(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5214473号
(P5214473)
(45)発行日　平成25年6月19日(2013.6.19)　(24)登録日　平成25年3月8日(2013.3.8)

(51)Int.Cl.　F I
G06F 9/46 (2006.01)　G06F 9/46 350

請求項の数 10 (全 21 頁)

(21)出願番号	特願2008-557282 (P2008-557282)
(86)(22)出願日	平成19年2月13日 (2007.2.13)
(65)公表番号	特表2009-528620 (P2009-528620A)
(43)公表日	平成21年8月6日 (2009.8.6)
(86)国際出願番号	PCT/US2007/004047
(87)国際公開番号	WO2007/100508
(87)国際公開日	平成19年9月7日 (2007.9.7)
審査請求日	平成22年1月7日 (2010.1.7)
(31)優先権主張番号	11/363,897
(32)優先日	平成18年2月28日 (2006.2.28)
(33)優先権主張国	米国 (US)

前置審査

(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　100096013
弁理士　富田　博行
(74)代理人　100092967
弁理士　星野　修
(74)代理人　100153028
弁理士　上田　忠

最終頁に続く

(54)【発明の名称】ハードウェアデバイスなどのリソースを有する仮想マシンの移動システム

(57)【特許請求の範囲】
【請求項１】
リソース・サービスを提供するリソースと、
インスタンス化された第１及び第２の複数の仮想マシン（ＶＭ）を有する計算機であって、それぞれのＶＭは１又は複数のアプリケーションがその上でインスタンス化され得るオペレーティング・システムのインスタンスをホストし、前記第１のＶＭが前記リソースと前記リソースによって提供されるサービスとを当初から有するように前記第１のＶＭは当初から前記リソースに通信可能に結合され且つ前記リソースは前記第１のＶＭに当初から割り当てられており、前記第１のＶＭが保存可能且つ第１のプラットフォームから第２のプラットフォームに移動可能なこの計算機上のソフトウェア構成物であるものと、
を備える計算システムにおいて、
前記第１のＶＭが、
前記リソースに対応するリソース・スタックであって、このリソース・スタック経由で送られるアクセス・リクエストに従って前記リソースにアクセスするリソース・スタックと、
前記リソースに通信可能に結合された第１のポートと、
通信媒体に通信可能に結合された第２のポートと、
前記リソース・スタックと前記第１及び第２のポートとに通信可能に結合されており、前記リソース・スタックからのそれぞれのアクセスを前記第１及び第２のポートの一方においてキューされるように送るポート・リダイレクタと、

を含み、
前記ポート・リダイレクタは、前記第１のＶＭが保存されるか又は移動されるように命じられるまで前記リソース・スタックからのそれぞれのアクセス・リクエストを前記第１のポートにおいてキューされるように送り、前記第１のポートにおけるそれぞれのアクセス・リクエストは前記リソースによって作用を受ける前記リソースに更に送られ、
前記ポート・リダイレクタは、前記第１のＶＭが保存されるか又は移動されるとき及びその後に前記リソース・スタックからのそれぞれのアクセス・リクエストを前記第２のポートに送り、前記第２のポートにおけるそれぞれのアクセス・リクエストは、前記リソースが前記第１のポートにおいてキューされているすべてのアクセス・リクエストに作用し次いで前記第１のＶＭによって所有されている状態から削除された後でのみ更に送られ、
前記第２のＶＭが次いで前記リソースと前記リソースによって提供されるサービスとを所有するように前記第２のＶＭが次いで前記リソースに通信可能に結合され且つ前記リソースが次いで前記リソースが前記第１のＶＭから削除された後に前記第２のＶＭに割り当てられ、それによって前記リソースの所有者としての前記第２のＶＭが前記通信媒体を介して前記第１のＶＭの前記第２のポートに通信可能に結合され、前記第２のポートにおけるそれぞれのアクセス・リクエストが前記通信媒体を介して前記第２のＶＭに送られ、更に、前記リソースによって作用される前記第２のＶＭを介して前記リソースに送られ、
よって、前記リソースが前記第１のＶＭから削除され前記第２のＶＭに割り当てられた後でも前記第１のＶＭの前記リソース・スタックからのすべてのアクセス・リクエストが前記リソースによって作用されて、前記第１のＶＭの保存又は移動を完了することが可能であることを特徴とする計算システム。
【請求項２】
請求項１記載の計算システムにおいて、前記計算機は、前記第１のＶＭと前記第２のＶＭとを通信可能に結合する通信媒体としてＶＭバスを更に有することを特徴とする計算システム。
【請求項３】
請求項１記載の計算システムにおいて、前記計算機は前記リソースを有していることを特徴とする計算システム。
【請求項４】
請求項１記載の計算システムにおいて、前記リソースはハードウェア・リソースであることを特徴とする計算システム。
【請求項５】
請求項１記載の計算システムにおいて、前記リソースは記憶装置リソースであることを特徴とする計算システム。
【請求項６】
リソース・サービスを提供するリソースと、
インスタンス化された第１及び第２の複数の仮想マシン（ＶＭ）を有する計算機であって、それぞれのＶＭは１又は複数のアプリケーションがその上でインスタンス化され得るオペレーティング・システムのインスタンスをホストし、前記第１のＶＭが前記リソースと前記リソースによって提供されるサービスとを当初から有するように前記第１のＶＭは当初から前記リソースに通信可能に結合され且つ前記リソースは前記第１のＶＭに当初から割り当てられており、前記第１のＶＭが保存可能且つ第１のプラットフォームから第２のプラットフォームに移動可能なこの計算機上のソフトウェア構成物であるものと、
を備えており、前記第１のＶＭが、
前記リソースに対応するリソース・スタックであって、このリソース・スタック経由で送られるアクセス・リクエストに従って前記リソースにアクセスするリソース・スタックと、
前記リソースに通信可能に結合された第１のポートと、
通信媒体に通信可能に結合された第２のポートと、

前記リソース・スタックと前記第１及び第２のポートとに通信可能に結合されており、前記リソース・スタックからのそれぞれのアクセスを前記第１及び第２のポートの一方においてキューされるように送るポート・リダイレクタと、
を含む計算システムに関する方法であって、
前記ポート・リダイレクタが、前記第１のＶＭが保存されるか又は移動されるように命じられるまで前記リソース・スタックからのそれぞれのアクセス・リクエストを前記第１のポートにおいてキューされるように送るステップであって、前記第１のポートにおいてキューされているそれぞれのアクセス・リクエストが前記リソースによって作用を受ける前記リソースに更に送られる、ステップと、
前記第１のＶＭが保存されるか又は第１のプラットフォームから第２のプラットフォームに移動されるように命じられていると判断する第１のステップであって、前記第１の判断がされた時に、前記ポート・リダイレクタが、前記リソース・スタックからのそれぞれのアクセス・リクエストを前記第２のポートにおいてキューされるように送るステップと、
前記第１のポートにおいてキューされており前記第１のポートによって送られたすべてのアクセス・リクエストに前記リソースが作用していることを判断する第２のステップと、
を含んでおり、前記第２の判断がされた時に、
前記リソースを、前記第１のＶＭによって所有されている状態から削除し、次いで前記第２のＶＭが前記リソースと前記リソースによって提供されるサービスとを所有するように前記第２のＶＭを前記リソースに通信可能に結合し且つ前記リソースを前記第２のＶＭに割り当て、
更に、前記第２のＶＭを前記リソースの所有者として前記通信媒体を介して前記第１のＶＭの前記第２のポートに通信可能に結合し、
前記第２のポートにおいてキューされているそれぞれのアクセス・リクエストを前記通信媒体を介して前記第２のＶＭに送り、それぞれのアクセス・リクエストは前記リソースによって作用される前記第２のＶＭを介して前記リソースに送られ、
よって、前記リソースが前記第１のＶＭから削除され前記第２のＶＭに割り当てられた後でも前記第１のＶＭの前記リソース・スタックからのすべてのアクセス・リクエストが前記リソースによって作用されて、前記第１のＶＭの保存又は移動が完了することを特徴とする方法。
【請求項７】
請求項６記載の方法において、前記計算機は、前記第１のＶＭと前記第２のＶＭとを通信可能に結合する通信媒体としてＶＭバスを更に有することを特徴とする方法。
【請求項８】
請求項６記載の方法において、前記計算機は前記リソースを有していることを特徴とする方法。
【請求項９】
請求項６記載の方法において、前記リソースはハードウェア・リソースであることを特徴とする方法。
【請求項１０】
請求項６記載の方法において、前記リソースは記憶装置リソースであることを特徴とする方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ハードウェアデバイスやそれ以外の物理デバイスなどのリソースを有する仮想マシンを、第１のマシン又はプラットホームから第２のマシン又はプラットホームに移動するのに用いられる方法及び機構に関する。特に、本発明は、リソースに関する状態情報をまったく失うことなく、仮想マシンの移動を可能にする方法及び機構に関する。

