

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6244786号
(P6244786)

(45) 発行日 平成29年12月13日(2017.12.13)

(24) 登録日 平成29年11月24日(2017.11.24)

(51) Int.Cl.		F I			
G06F 13/10	(2006.01)	G06F 13/10	340A		
G06F 9/46	(2006.01)	G06F 9/46	350		
		G06F 13/10	330C		
		G06F 13/10	320Z		

請求項の数 6 (全 18 頁)

(21) 出願番号	特願2013-204657 (P2013-204657)	(73) 特許権者	000005223
(22) 出願日	平成25年9月30日(2013.9.30)		富士通株式会社
(65) 公開番号	特開2015-69524 (P2015-69524A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成27年4月13日(2015.4.13)	(74) 代理人	100092978
審査請求日	平成28年6月6日(2016.6.6)		弁理士 真田 有
		(74) 代理人	100112678
			弁理士 山本 雅久
		(72) 発明者	七野 伸幸
			石川県かほく市宇野気ヌ98番地の2 株式会社PFU内
		審査官	田中 啓介

最終頁に続く

(54) 【発明の名称】 情報処理装置、ストレージ制御装置、及びプログラム

(57) 【特許請求の範囲】

【請求項1】

第1オペレーティングシステムによって制御される第1デバイスと、
 第2オペレーティングシステムによって制御される第2デバイスと、
 前記第1オペレーティングシステムと前記第2オペレーティングシステムとを動作させ、
前記第1オペレーティングシステムに対して、前記第1オペレーティングシステムが行なう前記第1デバイスの運用時の制御に用いられる制御機能と、起動時および保守時の前記第1デバイスの制御に用いられ前記第1デバイスの状態を通知する機能を含む保守機能と、を有する第1ドライバをロードさせ、前記第1ドライバに前記第1デバイスの保守制御を実行させ、前記第1オペレーティングシステムに対して、起動時および保守時の前記第2デバイスの制御に用いられ前記第2デバイスの状態を通知する機能を含む保守機能のみを有する第2ドライバをロードさせ、前記第2ドライバに前記第2デバイスの保守制御を実行させる処理部と、
 を備えることを特徴とする情報処理装置。

【請求項2】

前記処理部は、前記第2オペレーティングシステムの動作の停止中に、前記第1オペレーティングシステムに対して、前記第2ドライバを用いて、前記第2デバイスに係る情報を取得し、前記第2デバイスの保守制御を実行させる、請求項1記載の情報処理装置。

【請求項3】

前記処理部は、前記第2オペレーティングシステムの動作の停止中に、前記第1オペレ

ーティングシステムに対して、前記第2ドライバを用いて、前記第2デバイスの設定制御を実行させる、請求項1または請求項2に記載の情報処理装置。

【請求項4】

前記第1オペレーティングシステムは、S A N (Storage Area Network) 制御を行なうS A N用オペレーティングシステムであり、

前記第2オペレーティングシステムは、N A S (Network Attached Storage) 制御を行なうN A S用オペレーティングシステムである、請求項1～請求項3のいずれか一項に記載の情報処理装置。

【請求項5】

ストレージユニットを制御するストレージ制御装置であって、

S A N (Storage Area Network) 制御を行なう第1オペレーティングシステムによって制御される第1デバイスと、

N A S (Network Attached Storage) 制御を行なう第2オペレーティングシステムによって制御される第2デバイスと、

前記第1オペレーティングシステムと前記第2オペレーティングシステムとを動作させ、前記第1オペレーティングシステムに対して、前記第1オペレーティングシステムが行なう前記第1デバイスの運用時の制御に用いられる制御機能と、起動時および保守時の前記第1デバイスの制御に用いられ前記第1デバイスの状態を通知する機能を含む保守機能と、を有する第1ドライバをロードさせ、前記第1ドライバに前記第1デバイスの保守制御を実行させ、前記第1オペレーティングシステムに対して、起動時および保守時の前記第2デバイスの制御に用いられ前記第2デバイスの状態を通知する機能を含む保守機能のみを有する第2ドライバをロードさせ、前記第2ドライバに前記第2デバイスの保守制御を実行させる処理部と、を備えることを特徴とするストレージ制御装置。

【請求項6】

コンピュータに、

第1デバイスを制御する第1オペレーティングシステムと第2デバイスを制御する第2オペレーティングシステムとを動作させ、

前記第1オペレーティングシステムに対して、前記第1オペレーティングシステムが行なう前記第1デバイスの運用時の制御に用いられる制御機能と、起動時および保守時の前記第1デバイスの制御に用いられ前記第1デバイスの状態を通知する機能と、を含む保守機能を有する第1ドライバをロードさせ、前記第1ドライバに前記第1デバイスの保守制御を実行させ、

前記第1オペレーティングシステムに対して、起動時および保守時の前記第2デバイスの制御に用いられ前記第2デバイスの状態を通知する機能を含む保守機能のみを有する第2ドライバをロードさせ、前記第2ドライバに前記第2デバイスの保守制御を実行させる、プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理装置、ストレージ制御装置、及びプログラムに関する。

【背景技術】

【0002】

ネットワークストレージシステムとして、N A S (Network Attached Storage) や S A N (Storage Area Network) が知られている。

既存のイーサネット(登録商標)ネットワークに接続するN A Sは、S A Nに比べて管理は簡単だが、性能面で劣り、又、容量不足に陥ることが多い。一方、S A Nは、一般にN A Sに比べて高速で、より多くのデータ保存が可能だが高価である。

これらのN A SとS A Nとのそれぞれの長所をうまく組み合わせ、データ保存の効率向上およびコスト削減を図るために、S A NとN A Sとの統合が実現されつつある。

10

20

30

40

50

【 0 0 0 3 】

このとき、S A N用デバイスは、S A N用オペレーティングシステム（Operating System；OS）の制御下に置かれ、N A S用デバイスは、N A S用OSの制御下に置かれる。しかし、両OS制御下のデバイスは、統合ストレージ装置のハードウェアとして実装されているので、各デバイスに対する保守は、共通の保守手順で行ないたい。なぜならば、OS毎に保守ガイド画面の構成や保守手順が異なると、ユーザにとって使用し難いためである。

