

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5718378号
(P5718378)

(45) 発行日 平成27年5月13日(2015.5.13)

(24) 登録日 平成27年3月27日(2015.3.27)

(51) Int.Cl.

F 1

G06F 9/50 (2006.01)
G06F 9/48 (2006.01)G06F 9/46 465D
G06F 9/46 465E
G06F 9/46 457
G06F 9/46 452C

請求項の数 9 外国語出願 (全 18 頁)

(21) 出願番号

特願2013-82(P2013-82)

(22) 出願日

平成25年1月4日(2013.1.4)

(65) 公開番号

特開2013-140593(P2013-140593A)

(43) 公開日

平成25年7月18日(2013.7.18)

審査請求日

平成25年10月30日(2013.10.30)

(31) 優先権主張番号

13/344,474

(32) 優先日

平成24年1月5日(2012.1.5)

(33) 優先権主張国

米国(US)

(73) 特許権者

500520743

ザ・ボーイング・カンパニー

The Boeing Company

アメリカ合衆国、60606-2016

イリノイ州、シカゴ、ノース・リバーサイ

ド・プラザ、100

(74) 代理人

100109726

弁理士 園田 吉隆

(74) 代理人

100101199

弁理士 小林 義教

(72) 発明者

ディケンズ、トマス ピー。

アメリカ合衆国 ワシントン 98008

、ベルビュー、160番 アヴェニュー

— サウスイースト 2810

最終頁に続く

(54) 【発明の名称】一又は複数のタスクを実行するために使用するシステム及び方法

(57) 【特許請求の範囲】

【請求項 1】

少なくとも一つの処理エンジン(24)によってタスクを実行するために使用するコンピュータによって実行される方法であって、

少なくとも一つの処理エンジン(24)を有する少なくとも一つのプロセッサ(14)に関連するタスク配分メトリックを決定するステップであって、前記少なくとも一つの処理エンジン(24)が各々が一又は複数のスレッド(31)を有する複数の処理コアを有する、前記タスク配分メトリックを決定するステップと、

前記タスクを前記一又は複数のスレッド(31)のうちの一つに割り当てるタスク割当を実行するステップであって、タスク配分メトリックに従って、

(a) 負荷バランス方法か(b) コアアフィニティ方法のいずれかを用いる、前記タスク割当を実行するステップとを含み、

前記負荷バランス方法は、前記タスク配分メトリックが、前記少なくとも一つのプロセッサ(14)に関連する負荷バランスが第1閾値を下回ることを示す場合に実行されるものであり、

前記負荷バランス方法は、

前記少なくとも一つの処理エンジン(24)によって、前記タスクが再入可能タスクであるか否かを決定するステップであって、前記タスクが前記少なくとも一つの処理エンジン(24)に関連する前記一又は複数のスレッド(31)のうちの任意の一つに以前割り当てられていた場合に前記タスクが再入可能タスクであるとされる、前記タスクが再入可

能タスクであるか否かを決定するステップと、

前記タスクが再入可能タスクである場合、前記少なくとも一つの処理エンジン（24）によって、前記タスクの停止基準が満たされたか否かを決定するステップであって、該停止基準は該タスクに従属するいくつかまたはすべてのサブタスクが完了したかどうかを含む、前記タスクの停止基準が満たされたか否かを決定するステップと、

再入可能タスクである前記タスクの前記停止基準が満たされていない場合に、前記少なくとも一つの処理エンジン（24）に関連するタスクのキュー（26）に前記タスクを再入させた後に、前記タスクが割り当てられていた前記一又は複数のスレッドのうちの一つを解放するステップと、を含み、

前記コアアフィニティ方法は、前記タスク配分メトリックが、前記少なくとも一つのプロセッサ（14）に関連するコア（29）アフィニティが第2閾値を下回ることを示す場合に実行されるものであり、

前記コアアフィニティ方法は、

前記少なくとも一つの処理エンジン（24）によって、前記タスクがメインタスクであるか否かを決定するステップであって、メインタスクとは少なくとも一つのサブタスクを有するが他のタスクのサブタスクでないものである、前記タスクがメインタスクであるか否かを決定するステップと、

前記タスクがメインタスクでない場合、前記少なくとも一つの処理エンジン（24）によって前記タスクにとっての親タスクの前記停止基準が満たされたか否かを決定するステップと、

前記タスクがメインタスクではなく、かつ、前記親タスクの前記停止基準が満たされている場合に、メインタスクではない前記タスクが割り当てられていたスレッドと同じスレッドに、前記親タスクを割り当てるステップと、を含む、方法。

【請求項2】

前記負荷バランス方法は、さらに、

前記タスクが再入可能タスクでない場合に、前記タスクは複数のサブタスクに分解可能であるか否かを決定するステップと、

前記タスクが分解可能である場合に、前記少なくとも一つの処理エンジン（24）によって前記タスクの複数のサブタスクを前記キュー（26）に挿入するステップを含む、請求項1に記載の方法。

【請求項3】

前記負荷バランス方法は、さらに、前記タスクの前記停止基準が満たされた場合に、前記少なくとも一つの処理エンジン（24）によって少なくとも一つの蓄積タスクを実行するステップを含み、

前記コアアフィニティ方法における前記親タスクを割り当てるステップの後で、前記親タスクに関連する少なくとも一つの蓄積タスクを実行するステップを含む、請求項1または2に記載の方法。

【請求項4】

前記タスクがメインタスクでない場合、前記少なくとも一つの処理エンジン（24）によって前記タスクにとっての親タスクの前記停止基準が満たされたか否かを決定するステップが、前記親タスクに従属する複数のサブタスクのそれぞれが完了したか否かを決定することを含み、

前記コアアフィニティ方法は、さらに、

前記タスクがメインタスクではなく、かつ、前記親タスクの前記停止基準が満たされていない場合、メインタスクではない前記タスクの状態を完了に設定し、

メインタスクではない前記タスクに割り当てられていた前記一又は複数のスレッド（31）のうちの一つを解放するステップを含む、請求項1乃至3のいずれか1項に記載の方法。

【請求項5】

前記コアアフィニティ方法は、さらに、

10

20

30

40

50

前記タスクがメインタスクか否かを決定する前に行われる、前記タスクが複数のサブタスクに分割可能であるか否かを決定するステップと、

前記タスクが分割可能である場合、前記タスクの前記複数のサブタスクをタスクのキュー(26)に挿入するステップを含む、請求項4に記載の方法。

【請求項6】

タスクの実行において使用するコンピューティング装置であって、前記コンピューティング装置は、

タスクのキュー(26)を記憶するメモリ(12)装置と、

前記メモリ(12)装置に結合し、それぞれが一又は複数のスレッド(31)を有する複数の処理コアを有する少なくとも一つの処理エンジン(24)を含むプロセッサ(14)とを含み、
10

前記少なくとも一つの処理エンジン(24)はタスクマネージャ(27)とキューマネージャ(25)を含み、

前記キューマネージャ(25)はタスクの前記キュー(26)を管理するように構成されており、

前記タスクマネージャ(27)は前記プロセッサ(14)に関連するタスク配分メトリックを決定するように構成されており、

前記タスクマネージャ(27)は、前記タスクを前記一又は複数のスレッド(31)のうちの一つに割り当てるに際して、タスク配分メトリックに従って、(a)負荷バランス方法か(b)コアアフィニティ方法のいずれかを用いるように構成されており、
20

前記負荷バランス方法は、前記タスク配分メトリックが、少なくとも一つの前記プロセッサ(14)に関連する負荷バランスが第1閾値を下回ることを示す場合に、前記タスクマネージャ(27)に実行されるものであり、

