(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3887314号
(P3887314)
(45)発行日　平成19年2月28日(2007.2.28)　(24)登録日　平成18年12月1日(2006.12.1)

(51)Int.Cl.　F I
G06F 9/46 (2006.01)　G06F 9/46 350
G06F 1/26 (2006.01)　G06F 1/00 334G

請求項の数 14　(全 20 頁)

(21)出願番号　特願2002-570071(P2002-570071)
(86)(22)出願日　平成14年2月27日(2002.2.27)
(65)公表番号　特表2004-523048(P2004-523048A)
(43)公表日　平成16年7月29日(2004.7.29)
(86)国際出願番号　PCT/GB2002/000867
(87)国際公開番号　WO2002/071215
(87)国際公開日　平成14年9月12日(2002.9.12)
審査請求日　平成15年10月3日(2003.10.3)
(31)優先権主張番号　09/798,167
(32)優先日　平成13年3月1日(2001.3.1)
(33)優先権主張国　米国(US)

(73)特許権者　390009531
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
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
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
(73)特許権者　503314532
アイビーエム　ユナイティット　キングダム　リミティット
イギリス国ハンプシャ　ピーオー６　３エーユー　ポーツマス　ノースハーバー　ピーオーボックス４１
最終頁に続く

(54)【発明の名称】データ処理システム中の論理パーティションの電源を切る、または論理パーティションをリブートする、あるいはその両方の方法および装置

(57)【特許請求の範囲】
【請求項１】
複数の論理パーティションを有するデータ処理システムが電源を制御する方法であって、
前記データ処理システム中の前記複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあるかどうかを判定するステップと、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがないと判定されたことに応答して、前記データ処理システム全体の電源を切るステップと、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあると判定されたことに応答して、前記論理パーティションをシャットダウンするステップと
を含む方法。
【請求項２】
複数の論理パーティションと各論理パーティションを管理するハイパーバイザとを有するデータ処理システムが電源を制御する方法であって、
前記ハイパーバイザが前記データ処理システム中の前記複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、前記データ処理

システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあるかどうかを前記ハイパーバイザが判定するステップと、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがないと判定されたことに応答して、前記ハイパーバイザが前記データ処理システム全体の電源を切るステップと、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあると判定されたことに応答して、前記ハイパーバイザが前記論理パーティションをシャットダウンするステップと
を含む方法。
【請求項３】
複数の論理パーティションと各論理パーティションを管理するハイパーバイザとを有するデータ処理システムが電源を制御する方法であって、
前記ハイパーバイザは各論理パーティションの管理プロファイルを格納する不揮発性メモリと前記不揮発性メモリにアクセス可能なサービス・プロセッサとを含み、
前記ハイパーバイザが前記データ処理システム中の前記複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあるかどうかを前記サービス・プロセッサが前記不揮発性メモリ内の管理プロファイルを参照することで判定するステップと、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがないと判定されたことに応答して、前記サービス・プロセッサが前記不揮発性メモリ内の管理プロファイルを更新することで前記前記データ処理システム全体の電源を切るステップと、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあると判定されたことに応答して、前記サービス・プロセッサが前記不揮発性メモリ内の管理プロファイルを更新することで前記論理パーティションをシャットダウンするステップと
を含む方法。
【請求項４】
前記シャットダウンするステップは、前記論理パーティションに割り当てられたプロセッサをリセットするステップを含む請求項１ないし３のいずれかに記載の方法。
【請求項５】
前記データ処理システム中で物理的な電源スイッチを使用不可にするステップをさらに含む請求項１ないし３のいずれかに記載の方法。
【請求項６】
前記要求は遠隔の端末から受け取られる請求項１ないし３のいずれかに記載の方法。
【請求項７】
前記要求は前記データ処理システム中のサービス・プロセッサによって受け取られる請求項１ないし３のいずれかに記載の方法。
【請求項８】
前記要求はハイパーバイザによって受け取られる請求項１ないし３のいずれかに記載の方法。
【請求項９】
前記電源を切るステップおよびシャットダウンするステップは前記ハイパーバイザによって行われる請求項１ないし８のいずれかに記載の方法。
【請求項１０】
前記複数の論理パーティション中のある論理パーティションをリブートする要求を受け取るステップと、
前記論理パーティションに割り当てられた各プロセッサに対してのみリセット信号をアクティブにするステップと

をさらに含む請求項１ないし３のいずれかに記載の方法。
【請求項１１】
前記データ処理システム中のリセット・スイッチを使用不可にするステップをさらに含む請求項１０に記載の方法。
【請求項１２】
前記受け取るステップおよびアクティブにするステップは、前記データ処理システム中のサービス・プロセッサによって行われる請求項１０または１１に記載の方法。
【請求項１３】
複数の論理パーティションを有し、電源を制御するデータ処理システムであって、
前記データ処理システム中の前記複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあるかどうかを判定する判定手段と、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがないと判定されたことに応答して、前記データ処理システム全体の電源を切る電源オフ手段と、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあると判定されたことに応答して、前記論理パーティションをシャットダウンするシャットダウン手段と
を含むデータ処理システム。
【請求項１４】
複数の論理パーティションを有するデータ処理システムの電源を制御するコンピュータ・プログラムであって、前記コンピュータ・プログラムは、前記データ処理システムで実行されると前記データ処理システムに、
前記データ処理システム中の前記複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあるかどうかを判定するステップと、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがないと判定されたことに応答して、前記データ処理システム全体の電源を切るステップと、
前記データ処理システム中で、前記複数の論理パーティションの中に別のアクティブなパーティションがあると判定されたことに応答して、前記論理パーティションをシャットダウンするステップと
を行わせるコンピュータ・プログラム命令を含むコンピュータ・プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般には、改良されたデータ処理システムに関し、詳細には、ネットワーク・データ処理システム内でデータを管理する方法および装置に関する。
【背景技術】
【０００２】
データ処理システム（プラットフォーム）内の論理パーティショニング・オプション（ＬＰＡＲ）は、単一のオペレーティング・システム（ＯＳ）の複数のコピー、または複数の異種のオペレーティング・システムを単一のデータ処理システム・プラットフォーム上で同時に実行することを可能にする。オペレーティング・システムのイメージが実行されるパーティションには、プラットフォーム・リソースの重複しないサブセットが割り当てられる。これらのプラットフォームが割り振ることが可能なリソースには、各自の割込み管理領域を備える１つまたは複数のアーキテクチャ上別個のプロセッサ、システム・メモリの領域、およびＩ／Ｏアダプタのバス・スロットが含まれる。パーティションのリソー

スは、ＯＳイメージに対してはそのパーティションのオープン・ファームウェア・デバイス・ツリーによって表される。

【０００３】

プラットフォーム内で実行される別個の各ＯＳ、またはＯＳのイメージは、１つの論理パーティションにおけるソフトウェア・エラーが、他のどのパーティションの適正な動作にも影響を与えないように互いから保護される。これは、各ＯＳイメージによって直接管理するプラットフォーム・リソースの離散したセットを割り振り、また、各種イメージが、そのイメージに割り振られていないリソースは一切制御できないようにする機構を提供することによって実現される。さらに、あるＯＳに割り振られたリソースの制御下にあるソフトウェア・エラーが、他のイメージのリソースに影響を及ぼさないようにする。このように、ＯＳの各イメージ（または異なる各ＯＳ）が、プラットフォーム中の割り振り可能なリソースの別個のセットを直接制御する。

【０００４】

これらの異なるパーティションの構成は、通例、ハードウェア・システム・コンソール（ＨＳＣ）などの端末を通じて管理される。それらの端末は、ＨＳＣで定義され修正されるプロファイルとも称するオブジェクトを使用する。プロファイルは、データ処理システム中でＬＰＡＲを設定するために使用される。複数のＨＳＣが存在することが可能であり、データ処理システム中でＬＰＡＲを維持し、設定するために使用することができる。ＬＰＡＲでデータ処理システムを設定するために使用されるこのプロファイルは、しばしば、そのデータ処理システムと通信するどのＨＳＣからもアクセス可能であることが必要とされる。それらのＨＳＣ間でプロファイルを維持することはしばしば難しく、各ＨＳＣのプロファイルの同期を維持するプロセスが必要とされる。

【発明の開示】

【発明が解決しようとする課題】

【０００５】

したがって、異なるＨＳＣのプロファイルを維持する改良された方法、装置、およびコンピュータ実施命令を有すると有利である。

【０００６】

同時に複数のパーティションを実行すると、データ処理システムをリセットするコマンドによりすべてのパーティションがリセットされる。同様に、データ処理システムのリセット・ボタンを押すことによっても、すべてのパーティションがリセットされる。さらに、システムの電源を切ると、すべての論理パーティションが適切にシャットダウンされていない場合にはエラーが生じる場合がある。こうした状況があるため、論理パーティションを使用したシステムでは、コンピュータに備わる電源およびリセット用の物理的なボタンは使用しない方がよい。

