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

特許第5585844号
(P5585844)

(45)発行日　平成26年9月10日(2014.9.10)　(24)登録日　平成26年8月1日(2014.8.1)

(51)Int.Cl.		F I		
G06F 13/10	(2006.01)	G06F	13/10	330C
G06F 9/46	(2006.01)	G06F	9/46	350
G06F 9/50	(2006.01)	G06F	9/46	462Z

請求項の数 14　(全 26 頁)

(21)出願番号　特願2011-67444(P2011-67444)
(22)出願日　平成23年3月25日(2011.3.25)
(65)公開番号　特開2012-203636(P2012-203636A)
(43)公開日　平成24年10月22日(2012.10.22)
審査請求日　平成25年6月3日(2013.6.3)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　100114236
弁理士　藤井　正弘
(74)代理人　100075513
弁理士　後藤　政喜
(74)代理人　100120260
弁理士　飯田　雅昭
(72)発明者　服部　直也
東京都国分寺市東恋ヶ窪一丁目２８０番地
株式会社日立製作所　中央研究所内
(72)発明者　澤　勇太
東京都国分寺市東恋ヶ窪一丁目２８０番地
株式会社日立製作所　中央研究所内
最終頁に続く

(54)【発明の名称】仮想計算機の制御方法及び計算機

(57)【特許請求の範囲】
【請求項１】
プロセッサとメモリと仮想Ｉ／Ｏアダプタを仮想機能として生成する物理機能と前記仮想機能と前記物理機能のそれぞれの状態を管理するレジスタとを有するＩ／Ｏアダプタを備えた物理計算機と、前記物理計算機の計算機資源を仮想計算機に提供する仮想化部と、前記仮想計算機で実行されるＯＳと、を備えて、前記仮想化部が前記仮想機能を割り当てた１つ以上の仮想計算機を生成し、前記仮想計算機でＯＳを稼働させ、前記ＯＳに前記仮想機能を利用させる仮想計算機の制御方法であって、
前記仮想化部が、前記Ｉ／Ｏアダプタのレジスタを参照し前記Ｉ／Ｏアダプタの障害の発生を検出する第１のステップと、
前記仮想化部が、前記障害の発生した前記Ｉ／Ｏアダプタの前記物理機能により提供される前記仮想機能が割り当られた仮想計算機を特定する第２のステップと、
前記仮想化部が、前記特定された仮想計算機上で稼働する前記ＯＳに前記仮想機能の利用中止を指示する第３のステップと、
前記仮想化部が、前記ＯＳから前記利用中止の指示に対する応答を受領した後、前記Ｉ／Ｏアダプタの前記物理機能の回復処理を実施する第４のステップと、
を含むことを特徴とする仮想計算機の制御方法。
【請求項２】
請求項１に記載の仮想計算機の制御方法であって、
前記Ｉ／Ｏアダプタは、

障害が発生したときに、障害情報を格納する障害情報レジスタを有し、
前記第１のステップは、
前記仮想化部が、前記障害を示す信号を受信したときに前記障害情報レジスタを参照することを特徴とする仮想計算機の制御方法。
【請求項３】
請求項２に記載の仮想計算機の制御方法であって、
前記仮想計算機は、
前記Ｉ／Ｏアダプタの挿抜要求と通知を格納するホットプラグレジスタをさらに有し、
前記第２のステップは、
全ての前記仮想機能について前記特定を行い、
前記第４のステップは、
前記仮想化部が、前記ホットプラグレジスタに前記仮想機能の取り外し要求を設定する第５のステップと、
前記ＯＳが、前記ホットプラグレジスタを読み込む第６のステップと、
前記特定された仮想計算機上で稼働する全ての前記ＯＳが、前記ホットプラグレジスタに前記仮想機能の取り外し要求に対して応答するまで待機する第７のステップと、
をさらに含むことを特徴とする仮想計算機の制御方法。
【請求項４】
請求項３に記載の仮想計算機の制御方法であって、
前記仮想化部は、前記ＯＳが前記仮想機能の利用を中止した後に、前記Ｉ／Ｏアダプタを再初期化する第８のステップと、
前記Ｉ／Ｏアダプタが、前記仮想機能を再度生成し、前記仮想化部が、再度生成された仮想機能を前記仮想計算機に再度割り当てる第９のステップと、
前記仮想化部が、前記ホットプラグレジスタにＩ／Ｏアダプタの取り付け通知を設定する第１０のステップと、
前記ＯＳが、前記ホットプラグレジスタを読み込んで前記Ｉ／Ｏアダプタの取り付けを検知する第１１のステップと、
前記ＯＳが、前記仮想機能の利用を再開する第１２のステップと、
をさらに含むことを特徴とする仮想計算機の制御方法。
【請求項５】
請求項１に記載の仮想計算機の制御方法であって、
前記仮想化部は、
前記Ｉ／Ｏアダプタを操作するドライバを有し、当該ドライバには予め障害発生時の処理の方針が設定され、
前記第４のステップは、
前記仮想化部が、前記障害発生時には前記ドライバに対して、障害対処の方針を問い合わせる第１３のステップと、
前記ドライバに設定された前記方針がリセット試行である場合に、前記仮想計算機で稼働する前記ＯＳに前記仮想機能の状態を伝達する第１４のステップと、
をさらに含むことを特徴とする仮想計算機の制御方法。
【請求項６】
請求項３に記載の仮想計算機の制御方法であって、
前記仮想化部は、
前記Ｉ／Ｏアダプタを操作するドライバを有し、当該ドライバには予め障害発生時の処理の方針が設定され、
前記仮想化部は、前記ＯＳが前記仮想機能の利用を中止した後に、前記Ｉ／Ｏアダプタを再初期化する第８のステップと、
前記Ｉ／Ｏアダプタが、前記仮想機能を再度生成し、前記仮想化部が、再度生成された仮想機能を前記仮想計算機に再度割り当てる第９のステップと、
前記仮想化部が、前記ホットプラグレジスタにＩ／Ｏアダプタの取り付け通知を設定す

る第１０のステップと、
前記ＯＳが、前記ホットプラグレジスタを読み込んで前記Ｉ／Ｏアダプタの取り付けを検知する第１１のステップと、
前記ＯＳが、前記仮想機能の利用を再開する第１２のステップと、
をさらに含み、
前記第４のステップは、
前記仮想化部が、前記障害発生時には前記ドライバに対して、障害対処の方針を問い合わせる第１３のステップと、
前記ドライバに設定された前記方針がリセット試行である場合に、前記仮想計算機で稼働する前記ＯＳに前記仮想機能の状態を伝達する第１４のステップと、
をさらに含むことを特徴とする仮想計算機の制御方法。
【請求項７】
請求項１に記載の仮想計算機の制御方法であって、
前記仮想計算機は、第１のＯＳが稼働する第１の仮想計算機と第２のＯＳが稼働する第２の仮想計算機を含み、
前記仮想化部は、前記第１の仮想計算機及び第２の仮想計算機に前記仮想機能をそれぞれ割り当てて、前記第１のＯＳ及び前記第２のＯＳに前記割り当てた前記仮想機能をそれぞれ利用させ、
前記第４のステップは、
前記仮想化部が、前記第１のＯＳからの前記仮想機能の利用中止の指示に対する応答と、前記第２のＯＳからの前記仮想機能の利用中止の指示に対する応答とを待ち合わせることを特徴とする仮想計算機の制御方法。
【請求項８】
プロセッサとメモリと仮想Ｉ／Ｏアダプタを仮想機能として生成する物理機能と前記仮想機能と前記物理機能のそれぞれの状態を管理するレジスタとを有するＩ／Ｏアダプタを備えた物理計算機と、
前記物理計算機の計算機資源を仮想計算機に提供する仮想化部と、
前記仮想計算機で実行されるＯＳと、を備えて、前記仮想化部が、前記仮想機能を割り当てた１つ以上の仮想計算機を生成し、前記仮想計算機で前記ＯＳを稼働させ、前記ＯＳに前記仮想機能を利用させる計算機であって、
前記仮想化部は、
前記Ｉ／Ｏアダプタのレジスタを参照し前記Ｉ／Ｏアダプタの障害の発生を検出する障害処理部と、
前記物理機能が生成した前記仮想機能について、当該仮想機能を割り当てた前記仮想計算機を管理する割り当て情報と、
を備え、
前記障害処理部は、
前記障害の発生した前記Ｉ／Ｏアダプタの前記割り当て情報を参照し、前記物理機能により提供される前記仮想機能が割り当られた前記仮想計算機を特定し、前記特定された仮想計算機上で稼働する前記ＯＳに前記仮想機能の利用中止を指示し、前記ＯＳから前記利用中止の指示に対する応答を受領した後、前記Ｉ／Ｏアダプタの前記物理機能の回復処理を実施することを特徴とする計算機。
【請求項９】
請求項８に記載の計算機であって、
前記Ｉ／Ｏアダプタは、
障害情報を格納する障害情報レジスタを有し、
前記障害処理部は、
前記Ｉ／Ｏアダプタで発生した障害を示す信号を受信したときに前記障害情報レジスタを参照することを特徴とする計算機。
【請求項１０】

請求項９に記載の計算機であって、
前記仮想計算機は、
前記Ｉ／Ｏアダプタの挿抜要求と通知を格納するホットプラグレジスタをさらに有し、
前記障害処理部は、全ての前記仮想機能について前記特定を行い、
前記障害処理部は、
前記ホットプラグレジスタに前記仮想機能の取り外し要求を設定し、前記特定された仮想計算機上で稼働する全ての前記ＯＳが、前記ホットプラグレジスタに前記仮想機能の取り外し要求に対して応答するまで待機することを特徴とする計算機。
【請求項１１】
請求項１０に記載の計算機であって、
前記障害処理部は、
前記ＯＳが前記仮想機能の利用を中止した後に、前記Ｉ／Ｏアダプタを再初期化し、前記ホットプラグレジスタにＩ／Ｏアダプタの取り付け通知を設定し
前記Ｉ／Ｏアダプタは、前記再初期化の後に前記仮想機能を再度生成し、
前記仮想化部が、再度生成された前記仮想機能を前記仮想計算機に再度割り当てて、
前記ＯＳが、前記ホットプラグレジスタを読み込んで前記Ｉ／Ｏアダプタの取り付けを検知した後に、前記仮想機能の利用を再開することを特徴とする計算機。
【請求項１２】
請求項８に記載の計算機であって、
前記仮想化部は、
前記Ｉ／Ｏアダプタを操作するドライバを有し、当該ドライバには予め障害発生時の処理の方針が設定され、
前記障害処理部が、前記障害発生時には前記ドライバに対して、障害対処の方針を問い合わせ、前記ドライバに設定された前記方針がリセット試行である場合に、前記仮想計算機で稼働する前記ＯＳに前記仮想機能の状態を伝達することを特徴とする計算機。
【請求項１３】
請求項１０に記載の計算機であって、
前記仮想化部は、
前記Ｉ／Ｏアダプタを操作するドライバを有し、当該ドライバには予め障害発生時の処理の方針が設定され、
前記障害処理部は、前記ＯＳが前記仮想機能の利用を中止した後に、前記Ｉ／Ｏアダプタを再初期化し、
前記Ｉ／Ｏアダプタが、前記仮想機能を再度生成し、
前記仮想化部が、前記再度生成した仮想機能を前記仮想計算機に再度割り当てて、
前記障害処理部が、前記ホットプラグレジスタにＩ／Ｏアダプタの取り付け通知を設定し、
前記ＯＳが、前記ホットプラグレジスタを読み込んで前記Ｉ／Ｏアダプタの取り付けを検知して、前記仮想機能の利用を再開し、
前記障害処理部が、前記障害発生時には前記ドライバに対して、障害対処の方針を問い合わせ、前記ドライバに設定された前記方針がリセット試行である場合に、前記仮想計算機で稼働する前記ＯＳに前記仮想機能の状態を伝達することを特徴とする計算機。
【請求項１４】
請求項８に記載の計算機であって、
前記仮想計算機は、第１のＯＳが稼働する第１の仮想計算機と第２のＯＳが稼働する第２の仮想計算機を含み、
前記仮想化部は、前記第１の仮想計算機及び第２の仮想計算機に前記仮想機能をそれぞれ割り当てて、前記第１のＯＳ及び前記第２のＯＳに前記割り当てた前記仮想機能をそれぞれ利用させ、
前記障害処理部は、
前記第１のＯＳからの前記仮想機能の利用中止の指示に対する応答と、前記第２のＯＳ