前記負荷バランス方法は、

前記タスクが再入可能タスクであるか否かを決定するステップであって、前記タスクが前記少なくとも一つの処理エンジン(24)に関連する前記一又は複数のスレッド(31)のうちの任意の一つに以前割り当てられていた場合に前記タスクが再入可能タスクであるとされる、前記タスクが再入可能タスクであるか否かを決定するステップと、

前記タスクが再入可能タスクである場合、前記タスクの停止基準が満たされたか否かを決定するステップであって、該停止基準は該タスクに従属するいくつかまたはすべてのサブタスクが完了したかどうかを含む、前記タスクの停止基準が満たされたか否かを決定するステップと、
30

再入可能タスクである前記タスクの前記停止基準が満たされていない場合に、前記キューマネージャ(25)を介して、前記少なくとも一つの処理エンジン(24)に関連するタスクのキュー(26)に前記タスクを再入させた後に、前記タスクが割り当てられていた前記一又は複数のスレッド(31)のうちの一つを解放するステップと、を含み、

前記コアアフィニティ方法は、前記タスク配分メトリックが、少なくとも一つの前記プロセッサ(14)に関連するコア(29)アフィニティが第2閾値を下回ることを示す場合に、前記タスクマネージャ(27)に実行されるものであり、

前記コアアフィニティ方法は、

前記タスクがメインタスクであるか否かを決定するステップであって、メインタスクとは少なくとも一つのサブタスクを有するが他のタスクのサブタスクでないものである、前記タスクがメインタスクであるか否かを決定するステップと、

前記タスクがメインタスクでない場合、前記タスクにとっての親タスクの前記停止基準が満たされたか否かを決定するステップと、

前記タスクがメインタスクではなく、かつ、前記親タスクの前記停止基準が満たされている場合に、メインタスクではない前記タスクが割り当てられていたスレッドと同じスレッドに、前記親タスクを割り当てるステップと、を含む、コンピューティング装置。

【請求項7】

前記負荷バランス方法は、さらに、

10

20

30

40

50

前記タスクが再入可能タスクでない場合に、前記タスクは複数のサブタスクに分解可能であるか否かを決定するステップと、

前記タスクが分解可能である場合に、前記キューマネージャ（25）を介して、前記タスクの複数のサブタスクを前記キュー（26）に挿入するステップを含む、請求項6に記載のコンピューティング装置。

【請求項8】

前記タスクマネージャ（27）がさらに、複数の前記サブタスクのうちの一つを前記一又は複数のスレッド（31）のうちの2番目のスレッドに割り当てて、複数の前記サブタスクのうちの前記一つが再入可能タスクであるか否かを決定するように構成されており、

前記スレッドは、前記停止基準が満たされた時に、少なくとも一つの蓄積タスクを実行するためのものである、請求項6又は7に記載のコンピューティング装置。

10

【請求項9】

前記コアアフィニティ方法は、さらに、

前記タスクがメインタスクか否かを決定する前に行われる、前記タスクが複数のサブタスクに分割可能であるか否かを決定するステップと、

前記タスクが分割可能である場合、前記キューマネージャ（25）を介して、前記タスクの前記複数のサブタスクをタスクの前記キュー（26）に挿入するステップとを含み、

前記スレッドは、前記コアアフィニティ方法における前記親タスクを割り当てるステップの後で、前記親タスクに関連する少なくとも一つの蓄積タスクを実行するステップを実行するためのものである、請求項6に記載のコンピューティング装置。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明の分野は概して、少なくとも一つの処理エンジンによってタスクを実行するために使用するシステム及び方法に関するものである。

【背景技術】

【0002】

プロセッサは概して、プロセッサに関連するオペレーティングシステムによって提供されるスレッドの使用を通してタスクを実行することが知られている。スレッドはオペレーティングシステムによって生成されるタスク、又は実行されるタスクのキューからタスクを行って、計算環境において一又は複数の計算、作業、工程、アクション及び／又は手順を達成する。

30

【0003】

しばしば、オペレーティングシステムはマルチスレッドを提供し、マルチスレッドにはそれらによって実行される様々な異なるタスクが割り当てられている。既知のオペレーティングシステムは、タスクを実行するスレッドにタスクを割り当てる。様々な実装形態において、タスクは一又は複数の他のタスク、又はサブタスクに依存する。タスクがスレッドに割り当たられた時に、全て必要な依存タスク及び／又はサブタスクが完了して、スレッドがタスクを実行することができるようになるまで、スレッドはタスクで占有されている。それ自体がマルチプルタスクに分割されたマルチプルタスクにタスクが分割されると、依存タスクが実行されている間、一又は複数のタスクはスレッドに長期間割り当てられる。タスク又はサブタスクがスレッドに割り当たられる時間、及び他のタスクが完了するまでの待機時間により、しばしばプロセッサの使用が非効率となり、ある場合にはスレッド飢餓状態につながる。

40

【発明の概要】

【0004】

ある態様では、少なくとも一つの処理エンジンによってタスクを実行するために使用する方法が提供されている。本方法は、少なくとも一つの処理エンジンを有する少なくとも一つのプロセッサに関連するタスク配分メトリックを決定することを含む。少なくとも一つの処理エンジンは、一又は複数のスレッドを有する。本方法はさらに、タスクを一又は

50

複数のスレッドのうちの一つに割り当て、タスク配分メトリックが、少なくとも一つのプロセッサに関連する負荷バランスが第1閾値を下回ることを示す場合、少なくとも一つの処理エンジンによって、タスクが少なくとも一つの処理エンジンに関連する一又は複数のスレッドのうちの任意のスレッドに以前割り当てられた再入可能タスクであるか否かを決定することを含む。本方法はまた、タスクが再入可能タスクである場合、少なくとも一つの処理エンジンによって、停止基準が満たされたか否かを決定し、停止基準が満たされておらず、タスクが再入可能タスクである場合、少なくとも一つの処理エンジンに関連するタスクのキューにタスクを再入させることも含む。本方法は、タスク配分メトリックが、少なくとも一つのプロセッサに関連するコアアフィニティが第2閾値を下回ることを示す場合、少なくとも一つの処理エンジンによって、タスクがメインタスクであるか否かを決定し、タスクがメインタスクではない場合、少なくとも一つの処理エンジンによって、停止基準が満たされたか否かを決定し、停止基準が満たされており、タスクがメインタスクではない場合、タスクに関連する親タスクを一又は複数のスレッドのうちの一つのスレッドにプルすることを含む。10

【0005】

別の態様では、タスクを実行するために使用するコンピューティング装置が提供されている。コンピューティング装置は、タスクのキューを記憶するメモリ装置と、メモリ装置に関連し、一又は複数のスレッドを有する少なくとも一つの処理エンジンを含むプロセッサを含む。少なくとも一つの処理エンジンは、タスクマネージャと、前記タスクのキューを管理するように構成されたキューマネージャを含む。タスクマネージャは、前記プロセッサに関連し、タスクを一又は複数のスレッドのうちの一つのスレッドに割り当てるタスク配分メトリックを決定するように構成されている。タスクマネージャは、タスク配分メトリックが前記プロセッサに関連する負荷バランスが第1閾値を下回ることを示す場合、タスクが前記一又は複数のスレッドのうちの任意のスレッドに以前割り当てられた再入可能タスクであるか否かを決定し、タスクが再入可能タスクである場合、停止基準が満たされたか否かを決定し、停止基準が満たされておらず、タスクが再入可能タスクである場合、前記キューマネージャを介してタスクをタスクのキューに再入させるように構成されている。タスクマネージャは、タスク配分メトリックが前記プロセッサに関連するコアアフィニティが第2閾値を下回ることを示す場合に、タスクがメインタスクであるか否かを決定し、タスクがメインタスクではない場合、停止基準が満たされたか否かを決定し、停止基準が満たされており、タスクがメインタスクではない場合、タスクに関連する親タスクを一又は複数のスレッドのうちの前記一つにプルするように構成されている。20

