

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6509895号  
(P6509895)

(45) 発行日 令和1年5月8日 (2019. 5. 8)

(24) 登録日 平成31年4月12日 (2019. 4. 12)

(51) Int. Cl.

F I

G 0 6 F 9 / 5 0 (2006. 01)

G 0 6 F 9 / 5 0 1 2 0 A

請求項の数 11 (全 31 頁)

(21) 出願番号 特願2016-556970 (P2016-556970)  
 (86) (22) 出願日 平成27年2月27日 (2015. 2. 27)  
 (65) 公表番号 特表2017-511940 (P2017-511940A)  
 (43) 公表日 平成29年4月27日 (2017. 4. 27)  
 (86) 国際出願番号 PCT/US2015/017892  
 (87) 国際公開番号 W02015/138152  
 (87) 国際公開日 平成27年9月17日 (2015. 9. 17)  
 審査請求日 平成30年2月6日 (2018. 2. 6)  
 (31) 優先権主張番号 61/950, 620  
 (32) 優先日 平成26年3月10日 (2014. 3. 10)  
 (33) 優先権主張国 米国 (US)  
 (31) 優先権主張番号 14/264, 838  
 (32) 優先日 平成26年4月29日 (2014. 4. 29)  
 (33) 優先権主張国 米国 (US)

(73) 特許権者 314015767  
 マイクロソフト テクノロジー ライセン  
 シング, エルエルシー  
 アメリカ合衆国 ワシントン州 9805  
 2 レッドモンド ワン マイクロソフト  
 ウェイ  
 (74) 代理人 100140109  
 弁理士 小野 新次郎  
 (74) 代理人 100075270  
 弁理士 小林 泰  
 (74) 代理人 100101373  
 弁理士 竹内 茂雄  
 (74) 代理人 100118902  
 弁理士 山本 修

最終頁に続く

(54) 【発明の名称】 デバイス固有又はユーザー固有のリソース使用プロファイルに基づくリソース管理

(57) 【特許請求の範囲】

【請求項 1】

コンピューティングデバイスのリソースを管理するためにコンピューティングデバイスによって実施される方法であって、

前記コンピューティングデバイスによって実行されている間、アプリケーションの第1のインスタンスによって使用されるリソースの第1の量を決定するステップであって、前記アプリケーションの前記第1のインスタンスに関連付けられるフォアグラウンドプロセス及び前記アプリケーションの前記第1のインスタンスに関連付けられる1つ又は複数のバックグラウンドタスクによって使用されるリソースの量を決定するステップを含む、ステップと、

前記第1の量に少なくとも基づいて、前記アプリケーションについての第1の予測されるリソース使用量を計算するステップと、

前記第1の予測されるリソース使用量をデータストアに格納するステップと、

前記データストアに格納された前記第1の予測されるリソース使用量に少なくとも基づいて、前記コンピューティングデバイスによって実行される前記アプリケーションの第2のインスタンスに割り当てられるリソースの推定される第2の量を決定するステップと、

前記リソースの前記第2の量が現在利用可能でないと判定することに応答して、前記アプリケーションの前記第2のインスタンスの実行を防止するステップと

を含む、方法。

【請求項 2】

10

20

前記リソースは、  
揮発性又は不揮発性のメモリーリソース、  
中央処理装置（ＣＰＵ）リソース、  
入出力（Ｉ／Ｏ）リソース、  
ネットワークリソース、  
電源リソース、及び  
センサーリソース  
のうちの少なくとも１つを含む、請求項１に記載の方法。

【請求項３】

前記第１の量に少なくとも基づいて、前記アプリケーションについての前記第１の予測されるリソース使用量を計算するステップは、

前記第１の量を、前記アプリケーションの１つ又は複数の以前に実行されたインスタンスに関連付けられる１つ又は複数の以前に決定された量と比較することによって、前記アプリケーションについての最大リソース使用量を得るステップと、

前記最大リソース使用量に基づいて、前記第１の予測されるリソース使用量を計算するステップと

を含む、請求項１に記載の方法。

【請求項４】

前記第１の量に少なくとも基づいて、前記アプリケーションについての前記第１の予測されるリソース使用量を計算するステップは、

前記第１の量及び前記アプリケーションの以前に実行されたインスタンスに関連付けられる１つ又は複数の以前に決定された量に基づいて、前記アプリケーションについての平均リソース使用量を得るステップと、

前記平均リソース使用量に基づいて、前記第１の予測されるリソース使用量を計算するステップと

を含む、請求項１に記載の方法。

【請求項５】

前記第１の量及び前記アプリケーションの以前に実行されたインスタンスに関連付けられる１つ又は複数の以前に決定された量に基づいて、前記アプリケーションについての前記平均リソース使用量を得るステップは、

前記第１の量を第１の重み付け係数で乗算して、第１の積を得るステップと、

少なくとも前記１つ又は複数の以前に決定された量に基づいて得られる移動平均量を第２の重み付け係数で乗算して、第２の積を得るステップと、

前記第１の積及び前記第２の積を合計するステップと

を含む、請求項４に記載の方法。

【請求項６】

前記第１の重み付け係数が前記第２の重み付け係数よりも大きい、請求項５に記載の方法。

【請求項７】

前記決定するステップ、前記計算するステップ及び前記格納するステップが、

第１のユーザーのために前記コンピューティングデバイスによって実行されている間に前記アプリケーションの前記第１のインスタンスによって使用される前記リソースの前記第１の量を決定するステップと、

前記第１の量に少なくとも基づいて、前記アプリケーション及び前記第１のユーザーについての前記第１の予測されるリソース使用量を計算するステップと、

前記第１の予測されるリソース使用量を前記データストアに格納するステップと、

前記データストアに格納された前記第１の予測されるリソース使用量に少なくとも基づいて、前記第１のユーザーのために前記コンピューティングデバイスによって実行される前記アプリケーションの前記第２のインスタンスに割り当てられる前記リソースの前記第２の量を決定するステップと

を含み、前記方法は、

第2のユーザーのために前記コンピューティングデバイスによって実行されている間に前記アプリケーションの第3のインスタンスによって使用される前記リソースの第3の量を決定するステップと、

前記第3の量に少なくとも基づいて、前記アプリケーション及び前記第2のユーザーについての第2の予測されるリソース使用量を計算するステップと、

前記第2の予測されるリソース使用量を前記データストアに格納するステップと、

前記データストアに格納された前記第2の予測されるリソース使用量に少なくとも基づいて、前記第2のユーザーのために前記コンピューティングデバイスによって実行される前記アプリケーションの第4のインスタンスに割り当てられる前記リソースの推定される第4の量を決定するステップと

をさらに含む、請求項1に記載の方法。

【請求項8】

コンピューティングデバイス上に実装されるシステムであって、

少なくとも1つのプロセッサと、

前記少なくとも1つのプロセッサによってアクセス可能なメモリーであって、前記少なくとも1つのプロセッサによる実行のためのコンポーネントを格納する、メモリーとを備え、前記コンポーネントは、

アプリケーションの第1のインスタンスによって使用される前記コンピューティングデバイスの少なくとも1つのリソースの第1の量を決定し、前記第1の量を決定することは、前記アプリケーションの前記第1のインスタンスに関連付けられるフォアグラウンドプロセス及び前記アプリケーションの前記第1のインスタンスに関連付けられる1つ又は複数のバックグラウンドタスクによって使用されるリソースの量を決定することを含み、

前記第1の量に少なくとも基づいて、前記アプリケーションについての第1の予測されるリソース使用量を計算し、

前記第1の予測されるリソース使用量をデータストアに格納し、

前記データストアに格納された前記第1の予測されるリソース使用量に少なくとも基づいて、前記コンピューティングデバイスによって実行される前記アプリケーションの第2のインスタンスに割り当てられるリソースの推定される第2の量を決定し、

前記リソースの前記第2の量が現在利用可能でないと判定することに応答して、前記アプリケーションの前記第2のインスタンスの実行を防止する

ように構成される、リソースマネージャー

を含む、システム。

【請求項9】

前記アプリケーションの前記第1のインスタンスに関連付けられる前記フォアグラウンドプロセスは、前記アプリケーションの前記第1のインスタンスのためのユーザーインターフェイスを提供する、請求項1に記載の方法。

【請求項10】

前記アプリケーションの前記第1のインスタンスに関連付けられる前記1つ又は複数のバックグラウンドタスクのうちの少なくとも1つは、前記フォアグラウンドプロセスと同時に実行される、請求項1に記載の方法。

【請求項11】

前記アプリケーションの前記第1のインスタンスに関連付けられる前記1つ又は複数のバックグラウンドタスクのうちの少なくとも1つは、前記フォアグラウンドプロセスによって開始される、請求項1に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、デバイス固有又はユーザー固有のリソース使用プロファイルに基づくリソース管理に関する。

10

20

30

40

50

## 【背景技術】

## 【0002】

[0001]今日のモバイルデバイス、特に今日の低価格のモバイルデバイスは、厳しいマルチタスクシナリオに見舞われている。規模において競争力があるようにするために、モバイルデバイスは、低価格及び／又は低電力の部品を用いて製造しなければならないことが多い。これは、特に、ランダムアクセスメモリー（ＲＡＭ）及びマルチメディアカード（ＭＭＣ）の両方を含む、モバイルデバイスを実装するために使用されるメモリーに対して当てはまる。このことの正味の影響は、メモリー管理についてページングに頼っている従来のモバイルデバイスが、同時に複数のプロセスを実行しようとするとき、すぐにフォールオーバーすることである。このことは、非常に乏しいユーザーエクスペリエンスや障害のあるマルチタスクシナリオにつながる。市場が製造コストの低減を求める一方で、モバイルデバイス用に開発されたアプリケーションやオペレーティングシステムがますます大きくなっているため、この問題は特に重要になってきている。

10

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0003】

[0002]従来、モバイルデバイス上での複数のプロセスの同時実行をサポートする十分なメモリーがないという問題は、以下の方法のうちの１つ又は複数の方法で扱われた：メモリー管理のためにページングを使用する；モバイルデバイス上でより多くのメモリーを設ける；アプリケーションやオペレーティングシステムのサイズに制限を設ける。上述したように、ページングは、非常に乏しいユーザーエクスペリエンスや障害のあるマルチタスクシナリオにつながる可能性がある。より多くのメモリーを設けると、モバイルデバイスのコスト、複雑さ及び消費電力が跳ね上がる。アプリケーションとオペレーティングシステムのサイズに制限を設けると、ユーザーにとって所望の機能を開発して提供することが極めて困難になる。少なくともこれらの理由から、モバイルデバイスのための既存のメモリー管理ソリューションは不十分である。今日のモバイルデバイスが、ユーザーエクスペリエンスに悪影響を与えず様々なマルチタスクシナリオを可能にする方法で、複数のプロセスを同時に実行することを可能にするために、より良いメモリー管理技術が必要とされている。

20

## 【課題を解決するための手段】

30

## 【0004】

[0003]モバイルコンピューティングデバイスなどのコンピューティングデバイス上で実行されるアプリケーションによる、揮発性及び不揮発性メモリー、中央処理装置（ＣＰＵ）、入出力（Ｉ／Ｏ）、ネットワーク、電力及びセンサーなどの様々なリソースの使用状況の追跡を可能にする、システム、方法、装置及びコンピュータプログラム製品が本願において説明される。本願で説明される技術は、アプリケーションに関連付けられるフォアグラウンドプロセス（例えば、アプリケーションのためのユーザーインターフェイスを現在提供しており、コンピューティングデバイスに関連付けられる表示の少なくとも一部を占有しているプロセス）によって使用されるリソース、並びに、コンピューティングデバイス上で別のプロセスとして実行されているアプリケーションに関連付けられる１つ又は複数のバックグラウンドタスクによって使用されるリソースの両方を追跡することによって、アプリケーションについてのリソースの使用を都合よく追跡する。アプリケーションによるリソース使用に関する履歴データは、アプリケーションのリソース使用量を予測するために利用され、それはその後格納される。アプリケーションのその後の起動は、予測されるリソース使用量が利用可能であるかどうかを条件とすることができる。予測されるリソース使用量が利用可能な場合、それをアプリケーションに割り当てることができ、アプリケーションを起動することができる。アプリケーションが予測されるリソース使用量を超える場合、それと同時に実行されている他のプロセスが、リソースを解放するために終了されてもよい。アプリケーションが実行する方法にかかわらず、アプリケーションについて新たなリソース使用データが取得され、その後、予測されるリソース使用量を調

40

50

整するために使用される。このように、アプリケーションのためのリソースの割り当ては、ユーザーの使用パターンに基づいて時間とともにユーザーに対して自動的に微調整することができ、コンピューティングデバイスは、特にマルチタスクシナリオ中、よりよいユーザーエクスペリエンス及び改善されたパフォーマンスを提供することが可能となる。複数のユーザーをサポートするコンピューティングデバイスのために、予測されるリソース使用量は、各ユーザー/アプリケーションの組み合わせについて格納することができる。

【0005】

[0004]本願に記載のシステム、方法、装置及びコンピュータープログラム製品はまた、特定のリソース使用行動に関連付けられる時刻、位置及びイベントなどの、アプリケーション及び/又はユーザーによるリソース使用に関連する他の情報を追跡することが可能である。そのような情報は、ユーザーがアプリケーションを起動しようとする前であっても、コンピューティングデバイスのリソースをアプリケーションに予測的に割り当てるために使用することができ、有利である。さらに、アプリケーション及び/又はアプリケーションデータは、ユーザーがそれらにアクセスすることが予期される前に、ある時点でメモリーへロードされてもよい。