からの前記仮想機能の利用中止の指示に対する応答とを待ち合わせることを特徴とする計算機。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、仮想計算機に関し、仮想計算機上で稼動するゲストＯＳを停止させずにＩ／Ｏデバイスの障害回復を処理する技術に関する。

【背景技術】

【０００２】

オープン系サーバの性能向上および機能拡充に伴い、物理サーバに搭載されるプロセッサコアを有効に活用する方法として、サーバ仮想化ソフトウェア（ハイパバイザやＶＭＭ）が広く用いられている。ハイパバイザは、１台の物理サーバ上で複数の仮想計算機を生成し、それぞれの仮想計算機でＯＳやアプリケーションを稼働させる。プロセッサの性能向上と相まって、近年では物理サーバ上で１０以上の仮想計算機（ＬＰＡＲ）を稼働させるケースも珍しくない。しかし、物理サーバ上で稼動する仮想計算機の数が増えるにつれ、Ｉ／Ｏデバイスに関する２つの問題が浮上していた。

【０００３】

問題１（性能問題）：物理Ｉ／Ｏアダプタを、複数の仮想Ｉ／Ｏアダプタとして振る舞わせる「Ｉ／Ｏ共有」を実現するためにハイパバイザの介入が不可欠だった。この場合、ハイパバイザの介入オーバヘッドが原因で、仮想計算機の利用できるＩ／Ｏ性能が限られていた。

【０００４】

問題２（信頼性問題）：オープン系サーバでは従来、Ｉ／Ｏアダプタ障害の内容をＯＳ等のソフトウェアに伝達する仕組みが存在しなかった。そのため、物理Ｉ／Ｏアダプタに障害が発生すると、障害の規模や種類が判断できないために必ず物理サーバ全体がダウンし、物理サーバ上で稼働する全ての仮想計算機に被害が及んでいた。

【０００５】

これらの問題を解決するためにＰＣＩ　Ｓｉｇから、ＩＯＶ(I/O Virtualization)とＡＥＲ (Advanced Error Reporting)という２つの規格が制定された。

【０００６】

ＩＯＶは、上記の性能問題を解決するために、「Ｉ／Ｏ共有」の主要部をハードウェアで提供する仕組みである。ＩＯＶを用いると、ハイパバイザの介入が初期化等の低頻度処理のみとなるため、仮想計算機で高いＩ／Ｏ性能を利用可能になる。

【０００７】

一方、ＡＥＲは、非特許文献１で示すように、上記信頼性問題を改善するために、Ｉ／Ｏアダプタの障害情報をＯＳ等のソフトウェアに伝達する仕組みである。ＡＥＲによって、ＯＳ等のソフトウェアは障害の深刻度を判断可能となり、軽度な障害であればリセット等の方法でＩ／Ｏアダプタを回復させ、物理サーバや仮想計算機の稼働を継続可能となる。

【０００８】

なお、ＩＯＶを利用する技術として、特許文献１にハイパバイザの基本的な動作が開示されている。特許文献１ではＩＯＶに対応するＩ／Ｏアダプタの物理機能（以下、ＰＦ：Ｐｈｙｓｉｃａｌ　Ｆｕｎｃｔｉｏｎ）の障害に対処する方法として、ＰＦのＨｏｔ Ｐｌｕｇ（サーバを稼働させた状態でＰＦを抜き差しする技術）を利用する技術が含まれている。特許文献１を使用すれば、ＰＦに障害が発生した場合には、保守要員や管理者等がＩ／Ｏアダプタを交換することができる。

【先行技術文献】

【特許文献】

【０００９】

【特許文献１】米国特許出願公開第２００９／１３３０１６号明細書

【非特許文献】
【００１０】
【非特許文献１】PCI Express(r) Base Specification Revision 2.1 (§7.10. Advanced Error Reporting Capability)
【発明の概要】
【発明が解決しようとする課題】
【００１１】
しかしながら、上記特許文献１には軽度な障害への対処方法が言及されていない。Ｉ／Ｏアダプタの電源投入直後、及びリセット直後は、従来との互換性の観点からＩＯＶ機能が無効化される。このため、ＩＯＶに対応したＩ／Ｏアダプタは、ＨｏｔＰｌｕｇの直後ではＩ／Ｏ共有の為の仮想Ｉ／Ｏアダプタ機能（以下、ＶＦ：Ｖｉｒｔｕａｌ　Ｆｕｎｃｔｉｏｎとする）が無効化されているため、特許文献１の技術を用いると以下の問題が生じる。
【００１２】
軽度な障害に対して、ＡＥＲの仕組みに則ってＩ／Ｏアダプタ（ＰＦ）をリセットすると、ＩＯＶ機能が無効化されるため、仮想計算機からＩＯＶによって生成されていた仮想Ｉ／Ｏアダプタ機能（ＶＦ）が消失する。仮想計算機上で動くＯＳ（以下、ゲストＯＳ）は、一般に物理Ｉ／Ｏアダプタ（ＰＦ）向けに設計されているため、Ｉ／Ｏアダプタが突然消失する事態を想定していない。そのためＶＦが突然消失するとゲストＯＳがクラッシュするケースがある。また、上記特許文献１では、軽度な障害に対してもＰＦをリセットしなければならず、障害から回復するために保守員による交換作業が発生し、保守の手間がかかる、と言う問題があった。
【００１３】
そこで本発明は、上記問題点に鑑みてなされたもので、ＩＯＶに対応するＩ／Ｏアダプタの高性能、高信頼性、保守容易性を充足することを目的とする。特に、Ｉ／Ｏアダプタのリセットが引き起こすＶＦの無断消失に起因するゲストＯＳのクラッシュを回避し、Ｉ／Ｏアダプタの軽度な障害から自動的に回復させることを目的とする。
【課題を解決するための手段】
【００１４】
本発明は、プロセッサとメモリと仮想Ｉ／Ｏアダプタを仮想機能として生成する物理機能と前記仮想機能と物理機能のそれぞれの状態を管理するレジスタとを有するＩ／Ｏアダプタを備えた物理計算機と、前記物理計算機の計算機資源を仮想計算機に提供する仮想化部と、前記仮想計算機で実行されるＯＳと、を備えて、前記仮想化部が前記仮想機能を割り当てた１つ以上の仮想計算機を生成し、前記仮想計算機でＯＳを稼働させ、前記ＯＳに前記仮想機能を利用させる仮想計算機の制御方法であって、前記仮想化部が、前記Ｉ／Ｏアダプタのレジスタを参照し前記Ｉ／Ｏアダプタの障害の発生を検出する第１のステップと、前記仮想化部が、前記障害の発生した前記Ｉ／Ｏアダプタの前記物理機能により提供される前記仮想機能が割り当られた仮想計算機を特定する第２のステップと、前記仮想化部が、前記特定された仮想計算機上で稼働する前記ＯＳに前記仮想機能の利用中止を指示する第３のステップと、前記仮想化部が、前記ＯＳから前記利用中止の指示に対する応答を受領した後、前記Ｉ／Ｏアダプタの前記物理機能の回復処理を実施する第４のステップと、を含む。
【発明の効果】
【００１５】
したがって、本発明は、Ｉ／Ｏ共有機能を有する物理Ｉ／Ｏアダプタを備えて、仮想化部は、仮想Ｉ／Ｏアダプタ（ＶＦ）を仮想計算機に割り当てる。障害発生時などの特定のケースを除けば仮想化部の動作頻度が低いため、高性能を実現できる。次に、ＶＦ消失の契機となる物理Ｉ／Ｏアダプタの機能（ＰＦ）のリセットに先だって、仮想計算機のＯＳからＶＦの取り外し許可を受ける。これにより、ＶＦ消失によるＯＳクラッシュを回避できる。従って高信頼である。ＰＦドライバが軽度な障害と判断した場合に、保守要員の手

を借りずにＰＦをリセットし、物理Ｉ／Ｏアダプタの回復を試みる。物理Ｉ／Ｏアダプタの回復に成功した場合は仮想計算機にＶＦを割り当て直し、仮想計算機を障害発生前の状態に戻す。従って保守が容易である。以上より、高性能、高信頼、保守容易性が充足される。

【図面の簡単な説明】

【００１６】

【図１】本発明の第１の実施形態を示し、計算機システムの一例を示すブロック図である。

【図２】本発明の第１の実施形態を示し、物理計算機及びソフトウェアの関係を示すブロック図である。

【図３】本発明の第１の実施形態を示し、物理計算機のメモリマップの一例である。

【図４】本発明の第１の実施形態を示し、アダプタ割り当て表の一例である。

【図５】本発明の第１の実施形態を示し、ＰＦ状態表の一例である。

【図６】本発明の第１の実施形態を示し、ＶＦ状態表の一例である。

【図７】本発明の第１の実施形態を示し、ハイパバイザの全体的な処理を説明するフローチャートの例である。

【図８】本発明の第１の実施形態を示し、ハイパバイザのＩ／Ｏ障害処理を説明するフローチャートの例である。

【図９】本発明の第１の実施形態を示し、ハイパバイザのＰＦ障害処理を説明するフローチャートの例である。

【図１０】本発明の第１の実施形態を示し、ハイパバイザのＶＦ障害処理を説明するフローチャートの例である。

【図１１】本発明の第１の実施形態を示し、ハイパバイザのＩ／Ｏ仮想化処理を説明するフローチャートの例である。

【図１２】本発明の第１の実施形態を示し、ハイパバイザのＰＦ回復処理を説明するフローチャートの例である。

【図１３】本発明の第１の実施形態を示し、ハイパバイザのＶＦ交換処理を説明するフローチャートの例である。

【図１４】本発明の第２の実施形態を示し、物理計算機及びソフトウェアの関係を示すブロック図である。

【図１５】本発明の第２の実施形態を示し、物理計算機のメモリマップの一例である。

【図１６】本発明の第２の実施形態を示し、ハイパバイザのＰＦ障害処理を説明するフローチャートの例である。

【図１７】本発明の第２の実施形態を示し、ハイパバイザのＶＦ障害処理を説明するフローチャートの例である。

【図１８】本発明の第２の実施形態を示し、ハイパバイザのＩ／Ｏ仮想化処理を説明するフローチャートの例である。

【図１９】本発明の第２の実施形態を示し、ハイパバイザのＰＦ回復処理を説明するフローチャートの例である。

【図２０】本発明の第２の実施形態を示し、ハイパバイザのＶＦ回復処理を説明するフローチャートの例である。

【図２１】本発明の第２の実施形態を示し、ハイパバイザのＶＦ交換処理を説明するフローチャートの例である。

【発明を実施するための形態】

【００１７】

以下、本発明の一実施形態を添付図面に基づいて説明する。

【００１８】

図１は、本発明の第１の実施形態を示し、ＩＯＶ（I/O Virtualization）とＡＥＲ（Advanced Error Reporting）に対応するＩ／Ｏアダプタを用いた仮想計算機システムの一例を示すブロック図である。本実施形態のＩ／Ｏアダプタ６０は、物理的なＩ／Ｏアダプ

