(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5269081号
(P5269081)

(45)発行日　平成25年8月21日(2013.8.21)　(24)登録日　平成25年5月17日(2013.5.17)

(51)Int.Cl.　F I
G06F 9/54 (2006.01)　G06F 9/06 640E
G06F 9/445 (2006.01)　G06F 9/06 610Q
G06F 12/00 (2006.01)　G06F 12/00 537H
G06F 12/00 545M

請求項の数 8　(全 15 頁)

(21)出願番号　特願2010-527015 (P2010-527015)
(86)(22)出願日　平成20年9月12日 (2008.9.12)
(65)公表番号　特表2010-541070 (P2010-541070A)
(43)公表日　平成22年12月24日 (2010.12.24)
(86)国際出願番号　PCT/US2008/076136
(87)国際公開番号　WO2009/042421
(87)国際公開日　平成21年4月2日 (2009.4.2)
審査請求日　平成23年9月5日 (2011.9.5)
(31)優先権主張番号　11/861,877
(32)優先日　平成19年9月26日 (2007.9.26)
(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　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　アルバート　シー．エス．シェン
アメリカ合衆国　９８０５２　ワシントン州　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　インターナショナル　パテンツ内

最終頁に続く

(54)【発明の名称】拡張可能な分散アプリケーションの作成とデプロイメント

(57)【特許請求の範囲】
【請求項１】
リモートサービスクラスタで利用可能なサービスのリストを、コンピュータにより、前記リモートサービスクラスタから受信するステップと、
前記利用可能なサービスのリストからユーザが選択したサービスを示す情報を、前記コンピュータにより、作成インタフェースから取得するステップと、
前記ユーザが選択したサービスで利用可能なコンポーネントのリストを、前記コンピュータにより、前記ユーザが選択したサービスから受信するステップと、
ユーザの操作入力に従って、前記リモートサービスクラスタの前記ユーザによって選択されたサービスで利用可能なコンポーネントのリストから、一群のコンポーネントを前記作成インタフェースにより選択するステップと、
前記ユーザによって選択された一群のコンポーネントを示す情報を、前記コンピュータにより、前記作成インタフェースから取得するステップと、
前記選択された一群のコンポーネントを示す情報を、前記コンピュータにより、前記選択されたサービスに送信するステップと、
前記選択されたコンポーネントを記憶デバイスにインストールするのに必要なデータを前記リモートサービスクラスタの前記ユーザによって選択されたサービスから、前記コンピュータにより受信するステップと、
受信したデータに応じて、命令のリストを前記コンピュータにより作成するステップと、

前記作成した命令のリストを前記記憶デバイスに前記コンピュータにより保存するステップと、
を備え、
前記命令のリストを保存するステップは、
前記受信したデータを前記命令のリストに前記コンピュータによりコード化するステップと、
前記命令のリストにウォーターマークを前記コンピュータによりコード化するステップと、
を備えることを特徴とする分散アプリケーションを作成する方法。
【請求項２】
前記一群のコンポーネントを選択するステップは、前記選択されたサービスで利用可能なコンポーネントの種類に特定の選択インタフェースデータを前記コンピュータにより受信するステップをさらに含むことを特徴とする請求項１に記載の方法。
【請求項３】
前記選択されたコンポーネントのインストールに必要なデータを前記コンピュータにより抽出するために、前記命令のリストを処理するステップと、
前記コンピュータにより、前記選択されたコンポーネントのインストールに必要なデータを前記リモートサービスクラスタに送信して、前記リモートサービスクラスタへの前記選択されたコンポーネントのインストールを可能にするステップと、
をさらに備えることを特徴とする請求項１又は２に記載の方法。
【請求項４】
前記命令のリストを処理するステップは、
前記選択されたコンポーネントの記述を前記選択されたコンポーネントに関連するサービスに前記コンピュータにより送信するステップと、
前記サービスからコンフリクト情報を前記コンピュータにより受信するステップと、
前記コンフリクト情報を集中処理して、前記選択されたコンポーネントにコンフリクトが存在するか否かを前記コンピュータにより決定するステップと、
コンフリクトが存在しないときに、前記選択されたコンポーネントのインストールに必要なデータを前記リモートサービスクラスタに前記コンピュータにより送信して、前記コンフリクト情報を集中処理するステップに応じて、前記リモートサービスクラスタへの前記選択されたコンポーネントのインストールを可能にするステップと、
をさらに含むことを特徴とする請求項３に記載の方法。
【請求項５】
前記命令のリストを処理するステップは、
コンフリクトが存在するとき、前記コンフリクトが自動で処理可能か否かを前記コンピュータにより決定するステップと、
コンフリクトが自動で処理可能か否かの前記決定に応じて、前記コンピュータによりコンフリクトを自動で処理するステップと、
をさらに含むことを特徴とする請求項４に記載の方法。
【請求項６】
請求項１から５のいずれか一項に記載の方法を前記コンピュータおよび前記リモートサービスクラスタに実行させるためのプログラム。
【請求項７】
請求項６に記載のプログラムを記録したコンピュータ可読記録媒体。
【請求項８】
コンピュータとリモートサービスクラスタとを含んで構成されるシステムであって、前記コンピュータと前記リモートサービスクラスタとが請求項１から５のいずれか一項に記載の方法を実行することを特徴とするシステム。
【発明の詳細な説明】
【技術分野】

