

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5201696号
(P5201696)

(45) 発行日 平成25年6月5日 (2013.6.5)

(24) 登録日 平成25年2月22日 (2013.2.22)

(51) Int. Cl.	F I
G 0 6 F 9/445 (2006.01)	G O 6 F 9/06 6 1 O A
G 0 6 F 9/46 (2006.01)	G O 6 F 9/46 3 5 O
G 0 6 F 21/62 (2013.01)	G O 6 F 21/24 1 6 5 C

請求項の数 13 (全 11 頁)

(21) 出願番号	特願2010-529375 (P2010-529375)	(73) 特許権者	390009531
(86) (22) 出願日	平成20年10月15日 (2008.10.15)		インターナショナル・ビジネス・マシーンズ・コーポレーション
(65) 公表番号	特表2011-501846 (P2011-501846A)		I N T E R N A T I O N A L B U S I N E S S M A C H I N E S C O R P O R A T I O N
(43) 公表日	平成23年1月13日 (2011.1.13)		アメリカ合衆国10504 ニューヨーク州 アーモンク ニュー オーチャードロード
(86) 国際出願番号	PCT/EP2008/063889		
(87) 国際公開番号	W02009/050206		
(87) 国際公開日	平成21年4月23日 (2009.4.23)	(74) 代理人	100108501
審査請求日	平成23年8月18日 (2011.8.18)		弁理士 上野 剛史
(31) 優先権主張番号	11/872,793	(74) 代理人	100112690
(32) 優先日	平成19年10月16日 (2007.10.16)		弁理士 太佐 種一
(33) 優先権主張国	米国 (US)	(74) 代理人	100091568
			弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】 仮想マシンを生成するための方法、システム、およびコンピュータ・プログラム

(57) 【特許請求の範囲】

【請求項 1】

仮想マシンを生成するための、コンピュータにより各ステップが実行される方法であって、

仮想マシン (VM) のための仮想マシン (VM) プロファイルおよび VM ファイル・システムを構成するステップであって、前記仮想マシン (VM) が、ハードウェア非依存の態様でソフトウェアを実行するためのソフトウェア生成仮想化環境であり、前記 VM プロファイルが、どのオペレーティング・システム (OS) を前記 VM によって用いるかを定義し、前記 VM ファイル・システムが、少なくとも 1 つの仮想アプリケーション・ディスクにおける仮想ファイルと少なくとも 1 つの物理ディスクにおける物理ファイルとの間のマップされた関係を定義する、前記構成するステップと、

前記少なくとも 1 つの仮想アプリケーション・ディスクにおけるファイルを、前記 VM における仮想 OS ディスクの特定のファイル・システム位置にリンクするステップであって、前記 VM が前記仮想 OS ディスクおよび前記少なくとも 1 つの仮想アプリケーション・ディスクを含む、前記リンクするステップと、
を有する、方法。

【請求項 2】

前記少なくとも 1 つの仮想アプリケーション・ディスクのためのセキュリティを構成するステップを更に有する、請求項 1 に記載の方法。

【請求項 3】

前記構成されたセキュリティが、前記少なくとも1つの仮想アプリケーション・ディスクと前記VM内の前記仮想OSディスクとの間の新たにリンクされたアプリケーションを利用することをどのユーザに許可するかを制御する、請求項2に記載の方法。

【請求項4】

前記構成されたセキュリティが、前記少なくとも1つの仮想アプリケーション・ディスクにおけるファイルについてのファイル設定を変更することをどのユーザに許可するかを制御する、請求項3に記載の方法。

【請求項5】

前記構成されたセキュリティ、および、前記少なくとも1つの仮想アプリケーション・ディスクにおける前記ファイルの仮想オペレーティング・システム(OS)・ディスクにおける前記特定のファイル・システム位置に対するファイルのリンクが、前記少なくとも1つの仮想アプリケーション・ディスクにおけるエージェントによって実行される、請求項4に記載の方法。

10

【請求項6】

前記VMがハイパーバイザによって活性化され、前記ハイパーバイザが、多数のオペレーティング・システムが同一のホスト・コンピュータ上で稼働することを可能とする仮想化プラットフォームである、請求項5に記載の方法。

