

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2017-521796

(P2017-521796A)

(43) 公表日 平成29年8月3日(2017.8.3)

(51) Int.Cl.

G06F 9/50 (2006.01)

F I

G06F 9/46 4 6 5 C

テーマコード (参考)

審査請求 未請求 予備審査請求 未請求 (全 21 頁)

(21) 出願番号 特願2017-503021 (P2017-503021)  
 (86) (22) 出願日 平成27年7月10日 (2015.7.10)  
 (85) 翻訳文提出日 平成29年1月13日 (2017.1.13)  
 (86) 国際出願番号 PCT/US2015/039993  
 (87) 国際公開番号 W02016/014263  
 (87) 国際公開日 平成28年1月28日 (2016.1.28)  
 (31) 優先権主張番号 14/340,332  
 (32) 優先日 平成26年7月24日 (2014.7.24)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 517014158  
 イニゲス, アリフォンソ  
 アメリカ合衆国 アリゾナ 85203,  
 メサ, イー, レビ ロード 732  
 (74) 代理人 100078282  
 弁理士 山本 秀策  
 (74) 代理人 100113413  
 弁理士 森下 夏樹  
 (74) 代理人 100181674  
 弁理士 飯田 貴敏  
 (74) 代理人 100181641  
 弁理士 石川 大輔  
 (74) 代理人 230113332  
 弁理士 山本 健策

最終頁に続く

(54) 【発明の名称】 動的に構成可能な先回りコプロセッシングセルを用いる並列処理のためのシステムおよび方法

## (57) 【要約】

並列処理アーキテクチャが、CPUと、CPUによって投入されるタスクプールと、それぞれ、タスクプールに先回りして問い合わせ、特定のコプロセッサに対して適切なタスクを読み出すように構成されるエージェントを有する、複数の自律的共処理セルを含む。各コプロセッサは、全てのシステムリソース間のデータ転送およびアービトレーションのための接続を促進する、スイッチングファブリックを通してタスクプールと通信する。各コプロセッサは、タスクまたはタスクスレッドが完了すると、これをタスクプールに通知し、その後、タスクプールは、CPUに通知する。

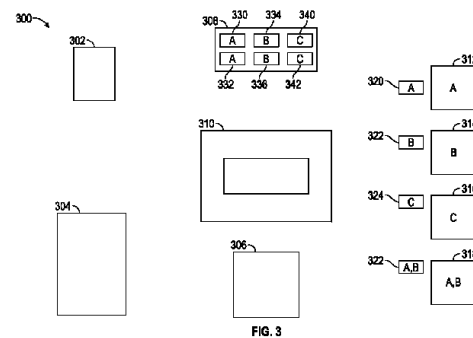


FIG. 3

**【特許請求の範囲】****【請求項 1】**

処理システムであって、  
タスクプールと、  
第 1 のタスクを前記タスクプールに投入するように構成される、コントローラと、  
前記タスクプールから前記第 1 のタスクを先回りして読み出すように構成される、第 1 のコプロセッサと、  
を備える、処理システム。

**【請求項 2】**

前記第 1 のコプロセッサは、前記コントローラと通信することなく、前記タスクプールから前記第 1 のタスクを読み出すように構成される、第 1 のエージェントを備える、請求項 1 に記載の処理システム。

10

**【請求項 3】**

前記第 1 のタスクは、第 1 のタスクタイプの印を含み、前記第 1 のコプロセッサは、前記第 1 のタイプのタスクを実施するように構成され、前記第 1 のエージェントは、前記第 1 のタイプのタスクに関して前記タスクプールを検索するように構成される、請求項 2 に記載の処理システム。

**【請求項 4】**

前記第 1 のコプロセッサはさらに、前記第 1 のものを処理し、前記第 1 のタスクの完了に応じて、前記タスクプールに通知するように構成される、請求項 1 に記載の処理システム。

20

**【請求項 5】**

前記タスクプールは、前記第 1 のタスクの完了に応じて、前記コントローラに通知するように構成される、請求項 1 に記載の処理システム。

**【請求項 6】**

前記コントローラおよび前記第 1 のコプロセッサは、相互に前記タスクプールを通してのみ通信するように構成される、請求項 1 に記載の処理システム。

**【請求項 7】**

前記コントローラおよび前記第 1 のコプロセッサは、相互に直接、前記タスクプールを通して通信するように構成される、請求項 1 に記載の処理システム。

30

**【請求項 8】**

前記第 1 のコプロセッサは、これが利用可能な処理容量を有していることを判定し、前記判定に回答して前記エージェントを前記タスクプールにディスパッチするように構成される、請求項 2 に記載の処理システム。

**【請求項 9】**

前記コントローラはさらに、第 2 のタスクを前記タスクプールに投入するように構成され、前記システムはさらに、前記タスクプールから前記第 2 のタスクを先回りして読み出すように構成される第 2 のエージェントを有する、第 2 のコプロセッサを備える、請求項 3 に記載の処理システム。

**【請求項 10】**

前記第 2 のタスクは、第 2 のタスクタイプの印を含み、前記第 2 のコプロセッサは、前記第 2 のタイプのタスクを実施するように構成され、前記第 2 のエージェントは、前記第 2 のタイプのタスクに関して前記タスクプールを検索するように構成される、請求項 9 に記載の処理システム。

40

**【請求項 11】**

前記コントローラおよび前記タスクプールは、モノリシック集積回路（IC）上に常駐し、前記第 1 のコプロセッサは、前記 IC 上に常駐しない、請求項 1 に記載の処理システム。

**【請求項 12】**

前記コントローラ、前記タスクプール、ならびに前記第 1 および第 2 のコプロセッサは

50

、モノリシック集積回路（ＩＣ）上に常駐する、請求項 9 に記載の処理システム。

【請求項 13】

第 1 のタスクタイプを有する第 1 のタスクをタスクプールに投入するように構成される、中央処理ユニット（ＣＰＵ）を含むタイプのネットワーク内の処理リソースを動的に制御する方法であって、

前記第 1 のタスクタイプを実施するように第 1 のセルをプログラムするステップと、

前記プログラムされた第 1 のセルを前記ネットワークに追加するステップと、

前記第 1 のセルから前記タスクプールに、第 1 のエージェントを先回りして送信するステップと、

前記第 1 のエージェントによって、前記第 1 のタイプのタスクに関して前記タスクプールを検索するステップと、

前記第 1 のエージェントによって、前記タスクプールから前記第 1 のタスクを読み出すステップと、

前記第 1 のエージェントによって、前記第 1 のセルに前記第 1 のタスクをトランスポートするステップと、

前記第 1 のセルによって、前記第 1 のタスクを処理するステップと、

前記第 1 のセルから前記タスクプールに、前記第 1 のタスクが完了した通知を送信するステップと、

を含む、方法。

【請求項 14】

前記タスクプールによって、前記第 1 のタスクを完了しているものとしてマーキングするステップと、

前記タスクプールから前記ＣＰＵに、前記第 1 のタスクが完了した通知を送信するステップと、

をさらに含む、請求項 13 に記載の方法。

【請求項 15】

前記第 1 のエージェントを前記タスクプールに先回りして送信するための述語として、前記第 1 のセルが利用可能な処理容量を有すると判定するように前記第 1 のセルを構成するステップをさらに含む、請求項 13 に記載の方法。

【請求項 16】

前記プログラムされた第 1 のセルを前記ネットワークに追加することに先立って、前記第 1 のセルを第 1 のデバイスに統合するステップをさらに含む、請求項 13 に記載の方法。

【請求項 17】

前記第 1 のデバイスは、センサ、電球、電力スイッチ、電化製品、バイオメトリックデバイス、医療デバイス、診断デバイス、ラップトップ、タブレット、スマートフォン、モータコントローラ、およびセキュリティデバイスのうちの 1 つを含む、請求項 16 に記載の方法。

【請求項 18】

前記プログラムされた第 1 のセルをネットワークに追加するステップは、前記第 1 のセルと前記タスクプールとの間に通信リンクを確立するステップを含む、請求項 13 に記載の方法。

