電力マネージャ314又はドメイン電力バジェットマネージャ326は、現在フェーズに割り当てられた電力リミットが複数の電力ドメインにわたってどのように空間的に分配されるかを決定するように構成することができる。他の実施形態では、図5に関して以下に説明するように、空間電力ステアリングアルゴリズムが使用されて、（時間電力ステアリングによって決定されるような）現在フェーズに対する特定の電力リミットについて複数の電力ドメインにわたる空間電力割り当てを最適化することができる。

【0062】

方法400はまた、システム変更、アプリケーション変更、電力上限変更又は選択されたパフォーマンス指標変更があるかを問い合わせるステップ422を含む。ない場合は、方法400は、現在の（又は次の）フェーズを検出するために、そして、フェーズのシーケンスにわたる電力の時間的割り当てを最適化することを続けるために、アプリケーションをモニタするステップ414を続ける。一方、システム変更、アプリケーション変更、電力上限変更又は選択されたパフォーマンス指標変更があると決定される場合、方法400は、現在フェーズを検出するためにアプリケーションをモニタするステップ414の前に、フェーズのシーケンスを決定するステップ410及び/又はパフォーマンス指標、目的関数及び測定間隔/イベントを選択するステップ412を再度行い、最適な時間的電力割り当てが達成されたかどうかを問い合わせるステップ418を再度行うということを繰り返す。

【0063】

上述したように、特定の実施形態は、フェーズ境界にわたる動的時間的電力ステアリングと、各フェーズに対する電力ドメイン間の動的空間電力ステアリングとの組み合わせを含む。特定のこのような実施形態では、動的空間電力ステアリングは、動的時間的電力ステアリングのための方法400内で入れ子にされてもよい。例えば、図5は、図4に示すような、複数のフェーズ間で電力を時間的に再割り当てするステップ420のためのフローチャートであり、一実施形態による動的空間電力ステアリングも含む。選択されたパフォーマンス指標毎に最適な時間的電力割り当てが達成されたかどうかを問い合わせるステップ418を行い、フェーズのシーケンスにわたる最適な時間的電力割り当てが達成されていない（又は、まだ改善されることができると決定した後、図5は、シーケンスのフェーズ間で電力を時間的に再割り当てするステップ420は、シーケンスのフェーズ間で新しい時間的電力割り当てを選択するステップ510と、新しい時間的電力分配に基づいて現在フェーズが省電力、加速、又はどちらもないに指定されているかどうかを問い合わせるステップ512と含み得ることを示す。特定の実施形態では、この問い合わせは、現在フェーズの実行時間が電力変動どのようにスケールリングするか（又は、どのようにスケールリングするかについての前の決定を洗練する）を決定することを含むことができる。

【0064】

現在フェーズが省電力のために指定されている場合、この方法は、現在フェーズ中は、

10

20

30

40

50

新しい時間的電力割り当てによりノード内の複数の電力ドメイン全体の電力リミットを下げるステップ514を含む。現在フェーズが加速のために指定されている場合、この方法は、現在フェーズ中は、新しい時間的電力割り当てによりノード内の複数の電力ドメイン全体の電力リミットを上げるステップ516を含む。現在フェーズが省電力にも加速のためにも指定されていない場合、この方法は、エネルギーバジェットにより現在フェーズの全体の電力リミットを維持する。

【0065】

図5は、さらに、選択されたパフォーマンス指標ごとに、複数の電力ドメイン間の最適な空間電力分配が現在フェーズに対して達成されたかどうかを問い合わせるステップ518を含む。測定が、最適な空間電力分配が達成されていない（又は、まだ改善されることができると決定する場合、この方法は、現在フェーズに対して、新しい時間的電力割り当て（すなわち、フェーズ間の新しい時間的電力割り当てに基づいた現在フェーズに対する新しい電力リミット）に基づいて、複数の電力ドメインの間で電力を空間的に再分配するステップ520を含む。この方法は、再分配された電力が複数の電力ドメイン間で最適な空間電力分配を提供するかどうかを問い合わせるステップ518を再度行う。測定が、最適又は十分な空間電力分配が達成されたと決定する場合、この方法は、現在フェーズに対応する空間電力分配を複数の電力ドメインに適用するステップ522を含む。次いで、この方法は、図4に示すように、すなわち、システム変更、アプリケーション変更、電力上限変更又は選択されたパフォーマンス指標変更があるかどうかを問い合わせるステップ422を行うことによって、続く。

