

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7301183号
(P7301183)

(45)発行日 令和5年6月30日(2023.6.30)

(24)登録日 令和5年6月22日(2023.6.22)

(51)国際特許分類

F I

G 0 6 F 11/34 (2006.01)

G 0 6 F 11/34 1 5 2

請求項の数 17 (全19頁)

(21)出願番号	特願2022-15570(P2022-15570)	(73)特許権者	310021766
(22)出願日	令和4年2月3日(2022.2.3)		株式会社ソニー・インタラクティブエン
(62)分割の表示	特願2020-69190(P2020-69190)の 分割		タテインメント
原出願日	平成29年3月23日(2017.3.23)		東京都港区港南1丁目7番1号
(65)公開番号	特開2022-58878(P2022-58878A)	(74)代理人	100105924
(43)公開日	令和4年4月12日(2022.4.12)		弁理士 森下 賢樹
審査請求日	令和4年3月4日(2022.3.4)	(72)発明者	シンプソン、デイヴィッド
(31)優先権主張番号	62/315,315		アメリカ合衆国、カリフォルニア州 9
(32)優先日	平成28年3月30日(2016.3.30)		4 4 0 4、サン マテオ、ブリッジボイ
(33)優先権主張国・地域又は機関	米国(US)		ント パークウェイ 2 2 0 7、ソニー
(31)優先権主張番号	62/315,345		インタラクティブ エンタテインメント
(32)優先日	平成28年3月30日(2016.3.30)	(72)発明者	アメリカ リミテッド ライアビリティ
(33)優先権主張国・地域又は機関			カンパニー内
最終頁に続く			ツェルニー、マーク エヴァン
			アメリカ合衆国、カリフォルニア州 9
			最終頁に続く

(54)【発明の名称】 後方互換性のためのアプリケーション固有動作パラメータの導出

(57)【特許請求の範囲】

【請求項1】

レガシーアプリケーションを新規システム上で実行しているときに、動作パラメータの調整のためにレガシーアプリケーションの性能を特徴付ける方法であって、

前記レガシーアプリケーションを前記新規システムよりも古いシステムアーキテクチャを有する旧システム上で実行することと、

前記レガシーアプリケーションを前記旧システム上で実行することに基づいて1つまたは複数の性能メトリックおよび他の性能情報を取得することであって、前記1つまたは複数の性能メトリックは、前記レガシーアプリケーションが前記旧システムよりも新しいシステムアーキテクチャを有する前記新規システム上で実行されるときに満たされなければならず、前記他の性能情報は、前記新規システム上で前記レガシーアプリケーションを実行しているときに、前記新規システムの動作パラメータの調整のために使用される、前記取得することと、

前記レガシーアプリケーションが前記新規システム上で動作する際に前記新規システムの前記動作パラメータを調整することであって、前記1つまたは複数の性能メトリックが境界範囲内にない場合、前記1つまたは複数の性能メトリックが前記境界範囲内に入るまで、前記動作パラメータが繰り返し調整される、前記調整することと、

を含む、方法。

【請求項2】

前記1つまたは複数の性能メトリックは、プログラム実行中にイベント間のクロックサ

イクルをカウントすることから判定される、請求項 1 に記載の方法。

【請求項 3】

前記 1 つまたは複数の性能メトリックは、前記アプリケーションの実行に関連する情報を追跡する、1 つまたは複数の専用プロセッサレジスタ内に記憶される値から判定される、請求項 1 に記載の方法。

【請求項 4】

前記 1 つまたは複数の専用プロセッサレジスタ内に記憶される前記情報が、カウンタ値を含む、請求項 3 に記載の方法。

【請求項 5】

前記カウンタ値が、プログラムカウンタ値である、請求項 4 に記載の方法。

10

【請求項 6】

前記カウンタ値が、メモリサイクル、演算論理ユニット（ALU）サイクル、またはピクセルのためのカウンタの値である、請求項 4 に記載の方法。

【請求項 7】

前記 1 つまたは複数の性能メトリックは、検出されるビジーウェイトから判定される、請求項 1 に記載の方法。

【請求項 8】

前記 1 つまたは複数の性能メトリックが、1 秒あたりのフレーム数を含む、請求項 1 に記載の方法。

【請求項 9】

20

前記 1 つまたは複数の性能メトリックが、プログラムカウンタ（PC）レンジにビン化される 1 サイクルあたりの命令数（IPC）を含む、請求項 1 に記載の方法。

【請求項 10】

前記他の性能情報は、間接的に導出される、請求項 1 に記載の方法。

【請求項 11】

前記他の性能情報が、単位時間あたりの平均並列処理ハードウェアスケジューリングユニット占有率、平均並列処理ハードウェアスケジューリングユニット寿命、メモリ動作についての平均レイテンシ、または単位時間あたりの対象をレンダリングするためのピクセル出力のカウントを含む、請求項 9 に記載の方法。

【請求項 12】

30

前記新規システムの動作パラメータを調節するために有用である 1 つまたは複数の性能情報値を判定することをさらに含む、請求項 1 に記載の方法。

【請求項 13】

前記新規システムの動作パラメータを調節するために有用である前記 1 つまたは複数の性能情報値を判定することが、性能情報値及び動作パラメータの変化の間の 1 つまたは複数の相関関係を判定することを含む、請求項 12 に記載の方法。

【請求項 14】

前記新規システムの動作パラメータを調節するために有用である前記 1 つまたは複数の性能情報値を判定することが、性能情報値及び動作パラメータの変化の間の 1 つまたは複数の相関関係を、多変量解析を通して判定することを含む、請求項 12 に記載の方法。

40

【請求項 15】

前記旧システム及び新規システムが、ビデオゲームシステムである、請求項 1 に記載の方法。

【請求項 16】

プロセッサ、メモリ、及び前記メモリ内で具現化されるプロセッサ実行可能命令を備えるシステムであって、前記命令が、レガシーアプリケーションを新規システム上で実行しているときに、動作パラメータの調整のためにレガシーアプリケーションの性能を特徴付ける方法を実施するように構成され、前記方法が、

前記レガシーアプリケーションを前記新規システムよりも古いシステムアーキテクチャを有する旧システム上で実行することと、

50

前記レガシーアプリケーションを前記旧システム上で実行することに基づいて1つまたは複数の性能メトリックおよび他の性能情報を取得することであって、前記1つまたは複数の性能メトリックは、前記レガシーアプリケーションが前記旧システムよりも新しいシステムアーキテクチャを有する前記新規システム上で実行されるときに満たされなければならない、前記他の性能情報は、前記新規システム上で前記レガシーアプリケーションを実行しているときに、前記新規システムの動作パラメータの調整のために使用される、前記取得することと、

前記レガシーアプリケーションが前記新規システム上で動作する際に前記新規システムの前記動作パラメータを調整することであって、前記1つまたは複数の性能メトリックが境界範囲内に入らない場合、前記1つまたは複数の性能メトリックが前記境界範囲内に入るまで、前記動作パラメータが繰り返し調整される、前記調整することと、を含む、システム。

10

【請求項17】

具現化されるコンピュータ可読命令を有する非一時的コンピュータ可読媒体であって、前記命令が、レガシーアプリケーションを新規システム上で実行しているときに、動作パラメータの調整のために前記レガシーアプリケーションの性能を特徴付ける方法を実施するように構成され、前記方法が、