【０００１】
本発明は、拡張可能な分散アプリケーションの作成とデプロイメントに関する。
【背景技術】
【０００２】
分散コンピューティングネットワークにインストールされたソフトウェアによって、ユーザは、別々のサーバクラスタ上にホストされる多数のサービスを活用したエンドツーエンドアプリケーションやエンドツーエンドソリューションを作成することができる。このエンドツーエンドアプリケーションは、ビジネスデータ、ページデザイン、ページレイアウト、ビジネスロジックなどのコンポーネントを含むことができ、それらは、各々、異なるサーバクラスタに分散されていてもよい。
【発明の概要】
【発明が解決しようとする課題】
【０００３】
エンドツーエンドアプリケーションは分散性なので、調整、認証、コンテンツの正確性、コンフリクト管理、スケーラビリティなどの点で、そのデプロイメント（注：ネットワークアプリケーションやＷｅｂサービスなどを、利用可能なように準備すること）には様々な困難が伴う。例えば、利用できるサービスの数が急速に増加すると、エンドツーエンドアプリケーションは、それまでに作成されたアプリケーションとの完全な後方互換性を維持しながら、その一方で、今後のサービスを容易にかつ柔軟に組み込めるように拡張可能でなければならない。
【課題を解決するための手段】
【０００４】
この概要は、発明の詳細な説明でより詳しく記載するコンセプトの一部を簡単に紹介するものである。この概要は、発明の主題の主要な特徴や不可欠な特徴を特定することを意図するものではないし、発明の主題の範囲を決定するための補助として用いられることを意図するものでもない。
【０００５】
分散アプリケーションを作成する方法には、リモートサーバクラスタで利用可能なコンポーネントのリストから一群のコンポーネントを選択することが含まれる。選択したコンポーネントのインストールに必要なデータは、リモートサーバクラスタから受信する。受信したデータに応じて、命令のリストを作成し、作成した命令のリストを保存する。
【０００６】
有形のコンピュータ可読媒体は、分散アプリケーションを作成するコンピュータ実行可能命令を有する。コンピュータ実行可能命令には、分散コンピュータシステムで利用可能なサービスのリストからサービスを選択することが含まれる。選択したサービスで利用可能なコンポーネントを選択する。選択したコンポーネントに関するインストールデータを、分散コンピュータシステムから受信する。受信したデータに応じてパッケージファイルを作成する。作成したパッケージファイルを保存する。
【０００７】
分散コンピューティングアプリケーションを集中制御するシステムは、プロセッサとコンピュータ可読媒体とを備える。そのシステムは、オペレーティング環境も備え、そのオペレーティング環境は、コンピュータ可読媒体に保存され、プロセッサで実行される。さらに、コンピュータ可読媒体に保存され、かつプロセッサで実行されるソリューションフレームワークも備える。ソリューションフレームワークは、サービスクラスタで利用可能なサービスのリストからサービスを選択するように構成される。選択されたサービスで利用可能なコンポーネントを選択する。選択したコンポーネント（複数可）に関するインストールデータを、選択したサービスから受信する。受信したデータに応じてパッケージファイルを作成する。次に、作成したパッケージファイルをコンピュータ可読媒体に保存する。
【０００８】

これらの特徴や利点及び他の特徴や利点は、下記の詳細な記載を読み、添付図面を参照することで明らかとなる。前述の一般的記載および下記の詳細な記載は、説明のためのものであり、範囲を制限する目的ではないことを理解されたい。とりわけ、本明細書に記載の様々な実施形態は、方法、装置、またはその両方の組合せとして、実現してもよい。同様に、様々な実施形態は、完全にハードウェアの実施形態、完全にソフトウェアの実施形態、もしくはソフトウェアとハードウェアの態様を組み合わせた実施形態、という形を取ってもよい。したがって、本明細書に開示する内容を限定的な意味で捉えるべきではない。

【図面の簡単な説明】

【０００９】

図面で、同じ番号は同じ要素を表す。

【図１】本明細書に記載のコンピュータ実装方法を実装するオペレーティング環境のブロック図である。

【図２】ソリューションフレームワークを実装するオペレーティング環境を示すブロック図である。

【図３】パッケージファイルの実装例を示す図である。

【図４】一群のコンポーネントを選択する操作を示すフロー図である。

【図５】パッケージファイルを作成する操作を示すフロー図である。

【図６】分散コンピューティング環境にパッケージファイルをデプロイする操作を示すフロー図である。

【発明を実施するための形態】

【００１０】

図面を参照しながら、様々な実施形態を記載する。図面では、同じ番号は同じ要素を表す。詳細には、図１および図１に対応する記載は、実施形態を実装し得る適切なコンピューティング環境を簡単かつ一般的に記載したものである。

【００１１】

一般に、エンドユーザとサービスクラスタ間のインタフェース接続に関与するソリューションフレームワークが設けられる。ソリューションフレームワークによって、分散アプリケーションの作成およびデプロイメントを集中化することができる。分散アプリケーションの作成にもデプロイメントにも多くの利点がある。例えば、分散アプリケーションは、多くの別々のサービスの機能を組み合わせて１つのアプリケーションにするために、多くの異なるリモートサーバに配置された多くのサービスのコンポーネントを利用することができる。

【００１２】

ソリューションフレームワークは、作成ユーザインタフェースとサービスクラスタ間の媒介として機能することにより、分散アプリケーションの作成を集中化する。しかしながら、ユーザが、特定のサービスを組み込んだパッケージファイルを作成するには、その前に、そのサービスの認証を受ける必要があるかもしれない。ソリューションフレームワークは、サーバクラスタの各々にユーザの信用情報を送って、この認証プロセスを集中化することもできる。そうすると、ソリューションフレームワークは、認証されたユーザが、サービスクラスタ内の利用できる様々なコンポーネントを選択して、分散アプリケーションを定義できるようにする。アプリケーションの定義に応じて、パッケージファイルが作成される。パッケージファイルのコンテンツの正確性は、ウォーターマークメタデータを使用して維持される。ウォーターマークメタデータは、パッケージファイル内に含まれており、パッケージファイルのコンテンツに基づいている。

【００１３】

分散アプリケーションをインストールする前に、パッケージファイルは、アプリケーションを定義するユーザから別のユーザに転送されてもよく、アプリケーションを定義するユーザが単にインストールしてもよい。インストールの間、ソリューションフレームワークがコンフリクト解消を集中的に制御してもよい。コンフリクトが解消されると、ソリュ

ーションフレームワークは、関連する全てのサービスクラスタに通信し、定義されたパッケージファイルに基づいて、正しいロケーションにコンポーネントをインストールすることができる。

【００１４】

図１を参照しながら、様々な実施形態で用いられるコンピュータ１００の例示のコンピュータアーキテクチャを記載する。図１のコンピュータアーキテクチャは、デスクトップコンピュータまたはモバイルコンピュータとして構成されてもよく、中央演算処理装置５（「ＣＰＵ」）と、ランダムアクセスメモリ９（「ＲＡＭ」）や読取専用メモリ１０（「ＲＯＭ」）などのシステムメモリ７と、そのシステムメモリをＣＰＵ５に接続するシステムバス１２とを備える。