【0066】

以下はさらなる実施形態の例である。例は、方法、方法の動作を実行する手段、機械によって実行されると、その機械に方法、又は、本明細書に記載されている実施形態及び例による拡張ビデオをレンダリングするための機械若しくはシステムの動作を実行させる命令を含む少なくとも一つの機械読み取り可能媒体等の主題を含むことができる。

【0067】

例1は、電力マネージャを使用して、ノード上で実行するためのタスクを含むアプリケーションに対応する、フェーズのシーケンスを決定するステップであって、該シーケンスはエネルギーバジェットに関連する時間間隔に対応する、ステップと、各フェーズに対して、前記ノード内の複数の電力ドメインに分配される電力の増加又は減少のため、実行時間において測定された応答を含む電力スケールリングを決定するステップと、各フェーズに対する前記電力スケールリングに基づいて、前記シーケンスのフェーズ間での時間的電力分配を決定して、前記エネルギーバジェットを満たし、前記ノードによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するステップと、を含む方法である。

【0068】

例2は、パフォーマンス指標、目的関数、及び前記パフォーマンス指標を評価するための測定間隔及び測定イベントのうちの一つを選択するステップをさらに含み、前記パフォーマンス指標は、前記ノードによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するように選択され、前記目的関数は、電力上限内でのパフォーマンスを最大化すること、パフォーマンスを最大化すること、電力上限内でエネルギー効率を最大化すること、及び効率を最大化することを含む群から選択される、例1に記載の主題である。

【0069】

例3は、現在フェーズを検出するために前記アプリケーションをモニタするステップと、前記現在フェーズにおいて、前記時間的電力割り当てが前記目的関数を満たさないと決定するステップと、前記決定に応じて、前記シーケンスのフェーズ間で前記電力を時間的に再割り当てするステップと、をさらに含む、例2に記載の主題である。

【0070】

例4は、前記シーケンスのフェーズ間で前記電力を時間的に再割り当てするステップは、前記シーケンスのフェーズ間で新しい時間的電力割り当てを選択するステップと、前記

10

20

30

40

50

時間的電力分配に基づいて、前記現在フェーズが省電力又は加速のために指定されているかどうかを決定するステップと、前記現在フェーズが省電力のために指定されている場合、前記新しい時間的電力割り当てに従って前記ノード内の複数の電力ドメインに対する全体の電力リミットを下げるステップと、前記現在フェーズが加速のために指定されている場合、前記新しい時間的電力分配に従って前記ノード内の複数の電力ドメインの全体の電力リミットを上げるステップと、を含む、例 3 に記載の主題である。

【0071】

例 5 は、前記現在フェーズが省電力又は加速のために指定されているかどうかを示す、前記アプリケーションのコンピュータ実行可能コード内のアノテーションを検出するステップをさらに含む、例 4 に記載の主題である。

10

【0072】

例 6 は、前記ノード内の複数の電力ドメイン間の空間電力分配が前記パフォーマンス指標を満たさないと決定するステップと、前記決定に応じて、前記新しい時間的電力割り当てに基づいて、前記現在フェーズにおける前記複数の電力ドメイン間の前記電力を空間的に再分配するステップと、をさらに含む、例 4 に記載の主題である。

【0073】

例 7 は、前記アプリケーション、前記ノード、電力上限及び前記ノードを含むシステムのうちの少なくとも一つの変化を検出するステップと、検出された前記変化に応じて、前記シーケンスのフェーズ間の時間的電力分配を再構成するステップと、をさらに含む、例 1 から 6 のいずれか一つに記載の主題である。

20

【0074】

例 8 は、計算結合フェーズ、記憶結合フェーズ、通信結合フェーズ、アイドルフェーズ、アクティブフェーズ及びそれらのフェーズの任意のものの組み合わせのうち少なくとも 2 つを検出するステップをさらに含む、例 1 から 7 のいずれか一つに記載の主題である。

【0075】

例 9 は、前記シーケンスの各フェーズに対して、前記シーケンスの第 1 反復に基づいた前記時間的電力分配を決定するステップと、前記シーケンスの各フェーズに対して、前記シーケンスの第 2 反復に前記時間的電力割り当てを適用するステップと、をさらに含む、例 1 から 8 のいずれか一つに記載の主題である。