前記レガシーアプリケーションを前記新規システムよりも古いシステムアーキテクチャを有する旧システム上で実行することと、

前記レガシーアプリケーションを前記旧システム上で実行することに基づいて1つまたは複数の性能メトリックおよび他の性能情報を取得することであって、前記1つまたは複数の性能メトリックは、前記レガシーアプリケーションが前記旧システムよりも新しいシステムアーキテクチャを有する前記新規システム上で実行されるときに満たされなければならない、前記他の性能情報は、前記新規システム上で前記レガシーアプリケーションを実行しているときに、前記新規システムの動作パラメータの調整のために使用される、前記取得することと、

20

前記レガシーアプリケーションが前記新規システム上で動作する際に前記新規システムの前記動作パラメータを調整することであって、前記1つまたは複数の性能メトリックが境界範囲内に入らない場合、前記1つまたは複数の性能メトリックが前記境界範囲内に入るまで、前記動作パラメータが繰り返し調整される、前記調整することと、を含む、非一時的コンピュータ可読媒体。

30

【発明の詳細な説明】

【技術分野】

【0001】

[優先権の主張]

本出願は、2016年3月30日に出願された米国仮特許出願第62/315,315号の利益を主張し、その内容全体が参照によって本明細書に組み込まれる。本出願は、また、2016年3月30日に出願された米国仮特許出願第62/315,345号の利益を主張し、その内容全体が参照によって本明細書に組み込まれる。本出願は、また、2017年3月22日に出願された米国特許出願第15/466,759号の利益を主張し、その内容全体が参照によって本明細書に組み込まれる。本出願は、また、2017年3月22日に出願された米国特許出願第15/466,769号の利益を主張し、その内容全体が参照によって本明細書に組み込まれる。

40

【0002】

本開示の態様は、コンピュータシステム上でのコンピュータアプリケーションの実行に関する。特に、本開示の態様は、旧バージョンのコンピュータシステム用に設計されたアプリケーション/タイトルに後方互換性を提供するシステムまたは方法に関する。

【背景技術】

【0003】

新規のコンピュータアーキテクチャがリリースされるとき、以前のバージョンのアーキテクチャ用に書かれたアプリケーションが、新規アーキテクチャ上で完璧に動作すること

50

が望ましい。このケイパビリティは、しばしば「後方互換性」と呼ばれる。後方互換性を実装することは、対象のレガシーデバイスを新規のホストアーキテクチャ上でエミュレートすることを伴い、それによって、新規アーキテクチャは、レガシーデバイス用に書かれたプログラムの命令を実行し得る。コンピュータアーキテクチャは、経時的に変化して、バス、クロック速度、プロセッサアーキテクチャ、キャッシング、標準などにおける技術的進歩を活用する。1つのコンピュータアーキテクチャが、より新しいアーキテクチャに置き換わると、古いアーキテクチャは、いわゆるレガシーアーキテクチャになる。その開発の過程にわたって、ネットワークプロトコル、ユーザインタフェース、オーディオ処理、デバイスドライバ、グラフィック処理、メッセージ、ワードプロセッサ、スプレッドシート、データベースプログラム、ゲーム、及び他のアプリケーションなどのソフトウェアアプリケーションは、レガシーアーキテクチャ用に書かれている。このようなレガシーソフトウェアは、それらが新規アーキテクチャにアップグレードする場合であっても、依然としてそのユーザには価値がある。したがって、レガシーソフトウェアを新規アーキテクチャ上で実行することが可能である必要性が存在する。

10

【0004】

新規デバイスとレガシーデバイスとのハードウェアコンポーネントの性能の差が、新規デバイス上での同期エラーを引き起こし得る。それは、新規デバイスアーキテクチャ上で実行しているときに、レガシーアプリケーションをクラッシュさせるか、または誤った出力を生じさせ得る。このような性能の差は、例えば、新規デバイスとレガシーデバイスとの間のハードウェアアーキテクチャの差から起こり得る。

20

【発明の概要】

【0005】

このような状況の中で、本開示の態様が生じる。

【図面の簡単な説明】

【0006】

【図1A】本開示の態様による、アプリケーション固有動作パラメータの導出を示すフロー図である。

【図1B】本開示の態様による、アプリケーション固有動作パラメータのリアルタイム調整を示すフロー図である。

【図2A】本開示の態様による、後方互換モードで動作するように構成され得る中央処理装置(CPU)コアの実施例を示すブロック図である。

30

【図2B】本開示の態様による、CPUのための可能なマルチコアアーキテクチャの実施例を示すブロック図である。

【図3】本開示の態様による、後方互換モードで動作するように構成されるCPUを有するデバイスのブロック図である。

【発明を実施するための形態】

【0007】

〔序論〕

新規デバイス上でレガシーアプリケーションを実行する際に、ハードウェアの振る舞いの違いに起因して発生する問題に対処するために、新規ハードウェアは、レガシーアプリケーションを実行するように調節され得る。

40

【0008】

テスト段階の間、レガシーアプリケーションは、レガシーデバイス上でレガシーアーキテクチャを用いて実行され、性能情報が収集される。性能情報の実施例は、単位時間あたりのALU命令またはメモリ動作の数、及び平均並列処理ハードウェアスケジューリングユニット(例えば、ウェーブフロント)占有率または寿命を含む。性能情報は、レガシーデバイス上でゲーム及びアプリケーションを実行すること、ならびにカウンタを読み出すことによって、直接測定(ALU及びメモリ動作)されてもよい。代替的には、性能情報は、測定プロセスの一部として、そのようなカウンタ、または他のデータ出力を読み出すことから導出されてもよい。そのような導出の実施例として、平均ウェーブフロント占有

50

率及び寿命が、ウェーブフロントが開始及び停止するときの測定値から導出され得る。特定のアプリケーション、例えば、特定のビデオゲームについての結合された性能データが、そのアプリケーションについての性能特性とここでは呼ばれる。テスト段階においてアプリケーションについて判定される性能特性は、後方互換性を保証するために、新規システム上で同じアプリケーションを実行するための基準値として使用され得る。

【 0 0 0 9 】

新規デバイス上でのアプリケーションの性能は、新規デバイスの動作パラメータを調節することによって、レガシーデバイス上の同じアプリケーションの性能に密接に合致され得る。動作パラメータの実施例は、とりわけ、新規デバイスのクロック周波数、利用可能な汎用レジスタ（GPR）の数、命令起動レートなどを含む。アプリケーションは、アプリケーション固有性能特性を調整するようにその動作パラメータを調節しつつ、新規システム上で繰り返し実行され得る。新規システム上で十分な数のテストを行った後、新規システム上のアプリケーションの性能特性が、動作パラメータが変化するにつれてどのように収束するかを解析し得る。動作パラメータの新たなセットが、収束解析に基づいて生成され得る。このプロセスは、動作パラメータが、新規システム上のアプリケーションに対して最適に設定されるまで繰り返され得る。さらに最適化するために、アプリケーションが、それを機能しなくさせることなく新規ハードウェア上でより高速に実行され得るかどうかを見るために、新規ハードウェアの実行を調整し得る。

【 0 0 1 0 】

[アプリケーション固有性能特性の判定]

