

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-108917

(P2012-108917A)

(43) 公開日 平成24年6月7日(2012.6.7)

(51) Int.Cl.

G06F 9/50 (2006.01)

F I

G06F 9/46 462B

テーマコード (参考)

審査請求 未請求 請求項の数 12 O L (全 28 頁)

(21) 出願番号 特願2011-250635 (P2011-250635)  
 (22) 出願日 平成23年11月16日 (2011.11.16)  
 (31) 優先権主張番号 12/949, 494  
 (32) 優先日 平成22年11月18日 (2010.11.18)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 000005223  
 富士通株式会社  
 神奈川県川崎市中原区上小田中4丁目1番  
 1号  
 (74) 代理人 100070150  
 弁理士 伊東 忠彦  
 (74) 代理人 100146776  
 弁理士 山口 昭則  
 (72) 発明者 デベトロ・マシュー ジェイムス  
 アメリカ合衆国, カリフォルニア州 94  
 089, サニーヴェイル, ヒドゥン・レイ  
 ク・ドライブ 298番

(54) 【発明の名称】 ワークロード・メタデータの生成、分析、及び利用のための方法、プログラム、及びシステム

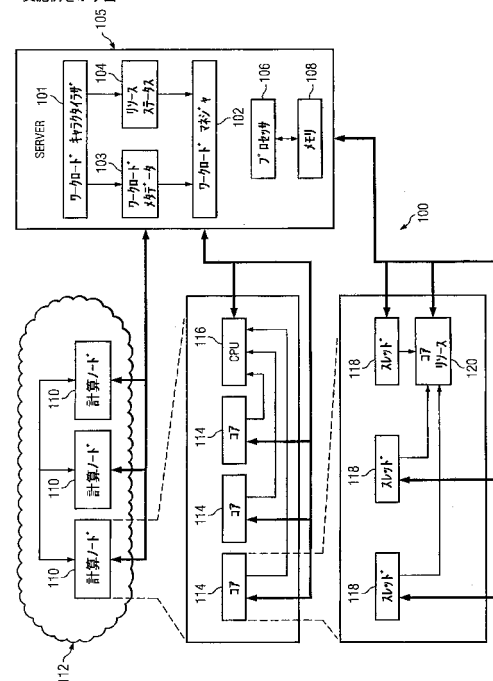
(57) 【要約】 (修正有)

【課題】ワークロード・リクエストの実行のために、リソースの割り当て及び優先付けを行う。

【解決手段】第1のワークロードのための第1のワークロード・メタデータと、第2のワークロードのための第2のワークロード・メタデータとの比較に基づいて、前記第1のワークロードと処理エンティティが利用できる計算リソースとの間でのリソース要求の潜在的な競合を特定し、前記処理エンティティのうちの1つによる実行のために前記第2のワークロードを割り当てる。ワークロードは、処理エンティティによって実行される。メタデータは、関連づけられたワークロードによって必要な計算リソースを特徴づける。前記第1のワークロードのメタデータが、前記第2のワークロード・メタデータより優先される。リソース要求の潜在的な競合は、前記第2のワークロードの前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在しない。

【選択図】 図1

計算ワークロード・メタデータの生成の分析及び利用のためのシステムの実施例を示す図



**【特許請求の範囲】****【請求項 1】**

計算リソースを管理する方法であって、

第 1 のワークロードのための第 1 のワークロード・メタデータを生成するステップと；

第 2 のワークロードのための第 2 のワークロード・メタデータを生成するステップであって；

前記第 1 のワークロード・メタデータ及び前記第 2 のワークロード・メタデータの各々は、関連づけられたワークロードによって必要な計算リソースを特徴づけ；

最初に、前記第 1 のワークロードのメタデータが、前記第 2 のワークロード・メタデータより優先され；かつ

前記第 1 のワークロード及び前記第 2 のワークロードは、一つ以上の処理エンティティによって実行される；

ステップと；

リソース・メタデータに対して前記第 1 のワークロード・メタデータ及び前記第 2 のワークロード・メタデータを比較するステップであって、前記計算リソースを特徴づけている前記リソース・メタデータは、一つ以上の前記処理エンティティにより利用可能である、ステップと、

リソース・メタデータに対するワークロード・メタデータの前記比較に基づいて、前記第 1 のワークロードと前記処理エンティティが利用できる前記計算リソースとの間でのリソース要求の潜在的な競合を特定するステップと；

前記処理エンティティのうちの一つによる実行のために前記第 2 のワークロードを割り当てるステップであって、リソース要求の前記潜在的な競合は、前記第 2 のワークロードの前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在しない、ステップと、

を有する方法。

**【請求項 2】**

前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータは、各々のワークロードに要求される計算リソースの相対的特徴を含む、請求項 1 記載の方法。

**【請求項 3】**

前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータは、各々のワークロードに要求される前記計算リソースの定量的特徴を含む、請求項 1 記載の方法。

**【請求項 4】**

前記一つ以上の処理エンティティは、計算ノードのプールを有する、請求項 1 記載の方法。

**【請求項 5】**

前記一つ以上の処理エンティティは、プロセッサ・コアのプールを有する、請求項 1 記載の方法。

**【請求項 6】**

前記一つ以上の処理エンティティは、プロセッサ・スレッドのプールを有する、請求項 1 記載の方法。

**【請求項 7】**

ワークロードのためのワークロード・メタデータを生成するステップは、

実行のために前記ワークロードをシステムにロードするステップと；

データの第 1 のサンプルによって前記ワークロードを実行するステップと；

前記ワークロードによって利用される前記計算リソースを測定するステップと；

を有する請求項 1 記載の方法。

**【請求項 8】**

ワークロードのためのワークロード・メタデータを生成するステップは、

データの第 2 のサンプルによって前記ワークロードを実行するステップと；

前記ワークロードによって利用される前記計算リソースを測定するステップと；

10

20

30

40

50

前記ワークロードによって使用された計算リソースを平均化するステップと；  
を更に有する請求項 7 記載の方法。

【請求項 9】

第 1 の処理エンティティ及び第 2 の処理エンティティに利用可能な計算リソースを特徴付けることによって、リソース・メタデータが生成され；

リソース・メタデータに対して前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータを比較するステップは、第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータを、前記第 1 の計算エンティティ及び前記第 2 の計算エンティティに利用可能な前記リソースに対して比較するステップを含み、

前記処理エンティティのうちの 1 つによる実行のために前記第 2 のワークロードを割り当てるステップは、

前記第 1 の処理エンティティが、前記第 2 のワークロードを実行するために、十分なリソースを持っているか否かを判断するステップと；

前記第 1 の処理エンティティが、不十分なリソースしか持たない場合、前記第 2 のワークロードを実行するために、前記第 2 の処理エンティティが十分なリソースを持つか否かを判断するステップと；

前記第 2 の処理エンティティが十分なリソースを持つ場合、前記第 2 の処理エンティティによる実行のために前記第 2 のワークロードを割り当てるステップと；

を含む、請求項 1 記載の方法。

【請求項 10】

第 1 の処理エンティティによる前記第 1 のワークロードの実行が、前記処理エンティティに対して、利用可能な計算リソースの所定のカテゴリの 100 パーセントを超えさせる場合、潜在的な競合が特定される、請求項 3 記載の方法。

【請求項 11】

コンピュータに、

第 1 のワークロードのための第 1 のワークロード・メタデータを生成するステップと；

第 2 のワークロードのための第 2 のワークロード・メタデータを生成するステップであって；

前記第 1 のワークロード・メタデータ及び前記第 2 のワークロード・メタデータの各々は、関連づけられたワークロードによって必要な計算リソースを特徴づけ；

最初に、前記第 1 のワークロードのメタデータが、前記第 2 のワークロード・メタデータより優先され；かつ

前記第 1 のワークロード及び前記第 2 のワークロードは、一つ以上の処理エンティティによって実行される；

ステップと；

リソース・メタデータに対して前記第 1 のワークロード・メタデータ及び前記第 2 のワークロード・メタデータを比較するステップであって、前記計算リソースを特徴づけている前記リソース・メタデータは、一つ以上の前記処理エンティティにより利用可能である、ステップと、

リソース・メタデータに対するワークロード・メタデータの前記比較に基づいて、前記第 1 のワークロードと前記処理エンティティが利用できる前記計算リソースとの間でのリソース要求の潜在的な競合を特定するステップと；

前記処理エンティティのうちの 1 つによる実行のために前記第 2 のワークロードを割り当てるステップであって、リソース要求の前記潜在的な競合は、前記第 2 のワークロードの前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在しない、ステップと、

を実行させるプログラム。

【請求項 12】

計算ワークロードを管理するシステムであって、

プロセッサと；

10

20

30

40

50

前記プロセッサに接続される計算機可読の媒体であって、コンピュータに、  
第1のワークロードのための第1のワークロード・メタデータを生成するステップと

;

第2のワークロードのための第2のワークロード・メタデータを生成するステップであ  
って;

前記第1のワークロード・メタデータ及び前記第2のワークロード・メタデータの各  
々は、関連づけられたワークロードによって必要な計算リソースを特徴づけ;

前記第1のワークロード及び前記第2のワークロードは、一つ以上の処理エンティテ  
ィによって実行される;

ステップと;

10

リソース・メタデータに対して前記第1のワークロード・メタデータ及び前記第2のワ  
ークロード・メタデータを比較するステップであって、前記計算リソースを特徴づけてい  
る前記リソース・メタデータは、一つ以上の前記処理エンティティにより利用可能である  
、ステップと、

リソース・メタデータに対するワークロード・メタデータの前記比較に基づいて、前記  
第1のワークロードと前記処理エンティティが利用できる前記計算リソースとの間でのリ  
ソース要求の潜在的な競合を特定するステップと;

前記処理エンティティのうちの一つによる実行のために前記第2のワークロードを割り  
当てるステップであって、リソース要求の前記潜在的な競合は、前記第2のワークロード  
の前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在  
しない、ステップと、

20

を実行させるプログラムを格納した媒体と;

を有するシステム。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本発明は、一般的に計算リソースの管理に関する。より詳細には、計算ワークロード・  
メタデータの生成、分析、及び利用のためのシステム及び方法に関する。

#### 【背景技術】

30

#### 【0002】

コンピュータのパフォーマンスを向上させるために、今日では、コンピュータのような  
システム及び電子デバイスは、巨大な計算電子デバイス、中央処理装置(CPU)、及び  
CPUコア等を利用している。

#### 【発明の概要】

#### 【発明が解決しようとする課題】

#### 【0003】

このようなリソースの管理をすることは、複雑であり、難しい作業である。ワークロ  
ード・リクエストの実行のために、このようなリソースの割り当て及び優先付けを行うよう  
構成されたソフトウェアは、現時点における要求に基づいて、計算デバイスを分散させる  
受け身的なアプローチが取られる。

40

#### 【課題を解決するための手段】

#### 【0004】

計算リソースを管理する方法は、第1のワークロードのための第1のワークロード・メ  
タデータを生成するステップと、第2のワークロードのための第2のワークロード・メ  
タデータを生成するステップと、リソース・メタデータに対して前記第1のワークロード・  
メタデータ及び前記第2のワークロード・メタデータを比較するステップとを有する。本  
方法は、リソース・メタデータに対するワークロード・メタデータの前記比較に基づいて  
、前記第1のワークロードと前記処理エンティティが利用できる前記計算リソースとの間  
でのリソース要求の潜在的な競合を特定するステップと、前記処理エンティティのうちの

50

1つによる実行のための前記第2のワークロードを割り当てるステップとを有する。前記第1のワークロード・メタデータ及び前記第2のワークロード・メタデータの各々は、関連づけられたワークロードによって必要な計算リソースを特徴づける。最初に、前記第1のワークロードのメタデータが、前記第2のワークロード・メタデータより優先される。前記第1のワークロード及び前記第2のワークロードは、一つ以上の処理エンティティによって実行される。前記計算リソースを特徴づけている前記リソース・メタデータは、前記処理エンティティが利用できる。リソース要求の前記潜在的な競合は、前記第2のワークロードの前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在しない。

