

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6458146号
(P6458146)

(45) 発行日 平成31年1月23日(2019.1.23)

(24) 登録日 平成30年12月28日(2018.12.28)

(51) Int.Cl.		F I			
G06F	9/50	(2006.01)	G06F	9/50	120Z
G06F	9/54	(2006.01)	G06F	9/54	A
G06F	9/455	(2006.01)	G06F	9/455	150

請求項の数 6 (全 16 頁)

(21) 出願番号	特願2017-527033 (P2017-527033)	(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
(86) (22) 出願日	平成27年7月8日(2015.7.8)	(74) 代理人	110001678 特許業務法人藤央特許事務所
(86) 国際出願番号	PCT/JP2015/069628	(72) 発明者	八田 ゆかり 東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
(87) 国際公開番号	W02017/006458	(72) 発明者	早川 典充 東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
(87) 国際公開日	平成29年1月12日(2017.1.12)	(72) 発明者	戸塚 崇夫 東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内
審査請求日	平成29年12月5日(2017.12.5)		

最終頁に続く

(54) 【発明の名称】 計算機及びメモリ領域管理方法

(57) 【特許請求の範囲】

【請求項1】

複数のオペレーティングシステムが稼働する計算機であって、
前記計算機は、計算機リソースとして、プロセッサ、前記プロセッサに接続されるメモリ、及び前記プロセッサに接続されるI/Oデバイスを備え、

前記計算機は、複数の記憶媒体を有するストレージ装置と接続し、
前記複数のオペレーティングシステムは、前記ストレージ装置に対するアクセスを制御する第1のオペレーティングシステム、及び少なくとも一つの仮想計算機を生成する第2のオペレーティングシステムを含み、

前記第1のオペレーティングシステム及び前記第2のオペレーティングシステムの各々には、占有的に使用できるように分割された前記計算機リソースが割り当てられ、

前記少なくとも一つの仮想計算機上では、アプリケーションを実行する第3のオペレーティングシステムが稼働し、

前記第2のオペレーティングシステムは、前記アプリケーションと前記第1のオペレーティングシステムとの間で通信するために用いられるメモリ領域である共有領域を管理する共有領域管理部と、前記アプリケーションに割り当てられる仮想的なメモリの位置を示すゲスト仮想アドレス空間に、前記共有領域をマッピングするためのアドレスを通知する処理部と、を有し、

前記第3のオペレーティングシステムは、前記アプリケーションからの要求に応じて前記第2のオペレーティングシステムに対して前記共有領域の確保を要求し、前記ゲスト仮

10

20

想アドレス空間に確保された前記共有領域をマッピングするエージェントを有し、

前記共有領域管理部は、

前記エージェントから前記共有領域の確保要求を受け付けた場合、所定のサイズの前記共有領域を確保し、

前記仮想計算機に含まれる論理メモリの物理的な位置を示すゲスト物理アドレス空間における、前記確保された共有領域にアクセスするための第1のアドレスを前記エージェントに通知し、

前記エージェントは、

前記第1のアドレスを用いて、前記ゲスト仮想アドレス空間に前記確保された共有領域をマッピングし、

前記ゲスト仮想アドレス空間にマッピングされた前記共有領域にアクセスするための第2のアドレスを前記アプリケーションに通知し、

前記アプリケーションは、前記第2のアドレスを受け付けた後、前記エージェントを介して前記処理部に前記共有領域を用いた処理の開始要求を送信し、

前記処理部は、

前記共有領域管理部から、前記第2のオペレーティングシステムが管理するホスト物理アドレス空間における、前記確保された共有領域にアクセスするための第3のアドレスを取得し、

前記第1のオペレーティングシステムに前記第3のアドレスを通知し、

前記第1のオペレーティングシステムは、前記第3のアドレスを用いて前記アプリケーションと通信することによって、前記共有領域を用いた処理を実行することを特徴とする計算機。

【請求項2】

請求項1に記載の計算機であって、

前記第2のオペレーティングシステムは、前記仮想計算機の起動時に、当該仮想計算機に割り当てられるメモリ領域の一部を前記共有領域として割り当てることを特徴とする計算機。

【請求項3】

請求項2に記載の計算機であって、

前記共有領域管理部は、

前記共有領域を複数のブロックに分割して管理し、

前記エージェントから前記共有領域の確保要求を受け付けた場合、少なくとも一つのブロックを前記アプリケーションに提供する共有領域として確保することを特徴とする計算機。

【請求項4】

複数のオペレーティングシステムが稼働する計算機におけるメモリ領域管理方法であって、

前記計算機は、計算機リソースとして、プロセッサ、前記プロセッサに接続されるメモリ、及び前記プロセッサに接続されるI/Oデバイスを備え、

前記計算機は、複数の記憶媒体を有するストレージ装置と接続し、

前記複数のオペレーティングシステムは、前記ストレージ装置に対するアクセスを制御する第1のオペレーティングシステム、及び少なくとも一つの仮想計算機を生成する第2のオペレーティングシステムを含み、

前記第1のオペレーティングシステム及び前記第2のオペレーティングシステムの各々には、占有的に使用できるように分割された前記計算機リソースが割り当てられ、

前記少なくとも一つの仮想計算機上では、アプリケーションを実行する第3のオペレーティングシステムが稼働し、

前記メモリ領域管理方法は、

前記アプリケーションが、前記第3のオペレーティングシステムを介して前記アプリケーションと前記第1のオペレーティングシステムとの間で通信するために用いるメモリ領

10

20

30

40

50

域である共有領域の確保要求を発行するステップと、

前記第2のオペレーティングシステムが、前記共有領域の確保要求を受け付けた場合、
所定サイズの前記共有領域を確保するステップと、

前記第2のオペレーティングシステムが、前記仮想計算機に含まれる論理メモリの物理
的な位置を示すゲスト物理アドレス空間における、前記確保された共有領域にアクセスす
るための第1のアドレスを前記第3のオペレーティングシステムに通知するステップと、

前記第3のオペレーティングシステムが、前記第1のアドレスを用いて、前記アプリケ
ーションに割り当てられる仮想的なメモリの位置を示すゲスト仮想アドレス空間に前記確
保された共有領域をマッピングするステップと、