【先行技術文献】

【特許文献】

【 0 0 0 4 】

【特許文献1】特開2011-65551号公報

【特許文献2】特開2012-9029号公報

【特許文献3】特開2000-207232号公報

【発明の概要】

【発明が解決しようとする課題】

【 0 0 0 5 】

したがって、複数のOSに対して保守制御機能を二重に実装しなければならず非効率的であるという課題がある。

【 0 0 0 6 】

一つの側面で、本発明は、複数のオペレーティングシステムによって管理されるそれぞれのデバイスに対応するデバイスの保守制御の効率化を図ることを目的とする。

なお、前記目的に限らず、後述する発明を実施するための最良の形態に示す各構成により導かれる作用効果であって、従来の技術によっては得られない作用効果を奏することも本件の他の目的の一つとして位置付けることができる。

【課題を解決するための手段】

【 0 0 0 7 】

本件の情報処理装置は、第1オペレーティングシステムによって制御される第1デバイスと、第2オペレーティングシステムによって制御される第2デバイスと、処理部とを備える。前記処理部は、前記第1オペレーティングシステムと前記第2オペレーティングシステムとを動作させ、前記第1オペレーティングシステムに対して、前記第1オペレーティングシステムが行なう前記第1デバイスの運用時の制御に用いられる制御機能と、起動時および保守時の前記第1デバイスの制御に用いられ前記第1デバイスの状態を通知する機能を含む保守機能と、を有する第1ドライバをロードさせ、前記第1ドライバに前記第1デバイスの保守制御を実行させ、前記第1オペレーティングシステムに対して、起動時および保守時の前記第2デバイスの制御に用いられ前記第2デバイスの状態を通知する機能を含む保守機能のみを有する第2ドライバをロードさせ、前記第2ドライバに前記第2デバイスの保守制御を実行させる。

【発明の効果】

【 0 0 0 8 】

一実施形態によれば、複数のオペレーティングシステムによって管理されるそれぞれのデバイスに対応するデバイスの保守制御の効率化を図ることができる。

【図面の簡単な説明】

【 0 0 0 9 】

【図1】本実施形態のストレージ制御装置（情報処理装置）を含むストレージ装置のハードウェア構成を模式的に示す図である。

【図2】シングルOSのストレージ制御装置（情報処理装置）におけるOSを模式的に示す図である。

【図3】本実施形態のストレージ制御装置（情報処理装置）におけるハイパバイザ上の複数のOSを模式的に示す図である。

【図4】図1に示すストレージ制御装置（情報処理装置）における保守ドライバ（第2保

10

20

30

40

50

守ドライバ)の動作を説明するフローチャートである。

【図5】図1に示すストレージ制御装置(情報処理装置)による起動シーケンスを説明する図である。

【図6】図1に示すストレージ制御装置(情報処理装置)による活性増設シーケンスを説明する図である。

【図7】図1に示すストレージ制御装置(情報処理装置)によるLinux稼働中の活性交換シーケンスを説明する図である。

【図8】図1に示すストレージ制御装置(情報処理装置)によるLinux停止中の活性交換シーケンスを説明する図である。

【発明を実施するための形態】

10

【0010】

以下に、図面を参照し、本願の開示する情報処理装置、ストレージ制御装置、及びプログラムの実施形態について、詳細に説明する。ただし、以下に示す実施形態は、あくまでも例示に過ぎず、実施形態で明示しない種々の変形例や技術の適用を排除する意図はない。すなわち、本実施形態を、その趣旨を逸脱しない範囲で種々変形して実施することができる。また、各図は、図中に示す構成要素のみを備えるという趣旨ではなく、他の機能を含むことができる。そして、各実施形態は、処理内容を矛盾させない範囲で適宜組み合わせることが可能である。

【0011】

(1)本実施形態のストレージ制御装置(情報処理装置)の構成

20

まず、図2を参照しながら、シングルOSのストレージ制御装置(情報処理装置)におけるOSについて説明する。なお、図2は、当該ストレージ制御装置におけるOSを模式的に示す図である。

【0012】

図2では、例えば、ストレージ制御装置のCPU(Central Processing Unit)においてVxWorks(登録商標)のシングルOS112が動作する場合が示されている。VxWorksは、SAN制御を行なうSAN用OSの一例である。また、図2では、OS112には、VxWorks用のデバイス120a~120cが接続されている。VxWorks用のデバイス120a~120cは、例えばNIC(Network Interface Card)、FC(Fibre Channel)カードであり、OS112によって管理される。

30

【0013】

このとき、OS112は、デバイス120a~120c用のデバイスドライバ121a~121cを有している。ここで用いられるデバイスドライバ121a~121cは、フル機能ドライバである。フル機能ドライバは、起動時や保守時のデバイスの制御に使用される保守機能と、運用時のデバイス120a~120cの制御に使用されるデバイスドライバ機能とを有する。ここで、保守機能は、各デバイス120a~120cの状態を通知するなどの保守制御に有用な機能を実現する。また、デバイスドライバ機能とは、デバイス120a~120cが例えばNICであれば、通信制御がデバイス120a~120cをデバイスとして使用する機能にあたる。

【0014】

40

このようなSAN用OS112を備えたSAN装置にNAS(Network Attached Storage)機能を追加するには、以下の手法が用いられる。その手法では、仮想化OS(ハイパーバイザシステム)を導入し、一つのシステムハードウェア上にVxWorksとLinux(登録商標)との二つのOSを搭載し、一方のOS(Linux)にNAS機能(例えばNFS(Network File System)、Samba)を実装することである。これにより、SAN機能とNAS機能との両方をもつ統合ストレージ装置(Unified Storage)が実現される。

【0015】

NAS機能のソフトウェアは、Linux上に安定したものがあるので、上記手法は合理的な手法である。ただし、NAS機能によって使用されるディスクは、VxWorks側にあるので、Linuxで受け取ったデータは、SAN機能によってVxWorks側のディスクに書き込まれ

50

る。

【 0 0 1 6 】

このとき、N A S用に新たにデバイス（例えばN I C）を追加すると、当該デバイスは、Linuxの制御下に置かれる。しかし、当該デバイスは統合ストレージ装置のハードウェアとして実装されているので、当該デバイスに対する保守は、当該デバイスを追加する前のVxWorksによる保守手順で行ないたい。なぜならば、O S毎に保守ガイド画面の構成や保守手順が異なると、ユーザにとって使用し難いためである。

【 0 0 1 7 】

つまり、N A S用に追加したデバイスについても、VxWorks上に実装されている保守制御機能と同じものが必要になる。そのため、

- (1) 保守に関する制御は、全てVxWorksで行なうか、もしくは、
- (2) Linux側に、VxWorksに実装された保守制御機能と同じ保守制御機能を実装する、必要がある。

【 0 0 1 8 】

しかしながら、上記(1)のように保守に関する制御を全てVxWorksで行なう場合、VxWorks側で必要とする情報の全てをLinuxが採取してVxWorksに通知しなければならない。また、例えばLinuxが停止している場合、VxWorksが望む情報を、VxWorksが望むタイミングで採取することができない。