タの物理機能（以下、ＰＦ：Ｐｈｙｓｉｃａｌ　Ｆｕｎｃｔｉｏｎ）と、「Ｉ／Ｏ共有」を実現する仮想Ｉ／Ｏアダプタ（以下、ＶＦ：Ｖｉｒｔｕａｌ　Ｆｕｎｃｔｉｏｎ）を生成する機能を有し、ＶＦを仮想計算機に割り当てる場合の構成及び動作を説明する。

【００１９】

＜１．ハードウェア構成＞

図１は、本発明の第１の実施形態となる仮想計算機システムを動作させる物理計算機の構成例を示す。物理計算機は、障害検出機能を有するＣＰＵ７０（またはプロセッサコア）を１つ以上有し、これらのＣＰＵ７０はＱＰＩ（Quick Path Interconnect）やＳＭＩ（Scalable Memory Interconnect）等のインターコネクト１１０を介してＣｈｉｐＳｅｔ１００やメモリ９０に接続される。

【００２０】

ＣｈｉｐＳｅｔ１００には、ＰＣＩｅｘｐｒｅｓｓ等のバス１２０を介してＩ／Ｏアダプタ６０が接続される。Ｉ／Ｏアダプタ６０はＬＡＮ１３０に接続されるネットワークアダプタ、ディスク装置１４０等に接続されるＳＣＳＩアダプタ、ＳＡＮ１５０（Storage Area Network）に接続されるファイバーチャネルアダプタ、コンソール８０に接続されるグラフィックコントローラなどで構成される。

【００２１】

ＣＰＵ７０はインターコネクト１１０を介してメモリ９０にアクセスし、ＣｈｉｐＳｅｔ１００からＩ／Ｏアダプタ６０にアクセスして所定の処理を行う。

【００２２】

メモリ９０には、仮想化部としてハイパバイザ２０がロードされ、ハイパバイザ２０が制御する仮想計算機３０－１～３０－ｎが、ゲストＯＳ４０－１～４０－ｎを実行する。ハイパバイザ２０は、物理計算機１０の計算器資源を分割または仮想化して仮想計算機に割り当てる。なお、以下では仮想計算機３０－１～３－ｎの総称を符号３０で示し、ゲストＯＳ４０－１～４０－ｎの総称を符号４０で示す。

【００２３】

＜２．ソフトウェア構成＞

次に、物理計算機１０上で仮想計算機３０を実現するソフトウェアの構成の主要部と、制御対象となるハードウェア要素について、図２を参照しながら詳述する。図２は、物理計算機及びソフトウェアの関係を示すブロック図である。

【００２４】

物理計算機１０上では、１つ以上の仮想計算機３０を制御するハイパバイザ２０が稼動する。物理計算機１０は、ＣｈｉｐＳｅｔ１００と、１つ以上のＩ／Ｏアダプタ６０を含む。

【００２５】

ＣｈｉｐＳｅｔ１００は、Ｉ／Ｏアダプタ６０の取り外し（挿抜）要求（ＨｏｔＲｅｍｏｖｅの要求）を保持するレジスタと、取り付け告知（ＨｏｔＡｄｄの通知）を保持するレジスタから成るＨｏｔＰｌｕｇレジスタ（以下、ＨＰレジスタ）１８５を含む。またＣｈｉｐＳｅｔ１００は、Ｉ／Ｏアダプタ６０の取り外し要求または取り付け告知あるいは障害を認識すると割り込みを発生して、物理計算機１０上で稼働するハイパバイザ２０に要求や障害の発生を通知する機能を有する。

【００２６】

１つ以上のＩ／Ｏアダプタ６０はＩＯＶ機能を有しており、物理的な機能（ＰＦ：Physical Function）１６０（１６０－１～１６０－ｍ）と仮想的な機能（ＶＦ：Virtual Function）１９０（１９０－１－１～１９０－ｍ－ｋ）から成る。ＰＦ１６０はいつでも利用できる機能だが、ＶＦ１９０はＩＯＶ機能が有効な場合のみ利用できる機能である。なお、仮想計算機１０は、ＩＯＶに非対応のＩ／Ｏアダプタ６０を含んでも構わない。

【００２７】

ＰＦ１６０は、物理計算機１０の外部とデータを送受信する機能を有し、ＩＯＶ機能を制御するＩＯＶレジスタ１７０と、ＰＦに関する障害の有無や種類といった障害情報を保

持するＡＥＲレジスタ１８０を含む。
【００２８】
ＶＦ１９０は、ＩＯＶ機能が有効な場合のみ、物理計算機１０の外部とデータを送受信する機能を有し、ＶＦに関する障害の有無や種類といった障害情報を保持するＡＥＲレジスタ１８０を含む。
【００２９】
ここで、ＡＥＲレジスタ１８０は、２種類存在し、ＰＦ１６０のＡＥＲレジスタ１８０とＶＦ１９０のＡＥＲレジスタ１８０が存在する。ＰＦ１６０のＡＥＲレジスタ１８０は物理機能、つまり物理Ｉ／Ｏアダプタ６０の障害に関する情報を保持し、ＶＦ１９０のＡＥＲレジスタ１８０は仮想機能、つまり仮想Ｉ／Ｏアダプタの障害に関する情報を保持する。なお、Ｉ／Ｏアダプタ６０は、障害発生時にＡＥＲレジスタ１８０に障害情報を設定するコントローラ（図示省略）を有する。
【００３０】
ハイパバイザ２０は、仮想計算機３０を生成し、仮想計算機３０上に対してＣｈｉｐＳｅｔ１００に相当する機能（仮想ＣｈｉｐＳｅｔ３００）を提供する。またハイパバイザ２０は、任意のＶＦ１９０を任意の仮想計算機３０に占有的に割り当て、当該仮想計算機３０上で稼働するゲストＯＳ４０に当該ＶＦ１９０の直接操作を許す機能（パススルー機能）を備える。このパススルー機能によって、複数の仮想計算機３０でひとつのＩ／Ｏアダプタ６０をＶＦ１９０から共有することができる。
【００３１】
ハイパバイザ２０は、仮想計算機３０に対するＶＦ１９０の割り当て関係を保持するアダプタ割り当て表２００と、物理計算機１０または物理計算機資源を監視して障害発生時に呼び出される障害処理部２１０と、仮想計算機３０の状態を保持する仮想計算機のエミュレーションデータと、ＰＦ１６０の種類に応じた制御手順を備えたＰＦドライバ２５０（２５０－１～２５０－ｍ）を含む。
【００３２】
障害処理部２１０は、ＩＯＶ機能に対応したＩ／Ｏアダプタ６０の障害を受け持つＩＯＶ障害処理部２１２を含み、ＩＯＶ障害処理部２１２はＰＦ１６０の障害処理状況を保持するＰＦ状態表２１４と、ＶＦ１９０の障害処理状況を保持するＶＦ状態表２１６を含む。
【００３３】
仮想計算機のエミュレーションデータ２２０は、仮想計算機３０に提供する仮想ＣｈｉｐＳｅｔ３００などの仮想的な部品またはデバイスに関する状態を保持するために、仮想ＣｈｉｐＳｅｔデータ２２２を含む。仮想ＣｈｉｐＳｅｔデータ２２２は、仮想ＣｈｉｐＳｅｔ３００が保持すべきＨＰレジスタ１８５などの状態を保持する。
【００３４】
ＰＦドライバ２５０は、ＩＯＶレジスタ１７０を操作する機能と、ＰＦ１６０で障害が発生した場合に軽度障害か重度障害かを判断する機能、及びＰＦリセット後に障害回復の成否を判断する機能を有する。
【００３５】
仮想計算機３０は、ハイパバイザ２０によって提供される仮想ＣｈｉｐＳｅｔ３００等の仮想的な部品と、占有的に割り当てられたＶＦ１９０を含む。仮想計算機３０の上ではゲストＯＳ４０が動作する。ゲストＯＳ４０は、ＶＦ１９０の種類に応じたＶＦドライバ２６０を用いてＶＦ１９０を操作する。
【００３６】
ＶＦドライバ２６０は、ＶＦ１９０で障害が発生した場合に軽度障害か重度障害かを判断する障害の程度を判定する機能、及びＶＦリセット後に障害回復の成否を判定する機能を有する。
【００３７】
図３はハイパバイザ２０が管理するメモリ９０のアドレスマップの一例を示す。

【００３８】
ハイパバイザ２０は、メモリ９０上に自身を配置する領域と、仮想計算機３０が使用する領域を割り当てる。例えば図３のように、ハイパバイザ２０は、自身にアドレスＡＤ０～ＡＤ１を割り当て、仮想計算機３０－１にアドレスＡＤ１＋１～ＡＤ２を、仮想計算機３０－ｎにアドレスＡＤ３～ＡＤ４を割り当てる。 各仮想計算機３０の領域には、ゲストＯＳ４０とＶＦドライバ２６０（２６０－１～２６０－ｎ）が格納される。
【００３９】
ハイパバイザ２０が使用する領域には、アダプタ割り当て表２００、障害処理部２１０、仮想計算機のエミュレーションデータ２２０と、ＰＦドライバ２５０が格納される。
【００４０】
図４は、アダプタ割り当て表２００の構成例である。アダプタ割り当て表２００は、ＶＦ１９０と仮想計算機３０の割り当て関係を保持する表である。本表ではＶＦ１９０を識別する手段として、ＶＦ１９０を生成したＰＦ１６０の通し番号（ＰＦ＃）４００と、ＶＦ１９０の通し番号（ＶＦ＃）４１０を用い、当該ＶＦ１９０を割り当てた仮想計算機３０の通し番号（仮想計算機番号）４２０を格納する。未割り当ての場合は仮想計算機番号４２０を"未割り当て"とする。
【００４１】
図５は、ＰＦ状態表２１４の構成例である。ＰＦ状態表２１４は、ＰＦ１６０の通し番号（ＰＦ＃）４００と、ＰＦ１６０の障害処理状況として、ＰＦ１６０の状態５１０と、当該ＰＦ１６０から生成された各ＶＦ１９０の取り外し許可の有無を格納する待ち合わせｂｉｔｍａｐ５２０からひとつのエントリが構成される。ＰＦ１６０の識別にはＰＦ＃４００を使用する。状態５１０は、障害処理が不要な状態（正常）、軽度障害に伴ってＰＦ１６０のリセットが必要となった状態（リセット保留）、重度障害に伴ってＰＦ１６０の交換が必要となった状態（交換保留）のいずれかを保持する。待ち合わせｂｉｔｍａｐ５２０は、各ｂｉｔが当該ＰＦ１６０の生成した各ＶＦ１９０に対応しており、ＶＦ１９０の取り外し許可をゲストＯＳ４０に要求中の場合に「１」、未要求または許可済みの場合に「０」が格納される。待ち合わせｂｉｔｍａｐ５２０の全ｂｉｔが０になるとＰＦのリセットが可能になる。
【００４２】
図６は、ＶＦ状態表２１６の構成例である。ＶＦ状態表２１６は、ＰＦ１６０の通し番号（ＰＦ＃）４００と、ＶＦ１９０の通し番号（ＶＦ＃）４１０と、ＶＦ１９０の障害処理状況として、ＶＦ１９０の状態６２０を保持する。ＶＦ１９０の識別には、ＰＦ＃４００と、ＶＦ＃４１０を使用する。状態６２０は、障害処理が不要な状態（正常）、軽度障害に伴ってＶＦ１９０のリセットが必要となった状態（リセット保留）、重度障害に伴ってＶＦ１９０の交換が必要となった状態（交換保留）のいずれかを保持する。
【００４３】
＜３．ハイパバイザによる処理＞
次に、ハイパバイザ２０が行う処理の一例について、以下、フローチャートを参照しながら説明する。
【００４４】
＜３．１．ハイパバイザによる処理の概要＞
図７は、ハイパバイザ２０が行う処理の全体像を示すフローチャートの例である。物理計算機１０の電源を投入するとハイパバイザ２０がメモリ９０にロードされ、ハイパバイザ２０自身と物理計算機１０を初期化し、Ｉ／Ｏアダプタ６０のＩＯＶ機能を有効化する（７００）。ステップ７００では、ＰＦ状態表２１４は全ＰＦが正常状態に初期化され、ＶＦ状態表２１６は全ＶＦが正常状態に初期化され、アダプタ割り当て表２００は全ＶＦが未割り当てに初期化される。
【００４５】
続いてハイパバイザ２０は、コンソール８０からの入力、或いは前回起動時の割り当て指示に基づいて、ＶＦ１９０を割り当てた仮想計算機３０を生成する（７１０）。ステッ

