

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号  
特許第5988621号  
(P5988621)

(45) 発行日 平成28年9月7日(2016.9.7)

(24) 登録日 平成28年8月19日(2016.8.19)

(51) Int.Cl.

G O 6 F 9 / 5 0 ( 2 0 0 6 . 0 1 )

F I

G O 6 F 9 / 4 6 4 6 5 Z

請求項の数 16 外国語出願 (全 27 頁)

(21) 出願番号	特願2012-50170 (P2012-50170)	(73) 特許権者	300015447
(22) 出願日	平成24年3月7日 (2012.3.7)		エスアーペー エスエー
(65) 公開番号	特開2012-230667 (P2012-230667A)		ドイツ連邦共和国, 69190 ハルドル
(43) 公開日	平成24年11月22日 (2012.11.22)		フ, ディートマルーホップーアレー 16
審査請求日	平成27年2月2日 (2015.2.2)		D i e t m a r - H o p p - A l l e e
(31) 優先権主張番号	13/094, 366		16, 69190 Walldorf,
(32) 優先日	平成23年4月26日 (2011.4.26)		Germany
(33) 優先権主張国	米国 (US)	(74) 代理人	100108453
			弁理士 村山 靖彦
		(74) 代理人	100064908
			弁理士 志賀 正武
		(74) 代理人	100089037
			弁理士 渡邊 隆
		(74) 代理人	100110364
			弁理士 実広 信哉

最終頁に続く

(54) 【発明の名称】 高負荷のビジネスプロセスのスケラビリティ

(57) 【特許請求の範囲】

【請求項 1】

イベントをメッセージとしてビジネスプロセスインスタンスに発行するための第1コンピュータノードの1つまたは複数のプロセッサによって遂行されるコンピュータ実装方法であって、

第1ビジネスプロセスインスタンスを実行する前記第1コンピュータノードでメッセージを受信するステップと、

前記メッセージに関連した第2ビジネスプロセスインスタンスを識別するステップであって、前記第2ビジネスプロセスインスタンスが前記メッセージを処理するために割り当てられる、ステップと、

前記第2ビジネスプロセスインスタンスが前記第1コンピュータノードではなく、前記第1コンピュータノード以外の第2コンピュータノードに配置される場合、前記第2ビジネスプロセスインスタンスによる検索のために前記メッセージをメッセージングキューに送信するステップと、

前記第2ビジネスプロセスインスタンスが第1コンピュータノードに配置される場合、前記メッセージングキューの前記メッセージを破棄して、前記第2ビジネスプロセスインスタンスに前記メッセージを提供するステップとを備えることを特徴とする方法。

【請求項 2】

前記第2ビジネスプロセスインスタンスをホストする前記第2コンピュータノードに、前記メッセージが前記メッセージングキューから検索されうること示す通知メッセージを

送信するステップをさらに備え、

任意で、前記第1コンピュータノードおよび前記第2コンピュータノードは、ビジネスプロセス管理スイートで利用されるコンピューティングデバイスであることを特徴とする請求項1に記載の方法。

【請求項3】

前記メッセージングキューは、前記第2ビジネスプロセスインスタンスのプロセスインスタンス識別子に基づいて識別されることを特徴とする請求項1または2に記載の方法。

【請求項4】

前記第1コンピュータノードで前記メッセージを受信するステップは、クラウドネットワークで受信されたメッセージを分配するロードバランサから前記メッセージを受信することを備えることを特徴とする請求項1～3のいずれか一項に記載の方法。

10

【請求項5】

前記メッセージに関連した前記第2ビジネスプロセスインスタンスを識別するステップは、前記メッセージに関連した複数のビジネスプロセスインスタンスを識別することをさらに備えることを特徴とする請求項1～4のいずれか一項に記載の方法。

【請求項6】

前記メッセージが共有される状態変数に対する変更を示すかどうかを判定するステップと

任意で、前記メッセージが前記共有される状態変数に対する変更を示す場合、前記共有される状態変数上のロックを取得するステップとをさらに備え、

20

前記共有される状態変数上の前記ロックを取得するステップは、他のコンポーネントおよび前記第2ビジネスプロセスインスタンス以外のビジネスプロセスインスタンスが前記共有される状態変数にアクセスすることを防ぐことを任意で備えることを特徴とする請求項1～5のいずれか一項に記載の方法。

【請求項7】

前記第2コンピュータノードから前記メッセージングキューへ開始されたポーリング要求に基づいて、検索のために前記メッセージングキューの中のメッセージを識別するステップと、

前記メッセージングキューから前記メッセージを検索し、キューから取り出すステップと、

30

前記メッセージに関連したビジネスプロセスインスタンスを使用して、前記第2コンピュータノードにおいて前記メッセージを処理するステップとをさらに備えることを特徴とする請求項1～6のいずれか一項に記載の方法。

【請求項8】

前記メッセージングキューへの複数の前記ポーリング要求は、前記第2コンピュータノードによって処理されるために割り当てられた着信メッセージが受信されたかどうかを判定するために、特定の間隔で開始されることを特徴とする請求項7に記載の方法。

【請求項9】

前記特定の間隔は、前記ビジネスプロセスインスタンスに関連したコンテキストに基づいて調整されるように動作可能であることを特徴とする請求項8に記載の方法。

40

【請求項10】

前記メッセージングキューの中の前記メッセージの利用可能性を示す通知が受信される場合、即座のポーリング要求が前記メッセージングキューに送信され、前記即座のポーリング要求は、その後の周期的な要求が前記特定の間隔で送信される前に送信されることを特徴とする請求項8に記載の方法。

【請求項11】

前記メッセージングキューから前記メッセージを検索する前に、前記ビジネスプロセスインスタンスに関連した共有される状態変数上のロックを取得するステップをさらに備え、

前記共有される状態変数上の前記ロックを取得するステップは、他のコンポーネントま

50

たは前記メッセージに関連した前記ビジネスプロセスインスタンス以外のビジネスプロセスインスタンスが前記共有される状態変数にアクセスすることを防ぐことを任意で備えることを特徴とする請求項 7 ~ 10 のいずれか一項に記載の方法。

【請求項 12】

請求項 1 ~ 11 のいずれか一項に記載の方法を前記第1コンピュータノードの前記1つまたは複数のプロセッサに遂行させるためのコンピュータ可読命令を備えることを特徴とする非一時的な有形のストレージ媒体上の符号化されたコンピュータプログラム。

【請求項 13】

システムで受信されたメッセージを保存するように動作可能なメモリであって、前記システムは第1ビジネスプロセスインスタンスを実行する第1コンピュータノードである、メモリと、

10

前記メッセージに関連した第2ビジネスプロセスインスタンスを識別し、前記第2ビジネスプロセスインスタンスは前記メッセージを処理するために割り当てられ、前記第2ビジネスプロセスインスタンスが前記第1コンピュータノードではなく、前記第1コンピュータノード以外の第2コンピュータノードに配置される場合、前記第2ビジネスプロセスインスタンスによる検索のために、前記メッセージをメッセージングキューに送信し、前記第2ビジネスプロセスインスタンスが第1コンピュータノードに配置される場合、前記メッセージングキューの前記メッセージを破棄して、前記第2ビジネスプロセスインスタンスに前記メッセージを提供するように動作可能な1つまたは複数のプロセッサとを備えることを特徴とするシステム。

20

【請求項 14】

前記1つまたは複数のプロセッサは、前記第2ビジネスプロセスインスタンスをホストする前記第2コンピュータノードに、前記メッセージが前記メッセージングキューから検索されることが可能であることを示す通知メッセージを送信するようにさらに動作可能であることを特徴とする請求項 13 に記載のシステム。

【請求項 15】

前記メッセージに関連した前記第2ビジネスプロセスインスタンスを識別するステップは、前記メッセージに関連した複数のビジネスプロセスインスタンスを識別することをさらに備えることを特徴とする請求項 13 または 14 に記載のシステム。

【請求項 16】

30

前記1つまたは複数のプロセッサは、前記第2ビジネスプロセスインスタンスに関連した共有される状態変数上のロックを取得するように動作可能であり、前記共有される状態変数上の前記ロックを取得するステップは、他のコンポーネントおよび前記第2ビジネスプロセスインスタンス以外のビジネスプロセスインスタンスが前記共有される状態変数にアクセスすることを防ぐことを備えることを特徴とする請求項 13 から 15 のいずれか一項に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、高負荷のビジネスプロセスのスケラビリティを提供するためのソフトウェア、コンピュータシステムおよびコンピュータ実装方法に関する。

40

【背景技術】

【0002】

高帯域ネットワークおよびデータ接続、ならびに大容量のデータストレージサーバの使用の増加は、クラウドコンピューティングソリューションなどの様々な展開モデルの実装をもたらした。クラウドコンピューティングソリューションでは、資源、サービス、強化機能またはソフトウェアは、ネットワークの間でクライアントコンピュータに提供されることが可能である。資源は、資源利用およびスケリング効果の向上を達成するために、仮想化技術を通じて多数のクライアントの間で共有されることが可能である。クラウドコンピューティングモデルはまた、データの共有アクセスおよび遠隔ストレージをユーザに

50

提供するために使用されることも可能である。クラウドコンピューティングソリューションでは、コンピューティング資源はインターネットなどのネットワークの間にホストされたサービスとして提供される。これらのサービスは、クライアントコンピュータへアプリケーションまたはソフトウェアをインストールすることなく、クラウドコンピューティングネットワークを介して提供されるオンデマンドのサービスを含むことが可能である。

#### 【0003】

企業は、中心となるビジネスプロセスおよび複雑な反復作業をモデル化、記録、自動化、統治、最適化、シミュレーションおよび監視するためにビジネスプロセス管理スイート(business process management suites)(BPMS)を採用する。いくつかの例では、オンデマンドのBPMSは、追加の作業付加を処理するために追加のクラウドインスタンス(コンピュータノード)を動的に割り当てることによって、スケーラビリティまたは弾力性を実現する。同時にクラウドベースのBPMSは、モバイルデバイス上で動作するクライアントソフトウェア、(例えば企業の資源計画システムなどの)店舗用ビジネスソフトウェア設備、ウェブベースのクライアント、その他のクラウドベースのビジネスソフトウェア、およびビジネスパートナーによって遂行される他のソフトウェアを含めて、広範な他のソフトウェアコンポーネントに接続される。BPMSシステムの中のビジネスプロセスは、それらの外部のソフトウェアコンポーネントとイベントを交換することができる。

#### 【発明の概要】

#### 【課題を解決するための手段】

#### 【0004】

本開示は、クラウドベースのインフラストラクチャで高負荷のビジネスプロセスのスケーラビリティを提供するための技術について述べる。1つまたは複数のプロセッサにオペレーションを遂行させるためのコンピュータ可読媒体を備えるコンピュータプログラム製品は、有形のストレージ媒体で符号化される。これらのオペレーションは、第1ビジネスプロセスインスタンスを遂行する第1コンピュータノードでメッセージを受信することを含むことが可能である。このメッセージに関連した第2ビジネスプロセスインスタンスが識別される。このメッセージは、第2ビジネスプロセスインスタンスが第1コンピュータノードに配置されていない場合に、第2ビジネスプロセスインスタンスによって検索されるためにメッセージングキューに送信される。

#### 【0005】

各々のデータを処理し、変換する有形の非一時的な媒体として具体化されるコンピュータ実装のソフトウェアとして一般に説明されているが、態様の一部またはすべてはコンピュータ実装方法であってもよく、またはさらに、この説明される機能を遂行するための各々のシステムまたはその他デバイスの中に含まれてもよい。これらおよびその他の態様の詳細、ならびに本開示の実行形態は、添付の図面および以下の説明の中で明記される。本開示のその他の特徴、目的および利点は、説明および図面、ならびに特許請求の範囲から明らかとなる。