【0076】

30

例 10 は、少なくとも一部にハードウェアを含むロジックを含む電力マネージャであって、該電力マネージャは、アプリケーションのフェーズのシーケンスを決定するフェーズモジュールであって、該シーケンスはエネルギーバジェットに関連する時間間隔に対応し、各フェーズは、他のフェーズの動作挙動とは異なる動作挙動を呈するアプリケーションのセグメントを有する、フェーズモジュールと、前記シーケンスのフェーズ間の時間的電力分配を決定して、前記エネルギーバジェットを満たし、前記アプリケーションの実行時間パフォーマンスを改善又は最適化する動的時間的電力スケジューラモジュールと、を含む装置である。

【0077】

例 11 は、前記電力マネージャは、パフォーマンス指標、目的関数、及び前記パフォーマンス指標を評価するための測定間隔及び測定イベントのうちの一つを選択するように構成され、前記目的関数は、前記アプリケーションの実行時間パフォーマンスを改善又は最適化するように選択される、例 10 に記載の主題である。

40

【0078】

例 12 は、前記フェーズモジュールは、さらに、現在フェーズを検出するために前記アプリケーションをモニタするように構成され、前記動的時間的電力スケジューラモジュールは、さらに、前記現在フェーズにおいて、前記時間的電力割り当てが前記目的関数を満たさないと決定し、前記決定に応じて、前記シーケンスのフェーズ間で電力を時間的に再割り当てするように構成されている、例 11 に記載の主題である。

【0079】

50

例 1 3 は、前記電力を時間的に再割り当てするために、前記動的時間的電力スケジューラモジュールは、さらに、前記シーケンスのフェーズ間で新しい時間的電力割り当てを選択し、前記時間的電力分配に基づいて、前記現在フェーズが省電力又は加速のために指定されているかどうかを決定し、前記現在フェーズが省電力のために指定されている場合、前記新しい時間的電力割り当てに従って、ノード内の複数の電力ドメインの全体の電力リミットを下げ、前記現在フェーズが加速のために指定されている場合、前記新しい時間的電力分配に従って前記ノード内の複数の電力ドメインの全体の電力リミットを上げる、ように構成されている、例 1 2 に記載の主題である。

【 0 0 8 0 】

例 1 4 は、前記動的時間的電力スケジューラモジュールは、さらに、前記現在フェーズが省電力又は加速のために指定されているかどうかを示す、前記アプリケーションのコンピュータ実行可能コード内のアノテーションを検出するように構成されている、例 1 3 に記載の主題である。

【 0 0 8 1 】

例 1 5 は、前記電力マネージャは、さらに、前記ノード内の複数の電力ドメイン間の空間電力分配が前記目的関数を満たさないことを決定し、前記決定に応じて、前記新しい時間的電力割り当てに基づいて、前記現在フェーズにおける前記複数の電力ドメイン間の前記電力を空間的に再分配する、ように構成されている例 1 3 に記載の主題である。

【 0 0 8 2 】

例 1 6 は、前記電力マネージャは、さらに、前記アプリケーション、前記ノード、電力上限及び前記装置を含むシステムのうちの少なくとも一つの変化を検出し、検出された前記変化に応じて、前記シーケンスのフェーズ間の時間的電力分配を再構成する、ように構成されている、例 1 0 から 1 5 のいずれか一つに記載の主題である。

【 0 0 8 3 】

例 1 7 は、システムであって、少なくとも一つのコアドメインと、少なくとも一つのアノードドメインとを含む複数の電力ドメインと、一つ以上のプロセッサによって実行されると、該一つ以上のプロセッサに動作を実行させる命令を記憶した少なくとも一つのコンピュータ読み取り可能記憶媒体と、を含み、該動作は、当該システムによって実行されるアプリケーションのフェーズのシーケンスを決定するステップであって、該シーケンスは、エネルギーバジェットに関連する時間間隔に対応する、ステップと、各フェーズに対して、当該システム内の前記複数の電力ドメインに分配される電力の増加又は減少のため、実行時間において測定された応答を含む電力スケーリングを決定するステップと、各フェーズに対する前記電力スケーリングに基づいて、前記シーケンスのフェーズ間での時間的電力分配を決定して、前記エネルギーバジェットを満たし、当該システムによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するステップと、を含む、システムである。

【 0 0 8 4 】

例 1 8 は、前記動作は、パフォーマンス指標、目的関数、及び前記パフォーマンス指標を評価するための測定間隔及び測定イベントのうちの一つを選択するステップをさらに含み、前記パフォーマンス指標は、当該システムによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するように選択される、例 1 7 に記載の主題である。

