

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6029660号
(P6029660)

(45) 発行日 平成28年11月24日(2016.11.24)

(24) 登録日 平成28年10月28日(2016.10.28)

(51) Int.Cl.

F 1

G06F 9/50 (2006.01)
G06F 9/46 (2006.01)G06F 9/46
G06F 9/46465 A
350

請求項の数 10 (全 16 頁)

(21) 出願番号 特願2014-514858 (P2014-514858)
 (86) (22) 出願日 平成24年6月7日(2012.6.7)
 (65) 公表番号 特表2014-516192 (P2014-516192A)
 (43) 公表日 平成26年7月7日(2014.7.7)
 (86) 国際出願番号 PCT/US2012/041434
 (87) 国際公開番号 WO2012/170746
 (87) 国際公開日 平成24年12月13日(2012.12.13)
 審査請求日 平成27年5月20日(2015.5.20)
 (31) 優先権主張番号 13/155,387
 (32) 優先日 平成23年6月8日(2011.6.8)
 (33) 優先権主張国 米国(US)

(73) 特許権者 314015767
 マイクロソフト テクノロジー ライセンシング、エルエルシー
 アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ
 (74) 代理人 100107766
 弁理士 伊東 忠重
 (74) 代理人 100070150
 弁理士 伊東 忠彦
 (74) 代理人 100091214
 弁理士 大貫 進介

最終頁に続く

(54) 【発明の名称】オペレーティングシステムと切り離される異種計算

(57) 【特許請求の範囲】

【請求項 1】

1つ又は複数のオペレーティングシステムスレッドを、異種処理コアを管理するハイパーバイザを通じてスケジュールする、コンピュータによって実装される方法であって、

オペレーティングシステムに仮想コアのセットを提示するステップであって、前記仮想コアのセットは、ハイパーバイザによって構成され、前記仮想コアは、コンピューティングデバイスに対してアクセス可能な物理的な処理コア間における1つ又は複数の機能の相違を知ることから前記オペレーティングシステムを分離する、ステップと、

前記ハイパーバイザによって提示された前記仮想コアのセットの特定の仮想コアにおいてスレッドの命令を実行するスレッドスケジューリング要求を、前記オペレーティングシステムから受信するステップと、

前記コンピューティングデバイスを操作するための1つ又は複数の目標を指定するスケジューリングポリシーにアクセスするステップであって、前記目標のうちの少なくとも1つが電力利用に関連する、ステップと、

前記受信したスケジューリング要求に関連付けられる前記スレッドを実行すべき物理的な処理コアを選択するステップであって、当該物理的な処理コアの選択は、前記ハイパーバイザによって、前記アクセスされたスケジューリングポリシー及び利用可能なシステム装置設備に基づいて行われる、ステップと、

前記選択された物理的な処理コアにおいて実行する前記スレッドをスケジュールするステップと、

10

20

を含む、方法。

【請求項 2】

スレッドスケジューリング要求を受信することは、前記オペレーティングシステムがスレッドの実行用に選択する前記特定の仮想コアに関わらず、前記ハイパーバイザが、1つ又は複数のハイパーバイザポリシーに従って前記スレッドを実行すべき任意の特定の物理的な処理コアを選択することを含む、

請求項 1 に記載の方法。

【請求項 3】

前記スレッドを実行する前に、前記受信したスケジューリング要求の前記スレッドに関する 1 つ又は複数の処理ニーズを決定するステップを更に含み、

10

前記ハイパーバイザは、前記スレッドに関連する前記処理ニーズと互換性のある機能を有する物理的な処理コアを選択し、

前記処理ニーズを決定することは、

特定の命令セット、利用可能なコプロセッサ、命令セット拡張又は前記スケジュールされたスレッドによって使用されるアーキテクチャ拡張を決定することと、

前記スレッドの 1 つ又は複数の性能要件を決定することと、

前記スレッドが、より低い電力利用での遅い実行に適切かどうかを決定することと、

追加の処理リソースが利用可能になるまで前記スレッドを遅らせることができるかどうかを決定することと、

特定のオペレーティングシステムに関する特有の知識にアクセスして、特定のスレッドの処理ニーズを決定することと、

20

のうちの少なくとも 1 つを含む、請求項 1 又は 2 に記載の方法。

【請求項 4】

前記スケジューリングポリシーにアクセスすることは、

当該ポリシーが、電力利用の最適化を要求し、利用可能な低電力の処理コアに対してスレッドをスケジューリングすることを指定していると決定することと、

当該ポリシーが、性能の最適化を要求し、利用可能な高性能の処理コアに対してスレッドをスケジューリングすることを指定していると決定することと、

のうち少なくとも一方を含む、請求項 1 乃至 3 のいずれかに記載の方法。

【請求項 5】

30

前記スケジューリングポリシーにアクセスすることは、前記コンピューティングデバイスが、利用可能な電力が制限されたモバイルデバイスであるか、熱制限されたデバイスであり、スレッドがスケジュールされるコアの選択を通して電力利用を最適化するか、熱を管理することを指定していると決定することを含む、

請求項 1 乃至 3 のいずれかに記載の方法。

【請求項 6】

前記物理的な処理コアを選択することは、

異なる機能及び性能 / 電力特性の複数の利用可能なコアのうち、システムがスレッドをスケジュールすることができる 1 つを選択することと、

前記コンピューティングデバイスの性能及び電力利用を管理するための 1 つ又は複数の目標を促進する手法で前記の選択を行うことと、

40

のうち少なくとも一方を含む、請求項 1 乃至 5 のいずれかに記載の方法。

【請求項 7】

前記スレッドと前記選択された物理的な処理コアとの間の 1 つ又は複数の機能的相違を処理するステップを更に含む、

請求項 1 乃至 6 のいずれかに記載の方法。

【請求項 8】

実行すべき前記スレッドをスケジュールすることは、出力を処理することと、前記出力を前記オペレーティングシステムに提供することとを含み、これにより、前記出力は、前記オペレーティングシステムに対して、該オペレーティングシステムが前記スレッドを割

50

り当てた前記仮想コアから来たように見える、
請求項 1 乃至 7 のいずれかに記載の方法。

【請求項 9】

プロセッサによって実行されると、該プロセッサに、請求項 1 乃至 8 のいずれかに記載の方法を実行させるコンピュータプログラム。

【請求項 10】

ハイパーバイザを通じて異種計算を切り離したオペレーティングシステムを提供するためのコンピュータシステムであって、

異種処理機能及び電力プロファイルを有する 1 つ又は複数の処理コアを含む、1 つ又は複数の処理ユニットと、

ハイパーバイザとオペレーティングシステムとの間を通信し、ハードウェアリソースへの配信及び当該ハードウェアリソースからの出力を受信するための命令を受信し、異なるコアが前記オペレーティングシステムに対して同じコアに見えるようにする、オペレーティングシステムインターフェースコンポーネントと、