【0005】

本発明及びその特徴及び効果の更なる詳細な理解のために、図面を参照しながら以下に説明する。

【図面の簡単な説明】

【0006】

【図1】計算ワークロード・メタデータの生成の分析及び利用のためのシステムの実施例を示す図である。

【図2】ワークロード・キャラクタライザによって特徴付けられ、システムの処理エンティティによりワークロードの優先付けを行うためのワークロード・マネージャによって利用される例示のシステムのリースを示す図である。

【図3】システムから発生したワークロードがどのように、非効率的にシステムの様々な処理ユニットに割り当てられるかを示す図である。

【図4】ワークロード・マネージャが、待ちのワークロードをジョブキューに割り当てる実施例の効率的な割り当て方式を示す図である。

【図5】ワークロード・マネージャが実行を優先付けするために、定性的又は相対的な手法によりワークロード・メタデータ及びリソース・ステータスを生成する実施例を示す図である。

【図6】ワークロード・キャラクタライザ及びワークロード・マネージャによる、定量的なカテゴリ分け及び優先付けの実施例を示す図である。

【図7】計算ワークロード・メタデータの生成、分析、及び利用のための方法の実施例を示す図である。

【発明を実施するための形態】

【0007】

更なる実施例においては、製造製品は、計算機可読媒体及び計算機実行可能な命令を含む。計算機実行可能な命令は、計算機可読媒体に格納される。この命令は、プロセッサによって読み取ることができる。この命令は、読み込まれ実行されると、プロセッサが、第1のワークロードのための第1のワークロード・メタデータを生成するステップと、第2のワークロードのための第2のワークロード・メタデータを生成するステップと、リソース・メタデータに対して前記第1のワークロード・メタデータ及び前記第2のワークロード・メタデータを比較するステップと、前記第1のワークロードと前記処理エンティティが利用できる前記計算リソースとの間でのリソース要求の潜在的な競合を特定するステップと、前記処理エンティティのうちの一つによる実行のための前記第2のワークロードを割り当てるステップとを実行する。前記第1のワークロード・メタデータ及び前記第2のワークロード・メタデータの各々は、関連づけられたワークロードによって必要な計算リソースを特徴づける。最初に、前記第1のワークロードのメタデータが、前記第2のワークロード・メタデータより優先される。前記第1のワークロード及び前記第2のワークロードは、一つ以上の処理エンティティによって実行される。前記計算リソースを特徴づけている前記リソース・メタデータは、前記処理エンティティが利用できる。リソース要求の前記潜在的な競合は、前記第2のワークロードの前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在しない。

【0008】

10

20

30

40

50

更にほかの実施例において、計算ワークロードを管理するシステムは、プロセッサと、前記プロセッサに結合された計算機可読媒体と、前記計算機可読媒体に格納された計算機実行可能な命令とを有する。この命令は、前記プロセッサによって読み取ることができる。この命令は、読み込まれ実行されると、プロセッサが、第1のワークロードのための第1のワークロード・メタデータを生成するステップと、第2のワークロードのための第2のワークロード・メタデータを生成するステップと、リソース・メタデータに対して前記第1のワークロード・メタデータ及び前記第2のワークロード・メタデータを比較するステップと、前記第1のワークロードと前記処理エンティティが利用できる前記計算リソースとの間でのリソース要求の潜在的な競合を特定するステップと、前記処理エンティティのうちの1つによる実行のための前記第2のワークロードを割り当てるステップとを実行する。前記第1のワークロード・メタデータ及び前記第2のワークロード・メタデータの各々は、関連づけられたワークロードによって必要な計算リソースを特徴づける。前記第1のワークロード及び前記第2のワークロードは、一つ以上の処理エンティティによって実行される。前記計算リソースを特徴づけている前記リソース・メタデータは、前記処理エンティティが利用できる。リソース要求の前記潜在的な競合は、前記第2のワークロードの前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在しない。

10

**【実施例】****【0009】**

図1は、計算ワークロード・メタデータの生成、分析、及び利用のためのシステム100の実施例を示している。システム100は、ワークロード・キャラクタライザ101及びワークロード・マネージャ102を含んでもよい。システム100は、一つ以上の計算ノード110を含んでもよい。計算ノード110はプロセッサを含んでもよい。そして、例えば、電子デバイス、コンピュータ、サーバ、又はマザーボードにインプリメントされてもよい。このようなデバイスは複数の計算ノード110（例えばラックマウント・サーバ構成）を含んでもよい。そして、複数のブレード・サーバ・ボードを含む。そして、各々のボードが計算ノードを含む。マルチプロセッサ環境において、各々のプロセッサは、計算ノード110であってもよい。システム110は、一つ以上のコア114を含んでもよい。コア114は、命令を読み込めかつ実行するよう構成されたプロセッサの一部にインプリメントされてもよい。プロセッサ、又は、計算ノード110は、一つ以上のコア114を含んでもよい。システム110は、一つ以上のスレッド118を含んでもよい。スレッド118は、一連の命令でインプリメントされてもよい。この一連の命令は、例えば、コア114又は計算ノード118で実行されるオペレーティングシステムによってスケジューリングされ得る。コア114は、一つ以上のスレッド118を含んでもよい。

20

30

**【0010】**

ワークロード・キャラクタライザ101は、例えば計算ノード110、コア114、又は、スレッド118等の、一つ以上の計算又は処理エンティティのオペレーションを評価するように構成されてもよい。処理エンティティによって処理される一組の計算、又は、処理タスクは、ワークロードとして束ねられてもよい。計算、又は、処理エンティティは、並行に一つ以上のワークロードを実行するように構成されてもよい。このような構成によると、並行的に処理されるワークロードが共有しなければならない計算リソースの不足を生じさせる場合がある。ワークロード・キャラクタライザ101は、処理エンティティによって処理されるワークロードのために必要なリソースを決定するように構成されてもよい。ワークロード・キャラクタライザ101は、所定の利用限度で全ての処理エンティティに利用可能な全てのリソースを評価するよう構成されてもよい。ワークロード・キャラクタライザは、システム100の処理エンティティの各種レベルのワークロードに対して必要とされる必要な計算リソースを測定してもよい。ワークロード・キャラクタライザ101は、このような測定の結果を記録し、かつ特徴メタデータの形でそれらをワークロードにアタッチするよう構成されてもよい。ワークロード・マネージャ102は、処理エンティティによる実行のためのワークロードを割り当て、優先付けするために、このような

40

50

記録されたメタデータ情報を利用するよう構成されてもよい。一つの実施例において、ワークロード・マネージャ 102 は、処理エンティティによって同時実行するための、ワークロードを割り当て、かつ、優先付けしてもよい。

#### 【0011】

ワークロード・キャラクタライザ 101 は、システム 100 のワークロードのリソース要求を特徴付けるための、いかなるスクリプト、実行形式、モジュール、ライブラリ、あるいはその他実行可能な適切なデジタルエンティティによってインプリメントされてもよい。ワークロード・マネージャ 102 は、ワークロードを割り当て、かつ優先付けするための、いかなるスクリプト、実行形式、モジュール、ライブラリ、あるいはその他実行可能なデジタルエンティティによってインプリメントされてもよい。ワークロード・キャラクタライザ 101 は、ワークロード・キャラクタライザ 101 をホスティングすることに適したシステム 100 のいかなる電子デバイスに存在してもよい。ワークロード・マネージャ 102 は、ワークロード・マネージャ 102 をホスティングすることに適したシステム 100 のいかなる電子デバイスに存在してもよい。実施例において、ワークロード・キャラクタライザ 101 及びワークロード・マネージャ 102 は、同じ電子デバイスに存在してもよい。別の実施例においては、ワークロード・キャラクタライザ 101 及びワークロード・マネージャ 102 は、異なる電子デバイスに存在してもよい。ワークロード・キャラクタライザ 101 及びワークロード・マネージャ 102 は、一つ以上のサーバ 105 に存在してもよい。サーバ 105 は、システム 100 の電子デバイスによってインプリメントされてもよい。サーバ 105、又はワークロード・キャラクタライザ 101 又はワークロード・マネージャ 102 が存在している電子デバイスは、メモリ 108 に接続されたプロセッサ 106 を含んでもよい。ワークロード・キャラクタライザ 101 は、プロセッサ 106 によって実行され、かつ、メモリ 108 に記憶されるよう構成されてもよい。ワークロード・マネージャ 102 は、プロセッサ 106 によって実行され、かつメモリ 108 に記憶されるよう構成されてもよい。実施例において、ワークロード・キャラクタライザ 101 及びワークロード・マネージャ 102 は、同じ実行形式、スクリプト、ファイル、モジュール又はライブラリにインプリメントされてもよい。別の実施例において、ワークロード・キャラクタライザ 101 及びワークロード・マネージャ 102 は、異なる実行形式、スクリプト、ファイル、モジュール又はライブラリにインプリメントされてもよい。

10

20

30

#### 【0012】

プロセッサ 106 は、例えばマイクロプロセッサ、マイクロコントローラ、デジタルシグナルプロセッサ (DSP)、特定用途向け集積回路 (ASIC)、又は、プログラム命令及び / 又はプロセス・データを解釈し、及び / 又は実行するよう構成された他のいかなるデジタル、又はアナログ回路をも含んでもよい。いくつかの実施例では、プロセッサ 106 は、メモリ 108 に記憶されたプログラム命令及び / 又はプロセス・データを解釈してもよく、及び / 又は実行してもよい。メモリ 106 は一部又は全てがアプリケーション・メモリ、又はシステムメモリ、又はその両方で構成されてもよい。メモリ 106 には、一つ以上のメモリーモジュールを持つよう構成された、いかなるシステム、デバイス、又は、装置を含んでもよい。各々のメモリーモジュールは、一定期間プログラム命令及び / 又はデータ (例えばコンピュータ可読のメディア) を保持するよう構成される、いかなるシステム、デバイス、又は、装置を含んでもよい。

40

#### 【0013】

ワークロード・キャラクタライザ 101 は、ワークロード・メタデータ 103 を生成するよう構成されてもよい。ワークロード・キャラクタライザ 101 は、システム 100 のリソースの状態を分析し、かつリソース・ステータス 104 において特徴づけるよう構成されてもよい。ワークロード・マネージャ 102 は、ワークロード・メタデータ 103 及び / 又はリソース・ステータス 104 にアクセスするよう構成されてもよい。ワークロード・マネージャは、システム 100 の処理エンティティ上のワークロードの実行を優先させ、かつ割り当てるために、ワークロード・メタデータ 103 及びリソース・ステータス 104 を使用するよう構成されてもよい。

50

## 【 0 0 1 4 】

ワークロード・キャラクタライザ 1 0 1 は、システム 1 0 0 の各種レベルから処理エンティティのワークロード及びリソースを特徴づけるように構成されてもよい。例えば、ワークロード・キャラクタライザ 1 0 1 は、ワークロード、及び一つ以上の計算ノード 1 1 0 のリソース特徴を分析するように構成されてもよい。計算ノード 1 1 0 は、ネットワーク 1 1 2 を介して通信するように構成されてもよい。ネットワーク 1 1 2 は、通信のためのいかなる適切なネットワークで実行されてもよい。例えば、インターネット、イントラネット、広域ネットワーク、ローカル・エリア・ネットワーク、バックホールネットワーク (back-haul-networks)、ピアツーピア・ネットワーク、又は、これらのいかなる組合せであってもよい

10

別の実施例において、ワークロード・キャラクタライザ 1 0 1 は、与えられた処理ユニット (例えば計算ノード 1 1 0) の一つ以上のコア 1 1 4 のワークロード及びリソースを分析するように構成されてもよい。このようなコアは単一の計算ノード 1 1 0 の各々で動作するものであってもよい、又は、複数のコアが単一の計算ノードで動作してもよい。さらに別の実施例では、ワークロード・キャラクタライザ 1 0 1 は、一つ以上のスレッド 1 1 8 のワークロード及びリソースを分析するように構成されてもよい。このようなスレッドは単一のコア 1 1 4 の各々で動作してもよい。あるいは、複数のスレッドは単一 CPU で動作してもよい。このようなスレッドは、コア 1 1 4、CPU、又は、他の処理エンティティによって並行的な実行として同時に動作していてもよい。