【0006】

[0005]具体的には、コンピューティングデバイスのリソースを管理するためにコンピューティングデバイスによって実施される方法が、本願に記載される。例えば、リソースは、揮発性又は不揮発性メモリーリソース、CPUリソース、I/Oリソース、ネットワークリソース、電源リソース及びセンサーリソースのうちの少なくとも1つを含むことができる。この方法によれば、コンピューティングデバイスによって実行されている間、アプリケーションの第1のインスタンスによって使用されるリソースの第1の量が決定される。第1の量を決定するステップは、第1のアプリケーションに関連付けられるフォアグラウンドプロセス及びアプリケーションの第1のインスタンスに関連付けられる1つ又は複数のバックグラウンドタスクによって使用されるリソースの量を決定するステップを含んでもよい。アプリケーションのための第1の予測されるリソース使用量は、少なくとも第1の量に基づいて計算される。第1の予測されるリソース使用量はデータストアに格納される。次いで、コンピューティングデバイスによって実行されるアプリケーションの第2のインスタンスに割り当てられるリソースの第2の量が、少なくとも、データストアに格納された第1の予測されるリソース使用量に基づいて決定される。

【0007】

[0006]前述の方法の一実施形態では、少なくとも第1の量に基づいてアプリケーションについての第1の予測されるリソース使用量を計算するステップは、第1の量を、1つ又は複数の以前に実行されたアプリケーションのインスタンスに関連付けられる1つ又は複数の以前に決定された量と比較することにより、アプリケーションについての最大のリソース使用量を得るステップと、最大リソース使用量に基づいて、第1の予測されるリソース使用量を計算するステップとを含む。

【0008】

[0007]別の実施形態では、少なくとも第1の量に基づいてアプリケーションについての第1の予測されるリソース使用量を計算するステップは、第1の量及びアプリケーションの以前に実行されたインスタンスに関連付けられる1つ又は複数の以前に決定された量に基づいてアプリケーションについての平均リソース使用量を得るステップと、平均リソース使用量に基づいて第1の予測されるリソース使用量を計算するステップとを含む。さらにこの実施形態によれば、アプリケーションの平均リソース使用量を得るステップは、第1の量を第1の重み付け係数によって乗算して第1の積を得るステップと、少なくとも1つ又は複数の以前に決定された量に基づいて得られた移動平均量を第2の重み付け係数によって乗算して、第2の積を得るステップと、第1の積及び第2の積を合計するステップとを含む。実施形態によって、第1の重み付け係数は、第2の重み付け係数より大きくてもよいし、小さくてもよいし、又は等しくてもよい。

【0009】

[0008]前述の方法はまた、リソースの第2の量が現在利用可能ではないと判定するステップと、それに応答して、アプリケーションの第2のインスタンスの実行を防止し及び/又はコンピューティングデバイス上で現在実行されている1つ又は複数のプロセスを終了するなどの特定のアクションを実行するステップとを含む。

【0010】

[0009]前述の方法は、さらに、コンピューティングデバイスによって実行されている間、アプリケーションの第2のインスタンスによって使用されるリソースの量が第2の量を超えると判定するステップと、それに応答して、コンピューティングデバイス上で現在実行されている1つ又は複数のプロセスを終了するステップとを含んでもよい。

【0011】

[0010]前述の方法の別の実施形態では、決定するステップ、計算するステップ及び格納するステップは、第1のユーザーのためにコンピューティングデバイスによって実行されている間、アプリケーションの第1のインスタンスによって使用されるリソースの第1の量を決定するステップと、少なくとも第1の量に基づいてアプリケーション及び第1のユーザーについての第1の予測されるリソース使用量を計算するステップと、第1の予測されるリソース使用量をデータストア内に格納するステップと、データストア内に格納された第1の予測されるリソース使用量に少なくとも基づいて、第1のユーザーのためにコンピューティングデバイスによって実行されるべきアプリケーションの第2のインスタンスに割り当てられるべきリソースの第2の量を決定するステップとを含む。この実施形態によれば、方法はさらに、第2のユーザーのためにコンピューティングデバイスによって実行される間、アプリケーションの第3のインスタンスにより使用されるリソースの第3の量を決定するステップと、第3の量に少なくとも基づいてアプリケーション及び第2のユーザーについての第2の予測されるリソース使用量を計算するステップと、第2の予測されるリソース使用量をデータストア内に格納するステップと、データストア内に格納された第2の予測されるリソース使用量に少なくとも基づいて、第2のユーザーのためにコンピューティングデバイスによって実行されるべきアプリケーションの第4のインスタンスに割り当てられるべきリソースの第4の量を決定するステップとを含む。

【0012】

[0011]コンピュータープログラム製品もまた本願に記載される。コンピュータープログラム製品は、コンピューティングデバイスの少なくとも1つのプロセッサによって実行されると少なくとも1つのプロセッサに動作を実行させる、コンピュータープログラムロジックが記録されたコンピューター読み取り可能なメモリーを含む。動作は、アプリケーションを起動する要求を受け取るステップと、アプリケーションについての予測されるリソース使用量を得るステップであって、当該予測されるリソース使用量は、コンピューティングデバイスによるアプリケーションの1つ又は複数の以前の実行中のコンピューティングデバイスのリソースについての監視された使用に基づく、ステップと、予測されるリソース使用量を現在利用可能なリソースの量と比較することによって、リソースの十分な量が利用可能であるか否かを判定するステップと、少なくともリソースの十分な量が利用可能であると判定することに応答して、アプリケーションが起動することを可能にするステップとを含む。リソースは、揮発性又は不揮発性のメモリーリソース、CPUリソース、I/Oリソース、ネットワークリソース、電源リソース及びセンサーリソースのうちの1つを含んでもよい。アプリケーションについてのリソースの監視された使用は、アプリケーションに関連付けられるフォアグラウンドプロセス及びアプリケーションに関連付けられる1つ又は複数のバックグラウンドタスクによる、リソースの監視された使用を伴ってもよい。

【0013】

[0012]前述のコンピュータープログラム製品の一実施形態では、動作はさらに、少なくとも十分な量のリソースが利用可能でないと判定することに応答して、アプリケーションを起動させないステップを含む。

【0014】

10

20

30

40

50

[0013]別の実施形態では、動作はさらに、少なくとも十分な量のリソースが利用可能でないと判定することに対応して、１つ又は複数のプロセスを終了させるステップを含む。

[0014]さらに別の実施形態では、動作はさらに、起動された後にアプリケーションによって使用されるリソースの量が予測されるリソース使用量を超えると判定することに対応して、コンピューティングデバイス上で同時に実行されている１つ又は複数のプロセスを終了するステップを含む。

【 0 0 1 5 】

[0015]さらに別の実施形態では、受け取るステップは、第１のユーザーのためにアプリケーションを起動するための要求を受け取るステップを含み、得るステップは、アプリケーション及び第１のユーザーについて予測されるリソース使用量を得るステップを含む。

10

【 0 0 1 6 】

[0016]コンピューティングデバイス上に実装されたシステムもまた本願に記載される。システムは、少なくとも１つのプロセッサと、少なくとも１つのプロセッサによりアクセス可能なメモリとを含む。メモリは、少なくとも１つのプロセッサによる実行のためのコンポーネントを格納する。コンポーネントは、アプリケーション及びユーザーのうちの１つ又は複数によるコンピューティングデバイスの少なくとも１つのリソースの使用のパターンを監視し、使用のパターンに関連付けられる情報を格納し、及び格納された情報を使用して、少なくとも１つのプロセッサによる少なくとも１つのプロセスの実行前に、当該少なくとも１つのプロセスに対してリソースのうちの少なくとも一部を予測的に割り当てるように動作可能なリソースマネージャーを含む。少なくとも１つのリソースは、揮発性又は不揮発性のメモリリソース、ＣＰＵリソース、Ｉ／Ｏリソース、ネットワークリソース、電源リソース及びセンサーリソースを含むことができる。

20

【 0 0 1 7 】

[0017]前述のシステムの一実施形態では、リソースマネージャーは、コンピューティングデバイスのユーザー又は任意の他のエンティティがアプリケーションを起動する前に、リソースの少なくとも一部をアプリケーションに対して予測的に割り当てるように動作可能である。少なくとも１つのリソースがメモリを含む前述のシステムの別の実施形態では、リソースマネージャーは、さらに、コンピューティングデバイスのユーザー又は任意の他のエンティティがアプリケーションを起動する前に、アプリケーション又はアプリケーションに関連付けられるデータの少なくとも一部を予測的に割り当てられたメモリへロードするように動作可能である。

30

【 0 0 1 8 】

[0018]この概要は、詳細な説明において以下でさらに説明される概念の選択を簡略化された形で導入するために提供される。この概要は、特許請求される主題の主要な特徴や不可欠な特徴を特定することを意図するものではなく、特許請求される主題の範囲を限定するために使用されることも意図していない。さらに、特許請求される主題は、詳細な説明及び／又は本文書の他のセクションで説明される特定の実施形態に限定されるものではないことに留意すべきである。そのような実施形態は、例示の目的のみのために本願に提示されている。さらなる実施形態は、本願に含まれる教示に基づいて、当業者にとって明らかとなる。

40

【 0 0 1 9 】

[0019]本願に含まれ、本願の一部を形成する添付の図面は、本発明の実施形態を示すものであり、さらに、詳細な説明とともに、本発明の原理を説明し、当業者が本発明を作成及び使用することを可能にするのに役立つものである。

【図面の簡単な説明】

【 0 0 2 0 】

【図 １】[0020]実施形態による、デバイス固有又はユーザー固有のリソース使用プロファイルに基づいてリソース管理を実行するコンピューティングデバイスのブロック図である。

【図 ２】[0021]実施形態による、デバイス固有又はユーザー固有のリソース使用プロファ

50

イルに従ってコンピューティングデバイスのリソースを管理するためにコンピューティングデバイスによって実施される方法のフローチャートを示す。

【図 3】[0022]一実施形態による、アプリケーションについての予測されるリソース使用量を計算する方法のフローチャートを示す。

【図 4】[0023]別の実施形態による、アプリケーションについての予測されるリソース使用量を計算する方法のフローチャートを示す。

【図 5】[0024]実施形態による、アプリケーションについての平均リソース使用量を得る方法のフローチャートを示す。

【図 6】[0025]図 2 に示すフローチャートの方法に従って行うことができるさらなるステップのフローチャートを示す。

【図 7】[0026]図 2 に示すフローチャートの方法に従って行うことができるさらなるステップのフローチャートを示す。

【図 8】[0027]図 2 に示すフローチャートの方法に従って行うことができるさらなるステップのフローチャートを示す。

【図 9】[0028]実施形態による、ユーザー固有のリソース使用プロファイルに従ってコンピューティングデバイスのリソースを管理するための、コンピューティングデバイスによって実施される方法のフローチャートを示す。

【図 10】[0029]実施形態による、デバイス固有又はユーザー固有のリソース使用プロファイルに従ってアプリケーションの起動を管理するための方法のフローチャートを示す。

【図 11】[0030]プロセスに対してリソースを予測的に割り当てるために、コンピューティングデバイスのリソースに関するユーザーの監視されるリソース使用パターン及びそれから導き出される情報を使用するための方法のフローチャートを示す。

【図 12】[0031]コンピューティングデバイス上で実行されるアプリケーションが、アプリケーションに関連付けられるフォアグラウンドプロセス及び 1 つ又は複数のバックグラウンドタスクに対してコンピューティングデバイスのリソースの割り当てを管理するのを助けることを可能にするための、1 つの方法のフローチャートを示す。

【図 13】[0032]コンピューティングデバイス上で実行されるアプリケーションが、アプリケーションに関連付けられるフォアグラウンドプロセス及び 1 つ又は複数のバックグラウンドタスクに対してコンピューティングデバイスのリソースの割り当てを管理するのを助けることを可能にするための、別の方法のフローチャートを示す。

【図 14】[0033]様々な実施形態を実装するために使用することができる例示的なモバイルデバイスのブロック図である。

【図 15】[0034]様々な実施形態を実施するために使用することができる例示的なプロセッサベースのコンピューターシステムのブロック図である。

#### 【0021】

[0035]本発明の特徴及び利点は、同様の参照文字が全体を通して対応する要素を特定する図面と併せて考慮される場合、以下に記載する詳細な説明からより明らかになるであろう。図面において、同様の参照番号は、一般に、同一の、機能的に類似の、及び/又は構造的に類似の要素を示す。要素が最初に現れる図面は、対応する参照番号の左端の数字によって示される。

#### 【発明を実施するための形態】

#### 【0022】

##### I. はじめに

[0036]以下の詳細な説明は、本発明の例示的な実施形態を示す添付の図面を参照する。しかしながら、本発明の範囲はこれらの実施例に限定されるものではなく、代わりに添付の特許請求の範囲によって規定される。したがって、示される実施形態の修正版などの、添付の図面に示されたもの以外の実施形態が、本発明に包含されてもよい。

#### 【0023】

[0037]明細書中での「一実施形態」、「実施形態」、「例示的な実施形態」などへの参照は、説明される実施形態が特定の特徴、構造又は特性を含み得るが、すべての実施形態

10

20

30

40

50



が必ずしも当該特定の特徴、構造又は特性を含まなくてもよいことを示す。また、そのような語句は、必ずしも同じ実施形態を指しているわけではない。さらに、特定の特徴、構造又は特性が実施形態に関連して説明される場合、明示的に説明されていようとなかろうと、他の実施形態に関連してそのような特徴、構造又は特性を実施することが当業者の知識の範囲内にあると考えられる。