#### 【図面の簡単な説明】

#### 【0006】

【図1】クラウドネットワークの中の分散型ビジネスプロセス管理スイートのための環境例を描く図である。

【図2】コンピュータノードおよびメッセージングシステムに含まれるコンポーネント例の略図である。

【図3】図2で説明されるシステムなどの適切なシステムを使用する、イベントをプロセスインスタンスに発行するためのプロセスの流れ図である。

【図4】図2で説明されるシステムなどの適切なシステムを使用する、外部コンポーネントからコンピュータノードで受信されたメッセージを処理するためのプロセスの流れ図である。

【図5】図2で説明されるシステムなどの適切なシステムを使用する、メッセージングキューから関連メッセージを検索するためのプロセスの流れ図である。

【図6】図2で説明されるシステムなどの適切なシステムを使用する、メッセージを受信し、そのメッセージをビジネスプロセスインスタンスに分配するためのビジネスプロセス例の略図である。

【図7】図2で説明されるシステムなどの適切なシステムを使用する、メッセージをクラウドインスタンスへ発行するためのプロセス例を描く略図である。

【図8】図2で説明されるシステムなどの適切なシステムを使用する、メッセージをクラウドインスタンスへ発行するためのプロセス例を描く略図である。

【発明を実施するための形態】

【0007】

本開示は一般に、クラウドベースのインフラストラクチャで高負荷のビジネスプロセスのスケラビリティを提供するためのコンピュータシステム、ソフトウェアおよびコンピュータ実装方法について説明する。クラウドコンピューティングまたはクラスタノードのインフラストラクチャでは、アプリケーションまたはサービスを外部のコンポーネントおよびユーザに提供するために多数のコンピュータノードまたはクラウドインスタンスが使用されることが可能である。第1クラウドまたはクラスタノードインスタンス(以下、「クラウドインスタンス」と呼ばれる)で受信されたイベントは、処理のために、第2の受信側クラウドインスタンスに転送される必要があってもよい。イベントは、アプリケーションおよび/またはビジネスプロセスの間で交換されるメッセージまたは要求である。イベントの処理を開始するために直ちに受信側クラウドインスタンスと通信する代わりに、イベントはデータベースに支援されるイベントキューの中に保存される。次いで、受信側クラウドインスタンスは、イベントの消費のために局所的に動作するプロセスインスタンスへ発行するために、イベントキューからイベントを検索してもよい。いくつかの実装では、受信側クラウドインスタンスへの通知コールは、受信側クラウドインスタンスが遅延を伴わずにイベントを検索するように動作させる。受信側プロセスインスタンスは、その内部状態に基づいて、適切な時間にイベントを消費することができる。

【0008】

ビジネスプロセスは、他のコンポーネントとプロセスを同期させる自動化アクティビティ、ユーザタスク、およびイベントなどのプロセスステップを統合するビジネスプロセス管理スイート(BPMS)によって遂行される。これらのプロセスは、しばしば外部のアプリケーションおよびデバイスと交信してもよい。例えば、ユーザタスクはユーザのモバイルデバイスに送信されてもよく、そこでユーザタスクは処理され、基盤となるビジネスプロセスにデータを返送する。他の例では、RFIDリーダはビジネスプロセスに信号を送信してもよく、そこで信号は、特定のフォローアップ行為を引き起こすためにイベントの中で消費される。他の例では、ビジネスプロセスは、自動化アクティビティから企業資源計画(ERP)システムを呼び出して、そこで管理される(例えば、請求書または材料のマスタデータなどの)ビジネスオブジェクトを変更する。

【0009】

いくつかの実装では、BPMSシステムは、オンデマンドのビジネスアプリケーションを支援し、クラウドインフラストラクチャの本質的な弾力性とスケラビリティ特性が伴う総所有コストの低さから恩恵を受けるために、クラウドコンピューティングネットワークの中でオンデマンドの設備として提供されることが可能である。技術的には、単一のBPMS設備は、基盤となるクラウドインフラストラクチャによって与えられるコンピュータ「ノード」のダイナミックレンジの間で分散される。これらのノードは一緒に、複数のビジネスプロセスを実行する。ノードの数は、一度に処理されるべき作業負荷がより多く存在する場合には常に増加し、また処理されるべき作業負荷がより少ない場合には常に縮小してもよい。ノードのうちのいくつかは、特定のビジネスプロセスの異なるインスタンスを実行してもよい一方で、他のノードは様々な実装の中で全く異なるビジネスプロセスを実行してもよい。

【0010】

図示された例を見ると、図1では、クラウドベースのインフラストラクチャの中でビジ

10

20

30

40

50

ネスプロセス管理スイート(BPMS)に関連したビジネスプロセスを遂行するための環境例100が図示されている。図に示された環境100は、クラウドネットワーク105などのネットワークの中で複数のコンポーネントを含むか、またはそれらに通信可能に接続される。一般に環境100は、モバイルデバイス180またはクライアント171などの外部アプリケーションおよびデバイスと同期して、クラウドネットワーク105内の自動化アクティビティ、ユーザタスクおよびイベントなどのプロセスステップを統合することができるシステムの構成例を示す。BPMSは、ノード110、120などを含めて、クラウドネットワーク105の中の多数のコンピュータノードのまで分散されることが可能である。本開示の中で使用する際、「コンピュータノード」および「クラウドインスタンス」という用語は、この開示の範囲から逸脱することなく、必要に応じて交互に使用されてもよい。クラスタコンピューティング環境(不図示)では、「コンピュータノード」および「クラウドインスタンス」という用語もまた「クラスタノード」と類似のものであってもよい。

10

#### 【0011】

ネットワーク105の中の各コンピュータノードは、いくつかのビジネスプロセスまたはプロセスインスタンスを実行するために必要とされる複数の異なるコンポーネントを含むことが可能である。例えば、図1に示すように、コンピュータノードはビジネスプロセス管理(BPM)ランタイム環境、メッセージングミドルウェアまたは通信アダプタを含むことが可能である。コンピュータノードの内部コンポーネントによって、コンピュータノードはBPMSに関連したプロセスステップを遂行し、他のコンピュータノードまたは外部コンポーネントと通信し、外部コンポーネントからのイベントを受信して応答し、ビジネスプロセスを実行することができる。クラウドコンピューティング環境の中でのBPMSの実装は、追加の作業負荷を処理するために必要なときに追加のコンピュータノードを割り当てることによって、BPMSに柔軟性およびスケーラビリティを提供する。

20

#### 【0012】

図1に示すように、クラウドベースのBPMSはまた、オンデマンドのサービスを外部のソフトウェアコンポーネントに提供するために、他の外部ソフトウェアコンポーネントに接続されることも可能である。例えばBPMSは、1つまたは複数のクライアント171、モバイルデバイス180、店舗用システム190およびその他のビジネスパートナーシステム192を含む外部のコンポーネントに接続されることが可能である。外部コンポーネントは、クラウドネットワーク105を介してBPMSと交信するクライアントソフトウェアコンポーネントを実行することができる。コンピュータノード110および120で動作するビジネスプロセスは、イベントを外部ソフトウェアコンポーネントと交換することができる。さらに、クラウドネットワーク105はまた、クラウドネットワーク105の中の通信を管理し、同期させるためのインターフェース140および/またはコンピュータノード間の作業負荷の分配を管理するためのロードバランサ150など、外部コンポーネントとコンピュータノードの間の通信を促進するためのコンポーネントも含むことができる。典型的ロードバランサ150は、作業負荷全体を固定サイズより小さな作業パッケージに分割してから、利用可能な作業プロセスに作業パッケージを割り当てるために使用されることが可能である。一般に、典型的ロードバランサ150はメッセージを受信して、そのメッセージを、必ずしも受信されたメッセージに関連した特定のノードまたは受信プロセスインスタンスが実行中のノードではないが、利用可能なノードに分配する。

30

40

#### 【0013】

ビジネスプロセスと外部コンポーネントの間のイベントの交換は、ビジネスプロセスが、それらの内部状態を受信されるイベントに一貫して同期させることが必要な場合がある。イベントがビジネスプロセスによって受信される場合、ビジネスプロセスは、ビジネスプロセスの制御フローおよびデータフロー上で意図した効果を達成するために、イベントに確実に対応する必要がある。したがって、ビジネスプロセスの状態は、ビジネスプロセスの外部コンポーネントとの整合性を維持するために、トランザクショナルな方法で同期される必要がある。すなわち、ビジネスプロセスの状態は、時間内の任意の離散点でビジネスプロセスと交信する外部コンポーネントの状態を反映しなければならない。

50

## 【 0 0 1 4 】

いくつかの例では、ビジネスプロセスと外部コンポーネントは、状態の整合性を確実にするために、同期的に結合させることが可能である。例えば、2フェーズコミット等の専用の分散型トランザクションプロトコルは、BPMSと外部ソフトウェアコンポーネントなどの2つのビジネスアプリケーションを同期的に結合する。すなわち、両方のアプリケーションはそれらの個々の状態を異なるコンピュータノード上で保持し、(例えば、データベース上でのそれらの状態のスナップショットを保持することなどの)単一の論理トランザクション行為を同時に遂行する。しかしながら、異なるソフトウェアコンポーネントと異なるコンピュータノードを同期的に結合することは、高い作業負荷を処理するビジネスアプリケーションにとっては効率的ではなく、処理のスループットおよび待ち時間に関してサービスレベル契約(SLA)に準拠しなければならない場合がある。要求側のアプリケーションと同時に行為を遂行するように他のアプリケーションに要求することによって、他のアプリケーションの現在の利用可能性も、その基盤となるインフラストラクチャも考慮されることはない。実際には、他のアプリケーションは現在要求に応じることができない場合があるので、トランザクション全体が遅延される。この問題は、コンピュータノードが一度に多数の要求を処理しなければならない場合に悪化する。本質的に、同期結合に依存する分散型トランザクションプロトコルは、クラウドベースのインフラストラクチャに拡張されない。

10

## 【 0 0 1 5 】

同時結合を回避するために、確実な非同期プロトコルが用いられてもよい。非同期プロトコルは、イベントが最終的に配信されることだけを保証する非同期に分離される仕方である。外部ソフトウェアコンポーネントからビジネスプロセスイベントを渡してもよい。同様に、ビジネスプロセスもまた、このようにして外部ソフトウェアコンポーネントにイベントを返してもよい。この非同期プロトコルは、分散型トランザクションの遮断特性を回避する。しかしながら、これらのプロトコルは、ビジネスプロセスと外部ソフトウェアコンポーネントの間の緩やかな(loose)結合を必要とする。例えば、着信イベントの待機などの非同期機能は、明示的にビジネスプロセスにモデル化される必要がある。さらに(例えばERPシステムなどの)外部ソフトウェアコンポーネントは、イベントの受信側ソフトウェアコンポーネントが何であるのか、または(ビジネスプロセスインスタンスのような)受信側ソフトウェアコンポーネントが現在実行中であるのはどの特定のコンピュータノードであるのかを理解するように構成されていない場合がある。したがって、イベントを受信側ソフトウェアコンポーネントに発行するために、(BPMSまたは他のメッセージングミドルウェアの一部であってもよい)特定のイベント関連メカニズムが必要とされる。

20

30

## 【 0 0 1 6 】

特定の実装では、クラウドベースのBPMS実装におけるスケーラビリティの問題に対処するために、クラスタ実行可能プロトコルが使用されることが可能である。クラスタ実行可能プロトコルは、2つのコンピュータノード間でプロセスインスタンス全体を転送するための削除アルゴリズム(eviction algorithm)に依存することが可能である。具体的には、受信側プロセスインスタンスは、イベントが受信されたノードに転送される。いくつかの例では、ソフトウェアコンポーネントは第1コンピュータノード上で要求を発行し、一方で要求を受信すると想定される、影響を受けるプロセスインスタンスは、第2の異なるコンピュータノード上で現在実行中である。イベントとのトランザクショナルな同期を維持しながらイベントを処理するために、第1コンピュータノードに関連したプロセスインスタンスは第1コンピュータノードから削除され、第2の異なるコンピュータノードに移動されてもよい。