前記第3のオペレーティングシステムが、前記ゲスト仮想アドレス空間にマッピングさ
れた前記共有領域にアクセスするための第2のアドレスを前記アプリケーションに通知す
るステップと、

前記アプリケーションが、前記第2のアドレスを受け付けた後、前記第3のオペレー
ティングシステムを介して前記第2のオペレーティングシステムに前記共有領域を用いた処
理の開始要求を送信するステップと、

前記第2のオペレーティングシステムが、前記処理の開始要求を受け付けた場合、前記
第2のオペレーティングシステムが管理するホスト物理アドレス空間における、前記確保
された共有領域にアクセスするための第3のアドレスを取得するステップと、

前記第2のオペレーティングシステムが、前記第1のオペレーティングシステムに前記
第3のアドレスを通知するステップと、

前記第1のオペレーティングシステムが、前記第3のアドレスを用いて前記アプリケ
ーションと通信することによって、前記共有領域を用いた処理を実行するステップと、を
含むことを特徴とするメモリ領域管理方法。

【請求項5】

請求項4に記載のメモリ領域管理方法であって、

前記第2のオペレーティングシステムが、前記仮想計算機の起動時に、当該仮想計算
機に割り当てられるメモリ領域の一部を前記共有領域として割り当てるステップを含むこ
とを特徴とするメモリ領域管理方法。

【請求項6】

請求項5に記載のメモリ領域管理方法であって、

前記第2のオペレーティングシステムが、前記共有領域を複数のブロックに分割して管
理するステップを含み、

前記所定サイズの前記共有領域を確保するステップでは、前記第2のオペレーティング
システムが、少なくとも一つのブロックを前記アプリケーションに提供する共有領域とし
て確保することを特徴とするメモリ領域管理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、複数のホストOSが稼働する計算機システムにおいて、ホストOSと、当該
ホストOS又は他のホストOSが管理するゲストOSとが共有するメモリ領域の制御技術
に関する。

【背景技術】

【0002】

近年、物理計算機上のハードウェアリソースを有効に活用するために、一台の物理計算
機上で二つ以上のOS（オペレーティングシステム）を起動する技術が注目されている（
例えば、特許文献1及び特許文献2参照）。

【0003】

特許文献1には、「マルチオペレーティングシステム計算機は、優先度が設定された複
数のタスクを優先度順に実行する第1のOS（オペレーティングシステム）と、前記第1
のOSと異なる第2のOSとを動作OSとして交互に動作させるマルチオペレーティング

10

20

30

40

50

システム計算機であって、前記第1のOSが動作OSとして動作しているときに前記複数のタスクのうち所定の優先度が設定された所定のタスクであって動作OSの切替え契機を特定するための切替え契機タスクとして用いられるタスクが実行された場合に、動作OSを前記第1のOSから前記第2のOSに切り替えるOS切替え部」を備えることが記載されている。

【0004】

また、特許文献2には、「第1のOSと第2のOSとの少なくとも2つのOSを起動するマルチOS起動装置において、(1)メモリ領域を有し、前記メモリ領域に対してメモリ空間が定義される1次記憶部と、(2)セカンドブートローダと、前記第2のOSとを記憶する2次記憶部と、(3)CPUに対する制御情報を示すコンテキストであって前記第1のOS用のコンテキストである第1のコンテキストで動作する前記第1のOSのもとで動作するファーストブートローダを前記第1のコンテキストで動作中の前記第1のOSのもとで動作させることにより、前記第1のOSが管理する第1のメモリ空間として前記第1のコンテキストにより前記1次記憶装置に対して定義された前記1次記憶部のメモリ領域に前記2次記憶部から前記セカンドブートローダと前記第2のOSとを前記ファーストブートローダにロードさせるOS実行部と、(4)前記第1のメモリ空間として定義されたメモリ領域にロードされた前記セカンドブートローダを前記第1のコンテキストで動作中の前記第1のOSのもとで実行することにより前記第2のOSが管理する第2のメモリ空間として定義されるメモリ領域と前記セカンドブートローダ及び前記第2のOSがロードされたメモリ領域とを含む第3のメモリ空間を前記1次記憶部に対して定義する前記セカンドブートローダ用のコンテキストを前記セカンドブートローダに生成させると共に生成された前記セカンドブートローダ用のコンテキストへ前記第1のコンテキストから切り替えさせ、前記セカンドブートローダ用のコンテキストのもとで前記セカンドブートローダを実行することにより前記第3のメモリ空間に含まれる前記第2のメモリ空間として定義された前記1次記憶部のメモリ領域に前記ファーストブートローダによって前記1次記憶部のメモリ領域にロードされた前記第2のOSを前記セカンドブートローダにロードさせると共に前記第2のOS用のコンテキストを生成させ、生成された前記第2のOS用のコンテキストへ前記セカンドブートローダ用のコンテキストから切り替えさせ、前記第2のOS用のコンテキストのもとで前記セカンドブートローダに前記第2のOSの起動を実行させるロード実行部と」を備えることが記載されている。

【0005】

このマルチOS技術を利用し、例えば、第1のOS(ホストOS1)としてホストOS1を搭載し、第2のOS(ホストOS2)としてハイバイザを搭載し、さらにハイバイザ上の複数の論理計算機上のOS(ゲストOS1~ゲストOSn)としてサーバ制御OSを搭載することによって、省スペース化・低価格化が期待できる。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開平11-149385号公報

【特許文献2】国際公開第2009/113394号

【発明の概要】

【発明が解決しようとする課題】

【0007】

一つの物理計算機と不揮発性の記憶媒体とが接続され、物理計算機に不揮発性の記憶媒体へのアクセスを制御するホストOS1及び複数のゲストOSを管理するホストOS2を搭載する計算機システムを一例にして考える。

【0008】

各ゲストOSはI/O処理の応答性能を高めるために、データをキャッシングするためのメモリを設けている。メモリは揮発性であるため、不意の電源遮断が発生した場合にデータが失われないように、ホストOS1又はホストOS2は、不揮発性の記憶媒体にデー