プ７１０では、アダプタ割り当て表２００の、ＶＦ１９０の行に割り当てられた仮想計算機３０の通し番号（識別子）を登録する。

【００４６】

続いてハイパバイザ２０は、仮想計算機３０の稼働を開始し、仮想計算機３０上でゲストＯＳ４０及びアプリケーション（以下、両者を総称してゲストと記載）を実行する（７２０）。ステップ７２０ではゲストが任意の命令コードを実行できるが、１命令を実行する毎にＣＰＵ７０がハイパバイザ２０の介入が必要なイベントが発生しているかをチェックする（７３０）。ハイパバイザ２０の介入が必要なイベントが発生していれば、ＣＰＵ７０がハイパバイザ２０を呼び出してステップ７４０に進む。ハイパバイザ２０の介入が必要なイベントが発生していなければ、ゲストの次の命令を実行するためにステップ７２０に進む。

【００４７】

ハイパバイザ２０が呼び出されると、ＣＰＵ７０から発生したイベントの情報を受け取って解析し、ハイパバイザ２０はイベントの内容を取得する（７４０）。

【００４８】

続いてハイパバイザ２０は、Ｉ／Ｏアダプタ６０に障害が発生したか否かを判定し（７５０）、Ｉ／Ｏアダプタ６０に障害が発生していればステップ７９０に進んでＩ／Ｏ障害処理を行う。本実施形態のＩ／Ｏアダプタ６０は、障害発生時に割込信号を発生する。ハイパバイザ２０は、この割込信号を障害発生のイベントとして受信する。

一方、ハイパバイザ２０は、Ｉ／Ｏアダプタ６０に障害が発生していなければ、ステップ７６０に進んでゲストが仮想チップセット３００等に対してＩ／Ｏ操作を行ったか否かを判定する。ゲストがＩ／Ｏ操作を行っていれば、ステップ７８０に進んでＩ／Ｏ仮想化処理を行う。ゲストがＩ／Ｏ操作を行っていなければステップ７７０に進み、ハイパバイザ２０の従来の処理（解析したイベントに応じた処理）を行う。ハイパバイザ２０の従来の処理については、公知または周知の処理を実施すればよいので、説明を省略する。

【００４９】

ステップ７７０、７８０、７９０の何れかで発生したイベントに対する処理が完了すると、ハイパバイザ２０はゲストの実行を再開させる（７９５）。

【００５０】

＜３．２．Ｉ／Ｏ障害処理＞

図８は、図７のステップ７９０で行われるＩ／Ｏ障害処理のフローチャートの例である。Ｉ／Ｏアダプタ６０に障害の発生を検出したハイパバイザ２０は、ＰＦ１６０又はＶＦ１９０が備えるＡＥＲレジスタ１８０を参照し、どのＩ／Ｏアダプタ６０で障害が発生したのかを特定する（８００）。

【００５１】

続いてハイパバイザ２０は、障害が発生したＩ／Ｏアダプタ６０の種類によって処理を分岐させる（８１０）。ハイパバイザ２０は発生した障害が、ＩＯＶに非対応のＩ／Ｏアダプタ（従来アダプタ）６０で発生した場合は、ステップ８２０に進んで従来の障害処理を行う。従来の障害処理については、公知または周知の処理を実施すればよいので、説明を省略する。

【００５２】

ハイパバイザ２０は発生した障害が、ＩＯＶ機能を有するＩ／ＯアダプタのＰＦ１６０部分で発生した場合は、ステップ８３０に進んでＰＦ障害処理を後述のように行う。一方、ハイパバイザ２０は発生した障害が、ＩＯＶ機能を有するＩ／ＯアダプタのＶＦ１９０部分で発生した場合は、ステップ８４０に進んでＶＦ障害処理を後述のように行う。

【００５３】

＜３．２．１．ＰＦ障害処理＞

図９は、図８のステップ８３０で行われるＰＦ障害処理のフローチャートの例である。障害が発生したＰＦ１６０を特定したハイパバイザ２０は、当該ＰＦ１６０の種類に応じ

たＰＦドライバ２５０に、障害からの回復が可能か否かを問い合わせ、障害の重度を取得する（９００）。Ｉ／Ｏアダプタ６０の障害の程度には３通りの重度があり、重い障害から順に、
（１）回復不能でＩ／Ｏアダプタ６０の交換が必要となる重度障害（PCI_ERS_RESULT_DISCONNECT）、
（２）ＰＦ１６０をリセットすれば回復できる可能性のある軽度障害（PCI_ERS_RESULT_NEED_RESET）、
（３）ＰＦ１６０のレジスタの一部を再設定する等の処理で回復が完了する障害（PCI_ERS_RESULT_RECOVERED）、
に分類される。

【００５４】
ハイパバイザ２０は、障害の重度に応じて処理を分岐させる（９１０）。

【００５５】
（１）回復不能な重度障害であれば、ＰＦ状態表２１４の該当するＰＦ１６０のエントリの状態５１０を、「交換保留」に変更する（９３０）。

【００５６】
（２）リセットで回復できる可能性がある軽度障害であれば、ハイパバイザ２０はＰＦ状態表２１４の該当するＰＦ１６０のエントリの状態５１０を、"リセット保留"に変更する（９６０）。

【００５７】
（３）回復完了した障害であれば、ハイパバイザ２０の回復処理は省略する。

【００５８】
上記（１）及び（２）のケースでは更に、ハイパバイザ２０がアダプタ割り当て表２００を参照し、障害が起きたＰＦ１６０から生成された全てのＶＦ１９０と、各ＶＦ１９０を割り当てた仮想計算機３０を特定する（９４０）。

【００５９】
続いてハイパバイザ２０は、各仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５にＶＦ１９０の稼働時取り外し（Ｈｏｔ　Ｒｅｍｏｖｅ）要求を設定する。そして、ハイパバイザ２０は、仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０にＶＦ１９０の利用中止と安全な取り外し（Ｓａｆｅｌｙ　Ｒｅｍｏｖｅ）を指令する（９５０）。

【００６０】
最後にハイパバイザ２０は、障害内容と対処結果をコンソール８０に出力する（９２０）。コンソール８０への出力により保守要員にＩ／Ｏアダプタ６０の保守を促す。Ｉ／Ｏアダプタ６０の障害の種類が（２）または（３）の場合は、直ちにＩ／Ｏアダプタ６０の交換を行う必要は無いが、Ｉ／Ｏアダプタ６０が経年劣化している可能性を考慮し、予防保守の観点から保守員に通報する。一方、障害の種類が（１）の場合は、直ちにＩ／Ｏアダプタ６０の交換を行う必要がある旨をコンソール８０に通知する。

【００６１】
＜３．２．２．ＶＦ障害処理＞
図１０は、図８の８４０で行われるＶＦ障害処理のフローチャートの例である。障害が発生したＶＦ１９０を特定したハイパバイザ２０は、ＶＦ状態表２１６の、当該ＶＦ１９０のエントリを“交換保留”に変更する（１０６０）。更に、ハイパバイザ２０がアダプタ割り当て表２００を参照し、障害が起きたＶＦ１９０を割り当てた仮想計算機３０を特定する（１０４０）。

【００６２】
続いてハイパバイザ２０は、当該仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５にＶＦ１９０の稼働時取り外し（Ｈｏｔ　Ｒｅｍｏｖｅ）要求を設定する。そして、ハイパバイザ２０は、仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０にＶＦ１９０の利用中止と安全な取り外し（Ｓａｆｅｌｙ　Ｒｅｍ

ｏｖｅ）を指令する（１０５０）。
【００６３】
最後にハイパバイザ２０は、障害内容と対処結果をコンソール８０に出力通報し、Ｉ／Ｏアダプタ６０の交換を促す（１０２０）。ＶＦ障害に対しては直ちに交換を行う必要は無いが、Ｉ／Ｏアダプタ６０が経年劣化している可能性を考慮し、予防保守の観点から保守員に通報する。
【００６４】
＜３．３．Ｉ／Ｏ仮想化処理＞
図１１は、図７のステップ７８０で行われるＩ／Ｏ仮想化処理のフローチャートの例である。ゲストによるＩ／Ｏ操作を検出したハイパバイザ２０は、操作対象（仮想ＣｈｉｐＳｅｔ３００等のレジスタ）及び操作内容（読み出し、書き込み）に応じて従来のＩ／Ｏエミュレーション処理を行う（１１００）。従来のＩ／Ｏエミュレーション処理については、公知または周知の処理を実施すればよいので、説明を省略する。
【００６５】
続いてハイパバイザ２０は、ゲストによるＩ／Ｏ操作の内容がＶＦ１９０の取り外し（ＨｏｔＲｅｍｏｖｅ）許可であったか否かを判定する（１１１０）。つまり、ハイパバイザ２０は、ゲストがＶＦ１９０の取り外しを許可するまで待機する。なお、ＶＦ１９０の取り外し許可の通知は、ゲストが仮想ＣｈｉｐＳｅｔ３００のホットプラグレジスタ１８５に許可の応答を書き込むことで行う。ハイパバイザ２０はホットプラグレジスタ１８５の値が所定の許可応答であれば、ゲストによるＩ／Ｏ操作の内容がＶＦ１９０の取り外し許可であった場合となる。この場合のみ、ハイパバイザ２０は、ＰＦ状態表２１４を参照し、当該ＶＦ１９０を生成したＰＦ１６０がリセット保留中であるかを判定する（１１２０）。
【００６６】
ステップ１１２０に於いてＰＦ１６０がリセット保留中だった場合、ハイパバイザ２０は、ＰＦ状態表２１４の当該ＰＦ１６０のエントリについて、待ち合わせｂｉｔｍａｐ５２０の取り外し許可が得られたＶＦ１９０に対応するｂｉｔを「０」に変更する（１１３０）。
【００６７】
続いてハイパバイザ２０は、ＰＦ状態表２１４の当該ＰＦ１６０の待ち合わせｂｉｔｍａｐ５２０で全ｂｉｔが「０」に揃ったか否かを判定する（１１４０）。全ｂｉｔが「０」に揃った場合のみ、１１５０に進んでＰＦ回復処理を行う。
【００６８】
ステップ１１２０に於いてＰＦ１６０がリセット保留中でなかった場合は、ハイパバイザ２０がＶＦ状態表２１６を参照し、取り外し許可が得られたＶＦ１９０が交換保留中であるか否かを判定する（１１８０）。ハイパバイザ２０は、ＶＦ１９０が交換保留中であった場合のみ、ステップ１１９０に進んでＶＦ交換処理を行う。
【００６９】
＜３．３．１．ＰＦ回復処理＞
図１２は、図１１のステップ１１５０で行われるＰＦ回復処理のフローチャートの例である。リセット保留中のＰＦ１６０から生成された全てのＶＦ１９０について、ゲストＯＳ４０からの取り外し許可を検出したハイパバイザ２０は、当該ＰＦ１６０をリセットする（１２００）。これによって、当該ＰＦ１６０＝Ｉ／Ｏアダプタ６０はリセットされ再起動する。
【００７０】
続いてハイパバイザ２０は、リセットしたＰＦ１６０の種類に応じたＰＦドライバ２５０を呼び出し、ＰＦ１６０の回復に成功したか否かをＩ／Ｏアダプタ６０に問い合わせる（１２１０）。ハイパバイザ２０は、ＰＦ１６０の回復の成否に応じて処理を分岐する（１２２０）。
【００７１】

