

(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2011-221634

(P2011-221634A)

(43) 公開日 平成23年11月4日(2011.11.4)

(51) Int.Cl.

G06F 9/46 (2006.01)

F I

G06F 9/46 350

テーマコード (参考)

審査請求 未請求 請求項の数 15 O L (全 22 頁)

(21) 出願番号 特願2010-87728 (P2010-87728)
 (22) 出願日 平成22年4月6日(2010.4.6)

(71) 出願人 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 100075513
 弁理士 後藤 政喜
 (74) 代理人 100114236
 弁理士 藤井 正弘
 (74) 代理人 100120260
 弁理士 飯田 雅昭
 (72) 発明者 畑▲崎▼ 恵介
 神奈川県川崎市麻生区王禅寺1099番地
 株式会社日立製作所システム開発研究所
 内

最終頁に続く

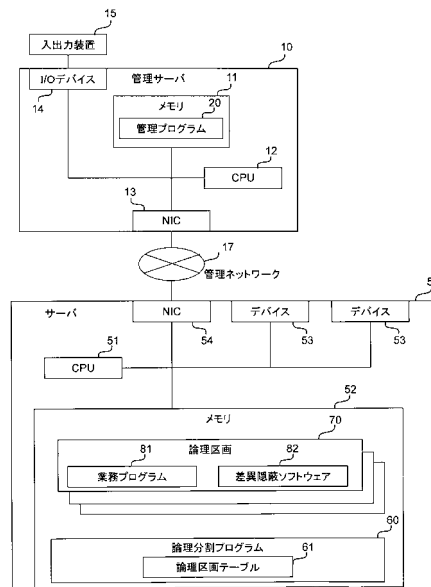
(54) 【発明の名称】 計算機システム、論理区画管理方法及び論理分割処理プログラム

(57) 【要約】

【課題】サーバ仮想化技術において、ハードウェアを仮想化するハードウェアのエミュレーションを実現し、論理分割による性能の安定性を保証する仮想化方法を提供する。

【解決手段】複数の計算機を管理する管理計算機及び一以上の論理区画を生成する論理分割処理部とを備える計算機を備える計算機システムであって、論理分割処理部は、仮想的なデバイスである仮想デバイスが割り当てられた論理区画の生成要求を受信した場合に、プロセッサを論理的に分割した第1の論理プロセッサと、仮想デバイスと、仮想デバイスを制御するための仮想デバイスドライバとを備える第1の副論理区画、及び、プロセッサを論理的に分割した第2の論理プロセッサと、デバイスを論理的に分割した論理デバイスと、論理デバイスを制御するためのデバイスドライバとを備える第2の副論理区画を含む論理区画を生成することを特徴とする。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

第 1 のプロセッサと、前記第 1 のプロセッサに接続される第 1 のメモリとを備え、複数の計算機を管理する管理計算機、及び、第 2 のプロセッサと、前記第 2 のプロセッサに接続されるデバイスと、前記第 2 のプロセッサ及び前記デバイスを論理的に分割することによって一以上の論理区画を生成する論理分割処理部とを備える計算機、を備える計算機システムであって、

前記論理分割処理部は、前記管理計算機から、仮想的なデバイスである仮想デバイスが割り当てられた前記論理区画の生成要求を受信した場合に、前記第 2 のプロセッサを論理的に分割した第 1 の論理プロセッサと、前記仮想デバイスと、前記仮想デバイスを制御するための仮想デバイスドライバとを備える第 1 の副論理区画、及び、前記第 2 のプロセッサを論理的に分割した第 2 の論理プロセッサと、前記デバイスを論理的に分割した論理デバイスと、前記論理デバイスを制御するためのデバイスドライバとを備える第 2 の副論理区画を含む前記論理区画を生成し、

前記第 1 の副論理区画では、所定の業務に対応する業務プログラムが実行され、

前記第 2 の副論理区画では、前記第 1 の副論理区画が備える前記仮想デバイスに対する命令を処理する仮想化処理プログラムが実行され、

前記仮想デバイスドライバは、

前記業務プログラムから前記仮想デバイスに対する第 1 の命令が発行された場合に、前記第 1 の命令を取得し、

前記取得された第 1 の命令を前記仮想化処理プログラムに送信し、

前記仮想化処理プログラムが、

前記第 1 の命令を取得し、

前記取得された第 1 の命令を前記論理デバイスに対する第 2 の命令に変換し、

前記変換され第 2 の命令の実行結果を前記業務プログラムに送信することを特徴とする計算機システム。

【請求項 2】

前記デバイスは、第 2 のメモリを含み、

前記第 1 の副論理区画は、前記第 2 のメモリが論理的に分割された第 1 の論理メモリを備え、

前記第 2 の副論理区画は、前記第 2 のメモリが論理的に分割された第 2 の論理メモリを備え、

前記第 1 の論理メモリに前記業務プログラムが格納され、

前記第 2 の論理メモリに前記仮想化処理プログラムが格納され、

前記論理分割処理部は、

前記論理区画を生成する場合に、前記第 1 の論理プロセッサと、前記第 1 の論理メモリとを前記第 1 の副論理区画に、割り当て、前記第 2 の論理プロセッサと、前記第 2 の論理メモリと、前記論理デバイスとを前記第 2 の副論理区画に、割り当て、

前記論理デバイスに対応する前記仮想デバイスを前記第 1 の副論理区画に割り当てることを特徴とする請求項 1 に記載の計算機システム。

【請求項 3】

前記第 2 の副論理区画は、前記第 1 の命令と前記第 2 の命令との対応関係を保持する命令変換情報を保持し、

前記仮想化処理プログラムは、

前記命令変換情報に基づいて、前記取得された第 1 の命令を前記第 2 の命令に変換し、

前記変換された第 2 の命令を前記論理デバイスドライバへ発行することを特徴とする請求項 2 に記載の計算機システム。

【請求項 4】

前記仮想デバイスドライバは、

前記第 1 の副論理区画上で前記業務プログラムが起動された後に、前記第 2 の論理プロ

10

20

30

40

50

セッサを識別するための識別子を取得し、

前記第 1 の命令が取得された場合に、前記取得された識別子を用いて、前記第 2 の論理プロセッサへの割り当てを発生させることによって前記取得された第 1 の命令を前記仮想化処理プログラムに送信することを特徴とする請求項 3 に記載の計算機システム。

【請求項 5】

前記第 1 の副論理区画及び前記第 2 の副論理区画は、前記第 2 のメモリが論理的に分割され、互いにアクセス可能な共有メモリを備え、

前記仮想デバイスドライバは、前記第 1 の命令が取得された場合に、前記第 2 の論理プロセッサに割り込みを発生させ、前記取得された第 1 の命令を前記共有メモリに書き込み、

10

前記仮想化処理プログラムは、

前記共有メモリから前記取得された第 1 の命令を取得することを特徴とする請求項 4 に記載の計算機システム。

【請求項 6】

前記論理デバイスドライバは、前記第 2 の副論理区画上で前記仮想化処理プログラムが起動された後に、前記第 1 の論理プロセッサを識別するための識別子を取得し、

前記仮想化処理プログラムは、

前記論理区画が受信した前記仮想デバイスに対する第 3 の命令を検知し、

前記命令変換情報に基づいて、前記検知された第 3 の命令を前記仮想デバイスへの第 4 の命令に変換し、

20

前記第 1 の論理プロセッサに割り込みを発生させ、前記変換された第 4 の命令を前記共有メモリに書き込み、

前記業務プログラムは、

前記共有メモリから前記変換された第 4 の命令を取得し、

前記変換された第 4 の命令を前記仮想デバイスドライバへ発行することを特徴とする請求項 4 に記載の計算機システム。

【請求項 7】

前記管理計算機は、情報を表示する表示部を含む入出力装置を備え、

前記論理分割処理部は

前記管理計算機から、前記論理区画へのアクセスを受信した場合に、

30

前記表示部に、前記第 1 の副論理区画を前記論理区画として表示することを特徴とする請求項 1 に記載の計算機システム。

【請求項 8】

前記管理計算機は、情報を表示する表示部を含む入出力装置を備え、

前記第 1 のメモリは、前記第 1 のプロセッサに実行され、前記論理区画を管理するための管理プログラムを格納し、

前記管理プログラムは、

前記仮想デバイスが割り当てられた論理区画を生成するか否かを選択する選択部と、前記第 1 の副論理区画を生成するために必要となる情報を入力する入力部と、前記第 2 の副論理区画を生成するために必要となる情報を入力する入力部と、を含む入力画面を前記表示部に表示させることを特徴とする請求項 1 に記載の計算機システム。