図 1 A は、アプリケーションがレガシーシステム上で動作するテスト段階中に、アプリケーション固有動作パラメータを導出する方法 1 0 を示している。アプリケーションは、レガシーシステム上で実行 1 2 され、コードブロック毎 1 4 に、性能情報が、記録または導出 1 6 される。アプリケーションを実行することは、例えば、キャプチャをロードすること及び入力無しでそれを実行させること、または特定のゲーム領域を通してプレイすることを含み得る。性能情報は、主要性能メトリック、及び他の性能情報を含む。主要性能メトリックは、アプリケーションが新規システム上で実行されるときに最も重要な性能情報のサブセットをいう。主要性能メトリックは、アプリケーションが新規ハードウェア上で実行されるときに満たされなければならないものである。主要性能メトリックの実施例は、1 秒あたりのフレーム数（例えば、ビデオゲームなどのビデオ集中型アプリケーションの場合）、及びプログラムカウンタ（PC）レンジにピン化された 1 サイクルあたりの命令数（IPC）を含むが、これらに限定されない。

【 0 0 1 1 】

他の性能情報は、PC ブロックレジデンス、単位時間あたりに発行される演算論理ユニット（ALU）命令の数（CPU 及び GPU）、単位時間あたりに発行されるメモリ動作の数（CPU 及び GPU）、単位時間あたりの平均並列処理ハードウェアスケジューリングユニット（例えば、ウェーブフロント、ワーブ、またはベクトル幅）占有率、平均並列処理ハードウェアスケジューリングユニット寿命、メモリ動作についての平均レイテンシ、単位時間あたりの対象をレンダリングするためのピクセル出力のカウント、及び、フレーム中のアクティブな総サイクル数（ALU カウントは、これの具体例）を含むが、これらに限定されない。

【 0 0 1 2 】

性能情報は、カウンタから直接読み出される値、または、プログラム実行中にイベント間のクロックサイクルをカウントするなど、そのような値もしくは他の情報から導出される値を含んでもよい。性能情報は、1 8 においてさらに解析されてもよく、選択される性能情報が、性能特性 1 9 のセットを判定するために結合されてもよく、それは、次いで、保存または転送 2 0 されてもよい。

【 0 0 1 3 】

ある性能情報値は、アプリケーションの実行に関連する情報を追跡する、専用プロセッサレジスタ内に記憶され得る。このような値の実施例は、プログラムカウンタ、ならびに

10

20

30

40

50

、中でもメモリサイクル、演算論理ユニット（ALU）サイクル、及びピクセルについてのカウンタなどのカウンタ値を含むが、これらに限定されない。プログラムカウンタ（PC）は、また、Intel x86 及び Itanium マイクロプロセッサにおける命令ポインタ（IP）とも呼ばれ、命令アドレスレジスタ（IAR）、または命令カウンタと呼ばれることもあり、コンピュータがそのプログラムシーケンス内にある場所を示す、プロセッサレジスタである。

【0014】

上述のように、単位時間あたりの平均並列処理ハードウェアスケジューリングユニット（例えば、ウェーブフロント、ワープ、もしくはベクトル幅）占有率、平均並列処理ハードウェアスケジューリングユニット寿命、メモリ動作についての平均レイテンシ、単位時間あたりの対象をレンダリングするためのピクセル出力のカウントといった、ある他の性能情報が、間接的に導出され得る。限定ではなく例として、1 サイクルあたりの命令数（IPC）は、最初のプログラムカウンタ値と最後のプログラムカウンタ値との差を、最初のプログラムカウンタ値と最後のプログラムカウンタ値との間のクロックサイクル数で割ることによって、導出され得る。また、平均並列処理ハードウェアスケジューリングユニット寿命を判定することは、そのようなスケジューリングユニットの起動及び完了を検出すること、ならびにその間のクロックサイクルをカウントすることを伴い得る。同様に、並列処理ハードウェアスケジューリングユニットの単位時間あたりの平均占有率を判定することは、所与の時間ウィンドウの間起動及び完了を記録すること、ならびにその時間ウィンドウ内の任意の所与の時間において平均で何回実行しているかを判定することの問題である。

【0015】

「並列処理スケジューリングユニット」という用語は、異なるプロセッサハードウェアの製造業者によって、コードの並列処理用の最小実行可能ユニットの概念を説明するために使用される、いくつかの異なる用語を包含する総称としてここでは使用される。例えば、GPU の文脈においては、並列処理スレッドは、スケジューリングの最も基本的なユニットとして、「ワープ」（NVIDIA ハードウェアの場合）、または「ウェーブフロント」（AMD ハードウェアの場合）と呼ばれることがあるものに束ねられる。その違いは主に、共にグループ化されるスレッドの数である。他の等価な定義は、「動作し得る最小実行可能ユニットコード」、または「その中のスレッドの全てにわたって単一命令により同時に処理するユニット」、または「SIMD 方式で処理されるデータの最小サイズ」を含む。CPU ハードウェアの場合、並列性の最も基本レベルの概念は、「ベクトル幅」（例えば、Intel 及び AMD プロセッサ上で SSE 命令を使用するとき）と呼ばれることが多い。単純化のため、「ウェーブフロント」という用語は、「並列処理スケジューリングユニット」の代用として、ここで使用されるものとする。ウェーブフロント内の全てのスレッドが、ロックステップで同一命令を実行し、違いは、その命令によって操作されるデータだけである。

【0016】

他の動作情報は、いくつかの異なる方法で動作レジスタ値から導出され得る。例えば、IPC は、プログラム実行時に実行されている命令の総数を含むカウンタをサンプリングすることによって導出されてもよい。例として、このカウンタは、N サイクル毎にサンプリングされてもよい。IPC 値は、 $(TIE_i + N - TIE_i) / N$ からの、最初の実行済み総命令数の値（ TIE_i ）及び N サイクル後のその後の値（ $TIE_i + N$ ）から導出されてもよい。実際には、アプリケーションの所与のセクション（例えば、コードのブロック）についての IPC 値が、その特定セクションのための PC レンジによってビン化されてもよい。さらに、アプリケーション内の各 PC レンジは、したがって、異なる潜在的振る舞いを有し、それに対応して異なる IPC 値を有してもよい。したがって、例えば、コードブロック番号により、IPC 値をプログラムコードの識別されるセクションと関連付けることは有用である。

【0017】

PCブロックレジデンスは、現在実行されているアプリケーションコードのブロックをいい、GPUは、典型的には、複数のコード部分を同時に実行するため、PCブロックレジデンスは、GPUよりもCPUにより関連し得る。PCブロックレジデンスは、Nサイクル毎にPCをサンプリングすること、及びサンプルがコードの同一ブロック内に入る回数をカウントすることによって導出され得る。

【0018】

ALUまたはメモリ動作の発行頻度は、そのような動作の発行を検出すること、及び所与の時間ウィンドウにわたって発行されるそのような動作の数をカウントすることによって、導出され得る。同様に、単位時間あたりの対象をレンダリングするためのピクセル出力のカウントは、所与の時間ウィンドウにわたるピクセル出力をカウントすることによって導出され得る。キャッシュレイテンシまたはメモリ動作レイテンシなどのレイテンシは、キャッシュ読み出し/書き込み及び/またはメモリアクセス命令の発行及び完了を検出すること、ならびに発行と完了との間のクロックサイクルをカウントすることによって、導出され得る。