【請求項7】

前記VMが複数の仮想アプリケーション・ディスクを含み、前記複数の仮想アプリケーション・ディスクの各々が異なるベンダーからのものである、請求項1に記載の方法。

20

【請求項8】

プロセッサと、
前記プロセッサに結合されたシステム・バスと、
前記システム・バスに結合されたメモリとを備え、
前記メモリに格納されたコンピュータ・プログラム・コードが含む命令が、前記プロセッサによって実行可能であり、前記命令によって前記プロセッサは、

仮想マシン(VM)のための仮想マシン(VM)・プロファイルおよびVMファイル・システムを構成するステップであって、前記仮想マシン(VM)が、ハードウェア非依存の態様でソフトウェアを実行するためのソフトウェア生成仮想化環境であり、前記VMプロファイルが、どのオペレーティング・システム(OS)を前記VMによって用いるかを定義し、前記VMファイル・システムが、少なくとも1つの仮想アプリケーション・ディスクにおける仮想ファイルと少なくとも1つの物理ディスクにおける物理ファイルとの間のマップされた関係を定義する、前記構成するステップと、

30

前記少なくとも1つの仮想アプリケーション・ディスクにおけるファイルを、前記VMにおける仮想OSディスクの特定のファイル・システム位置にリンクするステップであって、前記VMが前記仮想OSディスクおよび前記少なくとも1つの仮想アプリケーション・ディスクを含む、前記リンクするステップと、
を実行することによって仮想マシン(VM)を生成するために構成されている、システム。

【請求項9】

40

前記命令によって前記プロセッサは、
前記少なくとも1つの仮想アプリケーション・ディスクのためのセキュリティを構成するために更に構成されている、請求項8記載のシステム。

【請求項10】

前記構成されたセキュリティが、前記VMを利用することをどのユーザに許可するかを制御する、請求項9に記載のシステム。

【請求項11】

前記構成されたセキュリティが、前記少なくとも1つの仮想アプリケーション・ディスクにおけるファイルについてのファイル設定を変更することをどのユーザに許可するかを制御する、請求項10に記載のシステム。

50

【請求項 1 2】

前記 VM が複数の仮想アプリケーション・ディスクを含み、前記複数の仮想アプリケーション・ディスクの各々が異なるベンダーからのものである、請求項 8 に記載のシステム。

【請求項 1 3】

請求項 1 ないし請求項 7 の何れか 1 項に記載の方法の各ステップをコンピュータに実行させるためのコンピュータ・プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータの分野に関し、具体的にはソフトウェアに関する。更に具体的には、本発明は、仮想マシン（VM: Virtual Machine）に関する。

【背景技術】

【0002】

仮想マシン（VM）は、その特定のインスタンスが仮想イメージを用いる仮想アプライアンスと称され、ソフトウェアのための新しい配布手段（distribution vehicle）として現れたものである。各 VM は、典型的に、オープンソースの Linux（R）OS 等の固定オペレーティング・システム（OS）から成り、これが仮想コンテナ（純粋にソフトウェア・ベース）において 1 つ以上のアプリケーションを稼働させる。

【発明の概要】

【発明が解決しようとする課題】

【0003】

サード・パーティのベンダーからのコードを用いて仮想マシンを生成する方法およびシステムを提供する。

【課題を解決するための手段】

【0004】

各々を異なるサード・パーティのベンダー（供給業者）からのものとすることができる仮想オペレーティング・システム（OS）・ディスクおよび仮想アプリケーション・ディスクを用いて、仮想マシン（VM）を生成する。仮想アプリケーション・ディスクは、VM において交換可能であり、複数のアプリケーションから選択可能であり、VM プロファイルに従って VM において利用される。そして、仮想アプリケーション・ディスクにおける 1 つ以上のファイルを、仮想 OS ディスクにおける特定のファイル・システム位置（ノード）にリンクする。