40

【請求項 9】

第 1 のプロセッサと、前記第 1 のプロセッサに接続される第 1 のメモリとを備え、複数の計算機を管理する管理計算機、及び、第 2 のプロセッサと、前記第 2 のプロセッサに接続されるデバイスと、前記第 2 のプロセッサ及び前記デバイスを論理的に分割することによって一以上の論理区画を生成する論理分割処理部とを備える計算機、を備える計算機システムにおける論理区画管理方法であって、

前記方法は、

前記論理分割処理部が、前記管理計算機から、仮想的なデバイスである仮想デバイスが割り当てられた前記論理区画の生成要求を受信した場合に、前記第 2 のプロセッサを論理

50

的に分割した第 1 の論理プロセッサと、前記仮想デバイスと、前記仮想デバイスを制御するための仮想デバイスドライバとを備え、所定の業務プログラムが実行される第 2 の副論理区画、及び、前記第 2 のプロセッサを論理的に分割した第 2 の論理プロセッサと、前記デバイスを論理的に分割した論理デバイスと、前記論理デバイスを制御するためのデバイスドライバとを備え、前記第 1 の副論理区画が備える前記仮想デバイスに対する命令を処理する仮想化処理プログラムが実行される前記第 2 の副論理区画を含む前記論理区画を生成する第 1 のステップと、

前記仮想デバイスドライバが、前記業務プログラムから前記仮想デバイスに対する第 1 の命令が発行された場合に、前記第 1 の命令を取得する第 2 のステップと、

前記仮想デバイスドライバが、前記取得された第 1 の命令を前記仮想化処理プログラムに送信する第 3 のステップと、

前記仮想化処理プログラムが、前記第 1 の命令を取得する第 4 のステップと、

前記仮想化処理プログラムが、前記取得された第 1 の命令を前記論理デバイスに対する第 2 の命令に変換する第 5 のステップと、

前記仮想化処理プログラムが、前記変換された第 2 の命令の実行結果を前記業務プログラムに送信する第 6 のステップと、を含むことを特徴とする論理区画管理方法。

【請求項 10】

前記デバイスは、第 2 のメモリを含み、

前記第 1 の副論理区画は、前記第 2 のメモリが論理的に分割された第 1 の論理メモリを備え、

前記第 2 の副論理区画は、前記第 2 のメモリが論理的に分割された第 2 の論理メモリを備え、

前記第 1 の論理メモリに前記業務プログラムが格納され、

前記第 2 の論理メモリに前記仮想化処理プログラムが格納され、

前記第 1 のステップは、

前記論理分割処理部が、前記第 1 の論理プロセッサと、前記第 1 の論理メモリとを前記第 1 の副論理区画に、割り当てるステップと、

前記第 2 の論理プロセッサと、前記第 2 の論理メモリと、前記論理デバイスとを前記第 2 の副論理区画に、割り当てるステップと、

前記論理デバイスに対応する前記仮想デバイスを前記第 1 の副論理区画に割り当てるステップと、を含むことを特徴とする請求項 9 に記載の論理区画管理方法。

【請求項 11】

前記第 2 の副論理区画は、前記第 1 の命令と前記第 2 の命令との対応関係を保持する命令変換情報を保持し、

前記第 5 のステップは、

前記仮想化処理プログラムが、前記命令変換情報に基づいて、前記取得された第 1 の命令を前記第 2 の命令に変換するステップと、

前記仮想化処理プログラムが、前記変換された第 2 の命令を前記論理デバイスドライバへ発行するステップと、を含むことを特徴とする請求項 10 に記載の論理区画管理方法。

【請求項 12】

前記方法は、前記仮想デバイスドライバが、前記第 1 の副論理区画上で前記業務プログラムが起動された後に、前記第 2 の論理プロセッサを識別するための識別子を取得するステップを含み、

前記第 3 のステップは、前記仮想デバイスドライバが、前記取得された識別子を用いて、前記第 2 の論理プロセッサへの割り当てを発生させることによって前記取得された第 1 の命令を前記仮想化処理プログラムに送信するステップを含むことを特徴とする請求項 11 に記載の論理区画管理方法。

【請求項 13】

前記第 1 の副論理区画及び前記第 2 の副論理区画は、前記第 2 のメモリが論理的に分割され、互いにアクセス可能な共有メモリを備え、

10

20

30

40

50

前記第 3 のステップは、

前記仮想デバイスドライバが、前記取得された第 1 の命令を前記共有メモリに書き込むステップを含み、

前記仮想デバイスドライバが、前記第 2 の論理プロセッサに割り込みを発生させるステップと、を含み、

前記第 4 のステップは、

前記仮想化処理プログラムが、前記第 2 の論理プロセッサへの割り込みを検知するステップと、

前記仮想化処理プログラムが、前記共有メモリから前記取得された第 1 の命令を取得するステップと、を含むことを特徴とする請求項 1 2 に記載の論理区画管理方法。

10

【請求項 1 4】

前記方法は、

前記論理デバイスドライバが、前記第 2 の副論理区画上で前記仮想化処理プログラムが起動された後に、前記第 1 の論理プロセッサを識別するための識別子を取得するステップと、

前記仮想化処理プログラムが、前記論理区画が受信した前記仮想デバイスに対する第 3 の命令を検知するステップと、

前記仮想化処理プログラムが、前記命令変換情報に基づいて、前記検知された第 3 の命令を前記仮想デバイスへの第 4 の命令に変換するステップと、

前記仮想化処理プログラムが、前記変換された第 4 の命令を前記共有メモリに書き込むステップと、

20

前記仮想化処理プログラムが、前記第 1 の論理プロセッサに割り込みを発生させるステップと、

前記業務プログラムが、前記共有メモリから前記変換された第 4 の命令を取得するステップと、

前記業務プログラムが、前記変換された第 4 の命令を前記仮想デバイスドライバへ発行するステップと、を含むことを特徴とする請求項 1 2 に記載の論理区画管理方法。

【請求項 1 5】

第 1 のプロセッサと、前記第 1 のプロセッサに接続されるメモリと、前記第 1 のプロセッサに接続されるデバイスと、前記第 1 のプロセッサ、前記第 1 のメモリ及び前記デバイスを論理的に分割することによって一以上の論理区画を生成する計算機における論理分割処理プログラムであって、

30

論理分割処理プログラムは、仮想的なデバイスである仮想デバイスが割り当てられた前記論理区画の生成要求を受信した場合に、前記第 1 のプロセッサを論理的に分割した第 1 の論理プロセッサと、前記仮想デバイスと、前記仮想デバイスを制御するための仮想デバイスドライバとを備え、所定の業務プログラムが実行される第 1 の副論理区画、及び、前記第 1 のプロセッサを論理的に分割した第 2 の論理プロセッサと、前記デバイスを論理的に分割した論理デバイスと、前記論理デバイスを制御するためのデバイスドライバとを備え、前記第 1 の副論理区画が備える前記仮想デバイスに対する命令を処理する仮想化処理プログラムが実行される前記第 2 の副論理区画を含む前記論理区画を生成する手順を前記計算機に実行させることを特徴とする論理分割処理プログラム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、計算機システムに関し、特に計算機上で複数の仮想的な計算機を構築する方法に関する。

【背景技術】

【0002】

計算機システムにおいて、1 台のサーバに複数の仮想的なサーバを構築する方法として、サーバ仮想化がある。

50

【 0 0 0 3 】

サーバ仮想化の方法として、1台のサーバが備えるCPU、メモリ及びI/Oデバイスを複数の仮想的なサーバで共有し、必要に応じて、CPU、メモリ及びI/Oデバイスを各仮想的なサーバに割り当てる共有方式がある（例えば、特許文献1参照）。

【 0 0 0 4 】

また、別の方法として、1台のサーバの備えるCPU、メモリ及びI/Oデバイスなどを論理的に分割し、複数の論理的な区画（以下、論理区画と記載する）に分ける論理分割方式がある。論理分割方式では、論理区画上で稼働する業務プログラムが論理的に分割されたCPU、メモリ及びI/Oデバイス等を占有するため、論理区画上で稼働する業務プログラムの実行性能を安定化できるメリットがある。

10

【 0 0 0 5 】

しかし、前述した論理分割方式では、あるサーバ上で構築された業務プログラムを別のサーバに移行したいというニーズに対して、以下のような問題点がある。

【 0 0 0 6 】

すなわち、論理分割方式では、同一のCPUアーキテクチャを利用していても、それぞれのサーバが備えるチップセット又はI/Oデバイスなどのハードウェアデバイス（以下、デバイスと記載する）の違いによって、移行先のサーバ上において業務プログラムを改変することなく実行できない場合がある。