【０００７】

したがって、データ処理システムをリセットする、またはデータ処理システムの電源を切る、あるいはその両方のための改良された方法および装置を有すると有利であると思われる。

【課題を解決するための手段】

【０００８】

本発明は、複数の論理パーティションを有するデータ処理システム中で電源を制御する方法、装置、およびコンピュータ実施命令を提供する。データ処理システム中の複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、データ処理システム内で、複数の論理パーティション内に追加的なパーティションがあるかどうかを判定する。データ処理システム中で、複数の論理パーティション内に追加的な論理パーティションがないと判定されたことに応答してデータ処理システム内で電源を切る。データ処理システム内で、複数の論理パーティション内に追加的なパーティションがあると判定されたことに応答して論理パーティションをシャットダウンする。本発明の機構は、論理パーティションのリブートも提供する。複数の論理パーティション内の

ある論理パーティションをリブートする要求を受け取る。その論理パーティションに割り当てられた各プロセッサに対してのみリセット信号をアクティブにする。

【０００９】

第１の態様によれば、本発明は、複数の論理パーティションを有するデータ処理システム中で電源を制御する方法を提供し、この方法は、データ処理システム中の複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがあるかどうかを判定するステップと、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがないと判定されたことに応答して、データ処理システム内で電源を切るステップと、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがあると判定されたことに応答して、論理パーティションをシャットダウンするステップとを含む。

【００１０】

この方法はさらに、データ処理システム中で物理的な電源スイッチを使用不可にするステップを含むことが好ましい。

【００１１】

要求は、遠隔の端末から受け取ることがより好ましい。あるいは、データ処理システム内のサービス・プロセッサによって要求を受け取る。あるいは、ハイパーバイザによって要求を受け取る。

【００１２】

電源オフおよびシャットダウンのステップは、ハイパーバイザによって行うと適切である。より適切には、この方法は、複数の論理パーティション内のある論理パーティションをリブートする要求を受け取るステップと、その論理パーティションに割り当てられた各プロセッサに対してのみリセット信号をアクティブにするステップとを含む。さらに適切には、この方法はさらに、データ処理システム中でリセット・スイッチを使用不可にするステップを含む。

【００１３】

受け取りおよびアクティブ化のステップは、データ処理システム中のサービス・プロセッサによって行うと適切である。要求をハイパーバイザによって受け取り、サービス・プロセッサに送信してリセット信号をアクティブにするとより適切である。

【００１４】

好ましい実施形態によれば、データ処理システムが提供され、このデータ処理システムは、バス・システムと、バス・システムに接続された通信ユニットと、バス・システムに接続され、命令セットを含むメモリと、バス・システムに接続された処理ユニットとを含み、処理ユニットは、データ処理システム中の複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、命令セットを実行して、データ処理システム中で複数の論理パーティション内に追加的なパーティションがあるかどうかを判定し、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがないと判定されたことに応答してデータ処理システム内で電源を切り、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがあると判定されたことに応答してその論理パーティションをシャットダウンする。

【００１５】

別の好ましい実施形態によれば、データ処理システムが提供され、このデータ処理システムは、バス・システムと、バス・システムに接続された通信ユニットと、バス・システムに接続され、命令セットを含むメモリと、バス・システムに接続された処理ユニットとを含み、処理ユニットは、命令セットを実行して、複数の論理パーティション内のある論理パーティションをリブートする要求を受け取り、その論理パーティションに割り当てられた各プロセッサに対してのみリセット信号をアクティブにする。

【００１６】

第２の態様によれば、本発明は、複数の論理パーティションを有するデータ処理システ

ム中で電源を制御するデータ処理システムを提供し、このデータ処理システムは、データ処理システム中の複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、データ処理システム内で、複数の論理パーティション内に追加的なパーティションがあるかどうかを判定する判定手段と、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがないと判定されたことに応答して、データ処理システム内で電源を切る電源オフ手段と、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがあると判定されたことに応答して、論理パーティションをシャットダウンするシャットダウン手段とを含む。

【００１７】

データ処理システムはさらに、データ処理システム中で物理的な電源スイッチを使用不可にする使用不可手段を含むことが好ましい。要求は、遠隔の端末から受け取ることがより好ましい。あるいは、要求は、データ処理システム中のサービス・プロセッサによって受け取る。あるいは、要求はハイパーバイザによって受け取る。

【００１８】

電源オフ手段およびシャットダウン手段は、ハイパーバイザ中に配置することが好ましい。

【００１９】

好ましい実施形態によれば、データ処理システム中の論理パーティションをリブートするデータ処理システムが提供され、このデータ処理システムは、複数の論理パーティション内のある論理パーティションをリブートする要求を受け取る受け取り手段と、その論理パーティションに割り当てられた各プロセッサに対してのみリセット信号をアクティブにするアクティブ化手段とを備える。

【００２０】

データ処理システムはさらに、データ処理システム中でリセット・スイッチを使用不可にする使用不可手段を備えることが好ましい。受け取り手段およびアクティブ化手段は、データ処理システム中のサービス・プロセッサによって実行される命令セット中に配置することがより好ましい。

【００２１】

要求は、ハイパーバイザによって受け取られ、サービス・プロセッサに送信されてリセット信号をアクティブにする。

【００２２】

第３の態様によれば、本発明は、複数の論理パーティションを有するデータ処理システム内で電源を制御するコンピュータ・プログラム製品を提供し、このコンピュータ・プログラム製品は、コンピュータで実行されるとコンピュータに、データ処理システム中の複数の論理パーティション内のある論理パーティションの電源を切る要求を受け取ったことに応答して、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがあるかどうかを判定するステップと、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがないと判定されたことに応答して、データ処理システム内で電源を切るステップと、データ処理システム中で、複数の論理パーティション内に追加的なパーティションがあると判定されたことに応答して、論理パーティションをシャットダウンするステップとを行わせるコンピュータ・プログラム命令を含む。

【００２３】

以下で、本発明について、以下の図面に示す好ましい実施形態を参照して、単なる例として説明する。

【発明を実施するための最良の形態】

【００２４】

図１に、本発明を実施することが可能な分散データ処理システムの図を示す。

【００２５】

分散データ処理システム１００は、本発明を実施することが可能な、コンピュータのネットワークである。分散データ処理システム１００はネットワーク１０２を含み、ネット

ワーク１０２は、分散データ処理システム１００内で接続された各種の装置およびコンピュータ間に通信リンクを提供するのに使用される媒体である。ネットワーク１０２は、配線または光ファイバー・ケーブルなどの恒久的接続、または電話接続を通じて行われる一時的な接続を含むことができる。

【００２６】

図の例では、サーバ１０４がハードウェア・システム・コンソール１５０に接続されている。サーバ１０４は、記憶ユニット１０６とともにネットワーク１０２にも接続される。またクライアント１０８、１１０および１１２もネットワーク１０２に接続される。これらのクライアント１０８、１１０および１１２は、例えばパーソナル・コンピュータまたはネットワーク・コンピュータである。この応用例の目的では、ネットワーク・コンピュータとは、ネットワークに結合された別のコンピュータからプログラムまたは他のアプリケーションを受信する、ネットワークに結合された任意のコンピュータである。図の例では、サーバ１０４は、論理的にパーティションに区分されたプラットフォームであり、ブート・ファイル、オペレーティング・システム・イメージおよびアプリケーションなどのデータをクライアント１０８～１１２に提供する。ハードウェア・システム・コンソール１５０は、ラップトップ・コンピュータでよく、サーバ１０４で実行される各オペレーティング・システム・イメージから操作者に対するメッセージを表示するのに使用され、また操作者から受け取った入力情報をサーバ１０４に送信するために使用される。クライアント１０８、１１０および１１２はサーバ１０４に対するクライアントである。分散データ処理システム１００は、図にないサーバ、クライアント、および他のデバイスをさらに含むことができる。分散データ処理システム１００は、プリンタ１１４、１１６および１１８も含む。クライアント１１０などのクライアントは、直接プリンタ１１４に印刷を行うことができる。クライアント１０８および１１２などのクライアントは、直接接続されたプリンタを持たない。これらのクライアントは、サーバ１０４に接続されたプリンタ１１６か、または文書を印刷するのにコンピュータとの接続を必要としないネットワーク・プリンタであるプリンタ１１８に印刷を行うことができる。あるいは、クライアント１１０は、プリンタの種類と文書の要件に応じてプリンタ１１６またはプリンタ１１８に印刷を行ってもよい。