前記ハイパーバイザが前記オペレーティングシステムに提示する 1 つ又は複数の仮想コアを管理する、仮想コア管理コンポーネントであって、前記 1 つ又は複数の仮想コアは、前記処理コア間における 1 つ又は複数の機能の相違を知ることから前記オペレーティングシステムを分離する、仮想コア管理コンポーネントと、

オペレーティングシステムスレッドをスケジューリングし、前記 1 つ又は複数の処理ユニットに基づいて仮想コアを前記オペレーティングシステムに提示するための 1 つ又は複数のポリシーを管理する、ポリシーエンジンコンポーネントであって、前記ポリシーのうちの少なくとも 1 つが電力利用に関連する、ステップと、

前記オペレーティングシステムから前記 1 つ又は複数の仮想コアのうちの特定の仮想コアにおいてスレッドの命令を実行するスレッドスケジューリング要求を受信し、前記 1 つ又は複数のポリシー及び利用可能なシステム装置設備に基づいて、前記 1 つ又は複数の処理コアの中から、前記受信したスケジューリング要求に関連付けられる前記スレッドを実行すべき物理的な処理コアを選択し、前記選択された物理的な処理コアにおいて実行する前記スレッドをスケジュールする、スケジュールコンポーネントと、

前記ハイパーバイザと前記処理ユニットとの間を通信して、利用可能な物理的なコアにおいて実行するソフトウェア命令をスケジュールするハードウェアインターフェースコンポーネントと、

を備える、コンピュータシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、オペレーティングシステムと切り離される異種計算に関する。

【背景技術】

【0002】

エネルギー効率は、携帯電話からデータセンタにいたるまで、ますます重要な差別化要因になっている。消費者は、より長く続くモバイルデバイス経験に高い値段を払おうとするだけでなく、これらの同じデバイスからより向上した性能が得られることを望む。他方において、データセンタは、計算力をスケールアップし続けるが、効率的に冷却することが可能な温度制限に直面する。加えて、公衆は、エネルギー利用およびエネルギー使用の環境への影響についてますます意識を高めている。エネルギーの使用を効率的にすることは、したがって、多くのタイプのコンピューティングシステムにおいてより優先度の高い設計目標である。

【0003】

これらの技術的に相反する課題、すなわち、より良い性能を供給するが、より少ない電力しか使用しないことは、単一のシステムまたはシリコンチップ内において「大きな」計算コアを「小さな」計算コアに密接に結合する、という異種設計を産業的に試すこととな

10

20

30

40

50

った。これは、本明細書において、異種のコアまたは処理と呼ばれる。大きなコアは、より大きな電力エンベロープで高い性能を提供するように設計され、一方、小さなコアは、より小さな電力エンベロープで低い性能を提供するように設計される。従来の考え方としては、オペレーティングシステムのスケジューラが、次いで、作業量に応じて、スレッドを大きなコアまたは小さなコアに対して選択的にスケジュールすることとなる。1日の少なくとも一部の時間帯の間、オペレーティングシステムは、大きなコアを全体的にターンオフし、小さなコアを少しずつ消費する電源に頼ることが可能である。

【0004】

大きなコアおよび小さなコアは、同じ命令セットまたは特徴を共有しても、共有しなくてもよい。例えば、小さなコアは、減少した命令セットまたは他の差異を含み、この他の差異は、オペレーティングシステムにより、互換性のあるコアに対する処理をスケジュールさせる、さらなる決定を行うことを含む。従来の例の1つは、中央処理ユニット（CPU）およびグラフィックス処理ユニット（GPU）を含み、GPUを、アイドル状態または十分に活用されていないときにタスクを計算するのに使用できるシステムである。

10

【0005】

既存および現在の解決策は、オペレーティングシステムに、大きなコアおよび小さなコアの存在を「知らせる」ためにオペレーティングシステムのカーネルを修正することと、その個々のコアの性能および電力特性と、オペレーティングシステムが、どのコアに対して特定のスレッドをスケジュールするかを決定するためにモニタリングするシステム内の装置設備（例えば、CPU性能カウンタ、キャッシュのミス/ヒットカウンタ、バスアクティビティカウンタなど）による。このアプローチは、いくつかの欠点を有する。すなわち、1) このアプローチは、サポートされるオペレーティングシステム全てについてカーネルを修正する必要があり、2) 修正されたカーネルが、潜在的に異なるアーキテクチャ（例えば、N個の異なる実装をサポートすること）にわたって大きな／小さな設計における相違を理解することが必要であり、3) オペレーティングシステムカーネルのリリーススケジュールと、基礎となるコンピュータアーキテクチャを密に結合する。コンピュータアーキテクチャに対する変更は次いで、カーネルが新しいコアを商業的にサポートするまで、次にスケジュールされたオペレーティングシステムのリリース（すなわち、潜在的には数年またはそれ以上）を待つ必要がある（また、その逆もそうである）。

20

【発明の概要】

30

【0006】

本明細書において、オペレーティングシステムの認識または関与を必要とせずに、エネルギー効率を向上させ、または他の処理目標に合致するため、オペレーティングシステムスレッドのスケジューリングを大きなコアおよび小さなコアにわたって自立的に制御するハイパーバイザを提供する、異種処理システムを説明する。本システムは、仮想化された計算コアの有限のセットをオペレーティングシステムに提示し、本システムは、そのオペレーティングシステムに対して実行用のスレッドをスケジュールする。続いて、水面下で、ハイパーバイザは、各スレッドを実行すべきコア、すなわち大きいコアまたは小さいコア、の物理的な割り当ておよび選択を賢く制御して、エネルギー使用または他の処理要件を管理する。ソフトウェアハイパーバイザを使用して基礎となる大きなおよび小さなコンピュータアーキテクチャを抽象化することによって、コア間の性能および電力動作の相違は、オペレーティングシステムに対して不透明なままとなる。また、固有の間接性は、新たな機能を有するハードウェアのリリースを、オペレーティングシステムのリリーススケジュールと切り離す。

40

【0007】

この発明の概要は、以下の発明の詳細な説明においてさらに説明される概念の選択を、簡略化した形で紹介するために提供される。この発明の概要は、特許請求される主題の主要な特徴または本質的特徴を特定するようには意図されておらず、特許請求される主題の範囲を限定するのに用いられるようにも意図されていない。

【図面の簡単な説明】

50

【0008】

【図1】一実施形態における異種処理システムのコンポーネントを示すブロック図である。

【図2】一実施形態において、異種処理コアを有するコンピューティングデバイスを、異種処理コアとオペレーティングシステムとの間のハイパーバイザを用いて初期化する、異種処理システムの処理を示すフロー図である。