【 先行技術文献 】

【 特許文献 】

20

【 0 0 0 7 】

【 特許文献 1 】 米国特許第 6 4 9 6 8 4 7 号 明 細 書

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 0 8 】

前述した問題を解決するため、サーバが備えるデバイスに関わらず、サーバ上で稼働する業務プログラムに対して仮想的なデバイスに見せる方法が考えられる。

【 0 0 0 9 】

しかし、前述した方法では、仮想的なデバイスを見せるための処理によって発生するオーバーヘッドの影響を論理区画で稼働する業務プログラムが受けてしまうため、当該業務プログラムの実行性能が不安定になる。

30

【 0 0 1 0 】

本発明の目的は、サーバ仮想化技術において、サーバのハードウェアの違いを隠蔽する処理のオーバーヘッドを安定化する。すなわち、ハードウェアの仮想化を実現するとともに、ハードウェアの異なるサーバへの移行による影響を業務プログラムが受けにくいようなサーバ仮想化技術を提供することである。

【 課題を解決するための手段 】

【 0 0 1 1 】

本発明の代表的な一例を示せば以下の通りである。すなわち、第1のプロセッサと、前記第1のプロセッサに接続される第1のメモリとを備え、複数の計算機を管理する管理計算機、及び、第2のプロセッサと、前記第2のプロセッサに接続されるデバイスと、前記第2のプロセッサ及び前記デバイスを論理的に分割することによって一以上の論理区画を生成する論理分割処理部とを備える計算機、を備える計算機システムであって、前記論理分割処理部は、前記管理計算機から、仮想的なデバイスである仮想デバイスが割り当てられた前記論理区画の生成要求を受信した場合に、前記第2のプロセッサを論理的に分割した第1の論理プロセッサと、前記仮想デバイスと、前記仮想デバイスを制御するための仮想デバイスドライバとを備える第1の副論理区画、及び、前記第2のプロセッサを論理的に分割した第2の論理プロセッサと、前記デバイスを論理的に分割した論理デバイスと、前記論理デバイスを制御するためのデバイスドライバとを備える第2の副論理区画を含む前記論理区画を生成し、前記第1の副論理区画では、所定の業務に対応する業務プログラ

40

50

ムが実行され、前記第2の副論理区画では、前記第1の副論理区画が備える前記仮想デバイスに対する命令を処理する仮想化処理プログラムが実行され、前記仮想デバイスドライバは、前記業務プログラムから前記仮想デバイスに対する第1の命令が発行された場合に、前記第1の命令を取得し、前記取得された第1の命令を前記仮想化処理プログラムに送信し、前記仮想化処理プログラムが、前記第1の命令を取得し、前記取得された第1の命令を前記論理デバイスに対する第2の命令に変換し、前記変換され第2の命令の実行結果を前記業務プログラムに送信することを特徴とする。

【発明の効果】

【0012】

本発明によれば、論理分割によるサーバ仮想化において、ハードウェアの仮想化を実現し、かつ業務プログラムを実行する第一副論理区画の性能の安定化を保證できる。 10

【図面の簡単な説明】

【0013】

【図1】本発明の実施形態の計算機システムの構成例を説明するブロック図である。

【図2】本発明の実施形態のサーバの論理区画の構成を説明するブロック図である。

【図3】本発明の実施形態の業務プログラムの構成例を示す説明図である。

【図4】本発明の実施形態の差異隠蔽ソフトウェアの構成例を示す説明図である。

【図5】本発明の実施形態のデバイスマップテーブルの一例を示す説明図である。

【図6】本発明の実施形態の変換テーブルの一例を示す説明図である。

【図7】本発明の実施形態における、論理区画の生成処理を説明するフローチャートである。 20

【図8】本発明の実施形態の論理分割プログラムが実行する論理区画の生成処理を説明するフローチャートである。

【図9】本発明の実施形態における論理区画の起動処理を説明するフローチャートである。

【図10】本発明の実施形態の論理区画におけるデバイスに対する命令実行の処理を説明するフローチャートである。

【図11】本発明の実施形態の論理区画におけるデバイスに対するデータの受信処理を説明するフローチャートである。

【図12】本発明の実施形態における、論理区画生成のために表示されるUI (User Interface) の一例を示す説明図である。 30

【図13】本発明の実施形態の論理区画にユーザがアクセスする場合の処理について説明するブロック図である。

【発明を実施するための形態】

【0014】

以下に、本発明の実施形態について記載する。

【0015】

[実施形態]

図1は、本発明の実施形態の計算機システムの構成例を説明するブロック図である。

【0016】

本実施形態における計算機システムは、管理サーバ10、入出力装置15、管理ネットワーク17及びサーバ50から構成される。管理サーバ10とサーバ50とは管理ネットワーク17を介して互いに接続される。 40

【0017】

管理サーバ10は、CPU12、メモリ11、NIC (Network Interface Card) 13、及びI/Oデバイス14を備える。

【0018】

CPU12は、メモリ11上に展開されるプログラムを実行する。

【0019】

メモリ11は、CPU12によって実行されるプログラム及び当該プログラムを実行す 50

るために必要な情報を格納する。メモリ 11 は、管理プログラム 20 を格納する。

【0020】

管理プログラム 20 は、サーバ 50 上に生成される論理区画 70 を管理するプログラムである。

【0021】

なお、管理プログラム 20 によって実現される機能を、管理サーバ 10 が搭載するハードウェア、ファームウェア、又はこれらの組み合わせによって実装してもよい。

【0022】

N I C 13 は、管理ネットワーク 17 と接続するためのインタフェースである。I / O デバイス 14 は、入出力装置 15 と接続するためのインタフェースである。

【0023】

入出力装置 15 は、マウス、キーボード及びディスプレイなどの装置であり、管理サーバ 10 とユーザとの間で情報を入出力するために利用される。

【0024】

サーバ 50 は、C P U 51、メモリ 52、デバイス 53 及び N I C 54 を備える。

【0025】

C P U 51 は、メモリ 52 上に展開されるプログラムを実行する。

【0026】

メモリ 52 は、C P U 51 によって実行されるプログラム及び当該プログラムを実行するために必要な情報を格納する。メモリ 52 は、論理分割プログラム 60 を格納する。

【0027】

論理分割プログラム 60 は、サーバ 50 の C P U 51、メモリ 52、デバイス 53 を論理的に分割して、一以上の論理区画 70 を生成するためのプログラムである。ここで、デバイス 53 は、例えば、N I C、H B A (H o s t B u s A d a p t o r)、C A N (C o n v e r g e d N e t w o r k A d a p t e r) 及びチップセットなどを示す。

【0028】

C P U 51 を論理区画 70 に割り当てる方法としては、C P U 51 が複数のコアを備える場合、それぞれのコアを各論理区画 70 に割り当てる方法が考えられる。また、メモリ 52 を論理区画 70 に割り当てる方法としては、メモリ 52 のアドレス領域の一部を各論理区画 70 に割り当てる方法が考えられる。

【0029】

論理分割プログラム 60 によって生成された論理区画 70 に関する情報は、論理分割プログラム 60 が備える論理区画テーブル 61 に格納される。なお、論理区画テーブル 61 には、論理区画 70 を識別するための論理区画名を格納し、論理分割プログラム 60 は論理区画名によって論理区画 70 を識別する。

【0030】

また、メモリ 52 上には、論理分割プログラム 60 によって生成された一以上の論理区画のデータである論理区画 70 が格納される。各論理区画 70 は、それぞれ、業務プログラム 81 及び差異隠蔽ソフトウェア 82 を備える。

【0031】

業務プログラム 81 は、業務を実行するためのプログラムである。差異隠蔽ソフトウェア 82 は、デバイスの差異を隠蔽するための処理（エミュレーション処理）を実行するためのソフトウェアである。なお、差異隠蔽ソフトウェア 82 は、プログラムと当該プログラムを実行するために必要な情報とから構成される。

【0032】

業務プログラム 81 の詳細については、図 3 を用いて後述する。また、差異隠蔽ソフトウェア 82 の詳細については、図 4 を用いて後述する。

【0033】

なお、論理分割プログラム 60 及び差異隠蔽ソフトウェア 82 によって実現される機能

10

20

30

40

50

は、サーバ 50 が搭載するハードウェア、ファームウェア、又はこれらの組み合わせによって実装してもよい。

【0034】

図 2 は、本発明の実施形態のサーバ 50 の論理区画 70 の構成を説明するブロック図である。

【0035】

サーバ 50 は、論理分割プログラム 60 を実行することによって、サーバ 50 の CPU 51、メモリ 52 及びデバイス 53 を論理的に分割して、一以上の論理区画 70 に割り当てる。