【 0 0 2 4 】

[0038] モバイルコンピューティングデバイスなどのコンピューティングデバイス上で実行されるアプリケーションによる、揮発性及び不揮発性のメモリー、CPU、I/O、ネットワーク、電源及びセンサーなどの様々なリソースの使用の追跡を可能にする、システム、方法、装置及びコンピュータープログラム製品が本願で説明される。本願で説明される技術は、アプリケーションに関連付けられるフォアグラウンドプロセス（例えば、アプリケーションに対するユーザーインターフェイスを提供し、コンピューティングデバイスに関連付けられる表示を排他的に占有することができるプロセス）によって使用されるリソース、並びに、コンピューティングデバイス上で別のプロセスとして実行されている、アプリケーションに関連付けられる１つ又は複数のバックグラウンドタスクによって使用されるリソースの両方を追跡することによって、アプリケーションについてのリソース使用を有利に追跡する。アプリケーションによるリソース使用に関する履歴データは、アプリケーションのリソース使用量を予測するために利用され、その後格納される。アプリケーションのその後の起動は、予測されるリソース使用量が利用可能であるかどうかを条件とすることができる。予測されるリソース使用量が利用可能な場合、それをアプリケーションに割り当てることができ、アプリケーションを起動することができる。その後、アプリケーションが予測されるリソース使用量を超える場合、それと同時に実行されている他のプロセスが、リソースを解放するために終了されてもよい。アプリケーションが実行する方法にかかわらず、アプリケーションについて新たなリソース使用データが取得され、その後、予測されるリソース使用量を調整するために使用される。このように、アプリケーションのためのリソースの割り当ては、ユーザーの使用パターンに基づいて時間とともにユーザーに対して自動的に微調整することができ、コンピューティングデバイスは、特にマルチタスクシナリオ中、よりよいユーザーエクスペリエンス及び改善されたパフォーマンスを提供することが可能となる。複数のユーザーをサポートするコンピューティングデバイスのために、予測されるリソース使用量は、各ユーザー／アプリケーションの組み合わせについて格納することができる。

【 0 0 2 5 】

[0039] 本願に記載のシステム、方法、装置及びコンピュータープログラム製品はまた、特定のリソース使用行動に関連付けられる時刻、位置及びイベントなどの、アプリケーション及び／又はユーザーによるリソース使用に関連する他の情報を追跡することが可能である。そのような情報は、ユーザーがアプリケーションを起動しようとする前であっても、コンピューティングデバイスのリソースをアプリケーションに予測的に割り当てるために有利に使用することができる。さらに、アプリケーション及び／又はアプリケーションデータは、ユーザーがそれらにアクセスすることが予期される前に、ある時点でメモリーへロードされてもよい。

【 0 0 2 6 】

[0040] セクションⅡは、実施形態による、デバイス固有又はユーザー固有のリソース使用プロファイルに基づいてリソース管理を実行する例示的なコンピューティングデバイス、及び関連付けられる方法について説明する。セクションⅢは、セクションⅡの例示的なコンピューティングデバイスがマルチプロセスアプリケーションのための動的なリソース管理を可能にする方法とすることができる方法を説明する。セクションⅣは、本願に記載のリソース管理機能を実施することができる例示的なモバイルデバイスについて説明する。セクションⅤは、本願に記載のリソース管理機能を実施することができる例示的なデスクトップコンピューターについて説明する。セクションⅥは、いくつかの結びの言葉を提供する。

## 【 0 0 2 7 】

II. デバイス固有又はユーザー固有のリソース使用プロファイルに基づいてリソース管理を実行するための例示的なシステム及び方法

[0041] 図 1 は、実施形態による、デバイス固有又はユーザー固有のリソース使用プロファイルに基づいてリソース管理を実行するコンピューティングデバイス 100 のブロック図である。コンピューティングデバイス 100 は、一般に、ユーザーのためにアプリケーションを実行することが可能なプロセッサベースの電子デバイスを表すように意図される。一実施形態では、コンピューティングデバイス 100 は、携帯電話（例えば、スマートフォン）、ラップトップコンピューター、タブレットコンピューター又はネットブックなどのモバイルコンピューティングデバイスを含む。コンピューティングデバイス 100 は、考えられる限り、ウェアラブルコンピューター（例えば、ヘッドマウントコンピューター）、携帯メディアプレーヤー、パーソナルデジタルアシスタント、パーソナルナビゲーションアシスタント、ハンドヘルドゲームコンソール、又はユーザーのためにアプリケーションを実行することができる任意の他のモバイルデバイスなどの他の種類のモバイルコンピューティングデバイスを含んでもよい。コンピューティングデバイス 100 の機能を組み込むことができるモバイルデバイスの一例を、図 14 を参照して以下に説明する。別の実施形態では、コンピューティングデバイス 100 は、ユーザーのためにアプリケーションを実行することができる、デスクトップコンピューター又は他の非モバイルコンピューティングプラットフォームを含む。コンピューティングデバイス 100 の機能を組み込むことができる例示的なデスクトップコンピューターは、図 15 を参照して以下に説明される。

## 【 0 0 2 8 】

[0042] 図 1 に示すように、コンピューティングデバイス 100 は、アプリケーション 102、オペレーティングシステム 104 及びデータストア 106 を含む。1つの実施形態では、アプリケーション 102 及びオペレーティングシステム 104 の各々は、コンピューティングデバイス 100 のメモリーに記憶され、当該メモリーに通信可能に接続されたプロセッサ（例えば、ソフトウェア命令を実行するように動作可能なマイクロプロセッサ又は他の回路）によって当該メモリーから実行される、ソフトウェアコンポーネントを含む。このようなメモリー及びプロセッサは、コンピューティングデバイス 100 の一部を含むが、簡略化のためだけに、図 1 には示されていない。さらに、そのような実施形態によれば、アプリケーション 102 及びオペレーティングシステム 104 は、各々が、コンピューティングデバイス 100 の不揮発性メモリーに永続的に格納されてもよく、コンピューティングデバイス 100 の電力供給された動作中にプロセッサによって実行するために、コンピューティングデバイス 100 の揮発性メモリーに一時的に転送されてもよい。

## 【 0 0 2 9 】

[0043] アプリケーション 102 は、コンピューティングデバイスのユーザーのために機能を実行し及び/又は機能を提供するためにコンピューティングデバイス 100 上にインストールして実行することができる、多種多様なコンピュータープログラムのうちのいずれかを表すことを意図している。アプリケーション 102 は、例えば、そしてどんなものであれ限定なく、電話アプリケーション、電子メールアプリケーション、メッセージングアプリケーション、ウェブブラウジングアプリケーション、カレンダーアプリケーション、ユーティリティアプリケーション、ゲームアプリケーション、ソーシャルネットワーキングアプリケーション、音楽アプリケーション、生産性アプリケーション、ライフスタイルアプリケーション、リファレンスアプリケーション、旅行アプリケーション、スポーツアプリケーション、ナビゲーションアプリケーション、ヘルスケア・フィットネスアプリケーション、ニュースアプリケーション、写真アプリケーション、金融アプリケーション、ビジネスアプリケーション、教育アプリケーション、天気予報アプリケーション、電子書籍リーダーアプリケーション、医療アプリケーションなどを表してもよい。

## 【 0 0 3 0 】

[0044]オペレーティングシステム１０４は、コンピューティングデバイス１００上で実行するアプリケーションのために、リソースを集散的に管理し、共通のサービスを提供する、コンピュータープログラムのセットを含む。図１に示すように、オペレーティングシステム１０４は、ドライバ１１０、カーネル１２０及びシステムサービス１３０を含む。ドライバ１１０は、ＣＰＵ、システムメモリ及びＩ／Ｏデバイスなどのコンピューティングデバイス１００の様々なハードウェアリソースとオペレーティングシステム１０４が相互作用することを可能にするコンポーネントを含む。カーネル１２０は、ドライバ１１０を利用して、コンピューティングデバイス１００のそのようなハードウェアリソース及び他のリソースを管理し、他のプログラムがこれらのリソースを実行及び使用することを可能にする。カーネル１２０は、プロセスをＣＰＵに割り当てること、システムメモリを特定のプロセスに割り当てること、及びＩ／Ｏ要求を適切なデバイスに割り当てることなどの動作を行うことができる。システムサービス１３０は、カーネル１２０によって割り当てることができる様々なリソースについてのアプリケーションからの要求にサービスを提供するように動作するコンポーネントである。

10

#### 【００３１】

[0045]図１に示すように、カーネル１２０は、メモリーマネージャー１２２を含む。メモリーマネージャー１２２は、コンピューティングデバイス１００のシステムメモリが利用される方法を制御し、他のプロセスがシステムメモリを必要とするときに他のプロセスがシステムメモリにアクセスすることを可能にする動作を実行する。他の動作として、メモリーマネージャー１２２は、仮想アドレス方式を保持してもよく、物理的に利用可能であるよりも多くのシステムメモリをプロセスが使用できるように、ページングを実行してもよい。システムメモリを管理するために、メモリーマネージャー１２２は、ドライバ１１０の１つであるメモリードライバ１１２と相互作用する。

20

#### 【００３２】

[0046]さらに図１に示すように、システムサービス１３０は、リソースマネージャー１３２を含む。本願でより詳細に説明するように、リソースマネージャー１３２は、アプリケーションに割り当てべきリソースの量を決定するように動作するコンポーネントであり、そのような決定はアプリケーションについてのリソース使用プロファイルに基づいてもよく、また、利用可能である場合にリソースの量を割り当てるように動作するコンポーネントである。以下の説明においては、アプリケーションに割り当てられるリソースはシステムメモリであると仮定するが、しかしながら、本願に記載のリソース管理技術は、ＣＰＵ、システムメモリ以外の揮発性メモリ、不揮発性メモリ、ネットワークリソース、Ｉ／Ｏリソース、電源リソース（例えば、バッテリー電源又はいくつかの他の電源からの電力）及びセンサーリソース（例えば、カメラやマイク）を含むがこれらに限定されない、コンピューティングデバイス１００の任意の他のリソースにまで容易に拡張することができることを理解すべきである。

30

#### 【００３３】

[0047]データストア１０６は、コンピューティングデバイス１００の揮発性又は不揮発性メモリに格納されたデータの集合を含む。本願で説明するように、データストア１０６は、コンピューティングデバイス１００上にインストールして実行することができるアプリケーションについてのリソース使用プロファイルを記憶するために、リソースマネージャー１３２によって使用される。ある実施形態によれば、データストア１０６は、レジスタ又はデータベースを含んでもよい。

40

#### 【００３４】

[0048]（例えば、アプリケーションについてのユーザーの起動又はいくつかの他のイベントにตอบสนองして）アプリケーション１０２が起動されるとき、１つ又は複数のシステムサービス呼が、アプリケーション１０２のためにシステムサービス１３０に対して発呼されて、アプリケーション１０２を適切に実行するのに必要なリソースをアプリケーション１０２に割り当てることを要求する。当該呼にตอบสนองして、リソースマネージャー１３２は、必要なリソースが利用可能であるかどうかを判断し、そのようなリソースが利用可能であ

50

る場合には、それをアプリケーション 102 に割り当てるように動作する。

【0035】

[0049] システムメモリーの場合、リソースマネージャー 132 は、データストア 106 に記憶することができるアプリケーション 102 についてのリソース使用プロファイルにアクセスするのを試みることにより、アプリケーション 102 に割り当てるシステムメモリーの量を決定する。実施形態では、アプリケーション 102 についてのそのようなリソース使用プロファイルは、実行中にアプリケーション 102 が使用すると予測されるシステムメモリーの量を指定する。予測されたリソース使用量を含む。代替的に、アプリケーション 102 についてのリソース使用プロファイルは、そのような予測されるリソース使用量を導くことができるデータを含むことができる。さらなる実施形態では、データストア 106 内にアプリケーション 102 についてのリソース使用プロファイルが存在しない場合、デフォルトの予測されるリソース使用量がアプリケーション 102 に割り当てられてもよい。そのようなデフォルトの予測されるリソース使用量は、すべてのアプリケーションに対して同じであってもよく、又はアプリケーションの種類によって異なってもよい。

10

【0036】

[0050] リソースマネージャー 132 がアプリケーション 102 についての予測されるリソース使用量を決定すると、リソースマネージャー 132 は、次いで、予測されるリソース使用量が現在利用可能であるかどうかを判断するためにメモリーマネージャー 122 と通信する。予測されるリソース使用量が現在利用可能である場合、リソースマネージャー 132 は、予測されるリソース使用量をアプリケーション 102 に割り当て、アプリケーション 102 は実行することを許可される。しかし、予測されるリソース使用量が現在利用可能でない場合、リソースマネージャー 132 は、アプリケーション 102 が実行されることを防止することができる。さらに、予測されるリソース使用量が現在利用可能でない場合、リソースマネージャー 132 はまた、アプリケーション 102 に対してシステムメモリーを解放するために、コンピューティングデバイス上で現在実行されている 1 つ又は複数の他のプロセスを終了してもよい。そのような他のプロセスは、プロセス優先方式に従ってアプリケーション 102 よりも低い優先度が割り当てられたプロセスであってもよい。このようにして十分なシステムメモリーが取り戻されると、リソースマネージャー 132 は、アプリケーション 102 が実行されることを許可してもよい。

20

30

【0037】

[0051] アプリケーション 102 が実行を開始した後、リソースマネージャー 132 は、アプリケーション 102 が実際にどのくらいのシステムメモリーを使用しているかを監視する。このようにして、リソースマネージャー 132 は、予測されるリソース使用量に従って最初にアプリケーション 102 に割り当てられたものよりも多くのシステムメモリーをアプリケーション 102 が利用していると判定することができる。この場合、リソースマネージャー 132 は、アプリケーション 102 が適切に実行されるために十分なシステムメモリーが利用可能であることを確実にするために、コンピューティングデバイス 100 上で同時に実行されている 1 つ又は複数のプロセスを終了することを試みてもよい。