【図3】一実施形態において、異種処理コアを管理するハイパーバイザを通じて1つまたは複数のオペレーティングシステムスレッドをスケジューリングする、異種処理システムの処理を示すフロー図である。

【図4】一実施形態における異種処理システムの動作環境を示すブロック図である。

10

【発明を実施するための形態】**【0009】**

本明細書において、オペレーティングシステムの認識または関与を必要とせずに、エネルギー効率を向上させ、または他の処理目標に合致するため、オペレーティングシステムスレッドのスケジューリングを大きなコアおよび小さなコアにわたって自立的に制御するハイパーバイザを提供する、異種処理システムを説明する。本システムは、仮想化された計算コアの有限のセットをオペレーティングシステムに提示し、本システムは、そのオペレーティングシステムに対して実行用のスレッドをスケジューリングする。続いて、水面下で、ハイパーバイザは、各スレッドを実行すべきコア、すなわち大きいコアまたは小さいコア、の物理的な割り当ておよび選択を賢く制御して、エネルギー使用または他の処理要件を管理する。ソフトウェアハイパーバイザを使用して基礎となる大きなおよび小さなコンピュータアーキテクチャを抽象化することによって、コア間の性能および電力動作の相違は、オペレーティングシステムに対して不透明なままとなる。また、固有の間接性は、新たな機能を有するハードウェアのリリースを、オペレーティングシステムのリリーススケジュールと切り離す。ハードウェアのベンダは、更新されたハイパーバイザをリリースし、新たなハードウェアが、そのベンダにより選ばれた任意のオペレーティングシステムのバージョンで機能できるようにすることができる。

20

【0010】

ハイパーバイザの実装は、基礎となるコンピュータアーキテクチャと密に結合され、利用可能なシステムフィードバック（例えば、CPU利用、バス/キャッシュアクティビティなど）を使用して、適切なコアを要求される作業量に自立的に割り当てる。このアプローチは、基礎となるコンピュータアーキテクチャが、ソフトウェアハイパーバイザとの連携でしばしば変化し、この発展を、上記オペレーティングシステムから切り離すことを可能にする。異種処理システムは、オペレーティングシステムカーネル自体を変更することなく、簡単で粗粒な電力管理を提供する。したがって、異種処理システムは、より高速なハードウェア革新を可能にし、既存のデータセンタおよび他のインストールが今日、利用可能な異種処理ハードウェアからの利益を享受することを可能にする。

30

【0011】

異種計算は、システムにおいて利用可能な異なるタイプの計算コア（例えば、CPU、GPU、アクセラレータなど）に基づいて作業量の実行を最適化する、という目標を有する産業の新興分野である。最適化は、性能、電力、待ち時間、または他の目標について可能である。異種処理システムは、これらのより一般的な場合に適用可能である一方で、一意の機能的等価性を有するが、異なる性能/電力動作特性を有するコアを用いるシステムも対象とすることができます。典型的に、これらのシステムは、1つまたは複数の大きなコアと、1つまたは複数の小さなコアを有する。大きなコアは典型的に、深いパイプライン、順序不順の実行ファイル、大きなキャッシュ、高いクロックスピードを有し、そして高い漏出処理量（例えば、40G）を使用して操作される。小さなコアは典型的に、短いパイプライン、小さなキャッシュ、低クロックスピード、様々な電力レベルを有し、そして低い漏出処理量（例えば、40LP）を使用して操作される。

40

【0012】

50

一部の実施形態において、大きなコアおよび小さなコアは、アーキテクチャの等価物、マイクロアーキテクチャの等価物、グローバルインタラプトコントローラ、コヒーレンシー、および仮想化などを有する。アーキテクチャの等価物には、同じインストラクションセットアーキテクチャ (ISA : Instruction Set Architecture)、単一命令複数データ (SIMD : Single Instruction Multiple Data)、浮動小数点 (FP : Floating Point)、コプロセッサの利用可能性、およびISA拡張などが含まれ得る。マイクロアーキテクチャの等価物は、性能における相違は有するが、同じ構成可能な特徴 (例えば、キャッシュライン長) を含む。グローバルインタラプトコントローラは、全てのコアに対する中断を管理し、処理し、転送する能力を提供する。コヒーレンシーは、全てのコアが、必要に応じて転送して、他のコアからのデータにアクセス (キャッシュ) することができるこ¹⁰とを意味する。仮想化は、コアからの / コアへの作業量を切り替える / 移行するためのものである。

【0013】

一部の実施形態において、異種処理システムは、コア間のわずかな相違に対応することができる。例えば、ストリーミング SIMD 拡張命令 (SSE) (現在、4つの反復、SSE1、SSE2、SSE3、SSE4 が存在する) をサポートしない小さなコアは、他のインテルの x86 ベースのソフトウェアコードを扱うことができる。ハイパーバイザは、命令ストリーム内のサポートされていない命令を検出し、そのようなストリームを割り当てる適切なコアを起動することができる。他の命令ストリームは、任意のコアにおいて正確に動作する。少数のサポートされていない命令のみが使用される場合など、一部の場合において、ハイパーバイザは、なんらかのレベルのエミュレーションを含み、利用可能な命令セットにおいてサポートされていない命令をエミュレートする。例えば、ベクトル計算などの操作はしばしば分解されて、標準の計算命令を用いる低効率で実装される。²⁰

【0014】

ソフトウェアハイパーバイザは、デバイスのブートプロセスの間にオペレーティングシステム (OS) の初期化に先行して、ソフトウェアハイパーバイザ自身をインストールする。指定されたハードウェア構成が完了した後 (すなわち、メモリを構成し、仮想化装置設備を初期化するなど)、ハイパーバイザは、次いでコンピューティングデバイス内にインストールされた大きな処理コアおよび小さな処理コアを、ポリシーにより構成することができる。例えば、デバイスがモバイルデバイスである場合、ポリシーは、ハイパーバイザがオペレーティングシステムを、バッテリの需要のために利用可能かつ最適な最小限の性能で開始するように指示することができ、ハイパーバイザは続いて、オペレーティングシステムスレッドを 1 つまたは複数の小さなコアに対してスケジュールする。あるいは、デバイスが、データセンタブレードである場合、ポリシーは、ハイパーバイザがオペレーティングシステムを利用可能な最大限の性能で開始して、エネルギー効率を犠牲にするように指示することができ、ハイパーバイザは続いて、オペレーティングシステムスレッドを利用可能な大きなコアに対してスケジュールし、さらに場合によっては、利用可能な熱量に応じて小さなコアに対してもスケジュールすることもできる。初期化が完了した後、ソフトウェアハイパーバイザは、オペレーティングシステムブートマネージャをロードし、このオペレーティングシステムブートマネージャが次いでオペレーティングシステムをロードする。³⁰