【００１５】

基本入出力システムは、スタートアップ時などにコンピュータ内の要素間での情報転送を支援する基本ルーチンを含み、ＲＯＭ１０に保存されている。コンピュータ１００は、さらに大容量記憶装置１４を備え、大容量記憶装置１４は、オペレーティングシステム１６と、アプリケーションプログラム２４と、ソリューションフレームワーク２６とを保存する。これについては、下記に詳述する。

【００１６】

大容量記憶装置１４は、バス１２に接続された大容量記憶コントローラ（図示せず）を介してＣＰＵ５に接続される。大容量記憶装置１４とそれに関連付けられたコンピュータ可読媒体は、コンピュータ１００の不揮発性記憶装置となる。本明細書のコンピュータ可読媒体の記載は、ハードディスクやＣＤ－ＲＯＭドライブなどの大容量記憶装置に関するものであるが、コンピュータ可読媒体は、コンピュータ１００がアクセス可能な任意の入手可能な媒体であってよい。

【００１７】

限定ではなく一例として、コンピュータ可読媒体は、コンピュータ記憶媒体や通信媒体を含んでもよい。コンピュータ記憶媒体には、コンピュータ可読命令、データ構造、プログラムモジュールまたは他のデータなどの情報を保存する任意の方法または技術で実装される、揮発性媒体、不揮発性媒体、取り外し可能媒体、および取り外し不能媒体が含まれる。コンピュータ記憶媒体には、ＲＡＭ、ＲＯＭ、ＥＰＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリ、もしくはその他の固体メモリ技術、ＣＤ－ＲＯＭ、デジタル多用途ディスク（「ＤＶＤ」）、もしくはその他の光学式記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置、もしくは他の磁気記憶装置、または、コンピュータ１００がアクセス可能な所望の情報の保存に使用できる他の任意の媒体が含まれるが、それらに限定されない。

【００１８】

様々な実施形態によると、コンピュータ１００は、インターネットなどのネットワーク１８を介してリモートコンピュータへの論理接続を用いて、ネットワーク環境で動作することができる。コンピュータ１００は、バス１２に接続されたネットワークインタフェースユニット２０を介してネットワーク１８に接続してもよい。ネットワーク接続は、無線および／または有線であってよい。ネットワークインタフェースユニット２０を、他の種類のネットワークやリモートコンピュータシステムへの接続に用いることもできる。コンピュータ１００は、入出力コントローラ２２を備えて、キーボード、マウス、または、電子スタイラス（図１には示されていない）などの他の多くの装置からの入力を受信し、処理することもできる。同様に、入出力コントローラ２２は、表示画面２８、プリンタ、または他のタイプの出力装置に出力することができる。

【００１９】

簡単に上述したように、多くのプログラムモジュールやデータファイルを、オペレーティングシステム１６を含めて、コンピュータ１００の大容量記憶装置１４やＲＡＭ９に保存することができる。オペレーティングシステム１６は、ネットワーク化されたパーソナルコンピュータの操作を制御するのに適しており、ワシントン州レドモンドのＭＩＣＲＯＳＯＦＴ社製のＷＩＮＤＯＷＳ　ＶＩＳＴＡオペレーティングシステムなどがある。大容

量記憶装置１４とＲＡＭ９は、１つまたは複数のプログラムモジュールを保存してもよい。具体的に言うと、大容量記憶装置１４とＲＡＭ９は、１つまたは複数のアプリケーションプログラム２４を保存することができる。例えば、大容量記憶装置１４は、ソリューションフレームワーク２６を保存することができる。ソリューションフレームワーク２６は、分散アプリケーションの開発とインストールを集中化する。

【００２０】

図２は、ソリューションフレームワーク２６が動作する環境の実装例を示す。ソリューションフレームワーク２６は、サービスクラスタ２３０に結合することができる。サービスクラスタ２３０は、インターネットやエクストラネットなどのネットワークを介してソリューションフレームワーク２６に結合することができる。サービスクラスタ２３０は、サービス２３２、サービス２３４、サービス２３６などのＮ個のサービスを含むことができる。個々のサービスは、別々のロケーションに保存することができる。例えば、サービス２３０は、第１のロケーションの第１のサーバに保存され、サービス２４０は、第２のロケーションの第２のサーバに保存することができる。他の実装例においては、多くのサービスを１つのロケーションに保存することができる。したがって、ソリューションフレームワークは、個々のサービスのロケーションとは独立して動作することができる。

【００２１】

各サービスは、様々な形態の機能を提供することができる。例えば、あるサービスは、ビジネスデータ、ページデザインまたはページレイアウトを含むことができる。他の例においては、サービスはビジネスロジックを含むことができ、さらに他の例においては、サービスは他の任意の形態の機能を含むことができる。

【００２２】

ソリューションフレームワーク２６は、作成インタフェース２１０とデプロイメントインタフェース２２０に接続することもできる。作成インタフェース２１０は、一群の選択したサービスを含むパッケージファイルをユーザが定義できるようにするインタフェースを提供することができる。デプロイメントインタフェース２２０は、ユーザがパッケージファイルをデプロイして一群の選択したサービスをインストールするのを可能にするインタフェースを提供することができる。ある実装例においては、これらのインタフェースは、ハイパーテキストマークアップ言語（ＨＴＭＬ）または拡張マークアップ言語（ＸＭＬ）などのマークアップ言語でコード化されたウェブインタフェースであってよい。他の実装例においては、これらのインタフェースは、Ｃ＃やＪａｖａ（登録商標）などの他の言語でコード化することができる。

【００２３】

作成インタフェース２１０は、第１のロケーションに配置された第１のコンピューティング環境に置いて、第１のユーザがパッケージファイルを作成できるようにしてもよく、デプロイメントインタフェース２２０は、第２のロケーションに配置された第２のコンピューティング環境に置いて、第２のユーザがパッケージファイルをデプロイできるようにしてもよい。このような実装例において、パッケージファイルは、第１のコンピューティング環境から第２のコンピューティング環境に転送することができる。この転送は、任意のファイル転送手段を用いて達成することができる。例えば、パッケージファイルは、コンピュータ可読媒体、例えば、第２のコンピューティング環境に物理的に転送するディスクなどにコード化することができる。他の例では、パッケージファイルは、２つのコンピュータ要素間のネットワーク接続を介して、例えばインターネット送信する電子メールの添付ファイルなどで、電子的に送信することができる。