【0006】

上述の特徴、機能及び利点は、様々な実行形態において独立に実現することが可能であるか、さらに別の実行形態において組み合わせることが可能である。これらの実行形態について、以下の説明及び添付図面を参照してさらに詳細に説明する。

【図面の簡単な説明】

【0007】

【図1】一又は複数のタスクを実行するために使用する例示のコンピューティング装置のブロック図である。40

【図2】一又は複数のタスクを実行するために使用する例示の方法のフロー図である。

【図3】図2の例示の方法で使用される、一又は複数のタスクを実行するために使用する別の例示の方法のフロー図である。

【発明を実施するための形態】

【0008】

本明細書に記載される主題は概して、タスクのキューを管理し、タスクを実行するために一又は複数の処理エンジンに関連するスレッドを保っている間に、マルチプルサブタスクを有する一又は複数のタスクを実行するために使用するシステム及び方法に関するものである。

【0009】

50

ある実行形態では、本明細書に記載された方法、システム、及びコンピュータによって読み取可能な媒体の技術的効果は、(A) 一又は複数のスレッドを有する少なくとも一つの処理エンジンを有する少なくとも一つのプロセッサに関連するタスク配分メトリックを決定し、(B) タスクを一又は複数のスレッドのうちの一つのスレッドに割当てし、(C) タスク配分メトリックが、少なくとも一つのプロセッサに関連する負荷バランスが第1閾値を下回ることを示す場合、少なくとも一つの処理エンジンによって、タスクが少なくとも一つの処理エンジンに関連する前記一又は複数のスレッドのうちの任意のスレッドに以前割り当てられた再入可能タスクであるか否かを決定し、(D) タスクが再入可能タスクである場合、少なくとも一つの処理エンジンによって停止基準が満たされたか否かを決定し、(E) 停止基準が満たされておらず、タスクが再入可能タスクである場合、少なくとも一つの処理エンジンに関連するタスクのキューにタスクを再入させ、(F) タスク配分メトリックが、少なくとも一つのプロセッサに関連するコアアフィニティが第2閾値よりも下回ることを示す場合、少なくとも一つの処理エンジンによってタスクがメインタスクであるか否かを決定し、(G) タスクがメインタスクでない場合、少なくとも一つの処理エンジンによって、停止基準が満たされたか否かを決定し、(H) 停止基準が満たされており、タスクがメインタスクではない場合、タスクに関連する親タスクを一又は複数のスレッドのうちの前記一つのスレッドにプルすることのうちの少なくとも一つを含む。
10

【0010】

本明細書において、「1つの」という語から始まる単数形の要素又はステップは、複数の要素又はステップを除外することが明示的に記載されていない限り、複数の要素又はステップを除外しないと理解すべきである。さらに、本発明の「一実行形態」又は「例示的実行形態」への言及は、記載されている特徴を取り込む他の実行形態の存在を除外するものではない。
20

【0011】

図1は、例示のコンピューティング装置10のブロック図である。例示的な実行形態では、コンピューティング装置10は、メモリ12及びプログラムされた命令を実行するためのメモリ12と結合されたプロセッサ14を含む。プロセッサ14は一又は複数の処理エンジン（例えば、マルチコア構成で）を含む。コンピューティング装置10は、本明細書に記載される一又は複数の作業を実行するために、プロセッサ14において一又は複数の処理エンジンを確立するようにプログラミング可能である。例えば、プロセッサ14は、作業を一又は複数の実行可能な命令としてエンコードし、メモリ12で実行可能な命令を提供することによってプログラム可能である。
30

【0012】

プロセッサ14は、汎用中央処理装置（CPU）、マイクロコントローラ、縮小命令セットコンピュータ（RISC）プロセッサ、特定用途向け集積回路（ASIC）、プログラマブル論理回路（PLC）、及び／又は、本明細書に記載した機能を実行可能な任意の他の回路又はプロセッサを非限定的に含む。本明細書に記載されている方法は、限定するものではないが、記憶装置及び／又はメモリ装置を含む、コンピュータで読み込可能な媒体内に具現化される、実行可能な命令としてエンコードすることができる。このような命令は、プロセッサ14で実行された場合、本明細書に記載された方法の少なくとも一部をプロセッサ14に行わせることができる。上述の例は典型例にすぎず、したがって、プロセッサという用語の定義及び／又は意味を何らかの方法で制限することを意図していない。
40

【0013】

メモリ12は、本明細書に記載されているように、実行可能な命令及び／又は他のデータなどの情報が記憶され検索されることを可能にする一又は複数の装置である。メモリ12は、限定しないが、キャッシュ、ダイナミックランダムアクセスメモリ（DRAM）、静态ディスクランダムアクセスメモリ（SRAM）、半導体ディスク、及び／又はハードディスクなど、一又は複数のコンピュータで読み込可能な媒体を含みうる。図示したように、メモリ12は実行される一又は複数のタスクを記憶するキュー26を含む。メモリ12は限定しないが、作業中の命令、ファームウェア、ソフトウェア、サブタスクカウンタ、
50

アイドルタスク、及び／又は本明細書に記載された方法及びシステムで使用するのに好適なその他何らかの種類のデータを記憶するようにさらに構成される。さらに、メモリ12及びプロセッサ14は別々のものとして図示されており、当然ながら、プロセッサ14及びメモリ12を含む統合装置はいくつかのコンピューティング装置の実行形態に使用される。

【0014】

例示的な実行形態では、コンピューティング装置10は、プロセッサ14に結合されているプレゼンテーションインターフェース16を含む。プレゼンテーションインターフェース16は、限定しないが、インストールデータ、コンフィギュレーションデータ、テストデータ、エラーメッセージ、及び／又はユーザ20に対する任意の他の形式のデータなどの情報を出力（例：表示、印刷及び／又はその他の出力）する。例えば、プレゼンテーションインターフェース16は、陰極線管（CRT）、液晶ディスプレイ（LCD）、発光ダイオード（LED）ディスプレイ、有機LED（OLED）ディスプレイ、及び／又は「電子インク」ディスプレイなどのディスプレイデバイスに結合されたディスプレイアダプタ（図1に示せず）を含む。いくつかの実行形態では、プレゼンテーションインターフェース16は一よりも多い表示装置を含む。10

【0015】

例示的な実行形態では、コンピューティング装置10は、ユーザ20からの入力を受信する入力インターフェース18を含む。例示的な実行形態では、入力インターフェース18はプロセッサ14に結合されており、例えば、キーボード、リーダー、ポインティングデバイス、マウス、スタイルスペン、タッチセンサーパネル（例えば、タッチパッド又はタッチスクリーン）、ジャイロスコープ、加速度計、及び／又は音声入力インターフェースを含む。20

【0016】

オペレーティングシステム及びアプリケーションの命令は、本明細書に記載された一又は複数の工程を実行するためにプロセッサ14によって実行される持続性メモリ12の機能性の形態で位置決めされている。異なる実行形態のこれらの命令は、例えばメモリ12、又は別のメモリ等の異なる物理的又は有形のコンピュータによって読み取可能な媒体、例えば非限定的にフラッシュドライブ、CD-ROM、サムドライブ、フロッピー（登録商標）ディスク等を含むコンピュータによって読み取可能な媒体22で具現化される。30

