(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5462263号
(P5462263)
(45)発行日　平成26年4月2日(2014.4.2)　(24)登録日　平成26年1月24日(2014.1.24)

(51)Int.Cl.　　F I
G06F 13/00 (2006.01)　　G06F 13/00 520A
H04L 12/54 (2013.01)　　H04L 12/54

請求項の数 18　(全 21 頁)

(21)出願番号　特願2011-525077 (P2011-525077)
(86)(22)出願日　平成21年8月13日 (2009.8.13)
(65)公表番号　特表2012-507064 (P2012-507064A)
(43)公表日　平成24年3月22日 (2012.3.22)
(86)国際出願番号　PCT/US2009/053770
(87)国際公開番号　WO2010/027623
(87)国際公開日　平成22年3月11日 (2010.3.11)
審査請求日　平成24年8月2日 (2012.8.2)
(31)優先権主張番号　12/203,826
(32)優先日　平成20年9月3日 (2008.9.3)
(33)優先権主張国　米国 (US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　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　100080137
弁理士　千葉　昭男
(74)代理人　100096013
弁理士　富田　博行
(74)代理人　100113974
弁理士　田中　拓人

最終頁に続く

(54)【発明の名称】確率的な動的ルーター－サーバーメッシュルーティング

(57)【特許請求の範囲】
【請求項１】
コンピューティング環境中の処理ノードにおいて、信頼性のないルーティングデータを使用してメッセージをルーティングする方法であって、
コンピューター可読通信媒体からメッセージを受信するステップと、
前記メッセージの１つまたは複数の特性プロパティを計算して、前記メッセージの処理のためのサービスにおけるサービスインスタンスに対する状態要件を決定するステップと、
前記メッセージを処理するための前記状態要件を満たす適切なサービスインスタンスを取得することを試みるステップであって、前記メッセージを処理するための前記状態要件を満たす適切なサービスインスタンスを取得する試みが、適切なサービスインスタンスの取得に成功したとき、前記メッセージが前記状態要件に基づいて処理されるステップと、
前記メッセージを処理するための前記状態要件を満たす適切なサービスインスタンスを取得する試みが、適切なサービスインスタンスの取得に成功しないと判定し、結果として、ルーティング情報の信頼性のないローカルキャッシュを使用して処理ノード間の連携なしに前記メッセージをリダイレクトして、前記メッセージの処理のための前記状態要件を満たす適切なサービスインスタンスを有するかまたは適切なサービスインスタンスをうまく取得することのできるサーバーノードに前記メッセージを到達させることを試みるステップと、
前記メッセージの１つまたは複数の特性プロパティを計算する前またはした後に、前記

メッセージに対して前記サービスインスタンスから独立した処理を実施するステップと、
を含む方法。
【請求項２】
前記サービスインスタンスから独立した前記処理は、前記メッセージの１つまたは複数の特性プロパティを計算する前に前記メッセージに対して実施される、請求項１に記載の方法。
【請求項３】
前記メッセージを処理するための前記状態要件を満たす適切なサービスインスタンスを取得することを試みるステップが、インスタンスコーディネーターから適切なサービスインスタンスを取得することを試みるステップを含み、前記インスタンスコーディネーターが、ネットワーク中の処理ノードにおいてまだ保持されていない前記ネットワーク中の全てのサービスインスタンスを記憶する、請求項２に記載の方法。
【請求項４】
前記インスタンスコーディネーターにも前記ネットワーク中の前記処理ノードのいずれにも適切なサービスインスタンスが存在しないと前記インスタンスコーディネーターが判定したときに、新しい適切なサービスインスタンスを生み出す動作をさらに含む、請求項３に記載の方法。
【請求項５】
前記適切なサービスインスタンスが前記ネットワーク中の前記処理ノードのうちの１つによってロックされていると前記インスタンスコーディネーターが判定したときに、ロック所有側に関する情報を要求元処理ノードに提供する動作をさらに含む、請求項３に記載の方法。
【請求項６】
前記インスタンスコーディネーターが複数の特性をサービスインスタンスに関連付け、インスタンスコーディネーターから適切なサービスインスタンスを取得することを試みるステップが、１つまたは複数の追加の特性を送るステップを含み、前記追加の特性が、サービスインスタンスを取得するこの試みに対しては考慮されないが、適切なサービスインスタンスを取得するこの試みに前記処理ノードがもし成功した場合には、前記追加の特性が、前記サービスインスタンスを記述する前記複数の特性に加えられる、請求項３に記載の方法。
【請求項７】
前記インスタンスコーディネーターが複数の特性をサービスインスタンスに関連付け、インスタンスコーディネーターから適切なサービスインスタンスを取得することを試みるステップが、前記サービスインスタンスに関連する前記特性の全てを送らずに、前記特性のいずれか１つまたは複数を前記インスタンスコーディネーターに送るステップを含む、請求項３に記載の方法。
【請求項８】
コンピューティング環境中の処理ノードにおいて、信頼性のないルーティングデータを使用してメッセージをルーティングする方法であって、
コンピューター可読通信媒体からメッセージを受信するステップと、
ルーティング情報の信頼性のないローカルキャッシュを使用して、処理ノード間の連携なしに、前記メッセージに対する可能性の高いルートを決定するステップと、
前記メッセージに対する前記可能性の高いルートに従って、前記メッセージをサーバーノードまたは別のルーターノードに向けて送るステップと、
前記メッセージが異なる処理ノードに送信されるべきであることを示すリダイレクトメッセージを前記サーバーノードまたは他のルーターノードから受信するステップと、
メッセージを示すための前記信頼性のないローカルキャッシュを、前記異なる処理ノードにルーティングされるべき前記メッセージの前記１つまたは複数の特性プロパティによって更新するステップと、
を含む方法。

【請求項９】
前記メッセージをサーバーノードに向けて送るステップが、前記メッセージを前記処理ノードのサーバーコンポーネントに向けて送るステップを含む、請求項８に記載の方法。
【請求項１０】
前記メッセージをサーバーノードに向けて送るステップが、前記処理ノードに存在するサービスインスタンスのローカルカタログを参照するステップを含む、請求項９に記載の方法。
【請求項１１】
前記メッセージに対する可能性の高いルートを決定するステップが、メッセージをルーティングするための負荷平衡機構を使用するステップを含む、請求項８に記載の方法。
【請求項１２】
前記サーバーノードまたは他のルーターノードからの前記リダイレクトメッセージが、特性プロパティと処理ノードとの間の１つまたは複数の追加の関連付けを含み、前記追加の特性プロパティが、現在処理されている前記メッセージに関係せず、前記１つまたは複数の追加の関連付けに従って前記信頼性のないローカルキャッシュを更新する動作をさらに含む、請求項８に記載の方法。
【請求項１３】
前記メッセージを前記異なる処理ノードに向けて送ることによって、前記リダイレクトメッセージにどのように応答するかを決定するステップをさらに含む、請求項８に記載の方法。
【請求項１４】
前記メッセージの送信元であるシステムにリダイレクトメッセージを提供することによって、前記リダイレクトメッセージにどのように応答するかを決定するステップをさらに含む、請求項８に記載の方法。
【請求項１５】
ディレクトリサービスプロトコルを使用して前記リダイレクトメッセージを解決するステップをさらに含む、請求項８に記載の方法。
【請求項１６】
発見プロトコルを使用して前記リダイレクトメッセージを解決するステップをさらに含む、請求項８に記載の方法。
【請求項１７】
コンピューティング環境において、ネットワーク中でメッセージをルーティングするように構成されたシステムであって、
１つまたは複数のプロセッサーと、
プロセッサーによって実行されてコンピューターモジュールを実現するように構成されたコンピューター実行可能命令を記憶したコンピューター可読記憶媒体とを備え、前記コンピューターモジュールが、
メッセージの１つまたは複数の特性プロパティを計算して、前記メッセージの処理のためのサービスにおけるサービスインスタンスに対する状態要件を決定するように構成された計算機モジュールと、
ネットワーク中の処理ノードにおいてまだ保持されていない前記ネットワーク中の全てのサービスインスタンスを記憶するインスタンスコーディネーターモジュールと、
前記メッセージを処理するための前記状態要件を満たす適切なサービスインスタンスを前記インスタンスコーディネーターから取得することを試みるように構成されたサービスノードモジュールであって、前記メッセージを処理するための前記状態要件を満たす適切なサービスインスタンスを取得する試みが、適切なサービスインスタンスの取得に成功したとき、前記メッセージが前記状態要件に基づいて処理され、前記サービスノードモジュールはさらに、前記インスタンスコーディネーターにも前記ネットワーク中の前記処理ノードのいずれにも適切なサービスインスタンスが存在しないと前記インスタンスコーディネーターが判定したときに、新しい適切なサービスインスタンスを生み出すように構成さ