【0036】

論理分割プログラム 60 によって生成された論理区画 70 は、第一副論理区画 91 と第二副論理区画 92 とを含む。第一副論理区画 91 と第二副論理区画 92 とには、論理区画 70 に割り当てられた CPU 51、メモリ 52 及びデバイス 53 がそれぞれ割り当てられる。

【0037】

第一副論理区画 91 は、CPU 510、メモリ 520、論理デバイス 900 を備える。

【0038】

CPU 510 は、第一副論理区画 91 に割り当てられた CPU である。メモリ 520 は、第一副論理区画 91 に割り当てられたメモリである。メモリ 520 は、業務プログラム 81 を格納する。

【0039】

論理デバイス 900 は、論理分割プログラム 60 によって生成される論理的なデバイスである。すなわち、第一副論理区画 91 には、実際のデバイス 53 を割り当てるのではなく、仮想的なデバイスである論理デバイス 900 が割り当てられる。

【0040】

第二副論理区画 92 は、CPU 511、メモリ 521、デバイス 530 を備える。

【0041】

CPU 511 は、第二副論理区画 92 に割り当てられた CPU である。メモリ 521 は、第二副論理区画 92 に割り当てられたメモリである。メモリ 521 は、差異隠蔽ソフトウェア 82 を格納する。

【0042】

また、第一副論理区画 91 及び第二副論理区画 92 は、それぞれがアクセス可能な共有メモリ 540 を備える。共有メモリ 540 は、メモリ 52 のアドレス領域の一部が割り当てられる。

【0043】

本発明は、1つの論理区画 70 に対して、業務プログラム 81 を実行する第一副論理区画 91 と差異隠蔽ソフトウェア 82 を実行する第二副論理区画 92 とが含まれることに特徴がある。すなわち、業務プログラム 81 を実行するためのリソースと差異隠蔽ソフトウェア 82 を実行するためのリソースとがそれぞれ別個に割り当てられる。このため、業務プログラム 81 は差異隠蔽ソフトウェア 82 の影響を受けることなく実行可能となり、業務プログラム 81 の実行性能を安定化させることができる。

【0044】

図 3 は、本発明の実施形態の業務プログラム 81 の構成例を示す説明図である。

【0045】

業務プログラム 81 は、アプリケーション 811、OS (Operating System) 812 及び論理ドライバ 813 を含む。

【0046】

OS 812 は、第一副論理区画 91 上で稼働する OS である。アプリケーション 811 は、OS 812 上で実行されるアプリケーションである。アプリケーション 811 が実行されることによって、当該アプリケーション 811 に対応した業務が実行される。

10

20

30

40

50

【 0 0 4 7 】

論理ドライバ 8 1 3 は、第一副論理区画 9 1 の備える論理デバイス 9 0 0 に対応するデバイスドライバである。なお、論理ドライバ 8 1 3 は、C D - R O M やネットワークからのダウンロードなどの方法によってインストールできる。

【 0 0 4 8 】

図 4 は、本発明の実施形態の差異隠蔽ソフトウェア 8 2 の構成例を示す説明図である。

【 0 0 4 9 】

差異隠蔽ソフトウェア 8 2 は、制御プログラム 8 2 1、実ドライバ 8 2 2、デバイスマップテーブル 8 2 3 及び変換テーブル 8 2 4 を含む。

【 0 0 5 0 】

制御プログラム 8 2 1 は、第二副論理区画 9 2 を制御するためのプログラムである。実ドライバ 8 2 2 は、第二副論理区画 9 2 に割り当てられたデバイス 5 3 0 を制御するためのドライバである。実ドライバ 8 2 2 は、デバイス 5 3 のベンダによって提供されるドライバを用いてもよい。

【 0 0 5 1 】

デバイスマップテーブル 8 2 3 は、論理デバイス 9 0 0 とデバイス 5 3 0 との対応を管理するテーブルである。デバイスマップテーブル 8 2 3 の詳細については、図 5 を用いて後述する。

【 0 0 5 2 】

変換テーブル 8 2 4 は、論理デバイス 9 0 0 に対する命令と、デバイス 5 3 0 に対する命令との対応を管理するテーブルである。変換テーブル 8 2 4 は、論理デバイス 9 0 0 とデバイス 5 3 0 との対応毎に一つ存在する。変換テーブル 8 2 4 の詳細については、図 6 を用いて後述する。

【 0 0 5 3 】

図 5 は、本発明の実施形態のデバイスマップテーブル 8 2 3 の一例を示す説明図である。

【 0 0 5 4 】

デバイスマップテーブル 8 2 3 は、論理デバイス 9 0 0 とデバイス 5 3 0 との対応関係を格納する。デバイスマップテーブル 8 2 3 は、論理デバイス識別子 8 4 1 とデバイス識別子 8 4 2 とを含む。

【 0 0 5 5 】

論理デバイス識別子 8 4 1 は、第一副論理区画 9 1 が備える論理デバイス 9 0 0 を特定するための識別子を格納する。論理デバイス 9 0 0 を特定するための識別子は、例えば、論理デバイス 9 0 0 の B u s 番号、デバイス番号若しくは機能番号、又はこれらの組み合わせが考えられる。

【 0 0 5 6 】

デバイス識別子 8 4 2 は、第二副論理区画 9 2 が備えるデバイス 5 3 0 を特定するための識別子を格納する。デバイス 5 3 0 を特定するための識別子は、例えば、デバイス 5 3 0 の B u s 番号、デバイス番号若しくは機能番号、又はこれらの組み合わせが考えある。

【 0 0 5 7 】

図 6 は、本発明の実施形態の変換テーブル 8 2 4 の一例を示す説明図である。

【 0 0 5 8 】

変換テーブル 8 2 4 は、第一副論理区画 9 1 が備える論理デバイス 9 0 0 に対する命令と、第二副論理区画 9 2 が備えるデバイス 5 3 0 に対する命令との対応関係を格納する。変換テーブル 8 2 4 は、論理命令 8 3 1 と実命令 8 3 2 とを含む。

【 0 0 5 9 】

論理命令 8 3 1 は、第一副論理区画 9 1 が備える論理デバイス 9 0 0 に対する命令を格納する。実命令 8 3 2 は、論理命令 8 3 1 に格納される命令に対応する第二副論理区画 9 2 が備えるデバイス 5 3 0 の命令を格納する。

【 0 0 6 0 】

10

20

30

40

50

図 6 に示すように、論理命令 8 3 1 に格納される 1 つの命令が複数の命令に展開される場合は、実命令 8 3 2 の対応する行に複数の命令が格納される。

【 0 0 6 1 】

図 7 は、本発明の実施形態における、論理区画 7 0 の生成処理を説明するフローチャートである。

【 0 0 6 2 】

ユーザが、入出力装置 1 5 を用いて論理区画生成要求を管理プログラム 2 0 に送信する (ステップ 2 0 1)。論理区画生成要求には、論理区画 7 0 を生成するために必要なスペック情報 (CPU やメモリのリソース量、デバイス種別や数など) と、デバイスの差異隠蔽が必要か否かを示す情報とが含まれる。

10

【 0 0 6 3 】

以下、論理区画 7 0 を生成するために必要なスペック情報をスペック情報と記載し、デバイスの差異隠蔽が必要か否かを示す情報を差異隠蔽可否情報と記載する。

【 0 0 6 4 】

なお、スペック情報及び差異隠蔽可否情報を設定するための UI (User Interface) については、図 1 2 を用いて後述する。

【 0 0 6 5 】

論理区画生成要求を受信した管理プログラム 2 0 は、当該論理区画生成要求に含まれるスペック情報と、差異隠蔽可否情報とを取得する。

【 0 0 6 6 】

管理プログラム 2 0 は、取得されたスペック情報及び差異隠蔽可否情報を含む論理区画生成指示を論理分割プログラム 6 0 に送信する (ステップ 2 0 2)。

20

【 0 0 6 7 】

論理分割プログラム 6 0 は、論理区画生成指示を受信し、受信した論理区画生成指示に含まれる差異隠蔽可否情報に基づいて、差異隠蔽が必要か否かを判定する (ステップ 2 0 3)。

【 0 0 6 8 】

差異隠蔽が必要であると判定された場合、論理分割プログラム 6 0 は、論理区画生成指示に含まれるスペック情報に基づいて、第一副論理区画 9 1 と第二副論理区画 9 2 とから構成される論理区画 7 0 を生成し (ステップ 2 0 4)、論理区画 7 0 の生成が完了したことを示す応答を管理プログラム 2 0 に送信する。