【0017】

さらに命令は、例えば非限定的に、フラッシュドライブ、CD-ROM、サムドライブ、フロッピー（登録商標）ディスク等を含む持続性のコンピュータによって読み取可能な媒体22の機能性の形態で位置決めされている。コンピュータによって読み取可能な媒体22は、プロセッサ14によるアクセス及び／又は実行を可能にするためにコンピューティング装置10に選択的に挿入可能、及び／又は取り出し可能である。ある実行例では、コンピュータによって読み取可能な媒体22は、メモリ12及び／又はプロセッサ14に関連するCD/DVDドライブ又は他のデバイスに挿入又は配置される光ディスク又は磁気ディスクを含む。場合によっては、コンピュータで読み取り可能な媒体22は取り外し可能でなくともよい。40

【0018】

コンピューティング装置10は、本明細書に記載されるように実行されるように様々な処理環境において具現化される。具体的には、例えば、コンピューティング装置10は非限定的に、一又は複数のパソコン、組み込みシステム、ワークステーション、サーバ、サーバシステム、ネットワークコンピュータ、ポータブルコンピュータ（例：iPAD（登録商標））、スマートホン、タブレット、及び／又は一又は複数の作業を実行するためには好適な他の装置を含む。当然ながら、本明細書のコンピューティング装置10は、処理システム及び／又は装置の一又は複数の組み合わせによって提供される。

【0019】

一般に、プロセッサ14は一又は複数のタスクを実行するために提供される。各タスク50

はマルチブルサブタスクを含み、各サブタスクはさらにサブタスクを含むことができる。このように、タスクは処理エンジンによって実行されるサブタスクに分解できる。本明細書で使用する「タスク」という表現は、タスク及びサブタスクのいずれか、あるいは両方を指すものである。さらに、一つのタスクと一又は複数の他のタスクとの関係を示すために、階層的な表現（例：親又はメイン）をタスクと組み合わせて使用する場合がある。図1に示すように、例示の実行形態では、プロセッサ14は一又は複数の処理エンジン24をソフトウェア、ファームウェア、ハードウェアにおいて提供する。一又は複数の処理エンジン24は、キューマネージャ25とタスクマネージャ27を含む。

【0020】

例示の実行形態では、キューマネージャ25は、キューレベルを出入りするタスクの動きを管理するように構成されており、その一方タスクマネージャ27は、下記の図2及び3に示すように、スレッドに入った時にどの方法でタスクが実行されているか及び／又は処理されているかを決定するように構成されている。各処理エンジン24は、ソフトウェア及び／又はファームウェアによって提供された、処理コア29ごとに一又は複数のスレッド31を有する一又は複数の処理コア29を含む。タスクマネージャ27は、特定の一又は複数のタスクを実行するために、タスク及び／又はサブタスクをキューレベルから本明細書で記載される一又は複数のスレッド31に割り当てる機能を持つ。

10

【0021】

例示の実行形態では、タスクマネージャ27は、コアアフィニティ閾値及び／又は負荷バランス閾値に基づくタスク配分メトリックを決定するように構成される。さらに具体的には、タスクマネージャ27はキューマネージャ25によって表示されるように、プロセッサ14の現在の状態、及び／又はキューレベルから処理すべき一又は複数の次のタスクから予測されるプロセッサ14の状態に基づいて、一又は複数のタスク配分メトリックを決定する。さらに、タスクマネージャ27は、現在の及び／又は予測される状態に基づいたタスク配分メトリックをコアアフィニティ閾値及び／又は負荷バランス閾値と比較する。この比較に基づき、タスクマネージャ27は次のタスクをコアアフィニティ閾値の関数として特定の処理エンジン24に割り当てる、あるいは負荷バランス閾値の関数として別の処理エンジン24に割り当てる。加えて、又はこれに代えて、タスクマネージャ27は次のタスクが再入可能タスク及び／又はサブタスクに分割できるタスクであるか否かに基づいて、次のタスクを割り当てる。

20

【0022】

このように、タスクマネージャ27は、タスクの利用可能な処理エンジン24への配分を予測的に及び／又はリアルタイムで、キューマネージャ25（例：再入可能タスク、メインタスク）によって管理され、処理エンジン24によって実行されるタスクの種類の関数として制御する。本明細書のコアアフィニティ及び／又は負荷バランスの能力によりさらに、一又は複数の処理エンジン24間の負荷バランスを提供しながら、既知の方法に比べて利用可能な処理エンジン24全体のタスクの実行をより効率的に行うことが可能になる。

30

【0023】

図2は、一又は複数のタスクを実行するために使用する例示の方法100を示す。本明細書に記載される方法は、プロセッサ14に関連する少なくとも一つの処理エンジンによって実行される少なくとも一つのステップ及び／又は工程を含む。図2の例示の実行形態では、例えば、方法100は一又は複数のコンピューティング装置10において提供されるプロセッサ14に関連する一又は複数の処理エンジンによって実行される。

40

【0024】

タスクがコンピューティング装置10内で生成されると、タスクはキューレベルの適所へ挿入される。キューレベルは先入れ先出し（FIFO）キューであり、タスクがキューレベルに挿入された順番でスレッド31が利用可能になった場合にスレッド31にタスクが割り当てる。当然ながら、その他の種類のキューレベル（後入れ先出し（LIFO）キュー、又はその他の非FIFOキュー）及び／又は一又は複数のスレッドとのインタラクション

50

ン、及び複合キューをその他の実行形態において使用できる。さらに当然ながら、上述したように、キュー 26 はメモリ 12 内にあるところを図示されているが、キューはメモリ 12 に、またプロセッサ 14 の内部又は外部に含まれうる。

【0025】

方法 100 が開始されると、プロセッサ 14 に関連するタスク配分メトリックは、処理エンジン 24、又はプロセッサ 14 内の別の処理リソースによって決定される 101。タスク配分メトリックは例えば、処理エンジン 24 に含まれるスレッド内の一又は複数のタスクの配分／実行に関連する負荷バランス、コアアフィニティ、又はその他のメトリックを示すものである。タスク配分メトリックの決定 101 では、処理エンジン 24 はメトリックを一又は複数のスレッドと比較して、プロセッサ 14 に関連するコアアフィニティ及び／又は負荷バランスが、プロセッサ 14 の一又は複数の異なる所望の実行を行うために十分か、不十分か、最適か、又は非最適であるかを決定する。10

【0026】

例えば、タスク配分メトリックは、メモリへの任意の数の読み込み及び／又は書き込みを含む。閾値はメモリ 12 への任意の数の読み込み及び／又は書き込みであり、メモリ 12 への実際の読み込み及び／又は書き込みの数が第 1 所定数を超える場合、プロセッサ 14 に関連するコアアフィニティは第 2 閾値を下回る。逆に、メモリ 12 の実際の読み込み及び／又は書き込みの数が（第 1 所定数と同じ又は異なる）第 2 所定数よりも少ない場合、又はプロセッサ 14 のコアがアイドルである場合、プロセッサ 14 に関連する負荷バランスは第 1 閾値を下回る。プロセッサ 14 のその他さまざまな特徴及び／又は性能により、他の実行形態におけるコアアフィニティ及び／又は負荷バランスを示すタスク配分メトリックが提供される。20

【0027】

例えば少なくとも 1 つのプロセッサ 14 に関連する負荷バランスが第 1 閾値を下回ることをタスク配分メトリックが示す場合、タスクマネージャ 27 は処理するタスクをスレッドに割り当てる。タスクを受け取ると、スレッドは、タスクが以前複数のスレッドのうちの任意の一つに割り当てられた再入可能タスクであるか否かを決定 102 するように稼働する。この例示の実行形態にあるように、「再入可能」タスクという表現は、以前処理エンジン 24 のスレッドに割り当てられており、本明細書に記載される方法 100 の一又は複数の他のステップに依存するタスクを指すものである。30