20

## 【 0 0 1 5 】

各々のこれらの実施例において、処理エンティティは、システム・リソースとして有限値を有している。このようなリソースは、リソースの共有を試みる処理エンティティによって異なってもよい。例えば、コア 1 1 4 は、CPU 1 1 6 及び他の計算ノード 1 1 0 のリソースへのアクセスを共有しようとしてもよい。他の例では、スレッド 1 1 8 は、コア・リソース 1 2 0、又は、コア 1 1 4 の他の部分を共有するように構成されてもよい。ワークロード・キャラクタライザ 1 0 1 及びワークロード・マネージャ 1 0 2 は、コンピュータ計算ノード 1 1 0、コア 1 1 4、スレッド 1 1 8、各々のレベル、例えば CPU 1 1 6、又は、コア・リソース 1 2 0 のリソースに通信接続されてもよい。ワークロード・キャラクタライザ 1 0 1 は、所定のレベルの利用可能リソースを分析するように構成されてもよい。例えば、ワークロード・キャラクタライザ 1 0 1 は、計算ノード 1 1 0 が利用できるリソースを分析するように構成されてもよい。他の例では、ワークロード・キャラクタライザ 1 0 1 は、コア 1 1 4 (例えば CPU 1 1 6) が利用できるリソースを分析するように構成されてもよい。他の例では、ワークロード・キャラクタライザ 1 0 1 は、スレッド 1 1 8 (例えばコア・リソース 1 2 0) が利用できるリソースを分析するように構成されてもよい。ワークロード・キャラクタライザは、リソース・ステータス 1 0 4 にこの分析の結果を蓄積するように構成されてもよい。

30

## 【 0 0 1 6 】

ワークロード・マネージャ 1 0 2 は、所定のレベルの処理エンティティの実行の優先順位を決定するために、ワークロード・メタデータ 1 0 3 及びリソース・ステータス 1 0 4 にアクセスするように構成されてもよい。

40

## 【 0 0 1 7 】

図 2 は、ワークロード・キャラクタライザ 1 0 1 によって特徴づけられ、かつシステム 1 0 0 の処理エンティティによってワークロードの実行を優先させるためにワークロード・マネージャ 1 0 2 によって利用され得るシステム・リソースの具体例である。このようなリソースは、処理エンティティの性質に従って、異なるレベルに分けられてもよい。例えば、計算ノードのリソース 2 0 2 は、計算ノード 1 1 0 が利用できるシステム・リソースを示してもよい。他の例では、CPU リソース 2 1 2 は、コア 1 1 4 が利用できるリソースを示してもよい。さらに別の例では、コア・リソース 2 2 2 は、スレッド 1 1 8 によって実行できるリソースを示してもよい。

50

## 【 0 0 1 8 】

計算ノードのリソース 202 は、実行するために、計算ノード 110 に役立ついかなるリソースを含んでもよい。このような計算ノードのリソースは、例えば、利用できるシステムメモリ 204、ノード間バンド幅 205、与えられた計算ノード 206 の CPU 使用 (cpu usage)、パワー使用限度 207 及び / 又は I/O 接続 208 を含んでもよい。

#### 【0019】

CPU リソース 212 は、実行するために、コア 114 に適するいかなるリソースを含んでもよい。このようなリソースは、以下を含むが、これに限定されるものではない。すなわち、RAM と CPU とのバンド幅 214、CPU 間バンド幅 215、共用キャッシュ 216、メモリ使用状況 217、パワー使用限度 218 及び / 又はアドレス変換バッファ (「TLB」) 219 が挙げられる。CPU リソース 212 は、与えられた計算ノード 110 の CPU 使用状況 206 を示してもよい。

#### 【0020】

コア・リソース 222 は、実行するために、スレッド 118 に適したいいかなるシステム・リソースを含んでもよい。このようなリソースには、浮動小数点ユニット 224、整数ユニット 225、コア間バンド幅 226、パワー使用限度 227、さまざまなレジスタ 228、及びコア占有キャッシュ 229 が含まれ得る。なお、これらに制限されない。コア・リソース 222 は、CPU リソース 212 の与えられたコアのメモリ使用状況 217 を形成してもよい。

#### 【0021】

図 3 は、システムから発生したワークロードがどのように、非効率的にシステムの様々な処理ユニットに割り当てられるかを示す図である。ジョブスケジューラキュー 302 は、一つ以上のジョブ 310、又は、システム 100 で実行される予定のワークロードを含んでもよい。ジョブスケジューラキュー 302 は、ワークロード・マネージャ 102 によって維持されてもよい。図 3 の実施例において、ジョブスケジューラキュー 302 は、一つ以上の利用できる CPU 116 上での実行のためのさまざまなワーク・ジョブを含んでもよい。CPU<sub>0</sub>、CPU<sub>1</sub> 及び CPU<sub>2</sub> の 3 台の CPU が実行に利用できる。各々の CPU は、関連づけられたジョブキュー 304 - 308 を有してもよい。例えば、CPU<sub>0</sub> は CPU ジョブキュー # 1 304 に対応してもよい。CPU<sub>1</sub> は CPU ジョブキュー # 2 306 に対応してもよい。そして、CPU<sub>2</sub> は CPU ジョブキュー # 3 308 に対応してもよい。各々のジョブキュー 304 - 308 は、一つ以上のワークロードを有し、指定された CPU で実行される。ワークロード・マネージャ 102 は、ジョブスケジューラキュー 302 から、一つ以上のワークロード 312 を利用できるジョブキュー 304 - 308 に割り当てるように構成されてもよい。ワークロード・マネージャがワークロード・メタデータ 103、又はリソース・ステータス 104 にアクセスしない場合、ワークロード・マネージャ 102 は、特定のジョブが処理エンティティの、相当量のリソース、プールされた利用可能な特定のリソースにストレスを与え、これを要求している情報を持たないこととなる。例えば、いくつかのワークロード 310 は特に CPU と RAM との間のバンド幅にストレスを与えてもよい。他のものは共用キャッシュにストレスを与えてもよい。そして、他のものは CPU 間接続にストレスを与えてもよい。更に、各々の CPU ジョブキュー 302 - 308 には、異なるタイプのワークロードがロードされてもよい。CPU ジョブキュー 304 - 308 においてすでにロードされるワークロードの性質、及びジョブスケジューラキュー 310 から割り当てられるワークロード 312 の性質に関する情報なしでは、ワークロード・マネージャ 102 は、ジョブスケジューラキュー 310 から、所定の CPU ジョブキューに、ワークロード 312 の効果的な割当てをすることができない。例えば、CPU ジョブキュー 304 - 308 の中で、CPU ジョブキュー # 2 は、実行に指定された 2 つのワークロードしか持っていない。CPU ジョブキュー # 2 は、待機中ワークロードが最少の数であるため、次のワークロードの割り当てが受信されるであろう。しかしながら、この両者のワークロードは、CPU と RAM との間のバンド幅をかなり消費するワークロードである。ワークロード・スケジューラ 302 は、CPU と RAM との

10

20

30

40

50

間のバンド幅をかなり消費するワークロード310にどのCPUジョブキューを割り当てるべきかについて決めるが、ワークロード・マネジャ102は、CPUジョブキュー#1、又は、CPUジョブキュー#3の代わりにこのようなワークロードをCPUジョブキュー#2に送信する。これは、非効率的な割り当てである。そのようなワークロードがCPUジョブキュー#2よりも大きなワークロードであり、CPUジョブキュー#1及び#3は、ワークロード312を効率的に処理できるにもかかわらず、ワークロード・マネジャ102は、CPUジョブキュー#1及び#3が実行のための3つの待機中ワークロードを有することしか判断できない。

#### 【0022】

図4は、ワークロード・マネジャ102が待機中ワークロードをジョブキューに割り当てる効果的な割り当て方式についての実施例を示している。この実施例において、ワークロード・マネジャ102は、例えばワークロード・メタデータ103及びワークロード・マネジャ102がワークロード310をジョブキュー304-308に効率的に割り当てるリソース・ステータス104の情報にアクセスする。例えば、CPUジョブキュー#2は、CPUジョブキュー#1又はCPUジョブキュー#3よりも多い3つの待機中のワークロードを持っているとする。ワークロード・マネジャ102は所定のワークロード310を、いずれのジョブキューに割り当てるべきかについて決定するように構成され、割り当の際、リソースの競合を回避する。例えば、ワークロード310は、重要なCPUとRAMとの間のバンド幅リソースを消費する。ワークロード・メタデータ103及びリソース・ステータス104の情報を使用することにより、ワークロード・マネジャ102は、ワークロード310のメタデータの性質、これに加えて各々のジョブキュー304-308に待機中のワークロードの性質を確認し得る。その結果として、ワークロード・マネジャ102は、CPUジョブキュー#2には、CPUジョブキュー#1、又は、CPUジョブキュー#3より多くのジョブが待機しているにもかかわらず、待機中ワークロード310をCPUジョブキュー#2に割り当てるように構成されてもよい。CPUジョブキュー#1及び#2の両方が既に、実行においてCPUとRAMとの間のバンド幅にストレスを与えるような待機中ワークロードを有しているため、ワークロード・マネジャ102は上述のような割り当てを与える構成としてもよい。実施例において、ワークロード310は、並行して処理エンティティによって実行されてもよい。

#### 【0023】

図1に戻る。ワークロード・キャラクタライザ101は、システム100の処理エンティティのワークロードを特徴づけるよう、いかなる適切な方法によって構成されてもよい。図2で示すリソースだけでなく、ワークロード・キャラクタライザは、いかなる適切なリソースを特徴づけてもよい。一つの実施例において、ワークロード・キャラクタライザ101は、システム100の処理エンティティから、待機中ワークロードを、ワークロードのリソースの利用のプロファイルを作り特徴づけるための適切な電子デバイス上にロードするように構成されてもよい。ワークロード・キャラクタライザ101は、ワークロードのリソースの消費レベルを平均するために異なるサンプルデータを使用して、特徴づけられる各々のワークロードを複数回実行するように構成されてもよい。ワークロード・キャラクタライザ101は、システム100の仕様に基づいてこのような特徴づけを実行するように構成されてもよい。例えば、ワークロード・キャラクタライザ101が80%のCPUメモリ帯域幅を使用する所定のワークロードを分析した場合、ワークロード・キャラクタライザ101は、このように判断をするように構成されてもよい。すなわち、利用できるCPUメモリ帯域幅が1秒につき20ギガビット(「Gbps」)であり、ワークロードは16GB/sのCPUメモリ帯域幅を使用していた場合が挙げられる。このような技術を用いて、ワークロード・キャラクタライザ101は、所定のワークロードの必要性に基づいて、ワークロードを特徴づけるように構成されてもよい。ワークロード・キャラクタライザ101は、ボトルネックの原因となるリソースを特定するように構成されてもよい。

#### 【0024】

実施例において、ワークロード・キャラクタライザ 101 は、さまざまなワークロードの要求、及び定性的又は相対的な観点から処理エンティティが利用できるリソースを分析するように構成されてもよい。このような実施例では、ワークロード・マネージャ 102 は、ワークロード・メタデータ 103 及びリソース・ステータス 104 からこのような相対的情報に基づいてワークロードの実行を優先させるように構成されてもよい。別の実施例において、ワークロード・キャラクタライザ 101 は、さまざまなワークロードの要求及び処理エンティティが利用できるリソースを定量的方法で分析するように構成されてもよい。このような実施例では、ワークロード・マネージャ 102 は、ワークロード・メタデータ 103 及びリソース・ステータス 104 から定量的情報に基づいてワークロードの実行を優先させるように構成されてもよい。