【００２４】

他の実装例においては、作成インタフェース２１０およびデプロイメントインタフェース２２０を、同じコンピューティング環境に置いてもよい。このような実装例においては、同じユーザが、同じコンピューティング環境からパッケージファイルを作成することもデプロイすることもできる。さらに、このような実装例においては、上述の実装例では必要とされたパッケージファイルの転送を避けることができる。

【００２５】
したがって、ソリューションフレームワーク２６は、パッケージファイルの作成とパッケージファイルのデプロイメントの両方を管理し、集中化する。
【００２６】
図３は、パッケージファイル３００の実装例を示す。パッケージファイル３００は、ビット０からビット７６７までの７６８ビットを含むことができる。パッケージファイル３００は、ビット０からビット３１までのヘッダー部分を含むことができる。パッケージファイル３００は、選択したコンポーネントを記述するビット３２からビット６３までのマニフェストも含むことができる。マニフェストは、サーバクラスタ２３０から受信した情報を含むことができる。その情報はコンポーネントに関する情報を記述するもので、サービスは、その情報を用いて起こり得るコンフリクトやコンポーネントの利用可能性を検出することができる。例えば、マニフェストは、パックレット（小さいパケットデータ　ｐａｃｋｌｅｔ）作成時に各サービスクラスタが返信する情報の一部を含むことができる。その情報は、そのパックレット内にあるコンポーネントを記述するものである。マニフェストは、物理的なパックレットがなくても、マニフェスト内のメタ情報を用いてコンフリクトを検出できるように設計される。ヘッダーとマニフェストの次に、パッケージファイル３００は、ビット６４からビット１２７までのペイロードを含むことができる。図５を参照しながら下記に詳述するように、パッケージファイル３００のペイロード部は、選択したサービスから受信した情報を含む。
【００２７】
セキュリティのために、パッケージファイル３００は、ビット１２８からビット６３９までに、５１２ビット秘密鍵などの公開鍵を含むことができる。図６を参照しながら下記に詳述するように、さらなるセキュリティと、ファイルのインテグリティの検証を可能にするために、パッケージファイル３００は、ビット６４０からビット７６７に、暗号化された１２８ビットのセキュアハッシュアルゴリズム５（ＳＨＡ－Ｉ）のハッシュなどのウォーターマークを含むことができる。他の実施形態においては、ウォーターマークは、メッセージダイジェストアルゴリズム５（ＭＤ５）のハッシュを含むことができる。ハッシュは、パッケージファイル内のペイロードおよびマニフェストに任意のハッシュアルゴリズムを適用することによって、作成することができる。
【００２８】
ソリューションフレームワーク操作
図４を参照しながら、分散アプリケーションに含まれるようにコンポーネントのセットを定義する例示のプロセス４００を記載する。
【００２９】
本明細書に提示されるルーチンの記載を読むと、様々な実施形態の論理演算は、（１）コンピューティングシステム上で実行するコンピュータ実装された一連のアクトまたはプログラムモジュールとして、および／または（２）コンピューティングシステム内で互いに接続された機械論理回路または回路モジュールとして、実装されることを理解されるであろう。実装は、本発明を実装するコンピューティングシステムの性能要件に左右される選択事項である。したがって、本明細書に記載された実施形態を作成する例示の論理演算は、操作、構造装置、アクトまたはモジュールとして様々に参照される。これらの操作、構造装置、アクト、およびモジュールは、ソフトウェア、ファームウェア、専用デジタル論理、およびそれらの任意の組合せで実装することができる。
【００３０】
開始操作から、プロセスは操作４１０に進み、ソリューションフレームワーク２６でサービスリストを受信する。例えば、ユーザが作成インタフェース２１０のパッケージ作成ページを閲覧すると、このプロセスが開始されるようにしてもよい。次に、ソリューションフレームワークは、サービスクラスタ２３０に連絡して、利用可能なサービスのリストを要求することができる。あるサービスが利用可能か否かを決定するために、ソリューションフレームワーク２６は、作成インタフェース２１０と、サービスクラスタ２３０内の

各サービスの間の許可を集中的に仲介することができる。例えば、ソリューションフレームワーク２６は、サービスクラスタ２３０にユーザのプロファイルを送信し、返信に、そのユーザがアクセスしてよいサービスのセットを受信することができる。したがって、利用可能性は、ソリューションフレームワーク２６にネットワーク化されたサービスと、個々のユーザへの許可に左右され得る。

【００３１】

利用可能なサービスのリストを受信し、ソリューションフレームワーク２６で最終的に承認した後、プロセスは、操作４２０に進み、第１のサービスが選択される。

【００３２】

サービスは、作成インタフェース２１０から受信した、サービスを選択するようにというコマンドに応じて選択することができる。作成インタフェース２１０は、そのサービス内のコンポーネントを見るというユーザのリクエストを示している。サービスを選択すると、プロセスは操作４３０に進み、そこでコンポーネントの種類のリストが集められる。この操作の間、ソリューションフレームワーク２６は、選択したサービスに通信して、選択したサービスで利用可能な個々のコンポーネントの種類のリストを送信するようにリクエストする。種類情報には、利用可能なコンポーネントの種類を記述する情報や、個々のコンポーネントの選択に利用できるユーザインタフェースを記述する情報を含むことができる。ある実装例においては、この情報は、コンポーネントの種類や、そのサービスに特定のコンポーネントを選択するカスタムユーザインタフェースを含むことができる。

【００３３】

次にプロセスは操作４４０に進み、利用可能なコンポーネントのリストを集める。上述したサービスの利用可能性と同様に、コンポーネントの利用可能性は、コンポーネントがソリューションフレームワーク２６に接続されているか否かだけではなく、許可のチェックにも左右され得る。