【0028】

図 2 に示すように、タスクが再入可能タスクでない場合、方法 100 は、タスクがマルチプルサブタスクに分割可能か否かの決定 104 へ進む。タスクは、例えばタスク内の作業関係、タスク内の作業数、タスク内の作業の複雑さ等の一又は複数の異なる基準に基づいて分割可能である。ある実行形態では、タスクの分割可能性はさらに、使用されるスレッド、処理エンジン 24、及び／又はプロセッサ 14 の種類及び／又は容量に関連する。40

【0029】

タスクが分割可能である場合、スレッドはタスクのマルチプルサブタスクをキュー 26 へ挿入する 106。サブタスクがキュー 26 に入った後で、スレッドは、タスクがスレッドに割り当てられたことを示す状態を設定 108 する。さらに、スレッドは次の処理のためにタスクをキュー 26 に再入 110 させて、他のタスクのためにスレッドを開放 116 する。スレッドが、タスクが分割可能でないことを決定 104 した場合、スレッドはタスクを実行 112 する。タスクが実行された後で、スレッドはタスクの状態を完了と設定 114 し、一又は複数の他のタスクで使用するためにスレッドを開放 116 する。

【0030】

さらに、タスクがキュー 26 に再入したために、スレッドは再入タスクと関連する、又は関連していない他のタスクを自由に完了させることができる。したがって、方法 100 は、タスクが完了するまでスレッドに残ることを可能にする代わりに、スレッドを解放するところが既知の方法と異なるところである。一又は複数の実行形態において、方法 100 を用いて処理エンジン 24、プロセッサ 14、及び／又はコンピューティング装置 10 50

内のスレッド飢餓状態、デッドロック、及び／又はライブロックを減らす、制限する及び／又は除去することができる。さらに、当然ながら、方法100はステップ116で終了する、すなわちスレッドを解放するため、本明細書の方法100及び他の方法は、プロセッサ14内で割り当てられたリソース内で効率的にタスクの様々な複雑性を実行するのに好適である。少なくとも一つの実行形態では、処理エンジン24は、マルチプルサブタスクを有する少なくとも一つのタスクを有するマルチプルタスクを実行するための单ースレッドを含む。

【0031】

再び図2を参照する。スレッドが、タスクがスレッドに以前割り当てられていた（例えば、ステップ108で設定されたタスクの状態で示される）と決定102した場合、スレッドは停止基準が満たされたか否かを決定120する。停止基準は例えば、完了したタスクに関する一つの、幾つかの、ある割合の、及び／又は全てのサブタスク、又はタスクに関連する特定のタスクが完了したことを含む。少なくとも一つの実施形態では、タスクがマルチプルサブタスクに分割され、キュー26に挿入される106と、サブタスクカウンタが生成され、このカウンタはサブタスクのうちの一つが完了するたびにデクリメントすることができる。上記実施形態では、停止基準は、サブタスクカウンタがゼロと等しい時、すなわち、タスクに従属する各マルチプルサブタスクが完了した時を含むことができる。ある実行例では、停止基準は、少なくとも一つのマルチプルサブタスクが完了した時を含むことができる。様々な実施形態では、最小量の処理が実行され、タスクを進めることができることを示すために、多様な異なる基準が提供されている。他の実施形態では、適切なレベルの完了が達成されたこと、及び／又はタスクが完全に実行されたことを決定するために他の基準が使用される。当然ながら、停止基準は、実行されるタスクの種類、及び／又はタスクが実行される方法に関連する一又は複数のパラメータ（例えば精度）に基づいて変化する。

【0032】

停止基準が満たされない場合、スレッドは次の処理のためにタスクをキュー26に再入110させ、これによりスレッドは他のタスクのために解放される。そうでなく、停止基準が満たされた場合、スレッドは任意にタスクに関連する一又は複数の蓄積タスクを実行122する。例えば、蓄積タスクは非限定的に、複数の完了サブタスクの結果への実行、例えば多数の結果を加算する等を含む。少なくとも一つの実施形態では、ステップ122において蓄積タスクが実行されない。蓄積タスクが完了した時に、必要に応じて、スレッドはタスクの状態を完了に設定114して、一又は複数の他のタスクで使用するためにスレッドが開放116される。

【0033】

図2に示す方法100は、他のタスクが完了するのを待機しているアイドルタスクからスレッドを開放することによって、処理エンジンがタスクを処理できるようにする。方法100は、数千、数十万、数百万又はそれ以上のタスク及び／又はサブタスクを処理エンジン24に割り当てて、既知の方法よりもさらに迅速にさらに効率的にタスクを完了させるために使用される。例示の実行形態では、タスクの種類、又はタスクの他のタスクに対する関係を考慮せずに、スレッドが解放されキュー26からの次のタスクが割り当てられるため、方法100ではさらに、幾つかのスレッド間の負荷バランスが改善される。このため、方法100では、一又は複数の実施形態において処理エンジン24のコアアフィニティへのエンファシスを軽減し、いくつかの処理エンジン24間の負荷バランスの改善が強調される。

【0034】

さらに、多様性において、各スレッドはタスクを処理しているか、又はタスクをキュー26へ移動させるかのいずれかを実行しているため、本明細書の方法はスレッドを効率的に使用し、これにより、既知の処理方法においては問題（例：スレッド飢餓状態、デッドロック、及び／又はライブロック）が生じるタスクの実行に、より少ないスレッド（例えば单ースレッド）が使用されうる。

10

20

30

40

50

【 0 0 3 5 】

タスク配分メトリックが、方法 1 0 0 において決定 1 0 1 されたように、少なくとも一つのプロセッサ 1 4 に関連するコアアフィニティが第 2 閾値を下回ることを示す場合、処理エンジン 2 4 はノード A において、一又は複数のタスクを実行するために使用する例示の方法 2 0 0 へ分岐する。例示の方法 2 0 0 は図 3 に示されている。

【 0 0 3 6 】

方法 1 0 0 と同様に、タスクがコンピューティング装置 1 0 内で生成された時、タスクはキュー 2 6 の適所に挿入される。キュー 2 6 は、先入れ先出し（FIFO）キューであり、スレッドが利用可能になった時にタスクがキュー 2 6 に挿入された順でタスクがスレッドに割り当てられる。上述したように、当然ながら、他の種類のキュー 2 6 、及び／又はスレッドとのインタラクションを他の実施形態において使用可能である。処理エンジン 2 4 のスレッドが利用可能になると、処理のためにタスクがスレッドに割り当てられる。図 3 の例示の実行形態では、タスクを受け入れた時に、スレッドはタスクがマルチプルサブタスクに分割可能か否かの決定 2 0 2 を実行する。上述したように、タスクは一又は複数の異なる基準、例えばタスク内の作業の関係、タスク内の作業数等に基づいて分割可能である。

10

【 0 0 3 7 】

タスクが分割可能である場合、スレッドはタスクのマルチプルサブタスクをキュー 2 6 に挿入し 2 0 4 、他のタスクを実行するためにスレッドが開放 2 1 2 される。