【0005】

本発明の上述および追加の目的、特徴、および利点は、以下の詳細な記載において明らかとなろう。

【0006】

これより、以下の図面を参照して、一例としてののみ、本発明の好適な実施形態について説明する。

【図面の簡単な説明】

【0007】

【図 1】本発明を実装することができる例示的な物理コンピュータを示すブロック図である。

【図 2】本発明によって生成される例示的な仮想マシン（VM）を示す図である。

【図 3】図 2 に示した VM が用いる仮想アプリケーション・ディスクについて更に詳細に示す図である。

【図 4】サード・パーティのベンダーからのコードを用いて VM を生成するために本発明によって行われる例示的なステップを示す高レベルのフローチャートである。

【発明を実施するための形態】

【0008】

当業者には明らかであるように、本発明は、方法、システム、またはコンピュータ・プログラムとして具現化することができる。従って、本発明は、全体的にハードウェアの実施形態、全体的にソフトウェアの実施形態（ファームウェア、常駐ソフトウェア、マイクロコード等を含む）、またはソフトウェアおよびハードウェアの側面を組み合わせた実施形態という形態を取ることができ、これらは全て、本明細書においては一般的に「回路」、「モジュール」、または「システム」と称することができる。更に、本発明は、コンピュータ使用可能記憶媒体上に具現化されたコンピュータ使用可能プログラム・コードを有するコンピュータ・プログラムとして具現化することも可能である。

【0009】

いずれかの適切なコンピュータ使用可能またはコンピュータ可読媒体を利用することができる。例えば、コンピュータ使用可能またはコンピュータ可読媒体は、電子、磁気、光、電磁、赤外線、または半導体システム、装置、デバイス、または伝搬媒体等とすることができる。コンピュータ可読媒体の更に具体的な例は、1つ以上のワイヤを有する電氣的接続、携帯型コンピュータ・ディスク、ハード・ディスク、ランダム・アクセス・メモリ（RAM）、読み取り専用メモリ（ROM）、消去可能プログラマブル読み取り専用メモリ（EEPROMもしくはフラッシュ・メモリ）、光ファイバ、携帯型コンパクト・ディスク読み取り専用メモリ（CD-ROM）、光記憶デバイス、インターネットもしくはイントラネットをサポートする伝送媒体、または磁気記憶デバイス等を含む。コンピュータ使用可能またはコンピュータ可読媒体は、プログラムが印刷された紙または別の適切な媒体とすることも可能であることに留意すべきであり、この場合は、プログラムを、例えば紙または他の媒体の光学スキャンによって電子的に捕獲し、次いでコンパイルし、解釈、または必要な場合には他の適切な方法で処理し、次いでコンピュータ・メモリに記憶することができる。本明細書の文脈では、コンピュータ使用可能またはコンピュータ可読媒体は、命令実行システム、装置、もしくはデバイスによって用いるか、またはそれらと関連付けて用いるために、プログラムを保持し、格納し、伝達し、伝搬し、または移送することができるいずれかの媒体とすれば良い。コンピュータ使用可能媒体は、ベースバンドにおいてまたは搬送波の一部としてコンピュータ使用可能プログラム・コードが具現化された伝搬データ信号を含むことができる。コンピュータ使用可能プログラム・コードは、インターネット、ワイヤライン、光ファイバ・ケーブル、RF等を含むいずれかの適切な媒体を用いて、伝送することができる。

【0010】

本発明の動作を実行するためのコンピュータ・プログラム・コードは、Java（R）、Smalltalk、C++等のオブジェクト指向プログラミング言語で記述することができる。しかし、本発明の動作を実行するためのコンピュータ・プログラム・コードは、「C」プログラミング言語または同様のプログラミング言語等の従来の手順プログラミング言語で記述することも可能である。プログラム・コードは、全体的にユーザのコンピュータ上で、部分的にユーザのコンピュータ上で、スタンドアロンのソフトウェア・パッケージとして、部分的にユーザのコンピュータ上および部分的に遠隔コンピュータ上で、または全体的に遠隔コンピュータ上もしくはサーバ上で、実行することができる。後者の場合には、遠隔コンピュータを、ローカル・エリア・ネットワーク（LAN）もしくはワイド・エリア・ネットワーク（WAN）を介してユーザのコンピュータに接続することができ、または、この接続は、（例えば、インターネット・サービス・プロバイダを用いてインターネットを介して）外部のコンピュータに対して行うことも可能である。

【0011】

以下、本発明の実施形態に従った方法、装置（システム）、およびコンピュータ・プログラムのフローチャートおよび／またはブロック図を参照して、本発明を説明する。フローチャートおよび／またはブロック図の各ブロック、およびフローチャートおよび／またはブロック図におけるブロックの組み合わせは、コンピュータ・プログラム命令によって実装可能であることは理解されよう。これらのコンピュータ・プログラム命令を、汎用コンピュータ、特殊目的コンピュータ、または他のプログラマブル・データ処理装置のプロ

