

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6514687号
(P6514687)

(45) 発行日 令和1年5月15日 (2019.5.15)

(24) 登録日 平成31年4月19日 (2019.4.19)

(51) Int. Cl.

F I

G 0 6 F 9/54 (2006.01)
G 0 6 F 9/50 (2006.01)G 0 6 F 9/54 B
G 0 6 F 9/50 1 5 0 Z

請求項の数 21 (全 19 頁)

(21) 出願番号 特願2016-514966 (P2016-514966)
 (86) (22) 出願日 平成26年5月14日 (2014.5.14)
 (65) 公表番号 特表2016-522946 (P2016-522946A)
 (43) 公表日 平成28年8月4日 (2016.8.4)
 (86) 国際出願番号 PCT/US2014/038009
 (87) 国際公開番号 W02014/189737
 (87) 国際公開日 平成26年11月27日 (2014.11.27)
 審査請求日 平成29年4月13日 (2017.4.13)
 (31) 優先権主張番号 61/825, 284
 (32) 優先日 平成25年5月20日 (2013.5.20)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 511117392
 パックサイズ, エルエルシー
 アメリカ合衆国ユタ州84104, ソルト
 ・レイク・シティ, ウェスト・スマート・
 パック・ウェイ 3760
 (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
 (74) 代理人 100138759
 弁理士 大房 直樹

最終頁に続く

(54) 【発明の名称】 ローカルまたは分散型コンピュータ・システムにおける柔軟なノード構成方法およびシステム

(57) 【特許請求の範囲】

【請求項 1】

1つ以上のプロセッサと、
 システム・メモリと、
 コンピュータ実行可能命令が格納されている1つ以上のコンピュータ読み取り可能記憶媒体と、

を含むコンピュータ・システムであって、前記命令が前記1つ以上のプロセッサによって実行されると、前記コンピュータ・システムに、サービス・スケーリング方法を実行させ、前記方法が、

サービス間における通信を可能にする1つ以上のメッセージ・キューを維持するメッセージ・ブローカ・サービスを設置するステップであって、前記メッセージ・キューが、パブリッシャからメッセージを受信し、メッセージをサブスクライバに転送するように構成される、ステップと、

前記1つ以上のサービスの各々に対して指定されたメッセージ・キューを示すステップであって、前記指定されたメッセージ・キューが、当該サービスに対するメッセージを維持するように構成される、ステップと、

前記サービスの内少なくとも1つを、第2の異なるコンピュータ・システムに移動させるステップであって、前記指定されたメッセージ・キューが、前記移動されるサービスに対するメッセージを維持する、ステップと、

を含む、コンピュータ・システム。

10

20

【請求項 2】

請求項 1 記載のコンピュータ・システムであって、前記方法は、更に、
各メッセージに、当該メッセージについてのトピックを示すトピック・キーを割り当てる、ステップと、
前記メッセージの割り当てられたトピック・キーに基づいて、指定されたトピック・キーに関連付けられたメッセージが、前記メッセージ・ブローカによって、当該トピックのメッセージにサブスクライブしたサブスクライバに発送されるように、メッセージをサブスクライバに発送するステップと、
を含む、コンピュータ・システム。

【請求項 3】

請求項 1 記載のコンピュータ・システムにおいて、パブリッシャが、そのメッセージに誰がサブスクライブしているかについて把握しない、コンピュータ・システム。

【請求項 4】

請求項 1 記載のコンピュータ・システムにおいて、前記メッセージ・ブローカが、前記サービスの内任意のものが前記メッセージ・キューに発行することを可能にし、更に任意のサブスクライバが前記メッセージ・キューからのメッセージを受信することを可能にする、コンピュータ・システム。

【請求項 5】

請求項 4 記載のコンピュータ・システムにおいて、前記メッセージの内 1 つ以上が、前記メッセージの出所を検証するセキュリティ・トークンを含む、コンピュータ・システム。

【請求項 6】

請求項 5 記載のコンピュータ・システムであって、前記方法は、更に、
前記受信したメッセージの少なくとも 1 つに対する前記セキュリティ・トークンが無効であると判定するステップと、
前記無効のセキュリティ・トークンを含む前記メッセージを無視するステップと、
を含む、コンピュータ・システム。

【請求項 7】

請求項 1 記載のコンピュータ・システムにおいて、前記サービスの内少なくとも 1 つが、パブリッシャおよびサブスクライバの双方である、コンピュータ・システム。

【請求項 8】

請求項 1 記載のコンピュータ・システムにおいて、複数のサービスが、複数の異なるコンピュータ・システムに跨がって分散される、コンピュータ・システム。

【請求項 9】

請求項 8 記載のコンピュータ・システムにおいて、前記サービスの内少なくとも 1 つが、前記複数の異なるコンピュータ・システム上においてそれ自体の少なくとも 2 つの異なるインスタンスを有する、コンピュータ・システム。

【請求項 10】

請求項 9 記載のコンピュータ・システムであって、前記方法は、更に、前記サービスの前記少なくとも 2 つの異なるインスタンスを使用して、負荷均衡化を実行するステップを含む、コンピュータ・システム。

【請求項 11】

請求項 1 記載のコンピュータ・システムにおいて、1 つ以上の新たなサービスが、ランタイム中に動的に追加される、コンピュータ・システム。

【請求項 12】

請求項 1 記載のコンピュータ・システムにおいて、前記サービスの内 1 つ以上が、ランタイム中に動的に除去される、コンピュータ・システム。

【請求項 13】

請求項 1 記載のコンピュータ・システムにおいて、前記サービスの内 1 つ以上が、ランタイム中に、1 つのコンピュータ・システムから他のコンピュータ・システムに動的に移

10

20

30

40

50

動させられる、コンピュータ・システム。

【請求項 14】

請求項 1 記載のコンピュータ・システムであって、前記方法は、更に、制御システムを前記メッセージ・ブローカ・サービスに統合するステップを含む、コンピュータ・システム。

【請求項 15】

請求項 14 記載のコンピュータ・システムであって、前記方法は、更に、外部システム統合サービスをインスタンス化するステップを含み、前記外部システム統合サービスが、指定された登録済みメッセージにサブスクライブし、前記登録済みメッセージを 1 つ以上の外部システムに転送するように構成される、コンピュータ・システム。

10

【請求項 16】

請求項 1 記載のコンピュータ・システムであって、前記方法は、更に、メッセージ・パイプライニング・サービスをインスタンス化するステップを含み、前記メッセージ・パイプライニング・サービスが、

指定された順序で処理されるべき 1 つ以上のメッセージをメッセージ・パイプラインにおいて受信し、

受信したメッセージを発行し、当該メッセージの結果にサブスクライブし、

前記メッセージ結果を、前記パイプラインにおいて以後受信するメッセージ毎に、入力引数として追加し、

メッセージ・パイプラインにおいて最後のメッセージが受信されたと判定し、

前記メッセージ・パイプラインの結果を 1 人以上のサブスクライバに発行する、

ように構成される、コンピュータ・システム。

20

【請求項 17】

請求項 1 記載のコンピュータ・システムであって、前記方法は、更に、サービス監視サービスをインスタンス化するステップを含み、前記サービス監視サービスに基づいて、

前記メッセージ・ブローカ・サービスを含む前記サービスの動作状態を監視するステップであって、当該監視するステップが、少なくとも 1 つのステータス・チェックを前記サービスの内 1 つ以上に送ることを含む、前記監視するステップと、

前記サービスが前記ステータス・チェックに応答したか否か判定するステップと、

前記ステータス・チェックに応答していないサービス毎に、1 人以上のユーザまたは他のシステムに、前記サービスが正しく動作していないことを通知するステップを、

実行するように構成される、コンピュータ・システム。

30

【請求項 18】

請求項 17 記載のコンピュータ・システムであって、前記方法は、更に、

指定されたメッセージ・タイプのメッセージを待ち受けることと、指定されたメッセージ・タイプのメッセージを発行することと、指定されたメッセージ・タイプのメッセージにサブスクライブすることのうち、少なくとも 1 つを実行しているサービスがどれであるのかを判定するステップと、