20

【 0 0 3 8 】

逆に、タスクが分割可能でない場合、スレッドはタスクを実行し 2 0 6 、タスクがメインタスクであるか否かを決定する 2 0 8 。本明細書で使用する「メインタスク」という表現は、少なくとも一つのサブタスクを有するが、他のタスクに対するサブタスクではないタスクを指すものである。タスクがメインタスクであることが決定された場合、スレッドはタスクが完了したことを決定し、タスクの状態を完了に設定し 2 1 0 、他のタスクを実行するためにスレッドを解放する 2 1 2 。しかしながらタスクがメインタスクではない場合、スレッドは、親タスクの停止基準が満たされたか否かを決定する 2 1 4 。上述したように、停止基準は、タスクがマルチプルサブタスクに分割された時に生成されたサブタスクカウンタの値、又はタスク又はサブタスクの完了のその他様々な評価基準である。

30

【 0 0 3 9 】

親タスクの停止基準が満たされない場合、スレッドはタスクが完了したことを決定し、タスクの状態を完了に設定し 2 1 0 、他のタスクを実行するためにスレッドを解放する 2 1 2 。

【 0 0 4 0 】

逆に、親タスクの停止基準が満たされた場合、処理エンジン 2 4 は親タスクをスレッドにプルし 2 1 6 、必要に応じて親タスクに関連する蓄積タスクを実行する 2 1 8 。蓄積タスクが完了した際に、もしあれば、スレッドは親タスクがメインタスクであるか否かを決定する 2 0 8 ためにループバックする。本明細書に記載された工程は、メインタスクが完了したことが決定されるまで継続し、スレッドが開放される 2 1 2 。親タスクをスレッドにプルすることで、タスクを実行した後に、方法 2 0 0 は、マルチプルサブタスクを有するタスクを実行する既知の方法よりもエンハンストコアアフィニティを高め、一又は複数の実施形態における処理エンジン 2 4 間の負荷バランスへのエンファシスを軽減することができる。さらに具体的には、サブタスク、又は親タスクの下のタスクは同じスレッドによって実質的に実行され、これにより既知の方法に比べて、メインタスクの関連タスクが同じ処理エンジン 2 4 において実行される可能性がより高くなる。

40

【 0 0 4 1 】

さらに、停止基準が満たされる前に、他のタスクが完了するまで待機しているアイドルタスクからスレッドが開放される 2 1 2 ことで、方法 2 0 0 は既知の方法よりもさらに迅速にさらに効率的にタスクを完了させることができる。このように、各スレッドはタスクを処理している、又はタスクをキュー 2 6 へ移動させている、のいずれかであるため、処

50

理エンジン 2 4 はその一又は複数のスレッドを効率的に使用する。

【 0 0 4 2 】

上記効率的な使用により、より少ないスレッドを使用して、既知の処理方法において問題（例：スレッド飢餓状態、デッドロック、及び／又はライプロック）が生じるタスクを実行することができる。このように、本明細書に記載される方法は、スレッドの数が限定された特定の処理環境、及び／又は効率性が好まれるその他の処理環境において使用され、その一方で、スレッド飢餓状態、デッドロック、及び／又はライプロック状態の可能性が制限される及び／又は除去される。

【 0 0 4 3 】

当然ながら、方法 1 0 0 及び方法 2 0 0 は互いに連動する形で本明細書に記載されているが、一つ、又は両方の方法、及び／又はこれら の方法の置換をその他様々な実施形態で別々に使用可能である。例えば、負荷バランスが制御設計の検討事項であるコンピューティング装置においては、方法 1 0 0 を方法 2 0 0 と組み合わせずに単独に使用する。別の実行例では、コアアフィニティが制御設計の検討事項であるコンピューティング装置において、方法 2 0 0 を方法 1 0 0 と組み合わせずに単独に使用する。上記実施形態では、タスク配分メトリックを決定するステップは修正又は省略される。

10

【 0 0 4 4 】

説明及び図面において、少なくとも一つの処理エンジン 2 4 によってタスクを実行するために使用されるコンピュータによって実行可能な方法が開示されており、この方法は、少なくとも一つのスレッド 3 1 を有する少なくとも一つの処理エンジン 2 4 を有する少なくとも一つのプロセッサ 1 4 に関連するタスク配分メトリックを決定するステップと、タスク配分メトリックが、少なくとも一つのプロセッサ 1 4 に関連する負荷バランスが第 1 閾値を下回ることを示す場合に、一又は複数のスレッド 3 1 のうちの一つにタスクを割り当てるステップと、少なくとも一つの処理エンジン 2 4 によって、タスクが少なくとも一つの処理エンジン 2 4 に関連する前記一又は複数のスレッド 3 1 の任意の一つに以前割り当てられた再入可能タスクであるか否かを決定するステップと、タスクが再入可能タスクである場合、少なくとも一つの処理エンジン 2 4 によって停止基準が満たされたか否かを決定するステップと、停止基準が満たされておらず、タスクが再入可能タスクである場合、少なくとも一つの処理エンジン 2 4 に関連するタスクのキュー 2 6 にタスクを再入させるステップと、タスク配分メトリックが、少なくとも一つのプロセッサ 1 4 に関連するコア 2 9 アフィニティが第 2 閾値を下回ることを示す場合に、少なくとも一つの処理エンジン 2 4 によって、タスクがメインタスクであるか否かを決定するステップと、タスクがメインタスクでない場合、少なくとも一つの処理エンジン 2 4 によって停止基準が満たされたか否かを決定するステップと、停止基準が満たされており、タスクがメインタスクでない場合、タスクに関連する親タスクを一又は複数のスレッド 3 1 のうちの前記一つにプルするステップを含む。

20

【 0 0 4 5 】

ある変形例では、本方法において、タスクが再入可能タスクである場合に停止基準が満たされたか否かを決定するステップが、タスクに従属する少なくとも一つのマルチプルサブタスクが完了したか否かを決定することを含む。別の変形例では、方法はさらに、タスクをタスクのキュー 2 6 に再入させた後で、前記一又は複数のスレッド 3 1 のうちの前記一つを解放することを含む。さらに別の変形例では、本方法はさらに、タスクが再入可能タスクでない場合に、タスクを複数のサブタスクに分割可能であるか否かを決定することを含む。ある場合には、本方法はさらに、タスクが分割可能である場合、少なくとも一つの処理エンジン 2 4 によって、タスクの複数のサブタスクをキュー 2 6 に挿入することを含む。別の場合において、本方法はさらに、停止基準が満たされた場合に、少なくとも一つの処理エンジン 2 4 によって、少なくとも一つの蓄積タスクを実行することを含む。ある実行例において、本方法はさらに、親タスクが一又は複数のスレッド 3 1 のうちの前記一つにプルされた後で、親タスクに関連する少なくとも一つの蓄積タスクを実行することを含む。別の実行例では、本方法において、タスクがメインタスクでない場合に停止基準

30

40

50

が満たされたか否かを決定するステップが、タスクに従属するマルチプルサブタスクのそれぞれが完了したか否かを決定することを含む。さらに別の実行例では、本方法はさらに、停止基準が満たされておらず、タスクがメインタスクでない場合に、タスクをタスクのキュー26に再入させた後で、タスクの状態を完了に設定し、一又は複数のスレッド31のうちの前記一つを解放することを含む。ある代替例では、本方法はさらに、タスクがメインタスクであるか否かを決定する前に、タスクを複数のサブタスクへ分割可能であるか否かを決定することを含む。さらに別の代替例では、本方法はさらに、タスクが分割可能である場合、タスクの複数のサブタスクをタスクのキュー26へ挿入することを含む。