【0038】

40

[0052] リソースマネージャー 132 はまた、どのくらいのシステムメモリーがアプリケーション 102 によって使用されているかを監視することによって得た情報を利用して、アプリケーション 102 についてデータストア 106 内に記憶される予測されるリソース使用量を更新する。予測されるリソース使用量を更新するための様々なアルゴリズムが以下で詳細に説明される。このように、リソースマネージャー 132 は、アプリケーション 102 がどのように行動しており、コンピューティングデバイス 100 のユーザーによってどのように利用されているかに応じた方法で、時間とともに、アプリケーション 102 に対するリソースの割り当てを自動的に微調整することができる。このことは、コンピューティングデバイス 100 が、特にマルチタスクシナリオ中に、より良いユーザーエクスペリエンスと改善されたパフォーマンスを提供することを可能にするものであり、有益で

50

ある。

【 0 0 3 9 】

[0053]デバイス固有又はユーザー固有のリソース使用プロファイルに従ってリソースマネージャー 1 3 2 がコンピューティングデバイス 1 0 0 のリソースを管理するように動作する方法を、図 2 のフローチャート 2 0 0 を参照して説明する。フローチャート 2 0 0 のステップは図 1 のコンピューティングデバイス 1 0 0 のコンポーネントを引き続き参照して説明されるが、この方法がまったく別のコンポーネントによって実行されてもよいことが理解されるべきである。

【 0 0 4 0 】

[0054]図 2 に示すように、フローチャート 2 0 0 の方法は、ステップ 2 0 2 において開始し、リソースマネージャー 1 3 2 は、コンピューティングデバイス 1 0 0 によって実行されている間にアプリケーションのインスタンスによって使用されるリソースの第 1 の量を決定する。例えば、上述のように、リソースマネージャー 1 3 2 は、アプリケーション 1 0 2 の第 1 のインスタンスがコンピューティングデバイス 1 0 0 によって実行されている間、アプリケーション 1 0 2 の第 1 のインスタンスによって使用されているシステムメモリーの第 1 の量を決定してもよい。

【 0 0 4 1 】

[0055]実施形態では、リソースマネージャー 1 3 2 は、アプリケーションの第 1 のインスタンスに関連付けられたフォアグラウンドプロセス及びアプリケーションの第 1 のインスタンスに関連付けられた 1 つ又は複数のバックグラウンドタスクによって使用されるリソースの量を決定することによって、第 1 の量を決定する。フォアグラウンドプロセスは、現在、アプリケーションの第 1 のインスタンスのためのユーザーインターフェイスを提供し、コンピューティングデバイス 1 0 0 に関連付けられる表示の少なくとも一部を占有しているプロセスであってもよい。しかし、多くの現代のアプリケーションの実行はまた、1 つ又は複数のバックグラウンドプロセス（本願ではバックグラウンドタスクとも呼ぶ）の実行を伴う。このようなバックグラウンドタスクは、フォアグラウンドプロセスと同時に実行されることがあり、ある場合にはフォアグラウンドプロセスによって開始され得る。そのようなバックグラウンドタスクの例は、バックグラウンド音声の生成、ボイスオーバーインターネットプロトコル（V o I P）呼の実行、電子メールの同期、コンテンツ共有などを含むが、決してこれらに限定されるものではない。フォアグラウンドプロセスと同様に、これらのバックグラウンドタスクはシステムリソースを消費する。しかし、アプリケーションのリソース使用についての従来の概念は、通常、そのようなバックグラウンドタスクによって消費されるリソースを包含していない。実施形態は、実行中のアプリケーションに関連付けられるフォアグラウンドプロセス及び同じアプリケーションに関連付けられる 1 つ又は複数のバックグラウンドタスクの両方によるリソース使用を追跡し、両方の種類のリソース使用を組み合わせるアプリケーションについて総リソース使用量を取得することによって、このことに対処するものであり有利である。

【 0 0 4 2 】

[0056]ステップ 2 0 4 において、リソースマネージャー 1 3 2 は、第 1 の量に少なくとも基づいて、アプリケーションについての第 1 の予測されるリソース使用量を計算する。例えば、リソースマネージャー 1 3 2 は、第 1 の量に少なくとも基づいて、アプリケーション 1 0 2 についての第 1 の予測されるシステムメモリー使用量を計算してもよい。このステップを実行することができる方法を示すために、様々な例のアルゴリズムが以下で提供される。

【 0 0 4 3 】

[0057]ステップ 2 0 6 において、リソースマネージャー 1 3 2 は、データストア 1 0 6 内に第 1 の予測されるリソース使用量を記憶する。例えば、リソースマネージャー 1 3 2 は、データストア 1 0 6 内に、アプリケーション 1 0 2 についての第 1 の予測されるシステムメモリー使用量を記憶することができる。

【 0 0 4 4 】

[0058]ステップ206の実行後のある時点において、リソースマネージャー132は、アプリケーションの第2のインスタンスがコンピューティングデバイス100上で実行されるべきであることを通知される。ステップ208において、リソースマネージャー132は、データストア106に格納された第1の予測されるリソース使用量に少なくとも基づいて、アプリケーションの第2のインスタンスに割り当てられるべきリソースの第2の量を決定する。例えば、リソースマネージャー132は、データストア106内のアプリケーション102について記憶された第1の予測されるシステムメモリー使用量に少なくとも基づいて、アプリケーション102の第2のインスタンスに割り当てられるべきシステムメモリーの第2の量を決定してもよい。システムメモリーの第2の量を決定することは、例えば、システムメモリーの第2の量を第1の予測されるシステムメモリー使用量に設定することを含んでもよい。

10

【0045】

[0059]ステップ204においてアプリケーションについての第1の予測されるリソース使用量を計算するために、様々な方法を使用することができる。例えば、一実施形態では、第1の予測されるリソース使用量は、ある数の実行にわたってアプリケーションによって使用された最大のリソース量に基づいて計算される。この手法は、図3のフローチャート300によって表される。

【0046】

[0060]具体的には、図3は、実施形態による、アプリケーションについての予測されるリソース使用量を計算する方法のフローチャート300を示す。図3に示すように、フローチャート300の方法は、ステップ302で開始し、リソースマネージャー132は、ステップ202中に決定された第1の量を、アプリケーションの1つ又は複数の以前に実行されたインスタンスに関連付けられる1つ又は複数の以前に決定された量と比較することによって、アプリケーションについての最大のリソース使用量を取得する。ステップ304において、リソースマネージャー132は、最大のリソース使用量に基づいて、第1の予測されるリソース使用量を計算する。

20

【0047】

[0061]この手法の例がここで提供される。この例によれば、リソースマネージャー132は、5回の異なる実行におけるアプリケーション102によるシステムメモリー使用量を追跡し、そのような実行にわたるアプリケーション102による最大のシステムメモリー使用量を、予測されるリソース使用量として選択する。例えば、アプリケーション102の最後の4回の実行中に、アプリケーション102が、それぞれ100MB、120MB、160MB及び140MBのシステムメモリーを使用したと仮定する。次いで、ステップ202の間に、リソースマネージャー132は、アプリケーション102の現在実行中のインスタンスがシステムメモリーのうち150MBを使用していることを決定する。この場合、リソースマネージャー132は、第1の予測されるリソース使用量を160MB（又は、場合によっては160MBに加えてあるバッファ量）に設定してもよい。というのは、それが、最後の5回の実行にわたるアプリケーション102による最大のシステムメモリーの使用であるからである。

30

【0048】

[0062]別の実施形態では、第1の予測されるリソース使用量は、ある数の実行にわたるアプリケーションによって使用される平均リソース量に基づいて計算される。この手法は図4のフローチャート400によって表される。

40

【0049】

[0063]具体的には、図4は、実施形態による、アプリケーションについての予測されるリソース使用量を計算する方法のフローチャート400を示す。図4に示すように、フローチャート400の方法は、ステップ402で開始し、リソースマネージャー132は、ステップ202中に決定された第1の量及びアプリケーションの1つ又は複数の以前に実行されたインスタンスに関連付けられる1つ又は複数の以前に決定された量に基づいて、アプリケーションについて平均リソース使用量を取得する。ステップ404において、リ

50

ソースマネージャー 132 は、平均リソース使用量に基づいて、第 1 の予測されるリソース使用量を計算する。

【0050】

[0064]この手法の例がここで提供される。この例によれば、リソースマネージャー 132 は、5 回の別個の実行におけるアプリケーション 102 によるシステムメモリーの使用を追跡し、それらの値の平均値を計算し、その平均値を予測されるリソース使用量として使用する。例えば、アプリケーション 102 の最後の 4 回の実行中に、アプリケーション 102 が、それぞれシステムメモリーのうちの 100 MB、120 MB、160 MB 及び 140 MB を使用したと仮定する。次いで、ステップ 202 中に、リソースマネージャー 132 は、アプリケーション 102 の現在実行中のインスタンスがシステムメモリーのうち 150 MB を使用していることを決定する。この場合、リソースマネージャー 132 は、アプリケーション 102 についての平均リソース使用量が  $(100 \text{ MB} + 120 \text{ MB} + 160 \text{ MB} + 140 \text{ MB} + 150 \text{ MB}) / 5$  であり、134 MB に等しいと決定する。その結果、リソースマネージャー 132 は、第 1 の予測されるリソース使用量を、134 MB (又は、場合によっては、134 MB に加えてあるバッファ量) に設定してもよい。

【0051】

[0065]ステップ 402 で説明した平均リソース使用量を得ることができる別の方法をここで説明する。この手法によれば、リソース使用量の移動平均が、最新のリソース使用量と組み合わせられて、新たな長期の移動平均が得られる。この組み合わせは、より大きな重みが古い長期間にわたる移動平均又は最新のリソース使用量のいずれかに適用されるように重み付けすることができる。あるいは、両方の量が等しく重み付けされてもよい。この手法は、以下の式を用いて表すことができる。

【0052】

$$R_{AVG\_NEW} = R_{AVG\_OLD} + (1 - \alpha) R_{NEW}$$

$R_{AVG\_NEW}$  はリソース使用量の新たな移動平均を表し、 $R_{AVG\_OLD}$  はリソース使用量の古い移動平均を表し、 $R_{NEW}$  は最も最近観察されたリソース使用量を表し、 $\alpha$  は重み付けを決定する変数である。

【0053】

[0066]例えば、リソース使用量の古い移動平均  $R_{AVG\_OLD}$  が 130 MB に等しく、最も最近観察されたリソース使用量  $R_{NEW}$  が 160 MB に等しく、重み付け係数  $\alpha$  が 0.25 であると仮定する。この場合、リソース使用量の新たな移動平均  $R_{AVG\_NEW}$  は、 $(0.25 \times 130 \text{ MB}) + (0.75 \times 160 \text{ MB})$  であり、152.5 MB に等しくなる。その結果、リソースマネージャー 132 は、第 1 の予測されるリソース使用量を、152.5 MB (又は、場合によっては、152.5 MB とあるバッファ量の合計) に設定することができる。

【0054】

[0067]移動平均を使用してアプリケーションについての平均リソース使用量を得る上述の方法は、図 5 のフローチャート 500 に示される。図 5 に示すように、フローチャート 500 の方法は、ステップ 502 で開始し、リソースマネージャー 132 は、ステップ 202 中に得られた第 1 の量 (すなわち、最も最近に観察されたリソース使用量) を第 1 の重み付け係数で乗算する。ステップ 504 中、リソースマネージャー 132 は、1 つ又は複数の以前に決定された量 (すなわち、以前に観察されたリソース使用量) に少なくとも基づいて得られた移動平均リソース使用量を第 2 の重み付け係数で乗算して、第 2 の積を得る。ステップ 506 において、リソースマネージャー 132 は、第 1 の積と第 2 の積を合計して、新たな平均リソース使用量を得る。

【0055】

[0068]図 2 を参照して上述したように、フローチャート 200 のステップ 208 において、リソースマネージャー 132 は、第 1 の予測されるリソース使用量に少なくとも基づいて、コンピューティングデバイス 100 によって実行されるアプリケーションの第 2 の

インスタンスに割り当てられるべきリソース（例えば、システムメモリー）の第２の量を決定する。当該第２の量は予測アルゴリズムに基づく推定値であるので、アプリケーションの第２のインスタンスが割り当てられた第２の量より多くを消費する可能性が常にある。このシナリオに対処するために、リソースマネージャー１３２は、アプリケーションの第２のインスタンスがその割り当てられたリソース割り当てよりも多くを消費していることを判断するように構成されてもよく、これに応答して、リソースを解放するために、コンピューティングデバイス１００上で同時に実行されている他のプロセスを終了することができる。例えば、リソースマネージャー１３２は、プロセス優先方式に従って当該アプリケーションより低い優先度を割り当てられた特定のプロセスを終了させることができる。

10

#### 【００５６】

[0069]この処理は、図６のフローチャート６００に示されている。具体的には、図６に示すように、フローチャート６００の方法は、ステップ６０２で開始し、リソースマネージャー１３２は、コンピューティングデバイス１００によって実行されている間にアプリケーションの第２のインスタンスによって使用されるリソースの量が第２の量を超えていると判定する。この判定に응答して、ステップ６０４において、リソースマネージャー１３２は、コンピューティングデバイス上で同時に実行されている１つ又は複数のプロセスを終了する。追加的又は代替的に、リソースマネージャー１３２は、コンピューティングデバイス１００上で同時に実行されている１つ又は複数のプロセスと通信し、それらのプロセスに、リソースの消費を低減するように指示してもよい。

20

#### 【００５７】

[0070]リソースマネージャー１３２が、メモリーマネージャー１２２との通信を介して又はいくつかの他の手段を介して、アプリケーションの第２のインスタンスに割り当てべきリソースの第２の量が現在利用できないと決定することもあり得る。この場合、リソースマネージャー１３２は、アプリケーションの第２のインスタンスの実行を防止することができる。

#### 【００５８】

[0071]この処理は、図７のフローチャート７００に示されている。具体的には、図７に示すように、フローチャート７００の方法は、ステップ７０２で開始し、リソースマネージャー１３２は、リソースの第２の量が現在利用可能でないと判定する。この判定に응答して、ステップ７０４において、リソースマネージャー１３２は、アプリケーションの第２のインスタンスの実行を防止する。