【0019】

16において性能情報を記録すること/導出することは、ビジーウェイトを検出することを含み得る。ビジーウェイトは、典型的には、短いループとして実装される。カウンタの観点からは、これは、PCが非常に小さな範囲内にとどまっている（かつ繰り返している）ように見え、ループを通して毎回起こる、何らかの種類のメモリ読み出しまたはIO読み出し動作が存在することとなる。IPCは、ループのために大きくなり得ると考えられるが、より実際的には、メモリまたはIO動作の結果が返ってくるのを待機することによってループ内の時間が左右されることになるため、IPCは小さい可能性が高い。ビジーウェイトは、PCが非常に小さな範囲内にとどまる時間を探すことによって検出されてもよく、その時間は、メモリまたはIO動作が完了するのを待機することによって左右される。ビジーウェイトは、IPC及び他の性能情報測定値にずれを生じさせる傾向がある。ビジーウェイトに費やす時間が予測不可能であるため、ビジーウェイトの間に取られる測定値は、18において性能特性を判定するプロセスの一部として、性能情報から除去されてもよい。そうすることによって、動作パラメータを調整する後続のプロセスが、ビジーウェイトの存在によって影響を受けないこととなる。

【0020】

18において、記録または導出された性能情報を解析することは、概して、実行中のアプリケーションの振る舞いを大まかに特徴付ける性能特性19の有用なセットに、性能情報を絞り込むことを伴う。性能特性19は、以下で説明するように、1つまたは複数の主要性能メトリック、及び後の動作パラメータの判定に有用な他の性能情報を含むが、これらに限定されない。

【0021】

性能特性判定段階18は、例えば、多くの異なる性能情報値が、所与の動作パラメータの変化に応じて変化し得るときに、主要性能情報値及び動作パラメータの変化の間の相関関係を、多変量解析を通して判定することによって、動作パラメータを調節するためにどの性能情報値が有用であるかを判定し得る。

【0022】

[パラメータ調整プロセス]

図1Bは、アプリケーションが新規システム上で動作する際に動作パラメータを最適化するように、レガシーアプリケーションの実行を調整する方法30を説明する。新規システムは、性能特性19を使用して、レガシーアプリケーションの実行(32)時にリアルタイムで1つまたは複数の動作パラメータを調整し得る。各コードブロック34について、性能情報が、例えば上述したように導出36される。38において、1つまたは複数の主要性能メトリックが境界範囲内にない場合、それらが境界範囲内に入るまで、1つまたは複数の動作パラメータが繰り返し調整40され得る。38において、主要性能メトリックが、一旦境界範囲内に入れば、動作パラメータは、更新42され、さらなる調整40に

10

20

30

40

50

よって最適化され得る。更新／最適化された動作パラメータデータ 43 は、保存または転送 44 され得る。

【0023】

「動作パラメータ」という用語は、概して、主要性能メトリックを含む性能情報に影響を及ぼすように調整され得る、新規システム上でのアプリケーション実行の態様をいう。動作パラメータの実施例は、例えば、CPU、GPU、またはメモリのためのクロック周波数、命令の起動レート、ALU及び／またはメモリ動作の起動レート、例えば、汎用レジスタ（GPR）、ウェーブフロントスロット、読み出し及びストアキューサイズなどのリソース、特徴無効化、キャッシュパラメータ（例えば、キャッシュサイズ、ウェイの数、バンクの数、など）、ウェーブフロント起動レート、レンダーバックエンドからのピクセル出力レート、メモリ動作ストールを含んでもよいが、これらに限定されない。

10

【0024】

アルゴリズムマッチングは、新規システムアーキテクチャ用に書かれた新規かつ改善されたアルゴリズムの代わりに、レガシーシステムアーキテクチャからのアルゴリズムを使用して、新規システム上においてある動作を実行することをいう。このようなアルゴリズムマッチングの実施例が、レガシーシステム用の分岐予測器を使用して、新規システム上で分岐予測を実行することである。この例では、アルゴリズムマッチングパラメータは、レガシーアルゴリズムにおいて使用されるパラメータを含む。

【0025】

他の動作パラメータは、リソース制限に関連するパラメータ（例えば、参照により本明細書に組み込まれる、2015年7月27日に出願された米国特許出願第14/810,361号に記載されている）、ならびにアルゴリズムマッチング、特徴無効化、及びレイテンシまたはスループットのマッチングに関連するパラメータ（例えば、参照により本明細書に組み込まれる、2015年7月27日に出願された米国特許出願第14/810,334号に記載されている）も含み得る。

20

【0026】

40における動作パラメータの調整は、単純であってもよく、例えば、新規ハードウェア上の汎用レジスタ（GPR）の数を、レガシーハードウェアと同一の数に設定することであってもよい。代替的には、新規ハードウェアが、ある動作のためのレガシーアルゴリズムを使用してもよく、または、新規ハードウェアの特徴が、レガシーアプリケーションの動作のために無効化されてもよい。実行は、新規システム上のレイテンシをレガシーハードウェアのレイテンシに合致させるように調整されてもよい。

30

【0027】

動作パラメータの調整は、レガシーハードウェアと新規ハードウェアのアーキテクチャの差に起因して、より複雑であり得る。いくつかの場合において、例えば、わずかに大きな数のGPRを設定することによって、元のハードウェアよりも多くのリソースが、新規ハードウェア上に割り当てられ得る。

【0028】

下記の表Iは、アプリケーション固有動作パラメータ、それらの導出方法、及びそれらの調整方法の非限定的な例をリストにしたものである。

40

【0029】

【表 1】

表 I

性能情報	測定／導出方法	動作パラメータを調整することによる変更方法
1 サイクルあたりの命令数 (IPC)	フレームの間カウンタを用いて、発行される全命令を測定する (CPU 及び GPU)	命令起動レートを調整する (HW において行う必要がある)
ALU 動作の発行頻度	フレームの間カウンタを用いて ALU サイクルを測定する (CPU 及び GPU)	N サイクル毎に ALU 動作を許可しない (HW において行う必要がある)
メモリ動作の発行頻度	フレームの間カウンタを用いてメモリサイクルを測定する (CPU 及び GPU)	N サイクル毎にメモリ動作を許可しない (HW において行う必要がある)
単位時間あたりの平均ウェーブフロント占有率	カウンタでこれをサンプリングする、またはキャプチャを再生、ならびにウェーブフロントがいつ開始及び停止するかを見る (GPU)	選択的に GPR を割り当てる (SW において行い得る) か、またはウェーブフロント起動レートを抑制する (HW において行う必要がある)
平均ウェーブフロント寿命	キャプチャを再生、ならびにウェーブフロントがいつ開始及び停止するかを見る (GPU)	選択的に GPR を割り当てる (SW において行い得る) か、またはウェーブフロント起動レートを抑制する (HW において行う必要がある)
単位時間当たりの対象をレンダリングするためのピクセル出力	既存のカウンタで、単位時間あたりのピクセルカウントを見る (GPU)	レンダーバックエンド (これらはグラフィックパイプラインの下流において対象をレンダリングするためにピクセルを書き出す) からの出力レート (#ピクセル) を抑制する (HW において行う必要がある)
平均メモリ動作レイテンシ	メモリ命令が発行されるとき、及びそれが実行されるときを判定し、その間のクロックサイクルをカウントする (CPU 及び GPU)	メモリ動作を終了からストールする (HW において行う必要がある) か、または異なるレートでクロックを作動させる (SW において行い得る)
PC ブロックレジデンス	プログラムカウンタを読み出す (CPU)	動作パラメータ値が、ブロックレジデンスに強く相関する場合に、動作パラメータを調整する際の有用な情報