【 0 0 8 5 】

例 1 9 は、前記動作は、現在フェーズを検出するために前記アプリケーションをモニタするステップと、前記現在フェーズにおいて、前記時間的電力割り当てが前記目的関数を満たさないことを決定するステップと、前記決定に応じて、前記シーケンスのフェーズ間で前記電力を時間的に再割り当てするステップと、をさらに含む、例 1 8 に記載の主題である。

【 0 0 8 6 】

例 2 0 は、前記シーケンスのフェーズ間で前記電力を時間的に再割り当てするために、

10

20

30

40

50

前記動作は、前記シーケンスのフェーズ間で新しい時間的電力割り当てを選択するステップと、前記時間的電力分配に基づいて、前記現在フェーズが省電力又は加速のために指定されているかどうかを決定するステップと、前記現在フェーズが省電力のために指定されている場合、前記新しい時間的電力割り当てに従って当該システム内の複数の電力ドメインに対する全体の電力リミットを下げるステップと、前記現在フェーズが加速のために指定されている場合、前記新しい時間的電力分配に従って前記ノード内の複数の電力ドメインの全体の電力リミットを上げるステップと、をさらに含む、例 19 に記載の主題である。

【0087】

例 21 は、前記動作は、前記現在フェーズが省電力又は加速のために指定されているかどうかを示す、前記アプリケーションのコンピュータ実行可能コード内のアノテーションを検出するステップをさらに含む、例 20 に記載の主題である。

10

【0088】

例 22 は、前記動作は、前記ノード内の複数の電力ドメイン間の空間電力分配が前記パフォーマンス指標を満たさないと決定するステップと、前記決定に応じて、前記新しい時間的電力割り当てに基づいて、前記現在フェーズにおける前記複数の電力ドメイン間の前記電力を空間的に再分配するステップと、をさらに含む、例 20 に記載の主題である。

【0089】

例 23 は、前記フェーズのシーケンスを決定するために、前記動作は、前記アプリケーション、前記ノード、電力上限及び前記ノードを含むシステムのうちの少なくとも一つの変化を検出するステップと、検出された前記変化に応じて、前記シーケンスのフェーズ間の時間的電力分配を再構成するステップと、をさらに含む、例 17 から 22 のいずれか一つに記載の主題である。

20

【0090】

例 24 は、前記フェーズのシーケンスを決定するために、前記動作は、計算結合フェーズ、記憶結合フェーズ、通信結合フェーズ、アイドルフェーズ、アクティブフェーズ及びそれらのフェーズの任意のものの組み合わせのうち少なくとも 2 つを検出するステップをさらに含む、例 17 から 23 のいずれか一つに記載の主題である。

【0091】

例 25 は、前記動作は、前記シーケンスの各フェーズに対して、前記シーケンスの第 1 反復に基づいた前記時間的電力分配を決定するステップと、前記シーケンスの各フェーズに対して、前記シーケンスの第 2 反復に前記時間的電力割り当てを適用するステップと、をさらに含む、例 17 から 24 のいずれか一つに記載のシステムである。

30

【0092】

例 26 は、例 1 から 9 のいずれか一つに記載の方法を実装する又は装置を実現する機械読み取り可能命令を含む機械読み取り可能ストレージである。

【0093】

例 27 は、電力マネージャを使用して、ノード上で実行するためのタスクを含むアプリケーションに対応する、フェーズのシーケンスを決定する手段であって、該シーケンスはエネルギーバジェットに関連する時間間隔に対応する、手段と、各フェーズに対して、前記ノード内の前記複数の電力ドメインに分配される電力の増加又は減少のため、実行時間において測定された応答を含む電力スケーリングを決定する手段と、各フェーズに対する前記電力スケーリングに基づいて、前記シーケンスのフェーズ間での時間的電力分配を決定して、前記エネルギーバジェットを満たし、前記ノードによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化する手段と、を含む装置である。

40

【0094】

例 28 は、パフォーマンス指標、目的関数、及び前記パフォーマンス指標を評価するための測定間隔及び測定イベントのうちの一つを選択する手段をさらに含み、前記パフォーマンス指標は、前記ノードによる前記アプリケーションの実行時間パフォーマンスを改善又は最適化するように選択される、前記目的関数は、電力上限内でのパフォー