30

#### 【００５９】

[0072]リソースマネージャー１３２が、アプリケーションの第２のインスタンスに割り当てべきリソースの第２の量が現在利用可能でないと判断する場合、リソースマネージャー１３２はまた、リソースを解放するために、コンピューティングデバイス１００上で実行されている１つ又は複数のプロセスを終了してもよい。例えば、リソースマネージャー１３２は、プロセスの優先方式に従って当該アプリケーションより低い優先度を割り当てられた特定のプロセスを終了させることができる。このような他のプロセスによるリソース消費が低減された後、リソースマネージャー１３２は、アプリケーションの第２のインスタンスを実行することを可能にする。

40

#### 【００６０】

[0073]この処理は、図８のフローチャート８００に示されている。具体的には、図８に示すように、フローチャート８００の方法は、ステップ８０２で開始し、リソースマネージャー１３２は、リソースの第２の量が現在利用可能でないと判定する。この判定に응答して、ステップ８０４において、リソースマネージャー１３２は、コンピューティングデバイス１００上で現在実行されている１つ又は複数のプロセスを終了する。追加的に又は代替的に、リソースマネージャー１３２は、コンピューティングデバイス１００上で現在実行されている１つ又は複数のプロセスと通信し、それらのプロセスにリソースの消費を減らすよう指示してもよい。

50



## 【 0 0 6 1 】

[0074]スマートフォンなどの多くのコンピューティングデバイスにとって、典型的な使用モデルは、コンピューティングデバイスごとに1ユーザーである。この場合、コンピューティングデバイスについてのリソース使用プロファイルを記憶することは、本質的に、特定のユーザーのリソース使用プロファイルを記憶することと同じである。しかし、コンピューティングデバイスはまた、複数のユーザーをサポートするために使用することができる。例えば、同じ世帯のある数の異なるメンバーが、それぞれ、同じタブレットコンピューター、デスクトップコンピューター又はゲームコンソールを使用して、アプリケーションを実行することがある。これらのユーザーの各々は、異なるアプリケーションを使用することがあり、また、同じアプリケーションを異なる方法で使用することがあり、これはリソース消費のレベルが変化することにつながる。このシナリオに対処するために、特定の実施形態は、アプリケーションとユーザーの両方によるコンピューティングデバイスに対するリソース使用プロファイルを記憶し、その結果、リソースの予測割り当ては、コンピューティングデバイスを現在使用している特定のユーザーに合わせて調整することができる。これらのリソース使用プロファイルは、(前述のデバイス固有のリソース使用プロファイルとは対照的に)ユーザー固有のリソース使用プロファイルと考えることができる。

10

## 【 0 0 6 2 】

[0075]図9は、実施形態による、ユーザー固有のリソース使用プロファイルに従ってリソースを管理するためにコンピューティングデバイスによって実施される方法のフローチャート900を示す。フローチャート900の方法は、ここで、図1を参照して上述したコンピューティングデバイス100のコンポーネントを引き続き参照して説明される。しかしながら、この方法はその実施形態に限定されるものではない。

20

## 【 0 0 6 3 】

[0076]図9に示すように、フローチャート900の方法は、ステップ902で開始し、リソースマネージャー132は、第1のユーザーのためにコンピューティングデバイス100によって実行される間、アプリケーションの第1のインスタンスによって使用されるリソースの第1の量を決定する。例えば、リソースマネージャー132は、アプリケーション102の第1のインスタンスが第1のユーザーのためにコンピューティングデバイス100によって実行されている間、アプリケーション102の第1のインスタンスによって使用されているシステムメモリーの第1の量を決定することができる。アプリケーション102の第1のインスタンスが第1のユーザーのために実行されているという事実は、様々な方法のいずれかで決定することができる。ユーザーがコンピューティングデバイス100にログインすることができる実施形態では、現在のログイン情報が当該決定をするために使用されてもよい。しかし、この例は、限定することを意図するものではなく、コンピューティングデバイスのユーザーの同一性を決定するための任意の適切な方法を使用することができる。

30

## 【 0 0 6 4 】

[0077]ステップ904において、リソースマネージャー132は、第1の量に少なくとも基づいて、アプリケーション及び第1のユーザーについての第1の予測されるリソース使用量を計算する。例えば、リソースマネージャー132は、第1の量に少なくとも基づいて、アプリケーション102及び第1のユーザーについての第1の予測されるシステムメモリー使用量を計算してもよい。このステップを実行することができる方法を示すために、様々な例示的なアルゴリズムが既に提供された。

40

## 【 0 0 6 5 】

[0078]ステップ906において、リソースマネージャー132は、データストア106内に第1の予測されるリソース使用量を記憶する。例えば、リソースマネージャー132は、データストア106内に、アプリケーション102及び第1のユーザーについての第1の予測されるシステムメモリー使用量を記憶することができる。

## 【 0 0 6 6 】

50

[0079]ステップ906の実行後のある時点において、リソースマネージャー132は、アプリケーションの第2のインスタンスが第1のユーザーのためにコンピューティングデバイス100上で実行されることを通知される。ステップ908において、リソースマネージャー132は、データストア106内に記憶された第1の予測されるリソース使用量に少なくとも基づいて、アプリケーションの第2のインスタンスに割り当てられるべきリソースの第2の量を決定することができる。例えば、リソースマネージャー132は、データストア106内にアプリケーション102及び第1のユーザーについて記憶された第1の予測されるシステムメモリー使用量に少なくとも基づいて、アプリケーション102の第2のインスタンスに割り当てられるべきシステムメモリーの第2の量を決定してもよい。システムメモリーの第2の量を決定することは、例えば、システムメモリーの第2の量を第1の予測されるシステムメモリー使用量に設定することを含んでもよい。

10

#### 【0067】

[0080]ステップ910において、リソースマネージャー132は、第1のユーザーとは異なる第2のユーザーのためにコンピューティングデバイス100によって実行されている間、アプリケーションの第3のインスタンスによって使用されるリソースの第3の量を決定する。例えば、リソースマネージャー132は、アプリケーション102の第3のインスタンスが第2のユーザーのためにコンピューティングデバイス100によって実行されている間、アプリケーション102の第3のインスタンスによって使用されているシステムメモリーの第3の量を決定することができる。アプリケーション102の第3のインスタンスが第2のユーザーのために実行されているという事実は、様々な方法のいずれかで決定することができる。ユーザーがコンピューティングデバイス100にログインすることができる実施形態では、現在のログイン情報が当該決定をするために使用されてもよい。しかし、この例は、限定することを意図するものではなく、コンピューティングデバイスのユーザーの同一性を決定するための任意の適切な方法を使用してもよい。

20

#### 【0068】

[0081]ステップ912において、リソースマネージャー132は、第3の量に少なくとも基づいて、アプリケーション及び第2のユーザーについての第2の予測されるリソース使用量を計算する。例えば、リソースマネージャー132は、第3の量に少なくとも基づいて、アプリケーション102及び第2のユーザーについての第2の予測されるシステムメモリー使用量を計算してもよい。このステップを実行することができる方法を示すために、様々な例示的なアルゴリズムが既に提供された。

30

#### 【0069】

[0082]ステップ914において、リソースマネージャー132は、データストア106内に第2の予測されるリソース使用量を記憶する。例えば、リソースマネージャー132は、データストア106内に、アプリケーション102及び第2のユーザーについての第2の予測されるシステムメモリー使用量を格納することができる。

#### 【0070】

[0083]ステップ914の実行後のある時点において、リソースマネージャー132は、アプリケーションの第4のインスタンスが第2のユーザーのためにコンピューティングデバイス100上で実行されることを通知される。ステップ916において、リソースマネージャー132は、データストア106に格納された第2の予測されるリソース使用量に少なくとも基づいて、アプリケーションの第4のインスタンスに割り当てられるべきリソースの第4の量を決定する。例えば、リソースマネージャー132は、データストア106内のアプリケーション102及び第2のユーザーについて記憶された第2の予測されるシステムメモリー使用量に少なくとも基づいて、アプリケーション102の第4のインスタンスに割り当てられるべきシステムメモリーの第4の量を決定することができる。システムメモリーの第4の量を決定することは、例えば、システムメモリーの第4の量を第2の予測されるシステムメモリー使用量に設定することを含んでもよい。

40

#### 【0071】

[0084]図10は、実施形態による、デバイス固有又はユーザー固有のリソース使用プロ

50

ファイルに従ってアプリケーションの起動を管理するための方法のフローチャート 1000 を示している。フローチャート 1000 の方法は、ここで、図 1 を参照して上述したコンピューティングデバイス 100 のコンポーネントを引き続き参照して説明される。しかし、この方法はその実施形態に限定されるものではない。

【0072】

[0085] 図 10 に示すように、フローチャート 1000 の方法は、ステップ 1002 で開始し、アプリケーションを起動する要求が受け取られる。例えば、アプリケーション 102 を起動する要求は、例えば、システムサービス 130 によって及び/又はリソースマネージャー 132 によって受け取られてもよい。

【0073】

[0086] ステップ 1004 において、リソースマネージャー 132 は、アプリケーションについての予測されるリソース使用量を取得する。例えば、リソースマネージャー 132 は、アプリケーション 102 についての予測されるリソース使用量を取得してもよい。予測されるリソース使用量は、コンピューティングデバイス 100 によるアプリケーションの 1 つ又は複数の以前の実行中のコンピューティングデバイス 100 のリソースの監視された使用に基づく。予測されるリソース使用量、又はそのような予測されるリソース使用量を計算するために必要な情報は、アプリケーションについてのリソース使用プロファイルから得ることができ、データストア 106 内に記憶されてもよい。リソースは、揮発性又は不揮発性のメモリーリソース、CPU リソース、I/O リソース、ネットワークリソース、電源リソース及びセンサーリソースのうちの 1 つ又は複数を含んでもよい。予測されるリソース使用量は、アプリケーションに関連付けられるフォアグラウンドプロセス及びアプリケーションに関連付けられる 1 つ又は複数のバックグラウンドタスクの両方によるコンピューティングデバイス 100 のリソースの使用を監視することに基づいてもよい。

【0074】

[0087] 判定ステップ 1006 において、リソースマネージャー 132 は、予測されるリソース使用量を現在利用可能なリソースの量と比較することによって、十分な量のリソースが利用可能であるか否かを判定する。例えば、リソースがシステムメモリーである実施形態において、リソースマネージャー 132 は、メモリーマネージャー 122 から現在利用可能なシステムメモリーの量を得ることができ、次いで、アプリケーションについての予測されるシステムメモリー使用量を現在利用可能なシステムメモリーと比較して、十分な量のシステムメモリーが利用可能であるかどうかを判定することができる。

【0075】

[0088] 十分な量のリソースが利用可能であるとリソースマネージャー 132 が少なくとも判定した場合、次いで、リソースマネージャー 132 は、ステップ 1008 に示すように、アプリケーションが起動することを許可する。例えば、十分なシステムメモリーが利用可能であるとリソースマネージャー 132 が少なくとも判定した場合、次いで、リソースマネージャー 132 は、アプリケーション 102 が起動することを許可する。アプリケーションが起動することをリソースマネージャー 132 が許可するために満たされなければならない他の条件があり得ることに留意されたい。

【0076】

[0089] 十分な量のリソースが利用可能でないとリソースマネージャー 132 が判定する場合、ステップ 1010 に示すように、リソースマネージャー 132 は、アプリケーションが起動することを許可しない。例えば、十分なシステムメモリーが利用可能でないとリソースマネージャー 132 が判定する場合、リソースマネージャー 132 は、アプリケーション 102 が起動することを許可しない。加えて、リソースマネージャー 132 は、1 つ又は複数の現在実行されているプロセスを終了させるか、そうでなければ、それらのリソース使用量を低減させることによって、リソースを解放しようとしてもよい。

【0077】

[0090] 実施形態では、リソースマネージャー 132 はまた、特定のリソース使用行動に

10

20

30

40

50

関連付けられる時刻、場所及びイベントなどの、アプリケーション及び／又はユーザーによるリソース使用に関連する他の情報を追跡することが可能である。そのような情報は、ユーザーがアプリケーションを起動することを試みる前でさえ、コンピューティングデバイスのリソースをアプリケーションに予測的に割り当てるために都合よく使用することができる。さらに、アプリケーション及び／又はアプリケーションデータが、ユーザーがそれらにアクセスすると予期される前の時点でメモリーにロードされてもよい。

【 0 0 7 8 】

[0091]例えば、ユーザーが特定の時間において、特定の場所において又は特定のイベントに関連して、特定のアプリケーションを起動する可能性があると判断できる場合、リソースマネージャー 132 は、ユーザーがアプリケーションを起動すると予期される前、場合によってはその直前に、十分なリソースがアプリケーションに割り当てるために利用可能であることを確実にするように動作することができる。さらに、このような予測的な分析は、実際に、ユーザーがアプリケーション及びその機能への即時のアクセスを得ることができるように、アプリケーション又はアプリケーションのコンテンツをシステムメモリーにロードするために使用することができる。

10

【 0 0 7 9 】

[0092]このことを説明する助けとなるべく、図 11 は、コンピューティングデバイスのリソースに関するユーザーについての監視されるリソース使用パターン及びそれから導かれる情報を使用して、プロセスに対してリソースを予測的に割り当てるための方法のフローチャート 1100 を示す。フローチャート 1100 の方法は、ここでは、図 1 を参照して上述したコンピューティングデバイス 100 のコンポーネントを引き続き参照して説明される。しかし、この方法はその実施形態に限定されるものではない。