10

#### 【0025】

図5は、ワークロード・キャラクタライザ 101 による、定性的又は相対的な手法によるワークロード・メタデータ 103 及びリソース・ステータス 104 の生成の例を示す。その結果、ワークロード・マネージャ 102 は実行を優先させてもよい。コア・オペレーション・ステータス 502 は、システム 100 の所定のレベルで利用できるリソースの状態を示してもよい。コア・オペレーション・ステータス 502 は、処理エンティティによってすでに実行に割り当てられたワークロードのリソース要求を含んでもよい。ワークロードキュー 504 は、システム 100 の所定のレベルのさまざまな処理エンティティによって実行のために要求されるワークロードを含んでもよい。ワークロードキュー 504 は、一つ以上のワークロード 506 を含んでもよい。所定のワークロード 506 の、又は、コア・オペレーション・ステータス 502 のリソース情報は、ワークロード・キャラクタライザ 101 によって設定されるように構成されてもよい。このようなリソース情報は、ワークロード・メタデータ 103 及びリソース・ステータス 104 に対応してもよい。ワークロード・キャラクタライザ 101 は、定性的に所定のワークロード 506 の情報を特徴づけるように構成されてもよい。ワークロード・キャラクタライザ 101 は、例えば、ベンチマーク、他のワークロードの情報、又は、利用可能なシステム・リソースに関連して所定のワークロードの情報を特徴づけるように構成されてもよい。ワークロード・キャラクタライザ 101 は、所定のワークロードのリソース要求をいかなる適切な分割 (segmentation) によって特徴づけるように構成されてもよい。

20

#### 【0026】

例えば、ワークロード・キャラクタライザ 101 は、所定のワークロード 506 により必要とされるリソースを、一つ以上の可能な利用のカテゴリ (例えば、重い利用 (heavily usage)、軽い利用 (light usage)) に特徴付けるよう構成されてもよい。同様に、ワークロード・キャラクタライザ 101 は、利用に関する一つ以上の可能なカテゴリに、一つ以上の処理エンティティのオペレーションの状態を特徴づけるように構成されてもよい。ワークロード・キャラクタライザ 101 は、例えば、コア・オペレーション・ステータス 502 のコアにおいて利用できるリソースを、軽い利用、又は重い利用に特徴づけるように構成されてもよい。ワークロード・キャラクタライザ 101 は、ワークロードキュー 504 の各々のワークロード 506 のために、このような特徴を決定するように構成されてもよい。例えば、図5で、ワークロード・キャラクタライザ 101 は、コア・オペレーション・ステータス 502 によって示される所定のコアが、浮動小数点ユニット 224、整数ユニット 225、パワー利用限度 227 に関連づけられたそのリソースに関して、軽く利用されていると決定するように構成されてもよい。ワークロード・キャラクタライザ 101 は、そのコア間バンド幅 226、さまざまなレジスタ 228、及びコア占有キャッシュ 229 に関して、コアのリソースが重く利用されていると決定するように構成されてもよい。同様に、この例において、ワークロード・キャラクタライザ 101 は、ワークロード A は不動小数点ユニット 224、コア間バンド幅 226、及びコア占有キャッシュ 229 を重く利用すると決定するよう構成されてもよく、その他のリソースに関しては軽く利用していると決定してもよい。ワークロード・キャラクタライザ 101 は、ワークロード B、C 及び D に対して、同様の決定を行うよう構成されても

30

40

50

よい。

【0027】

ワークロード・マネージャ102は、例えばコア・オペレーション・ステータス502，及び例えばワークロードキュー404のようなワークロードキューから、最適化された、又は向上した所定のコアにおいて実行されるワークロードの割り当てを決定するよう構成されてもよい。ワークロード・マネージャ102は、所定の処理エンティティにおいてワークロードの実行を最適化するためのいかなる適切な方式を採用するよう構成されてもよい。一つの実施例において、ワークロード・マネージャ102は、重いリソースの利用特徴を有する複数のワークロード506を、所定の処理エンティティによる実行に割り当てることを回避するよう構成されてもよい。別の実施例において、ワークロード・マネージャ102は、所定のリソース・カテゴリの重いリソースの利用特徴を持つワークロード506を、その特徴に対するリソースがすでに重く圧迫されているコアによる実行に割り当てることを回避するよう構成されてもよい。別の実施例において、ワークロード・マネージャ102は、処理エンティティの実行のためのボトルネックを発生させることを回避するためにワークロードキュー504の一つ以上のワークロード506を評価するよう構成されてもよい。例えば、コア・オペレーション・ステータス502により示されるリソース・ステータス、及びワークロード506により示されるリソース要求が与えられた場合、システム100のより効率的な実行を提供するために、ワークロード・マネージャ102は、ワークロードA、又はワークロードBを送る前にワークロードC、又はワークロードDを送るよう構成されてもよい。ワークロード・マネージャ102は、ワークロードBをワークロードCと比較するよう構成されてもよい。ワークロードB及びワークロードCは両方とも数値演算プロセッサの重い利用を必要とする。しかし、ワークロードBはまた、コア占有キャッシュ229の重い利用を要求する。

ワークロードBが、初期の優先順位に関してワークロードキュー504においてワークロードCより前に位置するが、コア占有キャッシュの重い利用を行うワークロードBの要求がコア・オペレーション・ステータス502に示す利用可能リソースと競合を起こすため、ワークロード・マネージャ102は、ワークロードCの実行を優先させるよう構成されてもよい。特に、コアのコア占有キャッシュ229は、すでに重く利用される。

【0028】

図6は、定量的分類の例、及びワークロード・キャラクタライザ101及びワークロード・マネージャ102による優先順位である。所定の処理エンティティのリソースの利用ステータス、又はシステム100のレベルは、ノードのオペレーション・ステータス602によって反映されてもよい。

ノードのオペレーション・ステータス602は、処理エンティティによってすでに実行に割り当てられるワークロードのリソース要求を含んでもよい。ノード・オペレーション・ステータス602は、リソース(例えばノード間バンド幅205、CPU利用206、パワー利用限度207、利用できるシステムメモリ204及びI/O接続208)を反映した領域を含んでもよい。ノード・オペレーション・ステータス602は、このようなリソースの各々の利用の現在のレベルの表示、及びこのようなリソースの各々の最大能力の表示を含んでもよい。例えば、CPU<sub>0</sub>の現在の利用は85%、CPU<sub>1</sub>は0%、ノード間バンド幅2GB/s、パワー利用300ワット、利用できるシステムメモリ384ギガバイト、及びI/O接続5GB/sである。ワークロード・キャラクタライザ101は、例えば、一つ以上の計算ノード110のオペレーションに基づいて、ノード・オペレーション・ステータス602を決定するよう構成されてもよい。システム100の他の処理エンティティのためのオペレーション・ステータスを作成するよう構成されてもよい。

【0029】

ワークロード・キャラクタライザ101は、ワークロードキュー604に存在する1つ以上のワークロード606を特徴付けるよう構成されてもよい。ワークロードキュー604は、システム100のさまざまな処理エンティティによる実行のために待機中の、システム100のさまざまなワークロードを優先させるよう構成されるデータ構造であって

10

20

30

40

50

もよい。所定のワークロード606に対応するワークロードキュー604の各々のエンタリは、システム・リソースを指定する領域を含んでもよい。これは、ノード・オペレーション・ステータス602における指定に類似している。このような指定は、所定のワークロードのリソース要求を反映している。例えば、ワークロードAは、70%CPUのリソース、10GB/sのノード間バンド幅、最小のパワー利用、256ギガバイトのシステムメモリ及び1.5GB/sのI/O接続を必要としている。

【0030】

ワークロード・マネージャ102は、実行可能な利用できるそれらのリソース要求及びリソースと比較して、ワークロードキュー604の要素の優先順位を評価するように構成されてもよい。ワークロード・マネージャ102は、さまざまなワークロード606を実行するためのシステム・リソースの最も効率的な利用を決定するために、いかなる適切な優先順位メカニズムを利用するように構成されてもよい。一つの実施例において、ワークロード・マネージャ102は、ワークロード606を、ワークロードを適切に実行するための容量が存在するかどうかに基づいて、所定の処理エンティティによる実行に割り当てるように構成されてもよい。例えば、ワークロード・マネージャ102は、ノード・オペレーション・ステータス602において示されるように、利用可能リソースと比較してワークロードAを評価してもよい。このような実施例において、ワークロード・マネージャ102は、ノードが、CPUリソース、I/O接続及びパワー利用に関してワークロードAを実行する能力を有するが、リソースは、ノード間バンド幅及びシステムメモリに関しては、ワークロードAを実行するために十分ないと決定するように構成されてもよい。これに対して、ワークロード・マネージャ102は、ノード・オペレーション・ステータス602において示されるように、十分なリソースが利用できるので、ワークロードB及びワークロードCをワークロードAの前に送るよう決定するように構成されてもよい。このような実施例において、ワークロードBはCPU<sub>1</sub>によって実行に割り当てられ、かつワークロードCはCPU<sub>0</sub>によって実行が割り当てられてもよい。ワークロード・マネージャ102はワークロードDがCPU<sub>0</sub>上での実行のために送信されてもよいと決定するように構成されてもよい。しかしながら、ワークロードB、又はワークロードCのどちらとも一緒には、実行されないようにしてもよい。

10

20

【0031】

ワークロード・マネージャ102は、定性的、相対的、又は定量的評価のいずれの場合であっても、所定のワークロードキューのワークロードの当初の割り当てを考慮するように構成されてもよい。例えば、ワークロードB及びCが当初からワークロードDより高い優先順位でワークロードキュー504において配置されたために、ワークロードB及びワークロードCはワークロードDの前にコアによって実行されるよう割り当てられてもよい。

30

【0032】

例えば図2に示すような所定のリソースを特徴づけるために利用される容量及び容量の単位は、システム100の特定の特徴に従って異なってもよい。容量の単位、最大容量、最大容量に影響を与えるファクタ及び利用に関して、表1ないし表3に示す。

【0033】

【表 1】

リソース	容量単位	最大値の例	最大値に与えるファクタ
CPU 利用 (per CPU - CPU <sub>0</sub> , CPU <sub>1</sub> , ... CPU <sub>N</sub> )	Percent Utilization	100%	クロックレート、特定のデザイン、CPUアイドルリング
ノード間バンド幅	Gigabits per second (GB/s)	10 gbps	接続技術、チャネル数
利用可能なシステムメモリ	Gigabytes (GB)	512 GB	特定のデザイン、インストールされたモジュール数
パワー利用制限	Watts (W)	600 W	リソースアイドルリング、特定のデザイン
I/O接続	GB/s	5 gbps	I/Oチャネル数、特定の技術

10

20

表 1 は、システムを形成している複数ノードで共有される多ノード・システムのサンプル・リソースを示す。このようなリソースに対するメタデータは、ワークロード・マネージャ 102 によって決定されてもよい。そして、その結果として、このようなメタデータは、オペレーション、又はノードに対するワークロードの、又はこのようなノードによる共有リソースへのアクセスの割当てを優先させるために利用されてもよい。

## 【0034】

表 1 において、CPU 当たりの CPU 利用は、最大の利用を 100% として、パーセントでの利用として測定されてもよい。利用に影響を与えるファクタ及び最大値は、クロックレート、特定のプロセッサデザイン、及び CPU アイドリングにかかる時間を含んでもよい。ノード間バンド幅は、GB/s で測定されてもよい。例えば、最大 10 GB/s である。ノード間バンド幅に影響を与えるファクタとして、接続に用いられる技術のタイプ（例えば、インフィニバンド (Infiniband)、又は 10 ギガバイトのイーサネット (登録商標)）、及び利用できるチャネル数が含まれてもよい。利用できるシステムメモリは、ギガバイトで計測されてもよい。例えば 512 ギガバイトである。利用できるシステムメモリは、システム 100 の特定の設計、インストールされているメモリ数に影響される。パワー利用制限は、ワットで計測できる。例えば最大 600 ワットが挙げられる。パワー利用制限は、リソースのアイドルリング、リソース利用、及びシステム 100 の特定の設計に影響される。I/O 接続リソースは、GB/s によって計測されてもよい。例えば、最大 I/O 接続としては、5 GB/s が挙げられる。I/O 接続リソースは、入出力チャネル数、及び入出力接続に用いられる技術により制限される。