10

20

30

40

50

タを退避する。電源が復旧し、ゲストOSが再起動する場合、退避されたデータをメモリに戻すためには、ゲストOS上のアプリケーションは、不揮発性の記憶媒体にアクセス可能なホストOS 1との間で通信を行いホストOS 1が出力したデータをメモリ上に戻す必要がある。

【0009】

前述したような障害復旧を実現するためには、ホストOS 1とゲストOS上のアプリケーションとが通信するためのメモリ領域（共有領域）を確保する必要がある。

【0010】

独立した計算機リソースを有する複数のホストOSが稼働する計算機において、一つのホストOSが管理する仮想計算機と、当該ホストOS又は他のホストOSとの間の通信制御に用いるメモリ領域を確保する計算機、及びメモリ領域の管理方法を提供する。

10

【課題を解決するための手段】

【0011】

本願において開示される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、複数のオペレーティングシステムが稼働する計算機であって、前記計算機は、計算機リソースとして、プロセッサ、前記プロセッサに接続されるメモリ、及び前記プロセッサに接続されるI/Oデバイスを備え、前記計算機は、複数の記憶媒体を有するストレージ装置と接続し、前記複数のオペレーティングシステムは、前記ストレージ装置に対するアクセスを制御する第1のオペレーティングシステム、及び少なくとも一つの仮想計算機を生成する第2のオペレーティングシステムを含み、前記第1のオペレーティングシステム及び前記第2のオペレーティングシステムの各々には、占有的に使用できるように分割された前記計算機リソースが割り当てられ、前記少なくとも一つの仮想計算機上では、アプリケーションを実行する第3のオペレーティングシステムが稼働し、前記第2のオペレーティングシステムは、前記アプリケーションと前記第1のオペレーティングシステムとの間で通信するために用いられるメモリ領域である共有領域を管理する共有領域管理部と、前記アプリケーションに割り当てられる仮想的なメモリの位置を示すゲスト仮想アドレス空間に、前記共有領域をマッピングするためのアドレスを通知する処理部と、を有し、前記第3のオペレーティングシステムは、前記アプリケーションからの要求に応じて前記第2のオペレーティングシステムに対して前記共有領域の確保を要求し、前記ゲスト仮想アドレス空間に確保された前記共有領域をマッピングするエージェントを有し、前記共有領域管理部は、前記エージェントから前記共有領域の確保要求を受け付けた場合、所定のサイズの前記共有領域を確保し、前記仮想計算機に含まれる論理メモリの物理的な位置を示すゲスト物理アドレス空間における、前記確保された共有領域にアクセスするための第1のアドレスを前記エージェントに通知し、前記エージェントは、前記第1のアドレスを用いて、前記ゲスト仮想アドレス空間に前記確保された共有領域をマッピングし、前記ゲスト仮想アドレス空間にマッピングされた前記共有領域にアクセスするための第2のアドレスを前記アプリケーションに通知し、前記アプリケーションは、前記第2のアドレスを受け付けた後、前記エージェントを介して前記処理部に前記共有領域を用いた処理の開始要求を送信し、前記処理部は、前記共有領域管理部から、前記第2のオペレーティングシステムが管理するホスト物理アドレス空間における、前記確保された共有領域にアクセスするための第3のアドレスを取得し、前記第1のオペレーティングシステムに前記第3のアドレスを通知し、前記第1のオペレーティングシステムは、前記第3のアドレスを用いて前記アプリケーションと通信することによって、前記共有領域を用いた処理を実行することを特徴とする。

20

30

40

【発明の効果】

【0012】

複数のオペレーティングシステム間でメモリ領域を共有し、当該メモリ領域を介してデータの送受信が可能となる。前述した以外の課題、構成及び効果は、以下の実施形態の説明によって明らかにされる。

【図面の簡単な説明】

50

【 0 0 1 3 】

【図 1】実施例 1 の計算機システムのハードウェア構成の一例を示す図である。

【図 2】実施例 1 の計算機システムの論理構成の一例を示す図である。

【図 3】実施例 1 の管理情報の一例を示す図である。

【図 4】実施例 1 の共有領域のマッピング関係及び構造を示す図である。

【図 5 A】実施例 1 の計算機システムの共有領域を用いた処理の流れを示すシーケンス図である。

【図 5 B】実施例 1 の計算機システムの共有領域を用いた処理の流れを示すシーケンス図である。

【図 6】実施例 1 のエージェントが実行するマッピング処理の一例を説明するフローチャートである。 10

【発明を実施するための形態】

【 0 0 1 4 】

以下、図面を用いて実施例を説明する。

【実施例 1】

【 0 0 1 5 】

図 1 は、実施例 1 の計算機システムのハードウェア構成の一例を示す図である。

【 0 0 1 6 】

計算機システムは、物理計算機 1 0 及び外部ストレージ装置 2 0 から構成される。物理計算機 1 0 は、外部ストレージ装置 2 0 と直接、又は、ネットワークを介して接続される。ネットワークとしては、FC (F i b r e C h a n n e l) で構成された SAN 等が考えられる。 20

【 0 0 1 7 】

物理計算機 1 0 は、プロセッサ 1 1 0、RAM 1 2 0、チップセット 1 3 0、FPGA 1 4 0、内部ストレージデバイス 1 5 0、電源部 1 6 0、及び I / O デバイス 1 7 0 を備える。

【 0 0 1 8 】

プロセッサ 1 1 0 は、RAM 1 2 0 に格納されるプログラム (命令) を読み出し、読み出されたプログラム (命令) を実行する。プロセッサ 1 1 0 は、複数の CPU コア 1 1 1 を有する。プロセッサ 1 1 0 がプログラムを実行することによって、OS 等の機能が実現される。以下、プログラムを主体に処理を説明する場合、当該プログラムがプロセッサ 1 1 0 によって実行されていることを示す。 30

【 0 0 1 9 】

RAM 1 2 0 は、プロセッサ 1 1 0 によって実行されるプログラム及び当該プログラムの実行に必要な情報を格納する。また、RAM 1 2 0 は、各プログラムのワークエリアを含む。