れた、サービスノードモジュールと、
ルーティング情報の信頼性のないローカルキャッシュを使用して処理ノード間の連携なしに前記メッセージをリダイレクトして、前記メッセージの処理のための前記状態要件を満たす適切なサービスインスタンスを有するかまたは適切なサービスインスタンスをうまく取得することのできるサーバーノードに前記メッセージを到達させることを試みるように構成されたルーターモジュールと、
を含む、システム。
【請求項１８】
コンピューティング環境中の処理ノードにおいて、信頼性のないルーティングデータを使用してメッセージをルーティングする方法であって、
コンピューター可読通信媒体からメッセージを受信するステップと、
ルーティング情報の信頼性のないローカルキャッシュを使用して、処理ノード間の連携なしに、前記メッセージに対する可能性の高いルートを決定するステップと、
前記メッセージを前記処理ノードのサーバーコンポーネントに向けて送るステップと、
前記メッセージが異なる処理ノードに送信されるべきであることを示すリダイレクトメッセージを前記処理ノードから受信するステップと、
メッセージを示すための前記信頼性のないローカルキャッシュを、前記異なる処理ノードにルーティングされるべき前記メッセージの前記１つまたは複数の特性プロパティによって更新するステップと、
を含む方法。
【発明の詳細な説明】
【背景技術】
【０００１】
[0001]コンピューターおよびコンピューティングシステムは、現代生活のほぼあらゆる側面に影響を及ぼしてきた。コンピューターは一般に、仕事、休養、健康管理、移動、娯楽、家庭管理などに関わる。
【０００２】
[0002]さらに、コンピューティングシステムの機能は、ネットワーク接続を介して他のコンピューティングシステムに相互接続されるというコンピューティングシステムの能力によって強化することができる。ネットワーク接続は、以下のものに限定されないが、有線またはワイヤレスイーサネットを介した接続、セルラー接続、さらには、シリアル、パラレル、ＵＳＢ、または他の接続を介したコンピューター間接続を含むことができる。これらの接続により、コンピューティングシステムは、他のコンピューティングシステムにおけるサービスにアクセスすることができ、また他のコンピューティングシステムから素早く効率的にアプリケーションデータを受け取ることができる。
【０００３】
[0003]メッセージ処理システムを使用して、ネットワーク化されたコンピューター間でメッセージをルーティングすることができる。メッセージ処理システムの単純な一構成は、クライアントマシンから直接にメッセージを受信する単一のマシン上で稼動するサービスを含む。メッセージ数または１メッセージの平均処理コストを増やすことによってサービスによる必要リソース量が増加するのに伴い、処理負荷に対処できる単一のマシンを構築することは最終的に、非現実的に高価になる場合がある。一般的な慣行の１つは、多くのより安価なマシン間で処理負荷が分散されるように、サービスの複数のインスタンスをいくつかのマシンにまたがって実行することである。処理負荷を多くのマシン間で分散させることはまた、クリティカルポイントをなくしてシステムの全体的な信頼性を改善するためにも用いられる場合がある。
【０００４】
[0004]システムの並列効率は、より多くのマシンを追加することがどれだけ効果的かについての尺度となる。並列効率が１００％のときは、マシン数を２倍にすればマシンごとの処理負荷は半分になる。並列効率は、より小さいパーセンテージであるか、０であるか

、さらには負である場合もある。負の並列効率は、マシンを追加するオーバーヘッドが、総処理力における利得よりも大きいことを意味する。処理マシン間の連携は、低い並列効率の一般的な原因の１つである。例えば、いくつかのサービスを含むトポロジ中でメッセージがサービスにルーティングされる前に、このサービスがメッセージの処理に必要な状態情報を有することを確実にする必要がある場合がある。例示として、トポロジ中の特定のサービスにおいて、特定の電子商取引ショッピングカートに関する状態情報を含む状態バッグをロードすることができる。ショッピングカートを更新する（新規商品を追加する、商品を削除する、注文を完了するなど）メッセージを、トポロジにおいて受信することができる。トポロジ中の複数のサービスがショッピングカート機能を扱うことができる場合、どのサービスが当該の特定のショッピングカートに関する状態バッグを含むかを決定することがトポロジにおいて必要であろう。

【０００５】

[0005]メッセージを独立して処理することのできるサービスの例が多くある。メッセージ独立性により、メッセージを任意の利用可能な処理ノードに送信することができる。また、メッセージと処理ノードとの間の自明な関連付けを有するサービスの例もある。自明な関連付けにより、低い連携レベルで、入来メッセージを適切な処理ノードにルーティングすることが許される。自明な関連付けの一例は、ネットワーク接続が、共に処理しなければならない１組のメッセージの境界とちょうど等しいものである。しかし、メッセージとサービスとを連携させるために関連付けオーバーヘッドが必要な場合のある例が多くある。

【０００６】

[0006]本明細書で請求する主題は、前述のようないずれかの不都合を解決する実施形態、または前述のような環境のみで動作する実施形態に限定されない。そうではなく、この背景は、本明細書に述べるいくつかの実施形態を実践できる例示的な技術領域の１つを説明するために提供するに過ぎない。

【発明の概要】

【０００７】

[0007]本明細書に述べる一実施形態は、信頼性のないルーティングデータを使用してメッセージをルーティングする方法を対象とする。この方法は、コンピューティング環境中の処理ノードにおいて実践することができる。この方法は、コンピューター可読通信媒体からメッセージを受信することを含む。メッセージの１つまたは複数の特性プロパティが計算されて、メッセージの処理のためのサービスにおけるサービスインスタンスに対する状態要件が決定される。メッセージを処理するための状態要件を満たす適切なサービスインスタンスを取得することが試みられる。メッセージを処理するための状態要件を満たす適切なサービスインスタンスを取得する試みが、適切なサービスインスタンスの取得に成功したときは、メッセージは状態要件に基づいて処理される。この方法は、メッセージを処理するための状態要件を満たす適切なサービスインスタンスを取得する試みが、適切なサービスインスタンスの取得に成功しないと判定することを含む。結果として、メッセージは、ルーティング情報の信頼性のないローカルキャッシュを使用して処理ノード間の連携なしにリダイレクトされ、メッセージの処理のための状態要件を満たす適切なサービスインスタンスを有するかまたは適切なサービスインスタンスをうまく取得することのできるサーバーノードにメッセージを到達させることが試みられる。

【０００８】

[0008]この概要は、詳細な説明でさらに後述する概念の精選を単純化した形で紹介するために提供する。この概要は、特許請求する主題の鍵となる特徴または必須の特徴を識別するものとはせず、また、特許請求する主題の範囲を決定する助けとして使用されるものともしない。

【０００９】

[0009]追加の特徴および利点は、後続の記述に示すが、これらはこの記述から一部は自明であろうし、あるいは本明細書の教示の実践によって知ることができる。本発明の特徴

および利点は、添付の特許請求の範囲で特に指し示す手段および組合せによって理解および獲得することができる。本発明の特徴は、後続の記述および添付の特許請求の範囲からより完全に明らかになるであろうし、あるいは以下に示す本発明の実践によって知ることができる。

【００１０】

[0010]前述および他の利点および特徴を得ることのできる方式について述べるために、上に簡潔に述べた本主題のより具体的な記述を、添付の図面に示す特定の実施形態に関して提供する。これらの図面は典型的な実施形態のみを描写し、したがって範囲を限定するものと考えるべきではないことを理解した上で、添付の図面を使用して実施形態をさらに具体的かつ詳細に記述および説明する。

【図面の簡単な説明】

【００１１】

【図１】[0011]メッセージを処理するためのルーターノードおよびサービスノードを含むネットワーク環境を示す図である。

【図２Ａ】[0012]メッセージについて特性を計算することのできる環境を示す図である。

【図２Ｂ】[0013]クエリ最適化モジュールを示す図である。

【図２Ｃ】[0014]中間結果の計算前および計算後のデータ変換を示す図である。

【図２Ｄ】[0015]いくつかの実施形態での追加特性計算の詳細を示す図である。

【図３】[0016]メッセージを処理する方法を示す図である。

【図４】[0017]信頼性のないルーティングデータを使用してメッセージをルーティングする方法を示す図である。

【発明を実施するための形態】

【００１２】

[0018]いくつかの実施形態は、メッセージと処理ノードとの間に何らかの非自明な関連付けがあるサービスを対象とする。例えば、非自明な関連付けの一例は、メッセージ中の内容の位置およびフォーマットが、アプリケーションによって定義されるプロトコルに依存するような、かつ／または、メッセージ中の内容の値が、前のメッセージ中に見られる内容を反映するような、メッセージ中の内容に基づくものである。

【００１３】

[0019]いくつかの実施形態は、ルーティングエラーに対する耐性を処理ノードに持たせて楽観的ルーティングを採用することにより、非自明な関連付けのための連携レベルを低下させる。後の処理中に、誤ったルーティング決定が検出されることになる。誤ってルーティングされたメッセージに応答して、部分的に完了した作業が廃棄され、メッセージは再び楽観的方式でルーティングされる。加えて、メッセージを誤ってルーティングしたノードに情報を返すことができ、それにより、このノードによってルーティングされる将来のメッセージが正しくルーティングされる可能性が高くなるようにする。ルーティングエラーに遭遇するせいで、いくつかのメッセージは、処理すべき作業をより多く必要とする場合があるが、楽観的ルーティングは頻繁に正しくすることができ、ノード間の連携を低減させることができるので、メッセージ処理の平均コストは減少する。