40

## 【 0 0 1 7 】

例えば図1に描くように、クライアント171の外部デバイスで実行中の(例えばタスク管理ソフトウェアなどの)外部ソフトウェアコンポーネントは、ノード110および120を含む多数のノードの間に分散されたBPMSにイベントを提出することができる。最初に、イベントはロードバランサ150によって受信されることが可能であり、ロードバランサ150はイベ

50

ントを送信するために、その管理下にあるノードのうちの1つを選択する。この例では、イベントはノード110の特定のプロセスインスタンスへ送信されるが、イベントの消費は異なるノード120で遂行されなければならない場合がある。クラスタ実行可能プロトコルに基づいて、BPMSは、ユーザタスクが完了されるのを待つ場合など、アイドル状態に達するためにノード110のプロセスインスタンスを待つ。アイドル状態の間、その状態情報を含むノード110のプロセスインスタンスはノード110から削除され、データベース上で保持される。次いでノード120は、データベースから状態情報を読み込み、ノード120のプロセスインスタンスを再開することによって、プロセスインスタンスを回復する。受信されたイベントはその後、効果的にプロセス状態を同期させるノード120のプロセスインスタンスに渡される。

10

#### 【0018】

クラスタ実行可能プロトコルは、特定の状況では待ち時間およびスループットの問題につながる可能性がある。最初に、BPMSの中のビジネスプロセスのパフォーマンスは、プロセスインスタンスが複雑な状態に関連している場合、悪影響を受けることがある。多くの顧客シナリオには、深くネストされたサブフロー呼び出しを用いる大型のプロセスモデルが伴う。実際には、クラスタ輸送においてデータベースで保持され、そこからフェッチされなければならないプロセス状態は極めて大きい場合があり、データベース上でかなりの負荷を生成する場合がある。さらに、特定の要因は頻繁なクラスタ輸送を生じさせる場合があり、このことはさらにシステム資源を占有する可能性がある。いくつかのビジネスプロセスモデルは、クラスタ輸送を引き起こす場合がある多くのアーティファクト(artifact)を含む。クラスタ輸送を引き起こす場合があるアーティファクトの例は(例えばユーザタスクなどの)人間の活動、中間メッセージキャッチイベント、タイマーイベント、および同期プロセスインターフェースへ応答を送信することである。一般に、各々のこれらのアーティファクトの発生は、クラスタの中でのプロセスインスタンスの輸送を引き起こす場合があり、このことはシステム資源にとってコストのかかるオペレーションとなりうる。第3に、クラスタプロトコルは、本質的な利用可能性の制約のためにスケーラビリティを制限するノード間の同期通信を使用する。

20

#### 【0019】

さらに、多くのプロセスモデルは、クラスタ輸送を遂行するための前提条件であるアイドル状態にめったに遭遇しない場合がある。いくつかのアーティファクトは、連続または並列ループ、(例えばERP企業サービスなどの)長時間実行されるサービスを起動する自動化アクティビティ、および任意で複雑な場合がある顧客に提供されるデータマッピング機能など、アイドル状態を阻害し、したがって予測不可能な方法で処理時間を消費する場合がある。アーティファクトが呼び出しスタックの任意のサブフローの中の並列分岐に存在する場合、それらは一時的に、プロセスがクラスタ輸送の一部として削除されることを阻害する場合がある。実際には、プロセスにイベントを配信するための要求は失敗して、後で繰り返されなければならない、このことはメッセージのスループットを妨げる可能性がある。

30

#### 【0020】

クラウドベースのインフラストラクチャでは、外部コンポーネントによって送信されるイベントは特定のクラウドインスタンスに到達することができ、一方でイベントを処理する受信側プロセスインスタンスは、他のクラウドインスタンスに存在してもよい。分散型クラウドインフラストラクチャの中で一貫してイベントを受信側ビジネスプロセスに発行するためのプロトコルが提供されることが可能である。いくつかの実装では、このプロトコルはコストの高いプロトコルオーバーヘッドを取り込まなくてもよく、イベントを受信するために「アイドル」であるビジネスプロセスに依存する必要はない。イベントの数またはプロセスインスタンスの数のいずれかが増える場合、プロセスの応答時間とプロセス全体のエンド対エンドのスループットの両方は、追加の作業負荷を処理するために追加のクラウドインスタンスを割り当てることによって、簡単に補償されることが可能である。さらに、イベントを集中データベースで保持することによって、I/Oおよびネットワーク

40

50

の負荷は、受信側プロセスインスタンスがクラウドネットワークの中で、クラスタ間で輸送されることを必要としないことから、減らされることが可能である。また、正常にイベントをBPMSランタイムに配信することに関わる待ち時間は、大幅に削減される。イベントはもはや、配信トランザクションを完了するために、受信側プロセスインスタンスがクラスタ間で輸送されるのを待つ必要はない。最後に、ノードクラスタ間で輸送されることが不可能なプロセスインスタンスは、もはやイベントが配信されるのを阻害または差し控えることができないことから、イベントの配信を失敗する可能性もまた実質的に減少する。

#### 【 0 0 2 1 】

本開示は、物理的にも非同期的にも、イベントの受信とビジネスプロセスでの消費を分離することによって、クラウドコンピューティングインフラストラクチャの中の高負荷の処理に関連した課題に対処する。すなわち、イベントは第1クラウドインスタンスで受信される場合、第2クラウドインスタンスで実行中であるかもしれない受信側ビジネスプロセスのために、データベースに支援されるイベントキューの中で保持される。受信側ビジネスプロセスを実行する第2クラウドインスタンスは、キューから新たに到着するイベントを定期的にフェッチし、それらをイベントが消費される局所的に実行中のプロセスインスタンスに発行する。いくつかの例では、第2クラウドインスタンスは、イベントキューのポーリングに基づいて新しいイベントをフェッチする。いくつかの実装では、一旦イベントがキューの中に入ると、任意の通知呼び出しは、アクティブに第2クラウドインスタンスを動作させ、ポーリングの遅延を除外する。

#### 【 0 0 2 2 】

受信側プロセスインスタンスは、第1クラウドインスタンスでイベントを発行したトランザクションを遮断することなく、その内部状態および利用可能性に基づいてイベントを自由に消費する。例えば、タスク管理ソフトウェアは自立的に、他のクラウドインスタンス上で実行中のビジネスプロセスによって作り出されたユーザタスクの状態を、「進行中」から「完了」に設定してもよい。実際には、イベントは、別々の非同期的に分離されたトランザクションの中で、それ自体のクラウドインスタンス上で取り上げられるべき影響を受けるプロセスのために、生成され保持される(すなわちそのプロセスインスタンスのために、キューに加えられる)。特定の例では、外部コンポーネントはロック競合に陥らずに、イベントをプロセスに送信してもよい。まれな例では、外部コンポーネントとビジネスプロセスの両方が結合状態変数にアクセスする場合、ロックは中央ロックプロバイダから取得される必要がある。ロックは、任意の状態の変化を外部ソフトウェアコンポーネントがイベントを発行する場合にのみ生成される別々のイベントエンティティにパッケージ化することによって、回避されることが可能である。しかしながら一般に、ビジネスプロセスは私的な資源を管理し、直接外部資源にアクセスすることはない一方で、外部コンポーネントは一般に内部のプロセス資源を操作することはない。

#### 【 0 0 2 3 】

図2は、クラウドベースのインフラストラクチャで高負荷のビジネスプロセスのスケラビリティを提供するためのコンピュータノード202およびメッセージングシステム222の中のコンポーネント例を示す環境200を描く。環境200は1つまたは複数の遠隔システム250とコンピュータノード202とメッセージングシステム222とを含み、それらのうちの少なくともいくつかは、ネットワーク212間で通信する。一般に、環境200は、外部コンポーネントから受信されるイベントを処理するためのBPMSの中で使用されるコンポーネントの構成例を示す。コンピュータノード202は、図1に関して上で述べたようなBPMS実装の中のノード例を表す。BPMS実装は1つ以上のノードを含むことが可能であり、各ノードは実装によって、より少ない、より多い、または異なるコンポーネントを含んでもよい。特定の例では、ノード202とメッセージングシステム222は、クラウドコンピューティングネットワーク内で論理的にグループ化され、アクセス可能である。したがってBPMSは、従来のサーバ/クライアントシステムまたは遠隔システム250の局所アプリケーションと同様に、クラウドコンピューティングネットワークを通じたオンデマンドのソリューションとして提供されてもよい。

## 【 0 0 2 4 】

一般に、ノード202は、環境200に関連したデータおよび情報を受信、送信、処理、保存または管理するように動作可能なサーバ等の任意の電子コンピューティングデバイスであることが可能である。ノード202は、1つまたは複数のビジネスプロセスアプリケーション232を保存するサーバであることが可能であり、そこでビジネスプロセスアプリケーションの少なくとも一部が、描かれた図2の環境200の中にあり、そこに通信可能に結合されたユーザまたはクライアントへ送信される要求および応答を介して実行される。いくつかの例では、ノード202は複数の様々なビジネスプロセスアプリケーション232を保存することが可能であり、一方で他の例では、ノード202は単一のビジネスプロセスアプリケーション232だけを保存して実行するための専用のサーバであることが可能である。いくつかの例では、ノード202はウェブサーバを備えるか、またはウェブサーバに通信可能に結合されることが可能であり、ここでビジネスプロセスアプリケーション232は、ビジネスプロセスアプリケーション232のプログラムされたタスクまたはオペレーションを遂行するために、遠隔システム250によってネットワーク212を介してアクセスされ、実行される1つまたは複数のウェブベースのアプリケーションを示す。

10

## 【 0 0 2 5 】

図2に描いたノード202は、環境200の遠隔システム250に関連した1つまたは複数のクライアントアプリケーションまたはビジネスアプリケーションからアプリケーション要求(すなわちイベント)を受信するために責任を負うことが可能であり、ビジネスプロセスアプリケーション232の中の前記要求を処理し、受信された要求が同期要求である場合には、ビジネスプロセスアプリケーション232からの適切な応答を要求側クライアントアプリケーションに返送することによって、受信された要求に応答する。ノード202はまた、メッセージングシステム222または図2に描かれていない他のノードなどのネットワーク212上の他のコンポーネントからの要求を受信し、それに応答してもよい。代替として、ノード202のビジネスプロセスアプリケーション232は、ユーザが局所的にアクセス中のノード202からの要求を処理し、それに応答することができてよい。したがって、図2に描いた遠隔システム250からの要求に加えて、ビジネスプロセスアプリケーション232に関連した要求もまた、内部のユーザ、外部または第三者の顧客、他の自動化アプリケーションの他に、任意のその他の適切なエンティティ、個人、システムまたはコンピュータからも送信されてよい。

20

30

## 【 0 0 2 6 】

本開示で使用される際、「コンピュータ」という用語は任意の適切な処理デバイスを含むことが意図されている。例えば、図2には1つのコンピュータを備える単一のノード202が描かれているが、環境200は、サーバープールを含むサーバ以外のコンピュータとともに、1つまたは複数のノードを使用して実装されることが可能である。実際に、ノード202、遠隔システム250およびメッセージングシステム222は、例えばブレードサーバ、汎用パーソナルコンピュータ(PC)、マッキントッシュ(Macintosh)、ワークステーション、UNIX(登録商標)ベースのワークステーションまたは任意のその他の適切なデバイスなどの任意のコンピュータまたは処理デバイスであることが可能である。すなわち、本開示は、従来のオペレーティングシステムを伴わないコンピュータと同様に、汎用コンピュータ以外のコンピュータを企図する。さらに、描かれているノード202、遠隔システム250およびメッセージングシステム222は、Linux(登録商標)、UNIX(登録商標)、Windows(登録商標)、Mac OSまたは任意のその他の適切なオペレーティングシステムを含む任意のオペレーティングシステムを実行するために適合されてもよい。

40

## 【 0 0 2 7 】

本実装では、図2に示すように、ノード202はプロセッサ208、インターフェース205、メモリ211および1つまたは複数のビジネスプロセスアプリケーション232を含む。インターフェース205は、(例えば、ネットワーク212に通信可能に結合された他のシステムと同様に、遠隔システム250などの)ネットワーク212に接続されたクライアントサーバまたは(環境200の内部を含む)その他の分散型環境の中で他のシステムと通信するために、ノード20