【背景技術】
【０００２】
仮想マシンとは、ハードウェアシステムのエミュレーションを目的として計算機において動作するソフトウェア構築物である。必ずではないが典型的に、仮想マシンとは、アプリケーションなどであり、計算機において用いられてユーザアプリケーションなどをホストする。仮想マシンは、同時に、ユーザアプリケーションを計算機から隔離し、また、ユーザアプリケーションを計算機上にある他のアプリケーションから隔離する。複数の異なる計算機のそれぞれに対して様々な異なる仮想マシンを書くことができ、それにより、仮想マシンのために書かれた任意のユーザアプリケーションは異なる計算機の中の任意のものにおける動作が可能になる。したがって、それぞれの異なる計算機に対してであっても、異なる様々なユーザアプリケーションは不要である。
【０００３】
計算機のための新たなアーキテクチャと新たなソフトウェアとにより、単一の計算機が複数のパーティションをインスタンス化（instantiate）して動作させることが可能になる。ここで、複数のパーティションは、それぞれ、仮想マシンをインスタンス化して、その上で１又は複数のアプリケーションがインスタンス化されるオペレーティングシステムのインスタンスをホストするのに用いることができる。計算機は、必ずではないが典型的に、監督（overseer）アプリケーション又は「ハイパーバイザ」として機能する仮想マシンモニタを備えた仮想化レイヤを含む。ここで、仮想化レイヤは、それぞれの仮想マシンの管理的な側面を監督する、及び／又は、そうでない場合には管理し、それぞれの仮想マシンと当該仮想マシンの外部世界との間にある可能性のあるリンクとして機能する。
【０００４】
とりわけ、計算機上の特定の仮想マシンは、計算機と関連するリソースへのアクセスを要求することがある。そのようなリソースは、計算機と関連する可能性がある任意の種類のリソースでありうる。例えば、このリソースは、データを保存及び検索する記憶装置であり、記憶装置が用いられるどのような目的のための記憶装置でもかまわない。同様に、リソースは、ネットワーク、プリンタ、スキャナ、ネットワークドライブ、仮想ドライブ、サーバなど、任意の他の資産（アセット）でもありうる。したがって、リソースがどのようなものであっても、仮想マシンには、リソースによって提供されるサービスへのアクセスが与えられる。
【０００５】
複数のパーティションがインスタンス化されている計算機においては、その計算機の任意の特定のリソースは、特定のパーティション／仮想マシン（以下では、「仮想マシン」又は「ＶＭ」と称する）に動的に割り当てられる。それによって、この特定のＶＭは、当該リソースと当該計算機における他のＶＭからの当該リソースに対するサービスリクエストとを直接的に制御することができる。つまり、この特定のＶＭは、当該特定のリソースを「有する」リソースホストＶＭ（ＶＭ－Ｈ）として、リソース能力（capabilities）を提供するホストである。同様に、このＶＭ－Ｈは、そのような能力をリソースクライアントＶＭ（ＶＭ－Ｃ）として消費するクライアントである別のＶＭに、リソースサービスを提供する。このように、ＶＭ－ＣとＶＭ－Ｈとの組合せによって、特定のリソースの使用が必要となる動作が達成される。
【０００６】
計算機において動作する特定のＶＭ－Ｃは、現実のマシンであるかのように動作するように構築される。すなわち、特定のＶＭ－Ｃは、特定のリソースにアクセスする際には、特定のリソースが当該特定のリソースへの直接のリクエストによってアクセス可能であるかのように動作するのが典型的である。したがって、ＶＭ－Ｃは、そのようなリクエストが方向付けられるドライバの経路又はスタック（以下では、「スタック」と称する）を、特定のリソースが当該スタックの最後にあることを期待して、構築している場合がある。しかし、実際は、ＶＭ－Ｃは現実のマシンではなく、特定のリソースはスタックの最後に存在していない。

【０００７】
したがって、リソースが仮想化レイヤ／仮想マシンモニタによってスタックの最後に存在するものとしてエミュレートされる場合がある。実際には、仮想化レイヤは、リソースに対するリクエストを、当該リソースを有する又は当該リソースへのアクセスを有するＶＭ－Ｈに送る。同様に、ＶＭ－Ｃがそれ自体の仮想的存在を認識していて、ＶＭ－Ｃと当該リソースを有する又は当該リソースへのアクセスを有するＶＭ－ＨとをＶＭバス又はそれ以外の通信経路がその最後に存在している「拡張された（enlightened）」スタックを通じて、特定のリソースへのリクエストを送るという意味で、ＶＭ－Ｃに拡張された能力が与えられている場合がある。なお、ここで、ＶＭバスとは、仮想化レイヤをバイパスするものである。やはり同様に、拡張された能力を有するＶＭ－Ｃが、ＶＭバスを用いて実現されているＶＭ－ＣとＶＭ－Ｈとの間の仮想パイプを通じて、特定のリソースにリクエストを送る場合もある。どのような通信プロトコルが用いられるにしても、ＶＭ－Ｃは、ＶＭ－Ｈを通じて特定のリソースにアクセスするのであって、したがって、ＶＭ－Ｃによって特定のリソースに送られるそれぞれのリクエストは、対応するＶＭ－Ｈを通じて特定のリソースへの経路をたどる。

【０００８】
特に、特定のリソースを有するＶＭ－Ｈに関しては、このＶＭ－Ｈは、ＶＭ－Ｈに割り当てられたリソースのための適切なアダプタを通じて、リソースに直接にアクセスすることができる。必ずではないが典型的には、このアダプタは、ＶＭ－Ｈの計算機におけるハードウェア又はソフトウェアであり、このハードウェア又はソフトウェアはＶＭ－Ｈへの当該リソースのインタフェースを与える。例えば、このアダプタは、ネットワークインタフェースカードやビデオカードであり、あるいは、同等のソフトウェアでありうる。アダプタへの直接的なアクセスにより、ＶＭ－Ｈは、比較的高い効率及び性能をもって、リソースを用いることができる。特定のリソースが、それぞれが潜在的に特定のＶＭ－Ｈに割り当てられた複数の対応するアダプタを有していて、複数のＶＭ－Ｈが１つの特定のリソースを有することもありうる。しかし、どの時点においても、少なくとも典型的には、ただ１つのＶＭ－Ｈだけを特定のアダプタに割り当てる、すなわち、特定のアダプタを所有することができるのである。いずれにしても、典型的には、特定のアダプタの所有権は特定のアダプタ自体のリソースの所有権と同等であると想定できる。

【０００９】
ＶＭの品質保証（hallmark）とは、ＶＭが仮想的構築物として随意に停止させ再始動することができるということであり、更に、ＶＭが停止された場合には随意に保存し検索し再始動することができることである。特に、特定の計算機においてインスタンス化されているＶＭは、単一のソフトウェア構築物である。このソフトウェア構築物は、ＶＭに関係する動作データと状態情報とを含めて当該ソフトウェア構築物がＶＭに関係するすべてのデータを含む限り、適切なパッケージングが可能である。結果的に、第１の計算機におけるＶＭは、この第１の計算機においてＶＭを停止し、停止されたＶＭを第２の計算機に移動し、移動されたＶＭを第２の計算機において再始動することによって、第２の計算機に動かす、すなわち、「移動する」ことができる。より一般的には、あるＶＭを、同様の態様で、第１のプラットホームから第２のプラットホームに移動できる。ここで、これらのプラットホームは、複数の異なる計算機であるか、又は、同一の計算機の複数の異なる構成を表す。後者の場合には、例えば、追加的なメモリが加えられている、プロセッサが交換された、追加的な入力装置が提供されている、選択装置が取り外された、などの場合に、計算機は異なる構成を有することになる。

【００１０】
時には、ＶＭに関係する状態情報のすべてが当該ＶＭのソフトウェア構築物の中に含まれていないこともありうる。特に、リソース又はアダプタを有するＶＭ－Ｈは、当該リソース又は当該アダプタと共に記憶されているリソースに関係する特定の状態情報を有することがある。単なる例であるが、ネットワークであるリソースをＶＭ－Ｈが有しており、このＶＭ－Ｈが有する対応するアダプタがネットワークインタフェースカードである場合

には、ネットワークに対するリード及びライトコマンドのような状態情報は、少なくとも働きかけがあるまでは一時的に、ネットワークインタフェースカードに記憶される。他の例としては、リソースがアダプタを含む光ディスク読み出しドライブである場合には、ドライブに対するリードコマンドなどの状態情報は当該ドライブに記憶される。更に別の例であるが、リソースがプリンタであり対応するアダプタが印刷スプーラを含む場合には、プリンタに対するライトコマンドなどの状態情報は、そのスプーラに記憶される。

【００１１】

いずれにしても、ＶＭ－Ｈがリソースを有していて、有しているリソースに状態情報を記憶する場合のように、ＶＭの状態情報の一部がＶＭのソフトウェア構築物に含まれない場合には、ＶＭを第１のプラットホームから第２のプラットホームに移動することは、より困難になる。特に、そのような移動は、リソースにおける状態情報が失われないように処理され、又は、ＶＭから恒久的に分離されるように処理されるまで、生じさせるべきではない。

【００１２】

したがって、ＶＭ－Ｈを第１のプラットホームから第２のプラットホームに移動するときに、ＶＭ－Ｈの状態情報をＶＭ－Ｈが有するリソースにおいて処理する必要性が存在する。特に、移動の前にリソースにおける状態情報をその通常動作においてリソースから削除することができる、又は、別のＶＭ－Ｈによる処理が可能であるような方法及び機構に対する必要性が存在する。

【発明の概要】

【００１３】

上述した必要性は、本発明によって少なくとも部分的には達成される。本発明では、計算システムが、リソースサービスを提供するリソースと、第１及び第２の仮想マシン（ＶＭ）がその上でインスタンス化されている計算機とを有する。それぞれのＶＭは、１又は複数のアプリケーションがインスタンス化されるオペレーティングシステムのインスタンスをホストする。第１のＶＭは、最初に、リソースに通信可能な態様で結合されており、リソースは、最初に、第１のＶＭに割り当てられており、よって、第１のＶＭは、最初に、リソースとそれによって提供されるサービスとを有している。第１のＶＭは、計算機上のソフトウェア構築物であり、保存が可能で、第１のプラットホームから第２のプラットホームに移動することができる。