20

【 0 0 8 0 】

[0093]図 11 に示すように、フローチャート 1100 の方法は、ステップ 1102 で開始し、リソースマネージャー 132 は、アプリケーション及び／又はユーザーによるコンピューティングデバイス 100 の少なくとも 1 つのリソースの使用パターンを監視する。前述のように、使用パターンを監視することは、アプリケーション及び／又はユーザーによるリソース使用を監視すること、及び、特定のリソース使用行動に関連付けられる時間、場所及びイベントなどのそのようなリソース使用に関連付けられる他の情報を監視することを伴ってもよい。

30

【 0 0 8 1 】

[0094]ステップ 1104 において、リソースマネージャー 132 は、使用のパターンに関連付けられる情報を格納する。リソースマネージャー 132 は、例えばデータストア 106 内にそのような情報を格納することができる。

【 0 0 8 2 】

[0095]ステップ 1106 において、リソースマネージャー 132 は、コンピューティングデバイス 100 による実行の前に、少なくとも 1 つのリソースの少なくとも一部を少なくとも 1 つのプロセスに予測的に割り当てるために、格納された情報を使用する。一実施形態では、ステップ 1106 の間、リソースマネージャー 132 は、コンピューティングデバイス 100 のユーザー又は任意の他のエンティティがアプリケーションを起動する前に、リソースの少なくとも一部をアプリケーションに予測的に割り当てる。少なくとも 1 つのリソースがシステムメモリーを含む別の例示的な実施形態では、リソースマネージャー 132 は、コンピューティングデバイス 100 のユーザー又は任意の他のエンティティがアプリケーションを起動する前に、アプリケーション又は当該アプリケーションに関連付けられるデータの少なくとも一部を以前に割り当てられたシステムメモリーにロードしてもよい。これらの例は、例示のみのために提供されるものであり、限定することを意図するものではない。関連技術の当業者であれば、追加のリソース管理タスクを実行するため、リソース使用パターン情報を利用するために、さらに他の技術が使用され得ることを理解するであろう。

40

【 0 0 8 3 】

50

### III. マルチプロセスアプリケーションの動的リソース管理のための例示的なシステム及び方法

[0096]実施形態において、図1のコンピューティングデバイス100は、コンピューティングデバイス100のリソースの割り当てられた部分がアプリケーションに関連付けられる複数のプロセス間でどのように分割されるべきかを、コンピューティングデバイス100上で実行されるアプリケーション（例えば、アプリケーション102）が動的に制御することを可能にする。例えば、コンピューティングデバイス100上で実行されるアプリケーション102は、リソースマネージャー132とインタラクトして、コンピューティングデバイス100のリソースの割り当てられた部分が、アプリケーションに関連付けられるフォアグラウンドプロセス及びアプリケーションに関連付けられる1つ又は複数のバックグラウンドプロセスの間でどのように分割されるべきかを動的に制御してもよい。この機能を有するアプリケーションを提供することにより、本願に記載の実施形態は、アプリケーションが、変化するリソース割り当てに適応し、コンピューティングデバイス上で実行されていてもよい他のプロセスとより良好に共存し、それによって様々なマルチタスクシナリオをより良好に可能にすることを都合よく可能にする。

10

#### 【0084】

[0097]アプリケーション102及びリソースマネージャー132がこのような動的リソース管理を行うために相互作用することができる1つの方法が、ここで図12のフローチャート1200を参照して説明される。図12に示すように、フローチャート1200の方法は、ステップ1202で開始し、リソースマネージャー1202は、アプリケーション102に関連付けられるフォアグラウンドプロセスに対して、当該アプリケーションがフォアグラウンドで実行されているという通知を送る。

20

#### 【0085】

[0098]ステップ1204において、リソースマネージャー132は、フォアグラウンドプロセスから、フォアグラウンドプロセスに割り当てられるべき第1のリソース使用予算（バジェット）の第1の部分及びアプリケーションに関連付けられる少なくとも1つのバックグラウンドタスクに割り当てられるべき第1のリソース使用予算の第2の部分を指定する少なくとも1つのメッセージを受け取る。

#### 【0086】

[0099]ステップ1206において、リソースマネージャー132は、第1のリソース使用予算の第1の部分に等しいリソースの一部をフォアグラウンドプロセスに割り当て、第1のリソース使用予算の第2の部分に等しいリソースの一部を少なくとも1つのバックグラウンドタスクに割り当てる。リソースは、例えば、限定なしに、揮発性又は不揮発性メモリ、CPU、I/Oリソース、ネットワークリソース、電源リソース、センサーリソースを含むことができる。

30

#### 【0087】

[0100]ステップ1208において、リソースマネージャー132は、フォアグラウンドプロセスに、アプリケーションが中断又は終了されるという通知を送る。

[0101]ステップ1210において、リソースマネージャー132は、フォアグラウンドプロセスから、フォアグラウンドプロセスに割り当てられるべき第2のリソース使用予算の第1の部分及びアプリケーションに関連付けられる少なくとも1つのバックグラウンドタスクに割り当てられるべき第2のリソース使用予算の第2の部分を指定する少なくとも1つのメッセージを受け取る。第2のリソース使用予算は第1のリソース使用予算より小さくてもよい。

40

#### 【0088】

[0102]ステップ1212において、リソースマネージャー132は、第2のリソース使用予算の第1の部分に等しいリソースの一部をフォアグラウンドプロセスに割り当て、第2のリソース使用予算の第2の部分に等しいリソースの一部を少なくとも1つのバックグラウンドタスクに割り当てる。所定期間後、リソースマネージャー132は、フォアグラウンドプロセスによって使用されているリソースの量及び少なくとも1つのバックグラウ

50

ンドプロセスによって使用されているリソースの量の合計が第2のリソース使用予算を超えているかどうかを判定してもよい。合計が第2のリソース使用予算を超えていると判定することに応答して、リソースマネージャー132は、フォアグラウンドプロセス及び少なくとも1つのバックグラウンドプロセスのうちの1つ又は複数を終了させてもよい。

【0089】

[0103]一実施形態では、前述の方法でリソースマネージャー132とフォアグラウンドプロセスとの間でやり取りされる通信は、共有APIを介して交換される。そのようなAPIはアプリケーションの開発者に公開されてもよく、このため、そのような開発者は、これらの機能を利用することができるアプリケーションを開発することができる。

【0090】

10

[0104]図13は、フォアグラウンドプロセス及びアプリケーションに関連付けられる1つ又は複数のバックグラウンドタスクに対するコンピューティングデバイスのリソースの割り当てを管理する際にコンピューティングデバイス上で実行されるアプリケーションが支援することを可能にする、別の方法のフローチャート1300を示している。

【0091】

[0105]図13に示すように、フローチャート1300は、ステップ1302で開始し、リソースマネージャー132は、リソース使用予算を示す少なくとも1つのメッセージを、アプリケーションに関連付けられるフォアグラウンドプロセスに送る。

【0092】

[0106]ステップ1304において、リソースマネージャー132は、フォアグラウンドプロセスから、フォアグラウンドプロセスに割り当てられるべきリソース使用予算の第1の部分及びアプリケーションに関連付けられる少なくとも1つのバックグラウンドタスクに割り当てられるべきリソース使用予算の第2の部分を指定する少なくとも1つのメッセージを受け取る。

20

【0093】

[0107]ステップ1306において、リソースマネージャー132は、リソース使用予算の第1の部分に等しいリソースの一部をフォアグラウンドプロセスに割り当て、リソース使用予算の第2の部分に等しいリソースの一部を少なくとも1つのバックグラウンドタスクに割り当てる。

【0094】

30

[0108]実施形態では、フォアグラウンドプロセスに送られる少なくとも1つのメッセージ及びリソースマネージャーに送られる少なくとも1つのメッセージは、共有APIを介して送られる。

【0095】

[0109]別の実施形態では、リソース使用予算は、フォアグラウンドで実行されているアプリケーションについてのリソース使用予算を含み、リソース使用予算を示す少なくとも1つのメッセージは、アプリケーションがフォアグラウンドで実行されているという通知を含む。

【0096】

[0110]さらに別の実施形態では、リソース使用予算は、中断又は終了されるアプリケーションについてのリソース使用予算を含み、リソース使用予算を示す少なくとも1つのメッセージは、アプリケーションが中断又は終了されるという通知を含む。

40

【0097】

[0111]さらなる実施形態では、リソースマネージャー132は、所定の期間後、フォアグラウンドプロセスによって使用されているリソースの量及び少なくとも1つのバックグラウンドプロセスによって使用されているリソースの量の合計がリソース使用予算を超えているかどうかを判定し、合計がリソース使用予算を超えていると判定することに応答して、フォアグラウンドプロセス及び少なくとも1つのバックグラウンドプロセスのうちの1つ又は複数を終了させる。

【0098】

50

## I V . 例示的なモバイルデバイス実装

[0112]図14は、本願で説明される実施形態を実装することができる例示的なモバイルデバイス1402のブロック図である。図14に示すように、モバイルデバイス1402は、様々なオプションのハードウェアコンポーネント及びソフトウェアコンポーネントを含む。モバイルデバイス1402の任意のコンポーネントは任意の他のコンポーネントと通信することができるが、説明を簡単にするために、すべての接続が示されているわけではない。モバイルデバイス1402は、様々なコンピューティングデバイス（例えば、携帯電話、スマートフォン、ハンドヘルドコンピューター、パーソナルデジタルアシスタント（PDA）など）のうちの任意のものとしてでき、セルラー又は衛星ネットワークなどの1つもしくは複数のモバイル通信ネットワーク1404やローカルエリアもしくはワイドエリアネットワークとの無線双方向通信を可能にすることができる。

10

## 【0099】

[0113]図示されるモバイルデバイス1402は、信号符号化、データ処理、入出力処理、電力制御及び/又は他の機能などのタスクを実行するための、コントローラー又はプロセッサ1410（例えば、信号プロセッサ、マイクロプロセッサ、ASIC又は他の制御及び処理論理回路）を含むことができる。オペレーティングシステム1412は、モバイルデバイス1402のコンポーネントの割り当て及び使用を制御することができる、1つ又は複数のアプリケーションプログラム1414（「アプリケーション」又は「アプリ」とも呼ばれる）のサポートを提供することができる。アプリケーションプログラム1414は、一般的なモバイルコンピューティングアプリケーション（例えば、電子メールアプリケーション、カレンダー、コンタクトマネージャー、ウェブブラウザー、メッセージングアプリケーション）及び任意の他のコンピューティングアプリケーション（例えば、ワードプロセッシングアプリケーション、マッピングアプリケーション、メディアプレーヤーアプリケーション）を含むことができる。

20

## 【0100】

[0114]図示したモバイルデバイス1402はメモリー1420を含むことができる、メモリー1420は、取り外し不能なメモリー1422及び/又は取り外し可能なメモリー1424を含むことができる。取り外し不能なメモリー1422は、RAM、ROM、フラッシュメモリー、ハードディスク、又は他の周知のメモリーデバイスもしくは技術を含むことができる。取り外し可能なメモリー1424は、フラッシュメモリーもしくはGSM通信システムでよく知られている加入者識別モジュール（SIM）カード、又は「スマートカード」などの他の周知のメモリーデバイスもしくは技術を含むことができる。メモリー1420は、オペレーティングシステム1412及びアプリケーション1414を実行するためのデータ及び/又はコードを格納するために使用することができる。例示的なデータは、ウェブページ、テキスト、画像、音声ファイル、動画データ、又は、1つ又は複数の有線又は無線ネットワークを介して1つ又は複数のネットワークサーバー又は他のデバイスとの間で送受信される他のデータを含むことができる。メモリー1420は、国際モバイル加入者識別（IMSI）などの加入者識別子、及び国際モバイル機器識別子（IMEI）などの機器識別子を記憶するために使用することができる。このような識別子は、ユーザー及び機器を識別するために、ネットワークサーバーに送信することができる。

30

40

## 【0101】

[0115]モバイルデバイス1402は、タッチスクリーン1432、マイクロフォン1434、カメラ1436、物理キーボード1438及び/又はトラックボール1440などの1つ又は複数の入力デバイス1430、並びにスピーカー1452及びディスプレイ1454などの1つ又は複数の出力デバイス1450をサポートすることができる。タッチスクリーン1432などのタッチスクリーンは、様々な方法で入力を検出することができる。例えば、容量性タッチスクリーンは、物体（例えば、指先）が表面にわたって流れる電流を歪めたり中断したりするときに、タッチ入力を検出する。別の例として、タッチスクリーンは、光学センサーを使用して、光学センサーからのビームが遮断されたときにタ

50

タッチ入力を検出することができる。スクリーンの表面との物理的接触は、入力がいくつかのタッチスクリーンによって検出されるためには必要とされない。

【0102】

[0116]他の可能な出力デバイス（図示せず）は、圧電又は他の触覚出力デバイスを含むことができる。いくつかのデバイスは、複数の入出力機能を果たすことができる。例えば、タッチスクリーン1432及びディスプレイ1454は、単一の入出力デバイスへと組み合わせることができる。入力デバイス1430は、自然ユーザーインターフェイス（NUI）を含むことができる。

【0103】

[0117]無線モデム1460は、アンテナ（図示せず）に結合することができ、当技術分野で理解されるように、プロセッサ1410と外部デバイスとの間の双方向通信をサポートすることができる。モデム1460は、一般的に示されており、モバイル通信ネットワーク1404と通信するためのセルラーモデム1466及び/又は他の無線ベースのモデム（例えば、ブルートゥース（登録商標）1464及び/又はWi-Fi1462）を含むことができる。無線モデム1460の少なくとも1つは、典型的には、単一のセルラーネットワーク内、セルラーネットワーク間、又はモバイルデバイスと公衆交換電話網（PSTN）との間のデータ及び音声通信のためのGSM（登録商標）ネットワークなどの、1つ又は複数のセルラーネットワークとの通信のために構成される。

