

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-84029  
(P2008-84029A)

(43) 公開日 平成20年4月10日(2008.4.10)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 9/445 (2006.01)</b>	G06F 9/06 610A	5B176
<b>G06F 9/46 (2006.01)</b>	G06F 9/46 350	

審査請求 未請求 請求項の数 1 O L (全 9 頁)

(21) 出願番号	特願2006-263551 (P2006-263551)	(71) 出願人	000233055 日立ソフトウェアエンジニアリング株式会社 東京都品川区東品川四丁目12番7号
(22) 出願日	平成18年9月27日(2006.9.27)	(74) 代理人	100096954 弁理士 矢島 保夫
		(72) 発明者	中山 晃治 東京都品川区東品川4丁目12番7号 日立ソフトウェアエンジニアリング株式会社内
		(72) 発明者	青柳 慶光 東京都品川区東品川4丁目12番7号 日立ソフトウェアエンジニアリング株式会社内
		Fターム(参考)	5B176 AA04 AA13 AB09 AC03

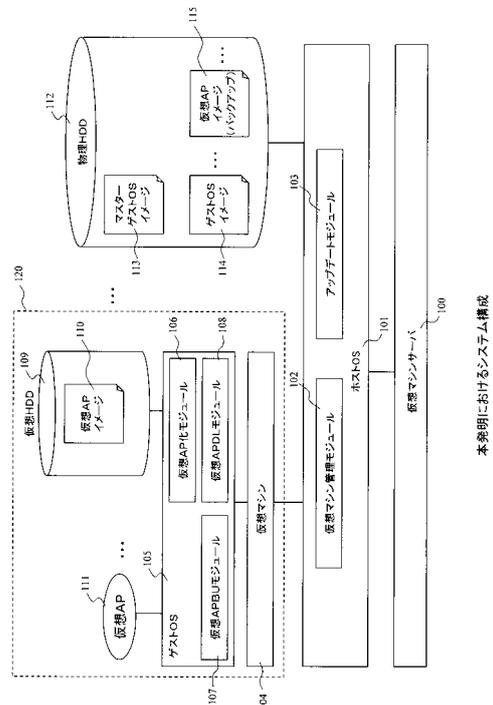
(54) 【発明の名称】 仮想マシン管理システム

(57) 【要約】

【課題】複数の仮想マシンを管理する仮想マシン管理システムにおいて、多様化するゲストOSのイメージファイルを一元化することで、ゲストOS及びゲストOS上で動作するAPの管理を容易にすること、並びにゲストOSのアップデートの際にAP環境を容易に元に戻せるようにすることを目的とする。

【解決手段】ゲストOS上で動作する複数のAPに関連するデータを1つの仮想APイメージファイルとして分離して保存し、ゲストOSのイメージファイルを一元的に管理する。ゲストOSをアップデートする場合は、一元化したゲストOSのイメージファイルを1回だけアップデートし、アップデート済みのイメージファイルをその他のゲストOSイメージファイルに上書きする。その後、アップデートが完了したゲストOS上に仮想APイメージファイルをダウンロードし、そのAPのイメージファイル上からデータを読み込んでAPを起動する。

【選択図】 図1



本発明におけるシステム構成

**【特許請求の範囲】****【請求項 1】**

複数の仮想マシンが動作する仮想マシンサーバにおいて、それら複数の仮想マシンを管理する仮想マシン管理システムであって、

前記各仮想マシン毎に、その仮想マシン上で動作するゲストOSが管理する仮想ハードディスク上に仮想A Pイメージファイルを作成する手段と、

任意の仮想マシン上で動作する任意のモジュールが、その仮想マシン上で動作するゲストOSが管理する仮想ハードディスク上の任意の領域に対して、任意のデータの書き込みまたは読み込みの処理を試みた場合に、該書き込みまたは読み込みが指示された領域から、前記仮想ハードディスク上に存在する仮想A Pイメージファイル上の所定の領域にリダイレクトを行い、書き込みまたは読み込み先を変更して書き込みまたは読み込みを実行することにより、前記任意のデータを前記仮想A Pイメージファイルに分離する手段と、