【００１４】

第１のＶＭは、リソースに対応するリソーススタックであって当該リソーススタックを通じて送られたアクセスリクエストに従ってリソースにアクセスするリソーススタックと、前記リソースに通信可能な態様で結合された第１のポートと、通信媒体に通信可能な態様で結合された第２のポートと、リソーススタックと第１のポートと第２のポートとに通信可能な態様で結合されたポートリダイレクタと、を含む。ポートリダイレクタは、それぞれのアクセスリクエストを、第１のポートと第２のポートとの一方においてキューされるリソーススタックから、送り出す。

【００１５】

特に、ポートリダイレクタは、第１のＶＭが保存される又は移動されることを命じられるまで、それぞれのアクセスリクエストを、第１のポートにおいてキューされるリソーススタックから送り出す。第１のポートにおけるそれぞれのアクセスリクエストは、リソースに向けて送られ、当該リソースによって作用を受ける。第１のＶＭが保存される又は移動されることを命じられると、及びそれ以降は、ポートリダイレクタは、それぞれのアクセスリクエストを、リソーススタックから第２のポートに送る。第２のポートにおけるそれぞれのアクセスリクエストは、第１のポートにおいてキューされたすべてのアクセスリクエストに対してリソースが働きかけ、それ以後に、第１のＶＭによる所有から取り除かれた後でのみ、送り出される。

【００１６】

リソースが第１のＶＭから取り除かれることにより第２のＶＭが当該リソースとそれに

よって提供されるサービスとを事後的に有することになった後で、第２のＶＭは事後的にリソースに通信可能な態様で結合され、リソースは事後的に第２のＶＭに割り当てられる。リソースの所有者としての第２のＶＭは、第１のＶＭの第２のポートに通信媒体を通じて通信可能な態様で結合され、第２のポートにおけるそれぞれのアクセスリクエストは、通信媒体を通じて第２のＶＭに送られ、更に、リソースによって働きかけられる第２のＶＭを通じてリソースに送られる。したがって、第１のＶＭのリソーススタックからのアクセスリクエストは、すべてが、リソースが第１のＶＭから取り除かれ第２のＶＭに割り当てられ保存又は移動が完了した後であっても、リソースによる働きかけを受ける。

【発明を実施するための最良の形態】

【００１７】

コンピュータ環境：

図１及び以下の説明では、本発明が実装可能である適切な計算環境についての簡潔な概説を提供することを意図している。しかし、ハンドヘルド、ポータブル及びあらゆる種類のそれ以外の計算機が本発明と関連して用いられることが想定されていると理解してほしい。以下では汎用コンピュータについて説明するが、これは単なる一例である。したがって、本発明は、非常にわずか又は最小限のクライアント・リソースだけしか含まれないネットワーク接続されホストされているサービス環境において実装されることがあるが、これは、例えば、クライアントデバイスがブラウザとしてすなわちワールドワイドウェブへのインタフェースとして機能するネットワーク環境である。

【００１８】

必須ではないが、本発明は、開発者によって用いられるために、アプリケーションプログラムインタフェース（ＡＰＩ）を介して実装することができる。及び／又は、本発明は、クライアントワークステーション、サーバ又は他のデバイスなど１又は複数のコンピュータによって実行されるプログラムモジュールなどのコンピュータ実行可能命令の一般的なコンテキストで説明されるネットワークブラウジングソフトウェアに含まれる。一般に、プログラムモジュールは、特定のタスクを実行する又は特定の抽象データタイプを実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含む。典型的には、プログラムモジュールの機能は、様々な実施例において希望されるように、組み合わされ、分配される。更に、当業者であれば理解するように、本発明は、他のコンピュータシステム構成でも実行可能である。本発明は、他の周知の計算システム、環境及び／又は構成においても使用が可能である。制限を意味しないが、パーソナルコンピュータ（ＰＣ）、自動預金受払機、サーバコンピュータ、ハンドヘルド又はラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースのシステム、プログラム可能な家電、ネットワークＰＣ、ミニコンピュータ、メインフレーム・コンピュータなどがその例である。本発明は、通信ネットワーク又は他のデータ伝送媒体によってリンクされる遠隔処理デバイスによってタスクが行なわれる分散コンピューティング環境中で実行されることもある。分散コンピューティング環境では、プログラムモジュールは、記憶装置デバイスを含むローカル及びリモートのコンピュータ記憶媒体の両方に位置することがある。

【００１９】

図１では、本発明が実装される適切な計算システム環境１００の例を図解している。上述したように、計算システム環境１００は適切なコンピュータ環境の単なる一例であり、本発明の使用又は機能の範囲に関するいかなる制限も示唆も意図もされていない。また、計算環境１００は、例示的な動作環境１００に図解されているコンポーネントのいかなる組合せにも限定されない。

【００２０】

図１を参照すると、本発明が実装される典型的なシステムは、汎用計算機をコンピュータ１１０の形で含んでいる。コンピュータ１１０の構成要素としては、限定的ではないが、演算処理装置１２０と、システムメモリ１３０と、システムメモリを含む様々なシステムコンポーネントを演算処理装置１２０に結合するシステムバス１２１とを含む。システ

ムバス１２１は、メモリバス又はメモリコントローラと周辺バスと様々なバスアーキテクチャの中の任意のものを用いるローカルバスとを含むいくつかのタイプのバス構造の中のいずれかである。例示であって限定を意味しないが、そのようなアーキテクチャには、インダストリ・スタンダード・アーキテクチャ（ＩＳＡ）バス、マイクロチャネル・アーキテクチャ（ＭＣＡ）バス、エンハンストＩＳＡ（ＥＩＳＡ）バス、ビデオ・エレクトロニクス・スタンダーズ・アソシエーション（ＶＥＳＡ）ローカルバス、ペリフェラル・コンポーネント・インタコネクト（ＰＣＩ）バス（メザニーン・バスとしても知られる）及びＰＣＩエクスプレスが含まれる。

【００２１】

コンピュータ１１０は、典型的には様々なコンピュータ可読媒体を含む。コンピュータ可読媒体は、コンピュータ１１０がアクセスできる任意の利用可能な媒体であり、揮発性及び不揮発性の媒体と、取り外し可能及び取り外し不可能な媒体を含む。例示であり限定は意味しないが、コンピュータ可読媒体は、コンピュータ記憶装置媒体及び通信メディアを含むことがある。コンピュータ記憶媒体は、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又はそれ以外のデータなどの情報の記憶のために任意の方法又は技術において用いられる揮発性・不揮発性、取り外し可能・取り外し不可能な媒体を含む。コンピュータ記憶媒体には、限定を意味しないが、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュ・メモリ又は他のメモリ技術、ＣＤＲＯＭ、ディジタルバーサタイルディスク（ＤＶＤ）又は他の光学ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置又は他の磁気記憶装置、又は所望の情報を格納するために使用することができコンピュータ１１０によってアクセスすることができるそれ以外の媒体も含まれる。通信媒体は、典型的には、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、又は、搬送波又は他の移送機構などの変調データ信号の中のそれ以外のデータを具体化するものであり、任意の情報配送媒体を含む。「変調データ信号」という用語は、信号中の情報をエンコードするように設定又は変更された特性の中の１又は複数を有する信号を意味する。例示であり限定ではないが、通信媒体には、有線ネットワークすなわちダイレクトワイアード接続又は光ファイバなどの有線媒体と、アコースティック、ＲＦ、赤外線、光、任意の波長で動作するフェーズドアレーアンテナ、任意の波長で動作する指向性及び非指向性型の電磁気エミッタ及び受信機、それ以外の無線媒体などが含まれる。上記のものの任意の組合せも、コンピュータ可読媒体の範囲に含まれる。

【００２２】

システムメモリ１３０は、コンピュータ記憶装置媒体を、リードオンリメモリ（ＲＯＭ）１３１及びランダムアクセスメモリ（ＲＡＭ）１３２など揮発性及び／又は不揮発性メモリの形で含む。始動中などにコンピュータ１１０内の要素間で情報を転送するのを支援する基本ルーチンを含む基本入出力システム１３３（ＢＩＯＳ）は、ＲＯＭ１３１に典型的に格納される。ＲＡＭ１３２は、典型的には、直ちにアクセス可能であり処理ユニット１２０が現在操作しているデータ及び／又はプログラムモジュールを含む。例示であり限定ではないが、図１は、オペレーティングシステム１３４、アプリケーションプログラム１３５、他のプログラムモジュール１３６及びプログラム・データ１３７を図解している。

【００２３】

コンピュータ１１０は、取り外し可能／不可能な、揮発性／不揮発性のコンピュータ記憶媒体なども含む。単なる例示であるが、図１は、取り外し不可能で不揮発性の磁気媒体との間で読み書きを行うハードディスクドライブ１４１と、取り外し可能で不揮発性の磁気ディスク１５２との間で読み書きを行う磁気ディスクドライブ１５１と、ＣＤＲＯＭ又は他の光学媒体など取り外し可能で不揮発性の光ディスク１５６との間で読み書きを行う光ディスクドライブ１５５とを図解している。他の取り外し可能／不可能、揮発性／不揮発性のコンピュータ記憶媒体も典型的な動作操作環境において使用される。例示でありこれらに限定されないが、磁気カセットテープ、フラッシュメモリカード、ディジタルバーサタイルディスク、ディジタルビデオテープ、ソリッドステートＲＡＭ、ソリッドステー

トＲＯＭなどが含まれる。ハードディスクドライブ１４１は、インタフェース１４０など取り外しが不可能なインタフェースを介してシステムバス１２１に典型的に接続される。また、磁気ディスク装置１５１及び光ディスクドライブ１５５は、インタフェース１５０などの取り外しが可能なメモリ・インタフェースによってシステムバス１２１に典型的に接続される。