30

40

## 【0035】

【表 2】

リソース	容量単位	最大値の例	最大値に与えるファクタ
コア利用 (per Core - Core <sub>0</sub> , Core <sub>1</sub> , ... Core <sub>N</sub> )	Percent Utilization	100%	クロックレート、特定のデザイン、コアアイドルリング
CPU間バンド幅	GB/s	30 GB/s	クロックレート、特定のデザイン
CPU—RAM間バンド幅	GB/s	20 GB/s	クロックレート、メモリチャネル数、特定のデザイン
共有キャッシュ	Megabytes (MB)	12 MB	特定のデザイン
パワー利用制限	W	125 W	クロックレート、ソース電圧、特定のデザイン、コアアイドルリング
TLB	# of entries	1000	特定のデザイン

10

20

表 2 は、処理ユニットのサンプル・リソースを示し、マルチコア処理ユニットにおいて、複数のコアによって共有される。このようなリソースに対するメタデータは、ワークロード・マネージャ 102 によって決定されてもよい。そして、このようなメタデータは、オペレーション、コアに割り当てるワークロード、又はコアに共有されるリソースへのアクセスの割り当ての優先順位を決めるために利用されてもよい。

## 【0036】

表 2 において、コア当たりで決定されるコア利用は、100%を最大とするパーセント利用率で評価されてもよい。リソースの利用に影響するファクタは、いかなるプロセッサのクロックレート、このようなプロセッサの特定の設計、及びコアがアイドルリングに要した時間を含んでもよい。CPU間バンド幅は、GB/sとして測定されてもよい。例えば、30GB/sである。クロックレート及び特定のコア又はプロセッサの設計は、CPU間バンド幅に影響する。CPUとRAMとの間のバンド幅は、GB/sにより測定されてもよい。例えば最大値20GB/sが例として挙げられる。クロックレート、メモリチャネル数、及びプロセッサ又はコアの特定の設計は、CPUとRAMとの間のバンド幅に影響する。共有キャッシュは、メガバイトで測定されてもよい。例としては12メガバイトが挙げられる。共有キャッシュは、システム100のプロセッサ又はコアの特定の設計に影響を受ける。パワー利用制限は、ワットで測定できる。例えば125ワットが挙げられる。クロックレート、電源電圧、システム100のプロセッサの特定の設計、所定のコアのアイドルリングにかかる時間は、利用及び最大パワー利用に影響する。TLBは、エントリの総数で測定され得る。最大1,000が例として挙げられる。このようなバッファの最大は、システム100の特定の設計によって決定されてもよい。

30

40

## 【0037】

【表 3】

リソース	容量単位	最大値の例	最大値に与えるファクタ
浮動小数点ユニット (FPU)	Cycles per Instruction (CPI)	2.0 CPI	クロックレート、特定のデザイン
整数ユニット (IntU)	CPI	1.5 CPI	クロックレート、特定のデザイン
コア間バンド幅	GB/s	1000 GB/s	クロックレート、特定のデザイン
コア占有キャッシュ	Kilobytes (KB)	128 KB	特定のデザイン
パワー利用制限	W	25 W	クロックレート、ソース電圧、特定のデザイン、リソースアイドリング
様々な特定のレジスタユニット	N/A	N/A	N/A

10

20

表 3 は、マルチスレッドのコアの複数のスレッドで共有し得るコアのサンプル・リソースを示す。このようなリソースに対するメタデータは、ワークロード・マネージャ 102 によって決定されてもよい。この結果、このようなメタデータは、スレッドのオペレーションの優先順位付け、又は、スレッドによる共有リソースへのアクセスのオペレーションの優先順位付けに用いられる。

## 【0038】

表 3 において、数値演算プロセッサの利用は、1 命令当たりのサイクル数（クロック数）によって測定されてもよい。浮動小数点ユニットのオペレーションの最大は、例えば 1 命令当たり 2.0 サイクルである。クロックレート及びシステム 100 の特定の設計が浮動小数点ユニットの利用及び最大値に影響する。整数ユニットは、1 命令当たりのサイクル数で計測されてもよい。例えば、1 命令当たり 1.5 サイクルが例としてあげられる。コア間バンド幅は、GB/s で測定されてもよい。最大値としては、1,000 GB/s が挙げられる。

30

コア間バンド幅と同様に、クロックレート及びシステム 100 の特定の設計は、整数ユニットの最大及び利用に影響を及ぼす。

コア占有キャッシュは、キロバイトで測定されてもよい。例えば、最大 128 キロバイトを有する。システム 100 の特定の設計は、コア占有キャッシュの利用及び最大値に影響を及ぼす。パワー利用制限は、ワットで測定されてもよい。例えば、最大 25 ワットが例として挙げられる。クロックレート、電源電圧、システム 100 の特定の設計は、利用及び最大パワー利用制限、システム 100 のリソースによるアイドリングに費やされる時間に影響を与える。その他の、最大値は、システム 100 の特定の設計に依存する。そして、このような利用、及びその最大は、システム 100 の処理エンティティのオペレーションを評価するために用いられる特定のレジスタの性質に従って変化する。

40

## 【0039】

動作において、ワークロード・キャラクタライザ 101 は、システム 100 の一つ以上の処理エンティティによって実行される 2 つ以上のワークロードのオペレーションの要求、及び計算リソースの要求を特徴づけてもよい。ワークロード・キャラクタライザ 101 は、このようなワークロードの実行のための利用できる計算リソースを特徴づけてもよい

50

。したがって、ワークロード・キャラクタライザ 101 は、ワークロード・メタデータ 101 及びリソース・ステータス 104 を生成してもよい。これらは、利用可能なリソースの効率的利用のために、ワークロードを処理エンティティに割り当てるためにワークロード・マネージャ 102 によって利用されてもよい。ワークロード・キャラクタライザ 101 及びワークロード・マネージャ 102 はサーバ 105、異なる電子デバイス、又は同じ電子デバイス上で動作してもよい。そして、この同じ電子デバイスは、サーバ 105 とは異なる電子デバイスであってもよい。

#### 【0040】

ワークロード・キャラクタライザ 101 及びワークロード・マネージャ 102 は、ワークロード・リクエスト、システム・リソースを評価してもよく、かつ、所定のレベルのシステム 100 のワークロード及び処理エンティティのために、ワークロードの実行を割り当ててもよい。例えば、ワークロード・キャラクタライザ 101 及びワークロード・マネージャ 102 は、一つ以上の計算ノード 110、一つ以上のコア 114、又は一つ以上のスレッド 118 の、ワークロード・リクエスト及びシステム・リソースを評価してもよい

計算ノード 110 は、ネットワーク化された構成、又は方式（例えばクラウド計算方式）において実行されてもよい。計算ノード 110 は、一部の共有リソースを共有しながら、効率的に一つ以上のこのようなワークロードを実行するために、ワークロード・マネージャ 102 によってワークロードを割り当てられてもよい。計算ノード 110 は、リソース、例えば利用できるシステムメモリ 204、ノード間バンド幅 205、所定の計算ノードの CPU 利用 206、パワー利用限度 207、及び / 又は I/O 接続バンド幅 208、を共有してもよい。計算ノード 110 の全体（プール）に利用できるこのようなリソースの量は、ワークロード・キャラクタライザ 101 によって特徴づけられてもよい。所定のワークロードによって必要なこのようなリソースの量は、ワークロード・キャラクタライザ 101 によって特徴づけられてもよい。

#### 【0041】

コア 114 は、単一の計算ノード 114 において動作する。各々のコア 114 は、一部の計算リソース（例えば CPU 116）を共有してもよい。ワークロード・マネージャ 102 は、一部の共有リソースを共有しながら、効率的に一つ以上のこのようなワークロードを実行するために、ワークロードをコア 114 に割り当ててもよい。このようなリソースは、CPU と RAM との間のバンド幅 214、CPU 間バンド幅 215、共用キャッシュ 216、メモリ利用 217、パワー利用限度 218 及び / 又はアドレス変換バッファ 219 を含んでもよい。コア 114 の全体（プール）で利用できるこのようなリソースの量は、ワークロード・キャラクタライザ 101 によって特徴づけられてもよい。所定のワークロードによって必要なこのようなリソースの量は、ワークロード・キャラクタライザ 101 によって特徴づけられてもよい。

#### 【0042】

スレッド 118 は、単一のコア 114 で動作してもよい。各々のスレッド 118 は、一部の計算リソース（例えばコア・リソース 120）を共有してもよい。ワークロード・マネージャ 102 は、一部の計算リソースを共有しながら、効率的に一つ以上のこのようなワークロードを実行するために、ワークロードをスレッド 114 に割り当ててもよい。このようなリソースは、浮動小数点ユニット 224、整数ユニット 225、コア間バンド幅 226、パワー利用限度 227、様々なレジスタ 228 及び / 又はコア占有キャッシュ 229 を含んでもよい。スレッド 118 の全体（プール）に利用できるこのようなリソースの量は、ワークロード・キャラクタライザ 101 によって特徴づけられてもよい。所定のスレッド 118 によって必要なこのようなリソースの量は、ワークロード・キャラクタライザ 101 によって特徴づけられてもよい。

#### 【0043】

ワークロード・マネージャ 102 は、例えば、ワークロード・メタデータ 103、ワークロード 506、又はワークロード 606 に、ワークロードリソース要求を特徴づける結果を記憶してもよい。ワークロード・マネージャ 102 は、処理エンティティのプールのため

10

20

30

40

50

の利用できる計算リソースを特徴づける結果を記憶してもよい。例えば、リソース・ステータス104、コア・オペレーション・ステータス502、又はノード・オペレーション・ステータス602が挙げられる。

【0044】

ワークロード・マネージャ102は、所定のワークロードを電子デバイスにロードして実行することによって、そのワークロードの要求を分析してもよい。ワークロード・マネージャ102は、特定のリソースのワークロードの利用を測定してもよい。ワークロード・マネージャ102は、リソースの利用を記録してもよい。一実施例において、サンプルデータが、ワークロードを実行するために利用されてもよい。ワークロード・マネージャ102は、異なるデータでワークロードを複数回実行し、かつ、その後リソース要求を平均化してもよい。

10

【0045】

一実施例において、ワークロード・マネージャ102は、ワークロードリソース要求及び利用できる計算リソースを定性的に特徴づけてもよい。別の実施例において、ワークロード・マネージャ102は、他のワークロードリソース要求に対して、ベンチマークに対して、又は利用できる計算リソースに対して相対的に、ワークロードリソース要求を特徴づけてもよい。ワークロード・マネージャ102は、所定のワークロード506のニーズを、一つ以上の閾値を用いて、リソース要求をセグメントに分けることによって、別個のセグメント又は利用のカテゴリに特徴付けてもよい。例えば、50%を超えるCPU利用は「重い」利用として分類され、50%未満のCPU利用が「低い」利用として分類されてもよい。同様に、ワークロード・マネージャ102は、リソースの利用可能性を、一つ以上の閾値を用いて、リソース要求をセグメントに分けることによって、たとえば、コア・オペレーション・ステータス502のように、別個のセグメント又は利用のカテゴリに特徴付けてもよい。例えば、CPUリソースが50%の容量より大きいレートで利用されている場合、リソースの負荷は「重い」と分類されてもよい。そして、CPUリソースが50%の容量より低いレートで利用されている場合、リソースの負荷は「低い」と分類されてもよい。

20

【0046】

別の実施例において、ワークロード・マネージャ102は、定量的にワークロードリソース要求及び利用できる計算リソースを特徴づけてもよい。ワークロード・マネージャ102は、所定のワークロード606のニーズを、ワークロードを実行するために必要とする容量を単位とする量に特徴づけてもよい。所定のワークロード606のための計算リソースの必要とされる量を示す方法は、必要とするリソースの性質に依存する。この点は、表1ないし3に示されている。例えば、計算リソースの必要な量は、利用できる計算リソースのパーセンテージ又は標準単位によって与えられていてもよい。