前記仮想A Pイメージファイルを、前記仮想ハードディスクから、前記仮想マシンサーバ上のホストOSが管理する物理ハードディスク上へ、バックアップする手段と、

前記物理ハードディスク上の前記仮想A Pイメージファイルを、前記仮想ハードディスク上にダウンロードする手段とを備え、

前記任意の仮想マシン上で動作するモジュールに関連するデータを前記ゲストOSのイメージファイルから分離することにより、前記ゲストOSのイメージファイルを一元的に扱うことを可能にしたことを特徴とする仮想マシン管理システム。

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、複数の仮想マシン及び仮想マシン上で使用されるOSイメージファイルの管理技術に関するものである。

**【背景技術】****【0002】**

仮想マシンは、ある特定のハードウェアをエミュレーションする環境、またはそれによって実現された仮想的なマシン環境のことであり、通常何らかのオペレーティング・システム（以後、OSと記述する）上でエミュレーションソフトを実行することにより実現される。また、仮想マシン上では、前記のOSとは異なるOSやプログラムを動作させることが可能である。このとき、仮想マシンそのものの実行に使われるOSのことを「ホストOS」、仮想マシンの上で実行されるOSのことを「ゲストOS」という。仮想マシンでは、1台の物理コンピュータ上に複数のマシン環境を構築できることから、リソースの効率的な使用を目的としたサーバ集約に利用されている。またその他にも、ゲストOSを、1つのイメージファイルとして扱うことが可能であり、イメージファイルのコピーのみで複数のマシン環境を構築することが可能なため、大規模なシステム開発におけるテスト環境の構築等にも利用される。

**【0003】**

仮想マシンの構築方法として、下記特許文献1では、ゲストOSに適用される設定や、仮想マシンの新規構築・削除等を容易にする手段を提供している。また、複数の仮想マシンを1つの管理コンソール画面上から管理するための仮想マシン管理ツール等も整備されつつある。さらに、Windows（登録商標）のActive Directory（登録商標）等のように、ネットワークで接続されたコンピュータからOSの設定を行う手段が存在し、仮想マシンの構築はいっそう容易になってきている。

**【特許文献1】特開2005-332223****【発明の開示】****【発明が解決しようとする課題】****【0004】**

しかし、仮想マシンやゲストOSの複製・設定が容易にできるようになったことで、仮想マシンは増殖し、ゲストOSにおける設定やアプリケーションのインストール状況は多

10

20

30

40

50

様となるため、管理者は大量、かつ、複雑な管理を行わなければならないという問題がある。

【0005】

また、従来の仮想マシン管理システムでは、ゲストOSをアップデートすると仮想マシン上のゲストOSイメージがアップデートしたものに書き替えられてしまうので、それまでに構築したアプリケーション環境を容易に元に戻せないという問題がある。アプリケーション環境を元に戻すためには、再びアプリケーションのインストールから作業を行う必要があった。

【0006】

本発明は、このような問題に鑑みて発明されたものであり、その目的は、多様化するゲストOSのイメージファイルを一元化することで（以後、一元化されたゲストOSのイメージファイルをマスターゲストOSイメージと呼ぶ）、ゲストOS、及び、ゲストOS上で動作するアプリケーションの管理を容易にすることである。また、ゲストOSのアップデートの際にアプリケーション環境を容易に元に戻せるようにすることを目的とする。

10

【課題を解決するための手段】

【0007】

上記目的を達成するため、本発明は、複数の仮想マシンが動作する仮想マシンサーバにおいて、それら複数の仮想マシンを管理する仮想マシン管理システムであって、前記各仮想マシン毎に、その仮想マシン上で動作するゲストOSが管理する仮想ハードディスク上に仮想APイメージファイルを作成する手段と、任意の仮想マシン上で動作する任意のモジュールが、その仮想マシン上で動作するゲストOSが管理する仮想ハードディスク上の任意の領域に対して、任意のデータの書き込みまたは読み込みの処理を試みた場合に、該書き込みまたは読み込みが指示された領域から、前記仮想ハードディスク上に存在する仮想APイメージファイル上の所定の領域にリダイレクトを行い、書き込みまたは読み込み先を変更して書き込みまたは読み込みを実行することにより、前記任意のデータを前記仮想APイメージファイルに分離する手段と、前記仮想APイメージファイルを、前記仮想ハードディスクから、前記仮想マシンサーバ上のホストOSが管理する物理ハードディスク上へ、バックアップする手段と、前記物理ハードディスク上の前記仮想APイメージファイルを、前記仮想ハードディスク上にダウンロードする手段とを備え、前記任意の仮想マシン上で動作するモジュールに関連するデータを前記ゲストOSのイメージファイルから分離することにより、前記ゲストOSのイメージファイルを一元的に扱うことを可能にしたことを特徴とする。