50

2によって使用される。一般に、インターフェース205は、適切な組み合わせのソフトウェアおよび/またはハードウェアの中で符号化され、ネットワーク212と通信するように動作可能なロジックを備える。より具体的には、インターフェース205は、ネットワーク212またはインターフェースのハードウェアが、描かれた環境200の内部および外側で物理信号と通信するように動作可能であるように、通信に関連した1つまたは複数の通信プロトコルを支援するソフトウェアを備えてもよい。

【0028】

いくつかの実装では、ノード202はまた、グラフィカルユーザインターフェース(GUI)などのユーザインターフェースを含んでもよい。GUIは、例えばサーバ202のユーザが、データを作成、準備、要求または分析することの他にも、ビジネス取引に関連したソース文書を閲覧およびそれにアクセスすることなどの任意の適切な目的のために、プラットフォームの少なくとも一部と交信することができるように動作可能なグラフィカルユーザインターフェースを備える。一般に、GUIはシステムによって提供される、またはシステム内で通信されるビジネスデータの、効率的かつユーザにとって扱いやすい表示を特定のユーザに提供する。具体的には、GUIは例えば、ビジネスプロセスから発したユーザタスクを示すために使用されてもよい。GUIはまた、ユーザがビジネスプロセスアプリケーション232の様々なサービスおよび機能にアクセスし、それを利用することができるようにする一般的な双方向要素も提供する。GUIはしばしば設定可能であり、表とグラフ(棒グラフ、線グラフ、円グラフ、ステータスダイアルなど)の組み合わせを支援し、タブがキー特性(例えばサイトまたはマイクロサイトなど)によって区切られるリアルタイムのポータルを構築することができる。したがってGUIは、プラットフォームの情報を処理し、効果的にその結果をユーザに視覚的に示す一般的なウェブブラウザとコマンドラインインターフェース(CLI)の組み合わせなどの任意の適切なグラフィカルユーザインターフェースを企図する。

【0029】

一般に、ノード例202は、環境200のコンポーネント間(すなわちノード202と遠隔システム250の間)の他にも、メッセージングシステム222、追加のクライアントサーバ、または図2に描かれていないがネットワーク212に通信可能に結合された他のデバイスなどの任意の他の局所または遠隔コンピュータとのワイヤレスまたはワイヤライン通信を促進するネットワーク212に通信可能に結合されてもよい。描かれている環境では、ネットワーク212は図2の中の単一のネットワークとして示されているが、ネットワーク212の少なくとも一部が送信者と受信者の間の通信を促進することができる限り、本開示の範囲から逸脱することのない連続または不連続のネットワークであってもよい。

【0030】

ネットワーク212は、企業のネットワークまたは保護されたネットワークのうちのすべてまたは一部であってもよく、一方で他の例では、ネットワーク212の少なくとも一部はインターネットへの接続を表してもよい。いくつかの例では、ネットワーク212の一部は、例えば遠隔システム250とノード202の間の接続など、仮想プライベートネットワーク(VPN)であってもよい。さらに、ネットワーク212のすべてまたは一部は、ワイヤラインまたはワイヤレスのいずれかのリンクを備えることが可能である。ワイヤレスリンクの例は802.11 a/b/g/n、802.20、WiMax、および/または任意のその他の適切なワイヤレスリンクを含んでもよい。すなわち、ネットワーク212は描かれている環境200の内側および外側の様々なコンピューティングコンポーネント間の通信を促進するように動作可能な任意の内部または外部ネットワーク、複数のネットワーク、サブネットワークまたはそれらの組み合わせを包含する。ネットワーク212は、例えばインターネットプロトコル(IP)パケット、フレームリレーフレーム、非同期転送モード(ATM)セル、音声、ビデオ、データ、およびネットワークアドレス間のその他の適切な情報を通信してもよい。ネットワーク212はまた、1つまたは複数のローカルエリアネットワーク(LANS)、無線アクセスネットワーク(RANS)、メトロポリタンエリアネットワーク(MANs)、ワイドエリアネットワーク(WANs)、インターネットのすべてまたは一部、および/または、1つまたは複数の場所での任意のその他

通信システムを含んでもよい。

【0031】

遠隔システム250は、ネットワーク212内のノード202などのリソースへのアクセスを有してもよい。特定の実装では、いくつかの例でのノード202を含むネットワーク212内のサーバは、クラウドベースのサービスを提供するためのクラウドコンピューティングプラットフォームを備えてもよい。「クラウド」、「クラウドコンピューティング」および「クラウドベース」という用語は、本開示の範囲から逸脱することなく適切に交互に使用されてもよい。クラウドベースのサービスは、クライアントコンピュータ上で局所的に実行されるアプリケーションを強化、補完、または置換するために、サーバによって提供され、ネットワークの間でクライアントプラットフォームに配信されるホストサービスであることが可能である。遠隔システム250は、別の方法ではリソースが遠隔システム250に配信可能になる前に非常に長い期間を必要とするソフトウェアのアップグレード、アプリケーションおよびその他リソースを迅速に受信するために、クラウドベースのサービスを使用することができる。さらに、他のデバイスもまた、ネットワーク212を通じてアクセス可能なサーバによって提供されるオンデマンドサービスなど、クラウドベースのサービスへのアクセスを有してもよい。さらに、クラウドプラットフォーム配置の実装は本開示の必須の要素ではなく、クラスターベースのシステムなどのその他の分散型インフラストラクチャもまた使用されることが可能である。

10

【0032】

本開示で述べられているように、オンデマンドのサービスは製品、実用的な分析、企業のポータル、管理されたウェブコンテンツ、複合アプリケーションなどの多数の種類のサービスおよびビジネスプロセス、またはビジネスアプリケーションを作成、統合、使用および提示する機能を含むことが可能である。例えば、クラウドベースの実装によって、遠隔システム250は機能を損なうことなく、古いユーザフェースインターフェースプラットフォームから新しいプラットフォームのリリースに透過的にアップグレードすることができる。

20

【0033】

図2に描かれているように、ノード202はプロセッサ208を含む。図2では単一のプロセッサ208として描かれているが、2つ以上のプロセッサは特定の必要性、要求または環境200の特定の実行形態によって使用されてもよい。各プロセッサ208は中央処理装置(CPU)、ブレード、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、またはその他の適切なコンポーネントであってよい。一般に、プロセッサ208はノード202のオペレーション、および具体的に1つまたは複数のビジネスプロセスアプリケーション232を遂行するために、命令を実行してデータを操作する。具体的には、サーバのプロセッサ208は、遠隔システム250およびそれらの各々のクライアントアプリケーション144からの要求を受信して、それに応答するために必要な機能とともに、ビジネスプロセスアプリケーション232の他のオペレーションを遂行するために必要な機能も実行する。

30

【0034】

特定の実装に関わらず、「ソフトウェア」は、少なくとも本明細書で説明されるプロセスおよびオペレーションを遂行するために実行される場合に動作可能な有形の固定媒体上にあるコンピュータ可読命令、ファームウェア、配線式またはプログラムされたハードウェア、もしくはそれらの任意の組み合わせを含んでもよい。実際に、各ソフトウェアコンポーネントはC、C++、Java(登録商標)、Visual Basic、アセンブラ、Perl、4GLの任意の適切なバージョンに加えて、その他のものも含む任意の適切なコンピュータ言語で全体的または部分的に書き込まれるか、または記述されてもよい。図2に描かれたソフトウェアの部分は、様々なオブジェクト、方法またはその他のプロセスを通じて様々な特徴および機能を実装する個々のモジュールとして示されているが、ソフトウェアは代わりに、いくつかのサブモジュール、第三者サービス、コンポーネント、ライブラリなどを必要に応じて含んでもよいということを理解されたい。反対に、様々なコンポーネントの特徴および機能は、必要に応じて単一のコンポーネントに結合されることが可能である。描かれてい

40

50

る環境200では、プロセッサ208は、ノード202上で1つまたは複数のビジネスプロセスアプリケーション232を実行する。

【0035】

高いレベルで、1つまたは複数のビジネスプロセスアプリケーション232の各々は、特に描かれた遠隔システム250およびその関連したクライアントアプリケーション254から、またはネットワーク212を通じた他のサーバまたはコンポーネントから受信された1つまたは複数の要求に応答して、およびそれに関連して、本開示による情報を実行、変更、削除、生成またはその他に管理してもよい任意のアプリケーション、プログラム、モジュール、プロセスまたはその他のソフトウェアであってよい。ある場合では、特定のノード202には1つだけのビジネスプロセスアプリケーション232が配置されてもよい。他の場合では、複数の関連した、および/または関連しないビジネスプロセスアプリケーション232は単一のノード202に保存されてもよく、または同様に複数の他のノード202の中に配置されてもよい。ある場合では、環境200は複合ビジネスプロセスアプリケーション232を実装してもよい。例えば、複合アプリケーションの部分はEnterprise Java(登録商標) Beans(EJBs)として実装されてもよく、または、デザイン時のコンポーネントはランタイムの実装を、特にJEE(Java(登録商標) Platform、Enterprise版)、ABAP(上級ビジネスアプリケーションプログラミング)オブジェクト、またはMicrosoftの.NETなどの異なるプラットフォームに生成する能力を有してもよい。

【0036】

さらに、ビジネスプロセスアプリケーション232のうちの1つまたは複数のは、(例えばインターネットなどを通じて)ネットワーク212を介して遠隔システム250またはクライアントアプリケーション254によってアクセスされ、実行されるウェブベースのアプリケーションを示してもよい。さらに、特定のビジネスプロセスアプリケーション232に関連した1つまたは複数のプロセスは、ノード202の内部にあるものとして描かれているが、遠隔から保存、参照または実行されてもよい。例えば、特定のビジネスプロセスアプリケーション232の一部は、遠隔から呼び出されるアプリケーションに関連したウェブサービスであってもよく、一方でビジネスプロセスアプリケーション232の他の部分は、遠隔システム250での処理のためにまとめられたインタフェースオブジェクトまたはエージェントであってもよい。さらに、ビジネスプロセスアプリケーション232のうちの任意のものまたはすべては、本開示の範囲から逸脱しない(図示されていない)他のソフトウェアモジュールまたは企業アプリケーションの子またはサブモジュールであってもよい。さらにまた、ビジネスプロセスアプリケーション232の部分は、ノード202で直接作業するユーザだけでなく、遠隔システム250で遠隔から作業するユーザによっても実行されてよい。

【0037】

描かれているように、ノード202はまた、ビジネスプロセスアプリケーション232を実行するためのサービス、ライブラリおよびツールを提供するビジネスプロセス管理(BPM)ランタイム234を含むことが可能である。ビジネスプロセスインスタンスは、特定のビジネスプロセスの実行中のインスタンスである。いくつかの場合では、同じビジネスプロセスの多数のインスタンスを実行すること可能である(例えば異なるビジネスプロセスインスタンスによって、多数の個別の購入注文が同時に生成されてもよい)。さらに、各ビジネスプロセスインスタンスはビジネスプロセスインスタンスをホストするノードに特有の情報に関連しているので、同じビジネスプロセスの多数のインスタンスを異なるノードで実行することができる。

【0038】

BPMランタイム234はまた、受信されたイベントに基づくプロセスステップの実行に関連した状態変化を含めて、ビジネスプロセスの任意の状態変化を処理することができる。ノード202はまた、メッセージングミドルウェア240も含む。メッセージングミドルウェア240は、分散型システム間のメッセージの送受信を促進し、トランザクショナル(フェールオーバー-セーフ)メッセージの配信、メッセージキューイングおよびパブリッシュ/サブスクライブ機能を提供するように構成されたソフトウェアまたはハードウェアインフラストラ