ＰＦ１６０の回復に成功した場合は、ハイパバイザ２０がＰＦドライバ２５０にＰＦ１６０の再初期化を依頼し、ＩＯＶ機能を再度有効化してＰＦ１６０にＶＦ１９０を再生成させる（１２３０）。続いてハイパバイザ２０は、アダプタ割り当て表２００を参照して、当該ＰＦ１６０から生成されていた全てのＶＦ１９０について、障害発生前にどの仮想計算機３０に割り当てられていたかを取得し、当該仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５にＶＦ１９０の稼働時取り付け（Ｈｏｔ　Ａｄｄ）告知（または通知）を設定する。そして、ハイパバイザ２０は、仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０にＶＦ１９０の増設認識（再認識）と初期化及び利用開始を促す（１２４０）。これにより、ＨｏｔＰｌｕｇ割り込みを受信したゲストＯＳ４０は、仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５を読み込んで、ＨｏｔＡｄｄが発生したことを取得し、ＰＦ１６０が再生成したＶＦ１９０の再検知を行う。ゲストＯＳ４０は既にロードしていたＶＦドライバ２６０で再検知したＶＦ１９０の利用を再開する。

【００７２】

続いてハイパバイザ２０は、ＰＦ状態表２１４の当該ＰＦ１６０に対応するエントリの状態５１０を正常状態に戻す（１２５０）。

【００７３】

一方、ステップ１２２０でＰＦ１６０の回復に失敗した場合は、ハイパバイザ２０がＰＦ状態表２１４の障害ＰＦのエントリの状態５１０を交換保留に変更する（１２７０）。ハイパバイザ２０によるＰＦ回復処理の最後に、当該処理の結果をコンソール８０に出力して保守要員によるＩ／Ｏアダプタ６０の交換を促す（１２６０）。回復に成功した場合は直ちに交換を行う必要は無いが、Ｉ／Ｏアダプタ６０が経年劣化している可能性を考慮し、予防保守の観点から保守要員に通報する。

【００７４】

＜３．３．２．ＶＦ交換処理＞

図１３は、図１１のステップ１１９０で行うＶＦ回復処理のフローチャートの例である。交換保留中のＶＦ１９０について、ゲストＯＳ４０からの取り外し許可を検出したハイパバイザ２０は、アダプタ割り当て表２００を参照し、当該ＶＦ１９０と同じＰＦ１６０から生成された別のＶＦ１９０を代替ＶＦとして割り当て可能かを判断する（１４００）。

【００７５】

代替ＶＦを割り当て可能な場合のみ、ハイパバイザ２０はアダプタ割り当て表２００を更新して、代替ＶＦを当該ゲストＯＳ４０が動いている仮想計算機３０に割り当て直す（１４２０）。続いてハイパバイザ２０は、当該仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５に代替ＶＦの稼働時取り付け（ＨｏｔＡｄｄ）の告知を設定させる。そして、ハイパバイザ２０は仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０に代替ＶＦの増設認識と初期化及び利用開始を促す（１４３０）。これにより、ＨｏｔＰｌｕｇ割り込みを受信したゲストＯＳ４０は、仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５を読み込んで、ＨｏｔＡｄｄが発生したことを取得し、ＰＦ１６０が再生成した他のＶＦ１９０の追加を行う。ゲストＯＳ４０は追加したＶＦ１９０の種類に応じたＶＦドライバ２６０で代替のＶＦ１９０の利用を開始する。

【００７６】

ハイパバイザ２０によるＶＦ交換処理の最後に、当該処理の結果をコンソール８０に出力し、保守要員によるＩ／Ｏアダプタ６０の交換を促す（１４１０）。交換に成功した場合は直ちに交換を行う必要は無いが、Ｉ／Ｏアダプタ６０が経年劣化している可能性を考慮し、予防保守の観点から保守員に通報する。

【００７７】

＜４．まとめ＞

以上の構成及び処理により、障害発生時を除いてＶＦ１９０をゲストＯＳ４０に直接操作させるため、高い性能を実現できる。

【００７８】
ＰＦ１６０で軽度な障害が発生した場合は、当該ＰＦ１６０のリセットによって消失する全てのＶＦ１９０を予め仮想計算機３０からＨｏｔＲｅｍｏｖｅして、ゲストＯＳ４０のクラッシュを回避できるため高信頼を実現できる。また、保守要員の手を借りずにＰＦ１６０を回復させ、再生成されたＶＦ１９０を自動的に仮想計算機３０にＨｏｔＡｄｄするため、Ｉ／Ｏアダプタ６０の保守を容易にすることができる。
【００７９】
また、ＶＦ１９０で軽度及び重度な障害が発生した場合についても、保守要員の手を借りずにハイパバイザ２０が当該ＶＦ１９０を仮想計算機３０からＨｏｔＲｅｍｏｖｅする。そして、ハイパバイザ２０は、代替ＶＦ１９０を割り当て直して仮想計算機３０にＨｏｔＡｄｄする。これにより仮想計算機３０は代替ＶＦ１９０で回復することができるため、ＶＦ１９０の障害に対しても高信頼と保守容易性を実現できる。
【００８０】
＜第２実施形態＞
本発明の第２実施形態では、ハイパバイザ２０がＶＦ１９０を利用してＶＦ１９０の機能を提供するEmulated Function３１０（以下、ＥＦとする）を生成し、仮想計算機３０にＥＦ３１０を割り当てる場合の構成及び動作を説明する。本第２実施形態では、ゲストＯＳ４０では無くハイパバイザ２０がＶＦ１９０を直接操作するため、ＶＦ障害に対してより適切な処理が可能となる。なお、ＥＦ３１０は、ＶＦ１９０と同様にゲストに対して仮想Ｉ／Ｏアダプタとして振る舞う。
【００８１】
＜５．ハードウェア構成＞
本第２の実施形態の計算機システムのハードウェア構成は、前記第１実施形態の図１と同様であるため説明を省略する。
【００８２】
＜６．ソフトウェア構成＞
次に、物理計算機１０上で仮想計算機３０を実現するソフトウェアの構成の主要部と、制御対象となるハードウェア要素について、図１４を参照しながら詳述する。実施形態１に於ける図２と同一の符号の同一構成要素については説明を省略する。
【００８３】
物理計算機１０は、前記第１の実施形態と同一である。ハイパバイザ２０は、任意のＶＦ１９０を用いてＥＦ３１０を生成し、任意の仮想計算機３０にＥＦ３１０を割り当てる。ＥＦ３１０はＶＦ１９０と１対１に対応し、ＶＦ１９０と同じ機能を提供する。ＥＦ３１０は機能を提供するインタフェースがＶＦ１９０と違っていても構わない。ハイパバイザ２０は、ＶＦ１９０を使用するためにＶＦ１９０の種類に応じたＶＦドライバ２６０（ＶＦ２６０－１～２６０－ｍ）を保持する。またハイパバイザ２０は、ＥＦ３１０を実現するために、仮想計算機３０のエミュレーションデータ２２０の内部にＥＦデータ２２４を保持する。
【００８４】
仮想計算機３０は、ハイパバイザ２０によって提供される仮想ＣｈｉｐＳｅｔ３００等の仮想的な部品と、ＥＦ３１０を含む。仮想計算機３０の上ではゲストＯＳ４０が動作する。ゲストＯＳ４０は、ＥＦ３１０に対応する仮想アダプタドライバ２７０を用いてＥＦ３１０を操作する。
【００８５】
図１５はハイパバイザ２０が管理するメモリ９０の一例を示す。前記第１実施形態に於ける図３と同一の構成については説明を省略する。 各仮想計算機３０の領域には、ゲストＯＳ４０と仮想アダプタドライバ２７０が格納される。
【００８６】
ハイパバイザ２０が使用する領域には、アダプタ割り当て表２００、障害処理部２１０、仮想計算機のエミュレーションデータ２２０と、ＰＦドライバ２５０に加えて、ＶＦド

ライバ２６０が格納される。
【００８７】
アダプタ割り当て表２００の構成は、前記第１実施形態の図４と同一であるため説明を省略する。また、ＰＦ状態表２１４の構成は、前記第１実施形態の図５と同一であるため説明を省略する。
【００８８】
ＶＦ状態表２１６の構成は、ＶＦ１９０とＥＦ３１０は１対１で対応するため、ＥＦ３１０にはＶＦ１９０と同じ通し番号を付与する。よって本第２実施形態では、前記第１実施形態の図６に示したＶＦ＃４１０は仮想アダプタ＃の意味を兼ねる。
【００８９】
＜７．ハイパバイザによる処理＞
次に、ハイパバイザ２０が行う処理の一例について、以下、フローチャートを参照しながら説明する。
【００９０】
＜７．１．ハイパバイザによる処理の概要＞
ハイパバイザ２０が行う処理の全体像を示すフローチャートは、前記第１実施形態の図７と同一であるため説明を省略する。
【００９１】
＜７．２．Ｉ／Ｏ障害処理＞
図７のステップ７９０で行われるＩ／Ｏ障害処理は、前記第１実施形態の図８の処理と同一であるため説明を省略する。
【００９２】
＜７．２．１．ＰＦ障害処理＞
図１６は、図８のステップ８３０で行われるＰＦ障害処理のフローチャートの例である。前記第１実施形態の図９と同一符号の同一処理については説明を省略する。
【００９３】
本第２の実施形態では前記第１実施形態で示した（１）重度障害及び（２）軽度障害のケースで更に、ハイパバイザ２０がアダプタ割り当て表２００を参照し、障害が起きたＰＦ１６０から生成された全てのＶＦ１９０と、各ＶＦ１９０と１対１に対応するＥＦ３１０を割り当てた仮想計算機３０を特定する（１７４０）。
【００９４】
続いてハイパバイザ２０は、各仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５にＥＦ３１０の稼働時取り外し（Ｈｏｔ　Ｒｅｍｏｖｅ）要求を設定させる。ハイパバイザ２０は、仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０にＥＦ３１０の安全な取り外し（Ｓａｆｅｌｙ　Ｒｅｍｏｖｅ）を指令する（１７５０）。
【００９５】
＜７．２．２．ＶＦ障害処理＞
図１７は、図８のステップ８４０で行われるＶＦ障害処理のフローチャートの例である。なお、以下では前記第１実施形態の図１０と同一符号の同一処理については説明を省略する。
【００９６】
障害が発生したＶＦ１９０を特定したハイパバイザ２０は、当該ＶＦ１９０の種類に応じたＶＦドライバ２６０に、障害からの回復が可能か否かを問い合わせ、障害の重度を取得する（１８００）。Ｉ／Ｏアダプタ６０の障害の程度には前記第１実施形態で述べたとおり３通りの重度があり、重い障害から順に、（１）回復不能でＶＦ１９０の交換が必要となる重度障害、（２）ＶＦ１９０をリセットすれば回復できる可能性のある軽度障害、（３）ＶＦ１９０のレジスタの一部を再設定する等の手段で回復を完了した障害に分類される。
【００９７】