20

30

【0008】

すなわち、本発明では、ゲストOSのイメージファイルはマスターゲストOSイメージとして一元化する。マスターゲストOSイメージをコピーして作成したゲストOS上で任意のモジュールが動作して仮想HDDに書き込み・読み込みをした場合、それらのデータは仮想APイメージファイルにリダイレクトされる。これにより、ゲストOSイメージと仮想APイメージとが明確に分離できる。

【0009】

本発明によれば、ゲストOSにおいてアップデート等の更新が容易である。例えば、ゲストOSをアップデートしたい場合は、マスターゲストOSイメージに対してアップデートを1回を行い、仮想マシンサーバ上で動作する任意のゲストOSのイメージファイルを、アップデート済みのマスターゲストOSイメージで上書きすることで、ゲストOSのアップデートが完了する。この際、アプリケーションに関連する実行コードやデータ等は上記仮想APイメージファイルとして分離されているので、アップデート済みのマスターゲストOSイメージで、ゲストOSのイメージファイルの上書きを行っても、ゲストOS上で動作していたアプリケーション等の任意のモジュールのデータは仮想APイメージファイルとして残されており消えることはない。通常、アプリケーションのインストール状況の異なるゲストOSのアップデートを行う場合、前記のゲストOSを起動して手動、または、自動でアップデート操作を行う必要があったが、本発明においては、マスターゲストOS

40

50

イメージで上書きするだけで仮想マシンサーバ上に存在する全てのゲストOSのアップデートを完了することが可能となる。また、本発明のシステムを使用すれば、起動していないゲストOSにおいて、ゲストOSを起動させずにアップデートすることも可能となる。

【発明の効果】

【0010】

本発明によれば、多様化するゲストOSのイメージファイルをマスターゲストOSイメージとして一元化することで、ゲストOS及びゲストOS上で動作するアプリケーションの管理が容易になる。また、ゲストOSのアップデートの際に、再びアプリケーションのインストールから作業を行う必要を無くし、アプリケーション環境を容易に元に戻すことが可能になる。

10

【発明を実施するための最良の形態】

【0011】

以下、図面を用いて、本発明の実施形態を説明する。

【0012】

図1は、本発明の一実施形態である仮想マシン管理システムのシステム構成を示す。なお、以後アプリケーションのことをAPと省略して記述する。

【0013】

図1のシステムは、仮想マシンサーバ100、ホストOS101、仮想マシン管理モジュール102、アップデートモジュール103、仮想マシン104、ゲストOS105、仮想AP化モジュール106、仮想APバックアップモジュール107（以後、仮想APBUモジュールと記述する）、仮想APダウンロードモジュール108（以後、仮想APDLモジュールと記述する）、仮想HDD109、仮想APイメージ110、仮想AP111、物理HDD112、マスターゲストOSイメージ113、ゲストOSイメージ114を備える。

20

【0014】

仮想マシンサーバ100は、複数台の仮想マシンを動作させるための物理的なコンピュータである。また、仮想マシンサーバ100は、1つの物理ハードディスク（HDD）112を具備している。本実施形態では、仮想マシン上の仮想HDDと区別するために、仮想マシンサーバ上でホストOSが管理している物理的なハードディスクを物理HDD112と呼ぶ。なお、図1で実際に物理的に存在し動作しているのは、仮想マシンサーバ100、ホストOS101、仮想マシン管理モジュール102、アップデートモジュール103、及び物理HDD112の部分である。点線120内の部分は仮想的に実現されている範囲を示す。また、本実施形態では1台の仮想マシンサーバ100のみを対象とした構成を挙げているが、複数台の仮想マシンサーバが存在する構成でもかまわない。