【００１４】

[0020]言及したように、いくつかの実施形態は、最良推量を用いることにより、処理ノード間の連携なしにメッセージの特性プロパティに基づいてメッセージをルーティングすることを可能にする。これは、ルーティングデータの信頼性のないローカルキャッシュを維持することによって達成することができる。ルーティングデータの信頼性のないローカルキャッシュは、例えば、メッセージ特性、メッセージメタデータ特性、または他の特性と、サービスノードとの関連付けを含むことができる。さらに、ルーティングデータは、受信されたリダイレクトメッセージに基づく更新を含むことができる。例えば、メッセージを誤って受信するノードは、そのデータに対する正しいノードを識別する情報を提供できるものとすることができる。実施形態はまた、元のメッセージ送達機構または特性に基づいてリダイレクトメッセージをどのように扱うかを決定する機能を含むことができる。

さらに、ルーティングは、最良推量ルーティングの定義の一部として、負荷平衡機構を組み込むことを含むことができる。

【００１５】

[0021]いくつかの実施形態は、メッセージの特性プロパティに基づいてインスタンスコーディネーターによって実施される原子ｃｒｅａｔｅ－ｌｏａｄプリミティブを含むことができる。例えば、原子ｃｒｅａｔｅ－ｌｏａｄ演算の一部として、特性とサービスインスタンスとの間に追加の関連付けを供給することができる。関連付けを供給するために実施されるルックアップは、複数の特性に基づくことができる。さらに、サービスインスタンスのローカルカタログを使用して、ルックアップの最適化を達成することができる。これらの実施形態のより詳細な記述は、本明細書においてさらに以下に含まれる。

【００１６】

[0022]実施形態は、失敗したｃｒｅａｔｅ－ｌｏａｄ演算の結果としてリダイレクトメッセージを生成する機能を含むことができる。いくつかの実施形態では、ディレクトリサービスまたは発見プロトコルを使用して、リダイレクトアドレスを解決することができる。さらに、他のサービスインスタンスに対するルーティングデータを、リダイレクトメッセージの一部として含めることができる。

【００１７】

[0023]次に図１を参照すると、相互接続された処理ノードのネットワーク１００が示されている。処理ノードは、適切なプロセッサー、メモリーデバイス、コンピューター記憶装置、本明細書において後で定義されるようにコンピューター可読媒体上に実装されたコンピューター実行可能命令などを含めた、コンピューティングハードウェアおよびソフトウェアを使用して実現することができる。処理ノードのいくつかは、アプリケーションサービスのインスタンスを実行することができる。いくつかの処理ノードは、本明細書ではサーバーノードと呼び、図１に示す例示的なサーバーノード１０２および１０４によって示す。いくつかの処理ノードは、処理ノード間でメッセージをルーティングすることができる。これらのノードは、本明細書ではルーターノードと呼び、ルーターノード１０６によって示す。時の経過につれて、また連携なしで、ノードの数またはノード間の接続またはノードの役割（例えばルーター機能またはサーバー機能の獲得または喪失）に変化がある場合がある。したがって、ネットワークのどんな描写も、特定の時点のスナップショットを表す。処理ノードは、必ずしも永続的にルーターとサーバーとに区分化されるとは限らない。特に、処理ノードは、両方として機能する場合もあり（ルーター－サーバーノード１１０によって示す。これを本明細書ではルーターノードまたはサーバーノードと呼ぶ場合がある）、あるいはいずれとしても機能しない場合もある。

【００１８】

[0024]１つまたは複数のサービスインスタンス１１６（本明細書では一般に１１６として示し、具体的に１１６－Ｘとして示す。Ｘは特定のサービスインスタンス１１６を示すための変数である）を含むアプリケーションが、ネットワーク１００上で実行される。アプリケーションは、サービスインスタンス１１６を進行中に作成および破壊することができる。したがって、サービスインスタンス１１６の現在のセットは、絶えず変化する場合がある。サービスインスタンス１１６は、メッセージを処理し、時の経過につれてインスタンス状態を蓄積することができる。この例では、サービスインスタンス１１６は、サーバーノードのうちの１つのみにしか同時に存在しない。サービスインスタンス１１６がどのサーバーノードにも位置しないとき、サービスインスタンス１１６は、全てのサーバーノードがアクセスできるインスタンスコーディネーター１１２によって保持される。インスタンスコーディネーター１１２の典型的な一実装形態は、サービスインスタンス１１６のインスタンス状態を耐久的に記憶するデータベースである。

【００１９】

[0025]アプリケーションは、様々な処理ノードにおいて外部ソースからメッセージ１１４を継続的に受信することができる。メッセージの処理は、既存のサービスインスタンス１１６のインスタンス状態を必要とする場合があり、あるいは前にメッセージを処理した

ことのない新しいサービスインスタンス１１６の白紙状態を必要とする場合があり、あるいはインスタンス状態要件を有さない場合がある。処理要件は、アプリケーションの定義の一部であることがあり、したがって、アプリケーションごとに異なる場合があり、メッセージから容易に明らかでない場合がある。いくつかの実施形態の機能の１つは、処理ノード間の過度の連携を必要とせずに、メッセージの処理要件を満たすサーバーノードに向けてメッセージを送ることである。いくつかの実施形態では、これは、メッセージ１１４の処理要件を決定するのに使用できる特性を計算することによって達成することができる。

【００２０】

[0026]特性計算は、例えば、本明細書と同時に出願されその全体が参照により本明細書に組み込まれる「Ｑｕｅｒｙ－Ｏｒｉｅｎｔｅｄ Ｍｅｓｓａｇｅ Ｃｈａｒａｃｔｅｒｉｚａｔｉｏｎ」という名称の米国出願第１２／２０３，７９０号に記載の技法を使用して実施することができる。図２Ａ～２Ｄに、いくつかの実施形態でどのように特性計算を実施できるかを示す。ここで図２Ａを参照すると、一例が示されている。図２Ａは、クエリエンジン２０２を示す。クエリエンジン２０２は、クエリ２０４を処理する機能を含み、クエリ２０４は、メッセージデータを含むメッセージ１１４または他のソースから入手可能な非メッセージデータ２１０など、様々なデータソースに対するクエリである。特に、いくつかの実施形態は、メッセージ内容、メタデータ、または他の情報に対するクエリを使用してメッセージ特性を指定できるように実践することができる。クエリエンジンは、様々な言語２０６に対するサポートを含むことができる。ある特定の例では、クエリは、クエリ言語としてＸＰａｔｈ表現を使用して定式化することができる。

【００２１】

[0027]ＸＰａｔｈなどのクエリ言語２０６はしばしば、限られた種類のフォーマットの、限られた種類のソースからの情報にアクセスするネイティブ機能を有するが、他の情報にアクセスする機能はネイティブに含まない。例えば、ＸＰａｔｈは、ＸＭＬを使用してフォーマットされたメッセージなど、ＸＭＬ構造化されたデータ構造の情報にアクセスするネイティブ機能を含むが、他のサービスからの他の情報を決定する機能は含まない場合がある。それにもかかわらず、拡張機能２１２を含めることによってクエリ言語を拡張して、他のサービスにアクセスする機能を含むようにすることができる。ＸＰａｔｈクエリ言語では、拡張機能はセレクターと呼ばれる。加えて、いくつかの実施形態は、クエリ言語への拡張機能を使用して種々の記憶位置へのアクセスを正規化する機能を含むことができる。いくつかの実施形態では、種々の記憶位置へのアクセスの正規化は、相互に合意されたデータ構造を使用することができる。クエリをマージして同時にまたは並行して実行することにより、後でより詳細に述べるように、同じメッセージについての複数の特性の計算の最適化を実施することができる。

【００２２】

[0028]図２Ａに示すように、情報ソースは、メッセージデータを含むメッセージ１１４を含むことができる。メッセージは、エンベロープデータ、メッセージ本文中のデータ、メッセージのヘッダ中のデータなどの情報を含むことができる。上に言及したように、クエリエンジン２０２は、メッセージデータを抽出する機能を備えることができる。例えば、一実施形態では、クエリエンジンは、ＸＭＬフォーマット化されたメッセージからデータを抽出するために、ＸＰａｔｈクエリ言語をサポートする機能を備えることができる。他のクエリ言語２０６を共にまたは代替として使用することもできる。特に、クエリエンジン２０２はまた、様々なアプリケーションプログラミングインターフェイス（ＡＰＩ）２１４を呼び出す機能を備えることもできる。ＡＰＩ２１４は、情報ソースと対話してソースからデータを得るためのプログラム済み機能を備える。特に、言語２０６は、いくつかの点でＡＰＩと考えることができる。

【００２３】

[0029]図２Ａはさらに、非メッセージデータ２１０も示す。非メッセージデータは、いくつかの異なるソースのうちのいずれか１つからのデータとすることができ、メッセージ