【 0 0 1 9 】

また、上記(2)のように、Linux側に、VxWorksに実装された保守制御機能と同じ保守制御機能を実装する場合、VxWorksとLinuxとの両方で、複数のO Sに対して共通の保守制御機能を二重に開発実装しなければならず非効率的である。

【 0 0 2 0 】

さらに、上記(1)および(2)に共通する課題として、以下のような課題がある。上述した統合ストレージ装置のようなシステム構成では、ディスクはVxWorks側で管理されており、二つのO SのうちLinuxだけが生き残ってもディスクがなくLinuxは機能できないため、Linuxだけが生き残ることはない。しかし、ディスクはVxWorks側で管理されているので、VxWorksだけが生き残る場合はある。その場合、Linuxが停止しているときでも、Linux側のデバイスの保守を可能にすることが望ましい。

【 0 0 2 1 】

以上の点から、上述した統合ストレージ装置のようなシステム構成において、Linux側に追加されるデバイスについての保守機能を、Linux側にもたせることは非合理であるといえる。

【 0 0 2 2 】

以下、図 1 を参照しながら、本実施形態のストレージ制御装置（情報処理装置）1 0 0 を含むストレージ装置 1 について説明する。なお、図 1 は、本実施形態のストレージ制御装置 1 0 0 を含むストレージ装置 1 のハードウェア構成を模式的に示す図である。

【 0 0 2 3 】

本ストレージ装置 1 は、ドライブエンクロージャ（D E : Drive Enclosure ; ストレージユニット）3 0 0 に格納された記憶装置 3 1 0 を仮想化して、仮想ストレージ環境を形成する。そしてストレージ装置 1 は、仮想ボリュームを上位装置である図示しないホスト装置に提供する。

【 0 0 2 4 】

本ストレージ装置 1 は、1 以上のホスト装置（図示略）に対して通信可能に接続されている。ホスト装置とストレージ装置 1 とは、後述するC A（Communication Adapter）1 0 1 , 1 0 2 により接続されている。

【 0 0 2 5 】

ホスト装置は、例えば、サーバ機能をそなえた情報処理装置であり、本ストレージ装置 1 との間において、N A S や S A N のコマンドを送受信する。このホスト装置は、例えば、ストレージ装置 1 に対してN A S におけるリード/ライト等のディスクアクセスコマン

10

20

30

40

50

ドを送信することにより、ストレージ装置 1 が提供するボリュームにデータの書き込みや読み出しを行なう。

【 0 0 2 6 】

そして、本ストレージ装置 1 は、ホスト装置からボリュームに対して行なわれる入出力要求（例えば、リードコマンドやライトコマンド）に応じて、このボリュームに対応する実ストレージに対して、データの読み出しや書き込み等の処理を行なう。なお、ホスト装置からの入出力要求のことを I / O コマンドもしくは I / O 要求という場合がある。

【 0 0 2 7 】

本ストレージ装置 1 は、図 1 に示すように、複数（本実施形態では 2 つ）の C M (Cont
roller Module) 1 0 0 を含むコントローラエンクロージャ 4 0 0 と、1 つ以上（図 1 に
示す例では 1 つ）のドライブエンクロージャ 3 0 0 とをそなえる。 10

ドライブエンクロージャ（ストレージユニット）3 0 0 は、1 以上（図 1 に示す例では
4 つ）の記憶装置（物理ディスク；ストレージユニット）3 1 0 を搭載可能であり、これ
らの記憶装置 3 1 0 の記憶領域（実ボリューム，実ストレージ）を、本ストレージ装置 1
に対して提供する。

【 0 0 2 8 】

例えば、ドライブエンクロージャ 3 0 0 は、複数段のロット（図示略）をそなえ、こ
れらのロットに、H D D (Hard Disk Drive) や S S D (Solid State Drive) 等の記憶
装置 3 1 0 を挿入することにより、実ボリューム容量を随時変更することができる。又、
複数の記憶装置 3 1 0 を用いて R A I D (Redundant Arrays of Inexpensive Disks) を 20

【 0 0 2 9 】

ドライブエンクロージャ 3 0 0 は、各 C M 1 0 0 における複数（本実施形態では 2 つ）
のデバイスアダプタ（Device Adapter：D A）1 0 3 のそれぞれに接続されている。そし
て、ドライブエンクロージャ 3 0 0 には、各 C M 1 0 0 からアクセスして、データの書き
込みや読み出しを行なうことができる。すなわち、ドライブエンクロージャ 3 0 0 の記憶
装置 3 1 0 に対して、各 C M 1 0 0 を接続することにより、記憶装置 3 1 0 へのアクセス
経路が冗長化される。

【 0 0 3 0 】

コントローラエンクロージャ 4 0 0 は、1 以上（図 1 に示す例では 2 つ）の C M 1 0 0 30
を備える。

各 C M 1 0 0 は、ストレージ装置 1 内の動作を制御するコントローラ（ストレージ制御
装置）であり、ホスト装置から送信される I / O コマンドに従って、ドライブエンクロ
ージャ 3 0 0 の記憶装置 3 1 0 へのアクセス制御等、各種制御を行なう。また、各 C M 1 0
0 は互いにほぼ同様の構成を有している。

なお、図中、同一の符号は同一もしくは略同一の部分を示しているもので、その詳細な説
明は省略する。

【 0 0 3 1 】

各 C M 1 0 0 は二重化されており、通常は、一方の C M 1 0 0 がプライマリとして各種
制御を行なう。しかし、このプライマリ C M 1 0 0 の故障時には、セカンダリの C M 1 0 40
0 がプライマリとして動作を引き継ぐ。

各 C M 1 0 0 は C A 1 0 1 , 1 0 2 を介して、それぞれホスト装置に接続される。そし
て、各 C M 1 0 0 は、ホスト装置から送信されるリード/ライト等のコマンドを受信し、
D A 1 0 3 等を介して記憶装置 3 1 0 の制御を行なう。また、各 C M 1 0 0 は、P C I (Peripheral Component Interconnect) Express 等の図示しないインタフェースを介して
相互に接続されている。

【 0 0 3 2 】

C M 1 0 0 は、図 1 に示すように、C A 1 0 1 , 1 0 2 と複数（図 1 に示す例では 2 つ）
の D A 1 0 3 , 1 0 3 とをそなえるとともに、C P U (Central Processing Unit) 1
1 0 , メモリ 1 0 6 , S S D 1 0 7 , I O C 1 0 8 およびロット 1 0 9 をそなえる。 50