【0030】

結果となる更新された動作パラメータ 43 は、アプリケーション固有性能情報の各項目についての下限を含み得る。それより上では、レガシーハードウェア上で実行しているレガシーアプリケーションの性能メトリックが、一貫して満たされる。新規ハードウェア上でさらにテストすることで、アプリケーション固有性能情報の各項目は、上限をさらに含み得る。それより上では、レガシーアプリケーションは、もはや適切に機能しないか、または、レガシーアプリケーションの主要性能メトリックが、新規システム上ではもはや満たされない。アプリケーション固有性能情報は、下記の表 I I における情報に対応し得る。

【0031】

10

20

30

40

50

【表 2】

表 I I

コードブ ロック	性能情報	最小値	最大値
CB 1	1サイクルあたりの命令数 (IPC)	IPC1 _{最小値}	IPC1 _{最大値}
CB 1	ALU動作の発行頻度	AOF1 _{最小値}	AOF1 _{最大値}
CB 1	メモリ動作の発行頻度	MOF1 _{最小値}	MOF1 _{最大値}
CB 1	単位時間あたりの平均ウェーブフロン ト占有率	AWO1 _{最小値}	AWO1 _{最大値}
CB 1	単位時間当たりの対象をレンダリング するためのピクセル出力	PORT1 _{最小値}	PORT1 _{最大値}
CB 1	平均メモリ動作レイテンシ	AML1 _{最小値}	AML1 _{最大値}
CB 1	PCブロックレジデンス	PB1 _{最小値}	PB1 _{最大値}
CB 2	1サイクルあたりの命令数 (IPC)	IPC2 _{最小値}	IPC2 _{最大値}
CB 2	ALU動作の発行頻度	AOF2 _{最小値}	AOF2 _{最大値}
CB 2	メモリ動作の発行頻度	MOF2 _{最小値}	MOF2 _{最大値}
CB 2	単位時間あたりの平均ウェーブフロン ト占有率	AWO2 _{最小値}	AWO2 _{最大値}
CB 2	単位時間当たりの対象をレンダリング するためのピクセル出力	PORT2 _{最小値}	PORT2 _{最大値}
CB 2	平均メモリ動作レイテンシ	AML2 _{最小値}	AML2 _{最大値}
CB 2	PCブロックレジデンス	PB2 _{最小値}	PB2 _{最大値}

【0032】

表 I I に示される実施例には、レガシープログラムにおける各コードブロックについての性能情報のための上限及び下限のセットがある。この情報は、新規ハードウェア上でのレガシーゲームの後続の動作において使用され得る。そのような後続の動作は、新規ハードウェアが最小値と最大値との間で性能情報を維持するように実行を調整して、図 1 B に記載されるように進行し得る。

【0033】

動作パラメータの導出及び調整は、図 2 A ~ 2 B 及び図 3 に示されるハードウェアの特徴に関連し得る。図 2 A は、CPU コア 100 の一般化されたアーキテクチャを示す。CPU コア 100 は、典型的には、分岐が取られるか否かの予測を試み、かつ（分岐が取られる場合に）分岐の行先アドレスの予測も試みる、分岐予測ユニット 102 を含む。これらの予測が正確である限り、投機的に実行されたコードの効率が上昇することとなり、したがって、より正確性の高い分岐予測が極めて望ましい。分岐予測ユニット 102 は、サブルーチンからの復帰アドレスを追跡する復帰アドレススタック 104、間接分岐の行先を追跡する間接ターゲットアレイ 106、ならびにそれらの結果となるアドレスをより正確に予測するために分岐の過去の履歴を追跡する分岐ターゲットバッファ 108 及びその関連予測ロジックなどの、高度に専門化されたサブユニットを含み得る。

【0034】

CPU コア 100 は、典型的には、命令フェッチ及び復号ユニット 110 を含み、命令フェッチ及び復号ユニット 110 は、命令フェッチユニット 112、命令バイトバッファ 114、及び命令復号ユニット 116 を含む。CPU コア 100 は、典型的には、いくつかの命令関連キャッシュ及び命令変換索引バッファ (ITLB) 120 も含む。これらは

、ページテーブルエントリ、ページディレクトリエントリなどの物理アドレス変換情報に仮想アドレスをキャッシュするITLBキャッシュ階層124を含み得る。この情報は、命令の仮想アドレスを物理アドレスへと変換するために使用され、それによって、命令フェッチユニット112が、キャッシュ階層から命令をロードすることができる。限定ではなく例として、プログラム命令は、コア内に存在するレベル1命令キャッシュ(L1 Iキャッシュ)122、及びCPUコア100外部の他のキャッシュレベル176を含むキャッシュ階層に従ってキャッシュされてもよい。命令の物理アドレスを使用して、これらのキャッシュは、まずプログラム命令について検索される。命令が見つからない場合、それらは、システムメモリ101からロードされる。アーキテクチャによっては、後述するように、復号済み命令を含むマイクロオペキャッシュ126も存在し得る。

10

【0035】

プログラム命令が一旦フェッチされると、それらは、典型的には、命令フェッチ及び復号ユニット110による処理を待機する命令バイトバッファ114内に配置される。復号は、非常に複雑なプロセスであり得る。各サイクルで複数の命令を復号することは困難であり、1サイクル内に復号され得る命令の数を制限する、命令アライメントまたは命令の種類についての制限が存在し得る。復号済み命令は、アーキテクチャによっては、マイクロオペキャッシュ126内に配置されてもよく(1つは、新規CPU上に存在する場合)、したがって、復号段階は、プログラム命令の後続の使用のためにバイパスされ得る。

【0036】

復号済み命令は、典型的には、ディスパッチ及びスケジューリング130のための他のユニットに渡される。これらのユニットは、リタイアキュー132を使用して、CPUパイプラインのリマインダ全体を通して命令の状態を追跡し得る。さらに、多くのCPUアーキテクチャにおいて利用可能な汎用及びSIMDレジスタの数が制限されることに起因して、レジスタリネーミングが実行されてもよく、そこでは、論理(アーキテクチャともいう)レジスタが、実行されている命令のストリームにおいて遭遇されるため、物理レジスタ140が、それらを表すために割り当てられる。物理レジスタ140は、単一命令複数データ(SIMD)レジスタバンク142及び汎用(GP)レジスタバンク144を含んでもよく、それは、特定のCPUアーキテクチャ上で利用可能な論理レジスタの数よりもかなり大きなサイズであってもよく、その結果、性能が相当に向上され得る。レジスタリネーミング134が実行された後、命令は、典型的には、スケジューリングキュー136に配置され、スケジューリングキュー136から、いくつかの命令が、実行ユニット150による実行のために(依存性に基づいて)各サイクルで選択され得る。

20

30

【0037】

実行ユニット150は、典型的には、SIMDレジスタバンク142に含まれる128ビットまたはそれ以上のSIMDレジスタに含まれる複数のデータフィールド上でいくつかの並列演算を実行するSIMDパイプ152と、GPレジスタバンク144に含まれるGPR上でいくつかの論理、算術、及び種々の演算を実行する演算論理ユニット(ALU)154と、メモリが記憶されまたはロードされるべきアドレスを計算するアドレス生成ユニット(AGU)156と、を含む。実行ユニットの各種類の複数のインスタンスが存在してもよく、インスタンスは、異なるレイバリティを有してもよく、例えば、特定のSIMDパイプ152は、浮動小数点加算動作ではなく浮動小数点乗算動作を実行することが可能であってもよい。