【0015】

ランタイムの間、異種処理システムは、仮想化されたコアのセットをオペレーティングシステムに提示する。コア間の動作特性および相違は、オペレーティングシステムに対して不透明であり、ソフトウェアハイパーバイザにより、定義された動作ポリシーに基づいて非公開で管理される。動作ポリシーは、システムの初期化中に設定されるか、またはランタイム中に動的に設定され得る。

【0016】

ハイパーバイザは、動作ポリシーを、利用可能なシステム装置設備 (例えば、CPU 性能カウンタ、キャッシュのミス / ヒットカウンタ、バスアクティビティカウンタなど) と⁴⁰

ともに使用し、どのコアをオペレーティングシステムスレッドにスケジュールすべきかを決定する。ハイパーバイザは、この情報を使用して、CPUコアの利用、経時的な傾向、情報の局所性、および入出力(I/O)パターンを理解することができる。この情報から、ハイパーバイザは、動的かつ投機的に、オペレーティングシステムスレッドを大きなコアおよび小さなコアにわたって適切に移行することができる。加えて、ハイパーバイザは、ダイナミック周波数・電圧スケーリング(DFS:dynamic frequency and voltage scaling)を、オペレーティングシステムの代わりにシステム実装に応じて制御することもできる。

【0017】

ここに、ハイパーバイザが制御することができる利用可能な動作ポリシーのサンプルを示す。最小電力(MiPo:Minimum Power)、最大性能(MaPe:Maximum Performance)、最小電力・オンデマンド性能(MiPoD:Minimal Power, Performance on Demand)、最大性能・アイドル状態電力低下(MaPe1:Maximum Performance, Power Down on Idle)。これらはそれぞれ、以下の段落において説明する。しかしながら、任意の特定の実装によって選択されるように、付加的な、より進化した動作ポリシーを実装することができる。

【0018】

最大電力(MiPo)は、スレッドをコアの最小のセットに対してスケジュールする。これは、典型的に、ハイパーバイザが、スレッドを小さいコアに対してスケジュールし、必要に応じてDFSを使用して、コア用の電力および性能動作ポイントを制御することができることを意味する。必要に応じて、追加の小さなコアに電力を供給して、スケジュールすることもできる。

【0019】

最大性能(MaPe)は、スレッドをコアの最大のセットに対してスケジュールする。これは、典型的に、ハイパーバイザが、スレッドを全ての利用可能なコア(大きなコアから開始する)に対してスケジュールして、必要に応じてDFSを使用し、コア用の電力および性能動作ポイントを制御することを意味する。利用可能な熱量によって許容される程度に、小さなコアにも電力を供給してスケジュールする。

【0020】

最小電力・オンデマンド性能(MiPoD)は、通常、利用可能な最小の電力状態(例えば、1つまたは複数の小さなコアにおいて)で動作するが、処理負荷の要求により性能を高める。これは一般に、動作の「ターボ」または「ブースト」モードと呼ばれ、大きなコアを動的に割り当ててスケジュールすることにより可能になる。作業量が完了すると、システムは最小の電力状態(例えば、1つの小さなコア)に戻る。

【0021】

最大性能・アイドル状態電力低下(MaPe1)は、通常、最大の利用可能な性能状態(例えば、1つまたは複数の大きなコア)で動作するが、アイドル閾値に達すると、低下した電力状態に従う。アイドル閾値はこの場合、典型的なほぼゼロのCPU利用ではないが、アイドル閾値は、何らかのドライストンMIPS(DMIPS)、またはポリシーによって定義されるCPU利用率において任意に定義することができる。アイドル状態になると、ハイパーバイザは動的に、小さなコアを割り当ててスケジュールし、未使用の大きなコアをスタンバイ/保留状態に置く。ポリシーおよび/または将来の処理量は、システムが最大の利用可能な性能状態(例えば、大きなコア)に戻るときを決定する。

【0022】

図1は、一実施形態における異種処理システムのコンポーネントを示すブロック図である。システム100は、1つまたは複数の中央処理ユニット110、オペレーティングシステムインターフェースコンポーネント120、仮想コア管理コンポーネント130、ポリシーエンジンコンポーネント140、ポリシーデータストア150、スケジューリングコンポーネント160、機能管理コンポーネント170、およびハードウェアインターフェースコンポーネント180を含む。これらのコンポーネントの各々を、ここでさらに詳細に

10

20

30

40

50

説明する。以下のコンポーネントを、コンピューティングデバイスのオペレーティングシステムとハードウェアリソースとの間に存在するソフトウェアハイパーバイザ内で実装することができる。

【0023】

1つまたは複数の中央処理ユニット110は、異種処理機能および電力プロファイルを有する1つまたは複数の処理コアを含む。典型的に、各CPU複合体は、単一のシリコン型上に配置され、CPU複合体の各コアは、シリコン型を共有する。ハードウェアを、様々なタイプのデバイス用の様々なパッケージで実装することができる。例えば、新しいモバイルデバイスおよびつい最近のデスクトッププロセッサは、相互間の効率的な通信と、より低い電力利用のために、同じチップ上にCPUとGPUを含む。各CPU複合体は1つまたは複数の大きなコアおよび小さなコアを含むことができる。あるいはまた、あるCPU複合体が大きなコア全てを含み、別のCPU複合体が小さなコアを全て含むこともある。ここで使用されるようなCPU複合体は、GPU、およびソフトウェア命令を実行することができる他のハードウェアに当てはまる。

10

【0024】

オペレーティングシステムインターフェースコンポーネント120は、ハイパーバイザとオペレーティングシステムとの間を通信し、ハードウェアリソースに配信してハードウェアリソースからの出力を受け取るための命令を受信する。オペレーティングシステムは、スレッドをスケジュールし、命令ストリーム（例えば、プログラムカウンタ（PC：program counter））へのポインタを提供し、命令をハードウェアに渡すメモリ領域への書き込みを行うことができる。オペレーティングシステムは典型的に、コンピューティングデバイス上のハードウェアと直接対話する。しかしながら、ハイパーバイザは、様々な目的で、オペレーティングシステムとハードウェアとの間に間接的なレイヤを挿入する。しばしば、ハイパーバイザを使用して仮想化を提供し、その結果、複数のオペレーティングシステムを同時に同じハードウェア上で実行することができる。ハイパーバイザを使用して、コンピューティングシステムにインストールされた実際のハードウェアとは異なる仮想ハードウェアを、オペレーティングシステムに提示することもできる。異種処理システム100の場合、これは、大きなコアと小さなコアがオペレーティングシステムに対して同じに見えるようにすることを含み得る。システム100は、デバイスに実際に存在するものとは異なる数のコアを、オペレーティングに提示することもできる。