CA101, 102は、ホスト装置等から送信されたデータを受信したり、CM100から出力するデータをホスト装置等に送信する。すなわち、CA101, 102は、ホスト装置等の外部装置との間でのデータの入出力(I/O)を制御する。

【0033】

CA101は、NASを介してホスト装置と通信可能に接続するネットワークアダプタであり、例えば、LAN(Local Area Network)インタフェース等である。

各CM100は、このCA101により図示しない通信回線を介してホスト装置等とNASにより接続され、I/Oコマンドの受信やデータの送受信等を行なう。図1に示す例においては、各CM100に2つのCA101, 101がそなえられている。

【0034】

CA102は、SANを介してホスト装置と通信可能に接続するネットワークアダプタであり、例えば、iSCSIインタフェースやFCインタフェースである。

CM100は、CA102により図示しない通信回線を介してホスト装置等とSANにより接続され、I/Oコマンドの受信やデータの送受信等を行なう。図1に示す例においては、各CM100に1つのCA102がそなえられている。

【0035】

DA103は、ドライブエンクロージャ300や記憶装置310等と通信可能に接続するためのインタフェースである。DA103は、ドライブエンクロージャ300の記憶装置310が接続され、各CM100は、ホスト装置から受信したI/Oコマンドに基づき、記憶装置310に対するアクセス制御を行なう。

【0036】

各CM100は、DA103を介して、記憶装置310に対するデータの書き込みや読み出しを行なう。また、図1に示す例においては、各CM100に2つのDA103, 103がそなえられている。そして、各CM100において、各DA103にドライブエンクロージャ300が接続されている。

これにより、ドライブエンクロージャ300の記憶装置310には、各CM100からデータの書き込みや読み出しを行なうことができる。

【0037】

SSD107は、CPU110が実行するプログラムや種々のデータ等を格納する記憶装置である。

メモリ106は、種々のデータやプログラムを一時的に格納する記憶装置であり、図示しないメモリ領域とキャッシュ領域とをそなえる。キャッシュ領域は、ホスト装置から受信したデータや、ホスト装置に対して送信するデータを一時的に格納する。メモリ領域には、CPU110がプログラムを実行する際に、データやプログラムを一時的に格納・展開して用いる。なお、プログラムは、CPU110を、図3を参照しながら後述する処理部として機能させるアプリケーションプログラムを含む。

【0038】

I/O C(Input Output Controller)108は、CM100内におけるデータ転送を制御する制御装置であり、例えば、メモリ106に格納されたデータをCPU110を介することなく転送させるDMA転送(Direct Memory Access)を実現する。

スロット109は、NIC等のデバイス20a, 20b, 30a, 30bを挿入接続されるもので、必要に応じて、デバイス20a, 20b, 30a, 30bを挿抜可能に構成されている。図1では、スロット109は一つだけ図示されているが、実際には、2以上のスロット109がそなえられ、各スロット109に対し、一つのカード状デバイスが挿抜される。

【0039】

CPU110は、種々の制御や演算を行なう処理装置であり、マルチコアプロセッサ(マルチCPU)である。CPU110は、SSD107等に格納されたOSやプログラムを実行することにより、図3を参照しながら後述する処理部としての機能を含む種々の機能を実現する。

10

20

30

40

50

【0040】

CPU110は、仮想化OSであるハイパバイザ(Hypervisor)11(図3参照)を実行し、このハイパバイザ11上で複数の仮想マシン(Virtual Machine; VM)を実行することで複数のOS12, 13(図3参照)を実行させることができる。複数のゲストOS12, 13はCPU110やメモリ106, SSD107, スロット109等のハードウェア資源を共有することができる。

なお、各CM100において複数のCPU110を備え、これらの複数のCPU110上でハイパバイザ11を実行してもよい。

【0041】

次に、図3を参照しながら、本実施形態の各CM100におけるハイパバイザ11上の複数(本実施形態では2つ)のOS12, 13について説明する。図3は、本実施形態の各CM100におけるハイパバイザ11上の複数のOS12, 13を模式的に示す図である。本実施形態においては、図3に示すように、ハイパバイザ11上で、SAN用OS(第1OS)12とNAS用OS(第2OS)13とを実行する。

【0042】

SAN用OS12は、SAN制御を行なうもので、NIC, FCカード等の第1デバイス20a, 20bを管理する。なお、図3では、2つの第1デバイス20a, 20bが図示されているが、本発明はこれに限定されるものではなく、第1デバイスは、1つだけそなえられてもよいし、3以上そなえられてもよい。

【0043】

NAS用OS13は、NAS制御を行なうもので、NIC等の第2デバイス30a, 30bを管理する。なお、図3では、2つの第2デバイス30a, 30bが図示されているが、本発明はこれに限定されるものではなく、第2デバイスは、1つだけそなえられてもよいし、3以上そなえられてもよい。

【0044】

そして、本実施形態の各CM100におけるCPU110は、以下のような処理部として機能する。

つまり、当該処理部は、SAN用OS12とNAS用OS13とを動作させる。また、当該処理部は、SAN用OS12に対して、第1デバイス20a, 20bにそれぞれ対応する第1保守ドライバ21a, 21bをロードさせ、当該第1保守ドライバ21a, 21bにそれぞれ第1デバイス20a, 20bの保守制御を実行させる。さらに、当該処理部は、SAN用OS12に対して、第2デバイス30a, 30bにそれぞれ対応する第2保守ドライバ32a, 32bをロードさせ、当該第2保守ドライバ32a, 32bにそれぞれ第2デバイス30a, 30bの保守制御を実行させる。

【0045】

また、当該処理部は、NAS用OS13の動作の停止中に、SAN用OS12に対して、第2保守ドライバ32a, 32bを用いて、それぞれ、第2デバイス30a, 30bに係る情報を取得し、第2デバイス30a, 30bの保守制御を実行させる。さらに、当該処理部は、NAS用OS13の動作の停止中に、SAN用OS12に対して、第2保守ドライバ32a, 32bを用いて、それぞれ、第2デバイス30a, 30bの設定制御を実行させる。

【0046】

また、第1保守ドライバ21a, 21bは、SAN用OS12が行なう第1デバイス20a, 20bの運用時の制御に使用される制御機能(デバイスドライバ機能)を含んでいる。つまり、第1保守ドライバ21a, 21bは、フル機能ドライバであり、起動時や保守時のデバイスの制御に使用される保守機能と、運用時の第1デバイス20a, 20bの制御に使用されるデバイスドライバ機能(制御ドライバ)とを有する。ここで、保守機能は、第1デバイス20a, 20bの状態を通知するなどの保守制御に有用な機能を実現する。また、デバイスドライバ機能とは、第1デバイス20a, 20bが例えばNICであれば、通信制御が第1デバイス20a, 20bをデバイスとして使用する機能にあたる。