【0046】

ある態様では、タスクの実行において使用するコンピューティング装置が開示されており、このコンピューティング装置は、タスクのキュー26を記憶するメモリ12装置と、メモリ12装置に結合し、一又は複数のスレッドを有し少なくとも一つの処理エンジン24を含むプロセッサ14を含み、前記少なくとも一つの処理エンジン24はタスクマネージャ27とキューマネージャ25を含み、キューマネージャ25は、タスクの前記キュー26を管理するように構成されており、前記タスクマネージャ27は前記プロセッサ14に関連するタスク配分メトリックを決定し、タスクを一又は複数のスレッド31のうちの一つに割り当てるように構成されており、前記タスクマネージャ27は、タスク配分メトリックが前記プロセッサ14に関連する負荷バランスが第1閾値を下回ることを示す場合に、タスクが前記一又は複数のスレッド31のうちの任意の一つに以前割り当てられた再入可能タスクであるか否かを決定し、タスクが再入可能タスクである場合、停止基準が満たされたか否かを決定し、停止基準が満たされておらず、タスクが再入可能タスクである場合に、タスクをタスクのキュー26に前記キューマネージャ25を介して再入させるように構成されており、前記タスクマネージャ27は、タスク配分メトリックが、前記プロセッサ14に関連するコア29アフィニティが第2閾値を下回ることを示す場合に、タスクがメインタスクであるか否かを決定し、タスクがメインタスクでない場合に、停止基準が満たされたか否かを決定し、停止基準が満たされており、タスクがメインタスクではない場合に、タスクに関連する親タスクを一又は複数のスレッド31のうちの前記一つにプルするように構成されている。

【0047】

ある変形例では、コンピューティング装置において、前記タスクマネージャ27はさらに、タスクが再入可能タスクでない場合、タスクを複数のサブタスクに分割可能であるか否かを決定するように構成されている。別の変形例では、コンピューティング装置において、前記タスクマネージャ27はさらに、タスクが分割可能である場合、前記キューマネージャ25を介してタスクの複数のサブタスクをキュー26に挿入するように構成されている。さらに別の変形例では、コンピューティング装置において、前記タスクマネージャ27はさらに、複数のサブタスクのうちの一つを一又は複数のスレッド31のうちの2番目の一つに割り当て、複数のサブタスクのうちの一つが再入可能タスクであるか否かを決定するように構成されている。ある場合には、コンピューティング装置において、前記タスクマネージャ27はさらに、停止基準が満たされた時に少なくとも一つの蓄積タスクを実行するように構成されている。別の場合において、コンピューティング装置では、タスクマネージャ27はさらに、タスクがメインタスクであるか否かを決定する前に、タスクを複数のサブタスクに分割可能であるか否かを決定するように構成されている。ある実施例では、コンピューティング装置において、前記タスクマネージャ27はさらに、タスクが分割可能である場合、前記キューマネージャ25を介してタスクの複数のサブタスクをキュー26に挿入するように構成されている。別の実施例では、コンピューティング装置において、前記タスクマネージャ27はさらに、親タスクが一又は複数のスレッド31のうちの前記一つにプルされた後で、親タスクに関連する少なくとも一つの蓄積タスクを実行するように構成されている。

【0048】

さらに、当然ながら、本発明の一又は複数の態様は、本明細書に記載された機能、方法

10

20

30

40

50

、及び／又は工程を実行するように構成された時に、汎用コンピューティング装置を特殊用途のコンピューティング装置に変換する。

【0049】

本明細書では、ベストモードを含む様々な実施形態を開示する実施例を使用しているため、当業者は任意の装置やシステムの作成ならびに使用、及び組込まれた任意の方法の実行を含め、実施形態を実行することができる。特許性を有する範囲は、特許請求の範囲によって規定され、当業者であれば想起できる他の実施例を含みうる。このような他の実施例は、それらが特許請求の範囲の文字言語から逸脱しない構造要素を有する場合、あるいは、それらが特許請求の範囲の文字言語との有意でない相違を有する等価な構造要素を含んでいる場合は、特許請求の範囲の範囲内にあることを意図している。

10

また、本願は以下に記載する態様を含む。

(態様1)

少なくとも一つの処理エンジン(24)によってタスクを実行するために使用するコンピュータによって実行される方法であって、

少なくとも一つの処理エンジン(24)を有する少なくとも一つのプロセッサ(14)に関連するタスク配分メトリックを決定するステップであって、前記少なくとも一つの処理エンジン(24)が一又は複数のスレッド(31)を有する、決定するステップと、前記タスクを前記一又は複数のスレッド(31)のうちの一つに割り当てるステップと、

前記タスク配分メトリックが、前記少なくとも一つのプロセッサ(14)に関連する負荷バランスが第1閾値を下回ることを示す場合、前記少なくとも一つの処理エンジン(24)によって、前記タスクが前記少なくとも一つの処理エンジン(24)に関連する前記一又は複数のスレッド(31)のうちの任意の一つに以前割り当てられた再入可能タスクであるか否かを決定するステップと、

前記タスクが再入可能タスクである場合、前記少なくとも一つの処理エンジン(24)によって、停止基準が満たされたか否かを決定するステップと、

前記停止基準が満たされておらず、前記タスクが再入可能タスクである場合に、前記少なくとも一つの処理エンジン(24)に関連するタスクのキュー(26)に前記タスクを再入させるステップと、

前記タスク配分メトリックが、前記少なくとも一つのプロセッサ(14)に関連するコア(29)アフィニティが第2閾値を下回ることを示す場合、前記少なくとも一つの処理エンジン(24)によって、前記タスクがメインタスクであるか否かを決定するステップと、

前記タスクがメインタスクでない場合、前記少なくとも一つの処理エンジン(24)によって停止基準が満たされたか否かを決定するステップと、

前記停止基準が満たされており、前記タスクがメインタスクでない場合、前記タスクに関連する親タスクを前記一又は複数のスレッド(31)の前記一つにプルするステップを含む方法。

(態様2)

前記タスクが再入可能タスクである場合に前記停止基準が満たされたか否かを決定するステップが、前記タスクに従属するマルチプルサブタスクのうちの少なくとも一つが完了したか否かを決定し、前記タスクをタスクのキュー(26)に再入させた後で、前記一又は複数のスレッド(31)のうちの前記一つを解放することを含む、態様1に記載の方法。

(態様3)

前記タスクが再入可能タスクでない場合に、前記タスクは複数のサブタスクに分解可能であるか否かを決定し、前記タスクが分解可能である場合に、前記少なくとも一つの処理エンジン(24)によって前記タスクの複数のサブタスクを前記キュー(26)に挿入することをさらに含む、態様1又は2に記載の方法。

(態様4)

20

30

40

50

前記停止基準が満たされた場合に、前記少なくとも一つの処理エンジン（24）によって少なくとも一つの蓄積タスクを実行し、前記親タスクを前記一又は複数のスレッド（31）のうちの前記一つにブルした後で、前記親タスクに関連する少なくとも一つの蓄積タスクを実行することをさらに含む、態様1乃至3のいずれか1項に記載の方法。

(態様5)

前記タスクがメインタスクでない場合に前記停止基準が満たされたか否かを決定するステップが、前記タスクに従属するマルチブルサブタスクのそれぞれが完了したか否かを決定することを含み、前記停止基準が満たされておらず、前記タスクがメインタスクでない場合、前記タスクをタスクのキュー（26）に再入させた後で、前記タスクの状態を完了に設定し、前記一又は複数のスレッド（31）のうちの前記一つを解放することをさらに含む、態様1乃至4のいずれか1項に記載の方法。

(態様6)

前記タスクがメインタスクか否かを決定する前に、前記タスクが複数のサブタスクに分割可能であるか否かを決定し、前記タスクが分割可能である場合、前記タスクの前記複数のサブタスクをタスクのキュー（26）に挿入することをさらに含む、態様5に記載の方法。

(態様7)