20

【0025】

仮想コア管理コンポーネント130は、ハイパーバイザがオペレーティングシステムに提示する1つまたは複数の仮想コアを管理する。仮想コアは、オペレーティングシステムに対してCPUコアのように見えるが、コンピューティングデバイスにおいて利用可能な物理的なハードウェアの特性とは異なるものとすることができます。例えば、仮想コアは、処理における相違または電力機能をオペレーティングシステムから隠すことができ、その結果、異種の大きなおよび小さなコアとともに機能するように設計されていないオペレーティングシステムが、そのように設計されたオペレーティングシステム用のやり方で動作することができる。このような場合、ハイパーバイザは、異種計算環境を用いるのに必要な特別なプログラミングを提供することができ、したがってオペレーティングシステムを修正する必要がない。

30

【0026】

ポリシーエンジンコンポーネント140は、オペレーティングシステムスレッドをスケジューリングし、利用可能な1つまたは複数の中央処理ユニットに基づいて仮想コアをオペレーティングシステムに提示するための、1つまたは複数のポリシーを管理する。ポリシーエンジンコンポーネント140は、特定のハイパーバイザの実装に特有のハードコードされたポリシーを含むことができ、あるいは、特定のインストール目標に適合するように修正することができる管理者構成可能なポリシーを含むことができる。ポリシーは、どのコアを最初にスケジュールするか、電力利用と処理目標のトレードオフ、電力を抑えるためにコアをどのようにシャットオフおよび起動するか、仮想コアをどのようにオペレー

40

50

ティングシステムに提示するかなど、を決定することができる。

【0027】

ポリシーデータストア150は、1つまたは複数のポリシーを、ブート時および実行時にハイパーバイザに対してアクセス可能な記憶装置に格納する。ポリシーデータストア150は、1つまたは複数のファイル、ファイルシステム、ハードドライブ、データベース、またはシステム100の実行セッションにわたってデータを保持するための他の記憶装置を含む。一部の実施形態において、管理者は、システム100に、ハイパーバイザによる使用のためのポリシーの初期のセットを格納する構成段階を行わせる、セットアップステップを実行する。

【0028】

スケジューリングコンポーネント160は、オペレーティングシステムからスレッドとして受信される1つまたは複数の命令ストリームを、コンピューティングシステムにインストールされた中央処理ユニットの1つまたは複数に対してスケジュールする。スケジューリングコンポーネントは、仮想コア識別をオペレーティングシステムから受信する。仮想コア識別は、オペレーティングシステムがスレッドをスケジュールするよう要求する仮想コアを識別する。スケジューリングコンポーネント160は、スケジュール要求を検査し、実行すべきスレッドをスケジュールする物理的なコアを決定する。例えば、コンポーネント160は、電力または処理がスレッドにより関連するかどうかを決定し、それに応じて適切な小さいコアまたは大きなコアをスケジュールすることができる。一部の場合において、コンポーネント160は、スレッドをある特定のコアに対してスケジューリングすることを回避し、これらのコアがパワーダウンして電力を抑えることができるようとする。

【0029】

機能管理コンポーネント170は、任意選択で、大きな処理コアと小さな処理コアとの間の1つまたは複数の相違を管理する。一部の場合において、システム100は、大きなコアと小さなコアが同じ機能を共有し、かつ機能管理コンポーネント170が必要はされない、処理ユニット上においてのみ動作することがある。別の場合において、システム100は、利用可能な処理コアの間の軽微な相違または主要な相違を扱う。例えば、システム100は、一部のコアによってサポートされない命令を観察し、対応するスレッドを、これらの命令をサポートするコアにおいてスケジュールすることができる。より高度な実装において、コンポーネント170は、電力目標または他のプロファイル目標を満たすように、大きなコアの機能を小さなコアにおいて（あるはその反対）仮想化またはエミュレートすることができる。

【0030】

ハードウェアインターフェースコンポーネント180は、ハイパーバイザと中央処理ユニットとの間を通信し、利用可能な物理的コア上で実行するソフトウェア命令をスケジュールすることができる。ハードウェアインターフェースコンポーネント180は、実際のメモリアドレス、あるいは実際のハードウェアにアクセスするための他の装置を含む。これらの他の装置は、他のコンポーネントから、特にハイパーバイザによって管理されるゲストオペレーティングシステムから隠される。

【0031】

異種処理システムが実装されるコンピューティングデバイスは、中央処理ユニット、メモリ、入力デバイス（例えば、キーボードおよびポインティングデバイス）、出力デバイス（例えば、ディスプレイデバイス）、ならびに記憶デバイス（例えば、ディスクドライブ、または他の非揮発性記憶媒体）を含むことができる。メモリおよび記憶デバイスは、システムを実装または可能にするコンピュータ実行可能命令（例えば、ソフトウェア）によりエンコードされる、コンピュータ読み取り可能記憶媒体である。さらに、データ構造およびメッセージ構造を、通信リンク上の信号などのデータ伝送媒体を介して格納または伝送することができる。様々な通信リンク、例えばインターネット、ローカルエリアネットワーク、広域ネットワーク、ポイントツーポイントダイアルアップ接続、携帯電話ネットワ

10

20

30

40

50

ークなどを使用することができる。

【0032】

本システムの諸実施形態を様々なオペレーティング環境において実装することができる。このようなオペレーティング環境には、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドもしくはラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースのシステム、プログラム可能な家庭用電化製品、デジタルカメラ、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、ならびに、上記システムまたはデバイス、セットトップボックス、およびシステムオンチップ(SOC)のいずれかを含む分散コンピューティング環境などが含まれる。コンピュータシステムは、携帯電話、PDA、スマートフォン、パーソナルコンピュータ、プログラム可能な家庭用電化製品、デジタルカメラなどとすることができます。10

【0033】

本システムを、1つまたは複数のコンピュータまたは他のデバイスによって実行されるプログラムモジュールなど、コンピュータ実行可能命令の一般的なコンテキストにおいて説明することができる。一般に、プログラムモジュールは、特定のタスクを実行し、特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含む。典型的に、プログラムモジュールの機能を、様々な実施形態の所望に応じて結合し、または分散することができる。

【0034】

図2は、一実施形態において、異種処理コアを有するコンピューティングデバイスを、その異種処理コアとオペレーティングシステムとの間のハイパーバイザを用いて初期化する、異種処理システムの処理を示すフロー図である。20