10

20

30

40

50

【 0 0 4 7 】

さらに、上記処理部は、N A S用OS 1 3に対して、第2デバイス3 0 a , 3 0 bにそれぞれ対応する制御ドライバ3 1 a , 3 1 bをロードさせる。ここで、制御ドライバ3 1 a , 3 1 bは、N A S用OS 1 3が行なう第2デバイス3 0 a , 3 0 bの運用時の制御に使用されるものである。したがって、S A N用OS 1 2に対してロードされた第2保守ドライバ3 2 a , 3 2 bは、それぞれ、第2デバイス3 0 a , 3 0 bの状態を通知するなどの保守制御に有用な機能を実現する保守機能のみを有する。また、N A S用OS 1 3に対してロードされた制御ドライバ3 1 a , 3 1 bは、それぞれ、運用時の第2デバイス3 0 a , 3 0 bの制御に使用されるデバイスドライバ機能を有する。

【 0 0 4 8 】

なお、図3に示すように、本実施形態の各CM（ストレージ制御装置）1 0 0のCPU 1 1 0においては、ハイパバイザ1 1 上に第1仮想マシンおよび第2仮想マシンが構築される。そして、第1仮想マシンには、S A N制御を行なうS A N用OS（例えばVxWorks；第1OS）1 2が搭載され、第2仮想マシンには、N A S制御を行なうN A S用OS（例えばLinux；第2OS）1 3が搭載される。

【 0 0 4 9 】

また、図3では、S A N用OS 1 2には、VxWorks用のデバイス（第1デバイス）2 0 a , 2 0 bが接続されている。VxWorks用のデバイス2 0 a , 2 0 bは、例えばNIC , F Cカードであり、S A N用OS 1 2によって管理される。同様に、図3では、N A S用OS 1 3には、Linux用のデバイス（第2デバイス）3 0 a , 3 0 bが接続されている。Linux用のデバイス3 0 a , 3 0 bは、例えばNICであり、N A S用OS 1 3によって管理される。

【 0 0 5 0 】

上述の構成により、S A N用OS 1 2は、第2保守ドライバ3 2 a , 3 2 bの保守機能を用いて、それぞれ、第2デバイス3 0 a , 3 0 bに係る情報を取得し、第2デバイス3 0 a , 3 0 bの保守制御を行なう。

【 0 0 5 1 】

また、S A N用OS 1 2は、第2保守ドライバ3 2 a , 3 2 bを用いて、第2デバイス3 0 a , 3 0 bの設定制御、例えばP C I Configuration（P C I構成）設定/リソース配置設定を行なう。

【 0 0 5 2 】

つまり、S A N用OS（VxWorks）1 2は、S A N用OS 1 2上の第2保守ドライバ3 2 a , 3 2 bを通して、N A S用OS（Linux）1 3の管理下における第2デバイス3 0 a , 3 0 bの情報を取得する。

【 0 0 5 3 】

これにより、S A N用OS（VxWorks）1 2は、取得した情報を用い、S A N用OS（VxWorks）1 2の管理下における第1デバイス2 0 a , 2 0 bに対する保守制御等の手順と同様にして、N A S用OS（Linux）1 3の管理下における第2デバイス3 0 a , 3 0 bに対する保守制御等を行なう。このとき、保守制御等の仲立ちはハイパバイザ1 1によって行なわれる。

【 0 0 5 4 】

また、本実施形態では、N A S用OS 1 3であるLinux側の第2デバイス3 0 a , 3 0 bの設定を、S A N用OS 1 2であるVxWorks側から制御することができる。ここで、第2デバイス3 0 a , 3 0 bとしてはP C I -Expressデバイスを想定している。P C I -ExpressのリソースはLinuxとVxWorksとで共用されており、リソースを自由に設定するとリソースの衝突が生じるため、Linuxは、P C I -Expressのリソースを自由に設定することはできないようになっている。しかし、本実施形態では、Linuxの起動前にP C I構成設定を済ませることで、リソースの衝突を発生させることなく、Linux側のリソース使用を制御することができる。

【 0 0 5 5 】

〔 2 〕本実施形態のストレージ制御装置（情報処理装置）の動作

次に、上述のごとく構成された本実施形態のストレージ制御装置（情報処理装置）である C M 1 0 0 の動作について、図 4 ~ 図 8 を参照しながら説明する。

まず、図 4 に示すフローチャート（ステップ S 1 ~ S 5 ）に従って、本実施形態の C M 1 0 0 における第 2 保守ドライバ 3 2 a , 3 2 b の動作について説明する。

【 0 0 5 6 】

第 2 保守ドライバ 3 2 a , 3 2 b は、S A N 用 O S 1 2 において情報採取の要求の有無を判断する（ステップ S 1 ）。第 2 保守ドライバ 3 2 a , 3 2 b は、S A N 用 O S 1 2 において情報採取の要求を受けると（ステップ S 1 の Y E S ルート）、それぞれ第 2 デバイス 3 0 a , 3 0 b に係る情報を取得する（ステップ S 2 ）。そして、第 2 保守ドライバ 3 2 a , 3 2 b は、採取した情報を要求元に受け渡したり、保守用の情報としてメモリ 1 0 6 等に記憶する（ステップ S 3 ）。この後、第 2 保守ドライバ 3 2 a , 3 2 b は、ステップ S 1 の処理に戻る。

10

【 0 0 5 7 】

また、情報採取の要求が無い場合（ステップ S 1 の N O ルート）、第 2 保守ドライバ 3 2 a , 3 2 b は、S A N 用 O S 1 2 において設定要求の有無を判断する（ステップ S 4 ）。第 2 保守ドライバ 3 2 a , 3 2 b は、S A N 用 O S 1 2 において設定要求を受けると（ステップ S 4 の Y E S ルート）、要求元からの要求内容に従って、それぞれ第 2 デバイス 3 0 a , 3 0 b の設定制御、例えば P C I 構成設定、あるいは、リソース配置設定を行なう（ステップ S 5 ）。この後、第 2 保守ドライバ 3 2 a , 3 2 b は、ステップ S 1 の処理に戻る。なお、設定要求が無い場合（ステップ S 4 の N O ルート）、第 2 保守ドライバ 3 2 a , 3 2 b は、ステップ S 1 の処理に戻る。