データに関するメタデータ、またはメッセージデータを直接に提示しない他のデータを含むことができる。メッセージ１１４中のデータに関連するメタデータは、メッセージ１１４の送信に使用されたプロトコルを示すプロトコルデータ、環境データ、ローカルプロパティ、時刻などの情報を含むことができる。

【００２４】

[0030]前に言及したように、図２Ａは、クエリエンジン２０２がデータソースに対してクエリ２０４を実施することを示す。クエリ２０４に基づいてクエリエンジン２０２は中間結果２１６を生成するが、中間結果２１６は、データのインスタンス値とすることができる。中間結果２１６は、データのテーブル、または他の形のデータを含むことができる。例えば、中間結果２１６は、特定の時刻（メッセージ１１４に関連する場合もありそうでない場合もある）、メッセージ１１４の送信に使用された特定のプロトコル、または他の情報などの情報を含むことができる。中間結果は通常、単位のない結果ではなく、何らかの特定の単位を表す。例えば、中間結果２１６は、時刻単位、プロトコル単位、トランスポート単位、または何らかの他の特定の単位を表すことができる。加えて、中間結果は、１つまたは複数の異なるデータ型であってもよい。例えば、中間結果は、整数、浮動小数点、文字列、または他のデータ型とすることができる。加えて、１組の中間結果が、種々のデータ型の混合を有することもできる。例えば、時間は１つまたは複数の整数として表現され、プロトコルは１つまたは複数の文字列として表現されてもよい。時間整数とプロトコル文字列の両方が、同じ１組の中間結果２１６に含まれてよい。

【００２５】

[0031]中間結果２１６を使用して、特性計算モジュール２２０によって特性２１８を生み出すことができる。特性２１８は、例えば、中間結果２１６に基づいて数を計算するためのハッシュアルゴリズムまたは他の数値方法を使用して計算された数とすることができる。例えば、一実施形態では、特性２１８は、大域的に一意の識別子を表す単位なしの１２８ビットのハッシュ数とすることができる。特性計算モジュール２２０は、ハッシュまたは他の表現、例えば数値表現を計算するように構成された、コンピューターハードウェアおよびソフトウェアを使用して具体化することができる。

【００２６】

[0032]以下により詳細に論じるように、いくつかの実施形態は、メッセージ特性２１８の計算とメッセージングインフラストラクチャとの間で連携が生じる場合に実践することができる。特に、メッセージングインフラストラクチャは、クエリ２０４について潜在的に供給できる情報をカタログすることができる。例えば、メッセージングインフラストラクチャは、トランスポートに関する情報やプロトコルに関する情報などを提供できる場合がある。メッセージングインフラストラクチャは、特定の時点における情報の利用可能性を約束することができる。いくつかの実施形態では、この約束は、メッセージングインフラストラクチャにおける、何らかの機能に、または何らかのアクションの実施に関係する。特性計算モジュール２２０における特性計算の前に、クエリ２０４の分析を実施して、どんな情報が必要になるかを決定することができる。情報の利用可能性に基づく制約に従ってより好都合な時点で特性計算のための算出を実施するために、特性計算の最適化を実施することができる。

【００２７】

[0033]以下により詳細に論じるように、いくつかの実施形態は、クエリの前および／または後に情報の変換が実施される場合に実践することができる。

[0034]図２Ａをもう一度参照すると、細部を含むより詳細な例が示されている。特性を計算することが望ましいメッセージ１１４を考えてみる。このメッセージ１１４の存在は、メッセージ１１４がどのように生成されるかまたは生成されたかに関わらず、前もって仮定することができる。したがってこのメッセージは、送信されつつあるメッセージ、受信されつつあるメッセージ、またはもしかすると、メッセージング操作とは何の関係もなく何もないところから作成されたメッセージですらある可能性がある。メッセージは、様々なフォーマットで表される場合がある。例として、シンプル・オブジェクト・アクセス

・プロトコル（ＳＯＡＰ）１．２フォーマットを使用して表されるメッセージを考えてみる。このようなメッセージは、メッセージエンベロープ、メッセージ本体、および任意の数のメッセージヘッダのための記憶位置を有することになる。メッセージにはまた、ローカルメッセージプロパティ、送達プロパティ、またはアンビエント環境中の情報など、メッセージエンベロープ内に含まれないメタデータが関連する場合がある。このメタデータは、２１０に示す非メッセージデータによって表すことができる。したがって、データのソースは、メッセージ内からの情報のソースか、あるいはメッセージ外からの情報のソースと呼ぶことができる。

【００２８】

[0035]メッセージ１１４の特性を計算するために、全ての利用可能な情報ソースを利用することができる。特性の計算はしばしば、利用可能な情報のサブセットのみを必要とすることになる。このサブセットは、１つまたは複数のクエリ２０４を含むクエリ仕様２０５によって記述される。各クエリは、識別子およびクエリプロシージャを含む。クエリプロシージャは、利用可能な情報からどのように値が抽出されるかを定義する。

【００２９】

[0036]クエリ仕様２０５の例として、一実施形態では、クエリプロシージャは、ＸＰａｔｈ表現を使用して指定される。例えば、メッセージはＳＯＡＰフォーマットの購入注文とすることができ、この断片は以下のとおりである。

```
＜ｓ：Ｅｎｖｅｌｏｐｅ＞
  ＜ｓ：Ｈｅａｄｅｒ＞
    ．．．ｈｅａｄｅｒ ｄａｔａ ｉｎｃｌｕｄｅｄ ｉｎ ｔｈｅ ｍｅｓｓａｇｅ．．．
  ＜／ｓ：Ｈｅａｄｅｒ＞
  ＜ｓ：Ｂｏｄｙ＞
    ＜ｐｏ：ＰｕｒｃｈａｓｅＯｒｄｅｒ ｐｕｒｃｈａｓｅＯｒｄｅｒＮｕｍｂｅｒ＝“１２３”＞
    ．．．ｐｕｒｃｈａｓｅ ｏｒｄｅｒ ｄａｔａ ｄｅｆｎｅｄ ｂｙ ｔｈｅ ａｐｐｌｉｃａｔｉｏｎ．．．
    ＜／ｐｏ：ＰｕｒｃｈａｓｅＯｒｄｅｒ＞
  ＜／ｓ：Ｂｏｄｙ＞
＜／ｓ：Ｅｎｖｅｌｏｐｅ＞
```

[0037]ＸＰａｔｈ表現「／ｓ：Ｅｎｖｅｌｏｐｅ／ｓ：Ｂｏｄｙ／ｐｏ：ＰｕｒｃｈａｓｅＯｒｄｅｒ／@ｐｕｒｃｈａｓｅＯｒｄｅｒＮｕｍｂｅｒ」が、メッセージの一部を指定する。この例では、このＸＰａｔｈ表現は、Ｅｎｖｅｌｏｐｅという名前の要素内の、Ｂｏｄｙという名前の要素内の、ＰｕｒｃｈａｓｅＯｒｄｅｒという名前の要素上の、ｐｕｒｃｈａｓｅＯｒｄｅｒＮｕｍｂｅｒという名前の属性の値を指定する。この例では、このＸＰａｔｈ表現は「ＰＯＮｕｍｂｅｒ」と名付けられて、識別子ＰＯＮｕｍｂｅｒと、このＸＰａｔｈ表現の値を求める結果として得られるファクト、すなわち購入注文単位１２３を表す数１２３との間の、関連付けが生み出される。

【００３０】

[0038]クエリ２０４を含むクエリ仕様２０５と、必要な情報ソース（メッセージ１１４、および／または、非メッセージデータ２１０を生成するソースへのアクセスなど）とをクエリエンジン２０２に供給すると、クエリエンジン２０２は、中間結果２１６中に示される名前付きクエリ結果のテーブルを計算する。

【００３１】

[0039]図示の例では、特性２１８の計算は名前付きクエリ結果２１６で定義されて、どのように情報がアクセスまたは編成されたかということから計算プロセスが抽象化される。新しい情報ソースを既存の情報ソースと統一することによって、あるいは新しいアクセスメソッドでクエリエンジンを拡張することによって、新しい情報ソースをシステムに追加することができる。例えば、標準的なＸＰａｔｈ言語は、メッセージデータへのアクセ

スを提供するだけである。拡張機能２１２によって示すように、ＸＰａｔｈ言語を新しい関数で拡張して、非メッセージデータにアクセスすることができる。

【００３２】

[0040]一実施形態では、ＨＴＴＰ　Ｒｅｆｅｒｅｒヘッダは、メッセージデータの一部ではないが、ＸＰａｔｈ表現「ｚ：ＧｅｔＰｒｏｔｏｃｏｌＤａｔａ（）／Ｒｅｆｅｒｅｒ」を使用して非メッセージデータ２１０の一部を指定することで、同様にしてこのヘッダにアクセスすることができる。この場合、プロトコルデータ中のＲｅｆｅｒｅｒプロパティの値はメッセージ内に含まれない。ＳＭＴＰ　Ｆｒｏｍヘッダは異なる情報ソースから来るが、このヘッダにもまた、ＧｅｔＰｒｏｔｏｃｏｌＤａｔａ関数を使用してアクセスすることができる。このように、情報を同じまたは異なるアクセスメソッドに分類することは、開発者の都合の良いように行うことができる。