30

【0015】

仮想マシン管理モジュール102は、仮想マシン104を管理するためのモジュール（プログラム）であり、ホストOS101上で動作することにより、仮想マシンの追加・削除・起動・停止・設定（例えば、仮想マシンの名前の決定等）等の管理機能を提供する。また、仮想マシン管理モジュール102は、管理者が仮想マシンを管理するためのGUIインターフェースの機能も提供し、任意のタイミングで前記の管理機能を行わせることが可能である。

40

【0016】

仮想マシン104は、仮想マシンサーバ100上では、複数台の仮想マシン104を動作させることが可能である。仮想マシン環境を新規に構築するためには、仮想マシン管理モジュール102により、新規に仮想マシン104を作成した後、物理HDD112上にゲストOSイメージ114を新規に作成し、マウントする必要がある。この際、1台の仮想マシン104に対して1つのゲストOSイメージ114が必要である。逆に言えば、1つのゲストOSイメージ114を物理HDD112上に作成することにより1台の仮想マシン104が実現される。ゲストOSイメージ114を新規に作成する方法は、マスターゲストOSイメージ113（後述）を複製することで作成することができる。

【0017】

50

ゲストOS 105上には、仮想AP化モジュール106、仮想APBUモジュール107、及び仮想APDLモジュール108が常に動作している。

【0018】

仮想AP化モジュール106とは、ゲストOS 105やアプリケーション111等の任意のモジュールが仮想HDD 109に対して行うデータの読み込み・書き込み等のファイルI/O操作を監視し、それらのファイルI/Oを仮想APイメージ110（後述）上の任意の領域へリダイレクトする機能と、そのリダイレクトした先を記憶しておき、前記ゲストOS 105やアプリケーション111等の任意のモジュールが以前に仮想HDD 109上に対して読み込み・書き込みを行ったデータ（実際には、仮想AP化モジュール106により、仮想APイメージ110上に保存されているデータ）に読み込み・書き込みを行った場合、仮想APイメージ110上の当該領域を参照するようにリダイレクトする機能とを持つ。これらの機能は、ゲストOS 105上のアプリケーション111に関連するデータ（例えば、図3で説明するような仮想マシン上にインストーラによってインストールされた実行ファイルやドライバやデータ、あるいは図4で説明するような仮想AP 111が保存した保存データなど）を1つの仮想APイメージ110として分離することを可能とするものである。

10

【0019】

仮想APBUモジュール107とは、仮想HDD 109上に保存されている仮想APイメージ110（後述）を、ホストOS 101が管理する物理HDD 112上に、バックアップする機能を持つモジュールである。仮想APBUモジュール107は、仮想マシン104が停止する前には必ず仮想APイメージ110のバックアップを行う。115はバックアップされた仮想APイメージを示す。また、仮想APBUモジュール107は、仮想マシン管理モジュール102から任意のタイミングで呼び出すことも可能である。

20

【0020】

仮想APDLモジュール108とは、物理HDD 112上にバックアップされた仮想APイメージ115を仮想HDD 109上にダウンロードしてくる機能を持つモジュールである。仮想APDLモジュール107は、仮想マシン104が起動した直後には必ず仮想APイメージのダウンロード（115から110へのダウンロード）を行う。また、仮想APDLモジュール108は、仮想マシン管理モジュール102から任意のタイミングで呼び出すことも可能である。

【0021】

仮想APイメージ110は、仮想マシン104上で動作する複数のアプリケーション111等のモジュールに関連するデータの内、従来の仮想マシン環境では仮想HDD 109に保存されるデータを、仮想AP化モジュール106がリダイレクトして書き込むためのイメージファイルである。従来の仮想マシンではこのような仮想APイメージ110は無かったため、仮想マシンにアプリケーションがインストールされたときの実行ファイルや、仮想APが保存したデータなどは、ゲストOSイメージ内に組み込まれてしまい容易に分離できないデータとなっていた。本実施形態のシステムでは、ゲストOS 105に対応するゲストOSイメージ114は変更することなく、アプリケーションに関連するデータを仮想APイメージ110として切り離して取り扱えるようにしている。仮想APイメージ110は、1つのゲストOS 105に対して1つ存在する。また、1つの仮想APイメージ110内には、ある仮想マシン104上で動作する複数のアプリケーション111に関連するデータが書き込まれている。仮想APイメージ110は、ゲストOS 105のインストールが終了した時から、新規にアプリケーション等の任意のモジュールがインストールされるまでの間の任意のタイミングで、仮想AP化モジュール106により1つだけ作成される。