【 0 0 2 0 】

チップセット 1 3 0 は、プロセッサ 1 1 0、RAM 1 2 0、及び I / O デバイス 1 7 0 間のデータの入出力を管理する。FPGA 1 4 0 は、所定の演算を行う回路である。本実施例では、FPGA 1 4 0 は、データ転送等を行う。なお、FPGA 1 4 0 は、RAM 1 4 1 を含む。 40

【 0 0 2 1 】

内部ストレージデバイス 1 5 0 は、不揮発性の記憶媒体であり、例えば、HDD (H a r d D i s k D r i v e) 又は SSD (S o l i d S t a t e D r i v e) 等が考えられる。本実施例では、不意に電源が遮断した時に、RAM 1 4 1 に格納されるデータが内部ストレージデバイス 1 5 0 に格納される。なお、RAM 1 1 2 0 に格納されるデータが内部ストレージデバイス 1 5 0 に格納されてもよい。

【 0 0 2 2 】

電源部 1 6 0 は、物理計算機 1 0 の電源を制御する。I / O デバイス 1 7 0 は、外部装置と接続することによって、外部から情報を入力し、外部に情報を出力するためのデバイ 50

スである。I/Oデバイス170は、例えば、NIC、FC HBA等が考えられる。

【0023】

外部ストレージ装置20は、OSのプログラム、OS上で動作する各種アプリケーションのプログラムコード、及びアプリケーションが扱うデータ等を格納する。外部ストレージ装置20は、複数の記憶媒体を備える。本実施例では、外部ストレージ装置20は、記憶媒体として四つのHDD190を備える。なお、物理計算機10は、複数の記憶媒体、及びコントローラを備えるストレージシステムと接続されてもよい。

【0024】

図2は、実施例1の計算機システムの論理構成の一例を示す図である。

【0025】

本実施例の計算機システムでは、物理計算機10の計算機リソースが分割され、ホストOS1(200)及びホストOS2(210)のそれぞれに分割された計算機リソースが割り当てられる。計算機リソースには、RAM120が含まれる。本実施例では、RAM120のメモリ領域(記憶領域)が二つに分割され、ホストOS1(200)及びホストOS2(210)のそれぞれに分割されたメモリ領域が割り当てられる。以下の説明では、分割されたメモリ領域を分割RAMとも記載する。

【0026】

ホストOS1(200)に割り当てられた計算機リソースは、ホストOS1(200)によって占有的に使用され、ホストOS2(210)に割り当てられた計算機リソースは、ホストOS2(210)によって占有的に使用される。

【0027】

本実施例のホストOS1(200)は、ストレージ制御OSに対応する。ストレージ制御OSは、外部ストレージ装置20に対する読出処理及び書込処理を制御する。ホストOS1(200)は、ゲストOS230と内部ストレージデバイス150との間のバックアップ処理及びリストア処理を実行する。

【0028】

本実施例のホストOS2(210)は、ハイパバイザ等のVMM(Virtual Machine Monitor)に対応する。VMMは、仮想マシン技術を用いて、ホストOS2(210)に割り当てられた計算機リソースの一部を用いて一つ以上の仮想計算機(LPAR220)を生成し、生成されたLPAR220上でゲストOS230を起動させる。ゲストOS230では、さらに、所定の処理を行うアプリケーション232が稼働する。

【0029】

ホストOS2(210)は、ホストOS2(210)に割り当てられた分割RAMのホスト物理アドレス空間と、LPAR220に割り当てられた論理RAMのゲスト物理アドレス空間との間のマッピングを管理する。

【0030】

分割RAMのホスト物理アドレス空間は、分割RAMの物理的な位置を示すアドレス空間である。論理RAMのゲスト物理アドレス空間は、論理RAMの物理的な位置を示すアドレス空間である。なお、ホストOS1(200)も、自身に割り当てられた分割RAMのホスト物理アドレスを管理する。

【0031】

ゲストOS230は、論理RAMのゲスト物理アドレス空間とゲスト仮想アドレス空間との間のマッピングを管理する。ここで、ゲスト仮想アドレス空間は、ゲストOS230がアプリケーション232に割り当てられる仮想的なRAMの位置を示すアドレス空間である。

【0032】

LPAR220は、論理プロセッサ、論理RAM(論理メモリ)、及び論理I/Oデバイスを有する。本実施例のホストOS2(210)は、LPAR220に対して共有領域221を割り当てる。共有領域221は、アプリケーション232とホストOS1(20

10

20

30

40

50

0)との間の通信等に用いられるメモリ領域である。

【0033】

本実施例のホストOS2(210)は、共有領域管理部211及び処理部212を有し、また、管理情報213を保持する。

【0034】

共有領域管理部211は、L P A R 2 2 0 に対して共有領域221を割り当て、当該共有領域221を管理する。共有領域管理部211は、共有領域221を複数のブロックに分割し、ブロック単位に共有領域221の状態を管理する。以下の説明ではブロック単位の共有領域221を単にブロックとも記載する。処理部212は、共有領域221を用いたホストOS1(200)とゲストOS230との間の通信を制御する。管理情報213は、共有領域221を管理するための情報であり、共有領域管理部211によって管理される。管理情報213の詳細については図3を用いて説明する。

10

【0035】

本実施例のゲストOS230は、エージェント231を含む。エージェント231は、アプリケーション232が発行した共有領域221に関する要求をホストOS2(210)に渡し、また、ホストOS2(210)に対して共有領域221の確保を依頼する。また、エージェント231は、仮想アドレス空間に確保された共有領域221をマッピングし、共有領域221がマッピングされた仮想アドレスをアプリケーション232に返す。これによって、アプリケーション232は、共有領域221を使用することができる。

【0036】

図3は、実施例1の管理情報213の一例を示す図である。

20

【0037】

管理情報213は、ブロックID301、ゲスト物理アドレス302、ホスト物理アドレス303、サイズ304、及び割当状態305を含む。

【0038】

ブロックID301は、ブロックの識別番号である。ゲスト物理アドレス302は、ゲスト物理アドレス空間上におけるブロックの開始アドレスである。ホスト物理アドレス303は、ホスト物理アドレス空間上におけるブロックの開始アドレスである。サイズ304は、ブロックのサイズである。本実施例では、全てのブロックのサイズが同一であるものとする。なお、各ブロックのサイズは異なってもよい。