40

【0038】

ストア及びロードは、典型的には、ストアキュー162及びロードキュー164にバッファされ、それによって、多くのメモリ動作が、並列で実行され得る。メモリ動作を支援するために、CPUコア100は、通常いくつかのデータ関連キャッシュ及びデータ変換索引バッファ(DTLB)170を含む。DTLBキャッシュ階層172は、仮想アドレスを、ページテーブルエントリ、ページディレクトリエントリなどの物理アドレス変換にキャッシュする。この情報は、メモリ動作の仮想アドレスを物理アドレスに変換するために使用され、それによって、データはシステムメモリに記憶され、またはシステムメモリ

50

からロードされ得る。データは、典型的には、コア内に存在するレベル 1 データキャッシュ (L1 D キャッシュ) 174、及びコア 100 の外部にある他のキャッシュレベル 176 にキャッシュされる。

【0039】

本開示のある態様によれば、CPU は、複数のコアを含んでもよい。限定ではなく例として、図 2B は、本開示の態様と併せて使用され得る、可能なマルチコア CPU 200 の例を示す。具体的には、CPU 200 のアーキテクチャは、M 個のクラスタ 201 - 1 ... 201 - M を含んでもよく、M は、0 より大きい整数である。各クラスタは、N 個のコア 202 - 1、202 - 2 ... 202 - N を有してもよく、N は、1 より大きい整数である。本開示の態様は、異なるクラスタが異なる数のコアを有する実施態様を含む。各コアは、1 つまたは複数の対応する専用ローカルキャッシュ (例えば、L1 命令、L1 データ、または L2 キャッシュ) を含み得る。ローカルキャッシュのそれぞれは、いかなる他のコアとも共有されないという意味で、特定の対応するコア専用であり得る。各クラスタは、また、対応するクラスタ内のコア間で共有され得るクラスタレベルキャッシュ 203 - 1 ... 203 - M を含み得る。いくつかの実施態様では、クラスタレベルキャッシュは、異なるキャッシュに関連付けられたコアによって共有されない。さらに、CPU 200 は、1 つまたは複数の上位レベルキャッシュ 204 を含んでもよく、上位レベルキャッシュ 204 は、クラスタ間で共有され得る。クラスタ内のコア間の通信を容易にするために、クラスタ 201 - 1 ... 201 - M は、コアのそれぞれ及びクラスタ用のクラスタレベルキャッシュに連結される、対応するローカルバス 205 - 1 ... 205 - M を含み得る。同様に、クラスタ間の通信を容易にするために、CPU 200 は、クラスタ 201 - 1 ... 201 - M、及び上位レベルキャッシュ 204 に連結される、1 つまたは複数の上位レベルバス 206 を含み得る。いくつかの実施態様では、上位レベルバス 206 は、また、他のデバイス、例えば、GPU、メモリ、またはメモリコントローラに連結され得る。さらに他の実施態様では、上位レベルバス 206 は、システム内の様々なデバイスに接続されるデバイスレベルバスに接続され得る。さらに他の実施態様では、上位レベルバス 206 は、クラスタ 201 - 1 ... 201 - M を上位レベルキャッシュ 204 に連結し得る。デバイスレベルバス 208 は、上位レベルキャッシュ 204 を他のデバイス、例えば、GPU、メモリ、またはメモリコントローラに連結し得る。限定ではなく例として、このようなデバイスレベルバス 208 を有する実施態様は、例えば、上位レベルキャッシュ 204 が、全 CPU コア用の L3 であるが GPU 使用のためではない場合に生じ得る。

【0040】

CPU 200 において、OS 処理は、主にあるコア、またはコアのあるサブセット上で発生し得る。同様に、アプリケーションレベルの処理は、主に特定のコア、またはコアのサブセット上で発生し得る。個々のアプリケーションスレッドは、あるコア、またはコアのあるサブセット上で動作するようにアプリケーションによって指定され得る。キャッシュ及びバスが共有されているため、所与のアプリケーションスレッドによる処理速度は、所与のアプリケーションスレッドと同一のクラスタ内で動作している他のスレッド (例えば、アプリケーションスレッド、または OS スレッド) によって発生している処理に依存して変化し得る。CPU 200 の詳細に依存して、コアは、一度にただ 1 つのスレッドのみを実行することが可能であってもよく、または複数のスレッドを同時に実行すること (「ハイパースレッディング」) が可能であってもよい。ハイパースレッド型 CPU の場合、アプリケーションは、また、どのスレッドが、他のどのスレッドと同時に実行され得るかを指定し得る。スレッドの性能は、同一コアによって実行されているいずれかの他のスレッドによって実行される、特定の処理によって影響を受ける。

【0041】

ここで図 3 を参照すると、本開示の態様に従って動作するように構成されるデバイス 300 の例示的な例が示されている。本開示の態様によれば、デバイス 300 は、組み込み型システム、モバイルフォン、パーソナルコンピュータ、タブレットコンピュータ、ポータブルゲームデバイス、ワークステーション、ゲームコンソールなどであり得る。

【 0 0 4 2 】

デバイス 3 0 0 は、概して、図 1 に示し上述した種類の 1 つまたは複数の CPU コア 3 2 3 を含み得る中央処理装置 (CPU) 3 2 0 を含む。CPU 3 2 0 は、図 2 の CPU 2 0 0 に示したような構成で、複数のそのようなコア 3 2 3、及び 1 つまたは複数のキャッシュ 3 2 5 を含み得る。限定ではなく例として、CPU 3 2 0 は、単一チップ上に CPU 3 2 0 及びグラフィック処理ユニット (GPU) 3 3 0 を含む高速処理ユニット (APU) 3 1 0 の一部であってもよい。代替的な実施態様では、CPU 3 2 0 及び GPU 3 3 0 は、別々のチップ上の別々のハードウェアコンポーネントとして実装されてもよい。GPU 3 3 0 は、また、2 つ以上のコア 3 3 2 及び 2 つ以上のキャッシュ 3 3 4、ならびに(いくつかの実施態様では) コア及びキャッシュ及びシステムの他のコンポーネント間の通信を容易にするための 1 つまたは複数のバスを含み得る。バスは、APU 3 1 0 用の内部バス 3 1 7、及び外部データバス 3 9 0 を含み得る。

10

【 0 0 4 3 】

デバイス 3 0 0 は、また、メモリ 3 4 0 を含み得る。メモリ 3 4 0 は、任意で、CPU 3 2 0 及び GPU 3 3 0 にアクセス可能なメインメモリユニットを含み得る。CPU 3 2 0 及び GPU 3 3 0 は、それぞれ 1 つまたは複数のプロセッサコア、例えば、単一コア、2 つのコア、4 つのコア、8 つのコア、またはそれ以上を含み得る。CPU 3 2 0 及び GPU 3 3 0 は、外部データバス 3 9 0 を使用して 1 つまたは複数のメモリユニットにアクセスするように構成されてもよく、いくつかの実施態様では、2 つ以上の異なるバスを含むことがデバイス 3 0 0 には有用であり得る。

20

【 0 0 4 4 】

メモリ 3 4 0 は、アドレス可能なメモリ、例えば、RAM、DRAMなどを提供する集積回路の形態の 1 つまたは複数のメモリユニットを含み得る。メモリは、レガシー CPU 上の実行用に当初作成されたアプリケーションを実行する際、デバイス 3 0 0 をタイミングテストモードで動作させるように判断するための実行時に、図 5 の方法のような方法を実施するように構成される実行可能命令を含み得る。さらに、メモリ 3 4 0 は、グラフィックリソースを一時的に記憶するための専用グラフィックメモリと、グラフィックバッファと、グラフィックレンダリングパイプライン用の他のグラフィックデータと、を含み得る。