判定に基づいて、スループットを高めるためにサービスの数を増やす、および処理コストを低減するためにサービスの数を減らす、の内、少なくとも 1 つを実行するステップと

40

を含む、コンピュータ・システム。

【請求項 19】

請求項 18 記載のコンピュータ・システムであって、前記方法は、更に、

1 つ以上の従属サービスを含むどのサービスが、メッセージ処理のために現在利用可能であるか識別するステップと、

前記サービスの判定された現在の利用可能性に基づいて、1 つ以上のメッセージ処理設定を制御するステップと、

を含む、コンピュータ・システム。

【請求項 20】

50

少なくとも1つのプロセッサとメモリとを含むコンピュータ・システムにおいて、サービスをスケールリングするためのコンピュータ実装方法であって、

サービス間における通信を可能にする1つ以上のメッセージ・キューを維持するメッセージ・ブローカ・サービスを設置するステップであって、前記メッセージ・キューが、パブリッシャからメッセージを受信し、メッセージをサブスクライバに転送するように構成される、ステップと、

前記1つ以上のサービスの各々に対して指定されたメッセージ・キューを示すステップであって、前記指定されたメッセージ・キューが、当該サービスのためにメッセージを維持するように構成される、ステップと、

前記サービスの内少なくとも1つを第2の異なるコンピュータ・システムに移動させるステップであって、前記指定されたメッセージ・キューが、前記移動されるサービスに対するメッセージを維持する、ステップと、

を含む、コンピュータ実装方法。

【請求項21】

請求項20記載の方法であって、更に、

各メッセージに、当該メッセージについてのトピックを示すトピック・キーを割り当てるステップと、

前記メッセージの割り当てられたトピック・キーに基づいて、指定されたトピック・キーに関連付けられたメッセージが、前記メッセージ・ブローカによって、当該トピックのメッセージにサブスクライブしたサブスクライバに発送されるように、メッセージをサブ

を含む、コンピュータ実装方法。

【発明の詳細な説明】

【背景技術】

【0001】

関連出願に対する相互引用

[0001] 本願は、"METHOD AND SYSTEM FOR FLEXIBLE NODE COMPOSITION ON LOCAL OR DISTRIBUTED COMPUTER SYSTEMS" (ローカルまたは分散型コンピュータ・システムにおける柔軟なノード構成方法およびシステム)と題し、2013年5月20日に出願された米国仮特許出願第61/825,284号の恩恵および優先権を主張する。この仮特許出願をここで引用したことにより、その内容全体が本願にも含まれるものとする。

【発明の概要】

【0002】

[0002] 本明細書において説明する実施形態は、スケールリング・サービス(scaling service)、第1サービス・バージョンから第2バージョンへの移行、および外部システム統合サービスの実装を対象とする。一実施形態では、コンピュータ・システムは、サービス間における通信を可能にするメッセージ・キューを維持するメッセージ・ブローカ・サービス(message broker service)を設置する(establish)。メッセージ・キューは、パブリッシャからのメッセージを受け入れ、メッセージをサブスクライバに転送する。このコンピュータ・システムは、サービス毎に指定されたメッセージ・キューを示し、指定されたメッセージ・キューは、そのサービスに対するメッセージを維持するように構成される。また、このコンピュータ・システムは、サービスの内少なくとも1つを、第2の異なるコンピュータ・システムに移動させるのを可能にしつつ、指定されたメッセージ・キューは、移動されるサービスに対するメッセージを維持する。

【0003】

[0003] 他の実施形態では、コンピュータ・システムは第1サービス・バージョンから第2サービス・バージョンに移行する。このコンピュータ・システムは、サービスの第1バージョンをインスタンス化し、同じサービスの第2バージョンをインスタンス化して、サービスの第1バージョンおよびサービスの第2バージョンが同時に実行することになる。次いで、コンピュータ・システムは、処理のためにサービスの第1バージョンに送られ

たメッセージ（または、いずれのサービスにも未だ送られていないメッセージ）をサービスの第2バージョンに送るべきであると判断し、サービスの第2バージョンを使用してこのメッセージを処理する。

【0004】

[0004] 更に他の実施形態では、コンピュータ・システムは、外部システム統合サービスを実装する。このコンピュータ・システムは、内部メッセージ・ブローカ・サービスが設置されていると判断する。メッセージ・ブローカ・サービスは、サービス間における通信を可能するメッセージ・キューを維持するように構成される。メッセージ・キューは、パブリッシャからのメッセージを受け入れ、メッセージをサブスクライバに転送する。コンピュータ・システムは、内部メッセージ・ブローカ・サービスと合同して、外部システム統合サービスをインスタンス化する。外部システム統合サービスは、指定された登録済みメッセージにサブスクライブし(subscribe)、登録済みメッセージを種々の外部エンティティに転送するように構成される。外部システム統合サービスの構成設定(configuration)は、コンピュータ・システムによって、または外部システム統合サービスを使用して外部システムによって実行することができる。また、コンピュータ・システムは、外部システム統合サービスが登録したメッセージを受信し、外部システム統合サービスは、受信した登録済みメッセージを、指定された外部エンティティに転送する。

10

【0005】

[0005] この摘要は、詳細な説明において以下で更に説明する概念から選択したものを、簡略化した形態で紹介するために設けられている。この摘要は、特許請求する主題の主要な特徴や必須の特徴を特定することを意図するのではなく、特許請求する主題の範囲を判断するときに補助として使用されることを意図するのでもない。

20

【0006】

[0006] 更に他の特徴および利点については、以下に続く説明において明記され、当業者にはその説明から部分的に明らかになり、または本明細書における教示の実施によって習得することもできる。本明細書において説明する実施形態の特徴および利点は、添付した特許請求の範囲において特定の指摘した手段および組み合わせによって実現し修得することができる。本明細書において説明する実施形態の特徴は、以下の説明および添付した特許請求の範囲から一層明らかになるであろう。

【図面の簡単な説明】

30

【0007】

[0007] 本明細書において説明する実施形態の以上のおよびその他の特徴を更に明確化するために、添付図面を参照しながら更に特定の説明を行う。尚、これらの図面は本明細書において説明する実施形態の例を図示するに過ぎず、したがってその範囲を限定するように見なしてはならないことは認められよう。実施形態について、更に具体的にそして詳細に、添付図面の使用によって説明する(described and explained)。

【図1】図1は、本明細書において説明する実施形態がスケーリング・サービスを含んで動作することができるコンピュータ・アーキテクチャを示す。

【図2】図2は、スケーリング・サービスのための方法例のフロー・チャートを示す。

【図3】図3は、第1サービス・バージョンから第2バージョンに移行するための方法例のフロー・チャートを示す。

40

【図4】図4は、外部システム統合サービスを実装するための方法例のフロー・チャートを示す。

【図5】図5は、受信したメッセージをサービスの第1バージョンからサービスの第2バージョンに送る実施形態を示す。

【図6】図6は、メッセージを外部エンティティに転送するために外部システム統合サービスを実装する実施形態を示す。

【図7】図7は、指定されたトピック・キーを使用して、メッセージ・ブローカ・サービスによってメッセージを送信する(route)実施形態を示す。

【発明を実施するための形態】

50

【 0 0 0 8 】

[0015] 本明細書において説明する実施形態は、スケーリング・サービス、第1サービス・バージョンから第2バージョンへの移行、および外部システム統合サービスの実施を対象とする。一実施形態では、コンピュータ・システムは、サービス間における通信を可能にするメッセージ・キューを維持するメッセージ・ブローカ・サービスを設置する。メッセージ・キューは、パブリッシャからのメッセージを受け入れ、メッセージをサブスクライバに転送する。このコンピュータ・システムは、サービス毎に指定されたメッセージ・キューを示し、指定されたメッセージ・キューは、そのサービスに対するメッセージを維持するように構成される。また、このコンピュータ・システムは、サービスの内少なくとも1つを、第2の異なるコンピュータ・システムに移動させるのを可能にしつつ、指定されたメッセージ・キューは、移動されるサービスに対するメッセージを維持する。