50

マンスを最大化すること、パフォーマンスを最大化すること、電力上限内でエネルギー効率を最大にすること、及び効率を最大にすることを含む群から選択される、例 27 に記載の主題である。

【0095】

例 29 は、現在フェーズを検出するために前記アプリケーションをモニタする手段と、前記現在フェーズにおいて、前記時間的電力割り当てが前記目的関数を満たさないことと決定する手段と、前記決定に応じて、前記シーケンスのフェーズ間で前記電力を時間的に再割り当てする手段と、をさらに含む、例 28 に記載の主題である。

【0096】

例 30 は、前記シーケンスのフェーズ間で前記電力を時間的に再割り当てする手段は、前記シーケンスのフェーズ間で新しい時間的電力割り当てを選択する手段と、前記時間的電力分配に基づいて、前記現在フェーズが省電力又は加速のために指定されているかどうかを決定する手段と、前記現在フェーズが省電力のために指定されている場合、前記新しい時間的電力割り当てに従って前記ノード内の複数の電力ドメインに対する全体の電力リミットを下げる手段と、前記現在フェーズが加速のために指定されている場合、前記新しい時間的電力分配に従って前記ノード内の複数の電力ドメインの全体の電力リミットを上げる手段と、を含む、例 29 に記載の主題である。

10

【0097】

例 31 は、前記現在フェーズが省電力又は加速のために指定されているかどうかを示す、前記アプリケーションのコンピュータ実行可能コード内のアノテーションを検出する手段をさらに含む、例 30 に記載の主題である。

20

【0098】

例 32 は、前記ノード内の複数の電力ドメイン間の空間電力分配が前記パフォーマンス指標を満たさないことと決定する手段と、前記決定に応じて、前記新しい時間的電力割り当てに基づいて、前記現在フェーズにおける前記複数の電力ドメイン間の前記電力を空間的に再分配する手段と、をさらに含む、例 30 に記載の主題である。

【0099】

例 33 は、前記アプリケーション、前記ノード、電力上限及び前記ノードを含むシステムのうちの少なくとも一つの変化を検出する手段と、検出された前記変化に応じて、前記シーケンスのフェーズ間の時間的電力分配を再構成する手段と、をさらに含む、例 27 から 32 のいずれか一つに記載の主題である。

30

【0100】

例 34 は、計算結合フェーズ、記憶結合フェーズ、通信結合フェーズ、アイドルフェーズ、アクティブフェーズ及びそれらのフェーズの任意のものの組み合わせのうち少なくとも二つを検出する手段をさらに含む、例 27 から 33 のいずれか一つに記載の主題である。

【0101】

例 35 は、前記シーケンスの各フェーズに対して、前記シーケンスの第 1 反復に基づいた前記時間的電力分配を決定する手段と、前記シーケンスの各フェーズに対して、前記シーケンスの第 2 反復に前記時間的電力割り当てを適用する手段と、をさらに含む、例 27 から 34 のいずれか一つに記載の主題である。

40

【0102】

上記の説明は、本明細書に説明されている実施形態の完全な理解のための多数の特定の詳細を提供する。しかし、当業者であれば、特定の詳細の一つ以上は省略されてもよく、他の方法、コンポーネント、又は材料が使用されてもよいことを認識するであろう。場合によっては、周知の特徴、構造、又は操作は、詳細には示されておらず、記載されていない。

【0103】

さらに、説明されている特徴、動作又は特性は、広範囲の異なる構成でアレンジ及び設計されることができ、及び / 又は一つ以上の実施形態において任意の適切な方法で組み合

50

わせられることができる。したがって、システム及び方法の発明を実施するための形態は、特許請求の範囲に記載されている本開示の範囲を限定するものではなく、本開示の可能な実施形態の単なる代表例である。さらに、開示された実施形態に関連して説明されている方法のステップ又は行動の順序は、当業者には明らかであるように変更され得ることは容易に理解されるものである。したがって、図面又は発明を実施するための形態の順序は、説明を目的とするのみであり、順序を必要とすると指定されない限り、順序が必要とされることを示唆することを意図していない。

【0104】

用語「結合された」は、本明細書では、問題となっているコンポーネント間の直接的又は間接的な任意のタイプの関係を示すために使用され、電氣的、機械的、流体的、光学的、電磁的、電気機械的、その他の接続に適用されることができる。追加的に、用語「第1」、「第2」等は、本明細書では説明を容易にするためにのみ使用されることができ、特に示されない限り、なんら特定の時間的又は時系列的な意味はない。