【００３３】

[0041]ここで図２Ｂを参照して、クエリエンジン２０２の追加の詳細について、特にクエリ処理の最適化に関して次に述べる。メッセージデータおよび非メッセージデータ２０８／２１０などの同じ情報ソースに対して複数のクエリ２０４が実施される場合、これらのクエリを１つずつではなく共に扱うと、クエリの集合をより効率的に実施することがしばしば可能である。一実施形態でこれを行うために、クエリエンジン２０２はクエリ最適化モジュール２２２を備える。クエリ最適化モジュール２２２は、まず元のクエリ仕様２０５を最適化済みクエリ仕様２２４に変換した後で、言語２０６（図２Ａに示すＡＰＩ２１４）を使用して最適化済みクエリ仕様２２４を実行する。最適化済みクエリ２２４は、処理されると、同じクエリ結果テーブル２１６を生成する。

【００３４】

[0042]一実施形態では、クエリエンジン２０２のクエリオプティマイザ２２２が、共通のサブ表現を有するクエリを結合して、単一の共通サブ表現の値が１回だけ求められるようにする。したがって、２つのクエリ「／ｓ：Ｅｎｖｅｌｏｐｅ／ｓ：Ｂｏｄｙ／ＰｕｒｃｈａｓｅＯｒｄｅｒ１」および「／ｓ：Ｅｎｖｅｌｏｐｅ／ｓ：Ｂｏｄｙ／ＰｕｒｃｈａｓｅＯｒｄｅｒ２」を含むクエリ仕様２０５に対して作用するクエリエンジン２０２は、両方のクエリを満たすために、メッセージ１１４のＥｎｖｅｌｏｐｅおよびＢｏｄｙ要素の中を１回走査するだけでよい。

【００３５】

[0043]次に図２Ｃを参照すると、処理の前後のデータ変換に関する、クエリエンジン２０２の追加の特徴が示されている。図示の実施形態では、クエリエンジン２０２は、他のコンポーネント２２８および２３０と共に処理パイプライン２２６の一部を構成する。これらのコンポーネント２２８および２３０は、それぞれエンジンの入力および出力に作用する。コンポーネント２２８においては、情報ソースがエンジンによって読み取られる前に１つまたは複数の変換を情報ソースに適用することができ、コンポーネント２３０においては、特性２１８（図２Ａ参照）が計算される前に１つまたは複数の変換をクエリ結果に適用することができる。情報ソース中の各ファクト、および各名前付き結果に、個別に作成された変換が適用されてもよく、あるいは、一群のファクトまたはクエリ結果に変換が適用されてもよい。

【００３６】

[0044]アプリケーションはしばしば、特性２１８を計算するのに好ましい時点を有する。アプリケーションが、行われている決定のタイプに応じてできるだけ遅くまたはできるだけ早く特性２１８を計算したいことはよくある。しかし、アプリケーションは、全ての必要な情報が利用可能になるまでは特性２１８を計算できない場合がある。この競合の一例は、メッセージを送信するときに生じる。メッセージの送信に対するいずれかの応答が確認される前に特性がわかるように、特性をできるだけ早く計算するのが望ましい場合がある。しかし、特性を計算するのに必要な情報は、メッセージが部分的にまたは完全に送信されるまで利用可能でない場合がある。ずっと後になるまで利用可能にならない情報の一例は、メッセージが配線上に書き込まれるときに送達システムによって割り当てられる

メッセージ識別子である。

【００３７】

[0045]次に図２Ｄを参照すると、これらの問題に対処する一実施形態の例が示されている。競合について推論するためには、どの情報が特性計算によって使用されることになるか、およびその情報がいつ利用可能になるかがわかるべきである。メッセージ１１４がアプリケーション２３２によって送信される前に、メッセージングインフラストラクチャ２３４の内部が観察されて、この特定の構成が生成することになる様々な情報が識別される。メッセージングインフラストラクチャ２３４はまた、いつ各ファクトが利用可能になるかに関する１つまたは複数のステートメントを作成することもできる。ステートメントは、ファクトが処理の特定の時点または段階で利用可能になることの約束とすることができる。メッセージ１１４が送信される前にはまた、クエリ仕様２０５（図２Ａ参照）の内部を観察して、この特定のクエリ仕様２０５が要求することになる様々な情報を決定することもできる。

【００３８】

[0046]図２Ｄには、時間軸Ｔに対してメッセージングインフラストラクチャ２３４を示す。時間軸Ｔは、下方向に進む時間を示す。T_{Start}において、メッセージ１１４がアプリケーション２３２からメッセージングインフラストラクチャ２３４に送信される。いくつかの実施形態では、メッセージ１１４が送信される頃、クエリ仕様２０５中のクエリ２０４によって必要とされることになる情報識別子のリストがメッセージ１１４に関連付けられる。特に、メッセージ１１４の送信前、またはメッセージ１１４の送信時、またはいくつかの実施形態ではメッセージ１１４の送信後に、情報識別子のリストを関連付けるように実施形態を実施することができる。加えて、メッセージ１１４は、クエリエンジン２０２および特性計算モジュール２２０（図２Ａ参照）を呼び出すコールバックに関連付けられる。コンポーネント２３６－１～２３６－Ｎが、メッセージ１１４に作用することができる。コンポーネント（ここでは一般に２３６と呼び、具体的に２３６－Ｘと呼ぶ。Ｘは特定のコンポーネントを識別する番号である）がメッセージ１１４に作用するとき、これらのコンポーネントは、概念的には、識別された各ファクトが利用可能になるのに伴って、クエリ２０４によって必要とされることになる情報識別子のリストにチェックマークを加える。一実施形態では、ファクトの値が得られる特定のプロセスを実行することが可能なとき、ファクトは利用可能になる。このプロセスは、単純にファクトの事前計算済みの値を返す場合もあり、あるいは追加の計算の実施を必要とする場合もある。したがって、ファクトが特定の時点でクエリエンジン２０２に利用可能になることがあっても、ファクトの値は、厳密な意味では、クエリエンジン２０２が後の時点でファクトの値を要求することをもし選択するならばそれまでは、クエリエンジン２０２にはわからないかもしれない。識別された情報の全てが利用可能になると、コールバックを呼び出して特性計算を完了することができる。図示の例では、図２Ｄは、メッセージ１１４に関する情報がクエリエンジン２０２に利用可能になるのを示す。時間T_1で、コンポーネント２３６－１によって提供された情報が、クエリエンジン２０２に利用可能になる。時間T_2で、コンポーネント２３６－２によって提供された情報が、クエリエンジン２０２に利用可能になる。時間T_Nで、コンポーネント２３６－Ｎ（これは、任意の数のコンポーネント２３６をメッセージングインフラストラクチャ２３４中で実装できることを意味する）によって提供された情報が、クエリエンジン２０２に利用可能になる。

【００３９】

[0047]時間軸Ｔは、メッセージを通信配線上に送信することなどによってメッセージ１１４がメッセージングインフラストラクチャ２３４の外に送信されるのを表す時点$T_{Transmit}$を含む。通信配線は、ネットワークケーブルまたはワイヤレストランスポート媒体を含めた種々の数の媒体のうちの任意の１つとすることができる。計算の完了は、コンポーネント２３６によってなされる約束に応じて、メッセージが送信されるよりも早くまたは遅く起こる場合がある。

【００４０】

一実施形態では、コールバックの完了を用いて、メッセージの送信と受信との間の競合が解決される。アプリケーション２３２は、前に送信されたメッセージ１１４の特性２１８に依存するかもしれないどんな受信メッセージも、これらの特性が全て計算されるまでは処理しないようにする。

【００４１】

[0048]次に、以下の考察では、いくつかの方法および実施できる方法動作に言及する。方法動作を、特定の順序で論じるか、または特定の順序で行われるようにフローチャート中に示す場合があるが、特定の順序付けは特に指定がない限り必ずしも必要とされず、あるいは、ある動作がその実施前に別の動作が完了することに依存するので特定の順序付けが必要とされることに留意されたい。

【００４２】

[0049]次に図３を参照すると、図１のノード１０２、１１０、および１０４などのサーバーノードの動作が示されている。図３は、方法３００を示す。方法３００は、メッセージを受信する動作（動作３０２）を含む。メッセージ（例えばメッセージ１１４）は、ネットワーク媒体、コンピューターバス、または他の通信媒体など、コンピューター可読通信媒体から受信することができる。

【００４３】

[0050]方法３００はさらに、メッセージの特性プロパティを計算して、メッセージの処理のためのサービス（サービス１０２、１０４、１１０のうちの１つなど）におけるサービスインスタンス（サービスインスタンス１１６のうちの１つなど）に対する状態要件を決定すること（動作３０４）を含む。特性プロパティの計算は、いくつかの実施形態では、上記の図２Ａ～２Ｄおよび付随する記述において例示したように実施することができる。