30

【0039】

割当状態305は、ブロックがアプリケーション232に対して割り当てられているか否かを示す情報である。本実施例の割当状態305には「ON」又は「OFF」のいずれかが格納される。割当状態305が「ON」の場合、ブロックがアプリケーション232に割り当てられていることを示す。割当状態305が「OFF」の場合、ブロックがアプリケーション232に割り当てられていないことを示す。

【0040】

本実施例では、一つのL P A R 2 2 0 に対して一つの共有領域221が割り当てられる。そのため、ホストOS2(210)は、L P A R 2 2 0 の数と同数の管理情報213を保持する。

40

【0041】

図4は、実施例1の共有領域のマッピング関係及び構造を示す図である。

【0042】

物理計算機10のRAM120のメモリ領域は、二つに分割され、ホストOS1及びホストOS2のそれぞれに分割されたメモリ領域が割り当てられる。ホストOS1(200)及びホストOS2(210)のそれぞれは、割り当てられたメモリ領域を独立したホスト物理アドレス空間として管理する。

【0043】

ホストOS2(210)は、割り当てられたメモリ領域の一部をホストOS2(210)の制御用の領域として確保する。また、ホストOS2(210)は、割り当てられたメ

50

メモリ領域の一部をLPAR220の領域として確保し、LPAR220に当該領域を割り当てる。

【0044】

LPAR220は、LPAR220の領域を、ゲストOS230の領域と、LPAR220の制御用の領域とに分割する。なお、ゲストOS230の領域には、MMIO用の領域、F/W用の領域、RAM用の領域、及びゲストOS230の制御用の領域が含まれる。

【0045】

ホストOS2(210)は、LPAR220を起動時に、制御用の領域に、共有領域221を割り当てる。すなわち、制御用の領域の一部が共有領域221として使用される。共有領域221のデータ構造は、データ領域及び通信用領域から構成される。データ領域は、ホストOS1(200)によってバックアップされたデータを用いたリストアを行うために、ホストOS1(200)から送信されたデータを格納するメモリ領域である。通信用領域は、アプリケーション232とホストOS1(200)との間で通信に用いるメモリ領域である。

10

【0046】

なお、図4に示す共有領域221のデータ構造は論理的な構造を示すものである。したがって、データ領域及び通信用領域は連続的なメモリ領域でなくてもよい。なお、通常起動時にも共有領域221を割り当てることによって、例えば、ホストOS2(210)及びアプリケーション232がメモリを共有すること等ができる。

20

【0047】

図4に示すように、共有領域221は、ゲストOS230の領域とは異なるため、ゲストOS230からは論理RAMとして認識されないメモリ領域である。また、共有領域221は、ホストOS2に割り当てられたメモリ領域に含まれるため、アプリケーション232との間の通信以外にはホストOS1(200)からアクセスされることはない。

【0048】

図5A及び図5Bは、実施例1の計算機システムの共有領域221を用いた処理の流れを示すシーケンス図である。本実施例では、共有領域を用いた処理の一例として、共有領域を用いたリストア処理について説明する。

【0049】

ホストOS1(200)は、物理計算機10の電源部160の障害等の電源遮断時にキャッシング用にRAM120に格納されるデータを外部ストレージ装置20にバックアップする(ステップS101)。その後、物理計算機10は再起動を行う。

30

【0050】

ホストOS2(210)の共有領域管理部211は、LPAR220の起動時に、LPAR220に共有領域221を割り当てる(ステップS102)。

【0051】

例えば、共有領域管理部211は、LPAR220の起動時に、LPAR220に割り当てられたメモリ領域のうち、制御用のメモリ領域から共有領域221用の領域を確保する。

40

【0052】

また、共有領域管理部211は、ホスト物理アドレス空間のアドレスが連続するメモリ領域を共有領域221として確保する。このとき、共有領域管理部211は、確保されたメモリ領域をブロック単位に分割し、各ブロックに識別番号を付与する。

【0053】

共有領域管理部211は、管理情報213にブロックの数だけエントリを生成し、ブロックID301に付与された識別番号を設定する。また、共有領域管理部211は、各エントリのホスト物理アドレス303に各ブロックの開始アドレスを設定し、また、サイズ304に各ブロックのサイズを設定する。さらに、共有領域管理部211は、各エントリの割当状態305に「OFF」を設定する。

50

【 0 0 5 4 】

共有領域管理部 2 1 1 は、L P A R 2 2 0 の起動時、確保されたメモリ領域のアドレスに基づいて各ブロックのゲスト物理アドレス及びホスト物理アドレスを取得し、各エントリのゲスト物理アドレス 3 0 2 及びホスト物理アドレス 3 0 3 に取得されたゲスト物理アドレス及びホスト物理アドレスを設定する。以上がステップ S 1 0 2 の処理の説明である。

【 0 0 5 5 】

L P A R 2 2 0 が起動し、さらに、L P A R 2 2 0 上でゲスト O S 2 3 0 が起動した後、アプリケーション 2 3 2 の処理が開始される。このとき、アプリケーション 2 3 2 は、バックアップ処理が実行されたか否かを判定する。例えば、ホスト O S 2 (2 1 0) を介して、ホスト O S 1 (2 0 0) に対して問い合わせることによって、前述した判定をすることができる。ここでは、バックアップ処理が実行されたと判定するものとする。

10

【 0 0 5 6 】

この場合、アプリケーション 2 3 2 は、エージェント 2 3 1 に対して共有領域 2 2 1 の確保要求を発行する (ステップ S 1 0 3)。なお、当該確保要求には、確保するメモリ領域のサイズが含まれる。

【 0 0 5 7 】

エージェント 2 3 1 は、共有領域管理部 2 1 1 にアプリケーション 2 3 2 から受け取った共有領域 2 2 1 の確保要求を出す (ステップ S 1 0 4)。