20

【 0 0 5 8 】

次に、図 5 を参照しながら、図 1 に示すストレージ制御装置（情報処理装置）である C M 1 0 0 による起動シーケンス（ステップ A 1 ~ A 1 0 , A 1 , A 2 , A 6 , A 1 0 ）について説明する。

【 0 0 5 9 】

起動時には、第 1 O S（以下 VxWorks という）1 2 および第 2 O S（以下 Linux という）1 3 の電源をオンする（ステップ A 1 , A 1 ）。これにより、VxWorks の起動を開始するとともに（ステップ A 2 ）、Linux も起動するが、Linux については、一旦起動した後、一時的に停止する（ステップ A 2 ）。

30

【 0 0 6 0 】

この後、VxWorks において P C I スキャンが開始される（ステップ A 3 ）。そして、P C I スキャンの結果に従って、第 1 保守ドライバ（フル機能ドライバ）2 1 a , 2 1 b や第 2 保守ドライバ（保守機能）3 2 a , 3 2 b がロードされる。また、Linux 管理下の第 2 デバイス 3 0 a , 3 0 b について、P C I 構成の設定およびデバイス情報の採取が、第 2 保守ドライバ 3 2 a , 3 2 b を用いて行なわれ、採取された情報が、保守用の情報としてメモリ 1 0 6 等に記憶される（ステップ A 4 ）。

【 0 0 6 1 】

以上で、VxWorks 側の P C I デバイスの初期設定を終了し（ステップ A 5 ）、Linux の停止状態（suspend 状態）を解除することで（ステップ A 6 ）、Linux の起動が再開される（ステップ A 6 ）。

40

これに伴い、Linux において P C I スキャンが開始される（ステップ A 7 ）。そして、P C I スキャンの結果に従って、制御ドライバ（デバイスドライバ機能）3 1 a , 3 1 b がロードされる。このとき、P C I 構成は設定済みであるので、それ以外の制御に必要な設定が実行されるとともに、デバイス情報が採取される（ステップ A 8 ）。

【 0 0 6 2 】

以上で、Linux 側の P C I デバイスの初期設定を終了し（ステップ A 9 ）、VxWorks および Linux は、運用を開始する（ステップ A 1 0 , A 1 0 ）。

次に、図 6 を参照しながら、図 1 に示すストレージ制御装置（情報処理装置）である C

50

M 1 0 0による活性増設シーケンス（ステップB 1 ~ B 1 0 , B 1 , B 1 0 ）について説明する。

【 0 0 6 3 】

活性増設時には、VxWorksが、P C I -Expressデバイス（第2デバイス）のマウントを検出する（ステップB 1）。このとき、Linuxは、デバイスのマウントに気付かずマウントを検出しない（ステップB 1）。

VxWorksは、P C I -Expressデバイスの電源をオンし（ステップB 2）、P C I スキャンを開始する（ステップB 3）。

【 0 0 6 4 】

そして、P C I スキャンの結果に従って、マウントされた第2デバイス用の第2保守ドライバ（保守機能のみ）がロードされる。また、Linux側でマウントされた第2デバイスについて、P C I 構成の設定およびデバイス情報の採取が、第2保守ドライバを用いて行なわれ、採取された情報が、保守用の情報としてメモリ106等に記憶される（ステップB 4）。

【 0 0 6 5 】

以上で、VxWorks側のP C I デバイスの初期設定を終了し（ステップB 5）、Linuxに対し、マウントされたデバイスの組込みの開始通知が行なわれる（ステップB 6）。

これに伴い、LinuxにおいてP C I スキャンが開始される（ステップB 7）。そして、P C I スキャンの結果に従って、マウントされた第2デバイス用の制御ドライバ（デバイスドライバ機能）がロードされる。このとき、P C I 構成は設定済みであるので、それ以外の制御に必要な設定が実行されるとともに、デバイス情報が採取される（ステップB 8）。

【 0 0 6 6 】

以上で、Linux側のP C I デバイスの初期設定を終了し（ステップB 9）、VxWorksおよびLinuxは、新たなデバイスを活性増設した状態で運用を開始する（ステップB 10 , B 10）。

次に、図7を参照しながら、図1に示すストレージ制御装置（情報処理装置）であるC M 1 0 0によるLinux稼働中の活性交換シーケンス（ステップC 1 ~ C 8）について説明する。

【 0 0 6 7 】

Linux稼働中にデバイス（第2デバイス）の活性交換を行なう場合、VxWorksからLinuxへP C I -Expressデバイスのデグレード通知が行なわれる（ステップC 1）。デグレード通知を受けたLinuxは、交換対象のデバイスのドライバ（制御ドライバ）をアンロードし（ステップC 2）、デグレード完了通知をVxWorksへ行なう（ステップC 3）。

【 0 0 6 8 】

Linuxからデグレード完了通知を受けたVxWorksは、交換対象のデバイスのドライバ（第2保守ドライバ）をアンロードし（ステップC 4）、交換対象のP C I -Expressデバイスの電源をオフにする（ステップC 5）。

これにより、P C I -Expressデバイスの保守準備が完了し（ステップC 6）、交換対象のP C I -Expressデバイスの挿抜が実行される（ステップC 7）。つまり、交換対象のデバイスが抜き取られ、新たなデバイスが挿し込まれる。新たなデバイスの挿し込みに応じて、図6の活性増設と同じ処理（ステップB 1 ~ B 10 , B 1 , B 10）が行なわれ（ステップC 8）、Linux稼働中の活性交換が完了する。

【 0 0 6 9 】

次に、図8を参照しながら、図1に示すストレージ制御装置（情報処理装置）であるC M 1 0 0によるLinux停止中の活性交換シーケンス（ステップD 1 ~ D 10）について説明する。

Linux停止中にデバイス（第2デバイス）の活性交換を行なう場合、VxWorksは、交換対象のデバイスのドライバ（第2保守ドライバ）をアンロードし（ステップD 1）、交換対象のP C I -Expressデバイスの電源をオフにする（ステップD 2）。

10

20

30

40

50

【 0 0 7 0 】

これにより、P C I -Expressデバイスの保守準備が完了し（ステップ D 3）、交換対象の P C I -Expressデバイスの挿抜が実行される（ステップ D 4）。つまり、交換対象のデバイスが抜き取られ、新たなデバイスが挿し込まれる。

新たなデバイスの挿し込みに応じて、VxWorksが、新たな P C I -Expressデバイス（第 2 デバイス）のマウントを検出する（ステップ D 5）。これに伴い、VxWorksは、P C I -Expressデバイスの電源をオンし（ステップ D 6）、P C I スキャンを開始する（ステップ D 7）。