30

【 0 0 6 9 】

なお、ステップ 2 0 4 における論理区画 7 0 の生成処理の詳細については、図 8 を用いて後述する。

【 0 0 7 0 】

差異隠蔽が必要でないとして判定された場合、論理分割プログラム 6 0 は、論理区画生成指示に含まれるスペック情報に基づいて、論理区画 7 0 を生成し (ステップ 2 0 6)、論理区画 7 0 の生成が完了したことを示す応答を管理プログラム 2 0 に送信する。

【 0 0 7 1 】

なお、ステップ 2 0 6 において生成される論理区画 7 0 には、第一副論理区画 9 1 及び第二副論理区画 9 2 は含まれない。

40

【 0 0 7 2 】

管理プログラム 2 0 は、論理分割プログラム 6 0 から論理区画 7 0 の生成が完了したことを示す応答を受信し、論理区画の生成が完了したことをユーザに通知し (ステップ 2 0 7)、処理を終了する。

【 0 0 7 3 】

なお、第一副論理区画 9 1 に業務プログラム 8 1 をインストールする処理は、ユーザが実行してもよいし、管理プログラム 2 0 が実行してもよい。

【 0 0 7 4 】

図 8 は、本発明の実施形態の論理分割プログラム 6 0 が実行する論理区画 7 0 の生成処

50

理を説明するフローチャートである。

【0075】

論理分割プログラム60は、管理プログラム20から受信したスペック情報に基づいて、論理区画70、第一副論理区画91及び第二副論理区画92の定義をメモリ52上に格納する(ステップ2041)。

【0076】

論理分割プログラム60は、取得されたスペック情報に基づいて、CPU51及びメモリ52を第一副論理区画91と第二副論理区画92とにそれぞれ割り当てる(ステップ2042)。

【0077】

論理分割プログラム60は、取得されたスペック情報に基づいて、第二副論理区画92にNIC、HBA、CNA及びチップセット等のデバイス530を割り当てる(ステップ2043)。なお、割り当てられるデバイス53は、物理的に存在する実デバイスである。

【0078】

論理分割プログラム60は、第二副論理区画92のメモリ521に、差異隠蔽ソフトウェア82をロードする(ステップ2044)。

【0079】

差異隠蔽ソフトウェア82をロードする方法は、例えば、メモリ52にあらかじめ差異隠蔽ソフトウェア82を格納しておき、メモリ52から取得してもよいし、補助記憶装置(図示省略)又は外部の記憶装置(図示省略)から取得してもよいし、または、ネットワークを介して取得してもよい。

【0080】

論理分割プログラム60は、第二副論理区画92に割り当てられたデバイス530と対応する論理デバイス900を生成し、生成された論理デバイス900を第一副論理区画91に割り当てる(ステップ2045)。

【0081】

論理分割プログラム60は、デバイスマップテーブル823及び変換テーブル824を更新する(ステップ2046)。

【0082】

具体体には、第一副論理区画91に割り当てられた論理デバイス900と第二副論理区画92に割り当てられたデバイス530との対応関係を第二副論理区画92のデバイスマップテーブル823に格納する。また、論理デバイス900に用いられる命令と、デバイス530に用いられる命令との対応関係を取得して、取得された対応関係を変換テーブル824に格納する。論理デバイス900に用いられる命令及びデバイス530に用いられる命令は、例えば、あらかじめメモリ52に格納しておき、必要に応じてメモリ52から取得してもよい。

【0083】

なお、プラグアンドプレイなどの方法によって、第二副論理区画92の起動後、動的にデバイス530の割り当てが追加された場合、論理分割プログラム60は、新たに割り当てられたデバイス530に対応する論理デバイス900を生成して、生成された論理デバイス900を第一副論理区画91に割り当てる。また、第二副論理区画92からデバイス530が削除された場合、論理分割プログラム60は、削除されたデバイス530に対応する論理デバイス900を第一副論理区画91から削除する。

【0084】

図9は、本発明の実施形態における論理区画70の起動処理を説明するフローチャートである。

【0085】

論理分割プログラム60は、管理プログラム20から起動対象となる論理区画70の識別情報とともに、起動要求を受け付ける(ステップ600)。論理区画70の識別情報は

10

20

30

40

50

、例えば、論理区画 70 の論理区画名が考えられる。

【0086】

起動要求を受け付けた論理分割プログラム 60 は、起動対象の論理区画 70 に含まれる第一副論理区画 91 及び第二副論理区画 92 の起動処理を開始する（ステップ 601）。

【0087】

論理分割プログラム 60 は、第一副論理区画 91 及び第二副論理区画 92 をそれぞれ起動させる（ステップ 911、ステップ 921）。

【0088】

起動した第二副論理区画 92 は、制御プログラム 821 の起動を開始し（ステップ 922）、起動が完了すると論理分割プログラム 60 に制御プログラム 821 の起動完了通知を送信する（ステップ 923）。 10

【0089】

起動した第一副論理区画 91 は、業務プログラム 81 の論理ドライバ 813 の組み込み処理を開始する（ステップ 912）。

【0090】

第一副論理区画 91 の論理ドライバ 813 は、論理デバイス 900 にアクセスすることによって、論理分割プログラム 60 に対して割り込み先となる第二副論理区画 92 に割り当てられた CPU 511 の識別子を問い合わせる（ステップ 913）。ここで、論理ドライバ 813 が論理デバイス 900 にアクセス方法は、論理デバイス 900 に対して特殊な命令を指示する方法等が考えられる。 20

【0091】

論理分割プログラム 60 は、論理デバイス 900 への問い合わせをトラップする（ステップ 602）。

【0092】

論理分割プログラム 60 は、第二副論理区画 92 の制御プログラム 821 の起動が完了を確認する（ステップ 603）。

【0093】

具体的には、論理分割プログラム 60 は、第二副論理区画 92 から制御プログラム 821 の起動完了通知を受信したか否かを判定する。第二副論理区画 92 から制御プログラム 821 の起動完了通知を受信していないと判定された場合、論理分割プログラム 60 は、制御プログラム 821 の起動完了が通知されるまで次に処理の実行を待機させる。 30

【0094】

第二副論理区画 92 から制御プログラム 821 の起動完了通知を受信していると判定された場合、論理分割プログラム 60 は、第二副論理区画 92 に割り当てられた CPU 511 の識別子を第一副論理区画 91 の論理ドライバ 813 に送信する（ステップ 604）。

【0095】

第一副論理区画 91 は、論理分割プログラム 60 から割り込み先の CPU 51 の識別子を受信し、第一副論理区画 91 のメモリ 520 上に当該 CPU 51 の識別子を格納し（ステップ 914）、論理ドライバ 813 の組み込み処理を完了する（ステップ 915）。 40

【0096】

一方、第二副論理区画 92 は、論理分割プログラム 60 に対して、割り込み先となる第一副論理区画 91 の CPU 510 の識別子を問い合わせる（ステップ 924）。具体的には、制御プログラム 821 が、論理分割プログラム 60 に割り込み先となる第一副論理区画 91 の CPU 510 の識別子を問い合わせる。

【0097】

論理分割プログラム 60 は、第二副論理区画 92 から受け付けた問い合わせに対して、第一副論理区画 91 に割り当てられた CPU 510 の識別子を送信する（ステップ 605）。具体的には、論理分割プログラム 60 は、制御プログラム 821 に第一副論理区画 91 に割り当てられた CPU 510 の識別子を送信する。 50

【0098】

第二副論理区画 9 2 の制御プログラム 8 2 1 は、割り込み先の CPU 5 1 0 の識別子を受信する（ステップ 9 2 5）。

【0099】

なお、図示しないが、第二副論理区画 9 2 は、制御プログラム 8 2 1 と論理分割プログラム 6 0 との間で情報を送受信するため（ステップ 9 2 3、9 2 4、9 2 5）の I/F を備える。

【0100】

図 1 0 は、本発明の実施形態の論理区画 7 0 におけるデバイスに対する命令実行の処理を説明するフローチャートである。

【0101】

OS 8 1 2 は、論理ドライバ 8 1 3 に対して命令実行を指示する（ステップ 7 1 0 1）。すなわち、論理デバイス 9 0 0 に命令（論理命令）が発行される。