10

【 0 0 0 9 】

[0016] 他の実施形態では、コンピュータ・システムは第1サービス・バージョンから第2サービス・バージョンに移行する。このコンピュータ・システムは、サービスの第1バージョンをインスタンス化し、同じサービスの第2バージョンをインスタンス化して、サービスの第1バージョンおよびサービスの第2バージョンが同時に実行することになる。次いで、コンピュータ・システムは、処理のためにサービスの第1バージョンに送られたメッセージ（または、いずれのサービスにも未だ送られていないメッセージ）をサービスの第2バージョンに送るべきであると判断し、サービスの第2バージョンを使用してこのメッセージを処理する。

20

【 0 0 1 0 】

[0017] 更に他の実施形態では、コンピュータ・システムは、外部システム統合サービスを実装する。コンピュータ・システムは、内部メッセージ・ブローカ・サービスが設置されていると判断する。メッセージ・ブローカ・サービスは、サービス間における通信を可能にするメッセージ・キューを維持するように構成される。メッセージ・キューは、パブリッシャからのメッセージを受け入れ、メッセージをサブスクライバに転送する。コンピュータ・システムは、内部メッセージ・ブローカ・サービスと合同して、外部システム統合サービスをインスタンス化する。外部システム統合サービスは、指定された登録済みメッセージにサブスクライブし、登録済みメッセージを種々の外部エンティティに転送するように構成される。外部システム統合サービスの構成設定(configuration)は、コンピュータ・システムによって、または外部システム統合サービスを使用している外部システムによって実行することができる。また、コンピュータ・システムは、外部システム統合サービスが登録したメッセージを受信し、外部システム統合サービスは、受信した登録済みメッセージを、指定された外部エンティティに転送する。

30

【 0 0 1 1 】

[0018] 次に、以下の論述は、実行することができる多数の方法および方法のアクトに言及する。尚、方法のアクトは、ある順序で論じられること、または特定の順序で行われるようにフロー・チャートにおいて示されることもあるが、特に述べられない限り、または要求されない限り、特定の順序付けは必ずしも必要とはされないことは注記してしかるべきである。何故なら、1つのアクトは、そのアクトが実行される前に完了する他のアクトに従属するからである。

40

【 0 0 1 2 】

[0019] 本明細書において説明する実施形態は、例えば、以下で更に詳しく論ずるように、1つ以上のプロセッサおよびシステム・メモリというようなコンピュータ・ハードウェアを含む特殊目的または汎用コンピュータを含むまたは利用することができる。また、本明細書において説明する実施形態は、コンピュータ実行可能命令および/またはデータ構造を搬送または格納するために、物理および他のコンピュータ読み取り可能媒体も含む。このようなコンピュータ読み取り可能媒体は、汎用または特殊目的コンピュータ・システムによってアクセスすることができる任意の入手可能な媒体とすることができる。コンピュータ実行可能命令をデータの形態で格納するコンピュータ読み取り可能媒体は、コン

50

ピュータ記憶媒体である。コンピュータ実行可能命令を搬送するコンピュータ読み取り可能媒体は、送信媒体である。つまり、一例として、そして限定ではなく、本明細書において説明する実施形態は、少なくとも2つの全く異なる種類のコンピュータ読み取り可能媒体、即ち、コンピュータ記憶媒体および送信媒体を含むことができる。

【0013】

[0020] コンピュータ記憶媒体は、RAM、ROM、EEPROM、CD-ROM、RAMに基づくソリッド・ステート・ドライブ(SSD)、フラッシュ・メモリ、相変化メモリ(PCM)、または他のタイプのメモリ、あるいは他の光ディスク記憶デバイス、磁気ディスク記憶デバイス、または他の磁気記憶デバイス、あるいはコンピュータ実行可能命令、データ、またはデータ構造の形態で所望のプログラム・コードを格納するために使用することができ、更に汎用または特殊目的コンピュータ・システムによってアクセスすることができる任意の他の媒体を含む。

10

【0014】

[0021] 「ネットワーク」とは、コンピュータ・システムおよび/またはモジュールおよび/または他の電子デバイス間における電子データの移送を可能にする1つ以上のデータ・リンクおよび/またはデータ・スイッチとして定義するものとする。ネットワークを通じて情報をコンピュータに転送または提供するとき(ハードワイヤ、ワイヤレス、あるいはハードワイヤまたはワイヤレスの組み合わせのいずれか)、コンピュータがこの接続を送信媒体と見なすのは適正である。送信媒体は、データまたは所望のプログラム・コード手段をコンピュータ実行可能命令の形態でまたはデータ構造の形態で搬送するために使用することができ、汎用または特殊目的コンピュータ・システムによってアクセスすることができるネットワークを含むことができる。以上のものの組み合わせも、コンピュータ読み取り可能媒体の範囲に含まれてしかるべきである。

20

【0015】

[0022] 更に、種々のコンピュータ・システムのコンポーネントに達したときに、コンピュータ実行可能命令またはデータ構造の形態であるプログラム・コード手段を自動的に送信媒体からコンピュータ記憶媒体に(またはその逆に)転送する(transfer)ことができる。例えば、ネットワークまたはデータ・リンクを通じて受信されたコンピュータ実行可能命令またはデータ構造を、ネットワーク・インターフェース・モジュール(例えば、ネットワーク・インターフェース・カードまたは「NIC」)内にあるRAMにバッファし、次いで最終的にコンピュータ・システムのRAMにおよび/またはコンピュータ・システムにおいてそれよりも揮発性が少ないコンピュータ記憶媒体に転送することができる。このため、コンピュータ記憶媒体は、送信媒体も利用する(または主に利用する場合も)コンピュータ・システム・コンポーネントに含めることができることは理解されてしかるべきである。

30

【0016】

[0023] コンピュータ実行可能(またはコンピュータ解釈可能)命令は、例えば、汎用コンピュータ、特殊目的コンピュータ、または特殊目的処理デバイスに、ある種の機能または1群の機能を実行させる命令を含む。コンピュータ実行可能命令は、例えば、バイナリー、アセンブリ言語のような中間フォーマット命令、またはソース・コードであってもよい。主題については、構造的特徴および/または方法論的アクトに特定の文言で説明したが、添付した特許請求の範囲において定められる主題は、必ずしも以上で説明した特徴やアクトには限定されないことは理解されよう。逆に、説明した特徴およびアクトは、特許請求の範囲を実現する形態例として開示したまでである。

40

【0017】

[0024] 当業者は、種々の実施形態を、多くのタイプのコンピュータ・システム構成を有するネットワーク計算環境において実施できることを認めよう。コンピュータ・システム構成には、パーソナル・コンピュータ、デスクトップ・コンピュータ、ラップトップ・コンピュータ、メッセージ・プロセッサ、ハンドヘルド・デバイス、マルチプロセッサ・システム、マイクロコンピュータ・ベースまたはプログラマブル消費者用電子機器、ネッ

50

トワークPC、ミニコンピュータ、メインフレーム・コンピュータ、移動体電話機、PDA、タブレット、ページャ、ルータ、スイッチ等が含まれる。また、本明細書において説明する実施形態は、分散型システム環境において実施することも可能である。分散型システム環境では、ネットワークを通じてリンクされた（ハードワイヤ・データ・リンク、ワイヤレス・データ・リンク、またはハードワイヤおよびワイヤレス・データ・リンクの組み合わせによって）ローカルおよびリモート・コンピュータ・システムの各々がタスク（例えば、クラウド・コンピューティング、クラウド・サービス等）を実行する。分散型システム環境では、プログラム・モジュールはローカルおよびリモート双方のメモリ記憶デバイスに配置されてもよい。

【0018】

10

[0025] この説明およびそれに続く特許請求の範囲において、「クラウド・コンピューティング」とは、構成変更可能な計算リソース（例えば、ネットワーク、サーバ、ストレージ、アプリケーション、およびサービス）の共有プールに対する、応需型(on-demand)ネットワーク・アクセスを可能にするモデルとして定義するものとする。「クラウド・コンピューティング」の定義は、このようなモデルが適正に展開されたときに得ることができる他の多数の利点のいずれにも限定されないものとする。