【 0 0 7 1 】

そして、P C I スキャンの結果に従って、マウントされた第 2 デバイス用の第 2 保守ドライバがロードされる。また、マウントされた第 2 デバイスについて、P C I 構成の設定およびデバイス情報の採取が、第 2 保守ドライバを用いて行なわれ、採取された情報が、保守用の情報としてメモリ 1 0 6 等に記憶される（ステップ D 8）。

以上で、VxWorks側の P C I デバイスの初期設定を終了し（ステップ D 9）、VxWorksは、Linuxの復旧待ちとなり（ステップ D 1 0）、Linux停止中の活性交換が完了する。

【 0 0 7 2 】

〔 3 〕本実施形態の効果

本実施形態によれば、ハイパバイザ 1 1 上の第 1 O S（VxWorks）1 2 に、第 2 O S（Linux）1 3 に接続された第 2 デバイス 3 0 a，3 0 b 用の第 2 保守ドライバ 3 2 a，3 2 b を備えることで、第 1 O S 1 2 に接続された第 1 デバイス 2 0 a，2 0 b の保守制御手順と同様の手順で、第 1 O S 1 2 から、第 2 O S 1 3 管理下の第 2 デバイス 3 0 a，3 0 b の保守制御等を行なうことが可能になる。

【 0 0 7 3 】

これにより、ストレージ制御装置（情報処理装置）である C M 1 0 0 においてハイパバイザ 1 1 を使用して新しい O S を搭載し当該 O S に新機能を追加して機能強化を図った際、追加した新機能の保守インタフェースを効率的に実装することができる。また、追加した新機能の保守インタフェースとして、従来と同じ保守ガイド画面の構成や保守手順を用いることができ、ユーザは、追加した新機能に対しても違和感なく保守等の操作を行なうことができる。つまり、複数の O S に対して保守制御機能を二重に実装する必要がなく、複数の O S によって管理されるそれぞれのデバイスに対応する保守制御が効率化される。

【 0 0 7 4 】

また、第 1 O S（VxWorks）1 2 は、第 2 保守ドライバ 3 2 a，3 2 b を用いて第 2 O S（Linux）1 3 管理下の第 2 デバイス 3 0 a，3 0 b の情報を採取することができる。したがって、第 2 O S（Linux）1 3 が停止している場合でも、第 1 O S（VxWorks）1 2 が望む情報を、第 1 O S（VxWorks）1 2 が望むタイミングで採取することができる。さらに、図 8 を参照しながら上述したように、第 2 O S（Linux）1 3 が停止しているときでも、第 2 O S（Linux）1 3 側の 第 2 デバイス 3 0 a，3 0 b の保守制御等を行なうことが可能になる。

【 0 0 7 5 】

〔 4 〕その他

以上、本発明の好ましい実施形態について詳述したが、本発明は、係る特定の実施形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲内において、種々の変形、変更して実施することができる。

【 0 0 7 6 】

上述した実施形態では、情報処理装置が、統合ストレージ装置用のストレージ制御装置である場合について説明したが、本発明は、これに限定されるものではない。本発明は、ハイパバイザ上に複数の O S を実装し各 O S にデバイスを接続するよう構成された情報処理装置であれば、上述した実施形態と同様に適用され、上述した実施形態と同様の作用効果を得ることができる。

【 0 0 7 7 】

10

20

30

40

50

また、上述した実施形態では、第1 O AがVxWorksであり、第2 O AがLinuxである場合について説明したが、本発明は、これに限定されるものではない。さらに、上述した実施形態では、第1 デバイスおよび第2 デバイスが2 個ずつ備えられる場合について説明したが、本発明は、これに限定されるものではない。

【0078】

上述したCPU110による処理部としての機能の全部もしくは一部は、コンピュータ(MPU(Micro-Processing Unit), CPU, 各種端末を含む)が所定のプログラムを実行することによって実現されてもよい。そのプログラムは、例えばフレキシブルディスク、CD(CD-ROM, CD-R, CD-RWなど)、DVD(DVD-ROM, DVD-RAM, DVD-R, DVD-RW, DVD+R, DVD+RWなど)、ブルーレイディスク等のコンピュータ読取可能な記録媒体に記録された形態で提供される。この場合、コンピュータはその記録媒体からプログラムを読み取って内部記憶装置または外部記憶装置に転送し格納して用いる。

10

【0079】

(5) 付記

以上の実施形態に関し、更に以下の付記を開示する。

(付記1)

第1オペレーティングシステムによって管理される第1デバイスと、
第2オペレーティングシステムによって管理される第2デバイスと、
前記第1オペレーティングシステムと前記第2オペレーティングシステムとを動作させ、前記第1オペレーティングシステムに対して、前記第1デバイスに対応する第1保守ドライバをロードさせ、前記第1保守ドライバに前記第1デバイスの保守制御を実行させ、前記第1オペレーティングシステムに対して、前記第2デバイスに対応する第2保守ドライバをロードさせ、前記第2保守ドライバに前記第2デバイスの保守制御を実行させる処理部と、
を備えることを特徴とする情報処理装置。

20

【0080】

(付記2)

前記処理部は、前記第2オペレーティングシステムの動作の停止中に、前記第1オペレーティングシステムに対して、前記第2保守ドライバを用いて、前記第2デバイスに係る情報を取得し、前記第2デバイスの保守制御を実行させる、付記1記載の情報処理装置。

30

【0081】

(付記3)

前記処理部は、前記第2オペレーティングシステムの動作の停止中に、前記第1オペレーティングシステムに対して、前記第2保守ドライバを用いて、前記第2デバイスの設定制御を実行させる、付記1または付記2に記載の情報処理装置。

【0082】

(付記4)

前記第1オペレーティングシステムは、SAN(Storage Area Network)制御を行なうSAN用オペレーティングシステムであり、

40

前記第2オペレーティングシステムは、NAS(Network Attached Storage)制御を行なうNAS用オペレーティングシステムである、付記1～付記3のいずれか一項に記載の情報処理装置。

【0083】

(付記5)

前記第1保守ドライバは、前記第1オペレーティングシステムが行なう前記第1デバイスの運用時の制御に使用される制御機能を含む、付記1～付記4のいずれか一項に記載の情報処理装置。

【0084】

(付記6)

50

前記第 2 オペレーティングシステムが行なう前記第 2 デバイスの運用時の制御に使用される制御ドライバを備える、付記 1 ~ 付記 5 のいずれか一項に記載の情報処理装置。

【 0 0 8 5 】

(付記 7)