ハイパバイザ２０は、Ｉ／Ｏアダプタ６０の障害の重度に応じて処理を分岐させる（１８１０）。（１）回復不能な重度障害であれば、ＶＦ状態表２１６の該当するＶＦ１９０のエントリの状態６２０を、「交換保留」に変更する（１８３０）。（２）リセットで回復できる可能性がある軽度障害であれば、ＶＦ状態表２１６の該当するＶＦ１９０のエントリの状態６２０を、「リセット保留」に変更する（１８６０）。（３）回復完了した障害であれば、ハイパバイザ２０に回復処理は省略する。

【００９８】

上記障害の程度が（１）及び（２）のケースでは更に、ハイパバイザ２０がアダプタ割り当て表２００を参照し、障害が起きたＶＦ１９０と１対１に対応するＥＦ３１０と、当該ＥＦ３１０を割り当てた仮想計算機３０を特定する（１８４０）。

【００９９】

続いてハイパバイザ２０は、各仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５にＥＦ３１０の稼働時取り外し（Ｈｏｔ　Ｒｅｍｏｖｅ）要求を設定させる。そして、ハイパバイザ２０は、仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０にＥＦ３１０の利用中止と安全な取り外し（Ｓａｆｅｌｙ　Ｒｅｍｏｖｅ）を指令する（１８５０）。

【０１００】

＜７．３．Ｉ／Ｏ仮想化処理＞

図１８は、前記第１実施形態の図７のステップ７８０で行われるＩ／Ｏ仮想化処理のフローチャートの例である。以下では、前記第１実施形態の図１１と同一符号の同一処理については説明を省略する。

【０１０１】

ゲストによるＩ／Ｏ操作を検出したハイパバイザ２０は、操作対象（仮想ＣｈｉｐＳｅｔ３００等のレジスタ）及び操作内容（読み出し、書き込み）に応じて従来のＩ／Ｏエミュレーション処理を行う（１９００）。この際、ＥＦ３１０のインタフェースが操作された場合は、ハイパバイザ２０が、ＶＦ１９０の同一機能のインタフェースを代行操作する。

【０１０２】

続いてハイパバイザ２０は、ゲストによるＩ／Ｏ操作の内容がＥＦ３１０の取り外し（Ｈｏｔ　Ｒｅｍｏｖｅ）許可であったか否かを判定する（１９１０）。ゲストによるＩ／Ｏ操作の内容がＶＦ１９０の取り外し許可であった場合のみ、ハイパバイザ２０は、ＰＦ状態表２１４を参照し、当該ＶＦ１９０を生成したＰＦ１６０がリセット保留中であるか否かを判断する（１１２０）。

【０１０３】

ステップ１１２０に於いてＰＦ１６０がリセット保留中でなかった場合は、ハイパバイザ２０がＶＦ状態表２１６を参照し、取り外し許可が得られたＥＦ３１０と１対１で対応するＶＦ１９０がリセット保留中であるか否かを判定する（１９６０）。当該ＶＦ１９０がリセット保留中であった場合のみ、１９７０に進んでＶＦ回復処理を行う。

【０１０４】

＜７．３．１．ＰＦ回復処理＞

図１９は、図１８のステップ１１５０で行われるＰＦ回復処理のフローチャートの例である。以下では、前記第１実施形態の図１２と同一符号、同一処理については説明を省略する。

【０１０５】

ＰＦ１６０の回復に成功した場合は、ハイパバイザ２０がＰＦドライバ２５０にＰＦ１６０の再初期化を依頼し、ＩＯＶ機能を再度有効化してＶＦ１９０を再生成させる（１２３０）。続いてハイパバイザ２０は、アダプタ割り当て表２００を参照して、当該ＰＦ１６０から生成されていた全てのＶＦ１９０について、各ＶＦ１９０と１対１に対応するＥＦ３１０が障害発生前にどの仮想計算機３０に割り当てられていたかを取得する。そして、ハイパバイザ２０は、ＥＦ３１０に対応する仮想計算機３０にＥＦ３１０を再初期化さ

せる（２０３５）。更にハイパバイザ２０は、当該仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５にＥＦ３１０の稼働時取り付け（Ｈｏｔ　Ａｄｄ）告知を設定する。そして、ハイパバイザ２０は、仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０にＥＦ３１０の増設認識と初期化及び利用開始を促す（２０４０）。

【０１０６】

＜７．３．２．ＶＦ回復処理＞

図２０は、図１８のステップ１９７０で行われるＶＦ回復処理のフローチャートの例である。リセット保留中のＶＦ１９０と１対１に対応するＥＦ３１０について、ゲストＯＳ４０からの取り外し許可を検出したハイパバイザ２０は、当該ＶＦ１９０をリセットする（２１００）。

【０１０７】

続いてハイパバイザ２０は、ＶＦ１９０の種類に応じたＶＦドライバ２６０を呼び出し、ＶＦ１９０の回復に成功したか否かをＶＦドライバ２６０に問い合わせる（２１１０）。ハイパバイザ２０は、ＶＦ１９０の回復の成否に応じて処理を分岐する（２１２０）。

【０１０８】

ＶＦ１９０の回復に成功した場合は、ハイパバイザ２０が当該ＥＦ３１０を再初期化するよう仮想計算機３０に指令する（２１３０）。続いてハイパバイザ２０は、当該ＥＦ３１０を有する仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５にＥＦ３１０の稼働時取り付け（Ｈｏｔ　Ａｄｄ）告知を設定する。そして、ハイパバイザ２０は仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０にＥＦ３１０の増設認識と初期化及び利用開始を促す（２１４０）。これにより、ＨｏｔＰｌｕｇ割り込みを受信したゲストＯＳ４０は、仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５を読み込んで、Ｈｏｔ　Ａｄｄが発生したことを取得し、初期化されたＥＦ３１０を加え、仮想アダプタドライバ２７０でＥＦ３１０の利用を再開する。

【０１０９】

続いてハイパバイザ２０は、ＶＦ状態表２１６の当該ＶＦ１９０に対応するエントリの状態６２０を正常状態に戻す（２１５０）。最後にハイパバイザ２０は、当該処理の結果をコンソール８０に出力し、保守要員によるＩ／Ｏアダプタ６０の交換を促す（２１６０）。回復に成功した場合は直ちに交換を行う必要は無いが、Ｉ／Ｏアダプタ６０が経年劣化している可能性を考慮し、予防保守の観点から保守員に通報する。

【０１１０】

一方、ステップ２１２０でＶＦ１９０の回復に失敗した場合は、ハイパバイザ２０がＶＦ状態表２１６の障害ＶＦのエントリの状態６２０を交換保留に変更する（２１７０）。続いてハイパバイザはＶＦ交換処理を行う（２１８０）。

【０１１１】

＜７．３．３．ＶＦ交換処理＞

図２１は、図１８のステップ１１９０及び図２０のステップ２１８０で行うＶＦ回復処理のフローチャートの例である。以下では、前記第１実施形態の図１３と同一符号、同一処理については説明を省略する。

【０１１２】

ステップ１４２０で代替ＶＦを仮想計算機３０に割り当て直したのち、ハイパバイザ２０は、代替ＶＦと１対１に対応する代替ＥＦ３１０を初期化する（２２２５）。更にハイパバイザは、当該仮想計算機３０に含まれる仮想ＣｈｉｐＳｅｔ３００のＨＰレジスタ１８５に代替ＥＦ３１０の稼働時取り付け（Ｈｏｔ　Ａｄｄ）告知を設定する。そして、ハイパバイザ２０は、仮想計算機３０にＨｏｔＰｌｕｇ割り込みを通知して、ゲストＯＳ４０に代替ＥＦ３１０の増設認識と初期化及び利用開始を促す（１４３０）。

【０１１３】

＜８．まとめ＞

以上の構成及び処理により、ハイパバイザ２０の処理がＥＦ３１０とＶＦ１９０のイン

タフェース変換のみとなるため、前記第１実施形態には劣るが、比較的高い性能を実現できる。

【０１１４】

ＰＦ１６０で軽度な障害が発生した場合は、当該ＰＦ１６０のリセットによって消失する全てのＶＦ１９０を予め仮想計算機３０からＨｏｔ　Ｒｅｍｏｖｅしておき、ゲストＯＳ４０のクラッシュを回避できるため高信頼を実現できる。また、保守要員の手を借りずにＰＦ１６０を回復させ、再生成されたＶＦ１９０を自動的に仮想計算機３０へＨｏｔＡｄｄするため、保守容易性を実現できる。

【０１１５】

また、ＶＦ１９０で軽度及び重度な障害が発生した場合についても、保守要員の手を借りずに当該ＶＦ１９０を仮想計算機３０からＨｏｔＲｅｍｏｖｅし、障害の重度に応じてＶＦ１９０をリセットまたは交換する対応をとることができる。そして、ＥＦ３１０を仮想計算機３０にＨｏｔＡｄｄして仮想計算機を回復させるため、ＶＦ１９０の障害に対しても高信頼性と保守容易性を実現できる。

【０１１６】

なお、上記各実施形態では、物理計算機１０上に仮想計算機を生成する仮想化部としてハイパバイザを適用した例を示したが、ＶＭＭ（Virtual Machine Monitor）を用いることができる。

【０１１７】

また、上記各実施形態では、物理計算機１０にチップセット１００がＣＰＵ７０から独立して存在する例を示したが、ＣＰＵ７０内にチップセット１００の機能を含めるようにしてもよい。