【００２７】

図の例では、分散データ処理システム１００はインターネットであり、ネットワーク１０２は、ＴＣＰ／ＩＰプロトコル・スイートを使用して互いと通信するネットワークおよびゲートウェイの世界規模の集合を表す。インターネットの中心となるのは、データおよびメッセージをルーティングする何千もの商用、行政、教育、およびその他のコンピュータ・システムからなる主要ノードまたはホスト・コンピュータ間を結ぶ高速データ通信回線のバックボーンである。言うまでもなく、分散データ処理システム１００は、例えばイントラネットやローカル・エリア・ネットワークなどいくつかの異なるタイプのネットワークとして実施してもよい。

【００２８】

次いで図２を参照すると、本発明によるデータ処理システムのブロック図を示している。データ処理システム２００は、クライアント、または図１に示すハードウェア・システム・コンソール１５０などのハードウェア・システム・コンソールの一例である。データ処理システム２００では、ペリフェラル・コンポーネント相互接続（ＰＣＩ）ローカル・バス・アーキテクチャを用いる。図の例ではＰＣＩバスを使用するが、Micro ChannelやＩＳＡなど他のバス・アーキテクチャを使用してもよい。プロセッサ２０２およびメイン・メモリ２０４は、ＰＣＩブリッジ２０８を通じてＰＣＩローカル・バス２０６に接続される。ＰＣＩブリッジ２０８は、内蔵メモリ・コントローラおよびプロセッサ２０２のキャッシュ・メモリも含むことができる。直接コンポーネント相互接続またはアドイン・ボードにより、ＰＣＩローカル・バス２０６に追加的な接続を行うことができる。図の例では、ローカル・エリア・ネットワーク（ＬＡＮ）アダプタ２１０、ＳＣＳＩホスト・バス・アダプタ２１２、および拡張バス・インターフェース２１４は、直接コンポーネント接

続によりＰＣＩローカル・バス２０６に接続される。これに対して、オーディオ・アダプタ２１６、グラフィクス・アダプタ２１８、およびオーディオ／ビデオ・アダプタ（Ａ／Ｖ）２１９は、拡張スロットに挿入されたアドイン・ボードによってＰＣＩローカル・バス２０６に接続される。拡張バス・インターフェース２１４は、キーボードおよびマウス・アダプタ２２０、モデム２２２、および追加メモリ２２４の接続を提供する。図の例では、ＳＣＳＩホスト・バス・アダプタ２１２は、ハード・ディスク・ドライブ２２６、テープ・ドライブ２２８、ＣＤ－ＲＯＭドライブ２３０、およびデジタル・ビデオ・ディスク読み取り専用メモリ・ドライブ（ＤＶＤ－ＲＯＭ）２３２の接続を提供する。典型的なＰＣＩローカル・バスの実装では、３つまたは４つのＰＣＩ拡張スロットまたはアドイン・コネクタがサポートされる。

【００２９】

オペレーティング・システムは、プロセッサ２０２で実行され、図２のデータ処理システム２００内の各種コンポーネントを調整し、その制御を提供するために使用される。このオペレーティング・システムは、ＯＳ／２など市販のオペレーティング・システムでよい。（ＯＳ／２は、International Business Machines社の登録商標。）Ｊａｖａ（Ｒ）（Ｊａｖａ（Ｒ）はSunMicrosystems社の商標）などオブジェクト指向のプログラミング・システムをオペレーティング・システムと併せて実行し、データ処理システム２００で実行されるＪａｖａ（Ｒ）プログラムまたはアプリケーションからオペレーティング・システムに対する呼出しを提供することができる。オペレーティング・システム、オブジェクト指向のオペレーティング・システム、およびアプリケーションまたはプログラムに対する命令は、ハード・ディスク・ドライブ２２６などの記憶装置に置かれ、プロセッサ２０２による実行のためにメイン・メモリ２０４にロードすることができる。

【００３０】

当業者は、図２のハードウェアは実装に応じて異なりうることを理解されよう。例えば、図２に示すハードウェアに加えて、あるいはその代わりに、光ディスク・ドライブなど他の周辺装置を使用することができる。例えば、本発明のプロセスは、マルチプロセッサのデータ処理システムに適用することができる。

【００３１】

次いで図３を参照すると、本発明による、図１のサーバ１０４など論理的に区分されたサーバとして実装することが可能なデータ処理システムのブロック図を示す。データ処理システム３００は、システム・バス３０６に接続された複数のプロセッサ３０１、３０２、３０３および３０４を含む対称型マルチプロセッサ（ＳＭＰ）システムでよい。例えば、データ処理システム３００には、ＩＢＭ　ＲＳ／６０００を使用することができる（ＲＳ／６０００はInternational Business Machines社の登録商標）。あるいは単一プロセッサ・システムを使用してもよい。システム・バス３０６には、複数のローカル・メモリ３６０～３６３とのインターフェースを提供するメモリ・コントローラ／キャッシュ３０８も接続される。Ｉ／Ｏバス・ブリッジ３１０は、システム・バス３０６に接続され、Ｉ／Ｏバス３１２とのインターフェースを提供する。メモリ・コントローラ／キャッシュ３０８およびＩ／Ｏバス・ブリッジ３１０は、図のように一体化することができる。

【００３２】

データ処理システム３００は、論理的にパーティションに区分されたデータ処理システムである。したがって、データ処理システム３００は、複数の異種のオペレーティング・システム（または単一オペレーティング・システムの複数のインスタンス）を同時に実行することができる。それら複数のオペレーティング・システムはそれぞれ、各自の内部で実行される任意数のソフトウェア・プログラムを有することができる。データ処理システム３００は、異なるＩ／Ｏアダプタ３２０～３２１、３２８～３２９、３３６～３３７、および３４６～３４７を異なる論理パーティションに割り当てることができるように論理的に区分されている。

【００３３】

したがって、例えば、データ処理システム３００が３つの論理パーティションＰ１、Ｐ

２、およびＰ３に分割されているとする。各Ｉ／Ｏアダプタ３２０～３２１、３２８～３２９、および３３６～３３７、各プロセッサ３０１～３０４、および各ローカル・メモリ３６０～３６４は、３つのパーティションの１つに割り当てられる。例えば、プロセッサ３０１、メモリ３６０、およびＩ／Ｏアダプタ３２０、３２８および３２９を論理パーティションＰ１に割り当て、プロセッサ３０２～３０３、メモリ３６１、およびＩ／Ｏアダプタ３２１および３３７をパーティションＰ２に割り当て、プロセッサ３０４、メモリ３６２～３６３、およびＩ／Ｏアダプタ３３６および３４６～３４７を論理パーティションＰ３に割り当てることができる。

【００３４】

データ処理システム３００で実行される各オペレーティング・システムは、異なる論理パーティションに割り当てられる。したがって、データ処理システム３００で実行される各オペレーティング・システムは、自身の論理パーティションにあるＩ／Ｏユニットにしかアクセスすることができない。したがって、例えば、Advanced Interactive Executive（ＡＩＸ）（ＡＩＸは、International BusinessMachines社の登録商標）オペレーティング・システムの１インスタンスをパーティションＰ１で実行し、ＡＩＸオペレーティング・システムの第２のインスタンス（イメージ）をパーティションＰ２で実行し、Ｗｉｎｄｏｗｓ（Ｒ）２０００（Ｗｉｎｄｏｗｓ（Ｒ）２０００はMicrosoft社の商標）オペレーティング・システムを論理パーティションＰ１で作動させることができる。

【００３５】

Ｉ／Ｏバス３１２に接続されたペリフェラル・コンポーネント相互接続（ＰＣＩ）ホスト・ブリッジ３１４は、ＰＣＩローカル・バス３１５とのインターフェースを提供する。ＰＣＩバス３１５にはいくつかのターミナル・ブリッジ３１６～３１７を接続することができる。典型的なＰＣＩバス実装では、拡張スロットまたはアドイン・コネクタを提供するために４つのターミナル・ブリッジがサポートされる。ターミナル・ブリッジ３１６～３１７はそれぞれ、ＰＣＩバス３１８～３１９を通じてＰＣＩ　Ｉ／Ｏアダプタ３２０～３２１に接続される。各Ｉ／Ｏアダプタ３２０～３２１は、データ処理システム３００と、例えばサーバ３００に対するクライアントである他のネットワーク・コンピュータなどの入出力装置とのインターフェースを提供する。ターミナル・ブリッジ３１６～３１７には、それぞれ１つのみのＩ／Ｏアダプタ３２０～３２１を接続することができる。ターミナル・ブリッジ３１６～３１７はそれぞれ、ＰＣＩホスト・ブリッジ３１４そしてより高いレベルのデータ処理システム３００にエラーが伝搬するのを防ぐように構成する。これにより、ターミナル・ブリッジ３１６～３１７のいずれかによって受け取られたエラーが、異なるパーティションにある可能性のある他のＩ／Ｏアダプタ３２１、３２８～３２９、および３３６～３３７の共有バス３１５および３１２から隔離される。したがって、あるパーティションのＩ／Ｏデバイスで発生したエラーは、別のパーティションのＯＳからは「見え」ない。したがって、あるパーティションにあるオペレーティング・システムの保全性は、別の論理パーティションで発生したエラーの影響を受けない。このようなエラーの隔離を行わないと、あるパーティションのＩ／Ｏデバイス内で発生したエラーにより、別のパーティションのオペレーティング・システムまたはアプリケーション・プログラムが動作を中止する、あるいは正常な動作を中止する可能性がある。