セッサに提供すると、当該コンピュータまたは他のプログラマブル・データ処理装置のプロセッサ上で実行される命令が、フローチャートおよび／またはブロック図の1つ以上のブロックにおいて特定された機能／行為を実装するための手段を生成することを目的として、機械を生成することができる。

【0012】

また、これらのコンピュータ・プログラム命令を、コンピュータ可読メモリに格納すると、当該コンピュータ可読メモリに格納された命令が、フローチャートおよび／またはブロック図の1つ以上のブロックにおいて特定された機能／行為を実装する命令手段を含む製品を生成することを目的として、当該コンピュータまたは他のプログラマブル・データ処理装置に対し特定の態様で機能するように指示することができる。

10

【0013】

また、コンピュータ・プログラム命令を、コンピュータまたは他のプログラマブル・データ処理装置上にロードすると、そのコンピュータまたは他のプログラマブル装置上で実行される命令が、フローチャートおよび／またはブロック図の1つ以上のブロックにおいて特定された機能／行為を実装するためのプロセスを提供することを目的として、コンピュータ実装プロセスを生成するように当該コンピュータまたは他のプログラマブル装置上で一連の動作ステップを実行させることができる。

【0014】

ここで図1を参照すると、本発明を利用することができる例示的なコンピュータ100のブロック図が示されている。コンピュータ100は、システム・バス106に結合されたプロセッサ・ユニット104を含む。また、システム・バス106には、ディスプレイ110を駆動／サポートするビデオ・アダプタ108も結合されている。システム・バス106は、バス・ブリッジ112を介して入出力(I/O)バス114に結合されている。I/Oバス114には、I/Oインタフェース116が結合されている。I/Oインタフェース116は、キーボード118、マウス120、コンパクト・ディスク読み取り専用メモリ(CD-ROM)ドライブ112、およびフラッシュ・メモリ・ドライブ126を含む様々なI/Oデバイスとの通信を提供する。I/Oインタフェース116に接続されたポートのフォーマットは、ユニバーサル・シリアル・バス(USB)ポート等を含む、当業者において周知のいずれかとすれば良い。

20

【0015】

コンピュータ100は、システム・バス106に結合されたネットワーク・インタフェース130を用いてネットワーク128を介してサーバ150と通信を行うことができる。ネットワーク128は、インターネット等の外部ネットワーク、イーサネット(R)または仮想プライベート・ネットワーク(VPN)等の内部ネットワークとすることができる。サーバ150は、コンピュータ100について示した態様でアーキテクチャ的に構成することができる。

30

【0016】

また、システム・バス106には、ハード・ドライブ・インタフェース132も結合されている。ハード・ドライブ・インタフェース132は、ハード・ドライブ134とインタフェースする。一実施形態においては、ハード・ドライブ134は、システム・メモリ136をポピュレートする。システム・メモリ136もシステム・バス106に結合されている。システム・メモリ136は、コンピュータ100における最下位レベルの揮発性メモリとして定義される。この揮発性メモリは、キャッシュ・メモリ、レジスタ、およびバッファを含む付加的な上位レベルの揮発性メモリ(図示せず)等を含むことができる。システム・メモリ136をポピュレートするコードは、オペレーティング・システム(OS)138およびアプリケーション・プログラム144を含む。

40

【0017】

OS138は、アプリケーション・プログラム144等のリソースに対するトランスペアレントなユーザ・アクセスを提供するためのシェル140を含む。一般に、シェル140(UNIX(R)において呼ばれているように)は、ユーザとオペレーティング・シス

50

テムとの間のインタープリタおよびインタフェースを提供するプログラムである。シェル 140 は、システム・プロンプトを提供し、キーボード 118、マウス 120、または他のユーザ入力媒体によって入力されたコマンドを解釈し、解釈した1つ以上のコマンドを処理するために適切な下位レベルのオペレーティング・システム（例えばカーネル 142）に送信する。図示するように、OS 138 はカーネル 142 も含み、これは OS 138 のための下位レベルの機能性を含む。カーネル 142 は、OS 138 の他の部分およびアプリケーション・プログラム 144 によって必要とされる不可欠なサービスを提供する。カーネル 142 が提供するサービスは、メモリ管理、プロセスおよびタスク管理、ディスク管理、および I/O デバイス管理を含む。