【0019】

[0026] 例えば、クラウド・コンピューティングは、現在、構成変更可能な計算リソースの共有プールに対する遍在的で便利な、応需型アクセスを提供するように、マーケットプレイスにおいて採用されている。更に、構成変更可能な計算リソースの共有プールは、仮想化によって素早くプロビジョニングすることができ、少ない管理の手間やサービス・プロバイダの介入で解放することができ、次いでこれらに応じてスケーリングすることができる。

20

【0020】

[0027] クラウド・コンピューティング・モデルは、例えば、応需型セルフ・サービス、ブロード・ネットワーク・アクセス、リソース・プーリング、即時性(rapid elasticity)、計測可能であること(measured service)等というような、種々の特性で構成することができる。また、クラウド・コンピューティング・モデルは、例えば、サービスとしてのソフトウェア(「SaaS」)、サービスとしてのプラットフォーム(「PaaS」)、およびサービスとしてのインフラストラクチャー(「IaaS」)というような、種々のサービス・モデルを露出することもできる。また、クラウド・コンピューティング・モデルは、プライベート・クラウド、コミュニティ・クラウド、パブリック・クラウド、ハイブリッド・クラウド等というような、異なる配備モデルを使用して配備することもできる。この説明および特許請求の範囲では、「クラウド・コンピューティング環境」とは、クラウド・コンピューティングが採用された環境である。

30

【0021】

[0028] 加えてまたは代わりに、本明細書において説明する機能は、少なくとも部分的に、1つ以上のハードウェア論理コンポーネントによって実行することができる。例えば、限定ではなく、使用することができるハードウェア論理コンポーネントの代表的なタイプは、フィールド・プログラマブル・ゲート・アレイ(FPGA)、特定プログラム集積回路(ASIC)、特定プログラム標準製品(ASSP)、チップ上システム・システム(SOC)(System-on-a-Chip System)、複合プログラマブル論理デバイス(CPLD)、および他のタイプのプログラマブル・ハードウェアを含む。

40

【0022】

[0029] 更にまた、本明細書において説明するシステム・アーキテクチャは、複数の独立したコンポーネントを含むことができ、その各々がシステム全体としての機能に寄与する。このモジュール性は、プラットフォームのスケラビリティの問題に取り組むときに、柔軟性を高めることを可能とし、このために、種々の利点が得られる。システムの複雑さおよび成長は、機能範囲が制限された小規模な(smaller-scale)部品の使用によって、一層容易に管理することができる。プラットフォームのフォールト・トレランスは、これ

50

らの疎結合モジュールの使用によって高められる。業務上の必要性から求められるに連れて、個々のコンポーネントを漸増的に増大させることができる。また、モジュール型の開発は、新たな機能を市場に投入するまでの時間の短縮となる。新たな機能は、コア・システムに影響を及ぼすことなく、追加または削除することができる。

【 0 0 2 3 】

[0030] 図 1 は、少なくとも 1 つの実施形態を採用することができるコンピュータ・アーキテクチャ 100 を示す。コンピュータ・アーキテクチャ 100 は、第 1 コンピュータ・システム 101 および第 2 コンピュータ・システム 108 を含む。第 1 および第 2 コンピュータ・システム 101 および 108 は、クラウド・コンピューティング・システムを含む、任意のタイプのローカルまたは分散型コンピュータ・システムであってもよい。これらのコンピュータ・システムは、種々の異なる機能を実行するモジュールを含むことができる。例えば、第 1 コンピュータ・システム 101 はメッセージ・ブローカ・サービス 102 を含む。メッセージ・ブローカ・サービスは、第 1 コンピュータ・システム上でインスタンス化され、サービス A、B、および C（それぞれ、105 A、105 B、および 105 C）を含むがこれらに限定されないサービスと相互作用するために使用することができる。サービス A ~ C は、任意のタイプのコンピュータ・サービス、アプリケーション、機能、または他のタイプのソフトウェア機能でもよい。

【 0 0 2 4 】

[0031] メッセージ・ブローカ・サービス 102 は、メッセージ・キュー 103 を維持するように構成することができる。メッセージ・キューは、サービス（105 A ~ 105 C）から発行されたメッセージを受け入れることができ、これらのメッセージのサブスクライバに、これらのメッセージを送ることができる。つまり、図 1 に示すように、サービス A はサービス B および C からのメッセージにサブスクライブされる。したがって、サービス B または C がメッセージ（例えば、発行メッセージ 106）を発行するときはいつでも、対応するメッセージ・キュー B および C（それぞれ、103 B および 103 C）が、サブスクライブされたいずれのメッセージ 107 もサービス A に転送する。同様に、サービス B はサービス C のメッセージにサブスクライブされたので、メッセージ・キュー C（103 C）は、サービス C によって発行されサブスクライブされたメッセージ 107 を、サービス B に転送する。更に、サービス C はサービス A のメッセージにサブスクライブされたので、メッセージ・キュー A（103 A）は、サービス A によって発行されサブスクライブされたメッセージ 107 をサービス C に転送する。場合によっては、メッセージ・ブローカ・サービス 102 は 1 つのキューまたは多数のキューを維持してもよいことは注記してしかるべきである。例えば、サービスは、トピックに基づいてメッセージにサブスクライブすることもできる。したがって、実際には 1 つのメッセージ・キューが多くの異なるトピックのメッセージを有するときでも、サービスにとっては、キューがそのトピックのメッセージのみをサービスするように思われるであろう。

【 0 0 2 5 】

[0032] 図 1 には 3 つのサービスを示すが、そしてこれらのサービスの各々は、パブリッシャおよびサブスクライバ双方として示されているが、実質的にいかなる数のサービスでもインスタンス化することができ、これらのサービスのいずれもが、パブリッシャのみ、サブスクライバのみ、またはパブリッシャおよびサブスクライバ双方になれることは理解されよう。このように、サービスが、ある種のサービスからメッセージを受信するためにサブスクライブすることができ、これらのサービスは、誰がサブスクライブされたか、そして誰にメッセージを送るのかについて心配する必要はない。むしろ、サービスは彼らのメッセージをメッセージ・ブローカに発行することができ、メッセージ・ブローカのメッセージ・キューが、メッセージを受信するためにサブスクライブしたサービスに、これらのメッセージを適正に転送する。

【 0 0 2 6 】

[0033] 尚、サービスを 1 つの計算システムから他に移動させてもよいことも注記してしかるべきである。例えば、サービス B（105 B）を第 1 コンピュータ・システム 10

10

20

30

40

50

1 から第 2 コンピュータ・システム 1 0 8 に移動させてもよい。サービスは、サービスが実行を停止した後に移動させればよいが、サービスが実行している間に、動的に移動させることもできる。したがって、コンピュータ・システム 1 0 1 のサービス移動モジュール 1 0 4 によってランタイム中に計算システム間でサービスを移動させるまたは移転させることができる。これらの概念については、それぞれ、図 2、図 3、および図 4 の方法 2 0 0、3 0 0、および 4 0 0 に関して、以下で更に説明する。

【 0 0 2 7 】

[0034] 以上で説明したシステムおよびアーキテクチャに関して、開示する主題にしたがって実装することができる方法(methodologies)は、図 2、図 3、および図 4 のフロー・チャートを参照すると、一層良く評価できよう。説明の簡単さのために、一連のブロックとして方法を示し説明する。しかしながら、特許請求する主題は、ブロックの順序によって限定されないことは理解され認められてしかるべきである。何故なら、あるブロックは異なる順序で現れてもよく、および/または本明細書において図示および説明する他のブロックと同時に現れてもよいからである。更に、図示するブロックの全てが、以下に説明する方法を実現するために必要ではない場合もある。

【 0 0 2 8 】

[0035] 図 2 は、スケーリング・サービスのための方法 2 0 0 のフロー・チャートを示す。これより、方法 2 0 0 について、環境 1 0 0 のコンポーネントおよびデータを頻繁に参照しながら説明する。