【 0 0 5 8 】

共有領域管理部 2 1 1 は、エージェント 2 3 1 から共有領域 2 2 1 の確保要求を受信した場合、管理情報 2 1 3 を参照して、アプリケーション 2 3 2 のゲスト仮想アドレス空間に割り当てる共有領域 2 2 1 (ブロック)を確保する (ステップ S 1 0 5)。すなわち、アプリケーション 2 3 2 に提供するブロックが選択される。さらに、共有領域管理部 2 1 1 は、マッピングするゲスト物理アドレスをエージェント 2 3 1 に通知する (ステップ S 1 0 6)。

20

【 0 0 5 9 】

具体的には、共有領域管理部 2 1 1 は、管理情報 2 1 3 の各エントリの割当状態 3 0 5 を参照し、管理情報 2 1 3 が「OFF」であるエントリが要求サイズ分連続している箇所を検索する。共有領域管理部 2 1 1 は、検索されたエントリ群を選択し、選択されたエントリ群の割当状態 3 0 5 に「ON」を設定する。共有領域管理部 2 1 1 は、選択されたエントリ群の先頭のブロック I D 3 0 1 及びゲスト物理アドレス 3 0 2 に格納される値をエージェント 2 3 1 に通知する。このとき、共有領域管理部 2 1 1 がエージェントに渡すアドレスは、マッピングするアドレスであるため、アプリケーション 2 3 2 に使用させる領域までのオフセットも合わせて通知する。

30

【 0 0 6 0 】

エージェント 2 3 1 は、共有領域管理部 2 1 1 からゲスト物理アドレスを受け取った場合、当該ゲスト物理アドレスをゲスト仮想アドレス空間にマッピングする (ステップ S 1 0 7)。本実施例では、O S の機能を用いてゲスト物理アドレスがゲスト仮想アドレスにマッピングされる。具体的な処理については図 6 を用いて説明する。

40

【 0 0 6 1 】

エージェント 2 3 1 は、マッピングしたゲスト仮想アドレスをアプリケーション 2 3 2 に通知する (ステップ S 1 0 8)。

【 0 0 6 2 】

具体的には、エージェント 2 3 1 は、マッピング結果及びオフセットに基づいて、アプリケーション 2 3 2 が使用する先頭ブロックのゲスト仮想アドレス、ゲスト物理アドレス、ファイルディスクリプタ、ブロックの識別番号、サイズ等の確保された共有領域の解放に必要な情報をアプリケーション 2 3 2 に通知する。

【 0 0 6 3 】

アプリケーション 2 3 2 は、エージェント 2 3 1 からゲスト物理アドレスを受け取った

50

場合、ホストOS 1 (2 0 0) との通信を開始するために、ホストOS 2 (2 1 0) に対して共有領域 2 2 1 を用いて処理の開始要求を発行する (ステップ S 1 0 9) 。なお、当該開始要求には、ステップ S 1 0 3 において発行された共有領域 2 2 1 の確保要求に含まれるサイズ、及び、エージェント 2 3 1 から受け取ったブロックの識別番号が含まれる。

【 0 0 6 4 】

エージェント 2 3 1 は、処理部 2 1 2 にアプリケーション 2 3 2 から受け取った処理の開始要求を発行する (ステップ S 1 1 0) 。

【 0 0 6 5 】

処理部 2 1 2 は、エージェント 2 3 1 から処理の開始要求を受け取った場合、共有領域管理部 2 1 1 に確保されたブロックのホスト物理アドレスを要求する (ステップ S 1 1 1) 。当該ホスト物理アドレスの要求には、ステップ S 1 0 8 において発行され、ステップ S 1 0 9 の開始要求で入力されたブロックの識別番号が含まれる。

10

【 0 0 6 6 】

共有領域管理部 2 1 1 は、ホスト物理アドレスの要求を受け取った場合、管理情報 2 1 3 に基づいて、確保されたブロックのホスト物理アドレスを処理部 2 1 2 に通知する (ステップ S 1 1 2) 。

【 0 0 6 7 】

具体的には、共有領域管理部 2 1 1 は、管理情報 2 1 3 を参照し、ブロック ID 3 0 1 がホスト物理アドレスの要求に含まれるブロックの識別番号と一致するエントリを検索する。共有領域管理部 2 1 1 は、検索されたエントリのホスト物理アドレス 3 0 3 に格納されるアドレスを処理部 2 1 2 に通知する。

20

【 0 0 6 8 】

処理部 2 1 2 は、共有領域管理部 2 1 1 からホスト物理アドレスを受け取った場合、ホストOS 1 (2 0 0) にホスト物理アドレスを通知する (ステップ S 1 1 3) 。このとき、処理部 2 1 2 は、ステップ S 1 0 9 の開始要求の入力に含まれるサイズもホスト物理アドレスと合わせて通知する。これによって、ホストOS 1 (2 0 0) は、アプリケーション 2 3 2 に割り当てられた共有領域 2 2 1 にアクセスが可能となる。

【 0 0 6 9 】

ホストOS 1 (2 0 0) は、ホスト物理アドレスの通知を受け取った場合、共有領域 2 2 1 を用いた処理を実行する (ステップ S 1 1 4) 。本実施例では、共有領域 2 2 1 を用いたリストア処理が実行される。リストア処理は公知のものであるため詳細な説明は省略する。

30

【 0 0 7 0 】

アプリケーション 2 3 2 は、リストア処理が完了した後、エージェント 2 3 1 に完了通知を発行する (ステップ S 1 1 5) 。エージェント 2 3 1 は、処理部 2 1 2 にアプリケーション 2 3 2 から受け取った完了通知を発行する (ステップ S 1 1 6) 。

【 0 0 7 1 】

処理部 2 1 2 は、エージェント 2 3 1 から完了通知を受け取った場合、ホストOS 1 (2 0 0) に完了通知を発行する (ステップ S 1 1 7) 。

【 0 0 7 2 】

40

ホストOS 1 (2 0 0) は、処理部 2 1 2 から完了通知を受け取った場合、これ以降、共有領域 2 2 1 へアクセスすることがないように、完了処理を実行する (ステップ S 1 1 8) 。完了処理は公知のものであるため詳細な説明は省略する。