タスクの実行において使用するコンピューティング装置であって、前記コンピューティング装置は、

タスクのキュー（26）を記憶するメモリ（12）装置と、

前記メモリ（12）装置に結合し、一又は複数のスレッドを有する少なくとも一つの処理エンジン（24）を含むプロセッサ（14）であって、前記少なくとも一つの処理エンジン（24）はタスクマネージャ（27）とキューマネージャ（25）を含み、前記キューマネージャ（25）はタスクの前記キュー（26）を管理するように構成されており、前記タスクマネージャ（27）は前記プロセッサ（14）に関連するタスク配分メトリックを決定し、前記タスクを前記一又は複数のスレッド（31）のうちの前記一つに割り当てるように構成されている前記プロセッサ（14）を含み、

前記タスクマネージャ（27）は、前記タスク配分メトリックが、前記プロセッサ（14）に関連する負荷バランスが第1閾値を下回ることを示す場合に、前記タスクが、前記一又は複数のスレッド（31）のうちの任意の一つに以前割り当てられた再入可能タスクであるか否かを決定し、前記タスクが再入可能タスクである場合、停止基準が満たされたか否かを決定し、停止基準が満たされておらず、前記タスクが再入可能タスクである場合に、前記タスクをタスクのキュー（26）に前記キューマネージャ（25）を介して再入させるように構成されており、

前記タスクマネージャ（27）は、タスク配分メトリックが、前記プロセッサ（14）に関連するコア（29）アフィニティが第2閾値を下回ることを示す場合に、前記タスクがメインタスクであるか否かを決定し、前記タスクがメインタスクでない場合に、停止基準が満たされたか否かを決定し、停止基準が満たされており、前記タスクがメインタスクではない場合に、前記タスクに関連する親タスクを一又は複数のスレッド（31）のうちの前記一つにブルするように構成されている、コンピューティング装置。

(態様8)

前記タスクマネージャ（27）がさらに、前記タスクが再入可能タスクでない場合に、前記タスクが複数のサブタスクに分解可能であるか否かを決定するようにさらに構成されており、前記タスクマネージャ（27）がさらに、前記タスクが分解可能である場合に、前記キューマネージャ（25）を介して、前記タスクの前記複数のサブタスクを前記キュー（26）に挿入するように構成されている、態様7に記載のコンピューティング装置。

(態様9)

前記タスクマネージャ（27）がさらに、前記複数のサブタスクのうちの一つを前記一又は複数のスレッド（31）のうちの2番目のスレッドに割り当てて、前記複数のサブタスクのうちの前記一つが再入可能タスクであるか否かを決定するように構成されており、

10

20

30

40

50

前記タスクマネージャ(27)がさらに、前記停止基準が満たされた時に、少なくとも一つの蓄積タスクを実行するように構成されている、態様7又は8に記載のコンピューティング装置。

(態様10)

前記タスクマネージャ(27)がさらに、前記タスクがメインタスクであるか否かを決定する前に、前記タスクが複数のサブタスクに分解可能であるか否かを決定するように構成されており、前記タスクマネージャ(27)がさらに、前記タスクが分解可能である場合に、前記キューマネージャ(25)を介して前記タスクの前記複数のサブタスクを前記キー(26)に挿入するように構成されており、前記タスクマネージャ(27)がさらに、前記親タスクを前記一又は複数のスレッド(31)のうちの前記一つにプルした後で、前記親タスクに関連する少なくとも一つの蓄積タスクを実行するように構成されている、態様7に記載のコンピューティング装置。

【符号の説明】

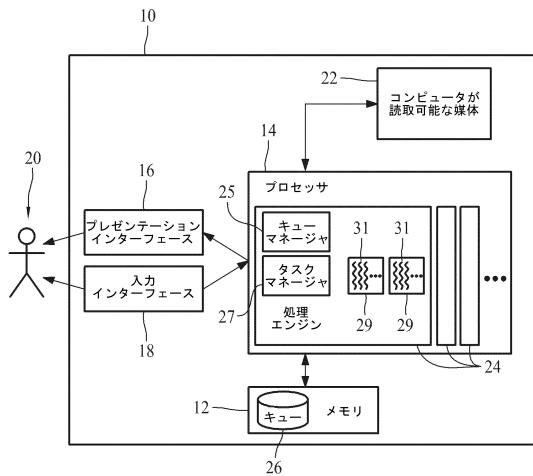
【0050】

10	コンピューティング装置
12	メモリ
14	プロセッサ
16	プレゼンテーションインターフェース
18	入力インターフェース
20	ユーザ
22	コンピュータが読取可能な媒体
24	処理エンジン
25	キューマネージャ
27	タスクマネージャ
29	処理コア
31	スレッド
100	方法
200	方法

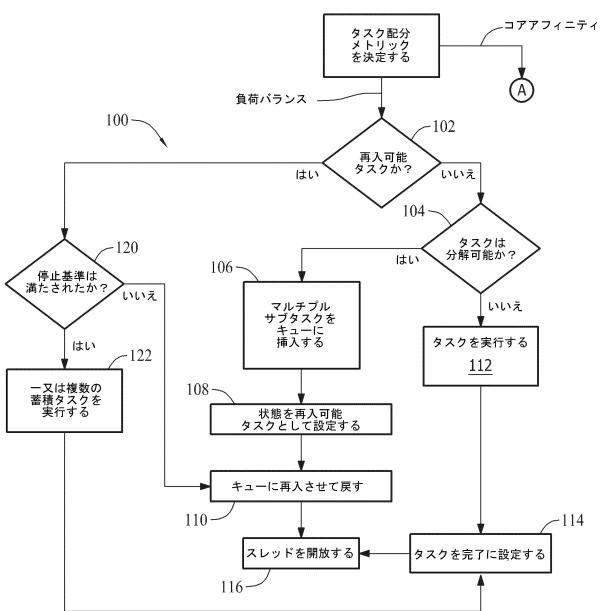
10

20

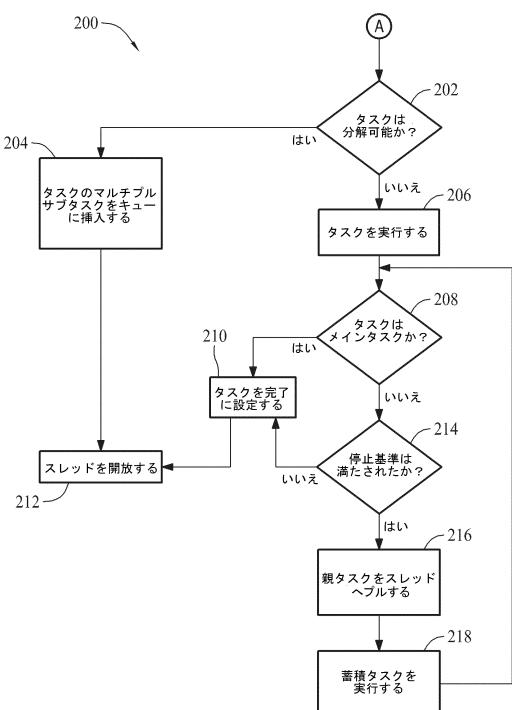
【図1】



【図2】



【図3】



フロントページの続き

(72)発明者 カッチア， デーヴィッド クリスチャン
アメリカ合衆国 ワシントン 98407, エバレット, 27番 ブレイス ウエスト 98
01

審査官 清木 泰

(56)参考文献 米国特許出願公開第2004/0054999(US, A1)
特開2008-047126(JP, A)
米国特許第05872972(US, A)
米国特許出願公開第2009/0187915(US, A1)

(58)調査した分野(Int.Cl., DB名)

G 06 F 9/46 - 9/54
G 06 F 15/16 - 15/177