30

【0047】

一つ以上の処理エンティティの全体（プール）に利用できる、必要とされる計算リソースの量の特徴付けるために、例えば、ノード・オペレーション・ステータス602に見られるように、これらのリソースの容量を単位とする量にワークロードを特徴付けてもよい。上記表1ないし3に記載されているように、処理エンティティ又は処理エンティティのプールのために利用できる計算リソースの量、例えばノード・オペレーション・ステータス602は、特定のリソースに依存する。例えば、利用できる計算リソースの量は、利用できる計算リソースのパーセンテージ、又は標準単位によって与えられていてもよい。

40

【0048】

ワークロード・マネージャ102は、一つ以上の特定の処理エンティティによって実行される一つ以上のワークロードを割り当てるために、システム100の利用可能なリソース及び待機中ワークロードによって要求されるリソースに関する情報を利用してよい。待機中ワークロードは、キュー（例えばワークロードキュー504及び604、又はジョブスケジューラキュー310）に存在してもよい。処理エンティティはキュー（例えばCPUジョブキュー304-308）を含んでもよい。そして、このキューに、ワークロード

50

・マネジャ 102 は実行のための待機中ワークロードを挿入してもよい。このような挿入は、処理エンティティに対するワークロードの割当てであってもよい。

【0049】

ワークロード・マネジャ 102 は、利用可能なあらゆる形態のリソースが、潜在的な制限因子、又はボトルネックを形成するか否かを判断するために、処理エンティティのプールに利用可能なリソースに関する情報（例えばコア・オペレーション・ステータス 502、ノード・オペレーション・ステータス 602、又はリソース・ステータス 102）を利用してもよい。

【0050】

例えば図 5 で、ワークロード・マネジャ 102 は、コア・オペレーション・ステータス 502 において示されるように、ワークロード A によって重い利用がなされるリソースのいずれかが、利用できるコアによってすでに重い利用がなされているかどうか判断するために、ワークロードキュー 504（例えばワークロード A）の最上位のワークロード 506 で必要とするリソースに関する情報を利用してもよい。ワークロード A は、数値演算プロセッサ 234 のリソースの重い利用と、コア間バンド幅 226 のリソースの重い利用との両方を必要とする。

【0051】

コア・オペレーション・ステータス 502 は、現在の数値演算プロセッサ 224 の利用は軽く、コア間バンド幅 226 の利用が重いことを示している。したがって、ワークロード・マネジャ 102 は、コア間バンド幅 226 が潜在的なボトルネックであると判定してもよい。ワークロード・マネジャ 102 は、重い利用を呈するいかなるリソースも潜在的なボトルネックであると判断してもよい。例えば、ワークロード・マネジャ 102 は、様々なレジスタ 228 の現在の利用、及びコア占有キャッシュ 229 が、同様に潜在的なボトルネックであると判断してもよい。

【0052】

他の例では、図 6 において、ワークロード・マネジャ 102 は、ノード・オペレーション・ステータス 602 によって示されるように、ワークロード 606 によって必要ないずれかのリソースが、利用可能なリソースを上回るか否かを判断するために、ワークロードキュー 604 の一つ以上のワークロード 606 によって必要とされるリソースに関する情報を利用してもよい。ワークロード A は、256 ギガバイトのシステムメモリ 204 及び 10GB/s のノード間バンド幅 205 を必要とし、これに対して、ノード・オペレーション・ステータス 602 に示されるように、ノードのプールのための利用可能リソースは、128 メガバイトの利用できるシステムメモリ 204（512 メガバイトの最大容量、384 メガバイトが現在使用中であり減少している）及び 8GB/s の利用できるノード間バンド幅 205 となっている。ワークロード・マネジャ 102 は、したがって、ワークロード A のアプリケーションが利用できるシステムメモリ 204 及びノード間バンド幅 205 に対して、リソースのボトルネックが発生すると判断してもよい。

【0053】

ワークロード・マネジャ 102 は、一つ以上の処理エンティティ上での実行のための一つ以上のワークロードを選択するために、潜在的なボトルネックに関する情報、処理エンティティのプールが利用できる計算リソース、及び一つ以上のワークロードによって必要とされる計算リソースを利用してもよい。

【0054】

一つの実施例において、ワークロード・マネジャ 102 は、利用できる処理エンティティ・リソースに対する潜在的な競合に基づいて、ワークロードを優先させてもよい。様々な実施例において、ワークロード・マネジャ 102 は、ワークロード要求と、利用可能システム・リソースとの、定性的な又は相対的評価を比較して判断し、競合を最小化することによって、このようなワークロードを優先させてもよい。例えば、図 5 において、ワークロード・マネジャ 102 は、所定のコア・オペレーション・ステータス 502 及びワークロードキュー 504 を与えられ、ワークロード B の実行がワークロード A よりも優先さ

10

20

30

40

50

れることを決定してもよい。なぜなら、ワークロード B は単に「軽い」コア間バンド幅を要求しているため、コア間バンド幅 226 の潜在的ボトルネックにより、ワークロード A よりもワークロード B は、不利に影響されないからである。なお、ワークロード A は、処理エンティティに対して、コア間バンド幅で「重い」要求を行っており、かつ、このようなリソースの利用において既に「重い」要求をしている。しかしながら、ワークロード A 及び B は両者とも、利用可能なコア占有キャッシュ 229 で潜在的な競合を呈しており、ワークロード A 及び B は、両者ともそのリソースに「重い」利用を要求している。そして、処理エンティティにおいてそのリソースは既に「重い」状態となっている。このため、ワークロード・マネージャ 102 は、ワークロード A 及び B よりも、ワークロード C 及び D を優先させてもよい。ワークロード C 及び D は、コア占有キャッシュ 229 について、潜在的な競合を持っていない。なぜなら、これらは、コア占有キャッシュ (core specific cache) 229 への要求が「軽い」からである。ワークロード・マネージャ 102 は、ワークロード D よりもワークロード C を選択してもよい。なぜなら、ワークロード C はワークロード D より、当初から上位の優先順位であり、ワークロード C 及び D の要求を見る限り、潜在的なボトルネックを持っていないからである。

10

**【0055】**

一実施例において、ワークロード・マネージャ 102 は、ワークロードの測定されたリソース要求、及び利用できる処理リソースに基づいて、ワークロードを優先させてもよく、又は所定のワークロードを所定の処理エンティティに割り当ててもよい。このような実施例において、ワークロード・マネージャは、さまざまな処理エンティティの容量を上回らないように、ワークロード及びターゲット処理エンティティを優先させてもよく、又は選択してもよい。例えば、ワークロード・マネージャ 102 は、図 6 で、ワークロード A が、256 GB のシステムメモリ 204 及び 10 GB/s のノード間バンド幅 205 を必要とし、それぞれ 128 GB 及び 8 GB/s だけが利用できる状態になっているため、ワークロード A がノード・オペレーション・ステータス 602 によって示されるノードのプールの CPU による実行のために送信されよう決定してもよい。ワークロード・マネージャ 102 は、ワークロード B 及びワークロード C が、処理エンティティのプールによる実行のために送信されるよう決定してもよい。なぜなら、これらのワークロードの要求のいずれも処理エンティティのプールで利用可能リソースを越えないからである。このような場合、ワークロード B は CPU<sub>1</sub> 上での実行のために送信されてもよい。なぜなら、それは 85 % の CPU リソース 206 を必要とし、かつ CPU<sub>1</sub> は 100 % の利用できるリソースを有するからである。ワークロード C は CPU<sub>0</sub>、又は CPU<sub>1</sub> のいずれに送信されてもよい。なぜなら、その追加によって、いずれの CPU の容量を超えることもないからである。

20

30

**【0056】**

図 7 は、計算ワークロード・メタデータの生成、分析及び利用のための方法 700 に関する実施例を示している。

**【0057】**

ステップ 705 において、システムの利用できる計算リソースが測定される。このようなリソースは、例えば、計算ノードの間で、コアの間で、又はスレッドの間で共有されるリソースを反映してもよい。ステップ 710 において、利用できる計算リソースが、特徴づけられる。一実施例において、利用できる計算リソースは、定量的方法で特徴づけられてもよい。別の実施例において、このような計算リソースは、定性的、又は、例えばベンチマークに対して、計算リソースの経時的な量 (historical amounts) に対して、又は様々なワークロードに要求される計算リソースに対して相対的に特徴付けられてもよい。

40

**【0058】**

ワークロードの実行に影響を及ぼすことがあり得る、いかなる適切な利用できる計算リソースが、測定され、かつ特徴づけられてもよい。このようなリソースは、所定のワークロードを実行している処理エンティティに従って変化してもよい。

50

## 【0059】

一実施例において、このような利用できる計算リソースは、定性的、又は、例えばベンチマークに対して、計算リソースの経時的量に対して、又は様々なワークロードに要求される計算リソースに対して相対的に特徴付けられてもよい。このような実施例では、一つ以上の閾値が、一つ以上のカテゴリにリソースの部分利用可能性に適用されてもよい。例えば、処理エンティティの現在のCPU利用のための50%の閾値が適用される。この閾値よりも大きければ、そのような現在の利用は「重い」と分類され、この閾値よりも小さければ、そのような現在の利用は「軽い」と分類される。別の実施例において、利用できる計算リソースは、定量的に特徴づけられてもよい。このような実施例では、利用できる計算リソースは、比較及び分析のために、単位当たりの量 (quantity of unit)、パーセンテージ、又は他の適切なメカニズムの量として評価されてもよい。

10

## 【0060】

一つ以上の待機中ワークロード (処理エンティティによる実行のために待機中のワークロード) に対して、ワークロードによって必要な計算リソースは、測定され、かつ特徴づけられる。ステップ715において、ワークロードは、評価のためにロードされてもよい。ステップ720において、ワークロードは、サンプルデータを利用して実行されてもよい。ステップ725において、ワークロードをロードするために必要なリソース、及び実行のための必要なリソースが測定されてもよい。ステップ730において、ステップ715 - 725は、一つ以上のサンプル・データ・セットに対して繰り返されてもよい。

## 【0061】

ステップ735において、ワークロードによって必要な計算リソースは、平均値が取られ、かつ特徴づけられてもよい。一実施例において、必要な計算リソースは、定性的に特徴づけられてもよく、又は例えば、他のワークロード、ベンチマーク、又は利用可能システム・リソースに対して相対的に特徴付けられてもよい。このような実施例では、リソースの利用を一つ以上の利用のカテゴリに分類するために、一つ以上の閾値が適用されてもよい。例えば、CPU利用に対して50%の閾値が適用され、その閾値より高ければ、「重い」と分類され、低ければ、そのような利用は、「軽い」と分類される。別の実施例において、必要な計算リソースは、定量的に特徴づけられてもよい。このような実施例では、必要な計算リソースは、比較及び分析のための単位当たりの量、割合、又は他の適切なメカニズムの量として評価されてもよい。

20

30

## 【0062】

ステップ740において、ワークロード要求は、他のワークロード要求に対して比較されてもよい。一実施例において、リソースの利用要求に従って、ワークロード要求は、ランクを付けられてもよい。

## 【0063】

ステップ745において、ワークロード要求は、利用できる計算リソースに対して比較されてもよい。ワークロード要求及び利用できる計算リソースが定性的又は相対的な手法で決定された場合、要求の各々の一組はこのような要求がシステム・リソースの重い利用を必要とするかどうか判定するために分析されてもよい。ここで、システム・リソースは、すでに重く利用されている。ワークロード要求及び利用できる計算リソースが定量的方法で決定された場合、要求の各々の一組はこのような要求が利用できるリソースを上回るかどうか判定するために分析されてもよい。