【 0 0 7 3 】

アプリケーション 2 3 2 は、エージェント 2 3 1 に解放要求を発行する (ステップ S 1 1 9) 。当該解放要求には、ステップ S 1 0 8 において、エージェント 2 3 1 から受け取った各種情報が含まれる。エージェント 2 3 1 は、共有領域管理部 2 1 1 にアプリケーション 2 3 2 から受け取った解放要求を発行する (ステップ S 1 2 0) 。

【 0 0 7 4 】

共有領域管理部 2 1 1 は、エージェント 2 3 1 から解放要求を受け取った場合、アプリ

50

ケーション232に対して割り当てられたブロックを全て解放する(ステップS121)。このとき、共有領域管理部211は、管理情報213を参照し、解放されたブロックに対応するエントリの割当状態305を全て「OFF」に更新する。

【0075】

共有領域管理部211は、アンマップ用のアドレスを算出するためのオフセットをエージェント231に通知する(ステップS122)。

【0076】

エージェント231は、共有領域管理部211からオフセットを受け付けた場合、当該オフセットを用いて、ゲスト仮想アドレス空間からブロックをアンマップする(ステップS123)。

10

【0077】

図6は、実施例1のエージェント231が実行するマッピング処理の一例を説明するフローチャートである。

【0078】

エージェント231は、スペシャルファイルをopenする(ステップS201)。エージェント231は、スペシャルファイルのopenに成功したか否かを判定する(ステップS202)。

【0079】

スペシャルファイルのopenに失敗したと判定された場合、エージェント231は、エラーと判定して処理を終了する。スペシャルファイルのopenに成功したと判定された場合、エージェント231は、ブロックのゲスト物理アドレスをゲスト仮想アドレスにマッピングするためにmmapを実行する(ステップS203)。エージェント231は、mmapが成功したか否かを判定する(ステップS204)。

20

【0080】

mmapが失敗したと判定された場合、エージェント231は、エラーと判定して処理を終了する。mmapが成功したと判定された場合、エージェント231は、mmapの実行結果として、アプリケーション232に対してゲスト仮想アドレスを返す(ステップS205)。

【0081】

なお、共有領域221をアンマップする場合、エージェント231は、munmapを実行する。

30

【0082】

以上で説明したように、本実施例によれば、それぞれ独立した計算機リソースを有するホストOS1(200)及びホストOS2(210)が稼働する計算機において、ホストOS2が管理するLPAR220上のアプリケーションと、ホストOS1(200)とが通信するための共有領域を確保することができる。

【0083】

したがって、独立した計算機リソースを使用するホストOS1(200)とホストOS2上のアプリケーション232との間で通信等の処理が可能となる。

【0084】

40

なお、共有領域の使用方法は、ホストOS1(200)とホストOS2上のアプリケーション232との間の通信に限定されるものではない。本発明を応用して、例えば、ホストOS2とアプリケーション232との間の通信、又はホストOS2とエージェント231との間の通信にも共有領域を使用することが可能である。

【0085】

なお、本発明は上記した実施例に限定されるものではなく、様々な変形例が含まれる。また、例えば、上記した実施例は本発明を分かりやすく説明するために構成を詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。また、各実施例の構成の一部について、他の構成に追加、削除、置換することが可能である。

50

【0086】

また、上記の各構成、機能、処理部、処理手段等は、それらの一部又は全部を、例えば集積回路で設計する等によりハードウェアで実現してもよい。また、本発明は、実施例の機能を実現するソフトウェアのプログラムコードによっても実現できる。この場合、プログラムコードを記録した記憶媒体をコンピュータに提供し、そのコンピュータが備えるプロセッサが記憶媒体に格納されたプログラムコードを読み出す。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施例の機能を実現することになり、そのプログラムコード自体、及びそれを記憶した記憶媒体は本発明を構成することになる。このようなプログラムコードを供給するための記憶媒体としては、例えば、フレキシブルディスク、CD-ROM、DVD-ROM、ハードディスク、SSD (Solid State Drive)、光ディスク、光磁気ディスク、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどが用いられる。

10

【0087】

また、本実施例に記載の機能を実現するプログラムコードは、例えば、アセンブラ、C/C++、perl、Shell、PHP、Java等の広範囲のプログラム又はスクリプト言語で実装できる。

【0088】

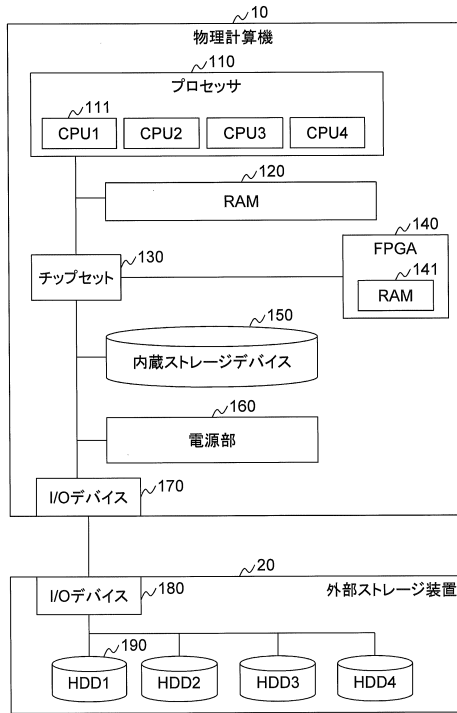
さらに、実施例の機能を実現するソフトウェアのプログラムコードを、ネットワークを介して配信することによって、それをコンピュータのハードディスクやメモリ等の記憶手段又はCD-RW、CD-R等の記憶媒体に格納し、コンピュータが備えるプロセッサが当該記憶手段や当該記憶媒体に格納されたプログラムコードを読み出して実行するようにしてもよい。