【請求項 19】

前記（ＣＰＵ）はさらに、第 2 のタスクタイプを有する第 2 のタスクをタスクプールに投入するように構成され、

前記第 2 のタスクタイプを実施するように前記第 2 のセルをプログラムするステップと、

、

前記第 2 のセルと前記タスクプールとの間に通信リンクを確立するステップと、

前記第 2 のセルから前記タスクプールに、第 2 のエージェントを先回りして送信するステップと、

10

20

30

40

50

前記第 2 のエージェントによって、前記第 2 のタイプのタスクに関して前記タスクプールを検索するステップと、

前記第 2 のエージェントによって、前記タスクプールから前記第 2 のタスクを読み出すステップと、

前記第 2 のエージェントによって、前記第 2 のセルに前記第 2 のタスクをトランスポートするステップと、

前記第 2 のセルによって、前記第 2 のタスクを処理するステップと、

前記第 2 のセルから前記タスクプールに、前記第 2 のタスクが完了した通知を送信するステップと、

前記タスクプールによって、前記第 2 のタスクを完了しているものとしてマーキングするステップと、

前記タスクプールから前記 CPU に、前記第 2 のタスクが完了した通知を送信するステップと、

をさらに含む、請求項 13 に記載の方法。

#### 【請求項 20】

モノのインターネット（IoT）コンピューティング環境内の分散処理リソースを制御するためのシステムであって、

集約コンピューティング要件を複数のタスクにパーティション化し、前記タスクをプール内に置くように構成される、CPU と、

それぞれ、前記 CPU との直接通信を伴わずに、前記プールからタスクを先回りして読み出すように構成される、一意の専用エージェントを有する、複数のデバイスと、

を備える、システム。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

（優先権データ）

本願は、2013 年 1 月 25 日に出願された米国出願第 13 / 750 , 696 号の継続出願であり、該米国出願は、参照により本明細書中に援用される。

#### 【0002】

本発明は、概して、並列プロセスコンピューティングに関し、特に、中央処理ユニットによって投入されるタスクプールからタスクを先回りして読み出すように構成される、自律的コプロセッサを伴う処理アーキテクチャに関する。

#### 【背景技術】

#### 【0003】

モノのインターネット（モノのクラウドとも称される）は、既存のインターネットインフラストラクチャ内の一意に認識可能な埋め込まれたコンピューティングデバイスのアドホックネットワークを指す。モノのインターネット（IoT）は、マシンツーマシン通信（M2M）に優る、デバイス、システム、およびサービスの高度な接続性を示す。IoT によって想定されるモノの範囲は、無制限であり、例えば、心臓監視インプラント、バイオチップトランスポンダ、自動車センサ、航空宇宙および防衛分野オペレーションデバイス、ならびに探索および救助活動において消防士を補助する公衆安全用途等のデバイスを含み得る。現在の市場の実施例は、遠隔監視のために Wi-Fi を利用するスマートサーモスタット、電球、および洗濯機 / 乾燥機を伴う家庭ベースのネットワークを含む。IoT において接続される物体の遍在性に起因して、2020 年までに 300 億個を上回るデバイスが、モノのインターネットに無線接続されるであろうことが推定されている。これらのデバイスと関連付けられるコントローラおよびプロセッサの処理容量を利用することが、本発明の目的のうちの 1 つである。

#### 【0004】

コンピュータプロセッサは、伝統的に、マシンコード化命令を直列的に実行する。複数のアプリケーションを並行して起動するために、単一のプロセッサが、種々のプログラム

からの命令をインターリーブし、それらを直列的に実行するが、ユーザの視点からは、アプリケーションは、並列に処理されているように見える。一方、真の並列またはマルチコア処理は、大規模な算出タスクを個々の算出ブロックに細分化し、それらを2つまたはそれを上回るプロセッサ間で分散する算出アプローチである。タスク並列性（並列処理）を使用するコンピューティングアーキテクチャは、大規模な算出要件を実行可能なコードの離散モジュールに分割する。モジュールは、次いで、それらのそれぞれの優先順位に基づいて、並行して、または連続的に実行される。

#### 【0005】

典型的なマルチプロセッサシステムは、中央処理ユニット（「CPU」）と、1つまたはそれを上回るコプロセッサを含む。CPUは、算出要件をタスクにパーティション化し、そのタスクをコプロセッサに分散する。完了したスレッドは、CPUに報告され、これは、必要に応じて付加的なスレッドをコプロセッサに分散し続ける。現在公知の多重処理アプローチは、タスク分散によって有意な量のCPU帯域幅が消費される点において不利であり、新しいタスクを分散する前にタスクが完了するまで待機し（多くの場合、以前のタスクに依存する）、タスクが完了すると、コプロセッサからの割込みに応答し、コプロセッサからの他のメッセージに応答する。加えて、コプロセッサは、多くの場合、CPUからの新しいタスクを待ちながら、アイドルしたままである。

#### 【0006】

したがって、CPU管理オーバーヘッドを低減させ、また、利用可能な共処理リソースをより効果的に利用および活用するマルチプロセッサアーキテクチャが、必要とされる。

#### 【発明の概要】

#### 【課題を解決するための手段】

#### 【0007】

並列処理コンピューティングアーキテクチャの種々の実施形態は、タスクプールに投入するように構成される、CPUと、タスクプールからスレッド（タスク）を先回りして読み出すように構成される、1つまたはそれを上回るコプロセッサを含む。各コプロセッサは、タスクの完了に応じてタスクプールに通知し、別のタスクが処理のために利用可能になるまでタスクプールをping確認する。このように、CPUは、タスクプールと直接通信し、タスクプールを通してコプロセッサと間接的に通信する。

#### 【0008】

コプロセッサはまた、自律的に作用することも可能であり得、つまり、それらは、CPUから独立してタスクプールと相互作用してもよい。好ましい実施形態では、各コプロセッサは、実施するタスクを探求するためにタスクプールに問い合わせるエージェントを含む。その結果、コプロセッサは、相互に、およびタスクプールと「連帯して」協働し、相互に関連する、またはしない場合もある個々のタスクを自律的に読み出し、完了することによって、集約算出要件を完了する。非限定的実施例として、タスクBは、経時的な平均温度を算出するステップを伴うと仮定する。経時的な温度読取値を捕捉するステップを含むようにタスクAを定義することによって、さらに、捕捉された読取値を取得するステップを含むようにタスクBを定義することによって、CPUおよび種々のコプロセッサは、それによって、タスクプールを介して相互に推測的に通信してもよい。

#### 【0009】

種々の実施形態では、コプロセッサは、自律的かつ先見的な連帯セルと称される。この文脈において、自律的という用語は、コプロセッサが、CPUによってまたはタスクプールによってそうするように命令されることなく、タスクプールと相互作用し得ることを含意する。先見的という用語は、各コプロセッサが、周期的にエージェントを送信し、そのコプロセッサに適切な利用可能なタスクに関してタスクプールを監視するように構成（例えば、プログラム）され得ることを示唆する。連帯という用語は、共処理セルが、タスクプール内の全ての利用可能なタスクを監視および実行する際に共通の目的を共有することを含意する。

#### 【0010】

連帯セル（コプロセッサ）は、汎用または専用プロセッサであってもよく、したがって、本システム内のＣＰＵおよび他の連帯セルと比較して、同一または異なる命令セット、アーキテクチャ、およびマイクロアーキテクチャを有してもよい。さらに、実行されるべきソフトウェアプログラムおよび処理されるべきデータは、１つまたはそれを上回るメモリユニット内に含有されてもよい。典型的なコンピュータシステムでは、例えば、ソフトウェアプログラムは、プログラムによって使用されるべきデータを要求し得る、一連の命令から成る。例えば、プログラムがメディアプレーヤに対応する場合、メモリ内に含有されるデータは、コプロセッサによって読み取られ、最終的にスピーカで再生される圧縮オーディオデータであってもよい。

#### 【００１１】

本システム内の各連帯セルは、ファブリックとしても公知である、クロスバスイッチを通してタスクプールとオーム的に、または無線で通信するように構成されてもよい。純粹に無線のメッシュ形態では、無線信号自体が、ファブリックを構成してもよい。種々の実施形態では、コプロセッサはまた、ＣＰＵと直接通信してもよい。スイッチングファブリックは、システムリソース間の通信を促進する。各連帯セルは、連帯セルが実施する処理を全く有していないとき、または代替として、連帯セルがその通常動作を妨げることなく処理サイクルに寄与することが可能であるとき、そのエージェントをタスクプールに送信することによって、実施するタスクを取得するという点において先見的である。非限定的実施例として、モノのインターネット（以下により詳細に議論される）の文脈では、電球等のデバイスと関連付けられるコプロセッサは、その通常動作として、マスターデバイス（スマートフォン等）から「オン」および「オフ」コマンドを待つようにプログラムされてもよいが、その処理リソースもまた、タスクプールを通して利用されてもよい。