30

40

【0022】

仮想AP 111とは、仮想AP化モジュール106により実行ファイルやドライバやデータ等が仮想APイメージ110上に分離されているアプリケーションのことである（図3で詳述する）。仮想マシン104上で、新規にアプリケーションのインストールを行うと、従来の仮想マシン上では、仮想HDD上の任意の場所に保存されるはずのデータが、図1のシステムでは仮想APイメージ110上に保存される。そのため、ゲストOS 105のインストール時にインストールされないアプリケーションは全て仮想化され仮想AP 111となる。

50

## 【 0 0 2 3 】

マスターゲストOSイメージ113は、全てのゲストOSイメージ114のもととなるゲストOSのイメージファイルである。実際の内容は、仮想AP化モジュール106、仮想APBUモジュール107、及び仮想APDLモジュール108以外の、アプリケーション等の任意のモジュールが新規インストールされていない状態のゲストOSのイメージファイルである。なお、アップデートや設定等に関しては、最新または適切な状態にしておくことが必要である。マスターゲストOSイメージ113は、仮想マシン104にマウントされているゲストOS105のアップデートを行うときに使用される。マスターゲストOSイメージ113のアップデートはアップデートモジュール103により管理されている。アップデートに関しては、動作の説明で詳細に述べる。

10

## 【 0 0 2 4 】

アップデートモジュール103は、マスターゲストOSイメージ113のアップデートを行う機能を持っている。アップデートモジュール103は、ゲストOS105にアップデートが生じた場合、マスターゲストOSイメージ113に対してアップデートを1回行い、アップデート済みのマスターゲストOSイメージ113で、物理HDD112上のゲストOSイメージ114を上書きすることで、アップデートを完了する。従来の仮想マシン管理システムでは、ゲストOSのイメージファイルを他のゲストOSのイメージファイルで上書きすると、アプリケーション等の任意のモジュールに関連するデータは消去されてしまうが、図1のシステムでは、仮想AP化モジュール106により、アプリケーション等の任意のモジュールに関連するデータは仮想APイメージ110に分離され、物理HDD113上に保存されるため、アプリケーションのデータが消去されることはない。

20

## 【 0 0 2 5 】

以下、本実施の形態における処理の流れについて、図2～図6を参照して詳細に説明する。

## 【 0 0 2 6 】

図2は、マスターゲストOSイメージの作成フローである。マスターゲストOSイメージ113を作成するためには、まず、任意の仮想マシンを起動する(s200)。仮想マシンが存在しない場合は、新規に仮想マシンを作成し起動する。仮想マシンが起動したら、任意のOSのインストールを行う(s201)。インストール作業が終了したら、ゲストOS上に仮想APBUモジュール107、仮想APDLモジュール108、及び仮想AP化モジュール106をインストールする(s202)。次に、アップデート及び設定を最適な状態にする(s203)。最後に、ゲストOSをシャットダウンし、仮想マシンを停止させる(s204)。以上の操作を行うと、物理HDD上にゲストOSイメージが作成され、これをマスターゲストOSイメージ113とする。

30

## 【 0 0 2 7 】

図3は、APインストール時におけるAPデータの分離(AP仮想化)を示した図である。ここでは、アプリケーションのインストール時にアプリケーションのデータをゲストOSのイメージファイルから分離する動作について解説する。

## 【 0 0 2 8 】

まず、任意のアプリケーションAとアプリケーションBをインストールするものとし、それぞれのインストーラーをAPインストーラーA300及びAPインストーラーB301とする。APインストーラーAからインストールを開始すると、従来の仮想マシン環境ならば、点線で示すように、インストールデータ(例えば、実行ファイル、ドライバ、レジストリに保存されるデータ等)はゲストOS105が管理する仮想HDD109内の任意の場所(例えば、CドライブのProgramフォルダの下のApplicationAフォルダなど)に保存される(s300)。図3では、実行ファイルA302、ドライバA303、データA304などがそれである。

40

## 【 0 0 2 9 】

一方、本実施形態のシステムでは、仮想AP化モジュール106が、本来保存されるべき領域から、仮想APイメージ110上の所定の領域にリダイレクトして保存する(s301)。