【産業上の利用可能性】

【０１１８】

以上のように、本発明は、ＩＯＶ機能とＡＥＲ機能を備えたＩ／Ｏアダプタを備えた仮想計算機や仮想計算機の管理システムに適用することができる。

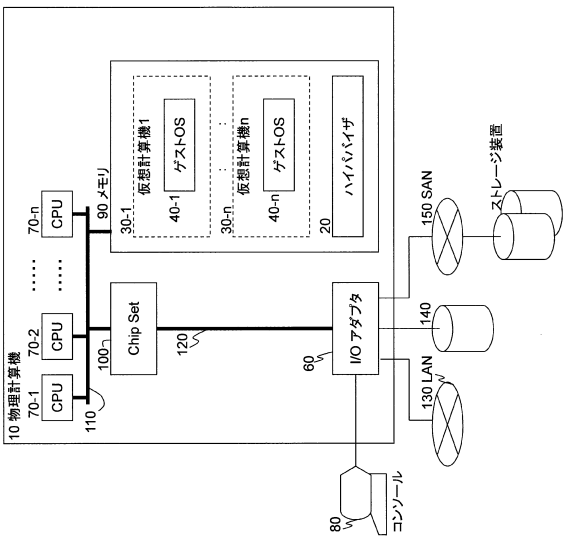
【符号の説明】

【０１１９】

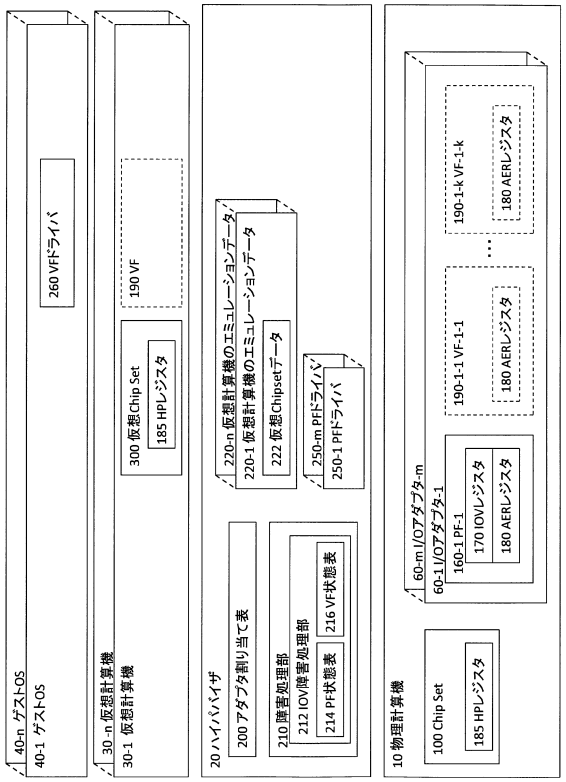
１０　物理計算機
２０　ハイパバイザ
３０　仮想計算機
４０　ゲストＯＳ
６０　Ｉ／Ｏアダプタ
７０　ＣＰＵ
８０　コンソール
９０　メモリ
１００　チップセット
１１０　インターコネクト
１２０　バス
１３０　ＬＡＮ
１４０　ディスク
１５０　ＳＡＮ
１６０　ＰＦ
１７０　ＩＯＶレジスタ
１８０　ＡＥＲレジスタ
１８５　ＨＰレジスタ（ＨｏｔＰｌｕｇレジスタ）
１９０　ＶＦ
２００　アダプタ割り当て表
２１０　障害処理部
２１２　ＩＯＶ障害処理部

２１４　ＰＦ状態表
２１６　ＶＦ状態表
２２０　仮想計算機のエミュレーションデータ
２２２　仮想Ｃｈｉｐｓｅｔデータ
２２４　仮想アダプタデータ
２５０　ＰＦドライバ
２６０　ＶＦドライバ
２７０　仮想アダプタドライバ
３００　仮想ＣｈｉｐＳｅｔ
３１０　仮想アダプタ

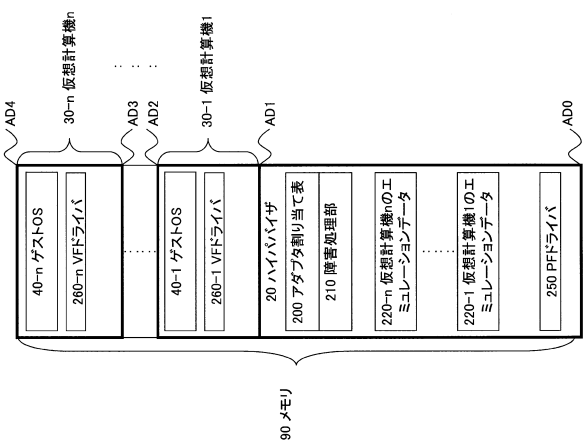
【図１】



【図２】



【図３】



【図４】

PF#	VF#	仮想計算機番号
PF1	VF1-1	0
PF1	VF1-2	1
：	：	：
PF8	VF8-8	未割り当て

400 410 420

【図５】

PF#	状態	520 待ち合わせbitmap
PF1	リセット保留	00000011
PF2	正常	00000000
PF3	交換保留	00010000
：	：	：
PF8	正常	00000000

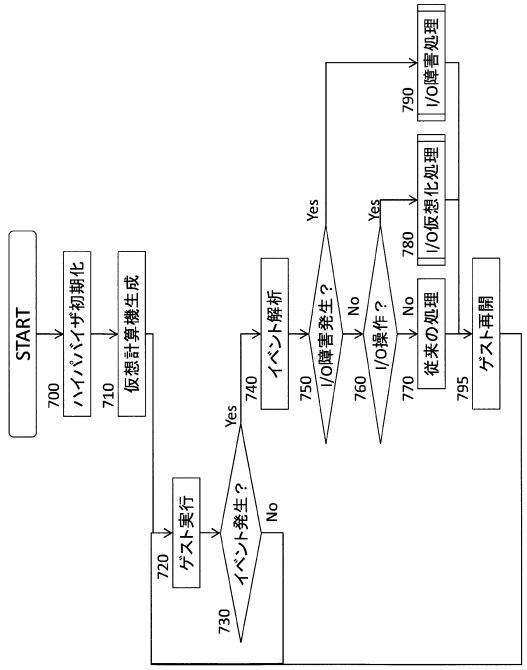
400 510 520

【図６】

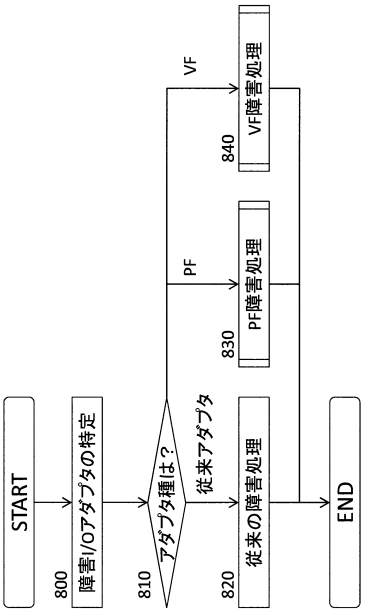
PF#	VF#	状態
PF1	VF1-1	リセット保留
PF1	VF1-2	正常
PF1	VF1-3	交換保留
：	：	：
PF8	VF8-8	正常

400 410 620

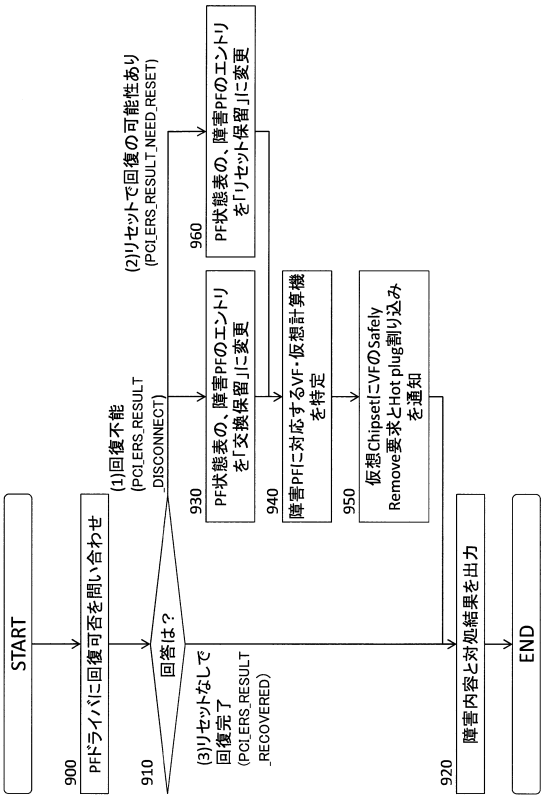
【図７】


START
700
ハイパバイザ初期化
710
仮想計算機生成
720
ゲスト実行
730
イベント発生？
Yes
No
740
イベント解析
750
I/O障害発生？
Yes
No
760
I/O操作？
Yes
No
770
従来の処理
780
I/O仮想化処理
790
I/O障害処理
795
ゲスト再開

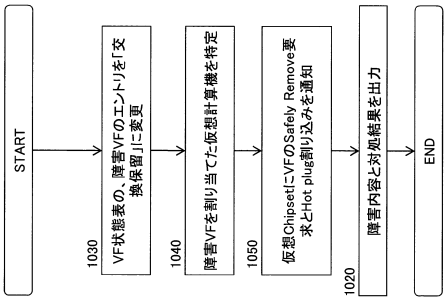
【図８】


START
800
障害I/Oアダプタの特定
810
アダプタ種は？
PF
VF
従来アダプタ
820
従来の障害処理
830
PF障害処理
840
VF障害処理
END

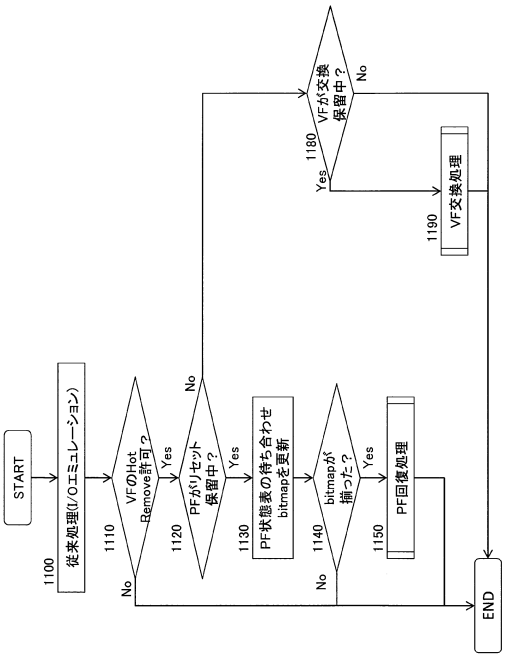
【図９】


START
900
PFドライバに回復可否を問い合わせ
910
回答は？
(1)回復不能
(PCI_ERS_RESULT
_DISCONNECT)
(2)リセットで回復の可能性あり
(PCI_ERS_RESULT_NEED_RESET)
(3)リセットなしで
回復完了
(PCI_ERS_RESULT
_RECOVERED)
930
PF状態表の、障害PFのエントリ
を「交換保留」に変更
940
障害PFに対応するVF・仮想計算機
を特定
950
仮想ChipsetにVFのSafely
Remove要求とHot plug割り込み
を通知
960
PF状態表の、障害PFのエントリ
を「リセット保留」に変更
920
障害内容と対処結果を出力
END

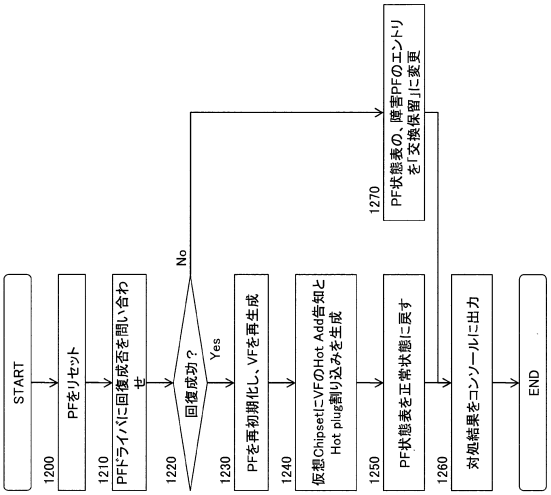
【図１０】


START
1030
VF状態表の、障害VFのエントリを「交
換保留」に変更
1040
障害VFを割り当てた仮想計算機を特定
1050
仮想ChipsetにVFのSafely Remove要
求とHot plug割り込みを通知
1020
障害内容と対処結果を出力
END