【００３６】

追加的なＰＣＩホスト・ブリッジ３２２、３３０および３４０は、追加的なＰＣＩバス３２３、３３１、および３４１とのインターフェースを提供する。追加的なＰＣＩバス３２３、３３１および３４１は各々、それぞれがＰＣＩバス３２６～３２７、３３４～３３５、および３４４～３４５によってＰＣＩ　Ｉ／Ｏアダプタ３２８～３２９、３３６～３３７、および３４６～３４７に接続された複数のターミナル・ブリッジ３２４～３２５、３３２～３３３、および３４２～３４３に接続される。したがって、各ＰＣＩ　Ｉ／Ｏアダプタ３２８～３２９、３３６～３３７、および３４６～３４７を通じて、例えばモデムやネットワーク・アダプタなどの追加的なＩ／Ｏデバイスをサポートすることができる。このように、サーバ３００は複数のネットワーク・コンピュータとの接続を可能にする。

図に示すように、Ｉ／Ｏバス３１２には、メモリマップ・グラフィクス・アダプタ３４８およびハード・ディスク３５０も直接または間接的に接続することができる。

【００３７】

論理パーティションの管理は、ハードウェア・システム・コンソール（ＨＳＣ）などの端末を通じて行われる。この例では、このアクセスは、サービス・プロセッサ３６６、不揮発性ランダム・アクセス・メモリ（ＮＶＲＡＭ）３６８、および入出力（Ｉ／Ｏ）アダプタ３７０を通じて提供される。ＨＳＣは、Ｉ／Ｏアダプタ３７０を通じてサービス・プロセッサ３６６に接続する。ＮＶＲＡＭ３６８は、データ処理システム３００中で論理パーティションを設定および管理するために使用されるプロファイルなどのオブジェクトを含む。これらの例では、ＮＶＲＡＭ３６８に格納されたプロファイルは、ＨＳＣがオンライン状態になる、すなわちＩ／Ｏアダプタ３７０を通じてデータ処理システム３００に接続すると、ＨＳＣに送信される。このアーキテクチャは、論理パーティションのプロファイルをＨＳＣに格納する必要性をなくす機構を提供する。さらに、このアーキテクチャでは、異なるＨＳＣで複製されたプロファイルを保持する同期機構も不要である。

【００３８】

当業者は、図３に示すハードウェアは異なってよいことを理解されよう。例えば、図のハードウェアに加えて、あるいはそれに代えて、光ディスク・ドライブなど他の周辺装置を使用することもできる。

【００３９】

次いで図４を参照すると、本発明を実施することが可能な、論理的にパーティションに区分された例示的プラットフォームのブロック図を示す。論理的に区分されたプラットフォーム５００のハードウェアは、例えば図３のサーバ３００として実装することができる。論理的に区分されたプラットフォーム４００は、区分されたハードウェア４３０、ハイパーバイザ４１０、およびオペレーティング・システム４０２～４０８を含む。オペレーティング・システム４０２～４０８は、プラットフォーム４００で同時に実行される単一オペレーティング・システムの複数のコピーであっても、複数の異種のオペレーティング・システムであってもよい。

【００４０】

区分されたハードウェア４３０は、複数のプロセッサ４３２～４３８、複数のシステム・メモリ・ユニット４４０～４４６、複数の入出力（Ｉ／Ｏ）アダプタ４４８～４６２、および記憶ユニット４７０を含む。プロセッサ４３２～４３８、メモリ・ユニット４４０～４４６、およびＩ／Ｏアダプタ４４８～４６２はそれぞれ、論理的に区分されたプラットフォーム４００中で複数のパーティションの１つに割り当てることができ、各パーティションはオペレーティング・システム４０２～４０８の１つに対応する。

【００４１】

ファームウェアとして実装されるハイパーバイザ４１０は、オペレーティング・システム・イメージ４０２～４０８に対して複数の機能およびサービスを行って、論理的に区分されたプラットフォーム４００のパーティション区分を作成し、実行させる。ファームウェアとは、例えば、読み取り専用メモリ（ＲＯＭ）、プログラマブルＲＯＭ（ＰＲＯＭ）、消去可能プログラマブルＲＯＭ（ＥＰＲＯＭ）、電気的に消去可能なプログラマブルＲＯＭ（ＥＥＰＲＯＭ）、および不揮発性ランダム・アクセス・メモリ（不揮発性ＲＡＭ）など、電力を用いずにその内容を保持するメモリ・チップに格納された「ハード・ソフトウェア」である。

【００４２】

ハイパーバイザ４１０は、例えば図３のＩ／Ｏバス３１２などの共有Ｉ／Ｏバス上の、例えば図３のＩ／Ｏアダプタ３２８などのＩ／Ｏアダプタごとに、例えば図４のＯＳイメージ４０２などそのＩ／Ｏアダプタに割り当てられたＯＳイメージに割り振られたメモリ・リソースへのセキュアなダイレクト・メモリ・アクセス（ＤＭＡ）ウィンドウを提供する。セキュアなＤＭＡウィンドウは、Ｉ／Ｏアダプタから、そのＩ／Ｏアダプタと同じパーティションに割り振られたメモリへのアクセスを提供し、同時にＩ／Ｏアダプタが異な

るパーティションに割り振られたメモリへのアクセス権を得ることを防止する。

【００４３】

一実施形態では、ＲＳ／６０００プラットフォーム・アーキテクチャで実装されるように、ハイパーバイザは既存の２つのハードウェア機構を利用する。これらのハードウェア機構は、変換制御エントリ（ＴＣＥ）機能およびＤＭＡ範囲レジスタ機能ブリッジと呼ばれるものである。一実施形態では、ＴＣＥ機能は、図３のＰＣＩホスト・ブリッジ３１４、３２２、３３０および３４０などのＰＣＩホスト・ブリッジ中で実施され、範囲レジスタ機能は、ターミナル・ブリッジ３１６～３１７、３２４～３２５、３３２～３３３、および３４２～３４３などのターミナル・ブリッジ中で実施される。

【００４４】

ＴＣＥ機能（図示せず）は、今日の大半のプロセッサが備える仮想メモリのアドレス変換機能に似る、Ｉ／Ｏの機能である。すなわち、ＴＣＥ機能は、Ｉ／Ｏバス上の隣接したアドレス空間を、隣接しない可能性のあるメモリ中の別のアドレス空間に変換する機構を提供する。ＴＣＥ機能はこの変換をプロセッサの変換機構と同様の方式で行い、すなわち、メモリのアドレス空間とＩ／Ｏバスのアドレス空間をページと称する小さな領域(chunk)に分割する。ＩＢＭ　ＰｏｗｅｒＰＣ（ＰｏｗｅｒＰＣは、International Business Machines社の登録商標）プロセッサを使用したプラットフォームでは、このサイズは一般には１ページ当たり４キロバイトである。各ページには、変換と制御のエントリが関連付けられる。この変換と制御のエントリをこのＩ／Ｏ変換機構ではＴＣＥと呼び、対応するプロセッサによる仮想変換機構では時にページ・テーブル・エントリと呼ぶ。この変換エントリは、プロセッサとＩ／Ｏとで異なるテーブルにある。

【００４５】

バスでＩ／Ｏ動作が開始すると、ＴＣＥ機能は、ＴＣＥテーブルのそのページのエントリにアクセスし、そのエントリのデータをメモリにアクセスするためのアドレスの最上位ビットとして使用し、最下位ビットはバス上のＩ／Ｏアドレスから得る。バスから使用するビット数はページのサイズによって異なり、そのページ内のバイト・レベルに合わせてアドレス指定するのに必要なビット数である（例えば４キロバイトのページ・サイズの例では、４キロバイトのページでバイト・レベルに合わせてアドレス指定するのに必要とされるビット数であることから、バスから取るビット数は１２になる）。このように、ＴＣＥは、メモリ中のどのページをアドレス指定するかを決定するビットを提供し、Ｉ／Ｏバスから取得するアドレス・ビットによりそのページ内のアドレスが決まる。