【００４４】

[0051]方法３００はさらに、メッセージ１１４を処理するための状態要件を満たす適切なサービスインスタンス（例えばサービスインスタンス１１６）の取得を試みることを含む。３０８で、判定ブロックが、この試みが成功したか否かに応じて実施される異なる動作を示す。メッセージを処理するための状態要件を満たす適切なサービスインスタンス１１６を取得する試みが、適切なサービスインスタンス１１６の取得に成功したときは、メッセージは状態要件に基づいて処理される（動作３１０）。あるいは、方法３００は、メッセージを処理するための状態要件を満たす適切なサービスインスタンス１１６を取得する試みが、適切なサービスインスタンス１１６の取得に成功しないと判定し、結果として、ルーティング情報の信頼性のないローカルキャッシュを使用して処理ノード間の連携なしにメッセージ１１４をリダイレクトして（動作３１２）、メッセージの処理のための状態要件を満たす適切なサービスインスタンス１１６を有するかまたは適切なサービスインスタンス１１６をうまく取得することのできるサーバーノードにメッセージ１１４を到達させるよう試みることを含むことができる。

【００４５】

[0052]図３には示されていないが、方法３００はさらに、サービスインスタンス１１６から独立した、メッセージ１１４に対する処理を実施することを含んでもよい。サービスインスタンス１１６から独立した処理の例は、静的に構成されたプロトコルおよびメッセージ変換を実施することである。

【００４６】

[0053]次に方法３００の細部をより詳しく例示するが、メッセージがノード１０２、１０４、または１１０などのサーバーノードに到着すると、サーバーノードがメッセージ１１４の処理要件を満たすという初期推定を行うことができる。メッセージ１１４に対していくらかの量の処理が実施された後、サーバーノードは、これ以上の処理がサービスインスタンス１１６を必要とするという点に到達する。

【００４７】

[0054]適切なサービスインスタンス１１６を識別するために、サーバーノードは、メッ

セージとローカルに利用可能な状態とを使用して、しかし処理ノード間の連携なしで、特性プロパティを計算することができる。例えば、メッセージ１１４がサーバーノード１１０に到着した場合、サーバーノード１１０は、メッセージ１１４とローカルに利用可能な状態とを使用して、しかし他の処理ノード１０６、１０２、および１０４間の連携なしで、特性プロパティを計算することができる。特性計算はアプリケーションによって定義される。特性は、メッセージ送達プロセス、メッセージの一部、さらにはメッセージ全体そのものからの情報である場合がある。例として、サーバーノードは、メッセージが特定のフォーマットの購入注文要求であることを識別し、この購入注文フォーマットが購入注文識別子を固定位置に含むことを識別し、この購入注文識別子を抽出して特性を形成することができる。

【００４８】

[0055]次いでサーバーノードは、インスタンスコーディネーター１１２と協議して、特性に対する適切な状態を有する適切なサービスインスタンス１１６を得る。この協議は、インスタンスコーディネーター１１２によって原子的に実施されるいくつかの動作を含むことができる。

【００４９】

[0056]特に、協議は、この特性に関連するサービスインスタンス１１６がすでに存在するかどうか判定することを含むことができる。例えば、インスタンスコーディネーター１１２は、適切なサービスインスタンス１１６がインスタンスコーディネーター１１２に、またはネットワーク１００中のいずれかの処理ノードに存在するかどうか判定することができる。特性を有するサービスインスタンス１１６が存在しない場合は、新しいサービスインスタンス１１６が生み出され、このサービスインスタンス１１６が特性に関連付けられる。特性を有するサービスインスタンス１１６が存在する場合は、協議は、どこにこのサービスインスタンス１１６が存在するかを決定することを含むことができる。サービスインスタンス１１６は、要求元サーバーノード（この例ではサーバーノード１１０）にすでに存在する場合がある。サービスインスタンス１１６がインスタンスコーディネーター１１２に存在する場合は、サービスインスタンス１１６は要求元サーバーノード１１０に転送される。サービスインスタンス１１６が、サーバーノード１０２または１０４など、他の何らかのサーバーノードに存在する場合は、インスタンスコーディネーター１１２は、ロード要求を拒否してエラーメッセージをサーバーノード１１０に送信することができる。エラーメッセージは、サービスインスタンス１１６が現在存在するサーバーノードの識別子（例えば１０２または１０４）を含む。

【００５０】

[0057]実施形態はまた、サービスインスタンス１１６に対してロックを設置できる環境で実施することもできる。これらの実施形態のいくつかでは、適切なサービスインスタンス１１６がネットワーク中の処理ノードのうちの１つによってロックされているとインスタンスコーディネーター１１２が判定したときに、ロック所有側に関する情報を要求元処理ノードに提供する機能を含めることができる。一実施形態では、インスタンスコーディネーター１１２は、サービスインスタンス１１６に対するロックに関する情報を要求元システムに提供することができる。

【００５１】

[0058]サービスインスタンス１１６を取得することの一部として、サーバーノード１１０は、サービスインスタンス１１６に関連付けられているよう望む１つまたは複数の追加の特性を供給することができる。これらの追加の特性は最初のルックアップの一部として使用されることはないが、関連付けを実施した後は、これらの追加の特性のうちの１つに対する将来のルックアップは、サービスインスタンス１１６と合致することになる。サーバーノード１１０が追加の特性を含む場合、これらの関連付けは、原子ｃｒｅａｔｅまたはｌｏａｄプロセスの一部として実施され、いくつかの実施形態では、ｃｒｅａｔｅまたはｌｏａｄプロセスがサービスインスタンス１１６の取得に成功した場合にのみ実施される。特に、インスタンスコーディネーター１１２から適切なサービスインスタンス１１６

を取得する試みは、１つまたは複数の追加の特性を送ることを含むことができる。これらの追加の特性は、サービスインスタンス１１６を取得するこの試みに対しては考慮されないが、適切なサービスインスタンス１１６を取得するこの試みに処理ノードがもし成功した場合には、サービスインスタンス１１６を記述する複数の特性にこれらの追加の特性が加えられる。例として、サーバーノード１１０は、購入注文識別子から導出された特性を使用してサービスインスタンス１１６をルックアップし、荷主追跡番号から導出された追加の特性を供給することができる。将来は、たとえ購入注文識別子がわからなかった場合でも、荷主追跡番号を使用してサービスインスタンス１１６を見つけることができる。このように、いくつかの実施形態では、インスタンスコーディネーターは、複数の特性をサービスインスタンスに関連付ける。インスタンスコーディネーターから適切なサービスインスタンスを取得する試みは、サービスインスタンスに関連する全ての特性を送らずに、いずれか１つまたは複数の特性をインスタンスコーディネーターに送ることを含むことができる。説明した例では、購入注文識別子と荷主追跡番号のいずれか一方を提供すればよく、他方の識別子または番号を提供する必要はない。

【００５２】

[0059]本発明のいくつかの実施形態では、インスタンスコーディネーター１１２は、ルックアップのための複数の特性を受諾し、特性のいずれか１つに合致するサービスインスタンス１１６を返す。特性がサービスインスタンス１１６を一意に定義しないときは、結果として、特性仕様のプロパティに基づいて特定のサービスインスタンス１１６を支持するか、サービスインスタンス１１６のプロパティに基づいて特定のサービスインスタンス１１６を支持するか、または、曖昧に指定されているとしてルックアップ動作を拒否する場合がある。

【００５３】

[0060]いくつかの実施形態では、サーバーノード１１０は、現在有するサービスインスタンス１１６のローカルカタログを維持する。このローカルカタログにより、サーバーノードは、実施すべき追加の関連付けがないときはインスタンスコーディネーター１１２と協議しないようにすることができる。

【００５４】

[0061]いくつかの実施形態では、サーバーノード１１０は、サービスインスタンス１１６の取得に成功した場合は、受信したメッセージ１１４をアプリケーションの残りに送り出す。そうでない場合は、サーバーノード１１０は、インスタンスコーディネーターから返されたアドレスを含むリダイレクトメッセージを構築し、この新しいアドレスにメッセージを再送するようメッセージの送信元に指示する。例えば、特定のサービスインスタンス１１６－３がサーバーノード１０４に位置することを、インスタンスコーディネーター１１２が示すと仮定する。サーバーノード１１０は、メッセージ１１４をサービスノード１０４に再送すべきであることを示すリダイレクトメッセージ１２０を構築することができ、リダイレクトメッセージ１２０は、ルーターノード１０６（メッセージ１１４をサーバーノード１１０に送信したノード）に送信される。

【００５５】

[0062]いくつかの実施形態では、新しいアドレスは、物理マシンアドレスではない。例えば、新しいアドレスは、メッセージの送信元にわかっているディレクトリ中でルックアップされることになる論理アドレスまたはサービス名である場合がある。別の例として、新しいアドレスは、ディレクトリサービスではなく動的発見プロトコルを使用して解決される場合がある。

【００５６】

[0063]いくつかの実施形態では、インスタンスコーディネーター１１２は、メッセージをルーティングするために厳密に必要とされる以外の、サービスインスタンス１１６に関するルーティング情報を提供して、全体的なルーティング品質を改善する。