【0035】

ブロック210において開始し、システムは、起動要求を受信してコンピューティングデバイスを初期化する。例えば、基本入出力システム(BIOS)、拡張可能ファームウェアインターフェース(EFI)、ブートローダ、または他の初期化デバイスソフトウェアは、異種コンピューティングシステムを実装するハイパーバイザをロードして起動することができる。一部の場合において、管理者が、ハイパーバイザをコンピューティングデバイスにインストールするインストール段階を以前に実行したことがある場合も、システムは、ネットワークブート、および、コンピューティングデバイスに共通して供給される他の非インストールのシナリオをサポートすることもできる。30

【0036】

ブロック220に続くと、システムは、コンピューティングデバイスの2つまたはそれ以上の物理的な処理コアを列挙する。一部の実施形態において、少なくとも2つのコアが、異なる性能および電力利用特性を提供する。しかしながら、非対称性が存在しないシステムを使用することもできる。例えば、電力管理にソフトウェアハイパーバイザを使用することは、オンダイのN個の物理的なCPUを有するが、K個のみを、気温、フォームファクタのエンクロージャ、利用可能な電力のコストなどの外部性に基づいて操作することができる、というシナリオにおいても適用可能である。ブートにおいて、ハイパーバイザは、この「ポリシー」情報を使用して、K個のコアの仮想化されたセットをオペレーティングシステムに報告することができ、これは、各ブートサイクルにおいて異なる可能性がある。ハイパーバイザは、このシナリオにおいて対称的なコアに対して同じタスクを実行しているであろう。システムは、BIOSまたは他の基礎となるレイヤを起動して、どれくらいの、およびどの種類のプロセッサをコンピューティングデバイスがインストールしたかを決定し、そして、CPUIDまたは他の同様の命令を実行して、プロセッサの処理機能に関する情報を決定することができる。一部の実施形態において、システムは拡張性インターフェースを含むことができ、この拡張性インターフェースを通じて、ドライバまたは他のハイパーバイザ拡張を実装して、ハイパーバイザ製造者または第三者によって追加することができ、必ずしもハイパーバイザ自体を更新する必要なく、新たな処理ハードウェアのサポートをハイパーバイザに追加することができる。4050

【0037】

ブロック230に続くと、システムは、それぞれの列挙された処理コアの機能を決定する。この機能は、各コアによって提供される1つまたは複数の電力プロファイル、各コアによってサポートされる1つまたは複数の命令セット、各コアの性能特性などを含むことができる。システムは、コア自体の情報のインターフェース（以前に言及されたCPUID命令など）、あるいはドライバまたは他の拡張によりハイパーバイザに提供された情報を用いて、各コアの機能を決定することができる。システムは、決定された機能を使用して、コアと互換性のあるスレッドを各コアに割り当てて、受信したポリシーおよび処理目標と整合する方法でスケジューリングを実行する。

【0038】

ブロック240に続くと、システムは、ハイパーバイザが、列挙された物理的コアに関するアクセスおよびスケジューリングを管理することとなる、1つまたは複数のオペレーティングシステムを識別する。システムは、ハードドライブ、フラッシュドライブ、またはコンピューティングデバイスの他の記憶装置にアクセスして、ハイパーバイザが初期化された後にどのオペレーティングシステムを起動すべきかを決定することができる。ハイパーバイザは、様々なオペレーティングシステムに関する情報を用いて指定することができ、そして拡張性を含むことができるので、ハイパーバイザ自体を更新することなく、新たなオペレーティングシステムをサポートすることができる。各オペレーティングシステム、およびオペレーティングシステムバージョンは、異なるスケジューリングセマンティクスを有し、あるいはオペレーティングシステムが仮想化された処理リソース上で正しく実行できるようにハイパーバイザが対処する他のニュアンスを有することがある。一部に場合において、ハイパーバイザは、複数のオペレーティングシステムが、列挙された物理的な処理コアを共有できるようにするのに必要とされ、ポリシーは、そのような共有をどのように扱うかを指示することができる。

【0039】

ブロック250に続くと、システムは、ハイパーバイザポリシー情報にアクセスする。ハイパーバイザポリシー情報は、列挙された物理的な処理コアにおいてオペレーティングシステムスレッドをスケジュールするための1つまたは複数の目標を指定する。この目標には、性能目標、電力利用目標、またはどのコア（1つまたは複数）がオペレーティングシステムスレッドを実行すべきかを決定するための他の方針を含むことができる。ポリシーを、コンピューティングデバイスに関連付けられた記憶デバイスに格納するか、またはハイパーバイザの実装にハードコードすることができる。ハイパーバイザは、ポリシーに対する更新を、管理者によって提供される管理インターフェースを通じて受信することができる。

【0040】

ブロック260に続くと、システムは、1つまたは複数の仮想コアを生成して、識別されたオペレーティングシステムに公開する。ここで、各仮想コアは、物理的な処理コア間における決定された機能の相違から、オペレーティングシステムを分離する。例えば、異種処理システムは、2つまたはそれ以上の大きなコアおよび小さなコアを、単一のタイプの統一仮想コアとしてオペレーティングシステムに提示することができる。仮想コアにおいてスレッドを実行するスケジューリング要求をオペレーティングシステムから受信すると、システムは、どの物理コアをジョブ用に選択すべきかを、アクセスしたハイパーバイザポリシーに基づいて決定する。ハイパーバイザポリシーは、例えば、低電力供給コアを用いることが支持され、少なくとも1つの高電力供給コアをシームレスに電力低下させることを可能にすることを目標とするときなどに、ハイパーバイザが物理コアではなく異なる数の仮想コアを提示すべきことを指定する。あるいは、システムは、オペレーティングシステムが、そのコアを使用することを選択した場合、または低電力コアだけでは満たすことのできないコア品質を使用することを選択した場合、オペレーティングシステムが認識したコアの電力を低下させるが、これらのコアを起動させておく。

【0041】

10

20

30

40

50

ブロック 270 に続くと、システムは、識別されたオペレーティングシステムを起動し、識別されたオペレーティングシステムを、列挙された物理的な処理コアから分離したまま、作成された仮想コアをオペレーティングシステムに提示する。オペレーティングシステムを起動することは、オペレーティングシステムローダを起動することと、ハイパーバイザを通常の BIOS の代わりに、またはオペレーティングシステムを基礎とする他のレイヤの代わりに提示することとを含むことができる。オペレーティングシステムは、物理的なハードウェア上で直接稼動しているように動作するが、ハイパーバイザは、オペレーティングシステムと、本明細書で説明されるオペレーティングシステムの知識を必要とせずにスケジューリングロジックを実行する、物理的なハードウェアとの間に存在する。ブロック 270 の後、これらのステップは終了する。

10

【 0042 】

図 3 は、一実施形態において、1つまたは複数のオペレーティングシステムスレッドを、異種処理コアを管理するハイパーバイザを通じてスケジュールする、異種処理システムの処理を示すフロー図である。