40

## 【0064】

ステップ750において、ワークロード要求と利用できる計算リソースとの比較に基づいて、ワークロードは効率的な実行のために優先されてもよい。一実施例において、ワークロードは、また、前の指定された又は決定されたいかなる順番又は優先順位に基づいて優先されてもよい。例えば、優先する第1のワークロードが、利用可能なリソースを上回る場合、当初は優先順位の低かった第2のワークロードを、第1のワークロードよりも更に優先させてもよい。この場合、第2のワークロードの実行は、利用可能なリソースを上回らないと評価されるとする。他の例では、低い利用可能性を持つ利用可能な計算リソース

50

に対応する一つ以上のボトルネックが特定されてもよい。このような実施例において、ワークロードは、ボトルネックを形成している計算リソースのそれらの要求に基づいて、実行のために優先されてもよい。

【0065】

さらに別の例では、第1のワークロードが、利用できる計算リソースの大量消費をするリソースであると特定された場合、第2のワークロードが第1のものよりも優先されてもよい。ここにおいて、第2のワークロードは、計算リソースのより低い消費要求をもつものとする。ステップ755において、前のステップの優先順位に基づいて、ワークロードは実行のための処理エンティティに割り当てられてもよい。

【0066】

方法700は図1ないし図3のシステム、又は方法700を実行するために利用できるいかなる他のシステムを利用して実行されてもよい。このように、方法700のための最適なインシャライズの位置、及びそのステップの順番は、選択されるインプリメンテーションに依存してもよい。いくつかの実施例では、一部のステップは、任意に省略されてもよく、繰り返され、又は組み合わされてもよい。例えば、方法700のオペレーションの最中に、新たに要求されたワークロードが受信されてもよい。そして、新たに要求されたワークロードのメタデータを計測しかつ分析するために、幾つかのステップ715-750は適切に繰り返されてもよい。いくつかの実施例では、方法700の一部のステップは、方法700の他のステップと並列に実行されてもよい。例えば、ステップ705-710は、ステップ710-740と並列に実行されてもよい。ある種の実施例では、方法700は、コンピュータ可読のメディアに格納されたソフトウェアによって、部分的に、又は完全に実行されてもよい。

【0067】

この開示された実施例のために、コンピュータ可読の媒体は、一定期間、データ及び/又は命令を保持する、いかなる手段、又は手段の集合をも含み得る。コンピュータ可読のメディアとしては、格納媒体、例えば、ダイレクトアクセス記憶媒体（例えば、ハードディスクドライブ、フロッピー（登録商標）ディスク）、シーケンシャルアクセス記憶デバイス（例えば、テープディスクドライブ）、コンパクトディスク、CD-ROM、DVD、RAM、ROM、EEPROM、及び/又は、フラッシュメモリを含む。加えて、通信媒体、例えば、電線、光ファイバ、及びその他の電磁気学的及び/又は光媒体、及び/又はこれらのいかなる組合せも含まれる。なお上記は、限定を目的とするものではない。本実施例が詳細に説明されたが、添付の請求項に定義された開示の精神及び技術的範囲を逸脱しない限り、これらに対して、様々な変更、置換、変形を行うことが可能である。

【0068】

以上の実施例に関し、更に以下の付記を開示する。

(付記1) 計算リソースを管理する方法であって、

第1のワークロードのための第1のワークロード・メタデータを生成するステップと；

第2のワークロードのための第2のワークロード・メタデータを生成するステップであって；

前記第1のワークロード・メタデータ及び前記第2のワークロード・メタデータの各々は、関連づけられたワークロードによって必要な計算リソースを特徴づけ；

最初に、前記第1のワークロードのメタデータが、前記第2のワークロード・メタデータより優先され；かつ

前記第1のワークロード及び前記第2のワークロードは、一つ以上の処理エンティティによって実行される；

ステップと；

リソース・メタデータに対して前記第1のワークロード・メタデータ及び前記第2のワークロード・メタデータを比較するステップであって、前記計算リソースを特徴づけている前記リソース・メタデータは、一つ以上の前記処理エンティティにより利用可能である、ステップと、

10

20

30

40

50

リソース・メタデータに対するワークロード・メタデータの前記比較に基づいて、前記第1のワークロードと前記処理エンティティが利用できる前記計算リソースとの間でのリソース要求の潜在的な競合を特定するステップと；

前記処理エンティティのうちの一つによる実行のために前記第2のワークロードを割り当てるステップであって、リソース要求の前記潜在的な競合は、前記第2のワークロードの前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在しない、ステップと、

を有する方法。

(付記2) 前記第1のワークロード・メタデータ及び第2のワークロード・メタデータは、各々のワークロードに要求される計算リソースの相対的特徴を含む、付記1記載の方法

10

(付記3) 前記第1のワークロード・メタデータ及び第2のワークロード・メタデータは、各々のワークロードに要求される前記計算リソースの定量的特徴を含む、付記1記載の方法。

(付記4) 前記一つ以上の処理エンティティは、計算ノードのプールを有する、付記1記載の方法。

(付記5) 前記一つ以上の処理エンティティは、プロセッサ・コアのプールを有する、付記1記載の方法。

(付記6) 前記一つ以上の処理エンティティは、プロセッサ・スレッドのプールを有する、付記1記載の方法。

20

(付記7) ワークロードのためのワークロード・メタデータを生成するステップは、実行のために前記ワークロードをシステムにロードするステップと；データの第1のサンプルによって前記ワークロードを実行するステップと；前記ワークロードによって利用される前記計算リソースを測定するステップと；を有する付記1記載の方法。

(付記8) ワークロードのためのワークロード・メタデータを生成するステップは、データの第2のサンプルによって前記ワークロードを実行するステップと；前記ワークロードによって利用される前記計算リソースを測定するステップと；前記ワークロードによって使用された計算リソースを平均化するステップと；を更に有する付記7記載の方法。

30

(付記9)

第1の処理エンティティ及び第2の処理エンティティに利用可能な計算リソースを特徴付けることによって、リソース・メタデータが生成され；

リソース・メタデータに対して前記第1のワークロード・メタデータ及び第2のワークロード・メタデータを比較するステップは、第1のワークロード・メタデータ及び第2のワークロード・メタデータを、前記第1の計算エンティティ及び前記第2の計算エンティティに利用可能な前記リソースに対して比較するステップを含み、

前記処理エンティティのうちの一つによる実行のために前記第2のワークロードを割り当てるステップは、

前記第1の処理エンティティが、前記第2のワークロードを実行するために、十分なリソースを持っているか否かを判断するステップと；

40

前記第1の処理エンティティが、不十分なリソースしか持たない場合、前記第2のワークロードを実行するために、前記第2の処理エンティティが十分なリソースを持つか否かを判断するステップと；

前記第2の処理エンティティが十分なリソースを持つ場合、前記第2の処理エンティティによる実行のために前記第2のワークロードを割り当てるステップと；

を含む、付記1記載の方法。

(付記10) 第1の処理エンティティによる前記第1のワークロードの実行が、前記処理エンティティに対して、利用可能な計算リソースの所定のカテゴリの100パーセントを

50

超えさせる場合、潜在的な競合が特定される、付記 3 記載の方法。

(付記 1 1)

コンピュータに、

第 1 のワークロードのための第 1 のワークロード・メタデータを生成するステップと；  
第 2 のワークロードのための第 2 のワークロード・メタデータを生成するステップであ  
って；

前記第 1 のワークロード・メタデータ及び前記第 2 のワークロード・メタデータの各  
々は、関連づけられたワークロードによって必要な計算リソースを特徴づけ；

最初に、前記第 1 のワークロードのメタデータが、前記第 2 のワークロード・メタデ  
ータより優先され；かつ

前記第 1 のワークロード及び前記第 2 のワークロードは、一つ以上の処理エンティテ  
ィによって実行される；

ステップと；

リソース・メタデータに対して前記第 1 のワークロード・メタデータ及び前記第 2 のワ  
ークロード・メタデータを比較するステップであって、前記計算リソースを特徴づけてい  
る前記リソース・メタデータは、一つ以上の前記処理エンティティにより利用可能である  
、ステップと、

リソース・メタデータに対するワークロード・メタデータの前記比較に基づいて、前記  
第 1 のワークロードと前記処理エンティティが利用できる前記計算リソースとの間でのリ  
ソース要求の潜在的な競合を特定するステップと；

前記処理エンティティのうちの一つによる実行のために前記第 2 のワークロードを割り  
当てるステップであって、リソース要求の前記潜在的な競合は、前記第 2 のワークロード  
の前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在  
しない、ステップと、

を実行させるプログラム。

(付記 1 2) 前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデー  
タは、各々のワークロードに要求される計算リソースの相対的特徴を含む、付記 1 1 記載の  
プログラム。

(付記 1 3) 前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデー  
タは、各々のワークロードに要求される前記計算リソースの定量的特徴を含む、付記 1 1 記  
載のプログラム。

(付記 1 4) 前記一つ以上の処理エンティティは、計算ノードのプールを有する、付記 1  
1 記載のプログラム。

(付記 1 5) 前記一つ以上の処理エンティティは、プロセッサ・コアのプールを有する、  
付記 1 1 記載のプログラム。

(付記 1 6) 前記一つ以上の処理エンティティは、プロセッサ・スレッドのプールを有す  
る、付記 1 1 記載のプログラム。

(付記 1 7) ワークロードのためのワークロード・メタデータを生成するステップは、  
実行のために前記ワークロードをシステムにロードするステップと；

データの第 1 のサンプルによって前記ワークロードを実行するステップと；

前記ワークロードによって利用される前記計算リソースを測定するステップと；

を有する付記 1 1 記載のプログラム。

(付記 1 8) ワークロードのためのワークロード・メタデータを生成するステップは、  
データの第 2 のサンプルによって前記ワークロードを実行するステップと；

前記ワークロードによって利用される前記計算リソースを測定するステップと；

前記ワークロードによって使用された計算リソースを平均化するステップと；

を更に有する付記 1 7 記載のプログラム。

(付記 1 9)

第 1 の処理エンティティ及び第 2 の処理エンティティに利用可能な計算リソースを特徴  
付けることによって、リソース・メタデータが生成され；

10

20

30

40

50

リソース・メタデータに対して前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータを比較するステップは、第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータを、前記第 1 の計算エンティティ及び前記第 2 の計算エンティティに利用可能な前記リソースに対して比較するステップを含み、

前記処理エンティティのうちの 1 つによる実行のために前記第 2 のワークロードを割り当てるステップは、

前記第 1 の処理エンティティが、前記第 2 のワークロードを実行するために、十分なリソースを持っているか否かを判断するステップと；

前記第 1 の処理エンティティが、不十分なリソースしか持たない場合、前記第 2 のワークロードを実行するために、前記第 2 の処理エンティティが十分なリソースを持つか否かを判断するステップと；

前記第 2 の処理エンティティが十分なリソースを持つ場合、前記第 2 の処理エンティティによる実行のために前記第 2 のワークロードを割り当てるステップと；

を含む、付記 11 記載のプログラム。

(付記 20) 第 1 の処理エンティティによる前記第 1 のワークロードの実行が、前記処理エンティティに対して、利用可能な計算リソースの所定のカテゴリの 100 パーセントを超えさせる場合、潜在的な競合が特定される、付記 13 記載のプログラム。

(付記 21) 計算ワークロードを管理するシステムであって、プロセッサと；

前記プロセッサに接続される計算機可読の媒体であって、コンピュータに、

第 1 のワークロードのための第 1 のワークロード・メタデータを生成するステップと

；

第 2 のワークロードのための第 2 のワークロード・メタデータを生成するステップであって；

前記第 1 のワークロード・メタデータ及び前記第 2 のワークロード・メタデータの各々は、関連づけられたワークロードによって必要な計算リソースを特徴づけ；

前記第 1 のワークロード及び前記第 2 のワークロードは、一つ以上の処理エンティティによって実行される；

ステップと；

リソース・メタデータに対して前記第 1 のワークロード・メタデータ及び前記第 2 のワークロード・メタデータを比較するステップであって、前記計算リソースを特徴づけている前記リソース・メタデータは、一つ以上の前記処理エンティティにより利用可能である、ステップと、