【００３４】

次にプロセスは操作４５０に進み、選択したコンポーネントに関する受信された種類情報が処理され、作成インタフェース２１０に送信されて選択インタフェースをユーザに提示する。この選択インタフェースは、利用可能なコンポーネントの具体的な種類の選択を容易にするためにカスタム選択インタフェースが表示されるように、選択したサービスから受信した種類情報に応じて作成される。このカスタムインタフェースは、選択したコンポーネントに任意のパラメータをユーザが入力する機構を含むことができる。これらのパラメータは、ソリューションフレームワークやサービスクラスタに対して、選択したコンポーネントをパッケージする方法を記述する、追加のメタ情報として働くことができる。

【００３５】

次にプロセスは操作４６０に進む。操作４６０で、ユーザは、図２の作成インタフェース２１０に提示された選択インタフェースを用いて所望のコンポーネントを選択する。一部の実装例においては、この操作の間、選択したコンポーネントのリストがサービスクラスタ２３０に送信され、サービスクラスタ２３０でそのリストを受信する。他の実装例においては、操作４７０の後、選択した各サービスに関するプロセスが終了するまで、選択したコンポーネントのリストはサービスクラスタ２３０に送信されない。

【００３６】

操作４７０に進んで、サービスをさらに選択するか否かの決定が行われる。ユーザがさらにサービスを選択する場合、プロセスは操作４２０に戻り、次に選択されたサービスに関してインタフェース生成およびコンポーネント選択プロセスが繰り返される。ユーザがさらなるサービスの選択を行わない場合、プロセスは操作４８０に続く。

【００３７】

図５を参照しながら下記に詳述するように、操作４８０に進むと、複数のパックレットは全てまとめられて、単一のパッケージファイルにフォーマットされる。次にプロセスは終了操作に進み、他のアクションの処理に戻る。

【００３８】

ここで図５を参照しながら、操作４８０で作成されたパッケージファイルなどの、パッケージファイルを作成する例示のプロセス５００を記載する。

【００３９】

開始操作の後、プロセスは操作５１０に進み、選択したコンポーネントのリストは、ソリューションフレームワーク２６によって処理される。選択したコンポーネントのリストは、ソリューションフレームワーク２６で、作成インタフェース２１０から受信することができる。選択したコンポーネントのリストは、例えば、図４に示されたプロセス４００にしたがって作成されたものでもよい。

【００４０】

次にプロセスは操作５２０に進む。操作５２０で、選択したコンポーネントが配置されている第１のサービスを選択する。この選択は自動プロセスの一部であり、ソリューションフレームワーク２６は、選択したコンポーネントのリストにあるコンポーネントが含まれるサービスの全てを通してこの自動プロセスを反復する。このように、サービスは、作成インタフェース２１０を通してユーザの介入やユーザからの入力なしに、ソリューションフレームワーク２６によって直接選択されてもよい。ソリューションフレームワーク２６が、第１のサービスを選択すると、プロセスは操作５３０に進む。操作５３０では、選択したサービスに関連するコンポーネントの記述が、ソリューションフレームワークから第１の選択したサービスに送信される。すなわち、ソリューションフレームワーク２６は、ユーザが選択したサービス内にあるコンポーネントのリストをサービスに送信する。

【００４１】

サービスがコンポーネントのリストを受信すると、プロセスは操作５４０に進む。操作５４０で、サービスは、選択したコンポーネントに関する情報を含むパックレットを送信することによってリクエストに応じる。パックレットは、リクエストされたデータを含むバイナリデータストリームと、リクエストされたデータに関するメタ情報を記述するマニフェストを含むことができる。

【００４２】

操作５５０において、サービスをさらに選択するか否かの決定が行われる。ソリューションフレームワーク２６が、選択したコンポーネントの全てを処理するのに必要な全てのサービスを一巡して処理し終えると、選択する必要のあるサービスがなくなり、プロセスは操作５６０に続く。ソリューションフレームワーク２６が、選択したコンポーネントに関するサービスの全てを一巡してはいない場合、プロセスは操作５２０に戻り、次のサービスが選択される。他の実施形態においては、このプロセスは、非同期的に実行することができる。すなわち、マルチスレッド環境のことで、ソリューションフレームワーク２６は、全てのサービスにパックレットを受信するよう同時に連絡し、それらが戻るときに集合させる。このように、このプロセスは、非同期的に実行してもよく、線形に／順次に行ってもよい。

【００４３】

操作５６０において、受信したパックレットがパッケージファイルに追加される。例えば、複数のパックレットを統合して、パッケージファイル３００のペイロード部に追加することができる。他の実装例においては、パッケージファイル３００のペイロード内に含める前に、パックレットのさらなる処理を実行することができる。

【００４４】

操作５７０において、ウォーターマークがパッケージファイル３００に追加される。一部の実装例においては、ウォーターマークは、単に、パックレットから作成されたハッシュであってよい。他の実装例においては、ウォーターマークは、パッケージファイルに含まれるパックレットや他のデータから作成することができる。パッケージファイルに含まれる他のデータとは、ヘッダー部、マニフェスト部、秘密鍵などのセキュリティ部などである。ある例においては、ウォーターマークは、ＳＨＡ－１ハッシュアルゴリズムを用いて作成することができる。したがって、ハッシュは、後に、パッケージファイル３００が改ざんされているか否かの決定を含めて、データのインテグリティを検証するのに用いる

ことができる。次に、プロセスは終了操作に進み、他のアクションの処理に戻る。

【００４５】

図６を参照しながら、パッケージファイル３００をデプロイする例示のプロセス６００を記載する。

【００４６】

開始操作後、プロセスは操作６１０に進む。操作６１０において、ソリューションフレームワーク２６で、パッケージファイル３００が、デプロイメントインタフェース２２０から受信される。ある例においては、パッケージファイル３００の作成中に、システムが受信したパッケージファイル３００を、システムからソリューションフレームワーク２６に直接送信することができる。これは、パッケージファイル３００を作成したユーザが、そのファイルのデプロイメントも行っているという状況で、起こりうる。他の例においては、パッケージファイル３００は、別のユーザからソリューションフレームワーク２６に送信することができる。例えば、パッケージファイル３００は、作成インタフェース２１０で、第１のユーザが受信してもよく、次に、デプロイメントインタフェース２２０で、コンパクトディスク（ＣＤ）または電子メールの形で第２のユーザに送ってもよい。そこから、ソリューションフレームワーク２６に送信する。