【００４６】

Ｉ／ＯアダプタがＩ／Ｏバスに置くことを許されるバス・アドレスの範囲は、範囲レジスタ機能によって制限される。範囲レジスタ機能は、Ｉ／Ｏアダプタがアクセスしようとしているアドレスと比較するアドレスを保持するいくつかのレジスタを含む。比較により、Ｉ／Ｏアダプタが、ファームウェアによって範囲レジスタにプログラムされたアドレス範囲の外側にアクセスしようとしていることが分かった場合、ブリッジはそのＩ／Ｏアダプタに応答せず、Ｉ／Ｏアダプタがアクセスを許可されていないアドレスにアクセスすることを効果的に阻止する。この実施形態では、これら２つのハードウェア機構はハイパーバイザの制御下に置かれる。

【００４７】

プラットフォーム４００を初期化すると、ハイパーバイザ４１０が個々のＩ／Ｏアダプタ４４８～４６２を独占的に使用するために、Ｉ／ＯバスのＤＭＡアドレスの離散した範囲が、Ｉ／Ｏアダプタ４４８～４６２それぞれに割り当てられる。ハイパーバイザ４１０は次いで、ターミナル・ブリッジの範囲レジスタ（図示せず）機能を構成して、この独占的な使用を実施する。ハイパーバイザ４１０は次いで、その割り振りをＯＳイメージ４０２～４０８のうち所有主であるイメージに伝える。ハイパーバイザはまた、特定のＩ／Ｏアダプタに関連付けられたＴＣＥテーブル部分中のすべてのエントリを初期化して、そのＩ／Ｏアダプタに割り振られたＯＳイメージが所有するイメージごとの確保されたページをポイントさせ、Ｉ／Ｏアダプタによるメモリへの無許可アクセスにより、他のＯＳイメ

ージ４０２～４０８の１つに影響を与えうるエラーが生じないようにする。
【００４８】
ＯＳイメージ４０２～４０８のうち所有主であるイメージが、ＤＭＡ動作のために自身のメモリの一部をマッピングすることを要求する際には、Ｉ／Ｏアダプタ、メモリ・アドレス範囲、およびマッピングしようとする、関連付けられたＩ／ＯバスのＤＭＡアドレス範囲を示すパラメータを含めてハイパーバイザ４１０に呼出しを行う。ハイパーバイザ４１０は、そのＩ／Ｏアダプタとメモリ・アドレス範囲が、ＯＳイメージ４０２～４０８のそのイメージに割り振られていることを確認する。ハイパーバイザ４１０は、Ｉ／ＯバスのＤＭＡ範囲が、そのＩ／Ｏアダプタに割り振られた範囲内であることも確認する。これらのチェックに通ると、ハイパーバイザ４１０は要求されるＴＣＥマッピングを行う。これらのチェックに通らない場合は、ハイパーバイザは要求を拒絶する。
【００４９】
ハイパーバイザ４１０は、各々がコンソールおよびオペレータ・パネルの仮想コピーである複数の論理パーティションで実行されるＯＳイメージ４０２～４０８も提供することができる。コンソールとのインターフェースは、従来技術における非同期のテレタイプ・ポート・デバイス・ドライバから、ポート・デバイス・ドライバをエミュレートするハイパーバイザ・ファームウェア呼出しのセットに変更する。ハイパーバイザ４１０は、各種のＯＳイメージからのデータをカプセル化して、ハードウェア・システム・コンソール・コンピュータ４８０などの端末に転送されるメッセージ・ストリームに置く。これらの例では、複数のハードウェア・システム・コンソールがサポートされる。図のように、ハードウェア・システム・コンソール４８２およびハードウェア・システム・コンソール４８４も存在する。
【００５０】
ハードウェア・システム・コンソール４８０～４８４は、図４に示すように論理的に区分されたプラットフォーム４００に直接接続するか、または、例えば図１のネットワーク１０２などのネットワークを通じて、論理的に区分されたプラットフォームに接続することができる。これらのハードウェア・システム・コンソールは、例えば、デスクトップ・コンピュータ、ラップトップ・コンピュータ、あるいは任意の他の端末でよく、図２のデータ処理システム２００を使用するように実施することができる。ハードウェア・システム・コンソール４８０は、メッセージ・ストリームを復号し、各種ＯＳイメージ４０２～４０８からの情報を各ＯＳイメージにつき少なくとも１つの個別のウィンドウに表示する。同様に、操作者からのキーボード入力情報は、ハードウェア・システム・コンソールによってパッケージされ、論理的に区分されたプラットフォーム４００に送信され、そこで復号し、その時点でハードウェア・システム・コンソール４８０でアクティブなウィンドウに関連付けられた、ハイパーバイザ４１０によってエミュレートされたポート・デバイス・ドライバを介して、適切なＯＳイメージに送られる。ハイパーバイザ４１０は、その他の機能およびサービスも行うことができる。
【００５１】
当業者は、図４に示すハードウェアおよびソフトウェアは様々に異なってもよいことを理解されよう。例えば、図４よりも多いまたは少ないプロセッサ、または図４よりも多いまたは少ないオペレーティング・システム・イメージ、あるいはその両方を使用することができる。
【００５２】
本発明は、データ処理システム内で電源を制御し、論理パーティションをリブートする方法、装置、およびコンピュータ実施命令を提供する。本発明のこの機構は、論理パーティションまたはシステム全体への電源を切る機能を提供する。この機構は、データ処理システム内のパーティションのリブートも可能にする。図の例では、物理的な電源スイッチおよびリセット・スイッチをデータ処理システム中で使用不可にする。この使用不可は、データ処理システムに電源が投入され、論理パーティションを実行した後に行われる。仮想的な電源スイッチおよび仮想的なリセット・スイッチを提供して、個々のパーティショ

ンの電源投入または切断を可能にし、個々のパーティションをリブートすることを可能にする。

【００５３】

次いで図５を参照すると、本発明の好ましい実施形態による、電源の制御および論理パーティションのリブートに使用するプロセスのフローチャートを示している。図５のプロセスは、図４のハイパーバイザ４１０などのパーティション・マネージャ中で実施することができる。パーティション・マネージャは、データ処理システムの論理パーティションの管理および制御に使用されるプロセスである。

【００５４】

プロセスは、データ処理システムの電源投入を検知することによって開始する（ステップ５００）。次いで、データ処理システムがＬＰＡＲモードになると、リセット・ボタンおよび電源ボタンを使用不可にする（ステップ５０２）。論理パーティションにブート・コマンドがあるかどうかを判定する（ステップ５０４）。このコマンドは、図３のＮＶＲＡＭ３６８の内容を調査または検査することによって識別することができる。ＨＳＣは、このＮＶＲＡＭにコマンドを入れて論理パーティションをブートすることができる。論理パーティションにブート・コマンドがある場合は、システム・リソースを割り振り、パーティションをインスタンス化してオペレーティング・システムをブートする（ステップ５０６）。これらの例のオペレーティング・システムはＡＩＸである。

【００５５】

その後、パーティション・マネージャが中断されているかどうかを判定する（ステップ５０８）。このステップを使用して、パーティションにより多くのプロセッサを追加するかどうかを判定する。例えば、１６プロセッサ・システムでは、特定のパーティションで独占的に使用するためにそのパーティションに３つのプロセッサを割り当てることができる。パーティション・マネージャが中断されている場合は、パーティション・マネージャのタスクを中断し（ステップ５１０）、その後プロセスが終了する。

【００５６】

再度ステップ５０４を参照すると、その論理パーティションにブート・コマンドがない場合は、パーティションが終了されているかどうかを判定する（ステップ５１２）。パーティションは、リブートされるか、シャットダウンされると終了する可能性がある。そのパーティションが終了されている場合は、システム・リソースを回収し、プロセッサ状態を実行から停止に変更し、パーティション・ステータスを実行から停止に変更し（ステップ５１４）、プロセスはステップ５０４に戻る。

【００５７】

再度ステップ５０８を参照すると、パーティション・マネージャが中断されていない場合は、プロセスはステップ５０４に戻る。再度ステップ５１２を参照すると、パーティションが終了されていない場合、プロセスはステップ５０４に戻る。

【００５８】

次いで図６に、本発明の好ましい実施形態により、パーティションをシャットダウンするために使用するプロセスのフローチャートを示している。図６のプロセスは、実行時抽象サービス（ＲＴＡＳ）で実施することができる。ＲＴＡＳは、オペレーティング・システム（ＯＳ）が、ＮＶＲＡＭや実時間クロックなどプラットフォームに固有のハードウェアにアクセスするための機能を提供する。このサービスは、サービス維持のために、パーティションのＯＳにハードウェア・エラー・リポートも提供する。ＲＴＡＳは、論理パーティションで実行されるシステム・ファームウェアのコンポーネントとして実装される。ＲＴＡＳは、実際にハードウェア・デバイスにアクセスしハードウェア・エラー・リポートを入手するためのハイパーバイザ呼出しを行う。