10

【0105】

本明細書において、「一実施形態」、「実施形態」、「例示的实施形態」等の言及は、実施形態に関連して説明される特定の特徵、構造又は特性が、少なくとも一つの実施形態に含まれることを意味している。本明細書の様々な場所においてそのような語句が出現するが、必ずしも全てが同じ実施形態を示しているとは限らない。さらに、特定の特徵、構造又は特性が任意の実施形態に関連して記載されている場合、他の実施形態に関連したそのような特徴、構造、又は特性に影響を及ぼすことは、当業者の理解の範囲内である。

20

【0106】

様々な実施形態は、ハードウェア要素、ソフトウェア要素及び/又はその両方の組み合わせを使用して実装されることができる。ハードウェア要素の例は、プロセッサ、マイクロプロセッサ、回路、回路要素(例えば、トランジスタ、抵抗器、コンデンサ、インダクタ等)、集積回路、特定用途向け集積回路(ASIC)、プログラマブル論理デバイス(PLD)、デジタル信号プロセッサ(DSP)、フィールドプログラマブルゲートアレイ(FPGA)、論理ゲート、レジスタ、半導体デバイス、チップ、マイクロチップ、チップセット等を含む。ソフトウェアの例は、ソフトウェアコンポーネント、プログラム、アプリケーション、コンピュータプログラム、アプリケーションプログラム、システムプログラム、機械プログラム、オペレーティングシステムソフトウェア、ミドルウェア、ファームウェア、ソフトウェアモジュール、ルーチン、サブルーチン、機能、方法、手順、ソフトウェアインタフェース、アプリケーションプログラムインタフェース(API)、命令セット、計算コード、コンピュータコード、コードセグメント、コンピュータコードセグメント、単語、値、記号、又はそれらの任意の組み合わせを含む。

30

【0107】

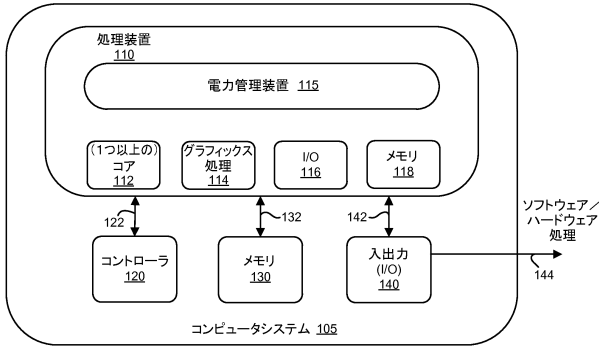
少なくとも一つの実施形態の一つ以上の態様は、プロセッサ内に様々なロジックを呈し、機械によって読み取られると、機械に本明細書で説明されている技術を実行するためのロジックを組み上げさせる機械読み取り可能媒体に格納された代表的な命令によって実装されることができる。「IPコア」として知られるそのような表現は、有形の機械読み取り可能媒体に記憶されることができ、様々な顧客又は製造施設に供給され、実際にロジック又はプロセッサを製造する製造機械にロードされる。

40

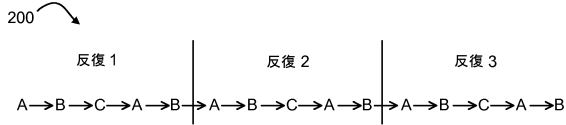
【0108】

多くの例示的な実施形態を参照して実施形態を説明したが、当業者によって、本開示の原理の精神及び範囲内に入る多くの他の修正及び実施形態が考案されることができることを理解されたい。より詳細には、本開示、図面、及び添付の特許請求の範囲の範囲内の主題の組み合わせアレンジメントのコンポーネントの部分及び/又はアレンジメントにおいて、様々な変更及び修正が可能である。コンポーネントの部品及び/又はアレンジメントの変更及び修正に加えて、代替の使用もまた当業者には明らかであろう。したがって、本発明の範囲は、添付の特許請求の範囲によってのみ決定されるべきである。