【 0043 】

ブロック 310 において始まり、システムは、オペレーティングシステムからスレッドスケジューリング要求を受信して、ハイパーバイザによって提示される、識別された仮想コアにおいてスレッドの命令を実行する。仮想コアは、コンピューティングデバイスに対してアクセス可能な 2つまたはそれ以上の物理的な処理コアの間の 1つまたは複数の機能上の相違から、オペレーティングシステムを分離する。オペレーティングシステムは、典型的に、各検出された処理コアに対するアイドルループを有し、オペレーティングシステムは、当該オペレーティングシステムがそのコアにおいて実行することを望む任意の命令をスケジュールし、配置することができる。オペレーティングシステムは、複数のアプリケーションスレッドを時間分割して、特定の処理コアにおいて実行することができる。オペレーティングシステムがスレッドの実行用に選択する特定の仮想コアに関わらず、ハイパーバイザは、任意の特定の物理的な処理コアを選択して、1つまたは複数のハイパーバイザポリシーに従ってスレッドを実行することができる。

20

【 0044 】

ブロック 320 に続くと、システムは、受信したスケジューリング要求の処理ニーズを決定する。例えば、システムは、スケジュールされたスレッドによって使用される特定の命令セット（例えば、1つまたは複数の命令セットの拡張、コプロセッサ、または他の機能が要求されているかどうか）、スレッドの性能要求、追加の処理リソースが利用可能になるまでスレッドが遅滞し得るかどうかなどを決定することができる。システムは、特定のオペレーティングシステムに関する特有の知識、またはポリシーを通じて受信した命令を使用して、特定のスレッドの処理ニーズを決定することができる。例えば、システムは、オペレーティングシステムの内部動作に関するスレッド、アプリケーションスレッドなどを識別し、それぞれをポリシーに従って対処することができる。

30

【 0045 】

ブロック 330 に続くと、システムは、デバイスを操作するための 1つまたは複数の目標を指定するスケジューリングポリシーにアクセスする。例えば、ポリシーは、電力利用、性能、またはこれら 2つの組み合わせの最適化を要求することがある。ポリシーは、デバイスに関連付けられたデータストアに格納されるか、ハイパーバイザの特定の実装にハードコードされる。例えば、システムは、高性能のタスクを実行するスレッドがオペレーティングシステムによってスケジュールされるまで、小さな処理コアを支持する電力使用バージョンのハイパーバイザを提示することできる。この時点で、システムは、高性能のタスクを大きなコアにおいてスケジュールし、次いで、タスクが完了した後は大きなコアをスリープに導くことができる。

40

【 0046 】

ブロック 340 に続くと、システムは、受信したスケジューリング要求に関連付けられたスレッドを実行すべき物理的な処理コアを選択する。この選択は、アクセスされたスケ

50

ジューリングポリシーに基づいて行うことができる。システムは、異なる機能、性能／電力特性の複数の利用可能なコアを有し、システムは、これらのコアに対してスレッドをスケジュールすることができる。システムのコアの選択に基づいて、コンピューティングデバイスは、より多くのまたはより少ない電力を使用して、より長いまたはより少ない時間でスレッドの実行を完了することとなる。ジョブまたはスケジューリングポリシーは、システムが、コンピューティングデバイスの性能、電力、または他の特性を管理するための1つまたは複数の目標を促進する手法で選択を行うのを可能にする。モバイルデバイスはより少ない電力使用を好むが、高性能なサーバは高性能を好む。一部の場合において、ポリシーは、時刻（例えば、電子的コストのピークと非ピーク）または他の考慮事項に基づいて異なるものとすることができ、その結果、ポリシーは、経時的に、または特定の条件に基づいて変化する。

【0047】

ブロック350に続くと、システムは、任意選択で、スレッドと、選択された物理的処理コアとの間の全ての機能的相違に対処する。例えば、スレッドが、選択されたコアにおいて利用可能でない命令を含む場合、システムは、その命令をエミュレートし、またはその命令を、選択されたコアによってサポートされる1つまたは複数の等価な命令で置換することができる。機能的相違を管理することは、顕著な複雑性をシステムに与え、ハイパーバイザ実装者は、任意の特定の実装が、処理コア間の機能においてどのくらい多くまたは（必要に応じて）少ない相違をサポートするかを選択することができる。

【0048】

ブロック360に続くと、システムは、選択された物理的な処理コアにおいて実行するスレッドをスケジュールすることができる。システムは、出力がオペレーティングシステムによりスレッドを割り当てられた仮想コアから来たように見えるように、出力を処理して、この出力をオペレーティングシステムに返す。したがって、オペレーティングシステムは、ハイパーバイザによって管理されるコアのタイプおよび数を知らないままであり、オペレーティングシステムが通常通り、ハイパーバイザおよび異種処理コアを用いないシステムに存在しているように、仮想コアのセットを使用することができる。ブロック360の後、これらのステップは終了する。

【0049】

図4は、一実施形態における異種処理システムの動作環境を示すブロック図である。ゲストオペレーティングシステム410は、本明細書において説明される異種処理システムを実装するソフトウェアハイパーバイザ430によって提示される、1つまたは複数の仮想処理コア420を見る。ソフトウェアハイパーバイザ430は異種処理ハードウェア440を管理する。異種処理ハードウェア440は、グローバルインターフェース450と、1つまたは複数の大きなコア460と、1つまたは複数の小さなコア470とを含む。例示の容易性のために、2つのコアタイプのみを示しているが（大きいものと、小さいもの）、システムは、任意の数の異なるコアで動作することができることに留意されたい。例えば、一部のプロセッサパッケージは、段々と電力利用および性能が減少する、いくつかの処理ストアを含むことができる。ソフトウェアハイパーバイザ430は、異種処理ハードウェア440を管理して、ゲストオペレーティングシステム410を、ハードウェア440を効果的に使用するのに必要ないずれの特別な知識または処理から分離することができる。したがって、ソフトウェア430は、ゲストオペレーティングシステム410などの未修正のレガシーオペレーティングシステムが、新たな異種処理ハードウェア440を利用できるようにする。オペレーティングシステム410が（より良い電力／性能特性は有するが）通常通りに動作し続けている間に、ハードウェア440の変更について行くようにハイパーバイザ430を修正することができる。

【0050】

一部の実施形態において、異種処理システムは、スレッドが既に実行された後に、当該スレッドを、ある物理的処理コアから別のコアへ移行する。一部の場合において、ハイパーバイザが、電力消費を低減する、性能を向上させる、あるいは他のポリシー目標を実行