【００５９】

このプロセスは、パーティション・オペレーティング・システムがシャットダウンを実行すると開始される。このシャットダウンは、データ処理システム内で仮想電源ボタンを起動することによって開始される。この仮想電源ボタンは、グラフィカル・ユーザ・イン

ターフェース中のボタンか、またはコマンドとして表すことができる。オペレーティング・システムから論理パーティションの電源オフ要求を受け取ったことに応答して、図４のハイパーバイザ４１０などのパーティション・マネージャに要求を渡す（ステップ６００）。ハイパーバイザが別のアクティブなパーティションがあるかどうかを判断するかどうかについて判定する（ステップ６０２）。ハイパーバイザが、別のアクティブなパーティションが存在すると判断した場合は、論理パーティションの電源オフをサービス・プロセッサに送り（ステップ６０４）、その後プロセスは終了する。これらの例では、サービス・プロセッサは、パーティションに割り当てられたプロセッサを処理する。これには、プロセッサを停止状態にすること、またはプロセッサをリセットすることが含まれる。

【００６０】

再度ステップ６０２を参照すると、ハイパーバイザがアクティブなパーティションは存在しないと判断した場合は、システムを完全にシャットダウンするためにシステム電源オフ要求をサービス・プロセッサに送り（ステップ６０６）、その後プロセスは終了する。

【００６１】

次いで図７に、本発明の好ましい実施形態による、論理パーティションのリブートに使用するプロセスのフローチャートを示す。図７のプロセスは、ＲＴＡＳで実施することができる。

【００６２】

このプロセスは、パーティション・オペレーティング・システムがオペレーティング・システムの再起動を指定してシャットダウンを実行すると開始される。プロセスは、ＲＴＡＳがオペレーティング・システムから論理パーティション・リブート要求を受け取り、その要求を図４のハイパーバイザ４１０などのパーティション・マネージャに渡すと開始する（ステップ７００）。次いで、サービス・プロセッサに論理パーティション・リブートを送信し（ステップ７０２）、プロセスはその後終了する。

【００６３】

以下で図８および９を参照して説明するプロセスは、ＨＳＣから開始された要求を処理するプロセスである。下記で説明する図１０のプロセスは、論理的に区分されたデータ処理システムのプロセッサで開始された要求を処理するプロセスである。

【００６４】

次いで図８に、本発明の好ましい実施形態による、論理パーティションをリセットするために使用するプロセスのフローチャートを示す。図８のプロセスは、図３のサービス・プロセッサ３６６によって実行されるコンピュータ命令として実施することができる。

【００６５】

プロセスは、サービス・プロセッサが論理パーティションに対する仮想リセット・スイッチの起動を受け取ると開始する（ステップ８００）。次いで、サービス・プロセッサは、ＮＶＲＡＭのプロセッサ・テーブルに設定されたターゲット・アドレスを更新して、そのパーティションのすべてのプロセッサについて０ｘ１００にする（ステップ８０２）。このアドレスは、すべてのプロセッサがリセット後に見る場所をポイントする。次いで、サービス・プロセッサは、パーティションのすべてのプロセッサに対してシステム・リセット信号をアクティブにし（ステップ８０４）、プロセスはその後終了する。そのパーティションに割り当てられていない、あるいは割り振られていないプロセッサは、このシステム・リセット信号を受信しない。

【００６６】

次いで図９に、本発明の好ましい実施形態により、パーティションの電源を切るために使用するプロセスのフローチャートを示す。図９のプロセスは、図３のサービス・プロセッサ３６６によって実行されるコンピュータ命令として実施することができる。

【００６７】

プロセスは、サービス・プロセッサが、ある論理パーティションに対する仮想電源オフ・スイッチ起動信号を受信することにより開始する（ステップ９００）。サービス・プロセッサは、ＮＶＲＡＭプロセッサ・テーブルを更新し、そのパーティションのすべてのプ

ロセッサについてターゲット・アドレスを０ｘＢ００に設定する（ステップ９０２）。このアドレスは、すべてのプロセッサがリセット後に見る場所をポイントする。次いで、サービス・プロセッサは、電源を切ろうとするパーティションのすべてのプロセッサに対してシステム・リセット信号をアクティブにし（ステップ９０４）、その後プロセスは終了する。

【００６８】

次いで図１０に、本発明の好ましい実施形態により、プロセッサをリセットするのに使用するプロセスのフローチャートを示す。図１０のプロセスは、図３のサービス・プロセッサ３６６によって実行されるコンピュータ命令セットとして実施することができる。

【００６９】

プロセスは、サービス・プロセッサが、論理パーティションのホスト・プロセッサから、論理パーティションの電源オフまたは論理パーティションのリブート要求を受け取ると開始する（ステップ１０００）。サービス・プロセッサは、要求からプロセッサＩＤおよびパーティションＩＤを取り出す（ステップ１００２）。次いで、サービス・プロセッサは、ＮＶＲＡＭプロセッサ・テーブルを更新し、取り出したＩＤとＩＤが一致したホスト・プロセッサのターゲット・アドレスを０ｘＢ００に設定し、そのパーティションのすべての他のプロセッサに対して、設定したターゲットを０ｘＡ００に加える（ステップ１００４）。ターゲット・アドレス０ｘＡ００を有するホスト・プロセッサは、中断されたパーティション・マネージャをウェークアップさせ、その実行を再開することを試みない。ターゲット・アドレスが０ｘＢ００のホスト・プロセッサは、中断したパーティション・マネージャを検出しようとする。パーティション・マネージャが実際に中断している場合は、ターゲット・アドレスＯｘＢ００のプロセッサがパーティション・マネージャを再開する。中断していない場合、ターゲット・アドレスＯｘＢ００のプロセッサのプロセッサは、ターゲット・アドレスが０ｘＡ００のプロセッサと同じ進路をとる。サービス・プロセッサは、パーティションのすべてのプロセッサに対するシステム・リセット信号をアクティブにし（ステップ１００６）、その後プロセスは終了する。

【００７０】

次いで図１１に、本発明の好ましい実施形態により、システム・リセット割込みを処理するために使用するプロセスのフローチャートを示す。図１１のプロセスは、特定パーティションのシステム・リセット割込み（ＳＲＩ）ハンドラで実施される。

【００７１】

プロセスは、パーティション・プロセッサがシステム・リセット割込みを受け取ることから開始する（ステップ１１００）。次いで、プロセッサは、システム・リセット割込み（ＳＲＩ）ハンドラを実行する（ステップ１１０２）。ＳＲＩハンドラは、ＮＶＲＡＭプロセッサ・テーブルからターゲット・アドレスを得る（ステップ１１０４）。このターゲット・アドレスは、図８～１０に関して説明したプロセスを使用して設定する。ターゲット・アドレスが０ｘ１００に等しいかどうかを判定する（ステップ１１０６）。ターゲット・アドレスが０ｘ１００に等しい場合は、パーティションのＳＲＩハンドラに制御を移す（ステップ１１０８）。仮想リセット・スイッチ動作を完了し（ステップ１１１０）、その後プロセスは終了する。仮想リセット・ボタンの機能は、パーティションのＳＲＩハンドラを起動することである。ハイパーバイザがパーティションのＳＲＩハンドラに制御を移すと、パーティションのＳＲＩハンドラは、そのパーティションのＯＳのＳＲＩ機能を呼び出してさらなる動作をとる。

【００７２】

再度ステップ１１０６を参照すると、ターゲット・アドレスが０ｘ１００に等しくない場合は、ターゲット・アドレスが０ｘＡ００または０ｘＢ００であるかどうかを判定する（ステップ１１１２）。このステップを使用して、プロセッサに各自の指定ターゲット・アドレスを指示する。プロセッサがターゲット・アドレス０ｘＡ００を得る場合、そのプロセッサは、中断しているパーティション・マネージャをウェークアップさせる競争に参加しない。プロセッサは単に、自身が現在ハイパーバイザ環境に戻っている、すなわちハ

イパーバイザ・メモリ中の各自の状態変数を停止状態値に設定していることをハイパーバイザに通知する。プロセッサは、自身を待ちループに入れる。待ちループで、これらのプロセッサは常に、ハイパーバイザ領域中の各自に一意に割り当てられたメモリ位置を読み出す。割り当てられたメモリ位置が非ゼロの値を含む場合は、その値をそのプロセッサが分岐するルーチンのアドレスとして扱う。そしてプロセッサは待ちループを抜け、指定の分岐アドレスで開始する実行を続行する。