#### 【００１２】

本明細書に説明される種々の実施形態の文脈では、エージェントという用語は、タスクプールと相互作用し、それによって、そのコプロセッサセルに対して適切である利用可能なタスクを取得するコプロセッサと関連付けられる、ネットワークパケットに類似するソフトウェアモジュールを指す。連帯セルは、タスクが以前のタスクの実行を条件とするとき、連続的に、または実行のために１つを上回る連帯セルが利用可能であり、１つを上回る整合タスクが利用可能であるとき、並列にタスクを実行してもよい。タスクは、（該当する場合）ＣＰＵによって提供されるタスクスレッド制限に応じて、独立して、または共同で実行されてもよい。タスクプール内の相互依存したタスクは、論理的に組み合わせられてもよい。タスクプールは、タスクスレッドが完了すると、これをＣＰＵに通知する。タスクスレッドが単一のタスクから成る場合、タスクプールは、そのようなタスクの完了時にＣＰＵに通知してもよい。タスクスレッドが複数のタスクから成る場合、タスクプールは、そのような一連のタスクの完了時にＣＰＵに通知してもよい。タスクスレッドは論理的に組み合わせられ得るため、論理的に組み合わせられたタスクスレッドの完了後に、タスクプールがＣＰＵに通知する場合があることも想定される。

#### 【００１３】

当業者は、ＣＰＵおよびコプロセッサ間の相互運用性が、種々のコプロセッサと関連付けられる、命令セットアーキテクチャから独立した抽象レベルでタスクを構成および／または構築するようにＣＰＵを構成し、それによって、命令レベルではなく、タスクレベルで構成要素が通信することを可能にすることによって促進され得ることを理解するであろう。したがって、デバイスおよびそれらの関連付けられるコプロセッサは、「プラグアンドプレイ」ベースでネットワークに追加されてもよい。本発明の別の側面は、異なる命令セットアーキテクチャを伴うＣＰＵの異種アレイ内の相互運用性を提供する。

#### 【００１４】

本発明の種々の特徴は、とりわけ、モノのインターネットデバイスおよびセンサのネットワーク、異種コンピューティング環境、高性能コンピューティング２次元および３次元モノリシック集積回路、ならびにモーション制御およびロボットに適用可能である。

#### 【図面の簡単な説明】

## 【 0 0 1 5 】

本発明は、以降では、添付される図面と併せて説明され、同様の番号は、同様の要素を表す。

## 【 0 0 1 6 】

【図 1】図 1 は、ある実施形態による、CPU と、メモリと、タスクプールと、ファブリックを通して通信するように構成される複数のコプロセッサとを含む、並列処理アーキテクチャの概略ブロック図である。

## 【 0 0 1 7 】

【図 2】図 2 は、ある実施形態による、例示的タスクプールの詳細を例証する、概略ブロック図である。

10

## 【 0 0 1 8 】

【図 3】図 3 は、ある実施形態による、タスクプールと相互作用する、共処理セルおよびそれらの対応するエージェントを含むネットワークの概略ブロック図である。

## 【 0 0 1 9 】

【図 4】図 4 は、ある実施形態による、利用可能なプラグアンドプレイデバイスを含む、モノのインターネットネットワークの概略配置である。

## 【 0 0 2 0 】

【図 5】図 5 は、ある実施形態による、近傍デバイスの動的利用を例証する、例示的モノのインターネットの使用事例の概略配置図である。

## 【 0 0 2 1 】

20

【図 6】図 6 は、ある実施形態による、例示的並列コンピューティング環境の動作を例証する、フローチャートである。

## 【発明を実施するための形態】

## 【 0 0 2 2 】

種々の実施形態は、例えば、モノのインターネットの文脈における、限定ではないが、データ暗号化；グラフィック、ビデオ、およびオーディオ処理；ダイレクトメモリアクセス；数学的算出；データマイニング；ゲームアルゴリズム；ネットワーク外のデータの構築、受信、および伝送を含む、イーサネット（登録商標）パケットおよび他のネットワークプロトコル処理；金融サービスおよびビジネス方法；検索エンジン；インターネットデータストリーミングおよび他のウェブベースアプリケーション；内部または外部ソフトウェアプログラムの実行；電化製品、電球、消費者用電子機器、および同等物のオンおよびオフ切替ならびに / または別様にそれらの制御もしくは操作を含む、単純な切替および制御機能から複雑なプログラムおよびアルゴリズムまでの並列処理コンピューティングシステムおよび環境に関する。

30

## 【 0 0 2 3 】

種々の特徴が、任意の現在公知の、または後に開発されるコンピュータアーキテクチャ中に組み込まれてもよい。例えば、同期、データセキュリティ、順不同実行、およびメインプロセッサ割込みに関する並列処理の問題は、本明細書に説明される発明的概念を使用して対処され得る。

## 【 0 0 2 4 】

40

ここで図 1 を参照すると、分散処理システム 10 が、単一またはマルチコア CPU 11 と、クロスバースイッチングファブリック 14 を通してタスクプール 13 と通信するように構成される、1 つまたはそれを上回る連帯もしくは共処理セル 12 A - 12 とを含む。連帯セル 12 はまた、スイッチングファブリック 14 を通して、または別個のセルバス（図示せず）を通して、相互に通信してもよい。CPU 11 は、直接、またはスイッチングファブリック 14 を通して、タスクプール 13 と通信してもよい。1 つまたはそれを上回るメモリユニット 15 が、それぞれ、データおよび / または命令を含有する。本文脈では、「命令」という用語は、CPU 11 による実行のためにコンパイルされ得るソフトウェアプログラムを含む。メモリユニット 15、セル 12、およびタスクプール 13 は、スイッチングファブリック 14 を介して、CPU と、および / または相互に通信するようにオ

50

ーム的に、または無線で相互接続されてもよい。いくつかの実施形態では、CPU 11は、タスクプールを通して間接的にのみセル12と通信する。他の実施形態では、CPU 11はまた、中間物としてタスクプールを使用することなく、セル12と直接通信してもよい。

#### 【0025】

いくつかの実施形態では、システム10は、1つを上回るCPU 11と、1つを上回るタスクプール13とを含んでもよく、その場合では、特定のCPU 11が、特定のタスクプール13と排他的に相互作用してもよい、または複数のCPU 11が、1つまたはそれを上回るタスクプール13を共有してもよい。さらに、各連帯セルは、1つを上回るタスクプール13と相互作用するように構成されてもよい。代替として、特定のセルが、例えば、高性能または高セキュリティの文脈において、単一の指定されたタスクプールと相互作用するように構成されてもよい。

10

#### 【0026】

種々の実施形態では、セルは、以下の3つの条件が満たされると、タスクプールとオーミックに（プラグアンドプレイ）、または無線で（オンザフライ）動的にペアリングされてもよい。

1) セルが、タスクプールとオーミックに、または無線で通信することが可能である。タスクプールへの接続は、タスクプール自体におけるポートを通して、またはタスクプールに接続されるスイッチングファブリックを通してであり得る。

2) タスクプールが、例えば、パスワードの有無にかかわらず、伝統的なWi-Fi、Bluetooth（登録商標）、もしくは類似するペアリングを通した、スマートフォンもしくはタブレット上で起動するグラフィカルソフトウェアプログラムを通した手動の、または任意の他のセキュアもしくは非セキュアな方法による、ユーザからの入力を使用して、セルによって送信されたエージェントを信頼できるものとして認識する。

20

3) タスクプール内の利用可能なタスクのうちの少なくとも1つが、連帯セルの能力と互換性がある。

#### 【0027】

複数のタスクプールを伴うマルチプロセッサ環境の場合では、前述の動的なペアリング条件が適用され、但し、所与のセルが、タスクプールのうちの1つのみと協働するようにロックまたは制限され得、そうでなければ、セルは、第1発見ベース、ラウンドロビンベース、または任意の他の選択スキームを使用して、1つまたはそれを上回るタスクプールと接続し得る。また、タスクプール内のタスクに優先順位を割り当てることも可能であり、それによって、セルは、優先順位の高いタスクを優先し、より優先順位の高いタスクに別様に従事しないとき、より優先順位の低いタスクを扱う。