【0018】

アプリケーション・プログラム 144 は、ブラウザ 146 を含む。ブラウザ 146 は、ワールド・ワイド・ウェブ（WWW）のクライアント（すなわちコンピュータ 100）が、ネットワーク・メッセージをインターネットに送信および受信することを可能とするプログラム・モジュールおよび命令を含む。コンピュータ 100 は、サーバ 150 との通信を可能とするために、ハイパーテキスト転送プロトコル（HTTP）メッセージングを利用することができる。また、システム・メモリ 136 におけるアプリケーション・プログラム 144 は、仮想マシン・アセンブラ（VMA）148 を含む。VMA 148 は、以下で図 2 ～ 図 4 を参照して説明するステップで生成される、仮想マシン（VM）を実行する。VMA 148 は、多数の動作システムを管理し監督するハイパーバイザ（仮想マシン・マネージャ（VMM）としても知られる）の機能性を含むことができるが、かかるハイパーバイザのユーザは本発明にとって不可欠ではないことに留意すべきである。すなわち、本発明にとって、ハイパーバイザが本明細書に記載される VM の動作に関与するか否かは重要でない。更に、VM が、仮想 OS ディスクにおける特定のノードにアプリケーションをリンクするための本明細書に記載した特徴を用いる限り、本発明の使用は、いずれかの特定の仮想マシンの実装に限定されない。

【0019】

一実施形態においては、コンピュータ 100 は、好ましくは「オン・デマンド」ベースで、サービス・プロバイダ・サーバ 150 から VMA 148 をダウンロードすることができる。

【0020】

コンピュータ 100 内に図示したハードウェア要素は、網羅的であることは意図されず、本発明によって必要な不可欠なコンポーネントを強調するために表されている。例えば、コンピュータ 100 は、磁気カセット、デジタル・バーサタイル・ディスク（DVD）、ベルヌーイ・カートリッジ（Bernoulli cartridge）等の代替的なメモリ記憶デバイスを含むことができる。本発明において、これらおよび他の変形が可能である。サービス・プロバイダ・サーバ 150 のためのハードウェア・アーキテクチャは、コンピュータ 100 について示したものと実質的に同様とすることができることに留意すべきである。

【0021】

ここで図 2 を参照すると、以下に説明する態様で異なるベンダーからのコードおよびファイルをアセンブルすることによって生成された仮想マシン（VM）202 の概要が示されている。図 1 に示した VMA 148 によって生成されてその一部となる VM 202 は、ハードウェア非依存型（hardware-agnostic）でソフトウェアを実行するためのソフトウェア生成仮想化環境であり、仮想オペレーティング・システム（OS）・ディスク 204 および 1 つ以上の仮想アプリケーション・ディスク 206 a ～ 206 b を含む。仮想 OS ディスク 204 および仮想アプリケーション・ディスク 206 a ～ 206 b は、全て論理ディスクであり、1 つ以上の物理ディスク 218 上に記憶されたデータの論理グループ化として定義されることに留意すべきである。VM プロファイル 208 は、物理ディスク 218 上に位置する多数の利用可能なアプリケーション・システムからの仮想アプリケーション・ディスク 206 a ～ 206 b の参照インスタンスを有する。一実施形態においては、仮想アプリケーション・ディスク 206 a は、仮想アプリケーション・ディスク 206

bにおいて見出されるアプリケーションとは異なるソフトウェア・アプリケーションであり、従って、異なるベンダー／ソースからのものとすることができる。このため、各仮想アプリケーション・ディスク206a～206bは、1つ以上の物理ディスク218上にある場合がある複数のパーティションから成る。

【0022】

各仮想OSディスク204および仮想アプリケーション・ディスク206から稼働するOSおよびアプリケーションは、ファイル・システムの観点からノードを有するツリーとして編成される。アプリケーションを適正に実行するために、各仮想アプリケーション・ディスク206における1つ以上のディレクトリ・ノードは、仮想OSディスク204におけるOSのディレクトリ・ノードにマップ（ファイル・リンク）される。例えば、仮想アプリケーション・ディスク206bが、WebSphere（R）という名前のアプリケーション・サービスのためのものであり、ノード220におけるファイル・ディレクトリ（例えば、／WebSphere）が、ノード222において見出されるLinuxOSディレクトリの下になければならない（例えば、／opt／IBM／WebSphere）と仮定する。従って、ノード220はノード222にマップされる。このマッピングは、仮想アプリケーション・ディスクに対するシステム・シンボリック・リンクによって実行される。この点で、／opt／IBM／WebSphereの下にあるOSに見えるプログラムは、実際は、仮想アプリケーション・ディスク206b上にある。