クチャを備えることが可能である。一般に、メッセージングミドルウェア240によって、アプリケーションモジュールは異種プラットフォーム上で分散され、アプリケーションの開発者を様々なオペレーティングシステムおよびネットワークインターフェースの細部から隔てることによって、多数のオペレーティングシステムおよびネットワークプロトコルにわたるアプリケーションを開発する複雑さを減らすことができる。いくつかの例では、メッセージングミドルウェア240は、メッセージングシステム222およびそのメッセージングキュー223へメッセージを送信し、そこからメッセージを受信するための方法および技術を提供することができる。ノード202のメッセージングミドルウェア240はまた、例えばノード202と異なるネットワーク間の種々のプラットフォームとの通信を可能にするJava(登録商標)メッセージサービス(JMS)アプリケーションプログラミングインターフェース(API)などのメッセージングミドルウェアAPI 242を提供することが可能である。

10

#### 【0039】

1つまたは複数の着信メッセージアダプタ236もまた、ノード202に含まれることが可能である。着信メッセージアダプタ236は、遠隔システム250、その他ノードまたはメッセージングシステム222などの外部コンポーネントから受信されるメッセージまたはイベントを受信するために使用されるハードウェアまたはソフトウェアコンポーネントを備える。着信メッセージアダプタはまた、メッセージアナライザモジュール238に結合されることが可能である。メッセージアナライザモジュール238は、受信されたイベントを分析して、イベントのための適切な受信者を判定するように構成された任意のアプリケーションであることが可能である。いくつかの例では、メッセージアナライザモジュール238は、受信されるイベントが送られるべきキューを判定することができる。イベントは、イベントを送信する外部コンポーネント、またはイベントに関連したその他のコンテキスト情報に基づいて、特定のノードで、または固有のプロセスインスタンスによって消費されなければならない場合がある。いくつかの例では、メッセージアナライザモジュール238は、受信されるイベントを、同じノード202上で実行中のビジネスプロセスインスタンスに関連したイベントとして識別してもよい。それらの例では、受信されるイベントまたはメッセージは、メッセージをメッセージングキュー223または他のシステムに転送または送信することなく、消費されることが可能である。

20

#### 【0040】

一般に、ノード202はまた、データおよびプログラム命令を保存するためのメモリ211を含む。メモリ211は任意のメモリまたはデータベースモジュールを含んでもよく、限定されないが、磁気媒体、光媒体、ランダムアクセスメモリ(RAM)、読み出し専用メモリ(ROM)、取り外し可能媒体、または任意のその他の適切な局所または遠隔のメモリコンポーネントを含む揮発性または不揮発性メモリの形態をとってもよい。メモリ211は、クラス、フレームワーク、アプリケーション、バックアップデータ、ビジネスオブジェクト、ジョブ、ウェブページ、ウェブページテンプレート、データベーステーブル、ビジネスおよび/または動的情報を保存するレポジトリ、ならびにノード202およびその1つまたは複数のビジネスプロセスアプリケーション232の目的に関連した任意のパラメータ、変数、アルゴリズム、命令、規則、成約、またはそれらへの参照を含むその他の適切な情報を含む様々なオブジェクトまたはデータを保存してもよい。

30

40

#### 【0041】

メモリ211はまた、ビジネスプロセスモデル214およびビジネスプロセスメタデータ216などのデータオブジェクトを保存することができる。ビジネスプロセスモデル214は様々な企業の態様またはプロセスを表すデータオブジェクトを含むことが可能であり、ビジネスプロセスメタデータ216は、ノード202が管理または通信するビジネスプロセスに関連した任意のメタデータを含むことが可能である。特に、メモリ211はインスタンス化されたプロセスコンテキスト、プロセストークンおよびその他のプロセスインスタンスデータなどのプロセスインスタンスデータを保持することができる。いくつかの実装では、ビジネスプロセスモデル214は、BPMN(ビジネスプロセスモデリング表記法)ベースのモデルまたはBPEL(ビジネスプロセス実行言語)ベースのモデルであることが可能である。

50

## 【 0 0 4 2 】

図2の描かれた環境もまた、1つまたは複数の遠隔システム250を含む。各遠隔システム250は、ワイヤラインまたはワイヤレス接続を使用して、少なくともノード202と、および/またはネットワーク212を介して接続または通信するように動作可能な任意のコンピューティングデバイスであってよい。さらに、図2に描いているように、遠隔システム250はプロセッサ256、インターフェース255、クライアントアプリケーション254およびメモリ258を含む。いくつかの例では、遠隔システム250はまた、グラフィカルユーザインターフェース(GUI)252を含むことが可能である。一般に、遠隔システム250は、図2の環境200に関連した任意の適切なデータを受信、送信、処理および保存するように動作可能な電子コンピューティングデバイスを備える。環境100に関連した、またはその外部にある任意の数の遠隔システム250が存在してもよいことを理解されたい。例えば、描かれている環境200は遠隔システム250を含むが、環境200の代替実装は、ノード202に通信可能に結合された多数のクライアント、または環境200の目的に適した任意の他の数のクライアントを含んでもよい。さらに、ネットワーク212を介して環境200と交信することができる、描かれた環境200の部分の外部にある1つまたは複数の追加の遠隔システムが存在してもよい。「遠隔システム」という用語はまた、任意のコンピュータ、アプリケーション、またはネットワーク212を通じて1つまたは複数のサーバに通信可能に結合されたモバイルデバイスなどのデバイスのことを指してもよい。さらに、各遠隔システム250は単一のユーザによって使用されるという観点から説明されているが、本開示は、多くのユーザが1つのコンピュータを使用してもよいということ、または1人のユーザが多数のコンピュータを使用して

## 【 0 0 4 3 】

いくつかの実装では、遠隔システム250はクライアントシステムであることが可能であり、GUI252は遠隔システム250に関連していてもよい。これらの例では、GUI252は、例えば遠隔システム250のユーザが、データを作成、準備、要求または分析することの他にも、ビジネス取引に関連したソース文書を閲覧およびそれにアクセスすることなどの任意の適切な目的のために、プラットフォームの少なくとも一部と交信することができるように動作可能なグラフィカルユーザインターフェースを備える。一般に、GUI252はシステムによって提供される、またはシステム内で通信されるビジネスデータの、効率的かつユーザにとって扱いやすい表示を特定のユーザに提供する。GUI252は双方向フィールド、プルダウンリストおよびユーザによって操作されるボタンを持つ複数のカスタマイズ可能なフレームまたはビューを備えてもよい。一般に、GUI252はまた、ユーザがアプリケーション254の様々なサービスおよび機能にアクセスし、それを利用することができるようにする一般的な双方向要素を提供してもよい。GUI252はしばしば設定可能であり、表とグラフ(棒グラフ、線グラフ、円グラフ、ステータスダイアルなど)の組み合わせを支援し、タブがキー特性(例えばサイトまたはマイクロサイトなど)によって区切られるリアルタイムのポータルを構築することができる。したがって、GUI252は、プラットフォームの情報を処理し、効果的にその結果をユーザに視覚的に示す一般的なウェブブラウザ、知的エンジンおよびコマンドラインインターフェース(CLI)の組み合わせなど、任意の適切なグラフィカルユーザインターフェースを企図する。しかしながら、GUI252は本開示の必須のコンポーネントではない。いくつかの例では、例えば遠隔システム250はGUIを必ずしも含まないERPシステムのサーバまたはその他のコンポーネントであってよい。

## 【 0 0 4 4 】

本開示で使用する際、遠隔システム250はパーソナルコンピュータ、タッチスクリーン端末、ワークステーション、ネットワークコンピュータ、キオスク、ワイヤレスデータポート、スマートフォン、携帯情報端末(PDA)、これらまたはその他のデバイス内の1つまたは複数のプロセッサ、または任意のその他の適切な処理デバイスを含むことが可能である。例えば、各遠隔システム250は、キーパッド、タッチスクリーン、マウス、またはユーザ情報を受け入れることができるその他デバイスなどの入力デバイス、およびデジタルデータ、視覚情報、クライアントアプリケーション254またはGUI252を含む、ノード202(お

よびビジネスプロセスアプリケーション232)または遠隔システム250自体のオペレーションに関連した情報を伝達する出力デバイス含むコンピュータを備えてもよい。入力デバイスと出力デバイスの両方は、磁気ストレージ媒体、CD-ROM、または表示すなわちGUI252を通じて遠隔システム250のユーザから入力を受信すること、および同ユーザに出力を与えることの両方のためのその他の適切な媒体など、固定または取り外し可能なストレージ媒体を含んでもよい。

#### 【0045】

いくつかの実装では、ノード202はまたメッセージングシステム222に通信可能に結合され、メッセージングシステム222は、着信イベントを保持するためにメモリ221の中の保存されるメッセージングキュー223を提供する。いくつかの例では、メモリ221は、不揮発性メモリまたはデータベースシステムであることが可能である。メッセージングシステム222は、他のコンポーネントから受信されるイベントまたはメッセージを受信、保存するため、またはそれらへのアクセスを提供するために構成された任意の電子コンピューティングデバイスであることが可能である。いくつかの例では、メッセージングシステム222はバックボーンまたはバックエンドシステムとして1つまたは複数のノード202に結合され、一方で他の例では、メッセージングシステム222はネットワーク212を通じて複数の他のノード202、デバイスおよびコンポーネントに接続された独立のシステムを表す。メッセージングシステム222は、プロセッサ228、インターフェース225、またはイベントを受信および管理するために使用されるその他のコンポーネントを含むことが可能である。いくつかの実装では、メッセージングシステム222は、メッセージングミドルウェアを通じて整合性およびフェールオーバー機能を含む。メッセージングシステム222のメッセージングミドルウェア226は、メッセージを損なうことなく、または重複するメッセージを配信することなく、トランザクショナルな方法でメッセージを受信し(キューに加え)、転送する(キューから取り除く)ことができる。さらに、メッセージングミドルウェア226はまた、先入れ先出し(FIFO)の順序付けなど、メッセージの順序付けを提供することができる。すなわち、メッセージングシステム222のメッセージングミドルウェア226は、プロセスインスタンスによって後で検索するために、着信イベントを保持するために使用されることが可能である。メッセージングミドルウェア226はメッセージングシステム222の中央データベースとして実装されることが可能であるが、局所保持などの任意の適切な手段を使用して、または遅延複製技術(lazy replication techniques)とともに実装されることが可能である。

#### 【0046】

例えば、遠隔システム250などの外部コンポーネントは、クラウドネットワークの中の特定のノード202にイベントまたは要求を送信することができる。イベントは異なる場所で消費されなければならない場合があるが、ノード202はメッセージングキュー223の中でイベントを保持するためにメッセージングシステム222へイベントを転送することができるので、適切なビジネスプロセスは、消費のためにメッセージングキュー223からイベントを検索することができる。受信されるイベントのためにキューを提供するメッセージングシステム222によって提供される機能は、メッセージングシステム224によって遂行されることが可能である。特定の实装では、メッセージングサービス224はまた、メッセージングキュー223に保存された特定のイベントを消費するために使用されるべきプロセスインスタンスを含む特定のノードに、通知メッセージを送信することができる。通知メッセージは、メッセージまたはイベントがメッセージングキュー223に送信される場合、(メッセージングミドルウェア240などを通じて)ノード202自体によって提供されることが可能である。メッセージングシステム222は、図2の中ではノード202に関して離れて配置されるものとして描かれているが、いくつかの実装では、メッセージングシステム222は、複数のノードのうちの1つの部分として配置されるか、またはBPMSの中の異なるノードの間で分散されることが可能である。

#### 【0047】

図2は、複数の要素を含むものとして、またはそれらに関連したものとして述べられて

10

20

30

40

50

いるが、図2の環境200内に描かれたすべての要素が、本開示の各々の代替実装の中で利用されなくてもよい。例えば、本明細書で述べられている要素のうちの1つまたは複数の環境200の外部に配置されてもよく、一方で他の例では、特定の要素は、描かれている実装の中で述べられていない他の要素とともに、他の述べられている要素のうちの1つまたは複数の一部の内部に、またはそれらの一部として含まれてもよい。さらに、図2に描かれている特定の要素は他のコンポーネントと結合されてもよく、また本明細書で述べられている目的に加えて、代替または追加の目的のために使用されてもよい。