30

#### 【0028】

CPU 11は、ソフトウェアプログラムを実行するために使用される、任意の単一もしくはマルチコアプロセッサ、アプリケーションプロセッサ、またはマイクロコントローラであってもよい。システム10は、パーソナルコンピュータ、スマートフォン、タブレット、またはモノのインターネットデバイス上に実装されてもよく、その場合では、CPU 11は、Intel（登録商標）Pentium（登録商標）または直近のコンピューティング環境にローカルである、もしくはそれからリモートであるマルチコアプロセッサ等の任意のパーソナルコンピュータ、中央プロセッサ、またはプロセッサクラスタであってもよい。代替として、システム10は、スーパーコンピュータ上に実装されてもよく、CPU 11は、縮小命令セットコンピュータ（「RISC」）プロセッサ、アプリケーションプロセッサ、マイクロコントローラ、または同等物であってもよい。

40

#### 【0029】

他の実施形態では、システム10は、Beowulfクラスタ等のローカル接続された一連のパーソナルコンピュータ上に実装されてもよく、その場合では、CPU 11は、ネットワーク化コンピュータの全て、そのサブセット、またはそのうちの1つの中央プロセッサを含んでもよい。代替として、システム10は、リモート接続されたコンピュータの

50



ネットワーク上に実装されてもよく、その場合では、CPU 11は、サーバまたはメインフレームのための現在公知の、または後に開発される中央プロセッサであってもよい。CPU 11が、ここで説明されるシステム 10内で対象並列処理方法を実施する特定の様式は、CPUのオペレーティングシステムによって影響を受け得る。例えば、CPU 11は、以下に説明されるように、タスクプール 13を認識し、これと通信し、コンピューティング要件をスレッドに分割するようにこれをプログラムすることによって、システム 10内で使用するために構成されてもよい。

#### 【0030】

システム 10は、本明細書に説明される機能性を実装するように修正または別様に構成され得るオペレーティングシステムを有する、任意のコンピュータまたはコンピュータネットワーク上に遡及的に実装され得ることがさらに想定される。当分野で公知であるように、処理されるべきデータは、例えば、ランダムアクセスもしくは読取専用メモリ、CPU 11のためのキャッシュメモリ、またはフラッシュメモリおよび磁気記憶装置等の他の形態のデータ記憶装置のアドレス指定可能領域または区画の文脈において、メモリユニット 15内に含有される。メモリユニット 15は、処理されるべきデータだけではなく、処理されたデータの結果を置く場所も含有する。例えば、システム 10にデータを戻し得るスマートメータおよび自動車計器の場合のように、またはある機構を作動し得るロボットおよびモータコントローラの場合のように、全てのタスクが、メモリユニット 15にアクセスすることを要求されるわけではない。

#### 【0031】

各セル 12は、1つまたはそれを上回るタスク/スレッドを実行することが可能である、概念的または論理的に独立した算出ユニットである。セル 12は、マイクロコントローラ、マイクロプロセッサ、アプリケーションプロセッサ、「ダム」スイッチ、または Beowulf クラスタ内のマシン等の独立型コンピュータであってもよい。

#### 【0032】

セル 12は、CPUの機能、または、例えば、周囲監視およびロボットアクチュエータ等のCPU 11とは無関係の機能を補完する、それらの全てを実施する、またはそれらの限定された範囲を実施するように構成される、汎用または専用コプロセッサであってもよい。専用プロセッサは、特殊化タスクを実施するように設計、プログラム、または別様に構成される、専用ハードウェアモジュールであってもよい、またはこれは、グラフィック処理、浮動小数点演算、もしくはデータ暗号化等の特殊化タスクを実施するように構成される、汎用プロセッサであってもよい。

#### 【0033】

ある実施形態では、専用プロセッサである任意のセル 12はまた、メモリにアクセスし、それに書き込み、以下に説明されるように、記述子を実行し、ならびに他のソフトウェアプログラムを実行するように構成されてもよい。

#### 【0034】

さらに、任意の数のセル 12が、異種コンピューティング環境、つまり、AMDベースおよび/またはIntelベースのプロセッサ、または32ビットおよび64ビットプロセッサの混合物等の1種類を上回るプロセッサを使用するシステムを含み得る。

#### 【0035】

各セル 12は、以下のイベントのシーケンスに例証されるように、1つまたは複数の特殊化タスクを実施するように構成される。ポーリングフェーズ中、各セルは、整合タスクが見出されるまで、タスクプールにエージェントを周期的に送信する。この整合を促進するために、セルおよびタスクプールは両方とも、送受信機を具備してもよい。タスクプールの場合は、送受信機は、タスクプール自体に、またはタスクプールが接続されるスイッチングファブリックに位置してもよい。タスクプール内にタスク整合が見出されると、タスクプールは、セルに肯定応答を伝送する。次のステップは、「通信チャネル」フェーズである。通信チャネルフェーズ中、セルは、タスクを受信し、そのタスクを実行し始める。一実装では、いったん第1のタスクが完了すると、通信チャネルは、連帯セルが「ボ

10

20

30

40

50

ーリング」および「肯定応答」フェーズを繰り返す必要なく別のタスクをフェッチし得るように維持される。

【0036】

システム10は、複数のセルを含んでもよく、セルのいくつかは、他のセルと同一のタスクタイプを実施することが可能であり、それによって、システム10内に冗長性をもたらし。所与のセル12によって実施されるタスクタイプのセットは、別のセルによって実施されるタスクタイプのセットのサブセットであってもよい。例えば、図1では、システム10は、集約算出問題をタスクの群に分割し、第1のタイプ、第2のタイプ、および第3のタイプのタスクをタスクプール13に投入してもよい。第1のセル12Aは、第1のタイプのタスクのみを実施することが可能であり得、第2のセル12Bは、第2のタイプのタスクを実施することが可能であり得、第3のセル12Cは、第3のタイプのタスクを実施することが可能であり得、第4のセル12Dは、第2または第3のタイプのタスクを実施することが可能であり得、第5のセル12Nは、全ての3つのタスクタイプを実施することが可能であり得る。システム10は、所与のセルがシステム10から除去される（または現在ビジーである、または別様に利用不可能である）場合、システム10がシームレスに機能し続け得るように、この冗長性ととも構成されてもよい。さらに、セルがシステム10に動的に追加される場合、システム10は、性能向上の利益を用いて、シームレスに機能し続け得る。

【0037】

ここで図1および2を参照すると、タスクプール13は、CPU11によってアクセス可能である物理メモリの領域を占有し得る。代替として、タスクプール13は、MACアドレスまたはIPアドレスによってアクセス可能であってもよい。複数の実施形態が、タスクプール13に関して想定され、これは、同一の2Dもしくは3DモノリシックIC内でCPUとともに物理的に位置してもよい、またはこれは、独立型ICとして実装され、コンピュータボード、スマートフォン、タブレット、ルータ、もしくはモノのインターネットデバイスに物理的に相互接続されてもよい。さらなる代替実施形態では、タスクプールは、複数のCPU11システム間で共有される、または所与のCPU11の専用であり得る、独立型マルチポート、有線、および/または無線接続されたデバイスであってもよい。タスクプール13はまた、セル12によってアドレス指定可能であってもよい。タスクプール13は、CPU11およびセル12によって最大アクセス速度を提供するために、専用ハードウェアブロック内に配置されてもよい。代替として、タスクプール13は、ソフトウェアベースであってもよく、タスクプール13のコンテンツは、ハードウェアベースの実施形態に類似して、メモリ内に記憶されるが、データ構造によって表される。