【 0 0 4 5 】

CPU 3 2 0 は、CPU コードを実行するように構成されてもよく、CPU コードは、オペレーティングシステム (OS) 3 2 1、またはアプリケーション 3 2 2 (例えば、ビデオゲーム) を含み得る。オペレーティングシステムは、ソフトウェア (例えば、アプリケーション 3 2 2) からの入力 / 出力 (I / O) 要求を管理し、それらを CPU 3 2 0、GPU 3 3 0、またはデバイス 3 0 0 の他のコンポーネントのためのデータ処理命令に変換するカーネルを含み得る。OS 3 2 1 は、また、ファームウェアを含んでもよく、ファームウェアは、不揮発性メモリに記憶され得る。OS 3 2 1 は、以下で詳細に説明するように、CPU 3 2 0 をタイミングテストモードで動作させるある特徴を実装するように構成され得る。CPU コードは、アプリケーション 3 2 2 の状態に基づいて、GPU 3 3 0 によって実装されるプログラムへのドローコマンドまたはドローコールを発行するためのグラフィックアプリケーションプログラミングインタフェース (API) 3 2 4 を含み得る。CPU コードは、また、物理的シミュレーション及び他の機能を実装し得る。OS 3 2 1、アプリケーション 3 2 2、または API 3 2 4 のうちの 1 つまたは複数のためのコードの一部は、メモリ 3 4 0、CPU 内部もしくは外部のキャッシュ、または CPU 3 2 0 にアクセス可能な大容量記憶デバイスに記憶され得る。

30

40

【 0 0 4 6 】

デバイス 3 0 0 は、メモリコントローラ 3 1 5 を含み得る。メモリコントローラ 3 1 5 は、メモリ 3 4 0 に向かう、またはメモリ 3 4 0 から来るデータの流れを管理するデジタル回路であってもよい。限定ではなく例として、メモリコントローラは、図 3 に示される例のように、APU 3 1 0 の不可欠な部分であってもよく、または別々のハードウェアコ

50

ンポーネントであってもよい。

【 0 0 4 7 】

デバイス 3 0 0 は、また、周知のサポート機能 3 5 0 を含んでもよく、サポート機能 3 5 0 は、例えばバス 3 9 0 を介して、システムの他のコンポーネントと通信し得る。このようなサポート機能は、入力 / 出力 (I / O) 素子 3 5 2 と、C P U 3 2 0、G P U 3 3 0、及びメモリ 3 4 0 それぞれのための別々のクロックを含み得る 1 つまたは複数のクロック 3 5 6 と、C P U 3 2 0 及び G P U 3 3 0 の外部にあり得る 1 つまたは複数レベルのキャッシュ 3 5 8 とを含んでもよいが、これらに限定されない。デバイス 3 0 0 は、任意で、プログラム及び / またはデータを記憶するための、ディスクドライブ、C D - R O M ドライブ、フラッシュメモリ、テープドライブ、ブルーレイドライブなどの大容量記憶デバイス 3 6 0 を含んでもよい。1 つの例では、大容量記憶デバイス 3 6 0 は、レガシー C P U を有するシステム上で動作するように当初設計されたレガシーアプリケーションを含む、コンピュータ可読媒体 3 6 2 を受け入れ得る。代替的には、レガシーアプリケーション 3 6 2 (またはその一部) は、メモリ 3 4 0 に記憶されてもよく、またはキャッシュ 3 5 8 に一部記憶されてもよい。

10

【 0 0 4 8 】

デバイス 3 0 0 は、また、G P U 3 3 0 によって準備される、レンダリングされたグラフィック 3 8 2 をユーザに提示するためのディスプレイユニット 3 8 0 を含み得る。デバイス 3 0 0 は、また、システム 1 0 0 及びユーザ間のインタラクションを容易にするためのユーザインタフェースユニット 3 7 0 を含み得る。ディスプレイユニット 3 8 0 は、フラットパネルディスプレイ、陰極線管 (C R T) スクリーン、タッチスクリーン、ヘッドマウントディスプレイ (H M D)、またはテキスト、数字、グラフィカルシンボル、もしくは画像を表示可能な他のデバイスの形態であってもよい。ディスプレイ 3 8 0 は、ここで説明する多様な技術に従って処理されるレンダリングされたグラフィック 3 8 2 を表示し得る。ユーザインタフェース 3 7 0 は、キーボード、マウス、ジョイスティック、ライトペン、ゲームコントローラ、タッチスクリーン、及び / またはグラフィカルユーザインタフェース (G U I) と併せて使用され得る他のデバイスなどの、1 つまたは複数の周辺機器を含み得る。ある実施態様では、アプリケーション 3 2 2 の状態及びグラフィックの元となるコンテンツは、例えば、アプリケーション 3 2 2 が、ビデオゲームまたは他のグラフィック集中型アプリケーションを含む場合の、ユーザインタフェース 3 7 0 を通じたユーザ入力に少なくとも一部によって判断されてもよい。

20

30

【 0 0 4 9 】

デバイス 3 0 0 は、また、デバイスがネットワークを介して他のデバイスと通信することを可能にするためのネットワークインタフェース 3 7 2 を含み得る。ネットワークは、例えば、ローカルエリアネットワーク (L A N)、インターネットなどの広域ネットワーク、B l u e t o o t h (登録商標) ネットワークなどのパーソナルエリアネットワーク、または他の種類のネットワークであってもよい。図示及び説明したコンポーネントのうちの様々なものが、ハードウェア、ソフトウェア、もしくはファームウェア、またはそれらのうち 2 つ以上の何らかの組み合わせで実装されてもよい。

【 0 0 5 0 】

下記の表 I I I は、図 2 A、図 2 B、及び図 3 に関して上述した特定ハードウェア要素が、性能情報及び対応する、調整する動作パラメータを判定するために、どのように使用され得るかの、いくつかの非限定的な例をリストにする。