【 0 0 2 9 】

[0036] 方法 2 0 0 は、サービス間において通信を可能にする 1 つ以上のメッセージ・キューを維持するメッセージ・ブローカ・サービスを設置するブロック(2 1 0)を含む。メッセージ・キューは、パブリッシャからメッセージを受信し、メッセージをサブスクライバに転送するように構成される。例えば、第 1 コンピュータ・システム 1 0 1 がメッセージ・ブローカ・サービス 1 0 2 を設置してもよい。メッセージ・ブローカ・サービス 1 0 2 は、サービス 1 0 5 A ~ 1 0 5 C (または図示しないその他)というようなサービス間における通信を容易にする。メッセージ・ブローカ・サービス 1 0 2 は、パブリッシャからのメッセージ(即ち、発行メッセージ 1 0 6)を受信し、メッセージ(即ち、サブスクライブされたメッセージ 1 0 7)をサブスクライバに転送する種々のメッセージ・キュー 1 0 3 を始動させ(initiate)管理する。つまり、パブリッシャ(例えば、サービス 1 0 5 A)が、その動作ステータスに係するメッセージ、その動作結果、通知メッセージ、または他のタイプのメッセージを発行することができる。これらのメッセージは、誰が実際にこれらのメッセージにサブスクライブしているかに関係なく、発行することができる。

【 0 0 3 0 】

[0037] ある場合には、メッセージ・ブローカ・サービス 1 0 2 と通信するサービスは、パブリッシャだけであってもよい(即ち、これらはいずれの他のサービスのメッセージにもサブスクライブされない)。他の場合では、サービスがサブスクライバだけであってもよい(即ち、これらは他のサブスクライバのためにメッセージを発行しない)。更に他の場合では、サービス 1 0 5 A ~ 1 0 5 C は、パブリッシャおよびサブスクライバ双方であってもよい(即ち、これらは、メッセージの発行、これらがサブスクライブされたメッセージ(例えば、サブスクライブされたメッセージ 1 0 7)の受信の双方を行う)。サービス 1 0 5 A ~ 1 0 5 C は、1 つのコンピュータ・システム、または多数のコンピュータ・システム上で実行することもできる。更に、サービスは、動作の前、後、または最中に、コンピュータ・システム間で移動させることができる。ある場合には、サービスが、同じまたは多数の異なるコンピュータ・システム上で実行する、それ自体の少なくとも 2 つの異なるインスタンスを有することもできる。例えば、メッセージ・ブローカ・サービス 1 0 2 は、フェイルオーバーまたは他の目的のために、多数の異なるコンピュータ・システム上において、それ自体の 2 つ以上の異なるインスタンスを有することができる。

【 0 0 3 1 】

[0038] 更に、方法 200 は、1 つ以上のサービスの各々に対して、指定されたメッセージ・キューを示すブロック (220) を含む。指定されたメッセージ・キューは、そのサービスのためにメッセージを維持するように構成される。メッセージ・ブローカ・サービス 102 は、多くの異なるメッセージ・キュー 103 を始動させることができる。これらのメッセージ・キューの一部または全部が、ある種のサービスのためのメッセージを処理するために特定の割り当てられてもよい。つまり、サービス A (105A) のメッセージを処理するために、メッセージ・キュー A (103A) を特定の指定することができる。したがって、メッセージ・キュー A は、サービス A の発行メッセージ 106 を受信し、サービス A のメッセージを受信するためにサブスクライブされたサービス (または他のエンティティ) (例えば、サービス C (105C)) にこれらを送る。同様に、メッセージ・キュー B (103B) が、サービス B のメッセージを処理するために割り当てられてもよく、メッセージ・キュー C (103C) が、サービス C のメッセージを処理するために割り当てられてもよい。

10

【0032】

[0039] 次いで、方法 200 は、サービスの内少なくとも 1 つを第 2 の異なるコンピュータ・システムに移動させるブロック (230) を含み、指定されたメッセージ・キューが、移動されるサービスに対するメッセージを維持する。第 1 コンピュータ・システム 101 のサービス移動モジュール 104 は、サービス 105 ~ 105C の内任意の 1 つ以上を他のコンピュータ・システムに移動させることができる。つまり、例えば、サービス移動モジュール 104 は、サービス B (105B) をコンピュータ・システム 101 から第 2 コンピュータ・システム 108 に移動させることができる。サービスは、サービスの動作中における任意の時点に移動させることができる。ランタイムの前、後、または最中に、追加のサービスを動的に追加することができる。同様に、ランタイムの前、後、または最中に、既存のサービスを動的に除去することもできる。スループットを高めるため、または追加のコンピュータ・システムを使用するコストを低減するため、負荷均衡化を補助するため、フェイルオーバーを補助するため、または他の理由のために、サービスを移動させる、追加する、または除去することができる。本明細書において説明する実施形態は、2 つ以上のサービスを伴う実施形態について説明するが、これは簡単さのために行われるに過ぎず、スループット、負荷均衡化、フェイルオーバー、または他の理由で補助するために、実質的にいかなる数のサービスでも移動させる、追加する、または除去することができる。

20

30

【0033】

[0040] 図 7 に示すようなある実施形態では、メッセージ・ブローカ・サービス 701 のトピック・キー割り当てモジュール 702 が、各メッセージ 703 にトピック・キー 704 を割り当てることができる。トピック・キー 704 は、そのメッセージ 703 についてのトピックを示す。トピックは、例えば、メッセージ 703 が通知メッセージである、または報告メッセージであることを示すことができる。これらのメッセージは、その対応するトピック・キーによって、メッセージの指定されたトピック・キーに基づいてサブスクライバに発送することができる。したがって、指定されたトピック・キー 704 に関連付けられたメッセージは、メッセージ・ブローカ・サービス 701 によって、そのトピックのメッセージにサブスクライブしたサブスクライバに発送される。例えば、サービス A (705A) がある種のトピック (例えば、「エラー」または「完了」または「動作ステータス」) のメッセージにサブスクライブした場合、これらのメッセージ主題に対応するトピック・キーを有するメッセージがサービス A に発送される。メッセージ・ブローカ・サービス 701 は、同様に、サービス B (705B) および C (705C) がどのメッセージ・トピックにサブスクライブしたかに基づいて、メッセージをこれらのサービスに発送する。このように、これらの実施形態では、サブスクライバは、ある種のパブリッシャからの全てのメッセージにサブスクライブすること、またはそのパブリッシャからのメッセージの内、指定されたトピック・キーを有するものだけにサブスクライブすることもできる。更に、サブスクライバは、単に、誰がパブリッシャであるかには関係なく、ある種

40

50

のトピック・キーを有するメッセージにサブスクライブすることもできる。このように、パブリッシャは、イベントが発生するに連れてメッセージを発行することができ、メッセージ・ブローカ・サービス 701 によってこれらのメッセージをサブスクライバに発送することができる。

【0034】

[0041] 少なくともある場合には、パブリッシャは、それらのメッセージに誰がサブスクライブしているのか全く知らなくてもよい。メッセージ・ブローカは、サービスの任意のものがメッセージをメッセージ・キューに発行することを可能にすることができ、更に、任意のサブスクライバがメッセージ・キューからメッセージを受信することを可能にすることができる。サブスクライブされたメッセージ 107 は、サブスクライバに、誰がメッセージを発行したか示すことができ、更にこのメッセージに割り当てられたトピック・キーを示すこともできる。ある場合には、メッセージは、メッセージの出所（即ちパブリッシャ）を検証するセキュリティ・トークンを含むこともできる。メッセージ・サブスクライバは、セキュリティ・トークンにアクセスし、それが有効か否か判定することができる。有効である場合、メッセージはメッセージ・キュー上に発行される。セキュリティ・トークンが無効である場合、メッセージには無効の印が付けられ、破棄または無視することができる。更に、無効なセキュリティ・トークンのパブリッシャ、および/またはサブスクライバに、そのセキュリティ・トークンが無効であることを通知することもできる。ある実施形態では、セキュリティ・ポリシーを立案し(initiate)、あるメッセージは、セキュリティ・トークンなしでアクセスし発送することを許可し、一方他の（恐らく優先度が高い）メッセージは、メッセージにアクセスできるようになる前に有効なトークンを要求する。更に、ある実施形態では、第1コンピュータ・システム 101 があるサービスから他のサービスに移行しなければならない場合もある。このような実施形態については、以下で図3と関連付けて説明する。