【0038】

CPU11によって投入されると、タスクプール13は、1つまたはそれを上回るタスクスレッド21を含有する。各タスクスレッド21は、CPU11に付与されたより大規模な集約算出要件の構成要素またはサブセットであり得る、算出タスクを表す。一実施形態では、CPU11は、並行して実行可能なスレッド21を伴うタスクプール13を初期化し、次いで、それに投入してもよい。各スレッド21は、1つまたはそれを上回る離散タスク22を含んでもよい。タスク22が、タスクタイプと、記述子とを有してもよい。タスクタイプは、どのセル12がタスク22を実施することが可能であるかを示す。タスクプール13はまた、同一タイプを有するタスク22に優先順位を付けるために、タスクタイプを使用してもよい。一実施形態では、タスクプール13は、システム10内に存在する連帯セル12、各セルが実施することが可能なタスク22のタイプ、および各セルが現在タスク22を処理しているかどうかを記録する、優先順位テーブル（図示せず）を維持してもよい。タスクプール13は、以下に説明されるように、適格なタスク22のどれを要求セルに割り当てるかを判定するために、優先順位テーブルを使用してもよい。

【0039】

いくつかの実施形態では、CPU11は、タスクプールからタスクまたはスレッドを読み出し、それを実行してもよい。さらに、CPU11は、古くなった、壊れた、膠着した

10

20

30

40

50

、または誤ったと判定される任意のタスクを中断してもよい。そのような場合では、CPU 11は、タスクをリフレッシュし、後続処理のために利用可能にしてもよい。例えば、人工知能によって要求され得るように、CPU 11が適応タスク管理を実装することを妨げるものはなく、その後、CPU 11は、未完成の既存のスレッド21内のタスクを追加、除去、または変更してもよい。

#### 【0040】

記述子は、存在する場合、実行されるべき具体的命令、実行のモード、処理されるべきデータの場所（例えば、アドレス）、およびタスク結果を置くための場所のうちの1つまたはそれを上回るものを含有してもよい。結果を置くための場所は、多くの場合、結果をメモリ内に記憶するのではなく、それらをディスプレイに提示する、アニメーションおよびマルチメディアタスクの場合等では、随意である。さらに、タスク記述子は、リンクされたリストのように、ともに連鎖されてもよく、したがって、処理されるべきデータは、記述子がともに連鎖されない場合よりも少ないメモリ呼び出しでアクセスされ得る。ある実施形態では、記述子は、ヘッダおよびメモリ場所の複数の参照ポインタを含有するデータ構造であり、タスク22は、データ構造のメモリアドレスを含む。ヘッダは、実行されるべき機能または命令を定義する。第1のポインタが、処理されるべきデータの場所を参照する。第2の随意的ポインタが、処理されたデータを置くための場所を参照する。記述子が、連続的に実行されるべき別の記述子にリンクされている場合、記述子は、次の記述子を参照する第3のポインタを含んでもよい。記述子がデータ構造である代替実施形態では、タスク22は、完全なデータ構造を含んでもよい。

#### 【0041】

スレッド21はさらに、タスク22が実施され得る順序および性能の順序に影響を及ぼす任意の条件を記述する、「レシピ」を備えてもよい。レシピによると、タスク22は、ブール演算に従って、連続的に、並行して、順不同に、相互依存的に、または条件的に実行されてもよい。例えば、図2では、スレッド21Aは、4つのタスク、すなわち、22A、22B、22C、および22Dを備える。例証される実施形態では、第1のタスク22Aは、第2のタスク22Bまたは第3のタスク22Cのいずれかが始まり得る前に完了しなければならない。レシピによると、いったん第2のタスク22Bまたは第3のタスク22Cのいずれかが完了すると、第4のタスク22Dが、始まり得る。

#### 【0042】

スレッド21はまた、相互依存的であってもよい。例えば、図2に示されるように、スレッド21B内のブール演算に起因して、完了したタスク22Cは、スレッド21B内のタスクの処理が継続することを可能にしてもよい。タスクプール13は、タスク22が、これが依存する別のタスク22の完了を待機している間、タスク22をロックしてもよい。タスク22がロックされると、これは、セルによって取得されることはできない。スレッド21のタスク22が完了すると、タスクプール13は、CPU 11に完了を通知してもよい。CPUは、次いで、完了したスレッド21を越えて処理を進めてもよい。

#### 【0043】

セルは、有利なこととして、相互に、およびCPU 11と連帯を維持し、それによって、システム10が、タスクプール13からタスクを自律的かつ先回りして読み出すことによって、複雑な算出を実施することに役立つ。セル12は、それらがCPU 11または任意の他のコプロセッサから独立して作用し得るという点において、自律的に作用する。代替として、セルは、CPUに依拠して作用する、またはそれによって直接命令されてもよい。各セルは、さらなる処理のためにセルが利用可能になるとすぐに、タスクプール13からタスク22を探求するという点において、先回りして作用する。

#### 【0044】

より具体的には、ある実施形態では、セル12は、エージェント30を送信し、タスクプールに問い合わせ（検索し）、完了を要求する、ロックされていない、セルによって実施され得るタスクタイプを有する利用可能なタスク22を読み出すことによって、タスクプールからタスクを取得する。典型的には、システム10は、連帯共処理セルと同数のエ

ージェントを有する。本文脈では、エージェントは、概して、エージェントがソースアドレス、宛先アドレス、およびペイロードを具備し得るという点で、ネットワークの意味におけるデータフレームと類似する。ある実施形態では、宛先アドレスは、エージェント30がタスク22を探求しているとき、タスクプール13のアドレスであり、宛先アドレスは、エージェント30がタスク22を伴うそのセルに戻っているとき、対応するセル12のアドレスである。対応して、ソースアドレスは、エージェント30がタスク22を探求しているとき、セル12のアドレスであり、ソースアドレスは、エージェント30がタスク22を伴うそのセルに戻っているとき、タスクプール13のアドレスである。

#### 【0045】

加えて、ソースおよび宛先アドレスは、フレーム同期を促進し得る。つまり、システム10は、ペイロードデータから明確にアドレスを区別するように構成されてもよく、したがって、エージェント30のコンテンツが読み取られると、宛先アドレスは、フレームの開始を示し、ソースアドレスは、フレームの終了を示し、逆もまた同様である。これは、ペイロードが、アドレス間に置かれるとき、サイズを変動することを可能にする。可変サイズのペイロードの別の実施形態では、エージェント30は、ペイロードサイズを示すヘッダを含んでもよい。ヘッダ情報は、データ完全性を検証するために、ペイロードと比較されてもよい。さらに別の実施形態では、ペイロードは、固定長であってもよい。エージェント30がそのコプロセッサセルによってタスクプール13にディスパッチされると、ペイロードは、セル12が実施し得るタスクのタイプの識別情報を含む。エージェント30がタスクプール13から戻ると、ペイロードは、メモリ場所または完全な記述子データ構造のいずれかの形態において、タスク22の記述子を含む。

#### 【0046】

他の実施形態では、エージェント30のいくつかまたは全ては、それらのそれぞれの対応するセル12の自律的な代表である。つまり、各エージェント30は、その対応するセル12によってディスパッチされ、セルがアイドルしている、または付加的な処理を実施することが可能であるときはいつでも、タスク22を読み出してもよい。このように、連帯セル12の処理容量は、セルがCPU11からの命令をアイドルして待機する必要がない限り、より完全に活用され得る。本アプローチは、タスクプールからタスクを読み出すために、セルに要求を送信する必要性からCPUを解放することによって、CPUオーバーヘッドを低減させる付加的な利益も有する。これらの利点は、補助モジュールおよびコプロセッサがメインCPUからの命令に依存する伝統的なコンピュータアーキテクチャよりも、システム10を効率的にする。

#### 【0047】

さらに、連帯セル12A - 12nは、スレッド自体の特定の組成に関して曖昧である。むしろ、エージェントは、その対応するセルの能力と、タスクプール13内の完了すべき利用可能なタスク22との間の整合を見出すことにのみ関与する。つまり、タスクプール13内に利用可能なタスク22が存在し、利用可能なタスク22がセルの能力と整合する限り、本システムは、セルの処理容量を効果的に利用し得る。