リソース・メタデータに対するワークロード・メタデータの前記比較に基づいて、前記第 1 のワークロードと前記処理エンティティが利用できる前記計算リソースとの間でのリソース要求の潜在的な競合を特定するステップと；

前記処理エンティティのうちの 1 つによる実行のために前記第 2 のワークロードを割り当てるステップであって、リソース要求の前記潜在的な競合は、前記第 2 のワークロードの前記リソース要求と前記処理エンティティが利用できる前記計算リソースとの間に存在しない、ステップと、

を実行させるプログラムを格納した媒体と；

を有するシステム。

(付記 22) 前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータは、各々のワークロードに要求される計算リソースの相対的特徴を含む、付記 21 記載のシステム。

(付記 23) 前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータは、各々のワークロードに要求される前記計算リソースの定量的特徴を含む、付記 21 記載のシステム。

(付記 24) 前記一つ以上の処理エンティティは、計算ノードのプールを有する、付記 21 記載のシステム。

10

20

30

40

50

(付記 25) 前記一つ以上の処理エンティティは、プロセッサ・コアのプールを有する、付記 21 記載のシステム。

(付記 26) 前記一つ以上の処理エンティティは、プロセッサ・スレッドのプールを有する、付記 21 記載のシステム。

(付記 27) ワークロードのためのワークロード・メタデータを生成するステップは、実行のために前記ワークロードをシステムにロードするステップと；データの第 1 のサンプルによって前記ワークロードを実行するステップと；前記ワークロードによって利用される前記計算リソースを測定するステップと；を有する付記 21 記載のシステム。

(付記 28) ワークロードのためのワークロード・メタデータを生成するステップは、データの第 2 のサンプルによって前記ワークロードを実行するステップと；前記ワークロードによって利用される前記計算リソースを測定するステップと；前記ワークロードによって使用された計算リソースを平均化するステップと；を更に有する付記 27 記載のシステム。

(付記 29)

第 1 の処理エンティティ及び第 2 の処理エンティティに利用可能な計算リソースを特徴付けることによって、リソース・メタデータが生成され；

リソース・メタデータに対して前記第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータを比較するステップは、第 1 のワークロード・メタデータ及び第 2 のワークロード・メタデータを、前記第 1 の計算エンティティ及び前記第 2 の計算エンティティに利用可能な前記リソースに対して比較するステップを含み、

前記処理エンティティのうちの 1 つによる実行のために前記第 2 のワークロードを割り当てるステップは、

前記第 1 の処理エンティティが、前記第 2 のワークロードを実行するために、十分なリソースを持っているか否かを判断するステップと；

前記第 1 の処理エンティティが、不十分なリソースしか持たない場合、前記第 2 のワークロードを実行するために、前記第 2 の処理エンティティが十分なリソースを持つか否かを判断するステップと；

前記第 2 の処理エンティティが十分なリソースを持つ場合、前記第 2 の処理エンティティによる実行のために前記第 2 のワークロードを割り当てるステップと；

を含む、付記 21 記載のプログラム。

(付記 30) 第 1 の処理エンティティによる前記第 1 のワークロードの実行が、前記処理エンティティに対して、利用可能な計算リソースの所定のカテゴリの 100 パーセントを超えさせる場合、潜在的な競合が特定される、付記 23 記載のシステム。

【符号の説明】

【0069】

100 システム

105 サーバ

110 計算ノード

112 ネットワーク

114 コア

118 スレッド

10

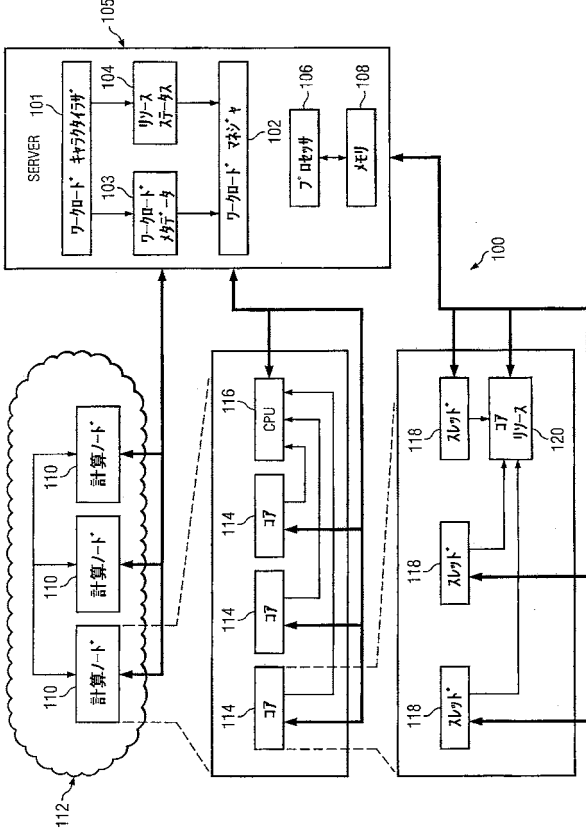
20

30

40

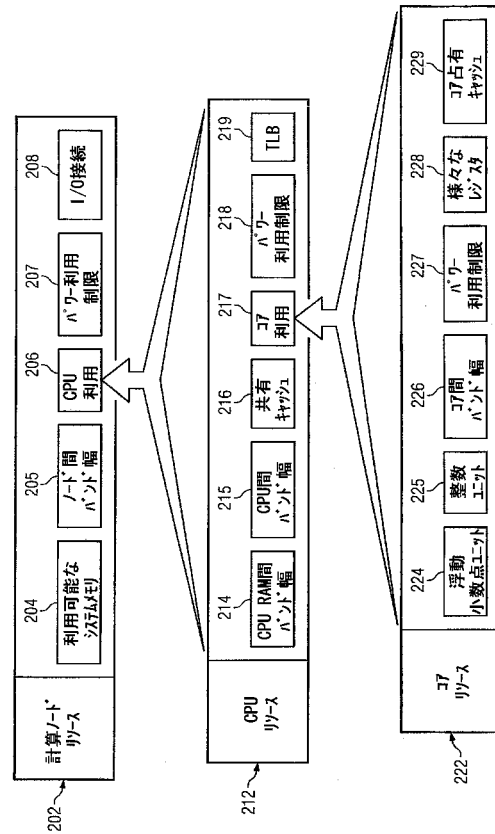
【 図 1 】

計算ワークロード・メタデータの生成の分析及び利用のためのシステムの実施例を示す図



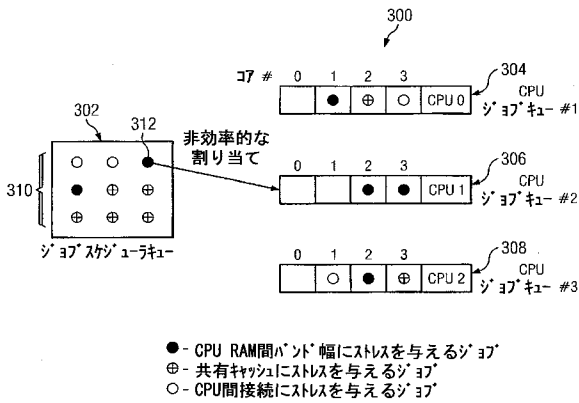
【 図 2 】

ワークロード・キャラクタライゼによって特徴付けられ、システムの処理エンティティによりワークロードの優先付けを行うためのワークロード・マネージャによって利用される例示のシステムのリースを示す図



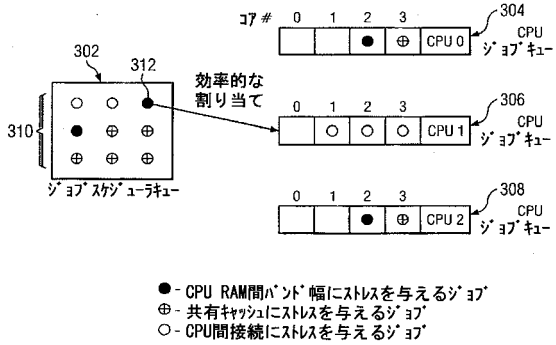
【 図 3 】

システムから発生したワークロードがどのように、非効率的にシステムの様々な処理ユニットに割り当てられるかを示す図



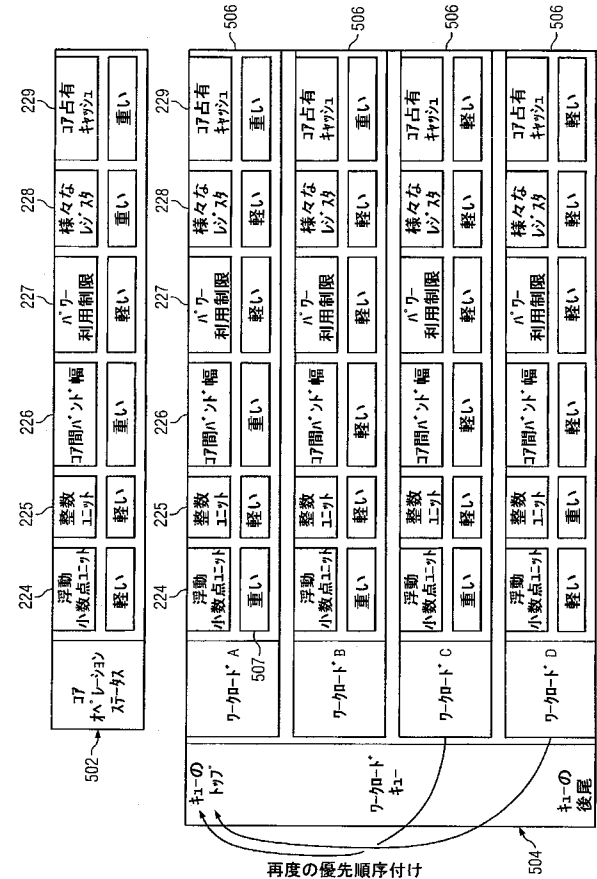
【 図 4 】

ワークロード・マネージャが、待ちのワークロードをジョブキューに割り当てる実施例の効率的な割り当て方式を示す図



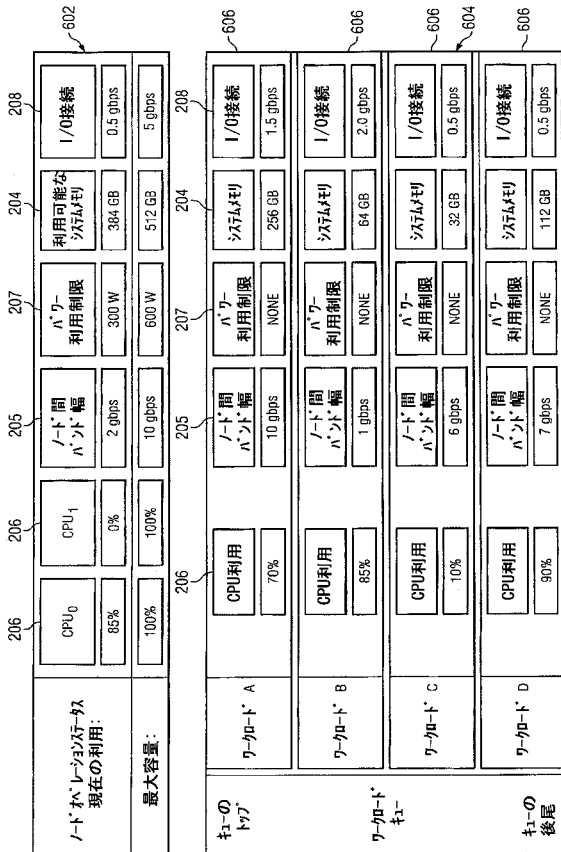
【図5】

ワークロード・マネージャが実行を優先付けるために、定性的又は相対的な手法によりワークロード・メタデータ及びリソース・ステータスを生成する実施例を示す図



【図6】

ワークロード・キャラクタライズ及びワークロード・マネージャによる、定量的なカテゴリ分け及び優先付けの実施例を示す図



【図7】

計算ワークロード・メタデータの生成、分析、及び利用のための方法の実施例を示す図