【００７３】

ターゲット・アドレスが０ｘＢ００である場合は、パーティション・マネージャが中断しているかどうかを判定する（ステップ１１１４）。パーティション・マネージャが中断している場合、プロセッサは競ってパーティション・マネージャをウェークアップさせる（ステップ１１１６）。図の例では、パーティション・マネージャは、図４のハイパーバイザ４１０を使用して実施することができる。パーティション・マネージャをウェークアップさせたプロセッサがあるかどうかを判定する（ステップ１１１８）。ウェークアップさせたプロセッサがある場合は、パーティション・マネージャのタスクを再開し（ステップ１１２０）、その後プロセスは終了する。

【００７４】

再度ステップ１１１４を参照すると、パーティション・マネージャが中断していない場合、上述のようにパーティションのプロセッサは停止状態に入り、回転（spinning）ループでアイドル状態となり（ステップ１１２２）、その後プロセスは終了する。再度ステップ１１１８を参照すると、パーティション・マネージャをウェークアップさせたプロセッサがない場合、プロセスはステップ１１２２に進む。再度ステップ１１１２を参照すると、ターゲット・アドレスが０ｘＡ００である場合、上述のようにプロセスは次いでステップ１１２２に進む。

【００７５】

このように、本発明は、論理パーティションの電源を切る要求、または論理パーティションをリブートする要求を処理する、あるいはその両方を行う改良された方法、装置、およびコンピュータ実施命令を提供する。本発明の機構は、データ処理システム内の他の論理パーティションに割り当てられた他のプロセッサをリブートまたはリセットすることなく、特定の論理パーティションに割り当てられたプロセッサをリブートまたはリセットする能力を提供する。この機構により、他の論理パーティションに影響を与えずに、ある論理パーティションのリソースを個別に扱うことが可能になる。

【図面の簡単な説明】

【００７６】

【図１】本発明を実施することが可能な分散データ処理システムの図である。

【図２】本発明によるデータ処理システムのブロック図である。

【図３】論理的にパーティションに区分したサーバとして実施することが可能なデータ処理システムのブロック図である。

【図４】本発明を実施することが可能な、論理的にパーティションに区分した例示的プラットフォームのブロック図である。

【図５】本発明の好ましい実施形態により、電源を制御し、論理パーティションをリブートするために使用するプロセスのフローチャートである。

【図６】本発明の好ましい実施形態により、論理パーティションをシャットダウンするために使用するプロセスのフローチャートである。

【図７】本発明の好ましい実施形態により、論理パーティションをリブートするために使用するプロセスのフローチャートである。

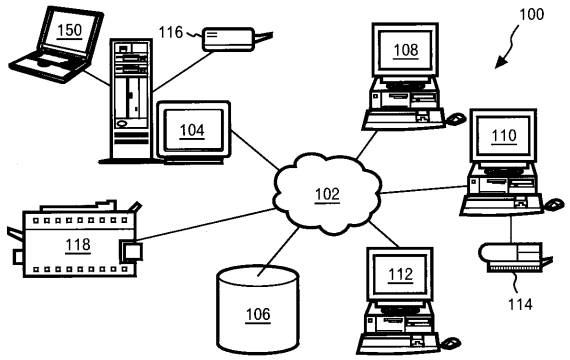
【図８】本発明の好ましい実施形態により、論理パーティションをリセットするために使用するプロセスのフローチャートである。

【図９】本発明の好ましい実施形態により、パーティションの電源を切るために使用するプロセスのフローチャートである。

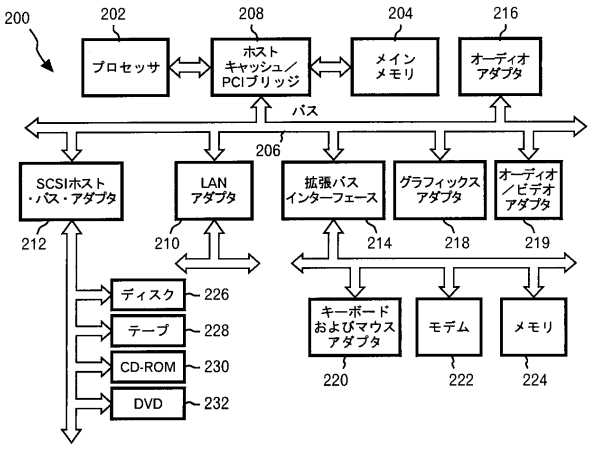
【図１０】本発明の好ましい実施形態により、プロセッサをリセットするために使用する