#### 【0048】

連帯セル12A - 12nのいくつかまたは全ては、相互に独立して機能してもよい、または直接、スイッチングファブリック14を通して、タスクプール13を通して、もしくはCPUからのコマンドもしくは要求に従って相互に通信し、別の連帯セルを呼び出し、データを処理、移動、もしくは伝送する際に補助してもよい。一実施形態では、エージェント30Aは、レディタスク22のタスクタイプと、セル12Aが実施することが可能であるタスクのタイプとの間の整合を検索してもよい。本アーキテクチャは、CPU11が生成するように構成されるタスクのタイプのハードコーディングを伴ってもよい。したがって、タスクプール13が3つのタイプのタスク22を含むし、大規模な算出要件が第4のタイプのタスクを含む場合、この第4のタイプのタスクは、第4のタイプのタスクを実施することが可能なセルがシステム10に含まれる、または追加される場合であっても、タスクプール13内に置かれない場合がある。その結果、CPU11は、利用可能な処理

リソースをより完全に活用するために、第４のタイプのタスクを生成する方法を「学習する」または教示されるように構成され得る。

【００４９】

別の実施形態では、エージェント３０Ａは、セル１２Ａが実施することが可能である命令のうちの１つと整合する実行可能な命令のためのタスク２２記述子を検索する。整合タスク２２が見出されると、エージェント３０Ａは、整合タスク２２の記述子をセル１２Ａに送達し、その後、セル１２Ａは、タスク２２を処理し始める。特に、エージェント３０Ａは、記述子のメモリアドレスをセル１２Ａに送達してもよく、セル１２Ａは、メモリからデータ構造を読み出す。代替として、記述子の全データ構造がタスク２２内に含有される場合、エージェント３０Ａは、処理のためにセル１２Ａに完全なデータ構造を送達してもよい。記述子は、セル１２Ａに、どの命令を実行するか、処理されるべきデータが見出され得るメモリユニット１５内の場所、および結果が置かれるべきメモリ１５内の場所を知らせる。タスク２２の完了に応じて、セル１２Ａは、選択されたタスク２２のステータスを「完了するべき」から「完了した」に変更するようにタスクプール１３に通知する。さらに、いったんセル１２Ａがタスク２２を終了すると、セルは、そのエージェント３０Ａをタスクプール１３にディスパッチし、別のタスク２２を探求してもよい。

10

【００５０】

エージェント３０Ａ - ３０ｎのいくつかまたは全ては、システム１０の特定のアーキテクチャおよび／または実装による、例えば、Wi-Fiネットワーク、無線イーサネット（登録商標）、無線USB、無線ブリッジ、無線中継装置、無線ルータ、Zigbee（登録商標）、ANT+（登録商標）、またはBluetooth（登録商標）ペアリングを使用して、有線または無線でシステム１０を通して進行してもよい。ある実施形態では、エージェント３０は、タスクプール１３にレセプタ特徴を含めることによって、さらに、セル１２に送信機特徴を含めることによって、無線でタスクプール１３に誘導されてもよい。同様に、タスクプールは、タスクプールに送信機を、連帯セルに受信機を具備することによって、セルに無線で応答してもよい。本様式では、セルは、スイッチングファブリックの使用の有無にかかわらず、タスクプールと無線で通信し得る。

20

【００５１】

しかしながら、好ましい実施形態では、ある形態のスイッチングファブリック１４が、使用される。スイッチングファブリック１４は、システムリソース間のデータ転送およびアービトレーションのための接続を促進する。スイッチングファブリック１４は、種々のセルとタスクプールとの間に接続性を提供する、ルータまたはクロスバースイッチであってもよい。スイッチングファブリック１４はさらに、各連帯セル１２Ａ - １２ｎと、CPU１１、メモリユニット１５、および限定ではないが、ダイレクトメモリアクセスユニット、送信機、ハードディスクおよびそれらのコントローラ、ディスプレイおよび他の入力／出力デバイス、ならびに他のコプロセッサを含む、伝統的なシステム構成要素等のシステムリソースとの間に接続性を提供してもよい。セル１２Ａ - １２ｎは、スイッチングファブリック１４に物理的に接続されてもよい、またはセルは、無線で接続されてもよい。

30

【００５２】

システム１０へのセルの無線接続は、システム１０内で使用するためのセルの動的な追加および／または除去を促進する。例えば、CPU１１は、他のセルシステムからセルを補充し、動的拡張および性能向上を可能にしてもよい。本様式では、２つまたはそれを上回るセルシステム（例えば、ネットワーク）が、連帯セルを共有し得る。一実施形態では、アイドル状態になるセルは、付加的な処理リソースの必要性を有する、すなわち、完了する必要がある利用可能な処理タスクを有する、別のシステムを探す、および／またはそれによって補充されてもよい。同様に、システム１０は、特定のタスクのための付加的なセルのクラスタを組み込むことによって、性能を拡張させてもよい。例えば、システム１０は、これらのタスクを実施することが可能な近傍セルを組み込むことによって、暗号化／復号化機能、またはオーディオおよび／もしくはビデオデータの処理の性能を増進させてもよい。

40

50

## 【 0 0 5 3 】

望ましくない接続を防止するために、CPU 11は、タスクプール13に、信頼できるおよび/または信頼できないセル、ならびに認証要件もしくはプロトコルのリスト、または代替として、それらを識別するための基準を提供してもよい。さらに、タスクプール自体は、低性能、信頼性のない接続、データ処理量の不良、または悪意のある、もしくは別様に不適切な活動の疑いに基づいて、特定のセルを除外してもよい。種々の実施形態では、セル12は、ユーザによって、スマートフォン、タブレット、または他のデバイスもしくはアプリケーションの使用を通して、タスクプール13に追加される、またはタスクプール13から除外されてもよい。一実施形態では、グラフィカルアプリケーションインターフェースは、ネットワークから特定のセルを追加または除去した結果、利用可能なセルおよび他のデバイスの場所、性能向上、または性能ペナルティ等の有用な統計および/またはアイコン情報をユーザに提供してもよい。

10

## 【 0 0 5 4 】

代替実施形態では、共処理セルのいくつかまたは全ては、通信のためのスイッチングファブリック14を要求しない有線構成等によって、タスクプール13に直接接続してもよい。セルの有線接続はさらに、有線接続が周辺機器デバイスの物理的（例えば、手動）統合および抽出であり得るが、上記に議論される無線構成に類似するシステム10の動的拡張および縮小を促進してもよい。いずれの場合も、システム10の変更を考慮して、コプロセッサがCPU 11を再プログラムすることなく追加および除去され得るため、本システムのスケーラビリティは、従来の並列処理スキームを上回って大幅に増進される。

20

## 【 0 0 5 5 】

ここで図3を参照すると、ネットワーク300が、CPU 302と、第1のメモリ304と、第2のメモリ306と、タスクプール308と、スイッチングファブリック310と、タイプAタスクを実施（実行）するように構成される第1の共処理セル312と、タイプBタスクを実施するように構成される第2のセル314と、タイプCタスクを実施するように構成される第3のセル316と、タイプAおよびタイプBタスクの両方を実施するように構成される第4のセル318とを含む。示されるように、タスクプール308は、（例えば、CPU 302によって）タスクタイプAのタスク（またはタスクスレッド）330および332、タスクタイプBのタスク334および336、ならびにタスクタイプCのタスク340および342を投入される。ある実施形態では、各セルは、好ましくは、一意の専用エージェントを有する。特に、セル312は、エージェント320を含み、セル314は、エージェント322を含み、セル316は、エージェント324を含み、セル318は、エージェント326を含む。各エージェントは、好ましくは、その関連付けられるセルが実施するように構成されるタスクのタイプ、例えば、単一タスクまたはタスクA、B、Cの組み合わせを識別する、情報フィールドまたはヘッダを含む。

30

## 【 0 0 5 6 】

動作中、セルがアイドル状態であるか、または別様に利用可能な処理容量を有するかのいずれかのとき、そのエージェントは、タスクプールに先回りして問い合わせ、その特定のセルに適切である任意のタスクがタスクキュー内にあるかどうかを判定する。例えば、セル312は、そのエージェント320をディスパッチし、タスクタイプAに対応するタスク330および332の一方または両方を読み出してもよい。同様に、セル314は、そのエージェント322をディスパッチし、タスクタイプBに対応するタスク334または336のいずれかを（それらの相対的優先順位に応じて）読み出してもよく、以下同様である。タスクタイプAおよびBを実施するように構成されるセル318等、1つを上回るタスクタイプを実施することが可能であるセルに関して、エージェント326は、タスク330、332、334、および/または336のうちのいずれか1つを読み出してもよい。