20

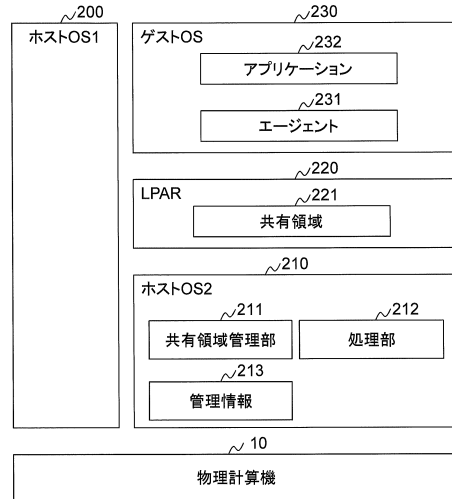
【0089】

上述の実施例において、制御線や情報線は、説明上必要と考えられるものを示しており、製品上必ずしも全ての制御線や情報線を示しているとは限らない。全ての構成が相互に接続されていてもよい。

【図1】



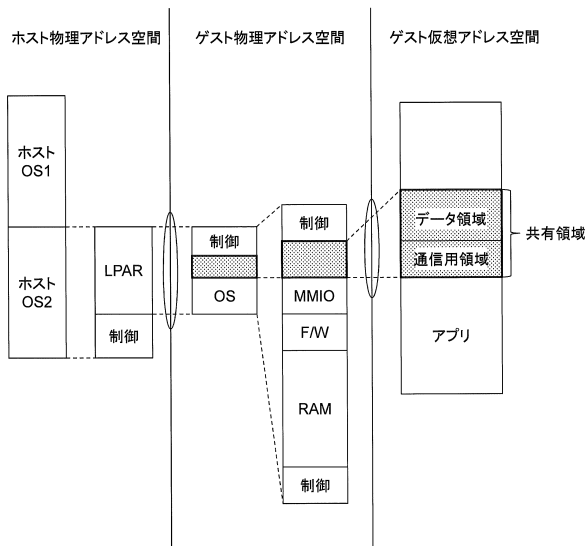
【図2】



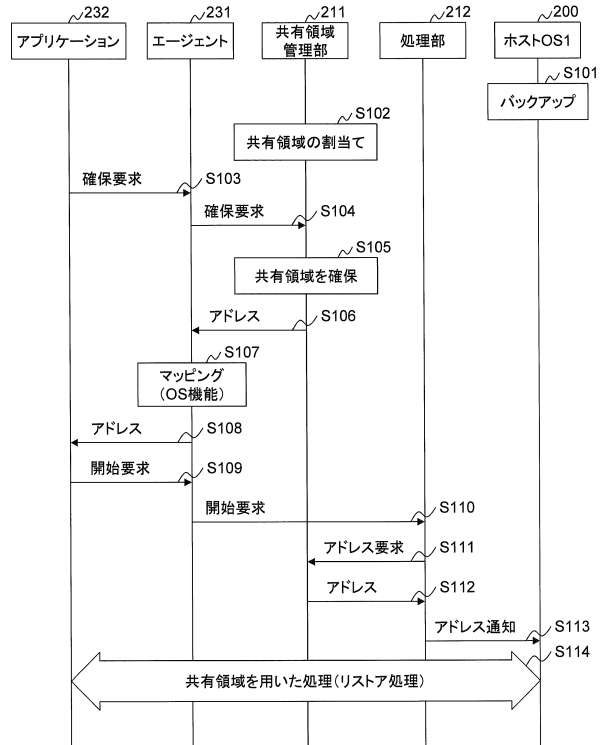
【図3】

301 ブロックID	302 ゲスト物理アドレス	303 ホスト物理アドレス	304 サイズ	305 割当状態
0	aaaaaaaa	xxxxxxxx	1kB	ON
1	aaaaaaab	xxxxxxy	1kB	OFF
⋮	⋮	⋮	⋮	⋮

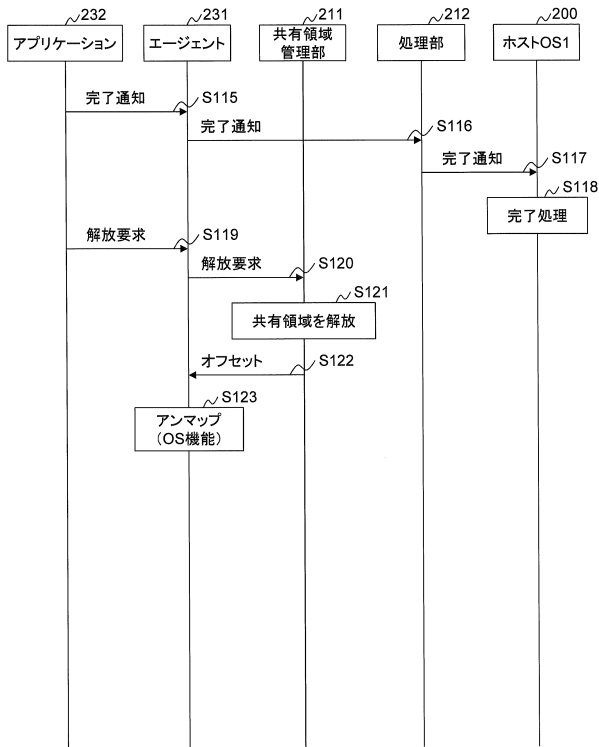
【図4】



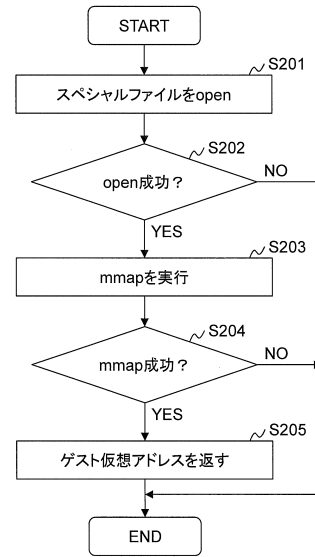
【図5A】



【図5B】



【図6】



フロントページの続き

(72)発明者 森木 俊臣

東京都千代田区丸の内一丁目6番6号 株式会社日立製作所内

(72)発明者 衣川 聡

神奈川県横浜市西区みなとみらい二丁目3番3号 株式会社日立情報通信エンジニアリング内

審査官 井上 宏一

(56)参考文献 米国特許出願公開第2015/0046924 (US, A1)

特開2001-243080 (JP, A)

特開2004-258698 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/455 - 9/54