プロセスのフローチャートである。
【図１１】本発明の好ましい実施形態による、システム・リセット割込みを処理するために使用するプロセスのフローチャートである。

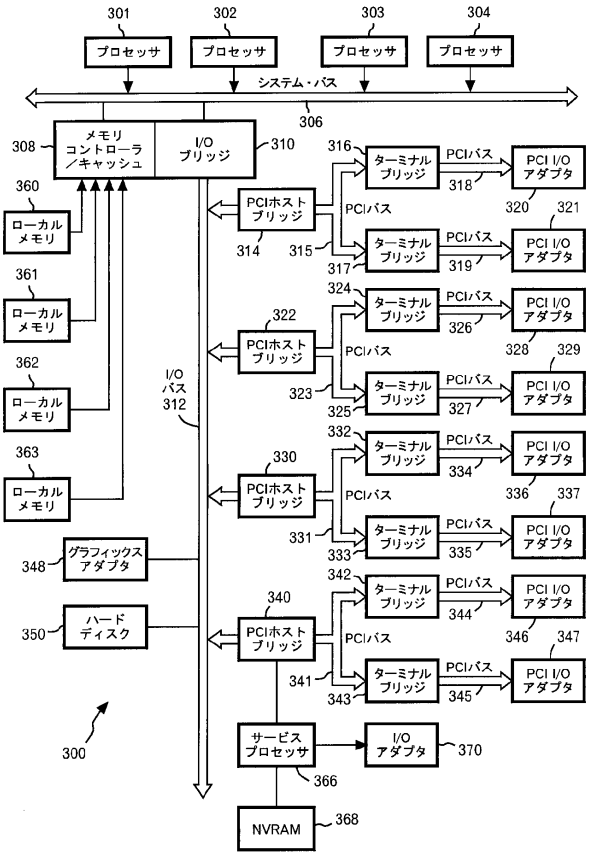
【図１】



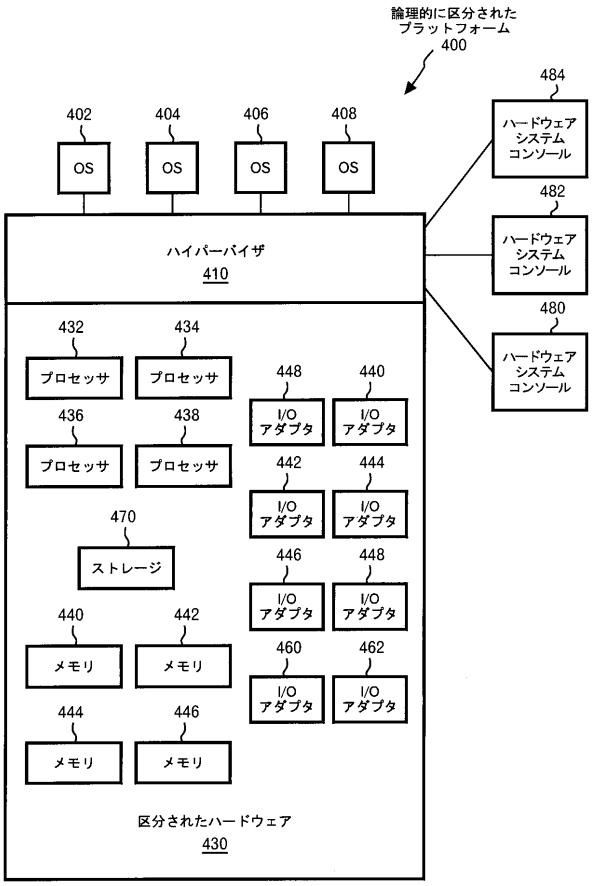
【図２】



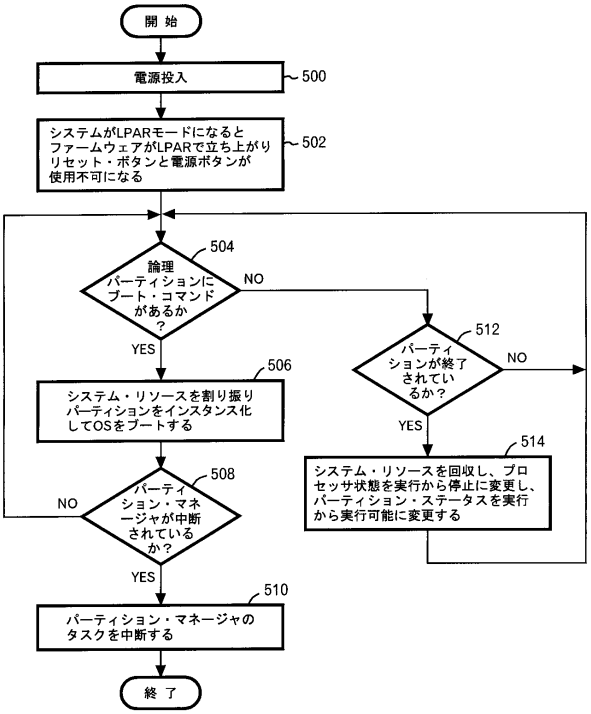
【図３】



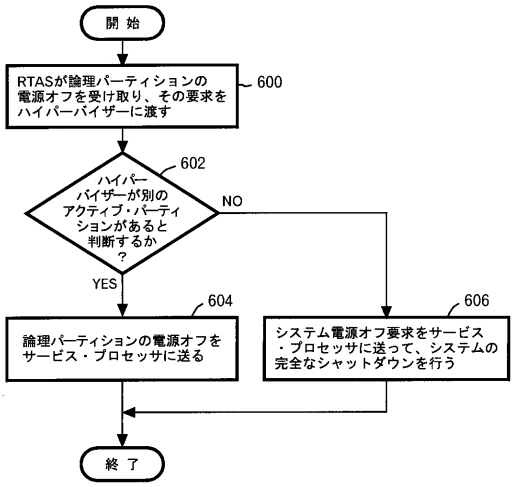
【図４】



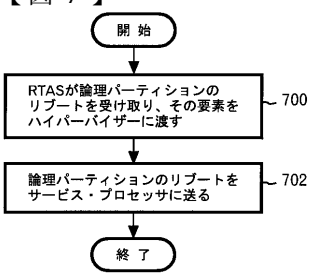
【図５】



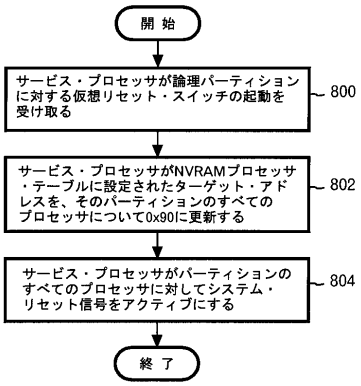
【図６】



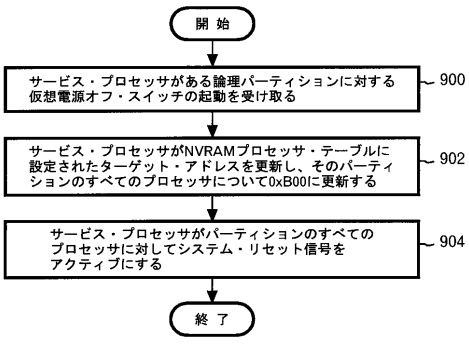
【図７】



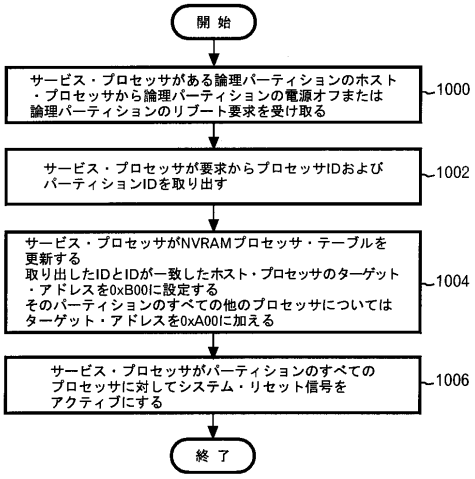
【図８】



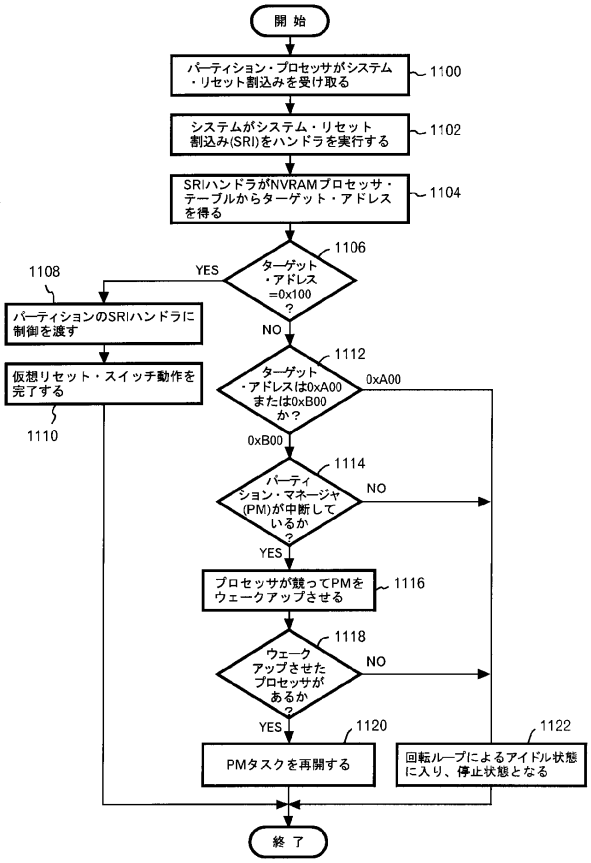
【図９】



【図１０】


開始
サービス・プロセッサがある論理パーティションのホスト・プロセッサから論理パーティションの電源オフまたは論理パーティションのリブート要求を受け取る
1000
サービス・プロセッサが要求からプロセッサIDおよびパーティションIDを取り出す
1002
サービス・プロセッサがNVRAMプロセッサ・テーブルを更新する
取り出したIDとIDが一致したホスト・プロセッサのターゲット・アドレスを0xB00に設定する
そのパーティションのすべての他のプロセッサについてはターゲット・アドレスを0xA00に加える
1004
サービス・プロセッサがパーティションのすべてのプロセッサに対してシステム・リセット信号をアクティブにする
1006
終了

【図１１】


開始
パーティション・プロセッサがシステム・リセット割込みを受け取る
1100
システムがシステム・リセット割込み(SRI)をハンドラを実行する
1102
SRIハンドラがNVRAMプロセッサ・テーブルからターゲット・アドレスを得る
1104
ターゲット・アドレス=0x100？
1106
YES
1108
パーティションのSRIハンドラに制御を渡す
仮想リセット・スイッチ動作を完了する
1110
NO
ターゲット・アドレスは0xA00または0xB00か？
1112
0xA00
0xB00
パーティション・マネージャ(PM)が中断しているか？
1114
NO
YES
プロセッサが競ってPMをウェークアップさせる
1116
ウェークアップさせたプロセッサがあるか？
1118
NO
YES
PMタスクを再開する
1120
回転ループによるアイドル状態に入り、停止状態となる
1122
終了

フロントページの続き

(74)代理人 100086243
弁理士 坂口 博
(74)代理人 100091568
弁理士 市位 嘉宏
(74)代理人 100108501
弁理士 上野 剛史
(72)発明者 ドーキンズ、ジョージ、ジョン、ジュニア
アメリカ合衆国７８７５９ テキサス州オースチン ブラックホーク・ドライブ １１６０１
(72)発明者 リー、バン、ホー
アメリカ合衆国７８６１３ テキサス州セダー・パーク ペブル・ブロック・ロード １１０３
(72)発明者 パテル、カニーシャ
アメリカ合衆国７８６１３ テキサス州セダー・パーク メスキート・ロード １２２２
(72)発明者 ファン、ピーター、ディン
アメリカ合衆国７８７５３ テキサス州オースチン ハリス・リッジ・ブルバード ナンバー ９３３
(72)発明者 ウィロビー、デビッド
アメリカ合衆国７８７３２ テキサス州オースチン マクネリー・トレイル １２８１８

審査官 ▲はま▼中 信行

(56)参考文献 特開平０６－２５０８５８（ＪＰ，Ａ）
特開２００１－０４３０９８（ＪＰ，Ａ）
特開平０９－３１９６３７（ＪＰ，Ａ）
特開平１０－０１１１７９（ＪＰ，Ａ）
特開２００１－２５６０６６（ＪＰ，Ａ）
江口三代一外，パーソナルコンピュータによる情報処理教育とＣＡＩのためのシステム，情報処理学会論文誌，１９９０年１１月１５日，第31巻，第11号，p.1556-1564

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 9/46
G06F 1/26