10

20

30

40

50

する、と決定すると、1つまたは複数のスレッドが既に実行中であることがある。コアがキャッシュストレージまたは他の装置設備を共有しているので、ハイパーバイザは、スレッドのデータへのアクセスに影響を与えることなく、スレッドを別のコアに移行することができる。したがって、ハイパーバイザは、スレッドの実行をインタラプトして、スレッドの命令ストリームを異なる物理的処理コアに移して、ターゲットコアにおいて実行を再開することができる。

【0051】

一部の実施形態において、異種処理システムは、プロセッサ電圧および周波数修正を用いて、異なるコアを選択する前に、電力を低減させるか性能を閾値まで向上させる。例えば、システムは、大きなコアにおいて特定のスレッドの実行を開始し、次いで、大きなコアの電力使用を、当該コアの動作電圧を低減することによって元のスケールに戻し、最終的に、大きなコアの作業を小さなコアに移行することができる。これは、システムが電力使用を徐々に減らし、熱エンベロープを管理するか、またはポリシーによって指定された他の計算目標を満足することを可能にする。

10

【0052】

一部の実施形態において、異種処理システムは、一部の処理タスクをクラウドコンピューティング装置設備に移行することを可能にする。システムは、クラウドコンピューティング装置設備を、単なる別の処理コアとして、スケジュールすべきタスクに提示する。適切なタスクのために、システムは、タスクをコンピューティングデバイスから全体的にオフロードし、後で、タスクの出力をゲストオペレーティングシステムに返すことができる。これは、システムが、コンピューティングデバイスにおいて低電力状態に入り、あるいは作業を、電力コストがピークのデータセンタから、電力コストのより低いデータセンタへ遷移するのを可能にする。

20

【0053】

一部の実施形態において、異種処理システムは、競合状態に対処し、ソフトウェアロックパラダイムを用いてオペレーティングシステムの期待値を管理する。多くの場合において、オペレーティングシステムは、特定のスレッド間の相互依存性または依存性の欠如に基づいてスレッドをスケジュールする。ソフトウェアは、ロック、ミューテックス、セマフォ、または、スレッドを複数同時に実行する環境でソフトウェアコードが正しく動作できるようにオペレーティングシステムによって提供される、他の同期化プリミティブを用いることができる。異種コンピューティングシステムは、オペレーティングシステムのスレッドの安全性および他の同期化に関する保証事項が確実に満たされるようにし、また新たな競合条件または他の問題が導入されないことを保証するように追加のロックを導入するか、スレッドのスケジューリングを決定することができる。

30

【0054】

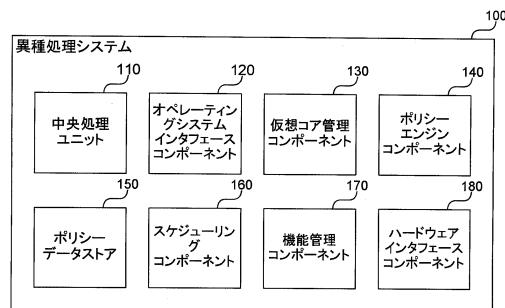
一部の実施形態において、異種処理システムは、ハードウェアハイパーバイザを含む。本明細書の諸実施例では、ソフトウェアハイパーバイザを使用したが、コンピューティングタスクを実装するためのハードウェアまたはソフトウェアの選択はしばしば、性能または他の目標に合致するように切り替えることが可能な実装の詳細であることが当業者には認識されよう。したがって、本システムを、ハードウェアハイパーバイザを用いて実装することができ、一部の処理ユニットは、処理ユニット自体に本システムを含むように生成されることもある。

40

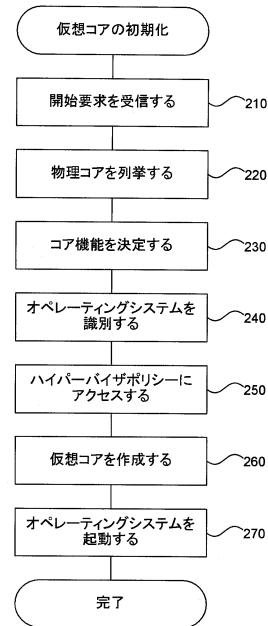
【0055】

以上のことから、本明細書では、例示の目的で、異種処理システムの特定の実施形態を説明してきたが、本発明の精神および範囲から逸脱することなく、様々な修正を行うことが可能であることが理解されよう。したがって、本発明は、添付の特許請求の範囲以外によって限定されることはない。

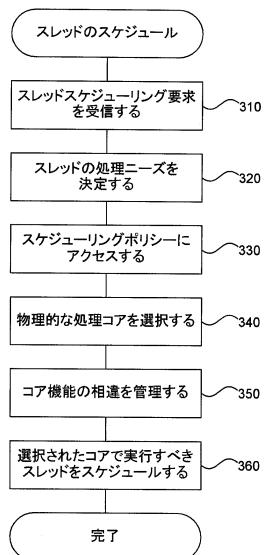
【図1】



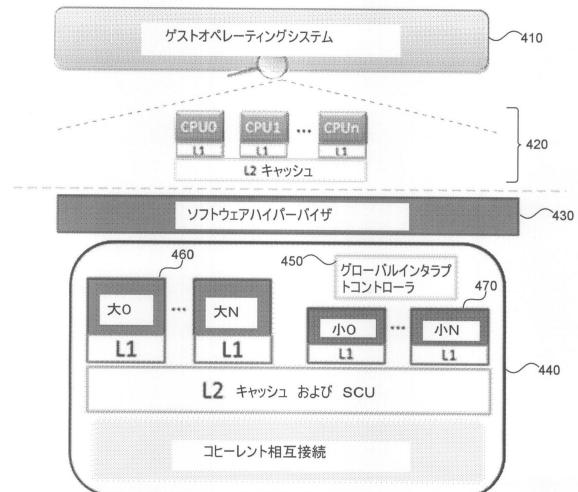
【図2】



【図3】



【図4】



フロントページの続き

(72)発明者 アーシエイム , ジェレッド

アメリカ合衆国 98052-6399 ワシントン州 レッドモンド ワン マイクロソフト
ウェイ マイクロソフト コーポレーション エルシーエー - インターナショナル パテンツ 内

審査官 篠塚 隆

(56)参考文献 特開2004-252900 (JP, A)

特開2011-34578 (JP, A)

Youngjin Kwon, Changdae Kim, Seungryoul Maeng, and Jaehyuk Huh, "Virtualizing Performance Asymmetric Multi-core Systems", ISCA '11 Proceedings of the 38th annual international symposium on Computer architecture, 米国, ACM, 2011年 6月 4日, p.45-56

(58)調査した分野(Int.Cl. , DB名)

G 06 F 9 / 46

9 / 48

9 / 50 - 9 / 52

9 / 54