【0102】

論理ドライバ 8 1 3 は、論理命令を取得（トラップ）する（ステップ 7 1 0 2）。

【0103】

論理ドライバ 8 1 3 は、取得された論理命令から論理デバイス 9 0 0 の識別子、当該論理命令の識別子及び当該論理命令の引数データを取得し、取得された情報を共有メモリ 5 4 0 に格納する（ステップ 7 1 0 3）。

【0104】

論理ドライバ 8 1 3 は、CPU 間割り込みを発生させ、第二副論理区画 9 2 の制御プログラム 8 2 1 に当該論理命令を送信する（ステップ 7 1 0 4）。

【0105】

ここで、CPU 間割り込みの発生先である CPU は、図 9 のステップ 9 1 3 において取得された第二副論理区画 9 2 に割り当てられた CPU 5 1 1 である。なお、CPU 間割り込みの発生先を CPU 5 1 1 に決定する方法としては、論理ドライバ 8 1 3 が論理分割プログラム 6 0 から取得しておく方法、論理分割プログラム 6 0 が CPU 間割り込みをトラップして、CPU の識別子を変換する方法が考えられる。

【0106】

また、CPU 間割り込みの代わりに、論理ドライバ 8 1 3 が共有メモリ 5 4 0 の特定領域にデータを書き込み、当該メモリ領域を制御プログラム 8 2 1 がポーリングして検知してもよい。

【0107】

制御プログラム 8 2 1 は、共有メモリ 5 4 0 から第二副論理区画 9 2 から受信した論理命令に関する情報を取得する（ステップ 7 1 0 5）。具体的には、論理デバイス 9 0 0 の識別子及び論理命令の識別子が取得される。

【0108】

制御プログラム 8 2 1 は、論理命令をデバイス 5 3 0 に対する命令（実命令）に変換する（ステップ 7 1 0 6）。

【0109】

具体的には、制御プログラム 8 2 1 は、取得された論理デバイス 9 0 0 の識別子、取得された命令の識別子、及びデバイスマップテーブル 8 2 3 に基づいて、論理デバイス 9 0 0 に対応する第二副論理区画 9 2 のデバイス 5 3 0 を特定する。

【0110】

さらに、制御プログラム 8 2 1 は、特定されたデバイス 5 3 0 に対応する変換テーブル 8 2 4 を参照して、論理命令を実命令に変換する。

【0111】

制御プログラム 8 2 1 は、特定されたデバイス 5 3 0 に対応する実ドライバ 8 2 2 に対して、変換された命令の実行を指示する。すなわち、特定されたデバイス 5 3 0 に対応する実ドライバ 8 2 2 に実命令が発行される。

【0112】

10

20

30

40

50

実ドライバ 8 2 2 は、実命令を実行し（ステップ 7 1 0 8）、当該実命令の実行が完了したことを検知すると、当該命令の実行に対する応答を制御プログラム 8 2 1 に送信する（ステップ 7 1 0 9）。

【 0 1 1 3 】

制御プログラム 8 2 1 は、CPU 間割り込みを発生させ、第一副論理区画 9 1 の OS 8 1 2 に命令完了を通知する（ステップ 7 1 1 0）。

【 0 1 1 4 】

ここで、CPU 間割り込みの発生先である CPU は、第一副論理区画 9 1 に割り当てられた CPU 5 1 0 である。なお、CPU 間割り込みの発生先を CPU 5 1 1 に決定する方法としては、制御プログラム 8 2 1 が論理分割プログラム 6 0 から取得しておく方法、論理分割プログラム 6 0 が CPU 間割り込みをトラップして、CPU の識別子を変換する方法が考えられる。

10

【 0 1 1 5 】

OS 8 1 2 は、制御プログラム 8 2 1 からの命令完了を検知し（ステップ 7 1 1 1）、処理を終了する。

【 0 1 1 6 】

図 1 1 は、本発明の実施形態の論理区画 7 0 におけるデバイスに対するデータの受信処理を説明するフローチャートである。

【 0 1 1 7 】

制御プログラム 8 2 1 は、第二副論理区画 9 2 に割り当てられたデバイス 5 3 0 からの割り込みを検知する（ステップ 7 2 0 1）。すなわち、デバイス 5 3 0 に対する受信命令が検知される。

20

【 0 1 1 8 】

実ドライバ 8 2 2 は、割り込みが検知されたデバイス 5 3 0 からデータをロードし、共有メモリ 5 4 0 に格納する（ステップ 7 2 0 2）。

【 0 1 1 9 】

制御プログラム 8 2 1 は、変換テーブル 8 2 4 を参照してデバイス 5 3 0 に対する受信命令を論理デバイス 9 0 0 に対する命令に変換し、変換された命令を共有メモリ 5 4 0 に格納するとともに、CPU 間割り込みを発生させ、第一副論理区画 9 1 の OS 8 1 2 にデータ受信を通知する（ステップ 7 2 0 3）。

30

【 0 1 2 0 】

ここで、CPU 間割り込みの発生先である CPU は、第一副論理区画 9 1 に割り当てられた CPU 5 1 0 である。なお、CPU 間割り込みの発生先を CPU 5 1 0 に決定する方法としては、制御プログラム 8 2 1 が論理分割プログラム 6 0 から取得しておく方法、論理分割プログラム 6 0 が CPU 間割り込みをトラップして、CPU の識別子を変換する方法が考えられる。

【 0 1 2 1 】

制御プログラム 8 2 1 からデータ受信の通知を受けた OS 8 1 2 は、割り込みを検知し、共有メモリ 5 4 0 から論理デバイス 9 0 0 に対する命令を取得し、取得された命令に基づいて、当該割り込みに対する処理を論理ドライバ 8 1 3 に実行するように指示する（ステップ 7 2 0 4）。

40

【 0 1 2 2 】

論理ドライバ 8 1 3 は、共有メモリ 5 4 0 からステップ 7 2 0 2 において保存されたデータを読み出す（ステップ 7 2 0 5）。

【 0 1 2 3 】

なお、論理分割プログラム 6 0 は、デバイス 5 3 0 の DMA (Direct Memory Access) によってメモリ 5 2 1 にデータが保存される先となるメモリアドレスを、共有メモリ 5 4 0 のアドレスとなるように制御してもよい。

【 0 1 2 4 】

本発明では、第二副論理区画 9 2 が、論理ドライバ 8 1 3 が受け取った命令を実ドライ

50

バ 8 2 2 の命令に変換する処理（ステップ 7 1 0 5 ~ ステップ 7 1 1 0）、及び、実ドライバ 8 2 2 が受け取った命令を論理ドライバ 8 1 3 の命令に変換する処理（ステップ 7 2 0 1 ~ ステップ 7 2 0 3）を実行している。すなわち、第二副論理区画 9 2 が、論理デバイス 9 0 0 とデバイス 5 3 0 との間のエミュレーション処理を実行する。

【 0 1 2 5 】

これによって、第一副論理区画 9 1 に割り当てられるデバイスを仮想化することが可能となる。したがって、サーバ 5 0 のデバイス 5 3 が変更されても、第一副論理区画 9 1 上で実行される業務プログラム 8 1 は影響を受けることなく実行される。

【 0 1 2 6 】

また、業務プログラム 8 1 は第一副論理区画 9 1 に割り当てられたリソースのみを用いて実行され、また、差異隠蔽ソフトウェア 8 2 は第二副論理区画 9 2 に割り当てられたリソースのみを用いて実行されるため、業務プログラム 8 1 及び差異隠蔽ソフトウェア 8 2 は、互いに干渉することなく実行される。

10

【 0 1 2 7 】

したがって、第一副論理区画 9 1 には前述したエミュレーション処理に伴う遅延等が発生しない。すなわち、第一副論理区画 9 1 は業務プログラム 8 1 のみを実行するため、業務プログラム 8 1 の実行性能が安定化する。さらに、第一副論理区画 9 1 には論理デバイス 9 0 0 が割り当てられているため、サーバ 5 0 のデバイス 5 3 の変更の影響を受けることなく業務プログラムを実行できる。

【 0 1 2 8 】

図 1 2 は、本発明の実施形態における、論理区画 7 0 生成のために表示される U I (U s e r I n t e r f a c e) の一例を示す説明図である。

20

【 0 1 2 9 】

論理区画 7 0 生成のための U I 2 5 0 は、ステップ 2 0 1 で用いられる。

【 0 1 3 0 】

U I 2 5 0 は、管理サーバ 1 0 に接続される入出力装置 1 5、又は管理サーバ 1 0 とネットワークを介して接続される他の端末の表示装置等に、ブラウザ、専用のプログラム、及びテキストなどを用いて表示される。