【0104】

[0118]モバイルデバイス1402は、さらに、少なくとも1つの入出力ポート1480、電源1482、全地球測位システム（GPS）受信機などの衛星ナビゲーションシステム受信機1484、加速度計1486、並びに/又は、USBポート、IEEE 1394（FireWire）ポート及び/もしくはRS-232ポートなどの物理コネクタ1490を含むことができる。当業者に認識されるように、任意のコンポーネントを削除することができ、他のコンポーネントを追加することができるので、モバイルデバイス1402の図示されたコンポーネントは、必須ものではなく、包括的なものではない。

【0105】

[0119]実施形態では、モバイルデバイス1402の特定のコンポーネントは、デバイス固有又はユーザー固有のリソース使用プロファイルに基づいてリソース管理を行い、マルチプロセスアプリケーションのための動的リソース管理を可能にすることに関連する、本願に記載の機能を実行するように構成される。例えば、一実施形態では、オペレーティングシステム1412は、デバイス固有又はユーザー固有のリソース使用プロファイルに基づいてオペレーティングシステム1412がリソース管理を実行することを可能にする、オペレーティングシステム104の機能を含む。さらなる例として、オペレーティングシステム1412及びアプリケーション1414の少なくとも1つは、それらのコンポーネントがマルチプロセスアプリケーションのための動的リソース管理を実行することを可能にする、オペレーティングシステム104及びアプリケーション102の機能を含むことができる。しかし、これは一例にすぎず、様々な機能が様々なコンポーネントによって実行されてもよい。

【0106】

[0120]デバイス固有又はユーザー固有のリソース使用プロファイルに基づいてリソース管理を行い、マルチプロセスアプリケーションのための動的リソース管理を可能にすることに関連する、本願に記載の機能を実行するためのコンピュータプログラムロジックが、メモリー1420に格納されて、プロセッサ1410によって実行されてもよい。このようなコンピュータプログラムロジックを実行することにより、プロセッサ1410は、図1を参照して上述したコンピューティングデバイス100のコンポーネントのうちの任意のものの機能のいずれかを実施することができる。また、このようなコンピュータプログラムロジックを実行することにより、プロセッサ1410は、図2-13に示されたフローチャートのうちのいずれか又はすべてのステップのうちのいずれか又はすべてを実行することができる。



## 【 0 1 0 7 】

## V . 例示的なコンピューターシステムの実施形態

[0121]図 1 5 は、本願に記載の様々な実施形態を実施するために使用され得る例示的なプロセッサベースのコンピューターシステム 1 5 0 0 を示す。例えば、システム 1 5 0 0 は、図 1 を参照して上述したコンピューティングデバイス 1 0 0 のコンポーネントを実装するために使用されてもよい。システム 1 5 0 0 はまた、図 2 - 1 3 に示されたフローチャートのうちのいずれか又はすべてのステップのいずれか又はすべてを実施するために使用されてもよい。本願で提供されるシステム 1 5 0 0 の説明は、例示の目的のために提供されるものであり、限定することを意図するものではない。実施形態は、当業者に知られているような、さらなる種類のコンピューターシステムで実現することができる。

10

## 【 0 1 0 8 】

[0122]図 1 5 に示すように、システム 1 5 0 0 は、処理ユニット 1 5 0 2、システムメモリー 1 5 0 4、及びシステムメモリー 1 5 0 4 を含む様々なシステムコンポーネントを処理ユニット 1 5 0 2 に結合するバスを含む。処理ユニット 1 5 0 2 は、1 つ又は複数のマイクロプロセッサ又はマイクロプロセッサコアを含むことができる。バス 1 5 0 6 は、メモリーバス又はメモリーコントローラー、周辺バス、加速グラフィックスポート及び様々なバスアーキテクチャーのうちの任意のものを使用するプロセッサ又はローカルバスを含む、いくつかの種類のバス構造のうちの 1 つ又は複数を表す。システムメモリー 1 5 0 4 は、読み出し専用メモリー ( R O M ) 1 5 0 8 及びランダムアクセスメモリー ( R A M ) 1 5 1 0 を含む。基本入出力システム 1 5 1 2 ( B I O S ) は R O M 1 5 0 8 に格納される。

20

## 【 0 1 0 9 】

[0123]システム 1 5 0 0 はまた、以下のドライブのうちの 1 つ又は複数を有する：ハードディスクからの読み出し及びそれへの書き込みのためのハードディスクドライブ 1 5 1 4、取り外し可能な磁気ディスク 1 5 1 8 からの読み出し又はそれへの書き込みのための磁気ディスクドライブ 1 5 1 6、及び C D R O M、D V D R O M、B L U - R A Y ( 商標 ) ディスク又は他の光学媒体などの取り外し可能な光ディスク 1 5 2 2 からの読み出し又はそれへの書き込みのための光ディスクドライブ 1 5 2 0。ハードディスクドライブ 1 5 1 4、磁気ディスクドライブ 1 5 1 6 及び光ディスクドライブ 1 5 2 0 は、それぞれ、ハードディスクドライブインターフェイス 1 5 2 4、磁気ディスクドライブインターフェイス 1 5 2 6 及び光ドライブインターフェイス 1 5 2 8 によってバス 1 5 0 6 に接続される。ドライブ及びその関連するコンピューター読み取り可能な媒体は、コンピューター読み取り可能な命令、データ構造、プログラムモジュール及びコンピューターのための他のデータについての不揮発性の記憶を提供する。ハードディスク、取り外し可能な磁気ディスク及び取り外し可能な光ディスクが説明されているが、フラッシュメモリーカード、デジタルビデオディスク、ランダムアクセスメモリー ( R A M )、読み取り専用メモリー ( R O M ) などの、他の種類のコンピューター読み取り可能なメモリーデバイス及びストレージ構造を、データを格納するために使用することができる。

30

## 【 0 1 1 0 】

[0124]多くのプログラムモジュールが、ハードディスク、磁気ディスク、光ディスク、R O M 又は R A M に格納されてもよい。これらのプログラムモジュールは、オペレーティングシステム 1 5 3 0、1 つ又は複数のアプリケーションプログラム 1 5 3 2、他のプログラムモジュール 1 5 3 4 及びプログラムデータ 1 5 3 6 を含む。様々な実施形態によれば、プログラムモジュールは、図 1 を参照して上述したコンピューティングデバイス 1 0 0 の機能及び特徴のうちのいずれか又はすべてを実行するために処理ユニット 1 5 0 2 によって実行可能なコンピュータープログラムロジックを含んでもよい。プログラムモジュールはまた、処理ユニット 1 5 0 2 によって実行されると、図 2 - 1 3 のフローチャートを参照して示した又は説明したステップ又は動作のうちの任意のものを実行する、コンピュータープログラムロジックを含むことができる。

40

## 【 0 1 1 1 】

50

[0125]ユーザーは、キーボード1538やポインティングデバイス1540などの入力デバイスを介して、コマンド及び情報をシステム1500に入力することができる。他の入力デバイス（図示せず）は、マイクロフォン、ジョイスティック、ゲームコントローラー、スキャナーなどを含むことができる。一実施形態では、タッチスクリーンが、タッチスクリーン上の1つ又は複数のポイントに対する（例えば、指又はスタイラスによる）タッチのアプリケーションを介してユーザーがユーザー入力を提供することを可能にするために、ディスプレイ1544に関連して設けられる。これら及び他の入力デバイスは、多くの場合、バス1506に結合されたシリアルポートインターフェイス1542を介して処理ユニット1502に接続されるが、パラレルポート、ゲームポート又はユニバーサルシリアルバス（USB）などの他のインターフェイスによって接続されてもよい。このようなインターフェイスは有線又は無線のインターフェイスであってもよい。

10

#### 【0112】

[0126]ディスプレイ1544はまた、ビデオアダプター1546などのインターフェイスを介してバス1506に接続される。ディスプレイ1544に加えて、システム1500は、スピーカー及びプリンターなどの他の周辺出力デバイス（図示せず）を含んでもよい。

#### 【0113】

[0127]システム1500は、ネットワークインターフェイスもしくはアダプター1550、モデム1552、又はネットワークを介して通信を確立するための他の適切な手段を介して、ネットワーク1548（例えば、ローカルエリアネットワーク又はインターネットなどのワイドエリアネットワーク）に接続される。モデム1552は、内蔵でも外付けでもよく、シリアルポートインターフェイス1542を介してバス1506に接続される。

20

#### 【0114】

[0128]本願で使用されるとき、「コンピュータープログラム媒体」、「コンピューター読み取り可能な媒体」及び「コンピューター読み取り可能な記憶媒体」という語は、ハードディスクドライブ1514に関連付けられるハードディスク、取り外し可能な磁気ディスク1518、取り外し可能な光ディスク1522のほか、フラッシュメモリーカード、デジタルビデオディスク、ランダムアクセスメモリー（RAM）、読み出し専用メモリー（ROM）などの他のメモリーデバイスやストレージ構造などの、メモリーデバイスやストレージ構造を広く指すために使用される。このようなコンピューター読み取り可能な記憶媒体は、通信媒体とは区別され、通信媒体とは重複しない（通信媒体を含まない）。通信媒体は、通常、搬送波などの変調データ信号に、コンピューター読み取り可能な命令、データ構造、プログラムモジュール又は他のデータを具体化する。「変調データ信号」という語は、信号内に情報を符号化するような方法で設定又は変更された1つ又は複数の特性を有する信号を意味する。限定ではなく例として、通信媒体は、音響、RF、赤外線及び他の無線媒体などの無線媒体を含む。実施形態は通信媒体にも向けられる。

30

#### 【0115】

[0129]上述のように、コンピュータープログラム及びモジュール（アプリケーションプログラム1532及び他のプログラムモジュール1534を含む）は、ハードディスク、磁気ディスク、光ディスク、ROM又はRAMに格納されてもよい。このようなコンピュータープログラムはまた、ネットワークインターフェイス1550、シリアルポートインターフェイス1542又は任意の他のインターフェイスタイプを介して受信することができる。このようなコンピュータープログラムは、アプリケーションによって実行又はロードされたときに、本願で説明した本発明の実施形態の機能をコンピューター1500が実施することを可能にする。したがって、このようなコンピュータープログラムは、システム1500のコントローラーを表す。

40

#### 【0116】

[0130]実施形態はまた、任意のコンピューター使用可能な媒体に格納されたソフトウェアを含むコンピュータープログラム製品に向けられる。このようなソフトウェアは、1つ

50

又は複数のデータ処理デバイスによって実行されると、データ処理デバイスを本願で説明された動作させる。本発明の実施形態は、現在又は将来において知られている、任意のコンピュータ使用可能又はコンピュータ読み取り可能な媒体を使用する。コンピュータ読み取り可能な媒体の例は、限定されないが、RAM、ハードドライブ、フロッピーディスク、CD ROM、DVD ROM、ZIPディスク、テープ、磁気ストレージデバイス、光ストレージデバイス、MEMS、ナノテクノロジーベースのストレージデバイスなどのメモリーデバイス及びストレージ構造を含む。

#### 【0117】

[0131]代替的な実施形態では、システム1500は、ハードウェア論理/電気回路又はファームウェアとして実装することができる。さらなる実施形態によれば、これらのコンポーネントのうちの1つ又は複数は、システムオンチップ(SoC)で実装されてもよい。SoCは、プロセッサ(例えば、マイクロコントローラー、マイクロプロセッサ、デジタル信号プロセッサ(DSP)など)、メモリー、1つ又は複数の通信インターフェイス、及び/もしくはさらなる回路、又はその機能を実行する組み込みファームウェアのうちの1つ又は複数を含む集積回路チップを含んでもよい。

#### 【0118】

#### VI. 結論

[0132]様々な実施形態が上で説明されたが、これらがほんの一例として提示されたものであって、限定ではないことが理解されるべきである。形態及び詳細の様々な変更が、本発明の趣旨及び範囲から逸脱することなくなされ得ることが、当業者には明らかであろう。したがって、本発明の広さ及び範囲は、上述した実施形態のいずれによっても限定されるべきではなく、以下の特許請求の範囲及びその均等物によってのみ規定されるべきである。