40

## 【 0 0 5 7 】

タスクプールからタスクを読み出すと、セルは、次いで、典型的には、第1のメモリ304内の特定の場所からデータを読み出し、そのデータを処理し、処理されたデータを第

50

2のメモリ306内の特定の場所に記憶することによって、そのタスクを処理し得る。タスクが完了すると、セルは、タスクプールに通知し、タスクプールは、完了したものとしてタスクをマーキングし、タスクプールは、タスクが完了したことをCPUに通知する。代替として、タスクプールは、タスクスレッドが単一タスク、一連のタスク、またはプール組み合わせのタスクを含み得る限り、タスクスレッドが完了すると、これをCPUに通知してもよい。有意なこととして、セルによるタスクの読み出しおよびデータの処理は、CPUと種々のセルとの間の直接通信を伴わずに生じ得る。

#### 【0058】

ここで図4を参照すると、モノのインターネットネットワーク400が、コントローラ(CPU)402と、タスクプール408と、種々のデバイス410-422とを含み、そのいくつかまたは全てが、処理容量を具現化する、集積回路(IC)チップまたは他の構成要素等の関連付けられる、または埋め込まれたマイクロコントローラを含む。非限定的実施例として、本デバイスは、電球410、サーモスタット412、電気レセプタクル414、電力スイッチ416、電化製品(例えば、トースター)418、車両420、キーボード422、およびネットワークとインターフェースをとることが可能な事実上任意の他のプラグアンドプレイデバイスまたはアプリケーションを含んでもよい。

#### 【0059】

例証される実施形態では、コントローラ402は、スマートフォン、タブレット、ラップトップ、またはネットワーク上の種々のデバイスとのユーザ相互作用を促進するためのディスプレイ404およびユーザインターフェース(例えば、キーパッド)406を含み得る他のデバイスであってもよい。コントローラ402の処理容量(例えば、帯域幅)がネットワークを適正にサポートするために不十分であり得る限りにおいて、コントローラは、例えば、図5と併せて以下に説明されるように、周辺機器デバイスからタスクプールを介して、処理リソースを効果的に採取または補充してもよい。

#### 【0060】

ここで図5を参照すると、モノのインターネットネットワーク500の使用事例が、近傍(または別様に利用可能な)デバイスの動的な利用を例証する。ネットワーク500は、主要制御ユニット502(例えば、ラップトップ、タブレット、またはゲームデバイス)と、タスクプール504と、第1のコプロセッサデバイス506と、第2のコプロセッサデバイス508とを含む。ネットワーク500の文脈における例示的使用事例が、ここで説明される。

#### 【0061】

ユーザが、自身のラップトップコンピュータ502でビデオゲームをプレイしていると仮定する。ビデオゲームは、詳細なコンピュータ生成イメージを要求し、おそらく、ラップトップ502における処理能力は、単一の写実的に見えるキャラクタをレンダリングするために十分であるが、第2のキャラクタが画面上に導入されると、画像品質は、劣化し、キャラクタの移動はもはや、連続的ではない。本発明は、ユーザの近傍内に位置する、または別様にユーザに利用可能な十分に利用されていないコンピュータリソースの処理能力を利用する方法を提案する。

#### 【0062】

付加的な処理能力の必要性に対処するために、ラップトップ502は、タスクプール504に接続する。この点で、ラップトップ自体は、タスクプールを具備してもよい、またはタスクプールは、ラップトップ502から無線到達範囲内に位置する外部デバイスまたはアプリケーションの形態であってもよい。外部タスクプールの場合では、タスクプール自体は、ポートを伴うスイッチングファブリックの役割を果たし、複数の共処理セルへの接続を可能にし得る。ラップトップ502は、算出集約的タスクをタスクプール504に投入する。スマートフォン508等の近傍の十分に利用されていないデバイスは、続けてタスクプール504に接続し、整合タスクタイプをフェッチするためにそのエージェントを送信する。その結果、スマートフォン508は、ラップトップ502をシームレスに補助するコプロセッサになり、それによって、ビデオゲーム体験を向上させる。同一の方法

10

20

30

40

50

は、他の十分に利用されていない処理リソースが存在し、必要とされる場合に繰り返されてもよい。実際、利用可能な電球 506 の処理能力であっても、ラップトップへのコプロセッサになり得る。

【0063】

図 6 は、例示的並列コンピューティング環境の動作を例証する、フローチャートである。特に、方法 600 が、タスクをタスクプールに投入するステップ（ステップ 602）と、1 つまたはそれを上回るエージェントを、1 つまたはそれを上回る対応するセルからタスクプールに先回りしてディスパッチするステップ（ステップ 604）と、タスクを読み出し、処理するステップ（ステップ 606）と、タスクスレッドが実施されたことをタスクプールおよび CPU に通知するステップ（ステップ 608）とを含む。方法 600 はさらに、必要に応じて、付加的なデバイスをネットワークに動的に組み込むステップ（ステップ 610）を含む。

10

【0064】

したがって、タスクプールと、第 1 のタスクをタスクプールに投入するように構成される、コントローラと、タスクプールから第 1 のタスクを先回りして読み出すように構成される、第 1 のコプロセッサとを含む処理システムが、提供される。

【0065】

ある実施形態では、第 1 のコプロセッサは、コントローラと通信することなく、タスクプールから第 1 のタスクを読み出すように構成される、第 1 のエージェントを備える。

【0066】

20

ある実施形態では、第 1 のタスクは、第 1 のタスクタイプの印を含み、第 1 のコプロセッサは、第 1 のタイプのタスクを実施するように構成され、第 1 のエージェントは、第 1 のタイプのタスクに関してタスクプールを検索するように構成される。

【0067】

ある実施形態では、第 1 のコプロセッサはさらに、第 1 のものを処理し、第 1 のタスクの完了に応じて、タスクプールに通知するように構成され、タスクプールは、第 1 のタスクの完了に応じて、コントローラに通知するように構成される。

【0068】

ある実施形態では、コントローラおよび第 1 のコプロセッサは、相互にタスクプールを通してのみ通信するように構成される。

30

【0069】

ある実施形態では、コントローラおよび第 1 のコプロセッサは、相互に直接、タスクプールを通して通信するように構成される。

【0070】

ある実施形態では、第 1 のコプロセッサは、これが利用可能な処理容量を有していることを判定し、判定に応答してエージェントをタスクプールにディスパッチするように構成される。

【0071】

ある実施形態では、コントローラはさらに、第 2 のタスクをタスクプールに投入するように構成され、本システムはさらに、タスクプールから第 2 のタスクを先回りして読み出すように構成される第 2 のエージェントを有する、第 2 のコプロセッサを備える。

40

【0072】

ある実施形態では、第 2 のタスクは、第 2 のタスクタイプの印を含み、第 2 のコプロセッサは、第 2 のタイプのタスクを実施するように構成され、第 2 のエージェントは、第 2 のタイプのタスクに関してタスクプールを検索するように構成される。

【0073】

ある実施形態では、コントローラおよびタスクプールは、モノリシック集積回路（IC）上に常駐し、第 1 のコプロセッサは、IC 上に常駐しない。

【0074】

別の実施形態では、コントローラ、タスクプール、ならびに第 1 および第 2 のコプロセ

50



ッサは、モノリシック集積回路（ＩＣ）上に常駐する。

【００７５】

第１のタスクタイプを有する第１のタスクをタスクプールに投入するように構成される、中央処理ユニット（ＣＰＵ）を含むタイプのネットワーク内の処理リソースを動的に制御する方法もまた、提供される。本方法は、第１のタスクタイプを実施するように第１のセルをプログラムするステップと、プログラムされた第１のセルをネットワークに追加するステップと、第１のセルからタスクプールに、第１のエージェントを先回りして送信するステップと、第１のエージェントによって、第１のタイプのタスクに関してタスクプールを検索するステップと、第１のエージェントによって、タスクプールから第１のタスクを読み出すステップと、第１のエージェントによって、第１のセルに第１のタスクをトランスポートするステップと、第１のセルによって、第１のタスクを処理するステップと、第１のセルからタスクプールに、第１のタスクが完了した通知を送信するステップとを含む。

10

【００７６】

ある実施形態では、本方法はまた、タスクプールによって、第１のタスクを完了しているものとしてマーキングするステップと、タスクプールからＣＰＵに、第１のタスクが完了した通知を送信するステップとを含む。