【0023】

ここで図3を参照すると、仮想アプリケーション・ディスク302（図2に示した仮想アプリケーション・ディスク206aまたは206bと同等）について更に詳細に提示されている。仮想アプリケーション・ディスク302内にエージェント304がある。エージェント304は、スマートなエージェントであり、セキュリティを構成し（システム内にリンクされているアプリケーション・ディスクに関連した新しいユーザを生成すること、または新しいもしくは既存のシステム・ユーザに関連したファイル・レベルのセキュリティ設定を変更することから成る）、仮想アプリケーション・ディスク302内にあるファイルについて、他の仮想ディスク（他の仮想アプリケーション・ディスクおよび仮想OSディスクの双方）に対するリンクを生成することができる。エージェント304は、仮想アプリケーション・ディスク302におけるファイルを、図2に示した仮想OSディスク204の特定のパスまたはディレクトリにリンクする柔軟性を有する。

【0024】

ここで図4を参照すると、仮想マシン（VM）を生成するために行われる例示的なステップのフローチャートが示されている。開始ブロック402は、仮想マシンの入力状態を表し、これは仮想OSディスクを保持するVMプロファイルを含む。上述のように、VMは、ハードウェア非依存の態様でソフトウェアを実行するためのソフトウェア生成仮想化環境である。VMプロファイルは、VM内で開始される仮想OSディスクを定義する。次いで、VMプロファイル内に新しい仮想アプリケーション・ディスクを構成し（ブロック404）、それらをVMに見えるようにする。次いで、仮想アプリケーション・ディスクにおける1つ以上のファイルおよび／またはディレクトリを、VMにおける仮想OSディスクの特定のファイル・システム位置（例えばノード）にリンクする（ブロック406）。従って、VMは、仮想OSディスクおよび少なくとも1つの仮想アプリケーション・ディスクを含み、当該仮想アプリケーション・ディスク内に保持される1つ以上のアプリケーションをVMに利用可能とする。次いで、図3において上述したように、適切なセキュリティを構成し（ブロック408）、終了ブロック410において、仮想OSディスクおよびリンクされた仮想アプリケーション・ディスクの出力がVM202として配布されると、プロセスは終了する。

【0025】

上述したように、本発明は、異なるベンダーからの仮想アプリケーション・ディスクを保持する単一の仮想イメージをアセンブルする機能を提供する。例えば、Linux（R）OS配布は、IBM（R）のWebSphere（R）アプリケーションを含む第2の

仮想アプリケーション・ディスクと一体にすることができる。各仮想アプリケーション・ディスクのインストールは1度だけ実行され、次いでLinux(R)ファイル・リンクを用いてアセンブルすることができる。このため、VMにおける特定のOSについて、各アプリケーションを別個に選択し、配布し、維持することができる。

【0026】

本発明によって提供される利点は、次の通りである。

1. 柔軟性 — アプライアンス(例えばVM)のベンダーは、使用可能なオペレーティング・システム配布について、より多くの選択肢を有する。これらのベンダーは、別個に生成し提供することができる完全サポート配布(例えばRed Hat(R))を選択することができる。従来技術においては、VMの一部として特定のOSが構築され配布されたので、ユーザは好みの別のOSを利用することができなかった。本発明は、アプリケーションの保守および配布からOSの保守および配布を分離するための方法を提供する。更に、VMのユーザ(カスタマ)は、OSのためのパッチ、サービス・パック、更新等を提供するために、もはやVMのプロバイダーに頼らなくて済む。

2. 1度のインストール/構成 — オペレーティング・システムならびにミドルウェアおよびアプリケーションは双方とも、1度インストールするだけである。簡単なアセンブリ・ステップがあり、その後は、イメージを何度も用いることができる。

3. ミドルウェアおよびアプリケーションからのOSのクリーンな分離。いかなるミドルウェアおよびアプリケーションからもオペレーティング・システムをクリーンに分離して、オペレーティング・システムまたはアプリケーションの完全な置換またはパッチングによる、自動化されて独立した更新を可能とする。これにより、カスタマの現在のシステム上で実行しているものとは異なるOSを有するVMを実行することに関するカスタマの懸念を取り除く。