#### 【0048】

図3は、スケーラブルなイベント発行のためのプロセス例300を描く。図3で示されているように、メッセージ(すなわちイベント)305は、310で第1クラウドインスタンス320(すなわちコンピュータノード320)上で受信される。イベント305は最初に、コンピュータノード320の特定のプロセスインスタンスに転送されることが可能である。特定の状況では、第1コンピュータノード320は、イベント305を消費するために割り当てられた、またはイベント305に関連付けられたプロセスインスタンスを持たなくてもよい。代わりに、他のコンピュータノードに配置された1つまたは複数のその他のビジネスプロセスインスタンスは、イベント305の適切な受信者であってもよい。したがって、影響を受けるプロセスインスタンスは、325で判定される。影響を受けるプロセスインスタンスの判定は、メッセージのペイロードおよびプロセスのデータコンテキストに基づいて、受信側プロセスインスタンスが着信メッセージと照合される関連手続きを含むことが可能である。その他の場合には、メッセージは前もって1つの固有のプロセスインスタンスを論理的に参照してもよいので、明確な相関は必要ではない。影響を受けるプロセスインスタンス345は第1コンピュータノード320または異なるコンピュータノード340に配置されてもよい。影響を受けるプロセスインスタンス345が異なるコンピュータノード340に配置される場合、イベント305は330でメッセージングミドルウェアを介してインスタンス固有のキューの中に入れられる。特定の状況では、多数のプロセスキューは特定のコンピュータノードでホストされることが可能であり、各プロセスキューは固有のビジネスプロセスインスタンスに関連している。したがって、メッセージングミドルウェアは、受信側プロセスインスタンスに関連したプロセスインスタンス識別子に基づいて、イベント305を保持するための固有のプロセスキューを識別するために使用されることが可能である。図3に示されているように、プロセスキューは、メッセージングミドルウェア335を通じてアクセスされるデータベースに支援されるプロセスキューであることが可能である。

#### 【0049】

任意のイベントでは、メッセージングミドルウェアは、受信側プロセスインスタンスによって後で検索されるために着信イベントを保持することを可能にするインターフェースを提供することができる。いくつかの実装では、メッセージングミドルウェア335は、各プロセスインスタンスが消費するためにイベントを検索するためのメッセージングキュー223へのアクセスを有しながら、多数のコンピュータノードの間の異なるプロセスインスタンスで利用可能なレポジトリまたはバックボーンシステムの中の集中データベースと接続して実装されることが可能である。代替として、メッセージングミドルウェア335は、着信イベントのために分散されたキューを提供するために、局所での持続性を備えた複製プロトコルなどの他の方法に依存することができる。影響を受けるプロセスインスタンスが最初にイベント305を受信したノード320と同じコンピュータノードに配置される場合、イベント305は、メッセージングミドルウェア335でイベント305を保持することなく、適切なプロセスインスタンスに配信されるか、またはそれによって消費されることが可能である。

#### 【0050】

いくつかの実装では、影響を受けるプロセスインスタンス345は、どのプロセスインスタンスが影響を受けるかが判定された後、およびプロセスキューの中にイベント305を保持した後、メッセージングミドルウェアを通じてアクティブに通知を受ける。いくつかの例では、コンピュータノード340への通知呼び出しは、コンピュータノード340のプロセス

インスタンス345によってイベント305を検索し、それを消費する際に遅延を回避することができる。いくつかの実装では、影響を受けるプロセスインスタンス345を含むコンピュータノード340は、特定のイベント305がメッセージングミドルウェア335で受信されたかどうかを判定するために、350で、メッセージングミドルウェア335でのプロセスキューの通常のポーリングを遂行することができる。コンピュータノード340は次いで、イベント305がプロセスインスタンス345による消費のためにコンピュータノード340で受信されたかどうかを判定した後で、メッセージングミドルウェア335からイベントを検索することができる。イベント305は一旦コンピュータノード340で受信されると、プロセスインスタンス345によって消費されることが可能である。

【0051】

図3に関して上で述べられたようなイベントのメッセージングミドルウェアへの転送は、BPMSの中の各ノードの間で実施されることが可能である。しかしながら、いくつかの例では、イベントは特定のBPMSのノード、特定のプロセスインスタンス、特定の受信されたイベント、または特定の状況下についてのみ、メッセージングミドルウェアの中で保持される。メッセージングシステム222のプロセスキュー223の中にイベントを保持することによって、特に特定のシナリオに関して、イベントを交換する場合のBPMSの性能は改善されることが可能である。ユーザタスクが、非常に頻繁にプロセスインスタンスを起動することと通信する状況では、プロセスキュー223の中に受信されたイベントを保持することは、プロセスインスタンスを頻繁に起動することに関連した待ち時間を減らすことができる。例えば、その形式でデータを入力し、完了後にその形式をプロセスインスタンスに返さなければならない複数のユーザに対して示される形式は、任意のユーザが引き起こすタスク状態の変更が、プロセスインスタンスに送信されるイベントにつながる可能性があることから、資源を占有する可能性がある。比較的長いユーザタスクの処理時間を前提とすると、メッセージングミドルウェアのポーリング手法を通じてタスク状態変更イベントをプロセスインスタンスに渡すことは、BPMSの性能にとって有益である場合がある。

【0052】

図4は、外部コンポーネントからコンピュータノードで受信されたメッセージを処理するためのプロセス例400を描く。最初に、第1コンピュータノードで受信されたメッセージは405で識別される。このメッセージは、遠隔システム250などの外部デバイスの外部ソフトウェアコンポーネントから受信されるイベントに関連したメッセージまたはその他情報であることが可能である。受信されたメッセージのコンテンツは、410で分析される。特に、メッセージに関連したビジネスプロセスインスタンスは、分析の間に415で識別される。例えば、特定のビジネスプロセスインスタンスは、メッセージを処理するため、またはメッセージに基づいた一定の行為を遂行するために割り当てられてもよい。いくつかの例では、メッセージはメッセージが関連した特定のビジネスプロセスインスタンスを特定してもよく、一方で他の例では、メッセージに関連した特定のビジネスプロセスインスタンスは、ルールセットまたはその他の関連方法に基づいて引き出されてもよい。メッセージは、1つまたは複数のビジネスプロセスインスタンスを実行する第1コンピュータノードで受信されてもよいが、特定の受信されるメッセージのために割り当てられるビジネスプロセスインスタンスは、同じコンピュータノードに配置されなくてもよい。したがって420で、識別されたビジネスプロセスインスタンスが第1ノードで実行中であるかどうかについて判定が行われる。

【0053】

識別されたビジネスプロセスインスタンスが第1ノードで実行中である場合、受信されたメッセージは440で識別されたビジネスプロセスインスタンスに提供され、ここでメッセージおよびそのコンテンツは、局所的にアクセスされて第1ノードで消費されることが可能である。識別されたビジネスプロセスインスタンスが第1ノードで実行中ではない場合、受信されたメッセージは第2コンピュータノードで処理される。しかしながら、第2コンピュータノードの位置はまだ識別されないかもしれない。したがって、メッセージは第2コンピュータノードによって検索されるために、425でメッセージングミドルウェアに送

10

20

30

40

50

信される。いくつかの実装では、アクティブ通知は、メッセージが第2コンピュータノードによって検索されるのを待機していることを第2コンピュータノードに通知するために、メッセージングミドルウェア内で有効にされることが可能である。したがって、430でアクティブ通知が有効になっているかどうか判定が行われる。アクティブ通知が有効になっている場合、435で通知メッセージは第2コンピュータノードに送信される。アクティブ通知は、435でメッセージングキューに送信される特定のメッセージに関連した情報、または第2コンピュータノードに関連したメッセージがさらなる詳細を伴わずにメッセージングキューで利用可能であるという通知を含むことが可能である。アクティブ通知が有効になっていない場合、プロセスは通常のオペレーションに戻り、さらなるメッセージの到着を待つ。

10

**【0054】**

図5は、メッセージングキューから関連したメッセージを検索するためのプロセス例500を描く。図4に関して上で述べたように、メッセージは第1コンピュータノードで受信されることが可能であるが、その後、そのメッセージが受信される第1コンピュータノード以外の第2コンピュータノードに配置されたビジネスプロセスインスタンスによって消費されるか、またはそれに関連するという判定の後で、メッセージングミドルウェアを使用して集中メッセージングキューに転送される。いくつかの例では、第2コンピュータノードに集中メッセージングキューから検索するためのメッセージの利用可能性を知らせるために、メッセージングミドルウェアによって通知が第2コンピュータノードに送信されることが可能である。したがって505で、検索するために利用できる可能なメッセージを示す通知が受信されたかどうかについての判定が第2コンピュータノードで行われる。

20

**【0055】**

いくつかの実装では、この通知方法はポーリング手法に結合される。受信側プロセスインスタンスは、一定の間隔で保留中のメッセージのためのメッセージキューをポーリングしてもよいが、第1コンピュータノードから通知が受信されていれば直ちにキューを確認してもよい。したがって、メッセージがメッセージキューの中で第2コンピュータノードのために利用可能であることを示す通知が受信されている場合、515で集中メッセージングキューは関連したメッセージについてポーリングされる。通知が受信されていない場合、510で、任意の検索するために利用可能なメッセージについて集中メッセージングキューにポーリングする時間かどうかについての判定が行われる。各ビジネスプロセスインスタンスのためのポーリング時間は、遂行されるビジネスプロセス間の差を許容するために異なってもよい。各ビジネスプロセスインスタンスは、例えばそのプロセスインスタンスが緊急を要するプロセスインスタンスなのか、またはそうではないのかによって、その特定のビジネスプロセスインスタンスに適したポーリング時間に関連付けられてもよい。いくつかの例では、ポーリング時間はユーザまたは管理者によって手動で変更されることが可能であり、デフォルト値に設定されるか、または新しいメッセージが受信される平均値または中央値の時間に関連した計算に基づいて、動的に変更される。いくつかの例では、デフォルトのポーリング時間が使用されてもよいので、メッセージは異なる時間でビジネスプロセスに送信されてもよい。メッセージングキューにポーリングする時間ではない場合、プロセス500は、通知が関連したビジネスプロセスノードから受信されたかどうかについての判定に戻る(505)。メッセージングキューにポーリングする時間である場合、第2コンピュータノードは515で関連したメッセージについて集中メッセージングキューにポーリングする。520で集中メッセージングキューの中に保存された関連したメッセージが存在しない場合、プロセス500は、メッセージングミドルウェアから通知が受信されたかどうかについての判定に戻る(505)。メッセージングキューの中に関連したメッセージが存在する場合、次いで525で関連したメッセージが集中メッセージングキューから検索される。メッセージはメッセージングキューから検索された後、530で、第2コンピュータノードで適切なビジネスプロセスインスタンスの中で消費される。

30

40

**【0056】**

図6は、着信メッセージを伴うビジネスプロセス例600を描く。図6に示されているよう

50

に、ビジネスプロセス例は、第1アクティビティ630と関連して開始される。ビジネスプロセスの間、中間メッセージイベント625は上位のプロセスブランチ650で着信メッセージを待機し、ユーザタスク620は処理者に発行されて、下位のブランチ660で完了されるのを待機する。両方のブランチは、同時に起動される。すなわち、中間メッセージイベント625のためのメッセージ610は、ユーザが下位ブランチからユーザタスク620を処理する間、その前、またはその後に受信されてもよい。(例えば、中間メッセージイベント625によって受信されるメッセージ610またはユーザタスク620でのタスク状態の変更などの)消費されるイベントのうちの任意のものが、任意のクラウドインスタンスで独立して受信される場合があるクラウドベース環境の本質的な複雑性を処理するための専用プロトコルは、使用されることが可能である。実際に、ビジネスプロセスは第1クラウドインスタンスで実行中であってよく、一方で(例えば汎用のロードバランサによって発行され、配信される場合など)中間メッセージイベントのためのメッセージは他のクラウドインスタンスで受信され、自分の受信箱からタスクを処理しているユーザからのウェブ要求は、第3クラウドインスタンスで受信される。両方のイベント(中間メッセージイベント625によって受信されるメッセージ610およびユーザ620からのタスク状態変更)は、大幅な性能の低下を取り入れることなく、またはクラウドネットワークのスケールアウト特性を変えることなく、信頼できる方法でビジネスプロセスに到達しなければならない。