【００２４】

上述した図１に図解されているドライブ及び関連するコンピュータ記憶媒体は、コンピュータ１１０のために、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及び他のデータためのストレージを提供する。図１では、例えば、ハードディスクドライブ１４１は、オペレーティングシステム１４４、アプリケーションプログラム１４５、他のプログラムモジュール１４６及びプログラム・データ１４７を格納するものとして図解される。これらのコンポーネントは、オペレーティングシステム１３４、アプリケーションプログラム１３５、他のプログラムモジュール１３６及び番組データ１３７と異なる場合も同一の場合もある。オペレーティングシステム１４４、アプリケーションプログラム１４５、他のプログラムモジュール１４６及びプログラム・データ１４７には異なる参照番号が与えられていて、少なくともそれらが異なるコピーであることが図解されている。ユーザは、キーボード１６２や、マウス、トラックボール又はタッチパッドと一般に呼ばれるポインティング・デバイス１６１などの入力装置を介してコンピュータ１１０にコマンドと情報とを入力する。他の入力装置（図示せず）として、マイクロホン、ジョイスティック、ゲームパッド、サテライトデッシュ、スキャナなども含まれる。これら及びそれ以外の入力装置は、多くの場合、システムバス１２１に結合されているユーザ入力インタフェースを通じて演算処理装置１２０に接続されるが、パラレルポート、ゲームポート又はユニバーサルシリアルバス（ＵＳＢ）などそれ以外のインタフェース及びバス構造によって接続されることもある。

【００２５】

モニタ１９１又は他のタイプの表示装置も、ビデオインタフェース１９０などのインタフェース経由でシステムバス１２１に接続される。ノースブリッジなどのグラフィックインタフェース１８２がシステムバス１２１に接続されることもある。ノースブリッジとは、ＣＰＵ（すなわちホスト処理ユニット１２０）と通信し、アクセラレイテッドグラフィックポート（ＡＧＰ）通信に関して責任を負うチップセットである。１又は複数のグラフィック処理装置（ＧＰＵ）１８４が、グラフィックインタフェース１８２と通信することがある。この点では、ＧＰＵ１８４は、レジスタ記憶装置のようなオンチップ記憶装置を含み、ビデオメモリ１８６と通信するのが一般的である。しかし、ＧＰＵ１８４は、コプロセサの一例に過ぎず、様々なコプロセシング装置がコンピュータ１１０に含まれていることがある。また、モニタ１９１又は他のタイプの表示装置が、ビデオインタフェース１９０などのインタフェース経由でシステムバス１２１に接続され、ビデオメモリ１８６と通信する。モニタ１９１に加えて、コンピュータには、更に、スピーカ１９７及びプリンタ１９６などの他の周辺出力装置が含まれることがあり、これらは出力周辺インタフェース１９５によって接続される。

【００２６】

コンピュータ１１０は、リモートコンピュータ１８０などの１又は複数のリモートコンピュータへの論理結合を用いて、ネットワーク接続された環境で動作することがある。リモートコンピュータ１８０とは、パーソナルコンピュータ、サーバ、ルータ、ネットワークＰＣ、ピアデバイス又はそれ以外の共通ネットワークノードであり、図１では記憶装置１８１だけが図解されているが、コンピュータ１１０に関して上述した中の多く又はすべての要素を含む。図１に描かれた論理結合は、ローカルエリアネットワーク（ＬＡＮ）１７１及び広域ネットワーク（ＷＡＮ）１７３を含むが、他のネットワークを含むこともある。そのようなネットワーキング環境は、オフィス、企業全体に渡るコンピュータネットワーク、イントラネット及びインターネットなどにおいて一般的である。

【００２７】

ＬＡＮネットワーキング環境の中で用いられるときには、コンピュータ１１０は、ネットワークインタフェースまたはアダプタ１７０を介してＬＡＮ１７１に接続される。ＷＡＮネットワーキング環境の中で用いられるときには、コンピュータ１１０は、典型的には、モデム１７２や、インターネットなどＷＡＮ１７３経由での通信を確立する他の手段を含む。モデム１７２は、内蔵も外付けもありうるが、ユーザ入力インタフェース１６０経由で、又は、他の適切な機構を経由して、システムバス１２１に接続される。ネットワーク接続された環境では、コンピュータ１１０又はその一部との関係で示されているプログラムモジュールは、遠隔的な記憶装置に格納されることがある。例示であり限定ではないが、図１は、メモリ・デバイス１８１に常駐するものとしてリモート・アプリケーション・プログラム１８５を図解している。示されているネットワーク接続は、コンピュータ間で用いられる例示的な通信リンクとして図解されている。

【００２８】

当業者であれば、コンピュータ１１０又は他のクライアントデバイスがコンピュータネットワークの一部として用いられることを認識するはずである。この点で、本発明は、任意の数のメモリ又は記憶装置、任意の数の記憶装置の全体で生じる任意の数のアプリケーション又は処理を有する任意のコンピュータシステムに関するものである。本発明は、ネットワーク環境中で用いられリモート又はローカルなメモリを有するサーバコンピュータとクライアントコンピュータとを備えた環境に適用が可能である。本発明は、更に、プログラミング言語機能、解釈及び実行能力を有するスタンドアロンの計算機にも適用可能である。

【００２９】

分散コンピューティングは、計算機とシステムとの間の直接的な交換により、コンピュータ・リソース及びサービスの共有を促進する。これらのリソース及びサービスには、情報の交換、キャッシュ記憶装置、そして、ファイルのためのディスク記憶装置が含まれる。分散コンピューティングは、ネットワーク接続の効果を利用することにより、クライアントが集合的な能力をてこ（レバレッジ）として用いて企業全体に役立てることを可能にする。この点に関しては、様々なデバイスがアプリケーション、オブジェクト又はリソースを有し、これらの間の相互作用により、信頼できるグラフィックスパイプラインのために本発明の認証技術を用いることができる。

【００３０】

図２では、ネットワークで接続された又は分散型の典型的なコンピュータ環境の概略図を提供されている。分散型の計算環境は、計算オブジェクト１０ａ、１０ｂ、計算オブジェクト又はデバイス１１０ａ、１１０ｂ、１１０ｃなどを含む。これらのオブジェクトは、プログラム、メソッド、データ・ストア、プログラマブルロジックなどを含む。これらのオブジェクトは、ＰＤＡ、テレビ、ＭＰ３プレーヤ、パーソナルコンピュータなどの同じ又は異なるデバイスの部分で構成される。それぞれのオブジェクトは、通信ネットワーク１４を経由して別のオブジェクトと通信できる。このネットワークは、それ自体、図２のシステムにサービスを提供する他の計算オブジェクト及び計算機を含む。本発明のある側面によると、それぞれのオブジェクト１０又は１１０は、信頼できるグラフィックスパイプラインのために本発明の認証テクニックを必要とするアプリケーションを含む。

【００３１】

１１０ｃなどのオブジェクトは、別の計算機１０又は１１０上においてホストされることもある。したがって、示されている物理環境は接続されているデバイスをコンピュータとして示しているが、そのような図解は単なる例示である。よって、この物理環境を、ＰＤＡ、テレビ、ＭＰ３プレーヤなどの様々なディジタル・デバイス、インタフェースなどのソフトウェア・オブジェクト、ＣＯＭオブジェクトなどを含むようにも図解してもかまわない。

【００３２】

分散計算環境をサポートするシステム、コンポーネント及びネットワーク・コンフィギュレーションには様々なものがある。例えば、複数の計算システムを、ローカル・ネット

ワーク又は広域分散ネットワークを経由し、有線又は無線によって、接続することがある。現在では、ネットワークの多くはインターネットに結合されているので、インターネットが、広域的に分散された計算のためのインフラストラクチャを提供し、多くの異なるネットワークを包含する。

【００３３】

ホームネットワーキング環境では、少なくとも４つの別個のネットワーク転送媒体が存在していて、それぞれが、電力線、データ（無線及び有線の両方）、ボイス（例えば、電話）及びエンタテイメント媒体などユニークなプロトコルをサポートしている。光スイッチ及び器具などほとんどの家庭用制御機器は、電力線を接続に使用している。データサービスは、ブロードバンド（例えばＤＳＬ又はケーブル・モデムのいずれか）として家庭に入り、無線（例えば、ホームＲＦ又は８０２．１１ｂ）又は有線（例えば、ホームＰＮＡ、Ｃａｔ５、更には電力線も）接続のいずれかを用いて家庭内でアクセス可能である。音声トラフィックは、有線（例えば、Ｃａｔ３）又は無線（例えば携帯電話）として家庭に入り、家庭の中ではＣａｔ３ワイアリングで分配される。エンタテイメント媒体は、サテライトかケーブルのいずれかを介して家庭に入り、典型的には同軸ケーブルを用いて家庭内で分配される。メディアデバイスのクラスターのためのディジタル相互接続としては、ＩＥＥＥ１３９４及びＤＶＩが普及している。プロトコル標準として普及しているこれらのネットワーク環境及び他のものは、すべて、インターネット経由で外部の世界に接続されるイントラネットを形成するように相互に接続できる。要するに、様々な異なるソースが、データの記憶及び伝送のために存在する。その結果、計算機は、データ処理パイプラインのすべての部分において、コンテンツを保護する複数の方法を必要とする。

【００３４】

「インターネット」とは、一般的に、ＴＣＰ／ＩＰプロトコルを利用するネットワーク及びゲートウエイの集合を指し、これは、コンピュータ・ネットワーキング技術で周知である。ＴＣＰ／ＩＰは「トランスミッション・コントロール・プロトコル／インターネット・プロトコル」の頭文字である。インターネットは、ユーザが互いに対話しネットワークを介して情報を共有することを可能にするコンピュータ処理ネットワークプロトコルによって相互に接続された地理的に分散するリモートコンピュータ・ネットワークのシステム、と説明できる。広範囲に情報を共有するために、インターネットなどの遠隔ネットワークは、オープンなシステムとしてこれまで成長してきた。オープンシステムでは、開発者たちは、ほぼ制限をうけることなく特定の動作又はサービスを実行するソフトウェアアプリケーションを設計することができる。