【図1】

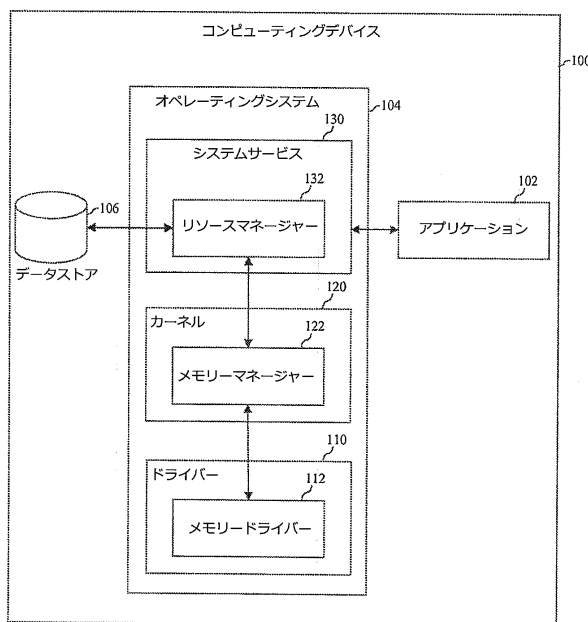


FIG. 1

【図2】

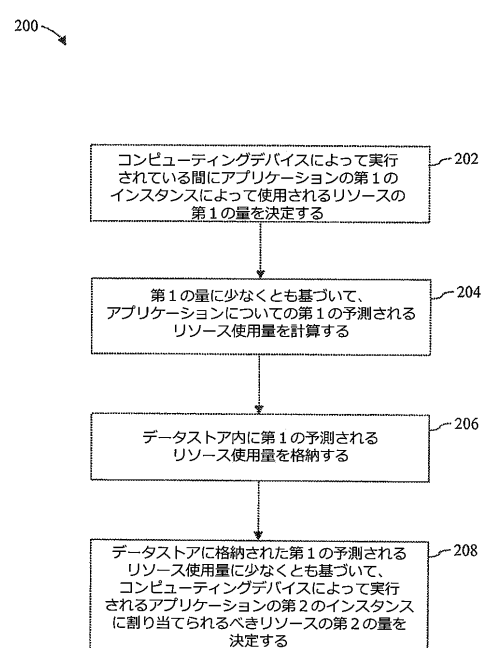


FIG. 2

【図 3】

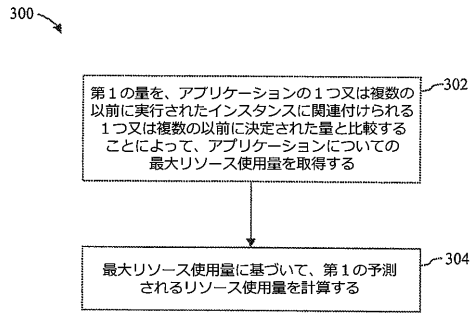


FIG. 3

【図 4】

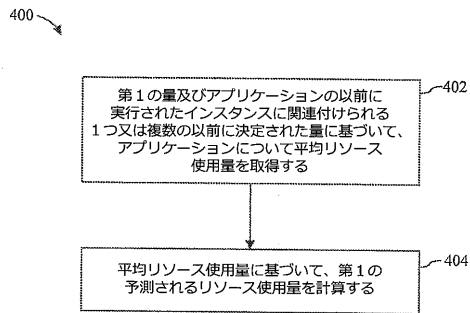


FIG. 4

【図 6】

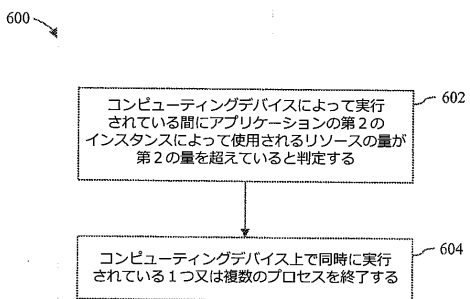


FIG. 6

【図 7】

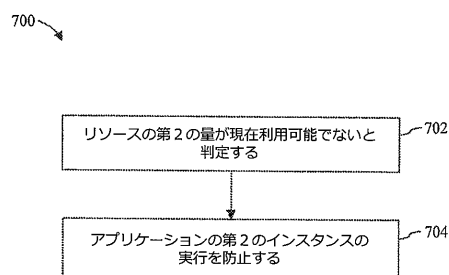


FIG. 7

【図 5】

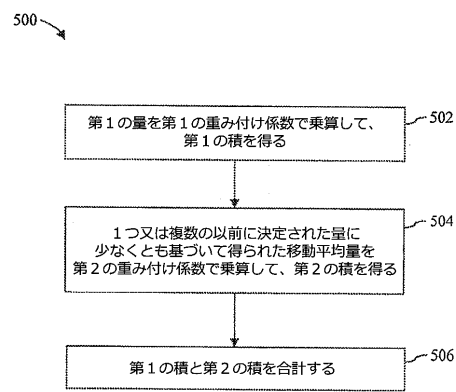


FIG. 5

【図 8】

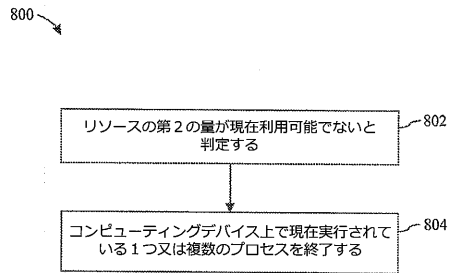


FIG. 8

【図 9】

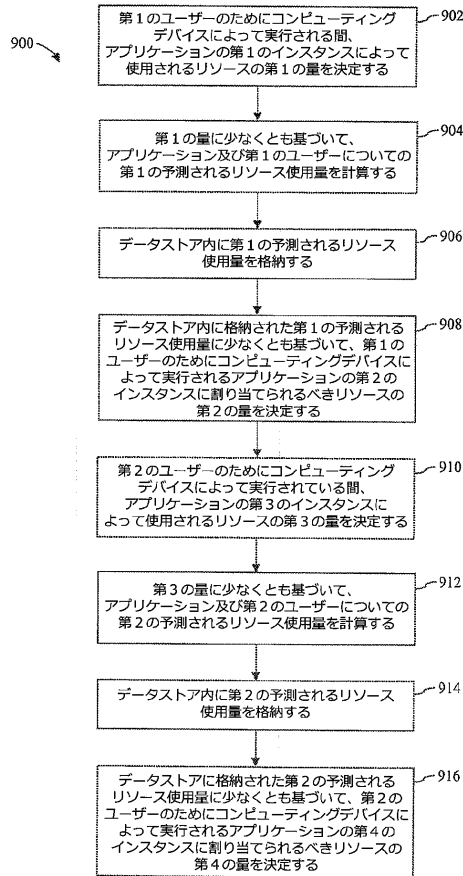


FIG. 9

【図 11】

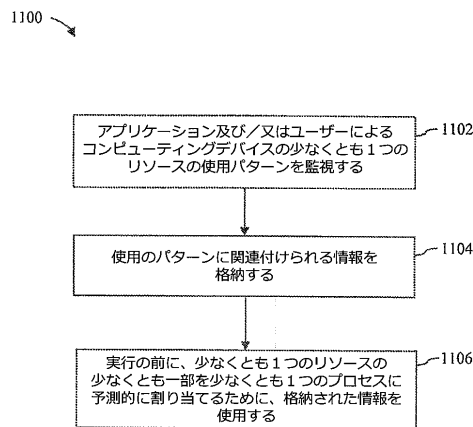


FIG. 11

【図 10】

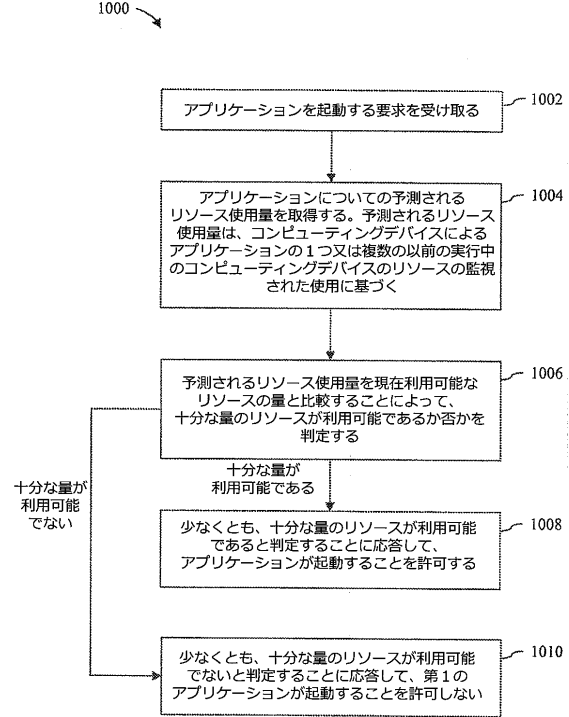


FIG. 10

【図 12】

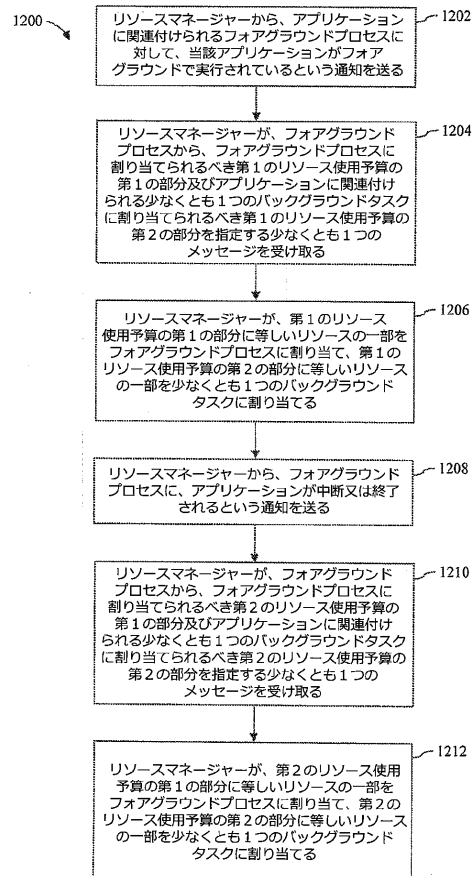


FIG. 12

【図 13】

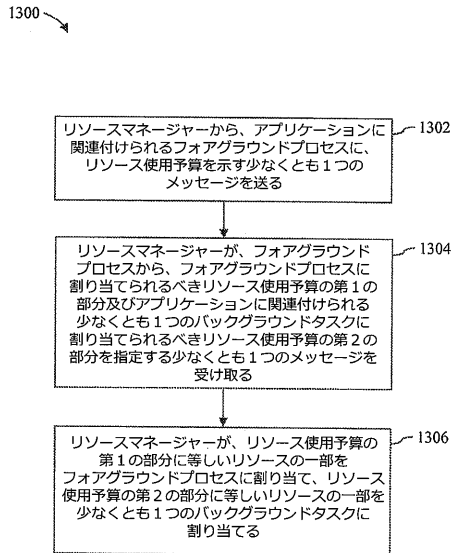


FIG. 13

【図 14】

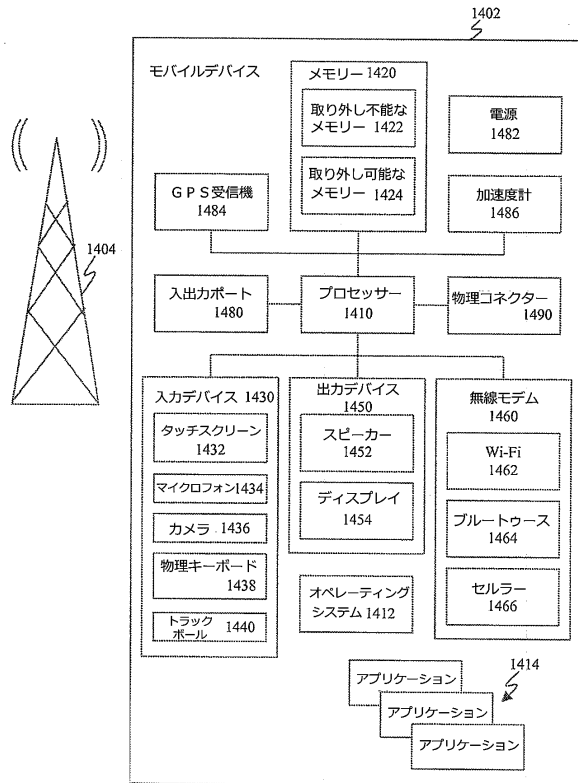


FIG. 14

【図 15】

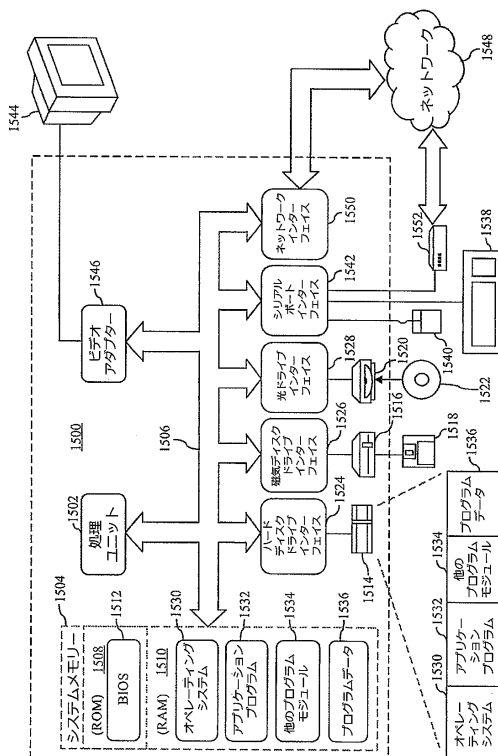


FIG. 15

## フロントページの続き

- (74)代理人 100147991  
弁理士 鳥居 健一
- (72)発明者 カーツマン, ティモシー・エム  
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ ( 8 / 1 1 7 2 )
- (72)発明者 ホワイトチャペル, デーヴィッド・エイ  
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ ( 8 / 1 1 7 2 )
- (72)発明者 グバデゲシン, アボレード  
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ ( 8 / 1 1 7 2 )
- (72)発明者 トルフィネスク, アディナ・エム  
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ ( 8 / 1 1 7 2 )
- (72)発明者 ロビンソン, ジェレミー・ビー  
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ ( 8 / 1 1 7 2 )
- (72)発明者 シャー, サマース・エイチ  
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ ( 8 / 1 1 7 2 )
- (72)発明者 カザ, アキレシュ  
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ ( 8 / 1 1 7 2 )
- (72)発明者 クリニック, アンドリュー  
アメリカ合衆国ワシントン州 9 8 0 5 2 - 6 3 9 9, レッドモンド, マイクロソフト コーポレーション, エルシーエイ - インターナショナル・パテンツ ( 8 / 1 1 7 2 )

審査官 大桃 由紀雄

- (56)参考文献 特開 2 0 0 6 - 2 8 5 8 7 1 ( J P , A )  
国際公開第 2 0 1 2 / 1 2 7 5 9 6 ( W O , A 1 )  
特表 2 0 1 0 - 5 3 3 9 2 4 ( J P , A )  
米国特許出願公開第 2 0 1 0 / 0 1 3 1 9 5 9 ( U S , A 1 )

- (58)調査した分野(Int.Cl. , D B 名)  
G 0 6 F 9 / 5 0