50

図3では、仮想APイメージ110上の実行ファイルA305、ドライバA306、データA307などである。以上により、アプリケーションに関連するデータを仮想APイメージ110上に分離することが可能となる。

【0030】

図4は、仮想AP化モジュールの動作例の図である。

【0031】

まず、仮想AP111が保存データBの書き込みを試みたとする(s400)。仮想AP111は、仮想HDD109上の保存データB401の領域へ書き込みを試みる(s402)。しかし、実際には、仮想AP化モジュール106によりリダイレクトされ、仮想APイメージ110上の保存データB404の領域へ書き込みを行うことになる(s403)。このとき、仮想AP化モジュール106は、本来データが保存される仮想HDD上の保存データB401の領域と、リダイレクトした仮想APイメージ110上の保存データB404の領域との対応関係を記憶しておく。その後、仮想AP111またはゲストOS105上の任意のモジュールが、前記仮想HDD109上の保存データB401に対して読み込み・書き込み処理を行った場合、仮想AP化モジュール106は、仮想APイメージ110上の保存データB404の領域にリダイレクトして読み込み・書き込みを行わせる。以上により、仮想APイメージ110を仮想HDD109から物理HDDへバックアップすることにより、アプリケーションのデータをゲストOSのイメージファイルから分離することが可能となる。よって、個々の仮想マシン上で使用される各ゲストOSのイメージファイルの管理が不要となりOSイメージの管理が容易になる。

10

【0032】

図5と図6は、ゲストOSのアップデートとそのフローを示している。

20

【0033】

ゲストOSをアップデートする必要がある場合、仮想マシン管理モジュール102は仮想マシン104に停止命令を出す。仮想マシン104上では停止する前に、仮想APBUモジュール107が仮想APイメージ110を物理HDD112上にバックアップする(s500)。その後、仮想マシン104を停止する(s501)。これにより、仮想マシン104上で動作するアプリケーション等の任意のモジュールに関連するデータが全て物理112上にコピーされたことになる。

【0034】

次に、アップデートモジュール103は、物理HDD112上に保存されているマスターゲストOSイメージ113のアップデート処理を行う(s502)。マスターゲストOSイメージ113のアップデートの方法に関しては、任意の仮想マシン上にマスターゲストOSイメージ113をマウントして、アップデートを行う方法等が挙げられる。マスターゲストOSイメージ113のアップデートが終了したら、アップデート済みのマスターゲストOSイメージ113でゲストOSイメージ114を上書きする(s503)。その後、仮想マシン管理モジュール102は、仮想マシン104にゲストOSイメージ114をマウントし起動させる(s504)。

30

【0035】

起動後、仮想APDLモジュール108は、物理HDD112上にバックアップされている仮想APイメージ115を仮想HDD109上にダウンロードし、仮想APイメージ110として配置する(s505)。仮想APイメージ110のダウンロード後、ゲストOS105は、仮想アプリケーション111の起動に必要な実行ファイル、ドライバ、データ等をメモリ上に配置してアプリケーションを起動する(s506)。この際、ゲストOS105は、仮想HDD109上の任意の場所へ実行ファイル、ドライバ、データの読み込みを試みるが、仮想AP化モジュール106のリダイレクト機能により、仮想HDD109上の仮想APイメージ110から読み込みが行われ、メモリ上に配置される。以上により、ゲストOSのアップデートが完了する。