【００７７】

ある実施形態では、本方法はまた、第１のエージェントをタスクプールに先回りして送信するための述語として、第１のセルが利用可能な処理容量を有すると判定するように第１のセルを構成するステップを含む。

20

【００７８】

ある実施形態では、本方法はまた、プログラムされた第１のセルをネットワークに追加することに先立って、第１のセルを第１のデバイスに統合するステップを含む。

【００７９】

ある実施形態では、第１のデバイスは、センサ、電球、電力スイッチ、電化製品、バイオメトリックデバイス、医療デバイス、診断デバイス、ラップトップ、タブレット、スマートフォン、モータコントローラ、およびセキュリティデバイスのうちの１つを含む。

【００８０】

ある実施形態では、プログラムされた第１のセルをネットワークに追加するステップは、第１のセルとタスクプールとの間に通信リンクを確立するステップを含む。

30

【００８１】

ある実施形態では、（ＣＰＵ）はさらに、第２のタスクタイプを有する、第２のタスクをタスクプールに投入するように構成され、本方法はさらに、第２のタスクタイプを実施するように第２のセルをプログラムするステップと、第２のセルとタスクプールとの間に通信リンクを確立するステップと、第２のセルからタスクプールに、第２のエージェントを先回りして送信するステップと、第２のエージェントによって、第２のタイプのタスクに関してタスクプールを検索するステップと、第２のエージェントによって、タスクプールから第２のタスクを読み出すステップと、第２のエージェントによって、第２のセルに第２のタスクをトランスポートするステップと、第２のセルによって、第２のタスクを処理するステップと、第２のセルからタスクプールに、第２のタスクが完了した通知を送信するステップと、タスクプールによって、第２のタスクを完了しているものとしてマーキングするステップと、タスクプールからＣＰＵに、第２のタスクが完了した通知を送信するステップとを含む。

40

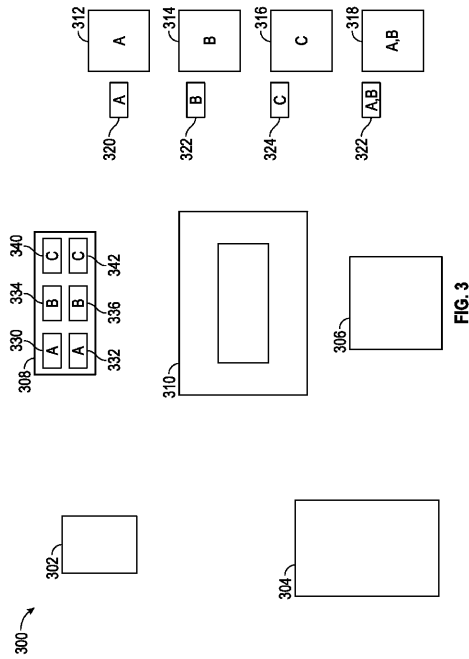
【００８２】

モノのインターネット（ＩｏＴ）コンピューティング環境内の分散処理リソースを制御するためのシステムもまた、提供され、集約コンピューティング要件を複数のタスクにパーティション化し、タスクをプール内に置くように構成される、ＣＰＵと、それぞれ、ＣＰＵとの直接通信を伴わずに、プールからタスクを先回りして読み出すように構成される、一意の専用エージェントを有する、複数のデバイスとを含む。

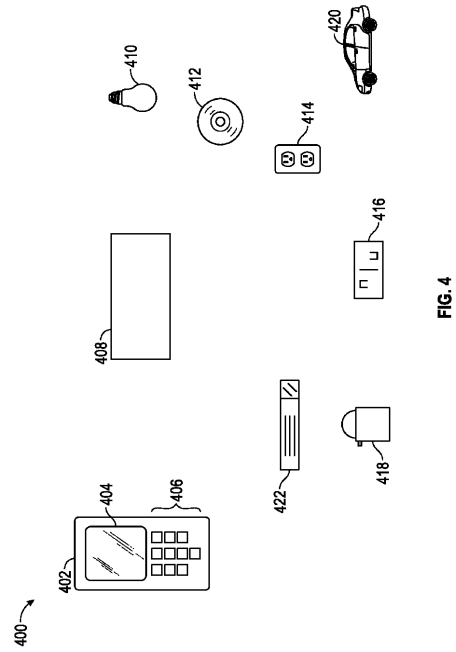
50



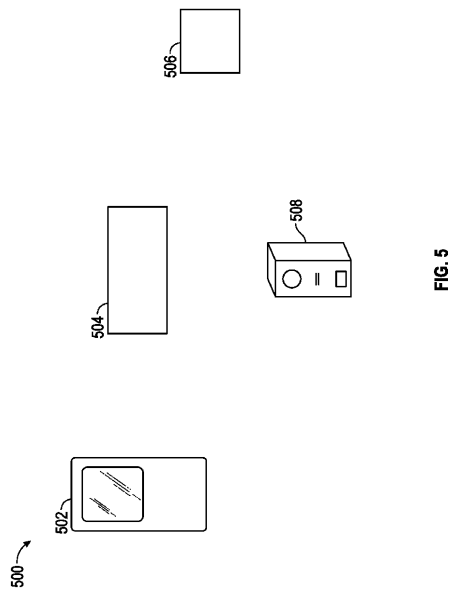
【図 3】



【図 4】



【図 5】



【図 6】

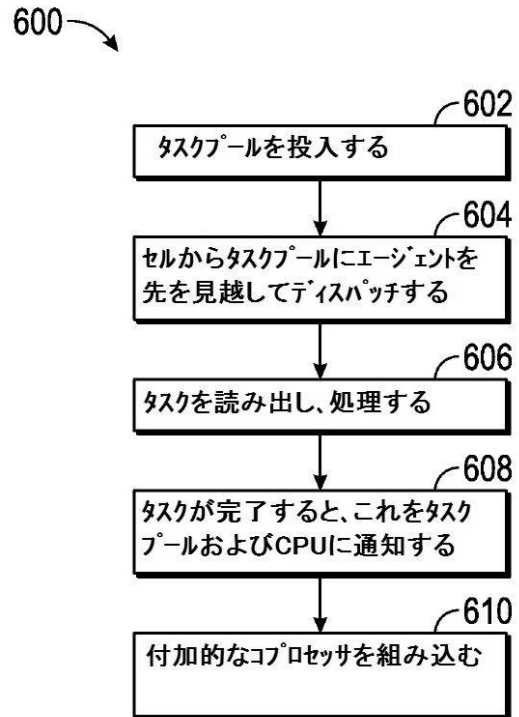


FIG. 6

## 【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US15/39993
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC(8) - G06F 9/46, 9/50, 9/54 (2015.01) CPC - G06F 9/4843, 9/5044, 9/546 According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) IPC(8): G06F 9/30, 9/46, 9/48, 9/50, 9/54 (2015.01) CPC: G06F 9/4843, 9/4881, 9/5044, 9/50, 9/546 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) PatSeer (US, EP, WO, JP, DE, GB, CN, FR, KR, ES, AU, IN, CA, INPADOC Data) Keywords: parallel, processor, thread, task, pool, core, call, notification, complete, scheduler, controller, dispatch, distribute, agent, message, queue		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2001/0000184 A1 (SEQUEIRA, W.) 05 April 2001; figures 2, 3B, 3C, 4A, 4; paragraphs [0043], [0044], [0055], [0057].	1-8, 9-11, 13, 16, 18, 20
Y		7-8, 12, 14-15, 17, 19
Y	US 2013/0160026 A1 (KUESEL, J. et al.) 20 June 2013; figure 3.	7
Y	US 2013/0036425 A1 (ZIMMERMANN, M.) 07 February 2013; figure 1; paragraphs [0024], [0026].	8, 14-15, 17, 19
Y	US 8,108,867 B2 (ROBISON, A.) 31 January 2012; figure 1; column 3, lines 57-62; column 4, lines 4-16.	12
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 12 September 2015 (12.09.2015)		Date of mailing of the international search report <div style="font-size: 1.5em; font-weight: bold; text-align: center;">06 OCT 2015</div>
Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300		Authorized officer Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

## フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(72)発明者 イニゲス , アリフォンソ  
アメリカ合衆国 アリゾナ 85203 , メサ , イー . レヒ ロード 732