【００３５】

したがって、以上のネットワークインフラストラクチャにより、クライアント／サーバ、ピアツーピア又はハイブリッドアーキテクチャなどネットワークトポロジのホストが可能になる。「クライアント」とは、それが関係していない別のクラス又はグループのサービスを利用するクラス又はグループのメンバである。よって、コンピューティングでは、クライアントとはプロセスであり、つまり、別のプログラムによって提供されるサービスを要求する命令又はタスクの集合である。クライアントプロセスは、別のプログラム又はサービス自体に関する作業上の詳細をまったく「知る」ことを必要とせずに、要求されたサービスを利用する。クライアント／サーバ・アーキテクチャ、特にネットワーク化されたシステムでは、クライアントは、通常、例えばサーバなど別のコンピュータによって提供され共有しているネットワークリソースにアクセスするコンピュータである。図２の例では、コンピュータ１１０ａ、１１０ｂなどはクライアントと考えることができ、コンピュータ１０ａ、１０ｂなどはサーバと考えることができる。なお、サーバ１０ａ、１０ｂなどは、クライアントコンピュータ１１０ａ、１１０ｂなどにおいて複製されるデータを維持管理する。

【００３６】

サーバは、典型的には、インターネットなどの遠隔ネットワークを経由してアクセス可能な遠隔コンピュータシステムである。クライアントプロセスは、第１のコンピュータシ

ステムにおいてアクティブであり、サーバプロセスは第２のコンピュータシステムにおいてアクティブであって、通信媒体経由で相互に通信し、分散型の機能を提供し、複数のクライアントがサーバの情報収集能力を利用することを可能にしている。

【００３７】

クライアントとサーバとは、プロトコル層によって提供される機能を利用して、相互に通信する。例えば、ハイパーテキスト・トランスファ・プロトコル（ＨＴＴＰ）は、ワールド・ワイド・ウェブ（ＷＷＷ）と共に使用される一般的なプロトコルである。典型的には、ユニバーサル・リソース・ロケータ（ＵＲＬ）又はインターネット・プロトコル（ＩＰ）アドレスなどのコンピュータネットワーク・アドレスは、サーバ又はクライアントコンピュータを相互に識別するのに用いられる。ネットワークアドレスは、ユニバーサル・リソース・ロケータ・アドレスとも呼ばれる。例えば、通信は、通信媒体を経由して提供することができる。特に、クライアントとサーバとは、大容量通信のためのＴＣＰ／ＩＰ接続を経由して、相互に接続可能である。

【００３８】

このように、図２には、本発明が実装されるネットワーク接続された又は分散型の環境が図解されていて、ネットワーク／バスを介してサーバとクライアントとが通信する。より詳細には、多くのサーバ１０ａ、１０ｂなどが、通信ネットワーク／バス１４を介して相互接続されている。ここで、通信ネットワーク／バス１４とは、ＬＡＮ、ＷＡＮ、イントラネット、インターネットなどである。そして、ポータブルコンピュータ、ハンドヘルドコンピュータ、シンクライアント、ネットワーク接続された器具、又は、ＶＣＲ、ＴＶ、オーブン、照明、ヒータなどなどの他のデバイスを含む多数のクライアント又は遠隔計算機１１０ａ、１１０ｂ、１１０ｃ、１１０ｄ、１１０ｅなどが本発明に従って相互に連結されている。したがって、本発明は、任意の計算機に適用可能であり、信頼できるソースからの安全なコンテンツを処理、記憶又はレンダリングする、更には、仮想マシンによって生成された高性能グラフィックスのレンダリングに用いることができる。

【００３９】

通信ネットワーク／バス１４が例えばインターネットであるネットワーク環境では、サーバ１０はウェブサーバであり、その場合、クライアント１１０ａ、１１０ｂ、１１０ｃ、１１０ｄ、１１０ｅなどは、ＨＴＴＰのような既知の多くのプロトコルの中の任意のプロトコルを介してウェブサーバと通信する。分散型の計算環境の特性として、サーバ１０は、クライアント１１０として機能することもある。通信は、適性に応じて、有線又は無線でありうる。複数のクライアントデバイス１１０は、通信ネットワーク／バス１４を介して通信する場合と通信しない場合とがあり、独立の通信が関連付けられていることもある。例えばテレビやＶＣＲの場合、その制御に関して、ネットワーク接続された特徴が存在する場合としない場合とがある。クライアントコンピュータ１１０とサーバコンピュータ１０とは、それぞれ、様々なアプリケーションプログラムモジュール又はオブジェクト１３５を備えているし、また、ファイルがそこに記憶される又はファイルがそこにダウンロード若しくは移動される様々なタイプの記憶素子又はオブジェクトへの接続又はアクセスを備えている。したがって、本発明は、コンピュータネットワーク／バス１４にアクセスして対話するクライアントコンピュータ１１０ａ及び１１０ｂと、クライアントコンピュータ１１０ａ及び１１０ｂやそれ以外のデバイス１１１やデータベース２０と対話するサーバコンピュータ１０ａ、１０ｂとを有するコンピュータネットワークにおいて、用いることができる。

分割（パーティション）された計算機：

ここで、図３を参照すると、計算機１０は、この計算機１０にリソースサービスを提供するのに用いられる特定のリソース１２へのアクセスを含む又は有する。そのような計算機１０、リソース１２及びリソースサービスは、本発明の精神及び範囲から逸脱することなく、任意の適切な計算機、リソース及びリソースサービスでありうる。例えば、計算機１０は、データ記憶サービスを提供するしソース１２としてハードディスクドライブを備えた、デスクトップ又はラップトップコンピュータなどのパーソナルコンピュータであり

うる。同様に、計算機１０は、表示サービスを提供するリソース１２としてディスプレイスクリーンを備えたポータブル型のオーディオ又はビデオプレーヤなどのポータブル再生装置でありうる。また、計算機１０は、通信サービスを提供するリソース１２としてデータ通信ネットワークを備えたサーバマシンでありうる。更にまた、サーバマシンは、それ自体がリソース１２にもなりうる。リソース１２は、特定のハードウェア、特定のハードウェアにアクセスするためのアダプタ、リモートサービス、ローカルサービス、及びこれらの組合せなどでもありうることに注意してほしい。

【００４０】

重要であるのは、計算機１０は、複数のパーティションを実行するように構成され、それぞれのパーティションは、仮想マシンをインスタンス化（instantiate）し、更に、１又は複数のアプリケーションがインスタンス化されるオペレーティングシステムのインスタンスをホストすることである。示されているように、そのような計算機１０では、当該計算機１０の特定のリソース１２は特定のパーティション又は仮想マシン１４（以下では、ＶＭ１４と称する）に割り当てられており、それにより、この特定のＶＭ１４は特定のリソース１２を直接に制御することができる。よって、この特定のＶＭ１４は、リソース能力を提供するホスト（ＶＭ－Ｈ１４）である。同様に、このようなＶＭ－Ｈ１４は１又は複数の他のＶＭ１６にリソースサービスを提供し、ＶＭ１６はそれぞれがそのようなサービスを消費するクライアント（ＶＭ－Ｃ１６）である。典型的には、それぞれのＶＭ－Ｃ１６とＶＭ－Ｈ１４とは、仮想マシン（ＶＭ）バス１８などのパイプ又はチャネル経由で通信し、リソースに関連する動作を実行する。

【００４１】

ＶＭバス１８は、本発明の精神及び範囲から逸脱することなく、計算機１０においてオブジェクトとしてそれ自体で確立されるか、又は、それ自体では存在しない概念的なオブジェクトとして確立される。後者の場合、そのようなＶＭ１４及び１６が相互間で通信を確立することを選択すると、概念的なＶＭバス１８は、ＶＭ１４とＶＭ１６との間の通信経路として現れることに注意すべきである。特に、そのようなＶＭバス１８の発生は、ＶＭ１４及び１６が相互間で通信することを選択しＶＭ１４のそれぞれが当該通信のために必要なサービスを確立し実際にそのようなサービスを用いて通信を行う際に、生じると考えられる。そのような場合、この通信は、本発明の精神及び範囲から逸脱することなく、計算機１０の中の任意の適切な通信手段を経由して生じる。

【００４２】

図３の計算機１０とＶＭ１４及び１６とは、ユーザモードとカーネルモードとの両方を含むように機能的に動作するが、これらのモジュールは、本発明の目的にとって絶対的に必要であるとは考えられない。いずれにしても、そして、理解されると思われるが、ユーザモードは一般的に非特権な状態（non-privileged state）であり、そのモードでは、実行コードは例えばそのコード自体に割り当てられていないメモリへの書込みなど特定の動作の実行がハードウェアによって禁止されている。一般に、それらの禁止されている動作は、ＶＭ１４及び１６のオペレーティングシステムを不安定にするか、又は、セキュリティ上のリスクを構成する動作である。オペレーティングシステムの点では、ユーザモードは類似の非特権的な実行モードであり、そのモードでは、実行コードは、システムコンフィギュレーションファイルへの書込み、他のプロセスを停止させること、システムのリブートなど潜在的に危険な動作の実行をカーネルによって禁止される。

【００４３】

更に注意すべきであるが、カーネルモードすなわち特権的なモードは、オペレーティングシステムと関連するコアコンポーネントとが動作するモードである。カーネルモードで動作するコードは、ＶＭ／パーティション１４及び１６に割り当てられているシステムメモリとリソースとに対する無制限のアクセスを有する。一般に、カーネルモードで動作するコードの量は、セキュリティ及びエレガンスさとの両方の目的のために、最小化される。概略的に述べると、計算機１０のユーザは、ユーザモードとそこで動作しているアプリケーションとを介して最も直接的にインタフェースし、他方で、計算機１０はカーネルモ