【 0 1 3 1 】

U I 2 5 0 は、論理区画名入力エリア 2 5 7、必要スペック入力エリア 2 5 1、差異隠蔽選択エリア 2 5 2、差異隠蔽スペック入力エリア 2 5 3、必要総リソース表示エリア 2 5 4、決定ボタン 2 5 5 及びキャンセルボタン 2 5 6 を含む。

30

【 0 1 3 2 】

論理区画名入力エリア 2 5 7 は、生成される論理区画 7 0 を識別するための名称を入力するエリアである。

【 0 1 3 3 】

必要スペック入力エリア 2 5 1 は、業務プログラム 8 1 を稼働させる論理区画 7 0 を生成するために必要なスペックを入力するエリアである。具体的には、CPU の数（コア数を含む）、メモリ容量、並びに、NIC 及び H B A の性能及び数等が入力される。

【 0 1 3 4 】

後述する差異隠蔽選択エリア 2 5 2 のチェックボックスがチェックされている場合には、必要スペック入力エリア 2 5 1 には第一副論理区画 9 1 を生成するためのスペックが入力される。

40

【 0 1 3 5 】

差異隠蔽選択エリア 2 5 2 は、サーバ 5 0 のデバイス 5 3 の変化を隠蔽する差異隠蔽が必要か否かを選択するエリアである。

【 0 1 3 6 】

図 1 2 に示す例では、ユーザは、差異隠蔽が必要である場合には差異隠蔽選択エリア 2 5 2 のチェックボックスをチェックし、差異隠蔽が不要な場合には差異隠蔽選択エリア 2 5 2 のチェックボックスのチェックをはずす。

50

【0137】

差異隠蔽選択エリア252のチェックボックスがチェックされている場合、第一副論理区画91と第二副論理区画92とを含む論理区画70が生成される。また、差異隠蔽選択エリア252のチェックボックスがチェックされていない場合、通常の論理区画70が生成される。

【0138】

差異隠蔽スペック入力エリア253は、第二副論理区画92を生成するために必要なスペックを入力するエリアである。

【0139】

具体的には、CPUの数(コア数を含む)及びメモリ容量が入力される。

10

【0140】

なお、差異隠蔽スペック入力エリア253の表示方法としては、差異隠蔽選択エリア252のチェックボックスがチェックされた場合に差異隠蔽スペック入力エリア253への入力を可能とする方法、差異隠蔽選択エリア252のチェックボックスがチェックされた場合に、差異隠蔽スペック入力エリア253を表示する方法が考えられる。

【0141】

必要総リソース表示エリア254は、必要スペック入力エリア251に入力されたスペックと、差異隠蔽スペック入力エリア253に入力されたスペックとを合わせた、論理区画70を生成するための全リソースを表示する。

20

【0142】

なお、必要総リソース表示エリア254の表示方法は、常に必要総リソース表示エリア254を表示する方法、差異隠蔽選択エリア252のチェックボックスがチェックされた場合に、必要総リソース表示エリア254を表示する方法が考えられる。

【0143】

なお、ユーザが容易に差異隠蔽を実現できるように、差異隠蔽スペック入力エリア253及び必要総リソース表示エリア254が表示されないUI250であってもよい。

【0144】

決定ボタン255は、各入力エリアに入力した情報に基づいて論理区画70の生成を実行する場合に操作するボタンである。

【0145】

キャンセルボタン256は、論理区画70の生成を中止する場合に操作するボタンである。

30

【0146】

差異隠蔽選択エリア252がチェックされている場合、必要スペック入力エリア251及び差異隠蔽スペック入力エリア253に入力された情報が、管理プログラム20に送信されるスペック情報となる。

【0147】

図13は、本発明の実施形態の論理区画70にユーザがアクセスする場合の処理について説明するブロック図である。

【0148】

ユーザは、管理サーバ10に接続される入出力装置15を用いて、論理区画名が「AAA」の論理区画70のコンソールにアクセスする。

40

【0149】

ユーザからのアクセスを検知した管理プログラム20は、論理分割プログラム60に論理区画70の識別子として論理区画名の「AAA」を送信する。

【0150】

論理区画名の「AAA」を受信した論理分割プログラム60は、受信した論理区画名が「AAA」である論理区画70に含まれる第一副論理区画91のコンソールと接続する。これによって、ユーザは、第一副論理区画91にアクセスできる。すなわち、ユーザは、第一副論理区画91を論理区画70として認識する。これによって、ユーザの管理を容易

50

にすることが可能となる。

【 0 1 5 1 】

本発明の一形態によれば、ユーザに第二副論理区画 9 2 を意識させることなく、第一副論理区画 9 1 のデバイスを仮想化することが可能となる。したがって、サーバ 5 0 のデバイス 5 3 が変更された場合、デバイス 5 3 変更の影響を受けることなく業務プログラム 8 1 の実行を継続することができる。

【 0 1 5 2 】

また、論理デバイス 9 0 0 とデバイス 5 3 0 との間の変換処理は第二副論理区画 9 2 によって実行されるため、第一副論理区画 9 1 上で実行される業務プログラム 8 1 の実行性能を安定化させることができる。

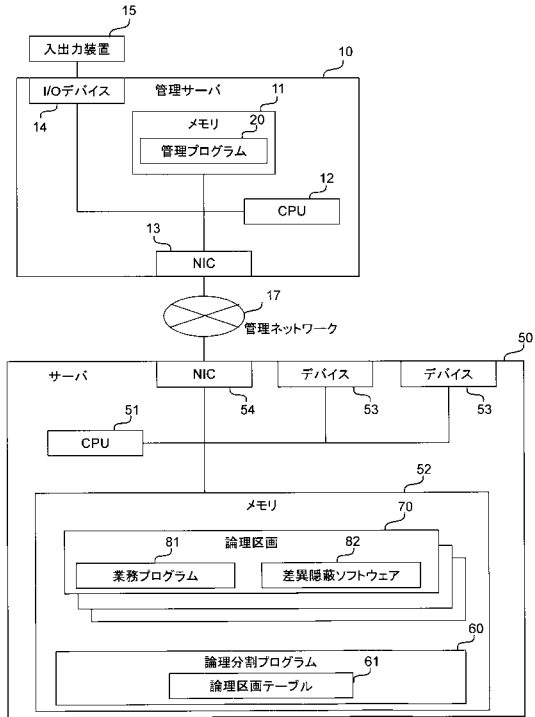
10

【符号の説明】

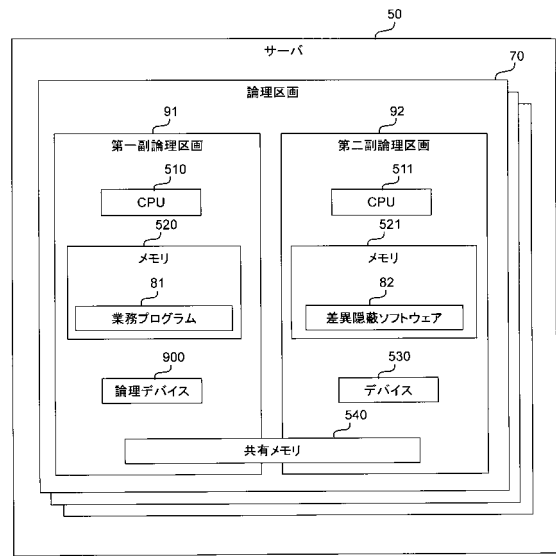
【 0 1 5 3 】

1 0	管理サーバ	
1 1	メモリ	
1 2	C P U	
1 3	N I C	
1 4	I / O デバイス	
1 5	入出力装置	
1 7	管理ネットワーク	
2 0	管理プログラム	20
5 0	サーバ	
5 1	C P U	
5 2	メモリ	
5 3	デバイス	
5 4	N I C	
6 0	論理分割プログラム	
6 1	論理区画テーブル	
7 0	論理区画	
8 1	業務プログラム	
8 2	差異隠蔽ソフトウェア	30
9 1	第一副論理区画	
9 2	第二副論理区画	
5 1 0	C P U	
5 1 1	C P U	
5 2 0	メモリ	
5 2 1	メモリ	
5 3 0	デバイス	
8 1 1	アプリケーション	
8 1 2	O S	
8 1 3	論理ドライバ	40
8 2 1	制御プログラム	
8 2 2	実ドライバ	
8 2 3	デバイスマップテーブル	
8 2 4	変換テーブル	
9 0 0	論理デバイス	

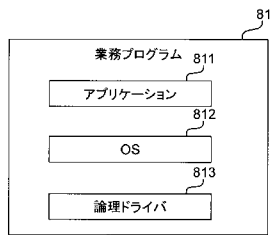
【 図 1 】



【 図 2 】



【 図 3 】

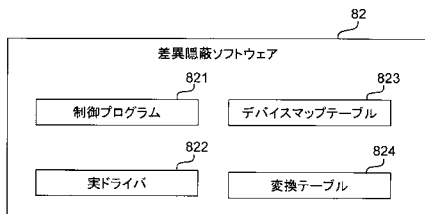


【 図 5 】

デバイスマップテーブル

841 論理デバイス識別子	842 デバイス識別子
論理デバイス1	デバイスA
論理デバイス2	デバイスB
論理デバイス3	デバイスC
....