40

【 0 0 5 1 】

【表 3】

表 I I I

性能情報	記録／導出方法	調整する動作パラメータ
1 サイクルあたりの命令数 (IPC)	実行された命令のカウンタを読み出す (図 2 A、リタイアキュー 1 3 2 の一部)	命令起動レートを調整する (図 2 A、CPU のためのスケジューリングキュー 1 3 6 の一部、及び図 3、GPU のための GPU コア 3 3 2 の一部)
ALU 動作の発行頻度	ALU カウンタを読み出す (図 2 A、CPU のための ALU 1 5 4 及び SIMD パイプ 1 5 2 の一部、ならびに図 3、GPU のための GPU コア 3 3 2 の一部)	N サイクル毎に ALU 動作を許可しない (図 2 A、CPU のためのスケジューリングキュー 1 3 6 の一部、及び図 3、GPU のための GPU コア 3 3 2 の一部)
メモリ動作の発行頻度	メモリ動作カウンタ (複数可) を読み出す (図 2 A、CPU のための AGU 1 5 6 の一部、及び図 3、GPU のための GPU コア 3 3 2 の一部)	N サイクル毎にメモリ動作を許可しない (図 2 A、CPU のためのスケジューリングキュー 1 3 6 の一部、及び図 3、GPU のための GPU コア 3 3 2 の一部)
単位時間あたりの平均ウェーブフロント占有率	GPU コアによって生成されるウェーブフロントの開始及び完了イベントを記録する (図 3、GPU コア 3 3 2 の一部)	選択的に GPR (図 3、GPR 3 3 6) を割り当てるか、またはウェーブフロント起動レートを抑制する (図 3、GPU コア 3 3 2 の一部)
平均ウェーブフロント寿命	GPU コアによって生成されるウェーブフロントの開始及び完了イベントを記録する (図 3、GPU コア 3 3 2 の一部) またはウェーブフロント寿命カウンタコアを読み出す (図 3、GPU コア 3 3 2 の一部)	選択的に GPR (図 3、GPR 3 3 6) を割り当てるか、またはウェーブフロント起動レートを抑制する (図 3、GPU コア 3 3 2 の一部)
単位時間当たりの対象をレンダリングするためのピクセル出力	ピクセルカウンタを読み出す (図 3、GPU コア 3 3 2 の一部)	レンダリアクティブエンドからの出力レート (#ピクセル) を抑制する (図 3、GPU コア 3 3 2 の一部)
平均メモリ動作レイテンシ	未処理のメモリ動作の長さをトラッキングする (図 2 A、CPU のためのストアキュー 1 6 2 及びロードキュー 1 6 4 の一部、ならびに図 3、GPU のための GPU コア 3 3 2 の一部)	メモリ動作を終了からストールする (図 2 A、CPU のためのストアキュー 1 6 2 及びロードキュー 1 6 4 の一部、ならびに図 3、GPU のための GPU コア 3 3 2 の一部) か、または異なるレートでクロックを動作させる (図 3、クロック 3 5 6)
分岐予測の成功及び失敗	分岐予測の成功及び失敗のカウンタを読み出す (図 2 A、分岐予測 1 0 2 の一部)	レガシー分岐予測アルゴリズムをマッチする (図 2 A、分岐予測 1 0 2 の一部)、または異なるレートでクロックを動作させる (図 3、クロック 3 5 6)
PC ブロックレジデンス	プログラムカウンタを読み出す (図 2 A、フェッチ及び復号ユニット 1 1 0 の一部)	直接調整する動作パラメータはないが、動作パラメータ値が、ブロックレジデンスと強く相関する場合、有用な情報である。

【0 0 5 2】

新規デバイス上でレガシーアプリケーションを実行しているときに記録または導出される性能情報が、レガシーデバイス上でレガシーアプリケーションを実行するための対応する性能情報を満たすか、または超えることが可能ではない場合があることに留意する。例えば、新規デバイス上の平均メモリレイテンシは、動作パラメータがどのように調整されるかに関わらず、レガシーデバイス上で測定される平均メモリレイテンシよりも高い可能性がある。新規システムについてのこの種の性能情報を知ることは、動作パラメータを調整する際に有用であり得るが、レガシー及び新規デバイス上で実行するアプリケーションの性能特性を比較する際に使用されるべきではない。1 秒あたりのフレーム数 (FPS)

及び1サイクルあたりの命令数（IPC）に限定されないが、それらのような主要性能メトリックだけが、実際に使用されるべきである。

【0053】

レガシーデバイス上で実行される同じアプリケーションの主要性能メトリックを満たすために、アプリケーションが新規デバイス上で動作する際に動作パラメータを調整するプロセスは、ビデオゲームを含む以下の実施例から理解され得る。まず、性能データは、その主要性能メトリックを判定するために、レガシーデバイス上で実行中のゲームについて収集される。次に、ゲームは、新規デバイスの動作パラメータを調整する間、新規デバイス上で実行される。新規デバイスの性能は、レガシーデバイス上でゲームを実行したときに行われたのと同じ性能情報を新規デバイス上で収集することによって、かつ、次いで、それら2つのデバイス上で実行中のゲームの主要性能メトリックを比較することによって、測定され得る。新規デバイスについての性能情報が、レガシーデバイスからの性能データに完全に合致することは、望ましいことであり得るが、これは、実際には可能でない場合がある。新規デバイス上の性能情報が、レガシーデバイス上の性能情報に可能な限り近接して合致することで十分である。しかしながら、新規デバイス上の主要性能メトリックが、レガシーデバイス上の主要性能メトリックよりも悪いことは容認できず、クラッシュし（典型的には、同期の問題に起因する）、または誤った出力を生成する（同じ理由で）アプリケーションまたはゲームも同様である。

【0054】

ここで实际的に言うと、ゲームが新規デバイス上で実行される最初の数回、動作パラメータは、レガシーデバイス上と同一であるように設定されると考えられる。ゲームが十分に新規デバイス上で実行され、その動作パラメータが調節されたら、そのエクスペリエンス及びデータが、さらなるゲームに使用され得るヒューリスティックを構築するために使用され得る。ヒューリスティックは、新規デバイス上の動作パラメータの初期値をゲームの性能特性に基づいて設定するために使用され得る。次いで、ゲームは、新規デバイス上で実行されることとなり、動作パラメータは、主要性能メトリックにさらによく合致するように修正され得る。新規デバイス上で測定される全ての性能データが、主要性能メトリックだけではなく、動作パラメータを調整するのを助けるために使用され得る。動作パラメータに対して行われるいかなる調整も、ヒューリスティックをさらに洗練するためにも使用され得る。

10

20

30

40

50

【 図 1 A 】



30



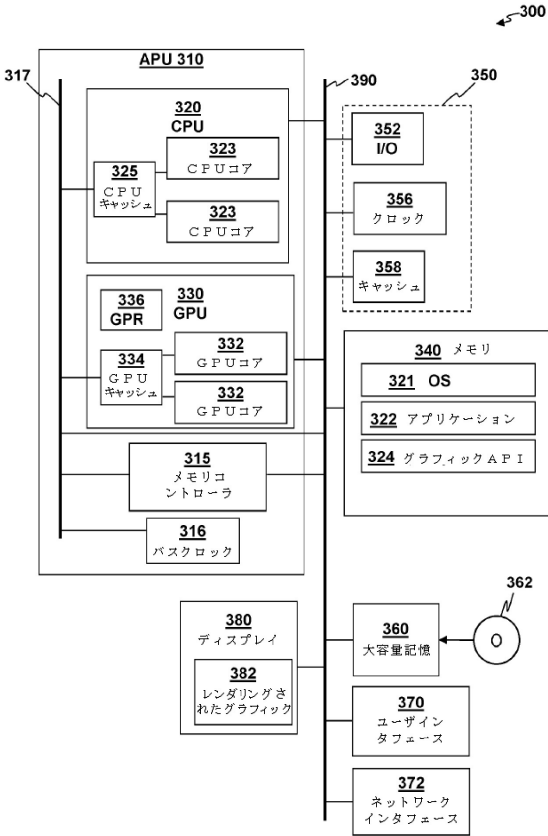
20

Page 10



40

【図 3】



10

20

30

40

50

フロントページの続き

米国(US)

(31)優先権主張番号 15/466,759

(32)優先日 平成29年3月22日(2017.3.22)

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 15/466,769

(32)優先日 平成29年3月22日(2017.3.22)

(33)優先権主張国・地域又は機関

米国(US)

4404、サン マテオ、ブリッジポイント パークウェイ 2207、ソニー インタラクティブ
エンタテインメント アメリカ リミテッド ライアビリテイ カンパニー内

審査官 三坂 敏夫

(56)参考文献 特開2014-085809(JP,A)

特開2011-258058(JP,A)

特開2007-299212(JP,A)

米国特許出願公開第2014/0351412(US,A1)

米国特許第08423976(US,B2)

(58)調査した分野 (Int.Cl., DB名)

G06F 11/07

11/28-11/36