【0035】

[0042] ある実施形態では、サービス監視サービスを実装することもできる。サービス監視サービスは、システムにおけるサービスの動作状態を監視することができる。サービス監視サービスは、1つの指定されたサービス、またはメッセージ・ブローカ・サービスを含む複数の異なるサービスを監視するように構成することができる。監視は、ステータス・チェックをサービスの内1つ以上に送ることを含んでもよい。これらのステータス・チェックは、サービスとサービス監視サービスとの間で送られる内部メッセージであってもよい。これらのステータス・チェックは、サービスが適正に機能しているか否か判定するために使用することができる。このように、サービス監視サービスは、サービスがステータス・チェックに回答したか否か判定し、ステータス・チェックに回答していないサービス毎に、そのサービスが正しく動作していないことを種々のユーザ（例えば、システム・アドミニストレータを含む）に通知することができる。

【0036】

[0043] サービス監視サービスは、更に、どのサービスが、指定されたメッセージ・タイプを待ち受けているか判定し、指定されたメッセージ・タイプを発行し、指定されたメッセージ・タイプにサブスクライブし、指定された数のメッセージを処理し、またはシステム全体によって処理されるメッセージの数を判定するように構成することができる。ある場合には、サービスが既知の量以上または以下に処理している場合、そのサービスは誤って機能しているおそれもある。同様に、ある種のメッセージ・タイプに対するパブリッシャまたはサブスクライバが、比較的大量のメッセージを受信または発行することもあり、つまり一層エラーを生じ易くなるおそれがある。サービス監視サービスからのデータは、サービスの数を増やしメッセージ処理スループットを高めるために使用することができ、またはサービスの数を減らし処理コストを低減することもできる。これらのサービスの増減は、自動的に、または処理の必要性が変化するに連れて動的に実施することができる。更に、サービス監視サービスは、手動でサービスを追加または除去するための推奨を与えるように構成することもできる。

【 0 0 3 7 】

[0044] 更にまた、サービス監視サービスは、従属サービスを含むどのサービスが、現在メッセージ処理のために利用可能か識別することもできる。次いで、この情報に基づいて、サービス監視サービスは、判定された現在のサービスの利用可能性に基づいて、種々のメッセージ処理設定を制御することができる。例えば、システム内部のサービス、または外部システム統合サービスを使用する外部システムにおけるサービスは、サービス監視サービスによって提示される利用可能性情報にしたがって、メッセージを異なって処理するように構成することができる。例えば、パブリッシャが、利用可能性情報から、現在利用可能なサービスにメッセージをどのようにして発送するか決定することができるのもよい。

10

【 0 0 3 8 】

[0045] 図 3 は、第 1 サービス・バージョンから第 2 バージョンに移行するための方法 3 0 0 のフロー・チャートを示す。これより、図 1 の環境 1 0 0 および図 5 の環境 5 0 0 のコンポーネントおよびデータを頻繁に参照しながら、方法 3 0 0 について説明する。

【 0 0 3 9 】

[0046] 方法 3 0 0 は、サービスの第 1 バージョンをインスタンス化するブロック (3 1 0) を含む。例えば、コンピュータ・システム 5 0 1 のサービス・インスタンス化モジュール 5 0 2 が、サービス 5 0 3 A の第 1 バージョンをインスタンス化することができ、更に同じサービスの第 2 バージョン 5 0 3 B をインスタンス化することができ、サービスの第 1 バージョンおよびサービスの第 2 バージョンが同時に実行することになる (3 2 0)。前述のように、このサービスは、3 つ以上のバージョンを含む、多数の異なるバージョンにインスタンス化することもできる (図 5 には示されていない)。これらのサービスは、サービス、アプリケーション、ソフトウェア機能、または他のソフトウェア・エレメントを含む、任意のタイプの計算機能とすることができる。ある場合には、第 1 バージョン 5 0 3 A が、サービスの新しいアップグレードされたバージョンであってもよく、一方第 2 バージョン 5 0 3 B は、古いアップグレードされていないバージョンであってもよい。これらの新旧バージョンは、同時に実行することができ、別個に実行することもできる。

20

【 0 0 4 0 】

[0047] 次に、方法 3 0 0 は、処理のためにサービスの第 1 バージョンに送られたメッセージをサービスの第 2 バージョンに送るべきことを決定するブロック (3 3 0) を含む。例えば、第 1 バージョンの方が新しいバージョンであり、第 2 バージョンの方が古いバージョンである先の例では、サービスの古い方のバージョンにのみ有効である、または新しい方のソフトウェア・バージョンによって認識されないまたは処理できない 1 つのエレメントを少なくとも含むメッセージ 5 0 6 が受信され場合があり得る。逆の場合も成り立つことができ、その場合、第 1 バージョン 5 0 3 A が古い方のバージョンであり、第 2 バージョン 5 0 3 B が新しい方のバージョンであり、メッセージ 5 0 6 は新しい方のバージョン (5 0 3 B) によってのみ処理することができる。1 つのバージョンが着信メッセージを処理することができればよく、他方のサービス・バージョンができないことには、多くの異なる理由があると考えられる。その理由には関係なく、コンピュータ・システム 1 0 1 は、メッセージ 5 0 6 をサービスの第 1 バージョン 5 0 3 A からサービスの第 2 バージョン 5 0 3 B に渡すべきことを決定することができる。次いで、メッセージ 5 0 6 を処理するために、サービスの第 2 バージョンを使用することができる (3 4 0)。

30

40

【 0 0 4 1 】

[0048] 一実施形態では、第 1 バージョンが誤動作したときに、メッセージ 5 0 6 をサービスの第 2 バージョンに転送することができる。つまり、サービスの第 2 バージョン 5 0 3 B は、第 1 サービス・バージョンに対するフェイルオーバーを設ける。このようなフェイルオーバーのシナリオでは、第 2 バージョンは第 1 バージョンよりも古くても新しくてもよい。ある場合には、これらのサービス・バージョンの一方 (例えば、バージョン 1 (5 0 3 A)) を、インストール時に自動的に実装することもできる。つまり、インスト

50

ールが終了すると直ぐに、着信メッセージ 5 0 6 を処理するために、新たにインストールされたサービス・バージョンを自動的に使用することができる。したがって、サービスをアップグレードし、アップグレードの後には自動的に実装し（例えば、サービス・アップグレード・モジュール 5 0 4 によって）、アップグレードのダウンタイムを大幅に短縮するか、または解消する。ある場合には、新たにアップグレードされたものが実行していると一旦判定されたなら、他方（例えば、第 2 ）のサービス・バージョンは自動的にアンインストールされる。

【 0 0 4 2 】

[0049] 更にまた、着信メッセージをサービスの第 1 バージョン 5 0 3 A から第 2 バージョン 5 0 3 B に処理のために送るべきか否かについての判定は、種々の論理的評価に基づくこともできる。例えば、第 1 バージョンがメッセージに応答するためには 2 秒以上かかる場合、コンピュータ・システム 1 0 1 は、第 1 バージョンは利用できないと判定することができる、次いでメッセージを同じサービスの第 2 バージョンに発送し直すことができる。他のロジックも同様に構成し実施することができる。以上は、論理的判定の一例に過ぎない。加えてまたは代わりに、メッセージ 5 0 6 を処理のためにサービスの第 1 バージョン 5 0 3 A から第 2 バージョン 5 0 3 B に送るべきか否かについての判定は、ユーザまたはシステム・アドミニストレータによって設定される構成設定 5 0 5 に基づくこともできる。