ードを介して最も直接的に特定のリソース１２を含むリソースとインタフェースする。

ハードウェアリソース１２を有するＶＭ－Ｈ１４の移動：

上に指摘したように、ＶＭ１４及び１６は、仮想的な構成物として、随意に停止、保存、検索、再開が可能である。その結果、第１のプラットホームの上に保存されているものとしてのＶＭ１４及び１６は、第２のプラットホームに、動かす、又は、「移動」させることが可能である。ここでプラットホームとは、異なる計算機又は同一の計算機の異なる構成を表わす。したがって、例えば、ウェブサーバであるＶＭ１４及び１６は、当該ウェブサーバのどのクライアントも当該ウェブサーバが移動されたことを知らないままに、第１の物理マシン上での動作を停止させ、移動させて、第２の物理マシンの上で再始動させることが可能である。そして、そのような移動により、第１の物理マシンを、その上で行われている作業を中断することなく、メンテナンスのため又は再コンフィギュレーションのために、ラインから取り外すことが可能である。更に、そのような移動により、物理マシンのグループが動的に負荷のバランスを取ることを可能になる。同様に、ユーザの個人的なワークスペースを表わすＶＭ１４及び１６を、当該ユーザが、職場や家庭などにおける複数の計算機１２の間で移動させることが可能になる。

【００４４】

しかし、ＶＭ１４及び１６に関する状態情報のすべてが当該ＶＭ１４及び１６のソフトウェア構成の内部に含まれているとは限らないことに注意しなければならない。特に、ハードウェア又はそのアダプタであるリソース１２を有するＶＭ－Ｈ１４は、リソース１２と関係のある特定の状態情報を当該リソース１２と共に記憶している場合がある。特にＶＭ－Ｈ１４の状態情報の一部がＶＭ－Ｈ１４自体のソフトウェア構成に含まれていない状況では、ＶＭ－Ｈ１４を第１のプラットホームから第２のプラットホームへＶＭ－Ｈ１４を移動させることはより困難となる。特に、リソース１２におけるＶＭ－Ｈ１４の状態情報が、リソース１２における当該状態情報が失われずに又はそうでなければＶＭ－Ｈ１４から恒久的に分離されるように処理されるまで、そのような移動は生じない。

【００４５】

ハードウェアリソース１２の使用への割込みをＶＭ－Ｈ１４が許容できるシナリオでは、リソース１２の状態情報は性質が比較的良性であり、移動のためにＶＭ－Ｈ１４を停止する前に、この状態情報に働きかけてリソース１２から削除することができる可能性が高い。例えば、リソース１２がプリンタであり状態情報が印刷ジョブに関係する場合には、印刷ジョブは、状態情報がプリンタによって消費されるように完了することが許容され、その後で、プリンタリソース１２の所有権はＶＭ－Ｈ１４から剥奪され、ＶＭ－Ｈ１４は移動されることになろう。

【００４６】

ハードウェアリソース１２の使用への割込みをＶＭ－Ｈ１４が許容できる別のシナリオでは、リソース１２の状態情報は性質がいくぶん低下しているが、移動のためにＶＭ－Ｈ１４を停止する前に、ＶＭ－Ｈ１４に移動することができる可能性が高い。例えば、リソース１２が先の例と同じプリンタであり状態情報が先の例と同じ印刷ジョブに関係しているが、当該印刷ジョブを合理的な時間内に完了することができない場合には、ＶＭ－Ｈ１４を移動のために停止する前に、印刷ジョブを停止させることができ、印刷ジョブに関係する残りの状態情報をＶＭ－Ｈに移動させることができ、その後で、プリンタリソース１２の所有権を再びＶＭ－Ｈ１４から剥奪し、ＶＭ－Ｈ１４を移動させることができる。そして、移動の後で、ＶＭ－Ｈ１４は再び同じプリンタリソース１２を所有すると想定して、印刷ジョブに関係する残りの状態情報をＶＭ－Ｈ１４からプリンタに移動して印刷ジョブを完了することができる。

【００４７】

しかし、これは、前述した仮定をすることができない場合でありえる。したがって、移動の後にＶＭ－Ｈ１４によって所有されるプリンタ資源１２は、別のタイプのプリンタなどまったく異なるプリンタ資源１２であるか、又は、同一のプリンタではあるが印刷コントローラが更新されているような僅かに異なるプリンタ資源１２でありうる。重要なこと

であるが、いずれの場合にも、異なるプリンタ資源がもはや状態情報を認識できない又は作用できない限り、印刷ジョブに関係する残りの状態情報をＶＭ－Ｈ１４からプリンタ自体に移動して印刷ジョブを完了することができない可能性がある。

【００４８】

また、中止すべきであるが、リソース１２を有するＶＭ－Ｈ１４は、ジョブリストや内部的ステートマシン変数などを含む状態情報を当該リソース自体において含むことがある。ＶＭ－Ｈ１４が任意に停止され再始動されれば、リソース１２の状態情報は異なる可能性があり、未解決のリクエストが最良の場合でも放棄され、その結果としてＶＭ－Ｈ１４はクラッシュする可能性がある。最悪の場合には、リソース１２のためのドライバなどは、状況を誤解し、プログラミングの際にリソース１２がリソース１２自体の中のメモリを損ない、やはり結果的にＶＭ－Ｈ１４をクラッシュさせ、おそらくは、計算機１０における他のＶＭ１４及び１６もまたクラッシュさせる可能性がある。

【００４９】

このように、ＶＭ－Ｈ１４が移動の後に同じ計算機１０の上で再始動され同じリソースが利用可能であることを保証できる場合には、ＶＭ－Ｈ１４のオペレーティングシステムにリソース１２からすべての状態情報を移動させ、移動させる前にＶＭ－Ｈ１４と共に記憶させることで十分である。ＶＭ－Ｈ１４が移動の後に再始動される場合、ＶＭ－Ｈ１４のオペレーティングシステムは、記憶されている状態情報を更なる作用のためにリソース１２に戻すことができる。

【００５０】

しかし、そのような保証ができない場合、又は、作業負荷のためすべての状態情報を移動の前に動かすことができない場合、移動の前にリソース１２をＶＭ－Ｈ１４による所有から削除する又は「取り外す」ことで十分である場合が多いと思われる。そのような取り外しは、ＶＭ－Ｈ１４のオペレーティングシステムに対する適切なリクエストをすることによって達成することができる。本発明のある１つの実施例では、そのような取り外しは、移動させるためにＶＭ－Ｈ１４を停止させる前に命令され、それにより、ＶＭ－Ｈ１４のオペレーティングシステムは、重要であると考えられリソース１２におけるすべての状態情報よりも少ないと想定される状態情報だけをリソース１２からＶＭ－Ｈ１４に移動させる。取り外しの後では、リソース１２は、ＶＭ－Ｈ１４に関する状態情報を有しておらず、移動の前にも後にもＶＭ－Ｈ１４によってもはや所有されておらず、又は、ＶＭ－Ｈ１４に対して利用可能でない。このように、移動の後では、ＶＭ－Ｈ１４は、どのような状態情報についても再度リソース１２に移動させようと試みることはなく、したがって、そのような移動に起因する可能性があるすべての問題が回避される。もちろん、ＶＭ－Ｈ１４は、利用可能でありそのように望むのであれば、リソース１２の所有権を再取得しなければならない。

【００５１】

しかし、更に別のシナリオでは、ＶＭ－Ｈ１４は、ハードウェアリソース１２の使用への割込みを許容することができず、したがって、リソース１２の状態情報はその性質から比較的危機的状態にある。例えば、リソース１２がディスクドライブなどＶＭ－Ｈ１４によって用いられる主記憶装置である場合、それへのアクセスはＶＭ－Ｈ１４を左右するほど重大であり、さもないと、ＶＭ－Ｈ１４のオペレーティングシステムがクラッシュする可能性がある。したがって、そのようなシナリオでは、ＶＭ－Ｈ１４が状態情報をそのようなリソース１２に向けることができなければならない限り、ディスクドライブリソース１２の所有権をＶＭ－Ｈ１４から剥奪する時間は存在しない。よって、このようなシナリオでは、何からの機構を用いて、ＶＭ－Ｈ１４が状態情報をリソース１２に向けることを許容しながら、リソース１２の動作を停止させ、ＶＭ－Ｈ１４のすべての状態情報をリソース１２から取り除かなければならない。

【００５２】

次に図４を参照すると、典型的なＶＭ－Ｈ１４が、記憶装置スタック２２を経由して記憶装置リソース１２などのリソース１２にアクセスする様子が示されている。ここで、記

憶装置スタック２２は、ファイルシステムドライバ、パーティションドライバ、ボリュームドライバ、ディスクドライバなどを含む。もちろん、本発明は、その精神及び範囲を逸脱することなく、記憶装置リソース１２には限定されず、任意のタイプのリソース１２でありうるのであって、適切な対応するスタック２２などをアクセスに用いることができる。

【００５３】

示されているように、図４の記憶装置スタック２２は、ポートドライバ２４など（以下では、「ポート２４」又は等価物）を経由して記憶装置リソース１２と通信する。注意すべきであり、典型的であるのだが、ポート２４は、記憶装置スタック２２からの総括的で理想化されたコマンド又はリクエストを記憶装置リソース１２に固有のコマンド又はリクエストに変換する。例えば、記憶装置リソース１２がＶＭ－Ｈ１４のオペレーティングシステムが存在するメモリの一部を含む場合には、ページングリクエストなどを受け取るために、記憶装置リソース１２は連続的に動作しなければならない。さもないと、ＶＭ－Ｈ１４のオペレーティングシステムはクラッシュする。端的に言えば、このような場合には、少なくとも、ＶＭ－Ｈ１４がアクセスリクエストなどの形式を有する状態情報を記憶装置リソースに常に送ることができなければならない限りにおいて、記憶装置リソース１２へのアクセスを中断することはできない。