【００４７】

ソリューションフレームワーク２６でパッケージファイル３００を受信すると、プロセスは、操作６２０に進む。操作６２０において、パッケージファイル３００が有効か否かを決定する。この決定は、ウォーターマークを参照して行うことができる。例えば、パッケージファイル３００が様々に送信される間に破損した場合、または、パッケージファイル３００が故意に改ざんされた場合、ウォーターマークがパッケージファイル３００内に含まれるデータに適切に合致しなくなる。他の例では、パッケージファイルが有効か否かの決定に他の基準を用いることができる。例えば、ファイルの拡張子が変更されているか否かを参照してもよい。他の例では、ファイルサイズを参照してもよい。すなわち、デフォルトの最大ファイルサイズを割り当てることができ、そのデフォルトのファイルサイズより大きいサイズは、無効として警告することができる。このように、パッケージファイル３００が有効か否かの決定は、一部はウォーターマークによってもよく、一部は、パッケージファイル３００の他のプロパティによってもよい。その結果、パッケージファイル３００が有効な場合、プロセスは操作６３０に進み、操作６３０で、処理が継続する。

【００４８】

パッケージファイル３００がもはや有効でない場合、プロセスは操作６８０に進み、操作６８０でデプロイメントが停止され、プロセスの流れは、終了操作に進む。ある例においては、停止操作６８０は、デプロイメントインタフェースで、ユーザへのエラーメッセージの提示を含むことができる。他の例においては、停止操作６８０は自動エラー修正を含むことができ、修正可能なエラーはソリューションフレームワーク２６により自動的に修正され、プロセスを継続することができる。

【００４９】

操作６３０において、パッケージファイル３００は処理され、ペイロードが抽出される。ペイロードは処理されて、どのサービスが要求されているか決定する。次に、ソリューションフレームワーク２６は、パッケージファイルが要求するサービス全てが利用可能か否かを決定する。パッケージファイル３００内に含まれるコンポーネントが特定のサービスに存在する場合は、そのサービスが要求される。例えば、パッケージファイル３００の作成とパッケージファイル３００のデプロイメントの間に時間が経過した場合、サービスクラスタ２３０で利用可能な１つまたは複数のサービスが、もはや利用不可能になっている場合がある。

【００５０】

上述のように、例えば、パッケージファイル３００をデプロイするユーザが、特定のサービスにアクセスする許可を持っていない場合、または、サービスが単にネットワークから切断されてしまった場合、そのサービスはもはや利用できない。要求されたサービスが

利用できない場合、プロセスはオペレーション６８０に進み、デプロイメントは停止される。要求されたサービスが全て利用可能な場合、プロセスは操作６４０に進む。

【００５１】

操作６４０において、第１のサービスが選択される。この選択は、自動プロセスの一部であり、ソリューションフレームワーク２６は、パッケージファイル３００のペイロードにあるコンポーネントが利用するサービスの全てを通して、この自動プロセスを反復する。このようにして、サービスは、デプロイメントインタフェース２２０を通してのユーザの介入や入力なしに、ソリューションフレームワーク２６によって直接選択することができる。ソリューションフレームワーク２６が第１のサービスを選択すると、プロセスは操作６５０に進み、コンフリクトチェックが実行される。

【００５２】

操作６５０において、パッケージファイル３００のコンテンツは、サービスクラスタ２３０に送信することができる。一部の実装例においては、費用のかかる送信時間を減らすためにマニフェストのみを送信することができる。マニフェストは、パックレット作成時に各サービスクラスタが返信する情報の一部を含むことができる。その情報はパックレットにあるコンポーネントを記述するものである。マニフェストは、物理的なパックレットが存在しなくても、マニフェスト内のメタ情報を用いてコンフリクトを検出できるように設計される。次に、各サービスは、パッケージファイル３００のコンテンツ（または他の実装例のマニフェスト）を調べ、コンポーネントの詳細、コンポーネントのコンフリクト、ユーザがパッケージファイル３００のデプロイメント中に遭遇する可能性のある他のなんらかのデプロイメントの問題をソリューションフレームワーク２６に報告する。この情報をソリューションフレームワーク２６に返すことによって、コンフリクトチェックプロセスの制御を集中化することができる。コンフリクトが存在する場合、プロセスは操作６７０に進む。コンフリクトが存在しない場合、プロセスは操作６６０に進み、デプロイメントプロセスは継続する。他の実施形態においては、コンフリクトチェックのプロセスは、非同期的に実行することができる。すなわち、マルチスレッドプロセスで、ソリューションフレームワーク２６は同時にコンフリクトを検出する。このように、このプロセスは、非同期的に実行してもよく、線形（順次に）実行してもよい。

【００５３】

操作６７０において、ソリューションフレームワークが自動で修正することが可能なコンフリクトか否かの決定が行われる。ソリューションフレームワークが自動でコンフリクトを修正できる場合、ソリューションフレームワークは、自動でコンフリクトを修正し、プロセスを継続して操作６６０に進むことができる。ある実装例においては、デプロイメントインタフェース２２０を介して、修正されたコンフリクトをユーザに知らせる警告メッセージを出すことができる。ソリューションフレームワーク２６が自動でコンフリクトを修正できない場合、致命的なコンフリクトであることを表すエラーメッセージをユーザに提示することができる。そしてプロセスは操作６８０に進み、デプロイメントは停止される。さらに他の実装例においては、ユーザに、自動でコンフリクトを無効にしてコンフリクトを起こしているコンポーネントを上書きする能力を与えてもよい。そのような実装例においては、ユーザは、パッケージファイル３００のデプロイメントを試みる前に、この作業を行うことが必要になるだろう。コンフリクトが発生すると、そのコンポーネントは、パッケージファイル３００の新しいコンポーネントで上書きすることができる。

【００５４】

操作６６０において、選択したサービスのコンポーネントは、選択したコンポーネントに関するパックレットをサービスに送信することによってデプロイされる。各パックレットは、そのサービスに特定の一連のバイナリデータを含むことができる。選択したサービスは、データストリームをデシリアライズしてデプロイメント用のデータに戻す方法を知っていてもよい。データがサービスにデプロイされると、プロセスは操作６９０に続く。