【図1】



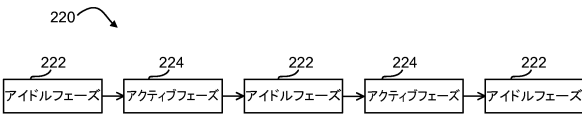
【図2A】



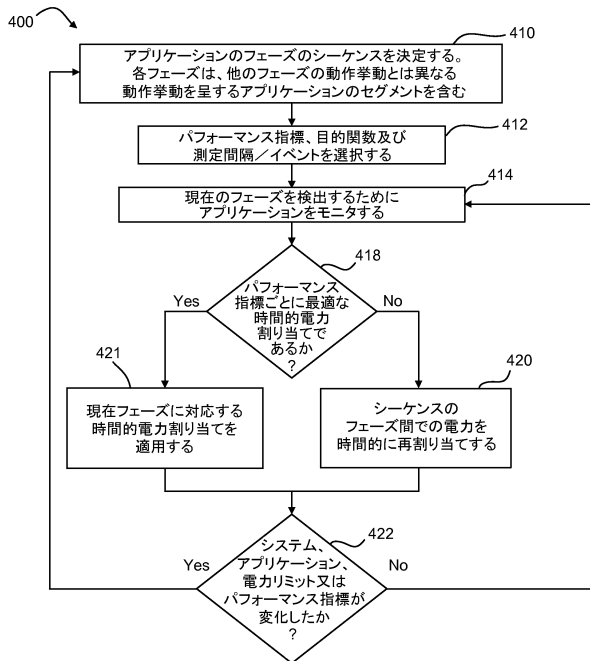
【図2B】



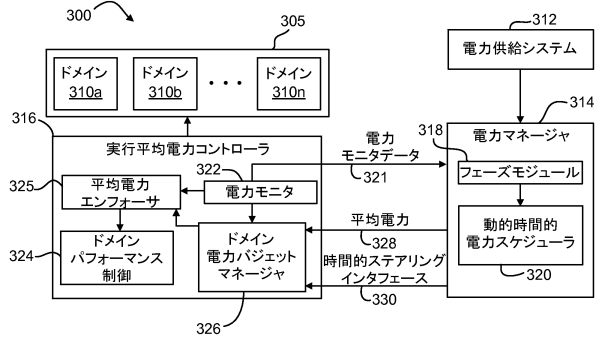
【図2C】



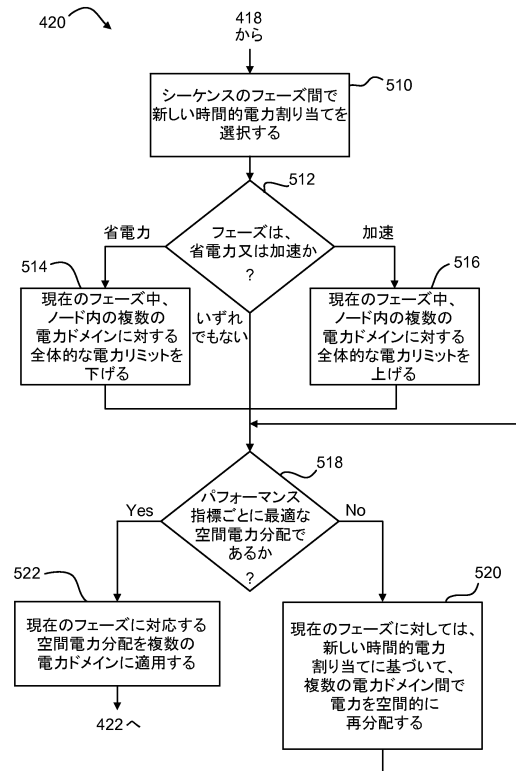
【図4】



【図3】



【図5】



フロントページの続き

- (72)発明者 イーステップ, ジョナサン, エム.
アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブール
バード 2200 インテル コーポレイション 内
- (72)発明者 バネルジー, ロヒット
アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブール
バード 2200 インテル コーポレイション 内
- (72)発明者 グレコ, リチャード, ジェイ.
アメリカ合衆国 95054 カリフォルニア州 サンタ クララ ミッション カレッジ ブール
バード 2200 インテル コーポレイション 内

審査官 田川 泰宏

- (56)参考文献 米国特許出願公開第2014/0089699(US, A1)
国際公開第2014/099741(WO, A1)
特表2013-536532(JP, A)
特開2011-100449(JP, A)
特表2014-527248(JP, A)
米国特許第05964879(US, A)
特開2009-181571(JP, A)
中国特許出願公開第102207769(CN, A)

(58)調査した分野(Int.Cl., DB名)

G06F 1/32 - 1/3296
H02J 1/00