【 0 0 4 3 】

[0050] ある場合には、メッセージを第 1 バージョンから第 2 バージョンに転送することができ、第 1 および第 2 バージョンの双方によって同時に処理することもできる。このような場合、同時に実行するサービスの第 1 および第 2 バージョンの出力を比較して、処理の間に発生するエラーを発見することもできる。このようにして、サービス・プロバイダが、アプリケーションの双方のバージョンを検査し、どのバージョンが最も少ないエラーでメッセージ 5 0 6 を処理するか判定することができるのでよい。第 2（新しい方、恐らくベータ）バージョンをインストールし、現在のバージョンと共に実行することもでき、プログラム開発者が新しい方のバージョンの結果を記録し、危険性を全く高めることなく、新しい方のバージョンを検査することになる。加えてまたは代わりに、サービス・プロバイダ（または他のユーザ）が、どのバージョンがどのエラーを発生したか判定できるであろう。このデータを使用して、サービス・プロバイダは、着信メッセージのタイプ毎に、どのサービス・バージョンが最良であるか指定することができるであろう。この指示は、着信メッセージを第 2 バージョン 5 0 3 B に転送するか、またはそれらを第 1 バージョン 5 0 3 A に留めるかの判定において、コンピュータ・システム 5 0 1 によって使用することができる。このように、コンピュータ・システム 1 0 1 は、受信するメッセージ毎に、どのサービス・バージョンを使用すべきか（または、使用するのが最良であるか）継続的に判定することができる。

【 0 0 4 4 】

[0051] 図 4 は、外部システム統合サービスを実現するための方法 4 0 0 のフロー・チャートを示す。これより、図 1 の環境 1 0 0 および図 6 の環境 6 0 0 のコンポーネントおよびデータを頻繁に参照しながら、方法 4 0 0 について説明する。

【 0 0 4 5 】

[0052] 方法 4 0 0 は、外部メッセージ・ブローカ・サービス 6 0 3 が設置されていることを判定するブロック（ 4 1 0 ）を含む。このメッセージ・ブローカ・サービスは、サービス間（例えば、パブリッシャ 6 0 6 および / またはサブスクライバ 6 0 7 ）における通信を可能にする 1 つ以上のメッセージ・キュー 6 0 4 を維持するように構成され、メッセージ・キューは、パブリッシャ 6 0 6 からメッセージを受信し、メッセージ 6 0 5 A / 6 0 5 B をサブスクライバ 6 0 7 に転送するように構成される。メッセージは、サブスクライブ(subscription)に基づいて、および / またはトピック・キー（図 7 に示すような）に基づいて、発送することができる。内部メッセージ・ブローカ・サービス 6 0 3 をインスタンス化するために、コンピュータ・システム 6 0 1 のサービス・インスタンス化モ

ジュール 6 0 2 を使用することもできる。内部メッセージ・ブローカ・サービスは、図 1 において説明したメッセージ・ブローカ・サービス 1 0 2 と実質的に同じでよい。また、メッセージ・キューも、図 1 において説明したメッセージ・キュー 1 0 3 と同様または同じでよい。メッセージは、製品を生産している機械についての情報を含むことができ、動作ステータス・メッセージ、計画ステータス・メッセージ、エラー・メッセージ、または他のタイプのメッセージを含む。

【 0 0 4 6 】

[0053] 更に、方法 4 0 0 は、内部メッセージ・ブローカ・サービス 6 0 3 と合同で、外部システム統合サービス 6 0 8 をインスタンス化するブロック (4 2 0) も含む。外部システム統合サービス 6 0 8 は、指定された、登録メッセージ 6 0 5 A / 6 0 5 B にサブスクライブし、登録メッセージ 6 0 5 C を 1 つ以上の外部エンティティ 6 1 1 に転送するように構成される。外部エンティティ 6 1 1 は、外部ユーザ、会社、コンピュータ・システム、データ・ストア、あるいはコンピュータ・システム 6 0 1 に対して外部である、または内部メッセージ・ブローカ・サービス 6 0 3 を動作させているところに対して外部である他のエンティティであってもよい。外部システム統合サービス 6 0 8 は、サービス・インスタンス化モジュール 6 0 2 によってインスタンス化することができ、パブリッシャ 6 0 6 とサブスクライバ 6 0 7 との間で転送されるメッセージを待ち受けるように構成することができる (例えば、図 1 のサービス A、B、または C (1 0 5 A ~ 1 0 5 C の内任意のもの))。

【 0 0 4 7 】

[0054] 次いで、外部システム統合サービス 6 0 8 は、メッセージの内どれを、指定された外部エンティティに送るべきか決定することができる。例えば、外部エンティティが、任意のエラー・メッセージ、またはエラーが発生したことを示す任意の動作ステータス・メッセージを受信したいことを指定することもできる。これらのメッセージは、「登録済み」メッセージ、即ち、外部エンティティが受信するためにサブスクライブした(subscribe)または登録した(register)メッセージであろう。外部システム統合サービスがサブスクライブされたメッセージは、外部システム統合サービスによって構成されても、または外部システム統合サービス内部から選択されても、外部システム統合サービスを使用している外部システム内部から選択されても、またはコンピュータ・システム 1 0 1 内部から選択されてもよい。このような登録済みメッセージが外部システム統合サービスによって受信されると (4 3 0)、外部システム統合サービス 6 0 8 は、受信した登録済みメッセージ 6 0 5 C を、種々の指定された外部エンティティに転送することができる (4 4 0)。

【 0 0 4 8 】

[0055] また、外部システム統合サービスは、指定された外部システム通信プロトコルまたはフォーマットへのメッセージの変換も処理するために使用することもできる。例えば、外部システムがある種のファイル・タイプまたはプロトコルを使用して通信する場合、外部システム統合サービス 6 0 8 は、内部メッセージをある種のフォーマット (例えば、コンマ分離フォーマット) に変換し(translate)、次いで指定されたプロトコルを使用してこのファイルを送信することができる。各外部エンティティ 6 1 1 は、異なるタイプのメッセージ、またはある種のパブリッシャからのメッセージ、あるいは双方を要求することもできる。外部システム統合サービス 6 0 8 は、メッセージング・キュー 6 0 4 を使用して、サービスの内任意のものと相互作用することができる。外部システム統合サービスは、更に、外部エンティティ 6 1 1 に利用可能にしたメソッドも発行することができる。

【 0 0 4 9 】

[0056] ある実施形態では、制御システム 6 0 9 が、サービス・インスタンス化モジュール 6 0 2 によって、サービスとしてインスタンス化されることも可能である。制御システムは、設置された内部メッセージ・ブローカ・サービス 6 0 3 および / または外部システム統合サービスと合同して、サービスとしてインスタンス化されてもよい。制御システ

ムは、１つ以上の指定された機械 6 1 0 のために機械機能を動作させるように構成することもできる。例えば、制御システムは、機械をいつオンまたはオフに切り替えるか、いつ機械がある種のジョブを処理すべきか、どのように機械がジョブを処理すべきか（ジョブ毎に構成設定値を含む）制御する、または他の機械動作を制御するように構成することができる。機械 6 1 0 は、コンピュータ・システム 6 0 1 に対してローカルでもリモートでもよく、付加的に手動制御によって制御されてもよい。

【 0 0 5 0 】

[0057] サービス・インスタンス化モジュール 6 0 2 は、少なくともある実施形態では、メッセージ・パイプラインング・サービス(message pipelining service)もインスタンス化することができる。メッセージ・パイプラインング・サービスは、指定された順序で処理されることになっているメッセージをメッセージ・パイプラインにおいて受信するように構成することができる。メッセージ・パイプラインング・サービスは、受信したメッセージを発行し、そのメッセージの結果にサブスクライブする。例えば、メッセージが機械コマンド（例えば、機械 6 1 0 に対する）である場合、メッセージ・パイプラインング・サービスはそのコマンドにサブスクライブし、このコマンドの結果として発生したことの結果を受信する。これらのメッセージ結果は、次いで、パイプラインにおいて以後受信するメッセージ毎に、入力引数として追加することができる。メッセージ・パイプラインング・サービスは、メッセージ・パイプラインにおける最後のメッセージを受信したと判断し、次いでメッセージ・パイプラインの結果をサブスクライバ 6 0 7 に発行する。このように、コールする側のサービス(calling service)が、任意の中間ステップをパイプライン・サービスに移譲することができる。