#### 【0057】

受信されたイベントをメッセージングキューの中に保持することによって、プロセスインスタンスは、プロセスインスタンスの存続期間の間特定のクラウドインスタンスに存在することが可能であり、このことはしばしばビジネスプロセスの「粘性」と呼ばれる。これに対する例外は、(例えば、負荷の一部を処理するために追加のクラウドインスタンスが割り当てられることなど)クラウドトポロジーの変更を含むことが可能である。ビジネスプロセスにトランザクショナルな方法でイベントを確実に一貫して受信させるために、(例えば図6のタスク状態の変更620およびメッセージ610などの)任意の到着するイベントは、到着イベントがBPMSランタイム234に配信されるときと同じトランザクションの中で、メッセージングキュー223とともにメッセージングミドルウェア上で局所的に保持される。イベントは、受信側プロセスが現在存在するクラウドインスタンスとは異なるクラウドインスタンスで受信される場合、イベントが発行されるべきプロセスインスタンスによる検索のために、データベースに支援されるキューの中で保持される。いくつかの例では、イベントは多数のプロセスインスタンスに発行されなければならない場合がある。

#### 【0058】

イベントは、受信側プロセスインスタンスが現在存在するクラウドインスタンスに配信される場合、直ちにプロセスインスタンスに配信され、メッセージングキューを迂回する。イベントは、イベントを消費するための適切なプロセスインスタンスに首尾よく配信されているので、ここではさらなるステップは必要とされなくてもよい。しかしながらイベントは、受信側プロセスインスタンスが存在しないクラウドインスタンスに配信される場合、イベントを受信側プロセスインスタンスに配信するために、集中メッセージングキューの中で保持されてもよい。

#### 【0059】

図7および8は、1つまたは複数のイベントを適切なクラウドインスタンスに発行するためのプロセス例700および800を描く。描かれている図7の例では、完了されたユーザタスクの表示は第1クラウドインスタンスで受信されることが可能である。完了されたユーザタスクは、ユーザに関連した特定のタスク状態が変更されなければならないというユーザの表示であってもよい。したがって710で、この表示は第1クラウドインスタンスで受信される。タスク状態の変数は720で、第1クラウドインスタンスに関連したBPMSランタイムからフェッチされることが可能である。次いで、第1クラウドインスタンスは730で、タスク状態を変更するためにBPMSランタイムに要求を提出する。共有される状態が関与する特定の例では、クラウドインスタンスによって受信されるイベントのタイプは、受信されたイベントに基づいてビジネスプロセス状態の変数が変化とともに更新されながら、ビジネス

プロセス状態に対する不要な変更を回避するために、プロセス状態変数を保護すること、または「ロックすること」が必要な場合がある。例えば、ある種のタイプのイベントは、新たな状態変数の作成を引き起こす。作成時に、新たな状態変数は既存のクラウドインスタンスにはまだわからないので、他のプロセスインスタンスが新たな状態変数に不要な変更を加えることはなく、ロック機構は必要とされない。しかしながら、特定のタイプのイベントは既存の状態変数の変更または削除を引き起こす場合がある。これらの例では、既存の状態変数をロックし、状態変数への不要なアクセスを回避するために、中央ロック機構が実装されることが可能である。しかしながら状態変数をロックすることは、その状態変数が多数のコンポーネント、プロセスなどによって一度に操作されることが可能な(すなわち状態変数がそれらの中で共有される)場合にのみ必要とされる。多くの場合、ロックプロトコルは必要ではない。

10

#### 【0060】

描かれている例を見ると、ユーザによって提出されたタスク状態に対する変更は、要求された変更が、タスクおよびユーザにタスクを提示するタスク管理コンポーネントを統合するプロセスインスタンス間で共有される既存の状態変数の変更につながることから、タスク状態に関連した状態変数のロックを必要とする場合がある。図7に見られるように、BPMSランタイムは、タスク状態変数に関して整合性違反を回避するために、740で中央ロックサービスにアクセスすることによってタスク状態変数上のロックを取得することができる。一旦タスク状態変数が中央ロックサービスによってロックされると、BPMSランタイムは、第1クラウドインスタンスから受信されたタスク変更要求に応じて、750で変更イベントを生成することができる。次いでイベントは、適切な受信側プロセスインスタンスにイベントを発行して、タスク状態変更を完了するために、図2に描かれたメッセージングキュー223の中など、760でメッセージングミドルウェアの中で保持されるか、またはキューに加えられる。イベントが中央データベースへ渡された後、770でタスク変数上のロックは中央ロックサービスによって解除されることが可能である。いくつかの実装では、クラウドインスタンスにイベントが検索のために利用可能であることを知らせるために、受信側プロセスインスタンスが配置されたクラウドインスタンスに通知呼び出しが送信されることが可能である。これらの例では、通知は780で信号イベントとして、受信側プロセスインスタンスに関連する第2クラウドインスタンスに関連したBPMSランタイムに送信されることが可能である。

20

30

#### 【0061】

図8では、信号イベントは780で、第2クラウドインスタンスのBPMSランタイムで受信されることが可能である。受信側プロセスインスタンスが存在するノードでは、局所のプロセスインスタンスがイベントを受信できるように、特定の構造が実装されることが可能である。いくつかの実装では、受信側プロセスインスタンスは着信イベントのために関連したデータベース上でポーリング方法または定期的な検査を遂行することができる。この検査は、局所のクラウドインスタンスに存在するすべてのプロセスエンティティで受信されることが可能なすべてのイベントのためのプロセスキューを検査する単一の定期的なデータベースルックアップに統合されることが可能である。このプロセス固有のデータベースルックアップは、局所のクラウドインスタンスに存在するすべてのプロセスインスタンスのためのイベントキューを検査する単一のトランザクションの一部であってもよい。代替として、各プロセスインスタンスは、異なるプロセス間のよりよい分離を達成し、個々のポーリング間隔を構成するために、それら自体のポーリングトランザクションを有してもよい。したがって、データベーストランザクションの数を最小に抑えることができる。ポーリング方法はクラウドインスタンスによって実装されてもよいが、中央データベースで着信イベントを示す信号イベントが受信される場合、クラウドインスタンスのBPMSランタイムは中央データベースから直ちにイベントを検索することが可能であり、このことはポーリングだけに頼ることによって生じるいくつかの遅延を減らすことができる。

40

#### 【0062】

50

メッセージキューから新たに到着したイベントをフェッチすることは、データベース検査の間の時間間隔が特定のプロセスインスタンスに対して設定可能な定期的なポーリング要求を使用して遂行されることが可能である(プロセスインスタンスのために間隔が設定されない場合、そのプロセスタイプまたはすべてのプロセスタイプのためにデフォルト値が適用されることが可能である)。いくつかの実装では、データベース検査の間の時間間隔は、以前に受信されたイベントの頻度、受信側ビジネスプロセスインスタンスに関連したビジネスプロセスタイプ、またはビジネスプロセスインスタンスに関連した任意のその他の要素に基づいて自動的に調整されることが可能である。他のクラウドインスタンスが、プロセスが存在するクラウドインスタンスに、イベントがそのクラウドインスタンスに関連したメッセージキューのうちの1つに含まれていたことをアクティブに通知する場合、ポーリング間隔は上書きされることが可能である。したがって、長いポーリング間隔から生じる増加した待ち時間は回避されることが可能である。通知メカニズムが省略されるか、または通知が失われた場合、次のポーリング間隔は最終的にメッセージキューからメッセージをフェッチすることになるので、整合性は依然として維持される。

#### 【0063】

描かれている例では、第2クラウドインスタンスのBPMSランタイムは、810で中央データベースのポーリングを開始し、820で新たにキューに入れられるイベントを探すために、中央データベースへのルックアップ呼び出しを起動する。ここで、図7に関して上で述べられたように、第1クラウドインスタンスによって中央データベースに提出されるイベントは、消費のために第2クラウドインスタンスによって検索されることが可能である。いくつかの例では、検索されたイベントはプロセス状態の一部になる必要があり、このことはプロセス状態変数の変更の中でイベントを具体化することによって達成されることが可能である。しかしながら、第2クラウドインスタンスで受信されたイベントは、受信側プロセスインスタンスに関連した状態変数のロックを必要とする場合がある。したがって、840でイベントが対応する状態変数に適用される前に、830で中央ロックサービスからロックメカニズムが要求される。次いで、イベントはメッセージキューからフェッチされ、メッセージキューから削除される(またはキューから取り除かれる)。プロセスインスタンスがキューからイベントをフェッチし、それを状態変数に適用したトランザクションが行われた後、850でロックは解除されることが可能である。

#### 【0064】

ここで、BPMSランタイムは次いで任意で、状態変数の変更に対応する連続したプロセスステップを起動することができる。これらのステップは通常、プロセスインスタンスの制御フローおよび/またはデータフローの側面に影響を及ぼす。特定の状況下では、これらのプロセスステップを起動することは遅延されるか、または他の状況によって異なってもよい。これらの場合では、具体化されたイベント(すなわちプロセス状態変数)は依然としてプロセス状態の一部であるが、実際にはただプロセスによって後で消費されてもよい。これらのうちのいくつかの場合では、プロセスはイベントを消費しなくてもよい。これらの場合では、BPMSランタイムは(1)プロセスが終了したときに具体化されたイベントを除去するため、または(2)その時点で他のプロセスインスタンスのためにイベントを解放するために構成されてもよい。例えば、各インスタンスが固定された数のメッセージだけを処理するプロセスインスタンスによってメッセージのストリームが消費されるシナリオでは、その数を越えるメッセージは後に続くプロセスインスタンスによって拾い上げられなければならない。その他の場合では、一旦プロセスが終了した後、イベントは実際には無意味なものとなる場合がある。例えば、関連したユーザタスクがまだ進行中であった間に、プロセスインスタンスはキャンセルされるかもしれない。そのユーザタスクが完了したとき、対応するイベントは他のプロセスインスタンスに発行される必要はなく、破棄されることが可能である。

#### 【0065】

先の図および添付の説明は、プロセス例およびコンピュータ実装可能な手法を描く。しかしながら環境100(またはそのソフトウェアもしくははその他コンポーネント)は、これら

10

20

30

40

50

およびその他タスクを遂行するための任意の適切な手法を使用、実装または実行することを企図する。これらのプロセスは単に例示を目的とするものであり、記述された、または類似の手法は、同時に、個別に、または組み合わせることを含めて、任意の適切なときに実行されてもよいということを理解されたい。さらに、これらのプロセスの中のステップの多くは同時に、および/または示されているものとは異なる順序で行われてもよい。さらに環境100は、方法が適切である以上、追加のステップ、より少ないステップおよび/または異なるステップのプロセスを使用してもよい。

【 0 0 6 6 】

換言すると、本開示は特定の実施形態および一般に関連した方法に関して述べられているが、これらの実施形態および方法の修正および置換は、当業者には明らかであろう。したがって、上の実施形態例の説明が本開示を定義または制約することはない。本開示の精神および範囲から逸脱することなく、その他の変更、代用および修正もまた可能である。