40

【図面の簡単な説明】

【0036】

【図1】本発明の実施形態のシステム構成図

【図2】マスターゲストOSイメージ作成フロー図

【図3】APインストール時におけるAPデータの分離(AP仮想化)図

50

【図4】仮想AP化モジュール及び仮想APイメージの使用例の図

【図5】本発明におけるゲストOSのアップデート図

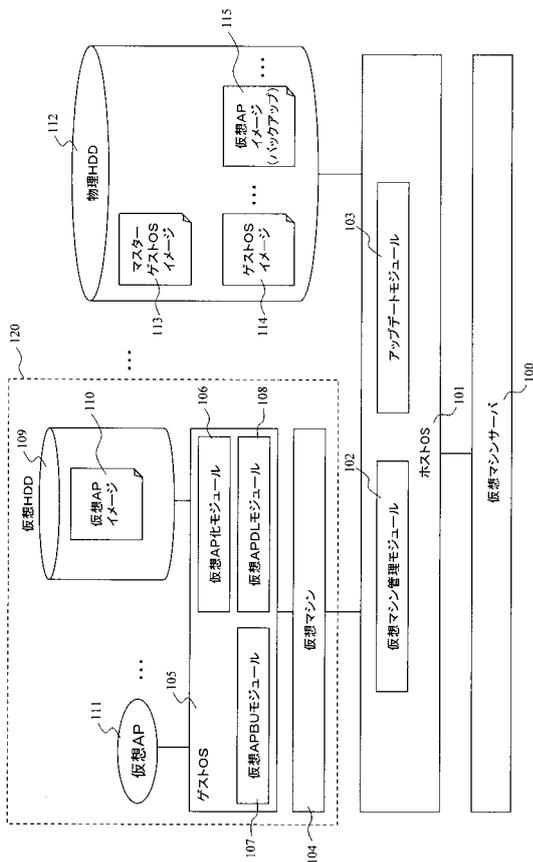
【図6】本発明におけるゲストOSのアップデートフロー図

【符号の説明】

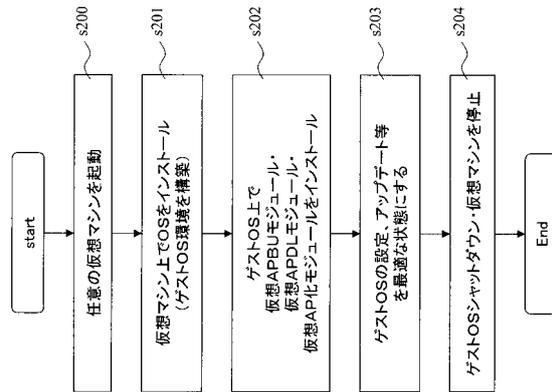
【0037】

100... 仮想マシンサーバ、101... ホストOS、102... 仮想マシン管理モジュール、103... アップデートモジュール、104... 仮想マシン、105... ゲストOS、106... 仮想AP化モジュール、107... 仮想APBUモジュール、108仮想APDLモジュール、109... 仮想HDD、110... 仮想APイメージ、111... 仮想AP、112... 物理HDD、113... マスターゲストOSイメージ、114... ゲストOSイメージ、115... 仮想APイメージ(バックアップ)