【００５５】

操作６９０において、サービスをさらに選択するか否かの決定が行われる。追加のコン

ポーネントを他の一つのサービスにデプロイする必要がある場合、プロセスは、操作６４０に戻り、次のサービスが選択される。コンポーネントが全てデプロイされた場合、プロセスは終了操作に進む。

【００５６】

他の実装例においては、パックレットがソリューションフレームワーク２６からサービスクラスタ２３０に送信される前に、コンフリクトをチェックするため選択したコンポーネントを全て処理することができる。例えば、全てのサービスを一巡して、コンフリクト情報を、情報を集中的に処理するソリューションフレームワークで集めてもよい。そして、コンフリクトがなくなると、パックレットはサービスクラスタ２３０に送信される。このようにして、全てのコンフリクトを集中的に処理することができる。

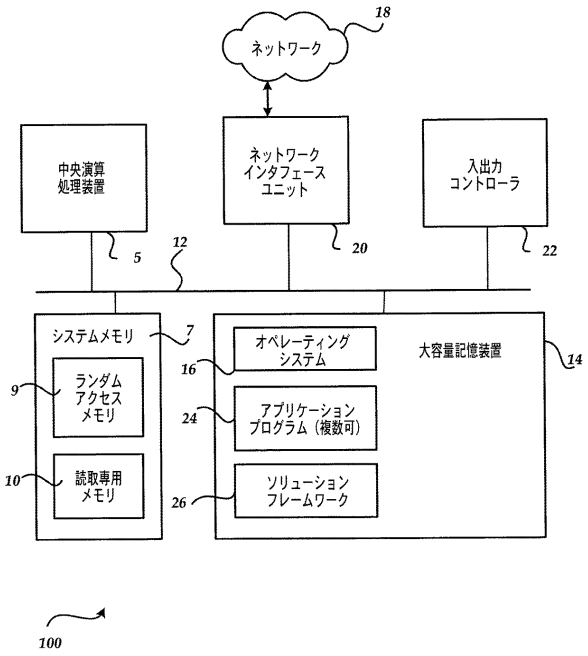
【００５７】

さらに、デプロイメント後のプロセスを実行することができる。例えば、パックレットをデプロイした後、各サービスクラスタは、デプロイされたコンポーネントに関する情報を報告することができる。このデプロイ後の情報は、次に、各サービスクラスタに返信することができる。このように、パッケージ全体のデプロイメントが完了すると、全てのサービスクラスタは、再度、通信を始める。こうすることによって、サービスクラスタは、システム全体にデプロイされた全てのコンポーネントの情報を得ることができる。このようにして、各クラスタは、その情報に基づいて、デプロイメント後の操作を実行することができる。例えば、第１のサービスクラスタのコンポーネントと第２のサービスクラスタのコンポーネントが、互いに密接に関連している場合があり、互いに関連する最終のデプロイメント情報を知ることによって、そのデプロイメントは、多くのばらばらの断片のデプロイメントではなく、エンドツーエンドの、密接につながった分散アプリケーションのデプロイメントであるという事実を強化するビジネスロジックを実行することができる。

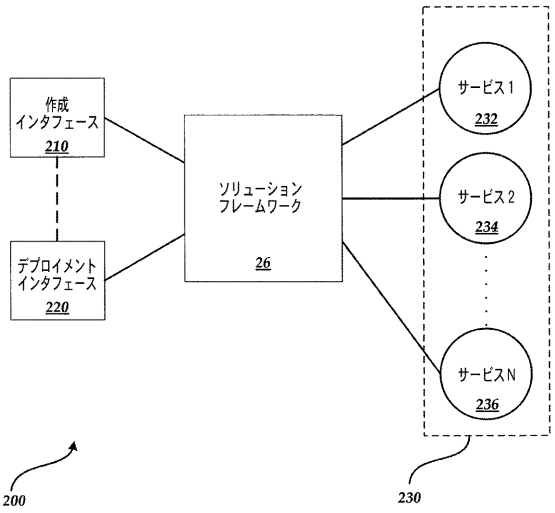
【００５８】

上述の記載、例、およびデータは、本発明の構成の製造および使用を完全に記述したものである。本発明の多くの実施形態は、本発明の精神と範囲を逸脱すること無しに行うことができるものであり、本発明は添付の請求項に帰する。

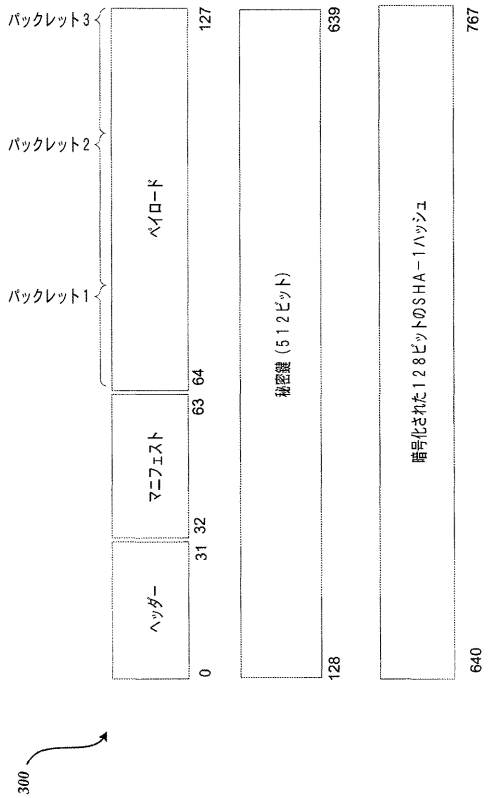
【図１】


ネットワーク
18
中央演算処理装置
ネットワークインタフェースユニット
入出力コントローラ
5
12
20
22
システムメモリ
7
ランダムアクセスメモリ
9
読取専用メモリ
10
オペレーティングシステム
16
アプリケーションプログラム（複数可）
24
ソリューションフレームワーク
26
大容量記憶装置
14
100