【００５４】

したがって、次に図５を参照すると、本発明のある実施例において、図４の単一のポート２４が１対のポート２４ａ及び２４ｂに置き換えられている。ポート２４ａは記憶装置スタック２２を記憶装置リソース１２と通信可能に結合し、ポート２４ｂは記憶装置スタック２２を通信媒体経由で別の目的地と通信可能に結合している。図示されているように、ここで、通信媒体とは、ＶＭバス１８である。別の目的地とは、計算機１０において動作している別のＶＭ１４、１６である。しかし、別の目的地も通信媒体も、任意の適切な別の目的地及び通信媒体でもよく、その場合でも本発明の精神及び範囲から逸脱しない。

【００５５】

本発明の実施例では、図５に示されているように、ポート２４ａ及び、２４ｂのそれぞれは、ポートリダイレクタ２６を経由して記憶装置スタック２２に結合される。ポートリダイレクタ２６は、記憶装置リソース１２へのアクセスリクエストを、ポート２４ａを経由して記憶装置リソース１２に方向付けるか、又は、ポート２４ｂ及び通信媒体／ＶＭバス１８を経由して別の目的地に方向付ける。重要な点は、ポートリダイレクタ２６が、それぞれのアクセスリクエストを以下で説明される態様でいかにして方向付けるかということである。

【００５６】

次に図６を参照すると、本発明の実施例において、記憶装置リソース１２などのリソース１２を有するＶＭ－Ｈ１４を移動するのに用いられる一連の動作が示されている。これら一連の動作は、ＶＭ－Ｈ１４を必ずしも移動させず保存するのにも用いられることに注意すべきである。移動でも保存でも、ＶＭ－Ｈ１４の通常の実行時間の動作の間、記憶装置リソース１２はＶＭ－Ｈ１４によって所有され、ポートリダイレクタ２６は、ＶＭ－Ｈ１４の記憶装置スタック２２からのアクセスリクエストなどを、ポート２４ａを経由して記憶装置リソース１２に方向付け、ポート２４ａと記憶装置リソース１２とは、そのようなアクセスリクエストなどをキュー・アップし、処理する（ステップ６０１）。しかし、いずれかの適切なソースによってＶＭ－Ｈ１４の保存又は移動が行われた後で（ステップ６０３）、ポートリダイレクタ２６は、ＶＭ－Ｈ１４の記憶装置スタック２２からのアクセスリクエストなどをポート２４ｂに方向付ける（ステップ６０５）。

【００５７】

まず、ポート２４ｂは受け取ったアクセスリクエストなどをキューし、他方で、ポート２４ａと記憶装置リソース１２とは任意の残りのアクセスリクエストなどを処理する（ステップ６０７）。こうして、ポート２４ａと記憶装置リソース１２とにおけるすべてのアクセスリクエストなどは完了することが可能になり、その後では、関係する状態情報が記

憶装置リソース１２に残ることはない（ステップ６０９）。ポート２４ａと記憶装置リソース１２とにおけるアクセスリクエストなどは完了することが可能になるが、更なるアクセスリクエストなどはポート２４ｂにおいてキューされ、そのようなアクセスリクエストなどが、否定され、無視され、又は拒絶されて、ＶＭ－Ｈ１４のオペレーティングシステムをクラッシュさせることはない。

【００５８】

いったんポート２４ａと記憶装置リソースとにおけるすべてのアクセスリクエストなどと記憶装置リソース１２とが完了し、関係する状態情報が記憶装置リソース１２に残らないようになると、記憶装置リソース１２は（第１の）ＶＭ－Ｈ１４から取り除かれ、おそらくはイジェクトなどによって、（第１の）ＶＭ－Ｈ１４は、もはや記憶装置リソース１２を有しなくなる（ステップ６１１）。記憶装置リソース１２は、次に、別のＶＭ１４、１６に割り当てられる。別のＶＭ１４、１６とは、記憶装置リソース１２を有する第２のＶＭ－Ｈ１４である。このような除去及び割り当ての実行は、本発明の精神及び範囲から逸脱することなく、任意の適切な方法によって、任意の適切な手段を用いて行うことができる。

【００５９】

本発明のある実施例では、いったん記憶装置リソース１２が第２のＶＭ－Ｈ１４に割り当てられると、ポート２４ｂは、ＶＭバス１８などを介して第２のＶＭ－Ｈ１４に結合され（ステップ６１５）、ポート２４ｂにおいてキューされているアクセスリクエストなどは、現在所有している記憶装置リソース１２において実行されるように、第２のＶＭ－Ｈ１４に送られる（ステップ６１７）。ある時点で、必要なアクセスリクエストなどはすべてが、第１のＶＭ－Ｈ１４の記憶装置スタック１２によって、記憶装置リソース１２に送られる。これは、保存又は移動の際に、ポート２４ａ又はポート２４ｂを経由してなされる。記憶装置リソース１２は、そのようにして送られたアクセスリクエストなどをすべて処理する（ステップ６１９）。このようにして、第１のＶＭ－Ｈ１４の保存又は移動は完了する。ここで、記憶装置リソース１２に対するＶＭ－Ｈ１４からのアクセスリクエストは、すべて、ポート２４ａによって直接的に又はポート２４ｂ及び第２のＶＭ－Ｈ１４によって間接的に作用を受けることがわかっている。また、記憶装置リソース１２が、第１のＶＭ－Ｈ１４の状態情報をまったく維持することなく第１のＶＭ－Ｈ１４から削除されることもわかっている（ステップ６２１）。

【００６０】

移動の後で記憶装置リソース１２を第１のＶＭ－Ｈ１４に再度割り当てることは、ステップ６０３－６２１において実行された削除の逆である。したがって、そのような再割り当ての詳細をここで述べる必要はないと思われる。

【００６１】

第１のＶＭ－Ｈ１４のポートの２４ｂは、受け取った一般的なコマンドを特定のコマンドに変換すべきでない。したがって、第２のＶＭ－Ｈ１４の記憶装置スタック２２は、第１のＶＭ－Ｈ１４の記憶装置スタック２２よりも機能的に上にある必要はない。特に、スタック２２は、共に、一般的なコマンドを生じ、第２のＶＭ－Ｈ１４のポート２４において受け取られたそのような一般的なコマンドは、特定のものでありうる。

結論

本発明は、少なくとも部分的にはホスト及びクライアントＶＭ１４、１６の観点から説明されているが、ハードウェアを有するパーティション又はＶＭが移動されるような任意の状況に関するものであることに注意すべきである。移動するＶＭは、他のクライアントに対するホストである場合が多いが、常に他のクライアントに対するホストであるとは限らない。また、移動の際には、ＶＭ－Ｈ１４は、ハードウェアデバイスの所有権を実際放棄するときには、少なくともこの出願において用いられる用語によれば、ＶＭ－Ｃ１６になる。本発明は、ホスト及びクライアントＶＭ１４及び１６のコンテキストで開示されているものの、「ホスト」又は「クライアント」などの用語に固執することなく、すなわち、ＶＭを有するハードウェアが移動され、移動の前に当該ハードウェアの所有権を別のＶ

Ｍに与えるＶＭという観点から考察されるべきである。

【００６２】

本発明との関係で実行されるプロセスを実現するのに必要なプログラミングは、比較的単純なものであり、関係するプログラミング業界にとっては明白であるはずである。特に、図５に示されているオブジェクトのそれぞれを構成するのに必要なプログラミングや、図６のステップを実現するのに必要なプログラミングは、オブジェクト及びステップのそれぞれに必要な機能に基づけば、明白であるはずである。したがって、それらのプログラミングは、ここに添付されていない。本発明を実現するには、本発明の精神及び範囲から離れることなく、任意の特定のプログラミングを用いることが可能である。

【００６３】

本発明では、ＶＭ－Ｈ１４が保存される又は第１のプラットホームから第２のプラットホームへ移動されるときに、有しているリソース１２においてＶＭ－Ｈ１４の状態情報を処理するための方法及び機構が提供される。リソース１２における状態情報は、移動の前に、通常の動作においてリソース１２から削除することができる。あるいは、移動の後でＶＭ－Ｈ１４が事後的に検索できるように記憶することもできる。更にまた、別のＶＭ－Ｈ１４による処理も可能である。

【００６４】

上述した複数の実施例に対し、その発明的概念から離れることなく、変更を行うことが可能である。一例を挙げると、本発明はＶＭ－Ｈ１４の状態情報を備えたハードウェアリソース１２の観点から説明されているが、リソース１２は、本発明の精神及び範囲から離れることなく、ソフトウェアリソース１２など、ＶＭ－Ｈ１４の状態情報を備えた別のタイプのリソースでもよい。別の例として、本発明はリソース１２にアクセスするスタック２２及びポート２４の観点から説明されているが、スタック２２及びポート２４は、本発明の精神及び範囲から離れずに、スタック及びポートだけではなくリソース１２にアクセスする任意の他のアクセス機構をそれぞれが含むことが意図されている。したがって、本発明は、ここで開示された特定の実施例に限定されることはなく、特許請求の範囲に記載された本発明の精神及び範囲内にある修正を含むことが意図されている。