【図1】

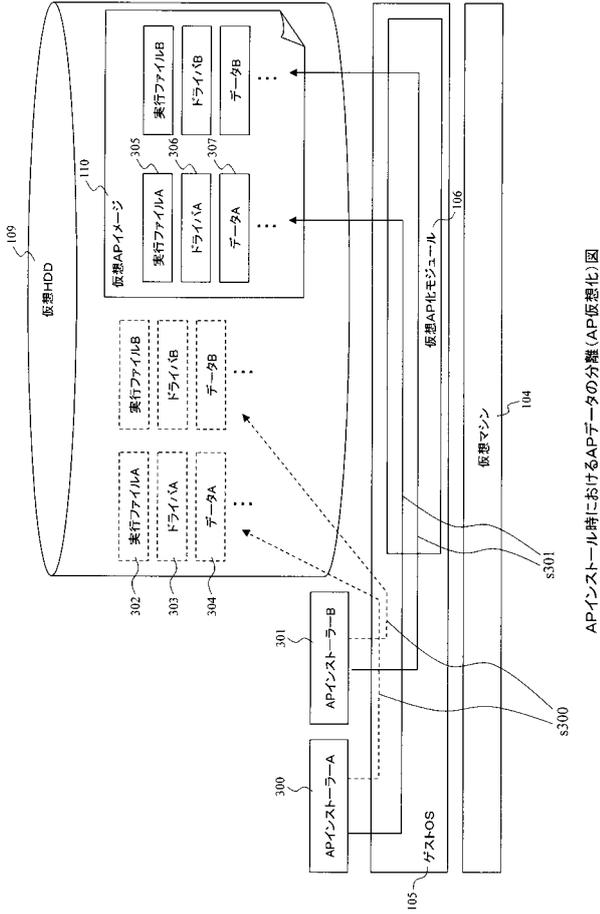


【図2】



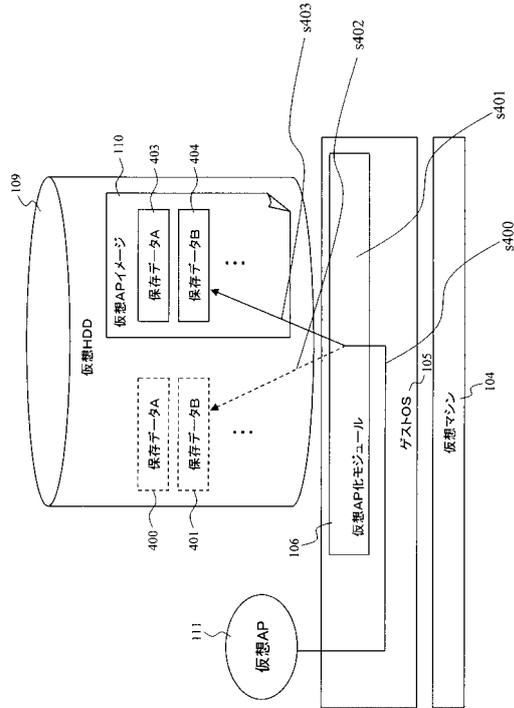
マスターゲストOSイメージ作成フロー

【図3】



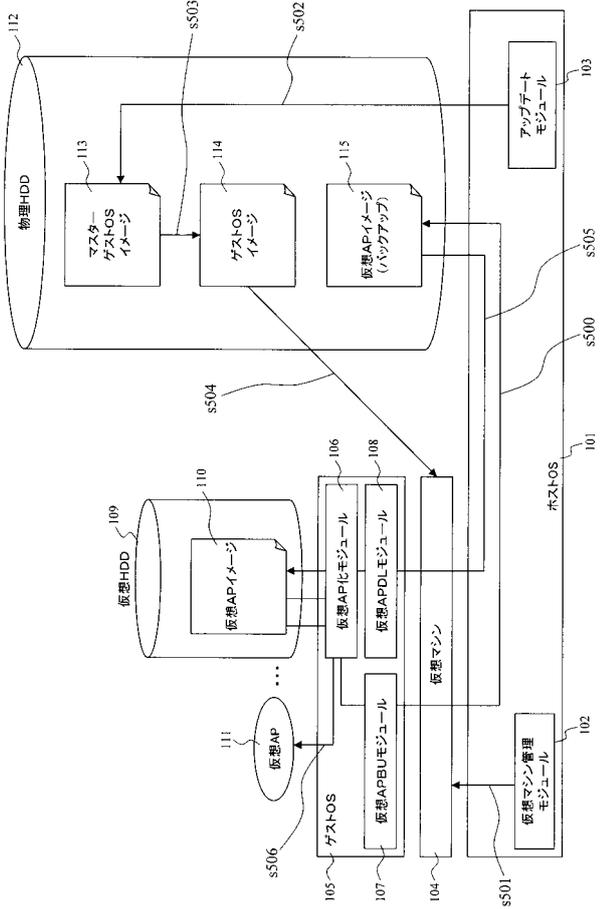
AP-インスタントル時におけるAPデータの分離 (AP仮想化) 図

【図4】



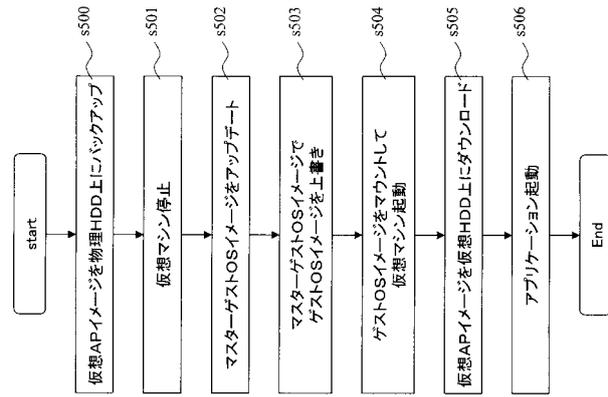
仮想AP化モジュール及び仮想APイメージの使用例の図

【図5】



本発明におけるゲストOSのアップデート図

【図6】



本発明におけるゲストOSのアップデートフロー