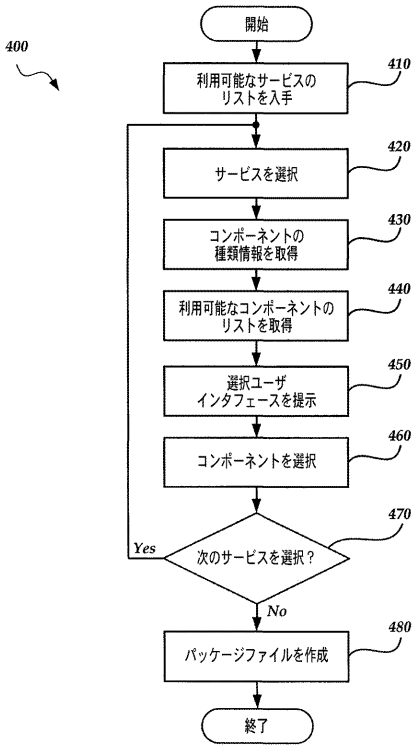
【図２】


作成インタフェース
210
デプロイメントインタフェース
220
ソリューションフレームワーク
26
サービス1
232
サービス2
234
サービスN
236
200
230

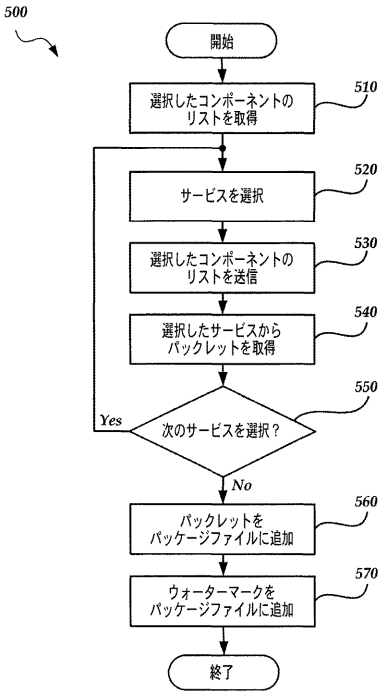
【図３】


バックレット3
バックレット2
バックレット1
ペイロード
マニフェスト
ヘッダー
127
64
63
32
31
0
639
128
767
640
秘密鍵（512ビット）
暗号化された128ビットのSHA-1ハッシュ
300

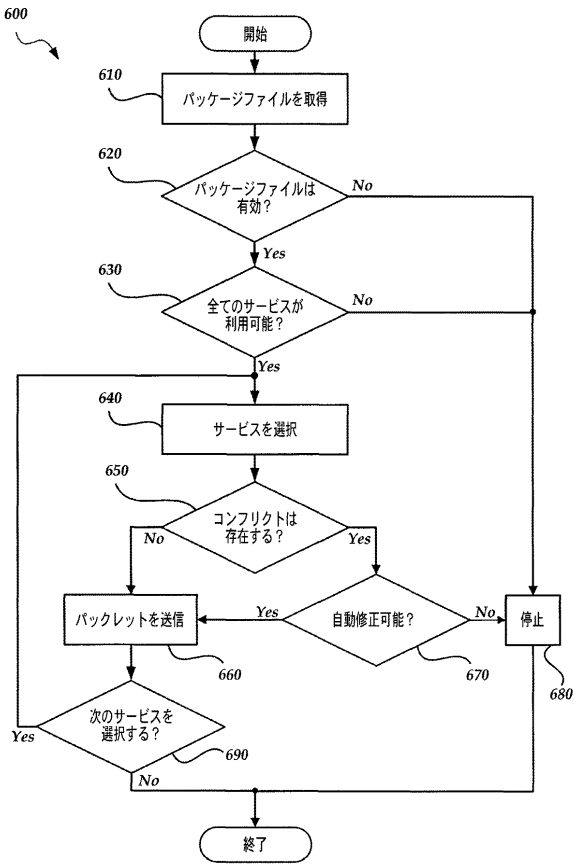
【図４】


400
開始
利用可能なサービスのリストを入手
410
サービスを選択
420
コンポーネントの種類情報を取得
430
利用可能なコンポーネントのリストを取得
440
選択ユーザインタフェースを提示
450
コンポーネントを選択
460
次のサービスを選択？
470
Yes
No
パッケージファイルを作成
480
終了

【図５】


500
開始
選択したコンポーネントのリストを取得
510
サービスを選択
520
選択したコンポーネントのリストを送信
530
選択したサービスからバックレットを取得
540
次のサービスを選択？
550
Yes
No
バックレットをパッケージファイルに追加
560
ウォーターマークをパッケージファイルに追加
570
終了

【図６】


600
開始
パッケージファイルを取得
610
パッケージファイルは有効？
620
No
Yes
全てのサービスが利用可能？
630
No
Yes
サービスを選択
640
コンフリクトは存在する？
650
No
Yes
パックレットを送信
660
Yes
自動修正可能？
670
No
停止
680
次のサービスを選択する？
690
Yes
No
終了

フロントページの続き

(72)発明者 クリストファー ジェイ．バイター
アメリカ合衆国 ９８０５２ ワシントン州 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 インターナショナル パテンツ内
(72)発明者 リチャード ダブリュ．トム
アメリカ合衆国 ９８０５２ ワシントン州 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 インターナショナル パテンツ内
(72)発明者 ラヴィクマール ビー．ゴピナート
アメリカ合衆国 ９８０５２ ワシントン州 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 インターナショナル パテンツ内
(72)発明者 ブライアン シー．ブロンクィスト
アメリカ合衆国 ９８０５２ ワシントン州 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 インターナショナル パテンツ内
(72)発明者 マドハビラサ ケニガンティ
アメリカ合衆国 ９８０５２ ワシントン州 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 インターナショナル パテンツ内
(72)発明者 デイビット チウ
アメリカ合衆国 ９８０５２ ワシントン州 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 インターナショナル パテンツ内

審査官 新井 寛

(56)参考文献 米国特許出願公開第２００３／０１８２６５６（ＵＳ，Ａ１）
特開２００６－３５０８５０（ＪＰ，Ａ）
特開２００７－０７２７１２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ９／５４
Ｇ０６Ｆ ９／４４５
Ｇ０６Ｆ １２／００