【図面の簡単な説明】

【００６５】

【図１】本発明が実装される例示的で制限的ではない計算環境を表すブロック図である。

【図２】本発明が実装される様々な計算機を有する例示的なネットワーク環境を表すブロック図である。

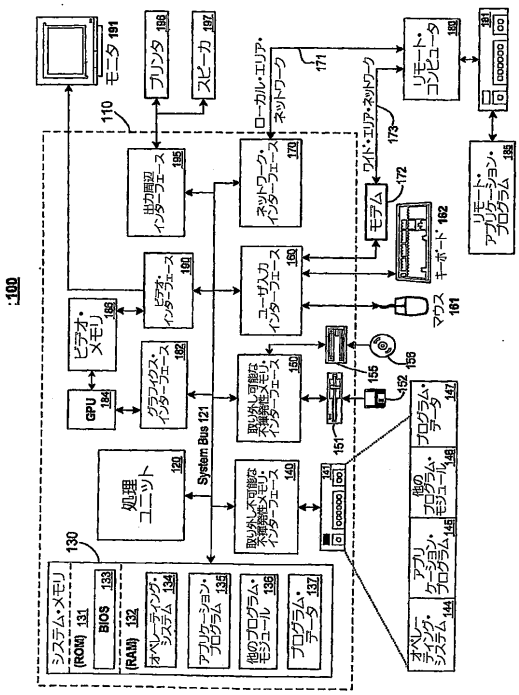
【図３】計算機を示すブロック図である。この計算機は、本発明の実施例に従って、特定のリソースを有しリソースサービスを提供する仮想マシン（ＶＭ－Ｈ）を備えたホストパーティションとＶＭ－Ｈのリソースサービスを用いる仮想マシン（ＶＭ－Ｃ）を備えたクライアントパーティションとを含む複数のパーティションを動作させる。

【図４】図３のＶＭ－Ｈを示すブロック図である。このＶＭ－Ｈが有するリソースに結合されている追加的な詳細が示されている。

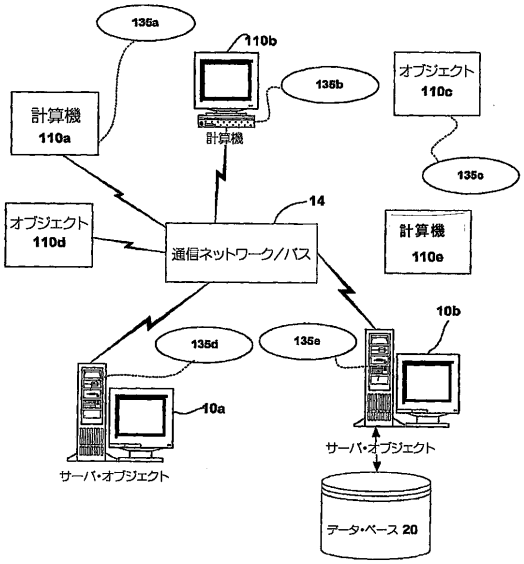
【図５】図３のＶＭ－Ｈを示すブロック図である。本発明のある実施例に従って、このＶＭ－Ｈが有するリソースに結合されている追加的な詳細が示されている。

【図６】図５のＶＭ－Ｈに関して実行されるキーとなるステップを示す流れ図である。リソースへのすべてのアクセスリクエストが本発明のある１つの実施例に従って適切に処理されることを保証しながら、ＶＭ－Ｈの保存及び移動が実行される。

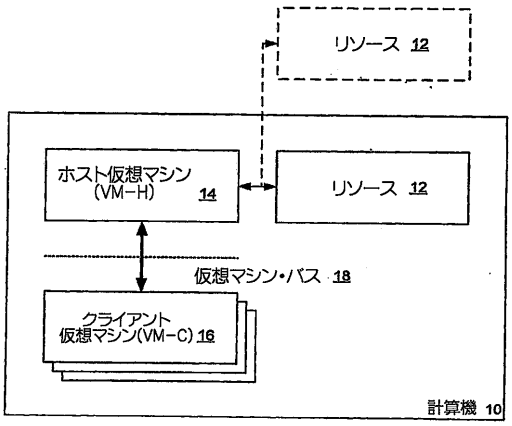
【図１】


100
110
モニタ 191
プリンタ 196
スピーカ 197
ローカル・エリア・ネットワーク 171
ワイド・エリア・ネットワーク 173
リモート・コンピュータ 193
出力周辺インターフェース 195
ネットワーク・インターフェース 170
モデム 172
リモート・アプリケーション・プログラム 185
ビデオ・インターフェース 190
ユーザ入力インターフェース 160
マウス 161
キーボード 162
ビデオ・メモリ 188
GPU 184
グラフィクス・インターフェース 182
取外し可能な不揮発メモリ・インターフェース 150
155
156
プログラム・データ 147
他のプログラム・モジュール 146
アプリケーション・プログラム 145
オペレーティング・システム 144
System Bus 121
処理ユニット 120
130
システム・メモリ
(ROM) 131
BIOS 133
(RAM) 132
オペレーティング・システム 134
アプリケーション・プログラム 135
他のプログラム・モジュール 136
プログラム・データ 137

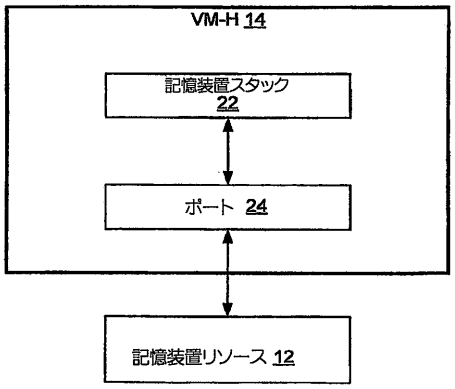
【図２】


135a
110b
135b
オブジェクト 110c
計算機 110a
計算機
135c
14
オブジェクト 110d
通信ネットワーク／バス
計算機 110e
10b
135d
135e
10a
サーバ・オブジェクト
サーバ・オブジェクト
データ・ベース 20

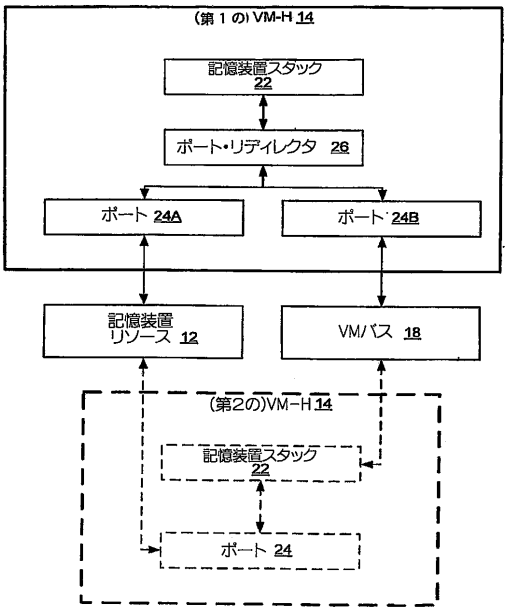
【図３】


リソース 12
ホスト仮想マシン (VM−H) 14
リソース 12
仮想マシン・バス 18
クライアント 仮想マシン(VM−C) 16
計算機 10

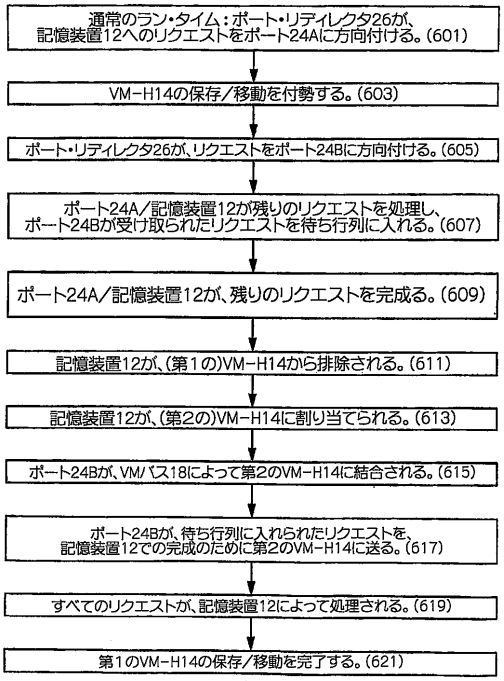
【図４】


VM-H 14
記憶装置スタック 22
ポート 24
記憶装置リソース 12

【図５】


(第１の)VM-H 14
記憶装置スタック 22
ポート・リディレクタ 26
ポート 24A
ポート 24B
記憶装置 リソース 12
VMバス 18
(第2の)VM−H 14
記憶装置スタック 22
ポート 24

【図６】


通常のラン・タイム：ポート・リディレクタ26が、記憶装置12へのリクエストをポート24Aに方向付ける。(601)
VM−H14の保存／移動を付勢する。(603)
ポート・リディレクタ26が、リクエストをポート24Bに方向付ける。(605)
ポート24A／記憶装置12が残りのリクエストを処理し、ポート24Bが受け取られたリクエストを待ち行列に入れる。(607)
ポート24A／記憶装置12が、残りのリクエストを完成する。(609)
記憶装置12が、(第１の)VM−H14から排除される。(611)
記憶装置12が、(第２の)VM−H14に割り当てられる。(613)
ポート24Bが、VMバス18によって第２のVM−H14に結合される。(615)
ポート24Bが、待ち行列に入れられたリクエストを、記憶装置12での完成のために第２のVM−H14に送る。(617)
すべてのリクエストが、記憶装置12によって処理される。(619)
第１のVM−H14の保存／移動を完了する。(621)

フロントページの続き

(72)発明者　オシンス，ジャコブ
アメリカ合衆国ワシントン州９８０５２，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

審査官　鈴木　修治

(56)参考文献　特開平１０－２８３２１０（ＪＰ，Ａ）
特開２００５－３２７２７９（ＪＰ，Ａ）
特開２００２－２１５４１６（ＪＰ，Ａ）
米国特許出願公開第２００６／０００５１８９（ＵＳ，Ａ１）
特開２００２－１８３０８８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４６－９／５４