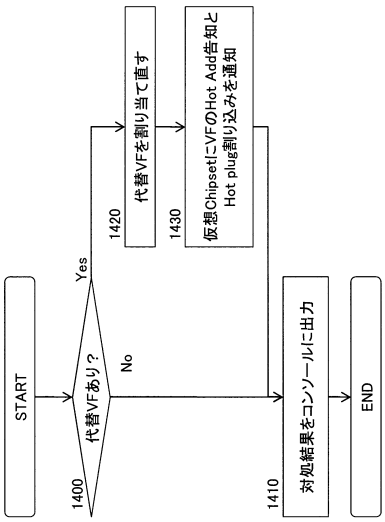
【図１１】



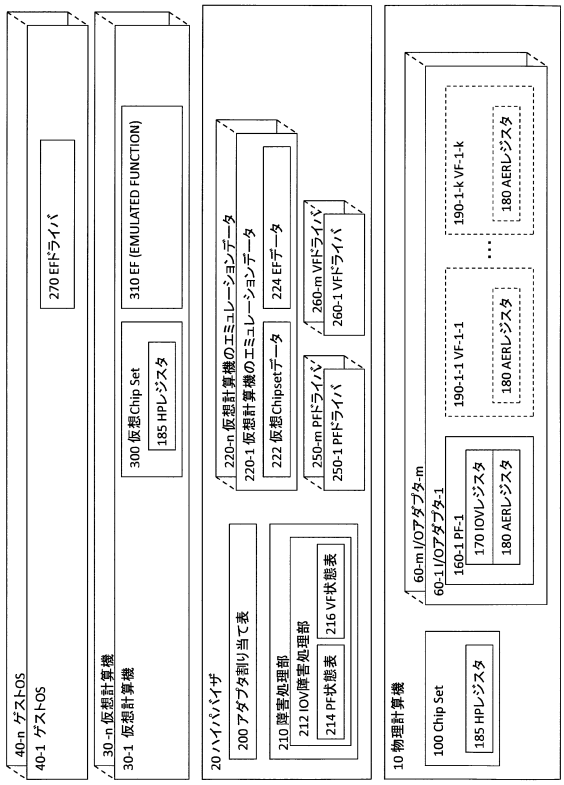
【図１２】



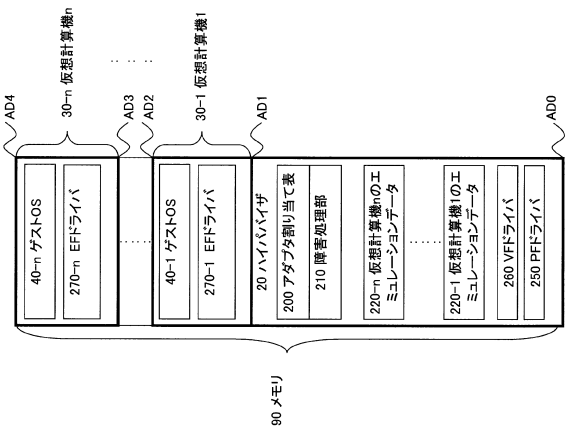
【図１３】



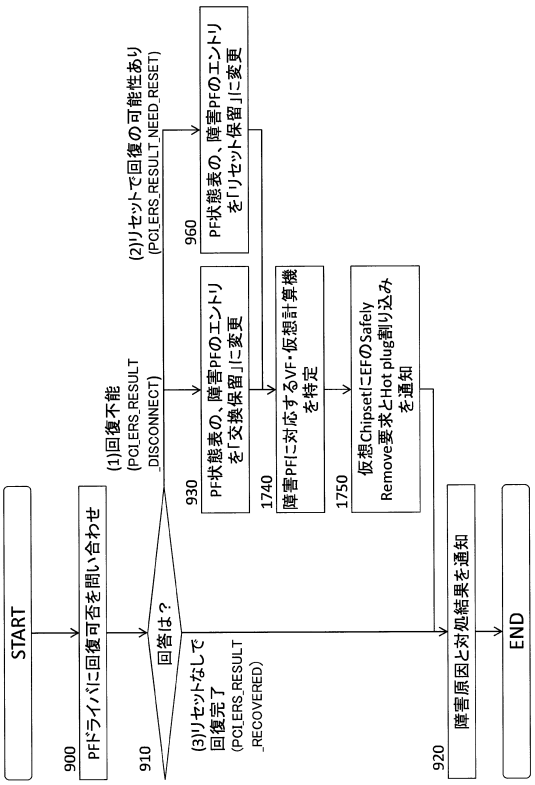
【図１４】



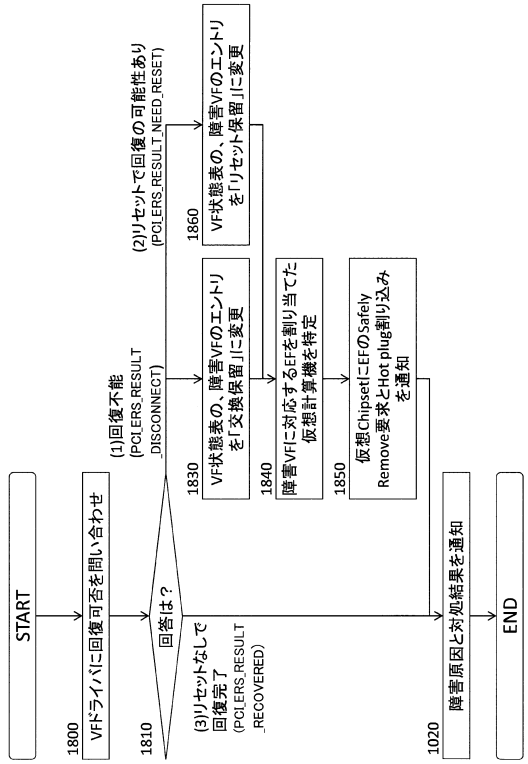
【図１５】



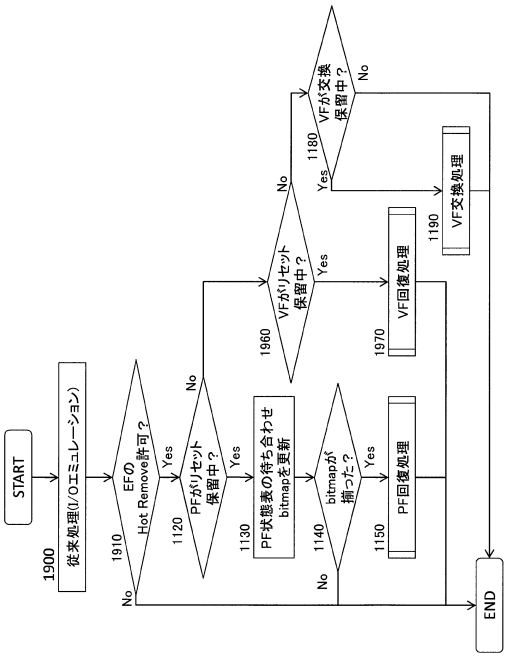
【図１６】



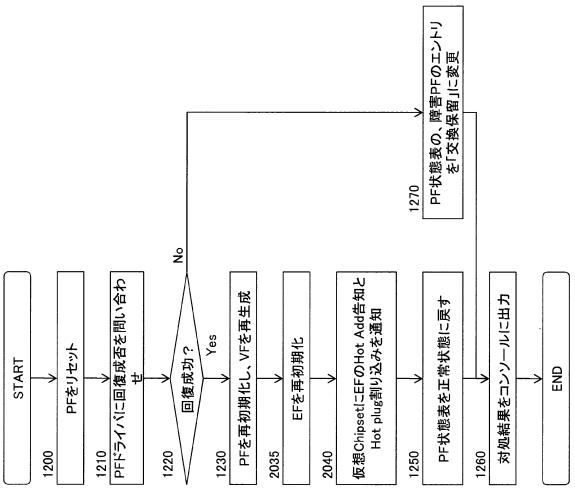
【図１７】



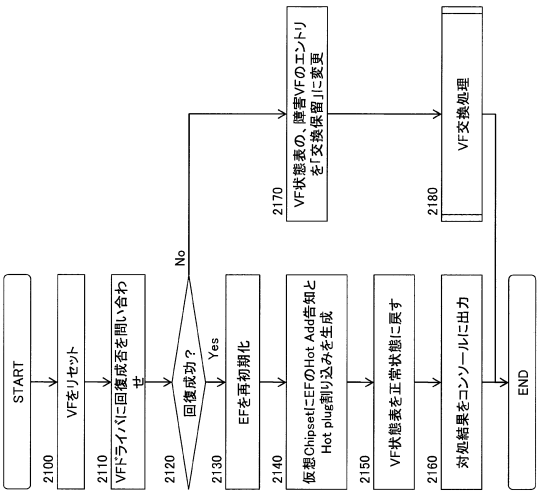
【図１８】



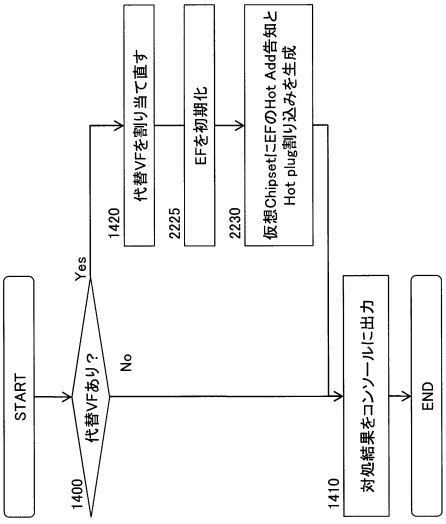
【図１９】


START
1200
PFをリセット
1210
PFドライバに回復成否を問い合わせ
1220
回復成功？
No
Yes
1270
PF状態表の、障害PFのエントリを「交換保留」に変更
1230
PFを再初期化し、VFを再生成
2035
EFを再初期化
2040
仮想ChipsetにEFのHot Add告知とHot plug割り込みを通知
1250
PF状態表を正常状態に戻す
1260
対処結果をコンソールに出力
END

【図２０】


START
2100
VFをリセット
2110
VFドライバに回復成否を問い合わせ
2120
回復成功？
No
Yes
2170
VF状態表の、障害VFのエントリを「交換保留」に変更
2130
EFを再初期化
2140
仮想ChipsetにEFのHot Add告知とHot plug割り込みを生成
2150
VF状態表を正常状態に戻す
2180
VF交換処理
2160
対処結果をコンソールに出力
END

【図２１】


START
1400
代替VFあり？
Yes
No
1420
代替VFを割り当て直す
2225
EFを初期化
2230
仮想ChipsetにEFのHot Add告知とHot plug割り込みを生成
1410
対処結果をコンソールに出力
END

フロントページの続き

(72)発明者　上原　敬太郎
東京都国分寺市東恋ヶ窪一丁目２８０番地　株式会社日立製作所　中央研究所内

審査官　井上　宏一

(56)参考文献　特開２００９－３０１１６２（ＪＰ，Ａ）
特開２０１０－１２８９１１（ＪＰ，Ａ）
特開２００８－１４６５６６（ＪＰ，Ａ）
特開２０１０－　３９７２９（ＪＰ，Ａ）
特表２０１３－５４６１１１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１３／１０
Ｇ０６Ｆ　　９／４６　－９／５４