【００５７】

[0064]サーバーノード１１０または他のルーターノードからのリダイレクトメッセージ

は、特性プロパティと処理ノードとの間の１つまたは複数の追加の関連付けを含むことができる。追加の特性プロパティは、現在処理されているメッセージに必ずしも関係しない。１つまたは複数の追加の関連付けに基づいて、信頼性のないローカルキャッシュを更新することができる。

【００５８】

[0065]以下の記述は、ルーターノード（ルーターノード１０６および１１０など）一般の機能に関するさらに多くの詳細を含む。加えて、リダイレクトメッセージ１２０の処理に関する詳細を含む場合もある。

【００５９】

[0066]図４に、方法４００を示す。方法４００は、メッセージを受信すること（動作４０２）を含む。メッセージは、コンピューター可読通信媒体から受信することができる。方法４００はさらに、メッセージの特性プロパティを計算して、メッセージの処理のためのサービスにおけるサービスインスタンス１１６に対する状態要件を決定すること（動作４０４）を含む。方法４００は、ルーティング情報の信頼性のないローカルキャッシュを使用して、処理ノード間の連携なしに、メッセージに対する可能性の高いルートを決定すること（動作４０６）を含む。メッセージは、メッセージに対する可能性の高いルートに従って、サーバーノードまたは別のルーターノードに向けて送られる。

【００６０】

[0067]方法４００は、サービスインスタンス１１６（図１参照）から独立した処理を行うように実施することができる。サービスインスタンス１１６から独立した処理の例は、静的に構成されたプロトコルおよびメッセージ変換を実施することである。

【００６１】

[0068]ノード１１０などのルーター－サーバーノードの場合、サーバーノードは、同じ処理ノードのサーバーコンポーネントであることがある。

[0069]ルーターノードの動作は、いくつかの点でサーバーノードの動作に類似する。しかし、インスタンスコーディネーター１１２など、サーバーインスタンス１１６に対する権威的なソースと協議する代わりに、ルーターノード１０６などのルーターノードは、潜在的に信頼性のないローカルルーティングデータと協議する。例えば、ルーターノード１０６は、潜在的に信頼性のないローカルルーティングデータを含むルーティングテーブル１１８－２を備えることができる。ルーターノード１０６は、時おり記憶消失に見舞われて、それによりルーティングデータのいくらかまたは全てを忘れてしまうことがあり、あるいは、正しくない推量のせいで誤ったルーティングデータが組み込まれることがある。処理ノードがネットワーク１００に追加されたかネットワーク１００から除去されたこと、処理ノードがそのタイプもしくは他の処理ノードとの接続性を変更したこと、またはサービスインスタンス１１６がある場所から別の場所に移動したことを、ルーターノード１０６が適時に通知される、それどころかいつか通知されるという予想はない。ルーターノード１０６は、ルーティングテーブル１１８－２中のルーティングデータを使用して、正しいルートの最良推量決定を行う。最良推量は、とりわけ、前にルーティングされたメッセージ１１４に、および、前にリダイレクトメッセージ１２０中で受け取ったインスタンスコーディネーター１１２からのルーティングデータに基づくことができる。

【００６２】

[0070]いくつかの実施形態では、最良推量は、ラウンドロビン負荷平衡などの負荷平衡機構を組み込むか、または、利用中として認識された処理ノードに向けてメッセージを送る。

【００６３】

[0071]ルーターノード１０６が、前にルーティングしたメッセージ１１４に対するリダイレクトメッセージ１２０を受信したとき、ルーターノード１０６は、どのようにこのルーティング失敗を扱うかを決定する。ルーターノード１０６は、送達機構、特性、およびルーティングデータに応じて、メッセージ１１４自体を再送することができる。他の場合では、ルーターノード１０６は、リダイレクトメッセージ１２０を元の送信元に返すこと

ができる。例えば、図１には示されていないが、追加のルーターノードがメッセージ１１４をルーターノード１０６に送信した場合がある。ルーターノード１０６は、テーブル１１８－２中の信頼性のないローカルルーティングデータを使用して、メッセージがルーターノード１１０に送信されるべきであると判定することができる。ルーターノード１１０は、メッセージ１１４が誤ってルーターノード１１０に送信されたと判定することができる。ルーターノード１１０は、この判定に応答して、リダイレクトメッセージ１２０をルーターノード１０６に送信する。いくつかの実施形態では、ルーターノード１１０はまた、メッセージ１１４に対する正しい処理ノードと思われるものを示す情報を送信することもできる。メッセージ１１４に対する正しい処理ノードと思われるものを示すこの情報は、ローカルルーティングテーブル１１８－１中の信頼性のない情報から導出することができ、あるいはインスタンスコーディネーター１１２からの信頼性のある連携情報から導出することができる。リダイレクトメッセージに応答して、ルーターノード１０６は、リダイレクトメッセージ１２０中で示される処理ノードにメッセージ１１４を再送することができる。

【００６４】

[0072]いくつかの実施形態では、ルーターノードは、可能なら常にリダイレクトメッセージを返す。単信メッセージソースから受信したメッセージは再送され、複信メッセージソースから受信したメッセージはリダイレクトメッセージを返す。

【００６５】

[0073]本発明の実施形態は、後でより詳細に論じるようにコンピューターハードウェアを備える専用または汎用コンピューターを含むかまたは利用することができる。本発明の範囲内の実施形態はまた、コンピューター実行可能命令および／またはデータ構造を搬送または記憶するための物理的および他のコンピューター可読媒体も含む。このようなコンピューター可読媒体は、汎用または専用コンピューターシステムによってアクセスできる任意の利用可能な媒体とすることができる。コンピューター実行可能命令を記憶するコンピューター可読媒体は、物理記憶媒体である。コンピューター実行可能命令を搬送するコンピューター可読媒体は、伝送媒体である。したがって、限定ではなく例として本発明の実施形態は、少なくとも２つの明確に異なる種類のコンピューター可読媒体、すなわち物理記憶媒体および伝送媒体を含むことができる。

【００６６】

[0074]物理記憶媒体は、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、ＣＤ－ＲＯＭ、もしくは他の光ディスク記憶装置、磁気ディスク記憶装置もしくは他の磁気記憶デバイス、または、所望のプログラムコード手段をコンピューター実行可能命令またはデータ構造の形で記憶するのに使用でき、汎用または専用コンピューターによってアクセスできる、任意の他の媒体を含む。

【００６７】

[0075]「ネットワーク」は、コンピューターシステムおよび／もしくはモジュール、ならびに／または他の電子デバイスの間で電子データの移送を可能にする１つまたは複数のデータリンクとして定義される。情報がネットワークまたは別の通信接続（ハードワイヤード、ワイヤレス、またはハードワイヤードとワイヤレスの組合せ）を介してコンピューターに転送または提供されるとき、コンピューターはこの接続を伝送媒体として正しく見なす。伝送媒体は、所望のプログラムコード手段をコンピューター実行可能命令またはデータ構造の形で搬送するのに使用でき、汎用または専用コンピューターによってアクセスできる、ネットワークおよび／またはデータリンクを含むことができる。以上の組合せもまた、コンピューター可読媒体の範囲に含まれるべきである。

【００６８】

[0076]さらに、コンピューター実行可能命令またはデータ構造の形のプログラムコード手段は、様々なコンピューターシステムコンポーネントに到達すると、伝送媒体から物理記憶媒体に（またはその逆に）自動的に転送されてよい。例えば、ネットワークまたはデータリンクを介して受け取ったコンピューター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イスモジュール（例えば「ＮＩＣ」）内のＲＡＭにバッファリングしてから、最終的にコンピューターシステムＲＡＭに、および／またはコンピューターシステムにおけるより揮発性の低い物理記憶媒体に、転送することができる。したがって、物理記憶媒体は、伝送媒体をも（さらには伝送媒体を主に）利用するコンピューターシステムコンポーネントに含めることができることを理解されたい。

【００６９】

[0077]コンピューター実行可能命令は、例えば、ある機能または機能グループを汎用コンピューター、専用コンピューター、または専用処理デバイスに実施させる命令およびデータを含む。コンピューター実行可能命令は、例えば、バイナリ、アセンブリ言語などの中間フォーマット命令、さらにはソースコードとすることができる。構造上の特徴および／または方法上の動作に特有の言語で本主題を述べたが、添付の特許請求の範囲に定義する本主題は、前述の特徴または動作に必ずしも限定されないことを理解されたい。そうではなく、前述の特徴および動作は、特許請求の範囲を実施する例示的な形として開示する。