【 図 4 】

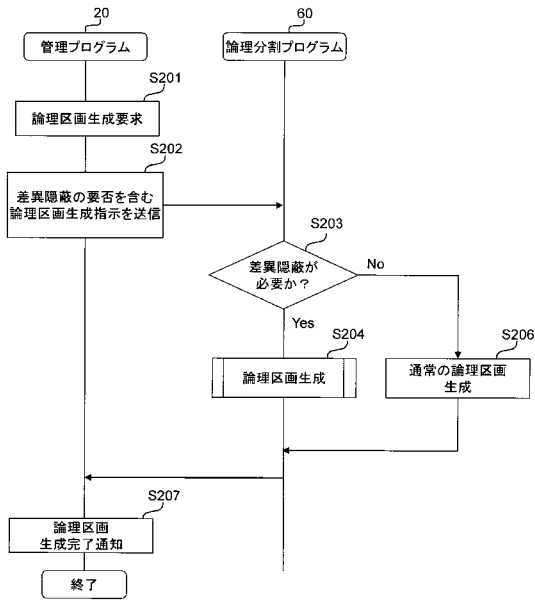


【 図 6 】

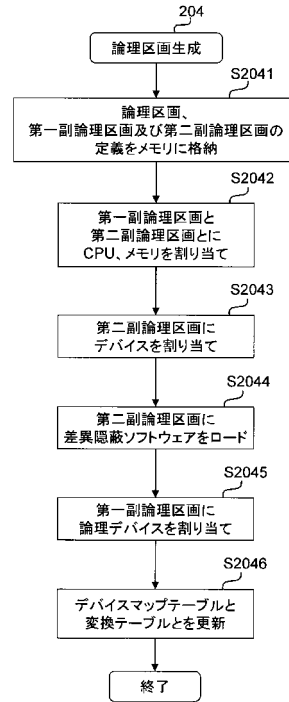
変換テーブル

831 論理命令	832 実命令
Open()	Read()
Close()	Write()
Test(a,b)	Set(a) Set(b) Sub(a,b)
....

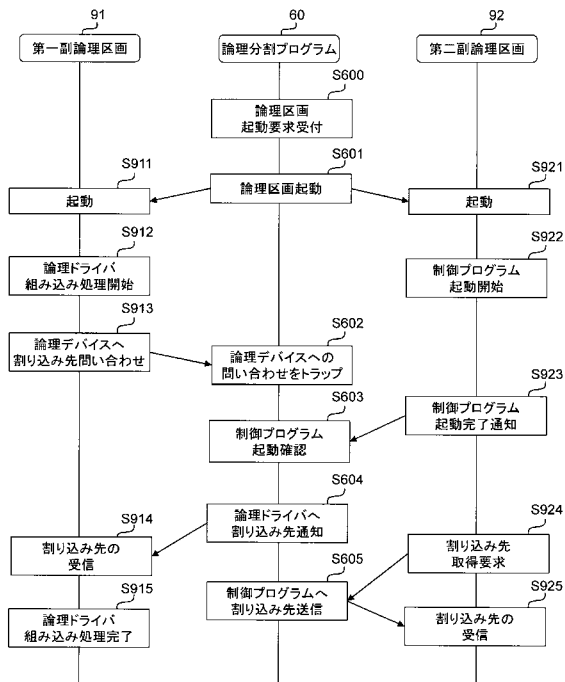
【 図 7 】



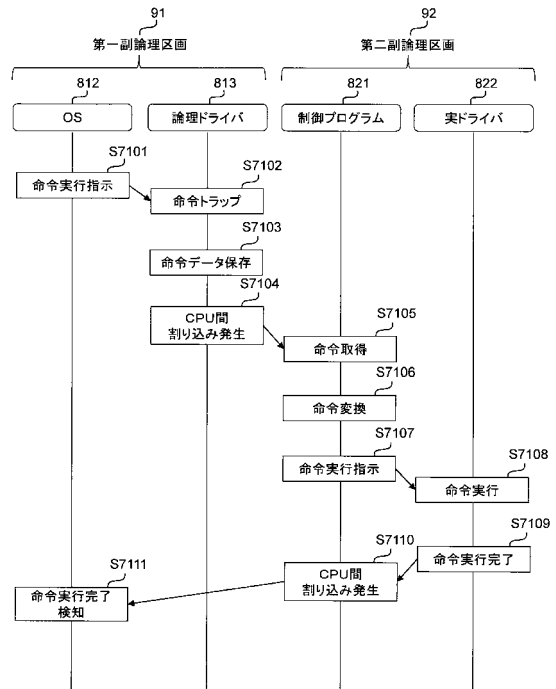
【 図 8 】



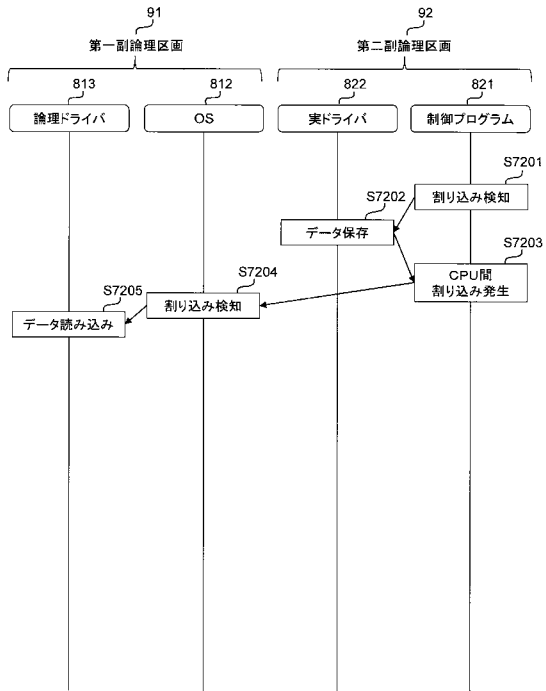
【 図 9 】



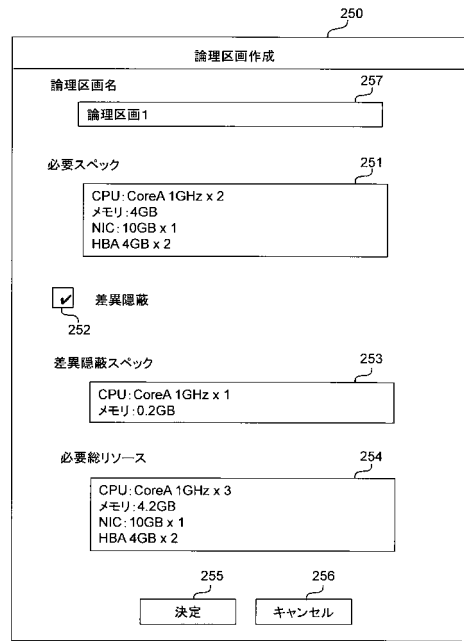
【 図 10 】



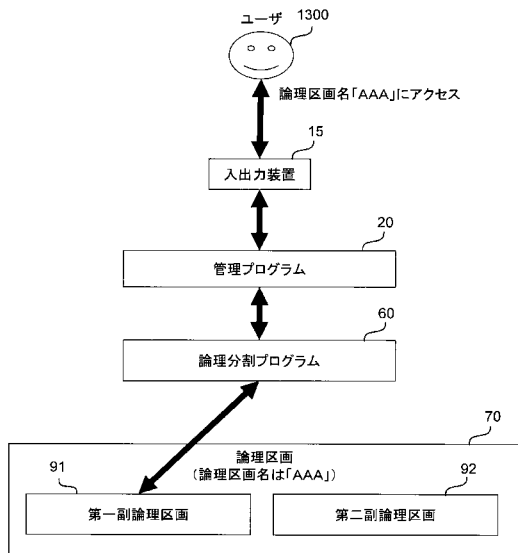
【 図 1 1 】



【 図 1 2 】



【 図 1 3 】



フロントページの続き

(72)発明者 関口 知紀

神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内