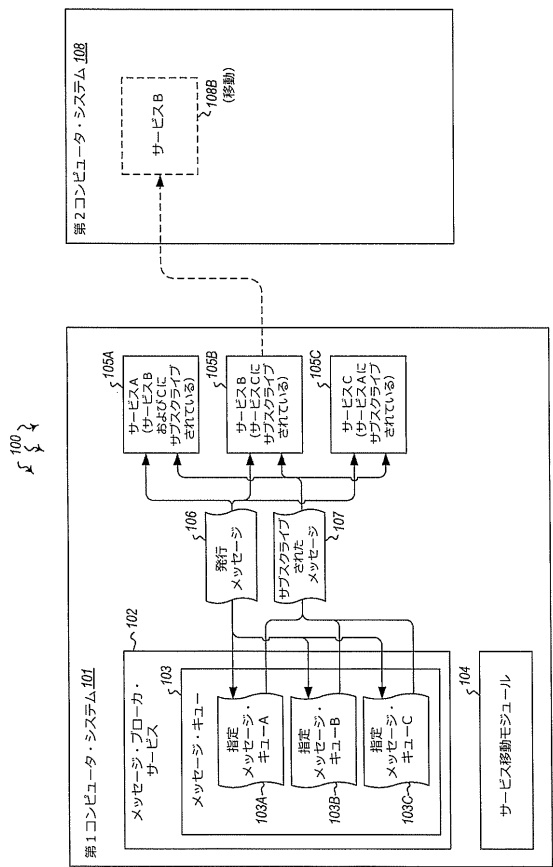
【 0 0 5 1 】

[0058] 以上によれば、サービスをスケーリングする、および／または複数のコンピュータ・システムにわたって移動させることを可能にする方法、システム、およびコンピュータ・プログラム製品を提供する。更に、１つのサービス・バージョンから他のバージョンに移行し、外部エンティティがサブスクライバによって発行されたメッセージを受信することを可能にする外部システム統合サービスを実装する方法、システム、およびコンピュータ・プログラム製品を提供する。

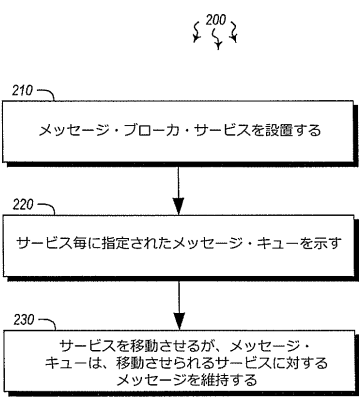
【 0 0 5 2 】

[0059] 本明細書において説明した概念および特徴は、それらの主旨や記述的特性(descriptive characteristics)から逸脱することなく、他の特定形態においても具体化することができる。説明した実施形態は、あらゆる観点において、限定ではなく例示として見なされるものとする。したがって、本開示の範囲は、以上の説明によってではなく、添付した請求項によって示される。請求項の均等の意味および範囲に該当するあらゆる変更は、その範囲内に包含されるものとする。

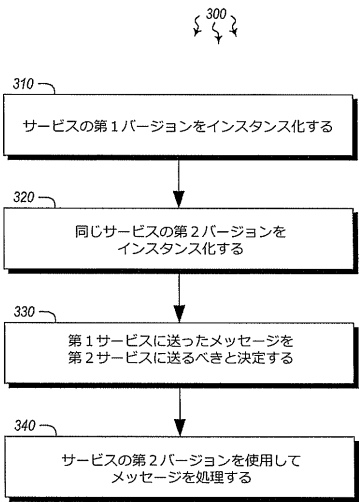
【図 1】



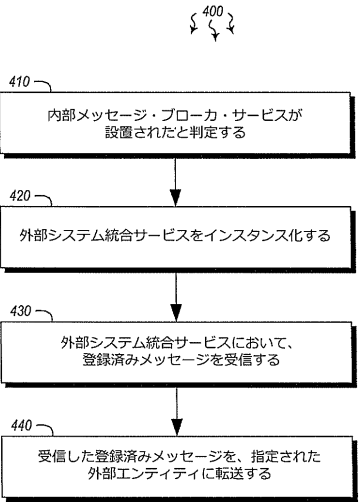
【図 2】



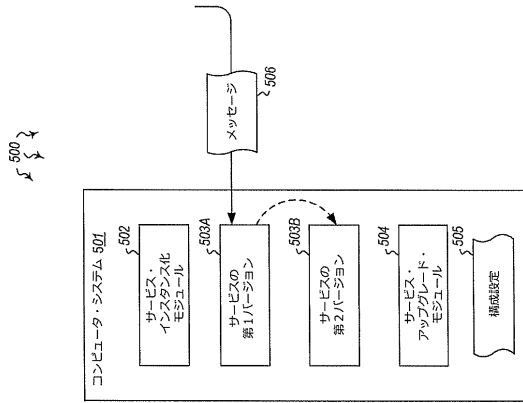
【図 3】



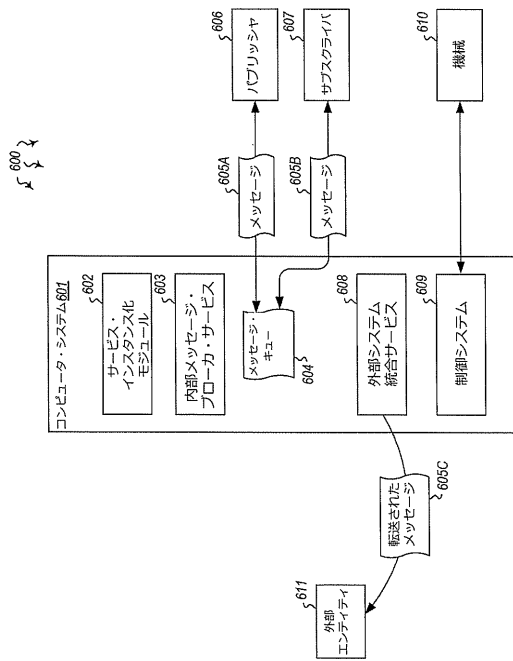
【図 4】



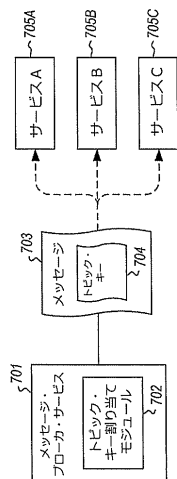
【図 5】



【図 6】



【図 7】



フロントページの続き

- (72)発明者 カールソン, ステファン
アメリカ合衆国ユタ州 8 4 1 2 1, ソルト・レイク・シティ, サウス・ワサッチ・ブルーバード
6 4 4 0
- (72)発明者 ハーネスク, アンドレアス
アメリカ合衆国ユタ州 8 4 1 2 1, ソルト・レイク・シティ, サウス・ワサッチ・ブルーバード
6 4 4 0

審査官 清木 泰

- (56)参考文献 特開 2 0 1 1 - 1 4 1 6 9 5 (J P , A)
国際公開第 2 0 1 0 / 1 1 6 6 7 6 (W O , A 1)
特開 2 0 0 9 - 2 2 3 4 0 3 (J P , A)
特表 2 0 0 6 - 5 2 3 9 0 5 (J P , A)
特開 2 0 0 5 - 0 0 4 6 7 6 (J P , A)
特開 2 0 0 0 - 2 5 9 5 8 1 (J P , A)
特表平 1 0 - 5 0 0 5 0 9 (J P , A)
Andrew S. Tanenbaum, Maarten van Steen, 水野忠則他訳, 分散システム 原理とパラダイム,
株式会社ピアソン・エデュケーション, 2 0 0 9 年 1 月 2 5 日, 第 2 版, Pages:150~154

(58)調査した分野(Int.Cl., D B 名)

G 0 6 F 9 / 4 5 5 - 9 / 5 4
G 0 6 F 1 5 / 1 6 - 1 5 / 1 7 7
G 0 6 F 1 3 / 0 0
H 0 4 L 1 2 / 0 0 - 1 2 / 2 6
H 0 4 L 1 2 / 5 0 - 1 2 / 9 5 5
G 0 6 F 1 1 / 1 6 - 1 1 / 2 0
B 6 0 W 1 0 / 0 0 - 1 0 / 3 0
B 6 0 W 3 0 / 0 0 - 5 0 / 1 6
G 0 6 F 9 / 4 4 - 9 / 4 4 5
G 0 6 F 8 / 0 0 - 8 / 7 7
G 0 6 F 1 1 / 3 6