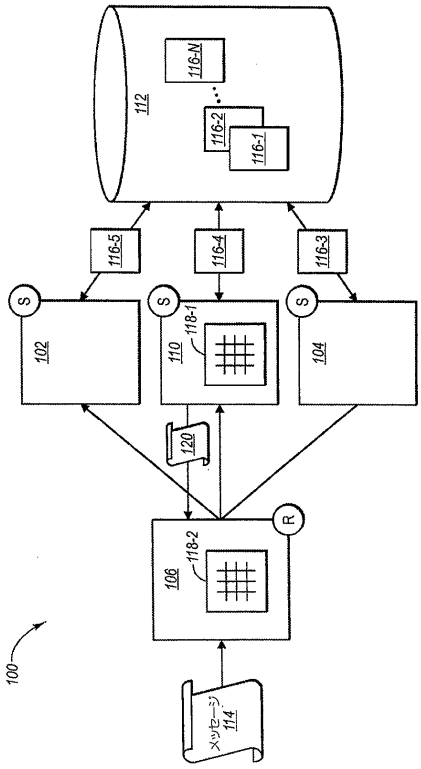
【００７０】

[0078]本発明は、パーソナルコンピューター、デスクトップコンピューター、ラップトップコンピューター、メッセージプロセッサー、ハンドヘルドデバイス、マルチプロセッサーシステム、マイクロプロセッサーベースのまたはプログラム可能な消費者電子機器、ネットワークＰＣ、ミニコンピューター、メインフレームコンピューター、携帯電話機、ＰＤＡ、ページャー、ルーター、スイッチなどを含めた、多くのタイプのコンピューターシステム構成を伴うネットワークコンピューティング環境で実践できることは、当業者なら理解するであろう。本発明はまた、ネットワークを介して（ハードワイヤードデータリンク、ワイヤレスデータリンク、またはハードワイヤードとワイヤレスのデータリンクの組合せによって）リンクされたローカルとリモートのコンピューターシステムが両方ともタスクを実施する、分散システム環境で実践することもできる。分散システム環境では、プログラムモジュールは、ローカルとリモートの両方のメモリー記憶デバイスに位置することができる。

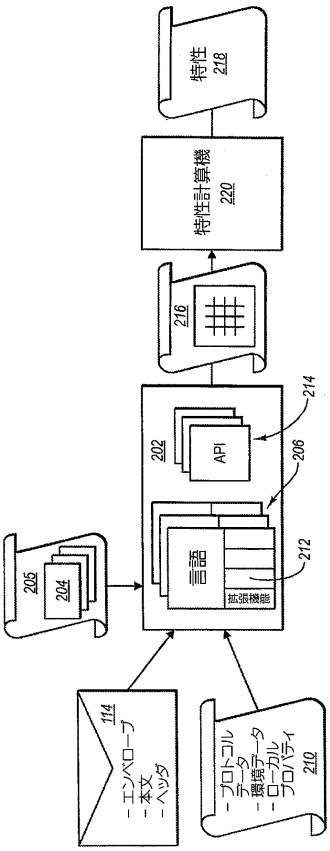
【００７１】

[0079]本発明は、本発明の趣旨または本質的な特性を逸脱することなく、他の特定の形で具体化することもできる。述べた実施形態は、あらゆる点で、例示的としてのみ考えるべきであり限定的として考えるべきではない。したがって、本発明の範囲は、以上の記述によってではなく添付の特許請求の範囲によって示す。特許請求の範囲の均等の意味および範囲の内に入るあらゆる変更は、その範囲内に包含されるべきである。

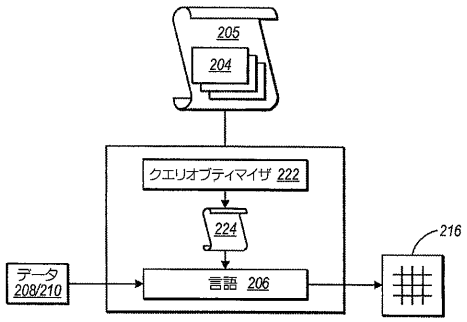
【図１】



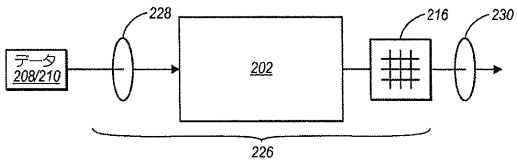
【図２Ａ】



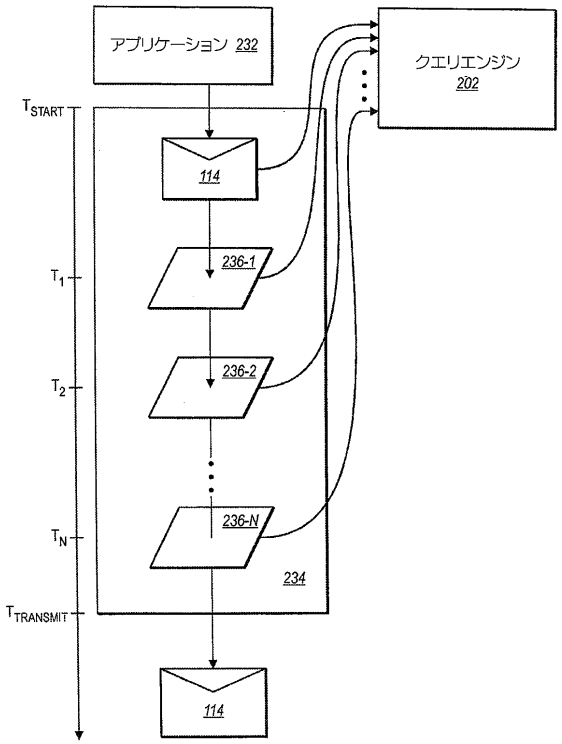
【図２Ｂ】



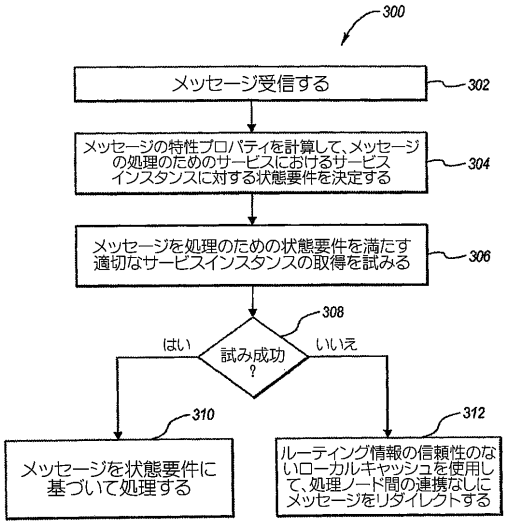
【図２Ｃ】



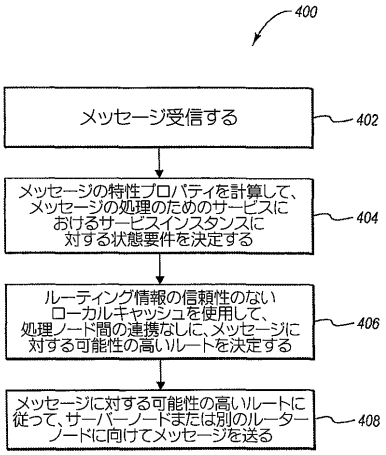
【図２Ｄ】



【図３】


300
メッセージ受信する
302
メッセージの特性プロパティを計算して、メッセージの処理のためのサービスにおけるサービスインスタンスに対する状態要件を決定する
304
メッセージを処理のための状態要件を満たす適切なサービスインスタンスの取得を試みる
306
308
試み成功？
はい
いいえ
310
メッセージを状態要件に基づいて処理する
312
ルーティング情報の信頼性のないローカルキャッシュを使用して、処理ノード間の連携なしにメッセージをリダイレクトする

【図４】


400
メッセージ受信する
402
メッセージの特性プロパティを計算して、メッセージの処理のためのサービスにおけるサービスインスタンスに対する状態要件を決定する
404
ルーティング情報の信頼性のないローカルキャッシュを使用して、処理ノード間の連携なしに、メッセージに対する可能性の高いルートを決定する
406
メッセージに対する可能性の高いルートに従って、サーバーノードまたは別のルーターノードに向けてメッセージを送る
408

フロントページの続き

(72)発明者　アレン，ニコラス・エイ
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，ワン・マイクロソフト・ウェイ，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
(72)発明者　バトレス，ステフェン・アール
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，ワン・マイクロソフト・ウェイ，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
(72)発明者　ブラウン，ジャスティン・ディー
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，ワン・マイクロソフト・ウェイ，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
(72)発明者　ピント，エドムンド・エスヴイ
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，ワン・マイクロソフト・ウェイ，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
(72)発明者　ラマン，カーシク
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，ワン・マイクロソフト・ウェイ，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
(72)発明者　テイラー，ジョン・エイ
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，ワン・マイクロソフト・ウェイ，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

審査官　木村　雅也

(56)参考文献　米国特許出願公開第２００６／０１２９６５０（ＵＳ，Ａ１）
米国特許出願公開第２００７／０１６８５４６（ＵＳ，Ａ１）
米国特許出願公開第２００２／０１８４３４４（ＵＳ，Ａ１）
米国特許出願公開第２００４／０１４８３３４（ＵＳ，Ａ１）
米国特許出願公開第２００６／０１２３４６７（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１３／００
Ｈ０４Ｌ　１２／５４