【 符号の説明 】

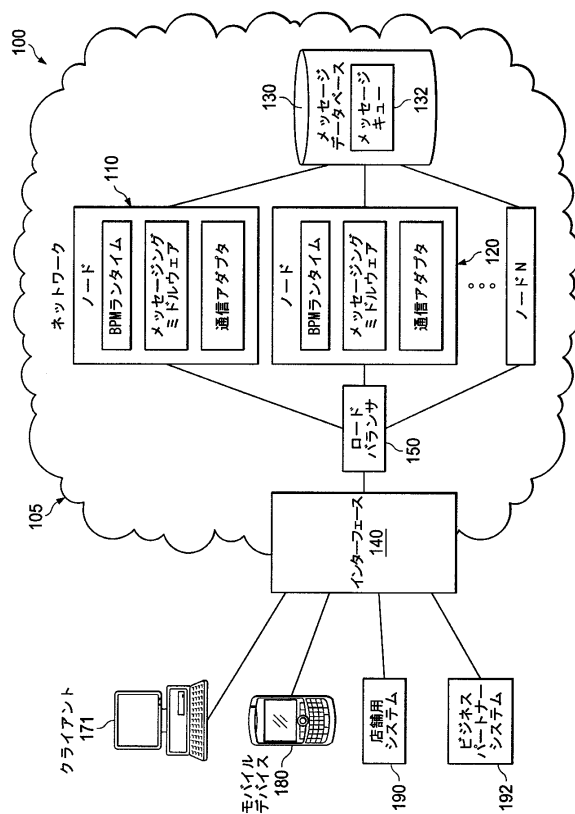
【 0 0 6 7 】

100	環境	
110	ノード	
105	クラウドネットワーク	
120	ノード	
140	インターフェース	
150	ロードバランサ	20
171	クライアント	
180	モバイルデバイス	
190	店舗用システム	
192	ビジネスパートナーシステム	
200	環境	
202	コンピュータノード	
205	インターフェース	
208	プロセッサ	
211	メモリ	
212	ネットワーク	30
214	ビジネスプロセスモデル	
216	ビジネスプロセスメタデータ	
221	メモリ	
222	メッセージングシステム	
223	メッセージングキュー	
224	メッセージングシステム	
225	インターフェース	
226	メッセージングミドルウェア	
228	プロセッサ	
232	ビジネスプロセスアプリケーション	40
234	BPMランタイム	
236	着信メッセージアダプタ	
238	メッセージアナライザモジュール	
240	メッセージングミドルウェア	
250	遠隔システム	
252	グラフィカルインターフェース	
254	クライアントアプリケーション	
255	インターフェース	
256	プロセッサ	
258	メモリ	50

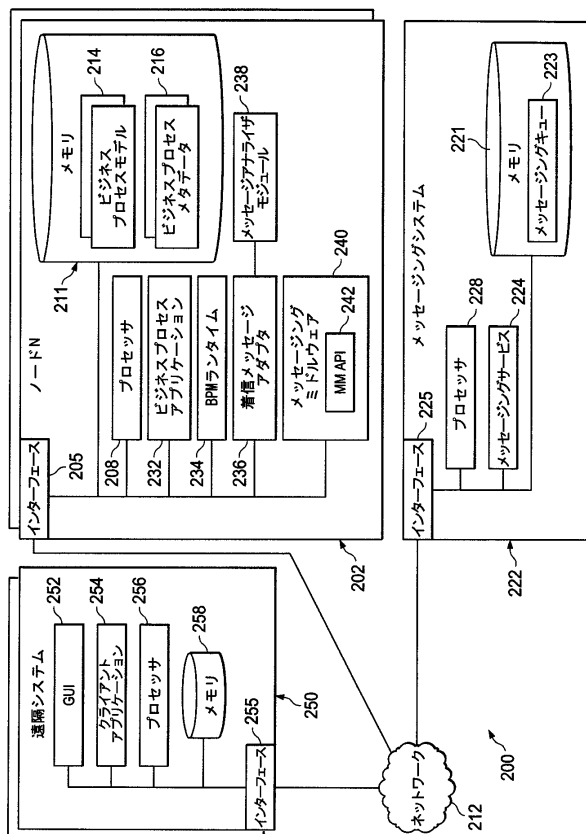
- 305 メッセージ(イベント)
- 320 第1クラウドインスタンス(コンピュータノード)
- 335 メッセージングミドルウェア
- 340 コンピュータノード
- 345 プロセスインスタンス
- 400 プロセス例
- 440 メッセージ
- 500 プロセス例
- 600 ビジネスプロセス例
- 610 メッセージ
- 620 ユーザタスク
- 625 中間メッセージイベント
- 630 第1アクティビティ
- 650 プロセスブランチ
- 660 下位のブランチ
- 700 プロセス例
- 800 プロセス例

10

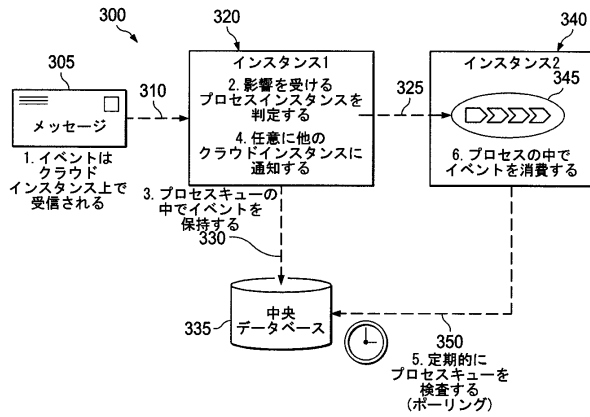
【図1】



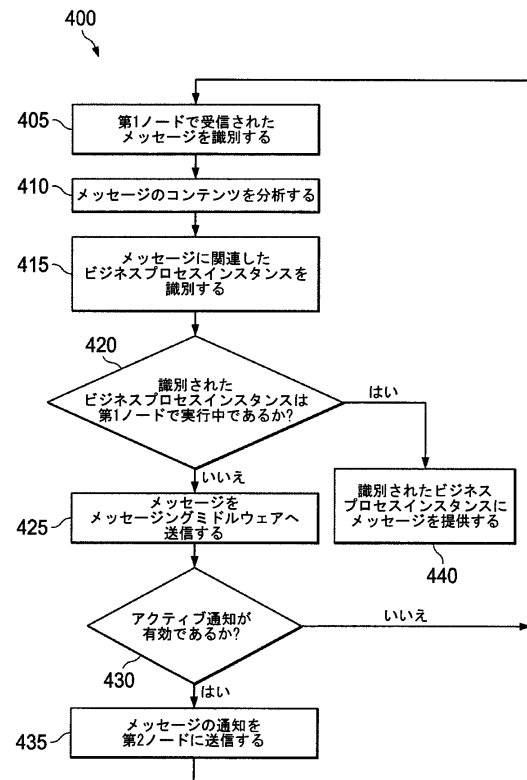
【図2】



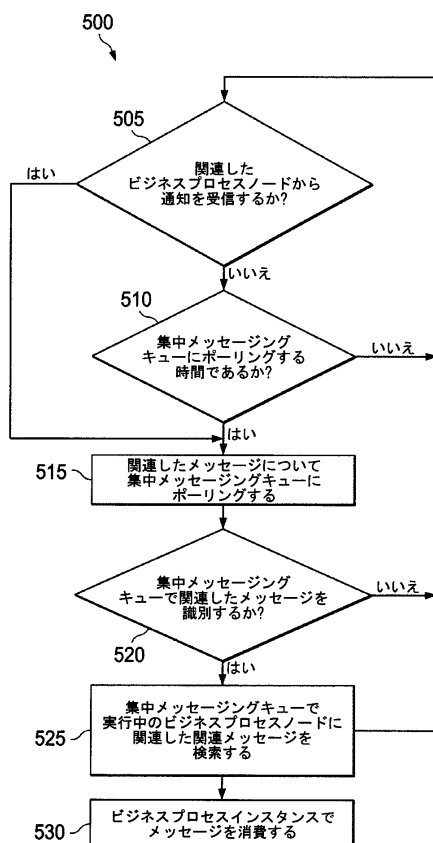
【図3】



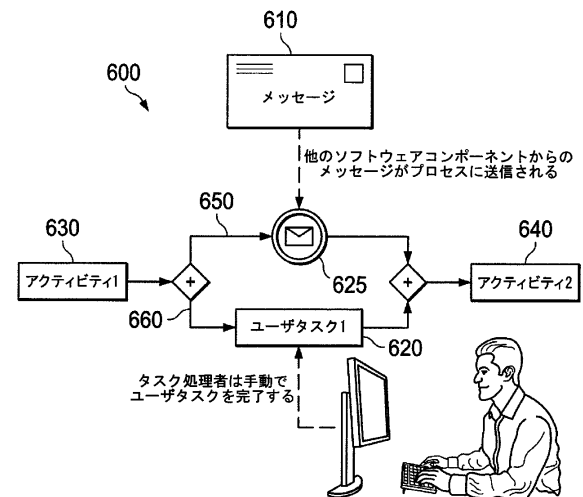
【図4】



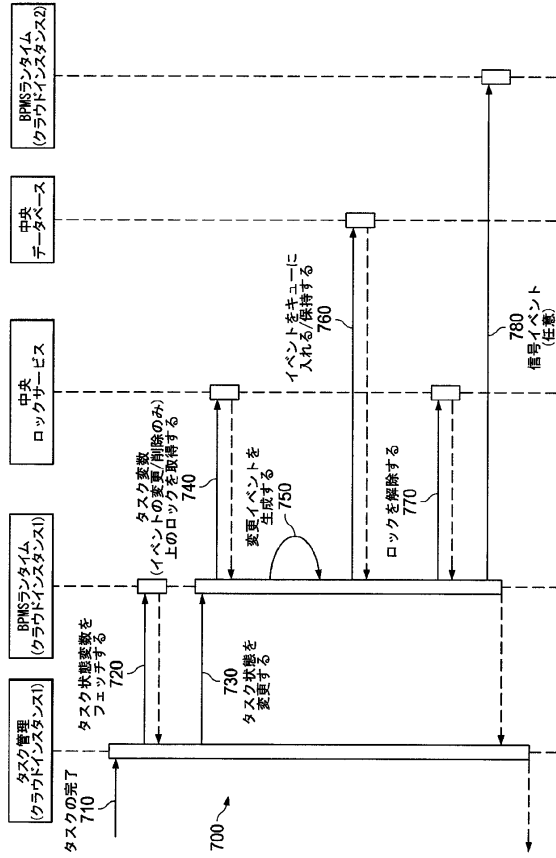
【図5】



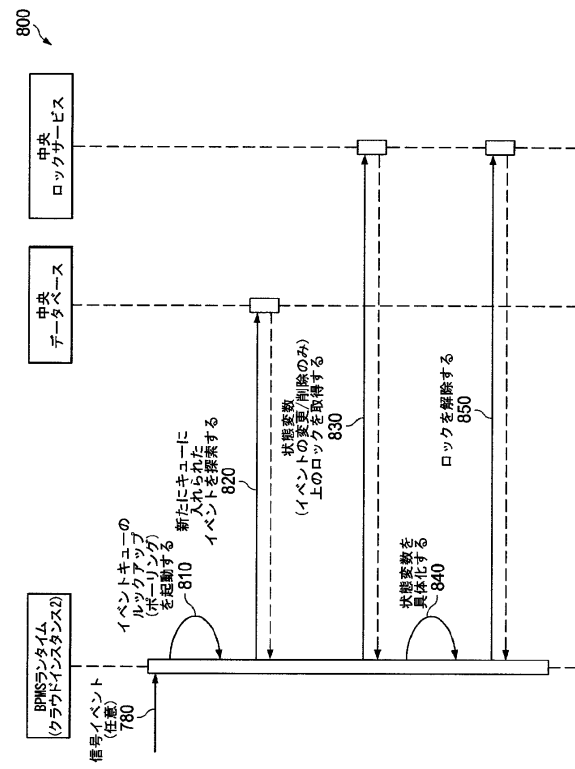
【図6】



【図 7】



【図 8】



---

フロントページの続き

(72)発明者 ゼレン・バルコ

ドイツ・69469・ヴァインハイム・ミュルハイマー・タールシュトラッセ・3 / 7

審査官 田中 幸雄

(56)参考文献 特開平9 - 34730 (JP, A)

特開2005 - 71031 (JP, A)

特開2005 - 309519 (JP, A)

特表2006 - 511857 (JP, A)

米国特許出願公開第2005 / 0265362 (US, A1)

(58)調査した分野(Int.Cl., DB名)

G06F 9 / 50