【0027】

図面におけるフローチャートおよびブロック図は、本発明の様々な実施形態に従ったシステム、方法、およびコンピュータ・プログラムの可能な実装のアーキテクチャ、機能性、および動作を例示することに留意すべきである。この点で、フローチャートまたはブロック図における各ブロックは、特定された1つ以上の論理機能を実装するための1つ以上の実行可能命令を含む、コードのモジュール、セグメント、または部分を表すことができる。また、いくつかの代替的な実装においては、ブロックに記載された機能は、図面に記載された順序に従わずに実行される場合があることに留意すべきである。例えば、連続して示された2つのブロックが、関連する機能性に応じて、実際には実質的に同時に実行されることがあり、または、これらのブロックが逆の順序で実行されることもある。また、ブロック図および/またはフローチャートの各ブロック、および複数ブロックの組み合わせは、特定された機能もしくは行為を実行する特殊目的ハードウェア・ベース・システムによって、または特殊目的ハードウェアおよびコンピュータ命令の組み合わせによって実装可能であることに留意すべきである。

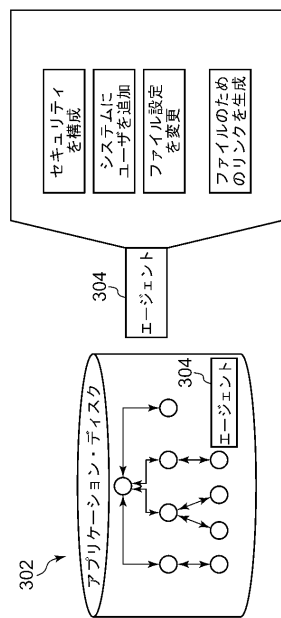
【0028】

本明細書において用いた用語は、特定の実施形態を説明するためだけのものであり、本発明を限定することは意図していない。本明細書において用いる場合、文脈によって明らかに他の場合が示されない限り、単数形は、複数の形態も含むように意図される。更に、本明細書において用いられる場合、「～を含む」および/または「～を含んでいる」という用語は、記載された特徴、整数、ステップ、動作、要素、またはコンポーネントあるいはそれら全ての存在を特定するが、1つ以上の他の特徴、整数、ステップ、動作、要素、コンポーネント、またはそのグループあるいはそれら全ての存在または追加を除外しないことは理解されよう。

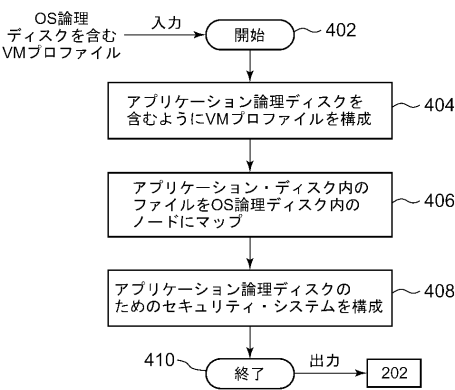
【0029】

特許請求の範囲において、全てのミーンズ・オア・ステップ・プラス・ファンクション要素の対応する構造、材料、行為、および均等物は、具体的に特許請求の範囲に記載された他の要素と組み合わせて機能を実行するためのいずれかの構造、材料、または行為を含

【図 3】



【図 4】



フロントページの続き

- (72)発明者 ウィレンボーク、ルース、エディス
アメリカ合衆国27523 ノースカロライナ州エイベックス ジェンクス・ロード 7617
- (72)発明者 ワン、チンボー
中華人民共和国100094 北京 ハイディアン・ディストリクト シェルチ ボヤデユアン
3-1-503
- (72)発明者 シュミット、ステファン、ジョン
アメリカ合衆国27614 ノースカロライナ州ラーレイ プラスフィールド・ロード 2504
- (72)発明者 ドイル、ロナルド、パトリック
アメリカ合衆国27615 ノースカロライナ州ラーレイ アボカド・サークル 10000

審査官 前田 浩

- (56)参考文献 米国特許出願公開第2007/0165937 (US, A1)
米国特許出願公開第2005/0268336 (US, A1)
米国特許第06195650 (US, B1)
特開2001-022713 (JP, A)
特開2006-155583 (JP, A)
特表2005-514699 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/445

G06F 9/46

G06F 21/62