ストレージユニットを制御するストレージ制御装置であって、

S A N (Storage Area Network) 制御を行なう第 1 オペレーティングシステムによって管理される第 1 デバイスと、

N A S (Network Attached Storage) 制御を行なう第 2 オペレーティングシステムによって管理される第 2 デバイスと、

前記第 1 オペレーティングシステムと前記第 2 オペレーティングシステムとを動作させ、前記第 1 オペレーティングシステムに対して、前記第 1 デバイスに対応する第 1 保守ドライバをロードさせ、前記第 1 保守ドライバに前記第 1 デバイスの保守制御を実行させ、前記第 1 オペレーティングシステムに対して、前記第 2 デバイスに対応する第 2 保守ドライバをロードさせ、前記第 2 保守ドライバに前記第 2 デバイスの保守制御を実行させる処理部と、
を備えることを特徴とするストレージ制御装置。

10

【 0 0 8 6 】

(付記 8)

前記処理部は、前記第 2 オペレーティングシステムの動作の停止中に、前記第 1 オペレーティングシステムに対して、前記第 2 保守ドライバを用いて、前記第 2 デバイスに係る情報を取得し、前記第 2 デバイスの保守制御を実行させる、付記 7 記載のストレージ制御装置。

20

【 0 0 8 7 】

(付記 9)

前記処理部は、前記第 2 オペレーティングシステムの動作の停止中に、前記第 1 オペレーティングシステムに対して、前記第 2 保守ドライバを用いて、前記第 2 デバイスの設定制御を実行させる、付記 7 または付記 8 に記載のストレージ制御装置。

【 0 0 8 8 】

(付記 1 0)

前記第 1 保守ドライバは、前記第 1 オペレーティングシステムが行なう前記第 1 デバイスの運用時の制御に使用される制御機能を含む、付記 7 ~ 付記 9 のいずれか一項に記載のストレージ制御装置。

30

【 0 0 8 9 】

(付記 1 1)

前記第 2 オペレーティングシステムが行なう前記第 2 デバイスの運用時の制御に使用される制御ドライバを備える、付記 7 ~ 付記 1 0 のいずれか一項に記載のストレージ制御装置。

【 0 0 9 0 】

(付記 1 2)

コンピュータに、

第 1 デバイスを管理する第 1 オペレーティングシステムと第 2 デバイスを管理する第 2 オペレーティングシステムとを動作させ、

前記第 1 オペレーティングシステムに対して、前記第 1 デバイスに対応する第 1 保守ドライバをロードさせ、前記第 1 保守ドライバに前記第 1 デバイスの保守制御を実行させ、

前記第 1 オペレーティングシステムに対して、前記第 2 デバイスに対応する第 2 保守ドライバをロードさせ、前記第 2 保守ドライバに前記第 2 デバイスの保守制御を実行させる、プログラム。

40

【 0 0 9 1 】

(付記 1 3)

前記コンピュータに、

50

前記第 2 オペレーティングシステムの動作の停止中に、前記第 1 オペレーティングシステムに対して、前記第 2 保守ドライバを用いて、前記第 2 デバイスに係る情報を取得し、前記第 2 デバイスの保守制御を実行させる、付記 1 2 記載のプログラム。

【 0 0 9 2 】

(付記 1 4)

前記コンピュータに、

前記第 2 オペレーティングシステムの動作の停止中に、前記第 1 オペレーティングシステムに対して、前記第 2 保守ドライバを用いて、前記第 2 デバイスの設定制御を実行させる、付記 1 2 または付記 1 3 に記載のプログラム。

【 0 0 9 3 】

(付記 1 5)

前記第 1 オペレーティングシステムは、S A N (Storage Area Network) 制御を行なう S A N 用オペレーティングシステムであり、

前記第 2 オペレーティングシステムは、N A S (Network Attached Storage) 制御を行なう N A S 用オペレーティングシステムである、付記 1 2 ~ 付記 1 4 のいずれか一項に記載のプログラム。

【 0 0 9 4 】

(付記 1 6)

前記第 1 保守ドライバは、前記第 1 オペレーティングシステムが行なう前記第 1 デバイスの運用時の制御に使用される制御機能を含む、付記 1 2 ~ 付記 1 5 のいずれか一項に記載のプログラム。

【 符号の説明 】

【 0 0 9 5 】

1 ストレージ装置

1 1 ハイパバイザ (仮想化 O S)

1 2 第 1 オペレーティングシステム (S A N 用 O S ; VxWorks)

1 3 第 2 オペレーティングシステム (N A S 用 O S ; Linux)

2 0 a , 2 0 b 第 1 デバイス

2 1 a , 2 1 b 第 1 保守ドライバ (フル機能ドライバ)

3 0 a , 3 0 b 第 2 デバイス

3 1 a , 3 1 b 制御ドライバ (デバイスドライバ機能)

3 2 a , 3 2 b 第 2 保守ドライバ (保守機能)

1 0 0 C M (ストレージ制御装置 , 情報処理装置)

1 0 1 , 1 0 2 C A

1 0 3 D A

1 0 6 メモリ

1 0 7 S S D

1 0 8 I O C

1 0 9 スロット

1 1 0 C P U (処理部)

3 0 0 ドライブエンクロージャ (ストレージユニット)

3 1 0 記憶装置 (ストレージユニット)

4 0 0 コントローラエンクロージャ

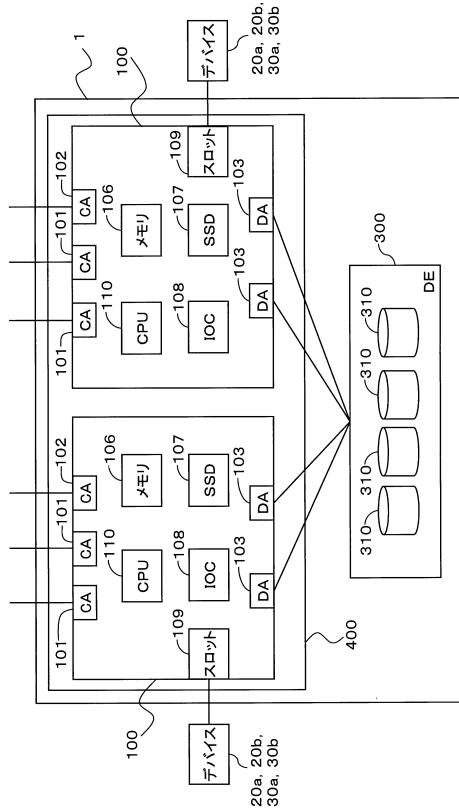
10

20

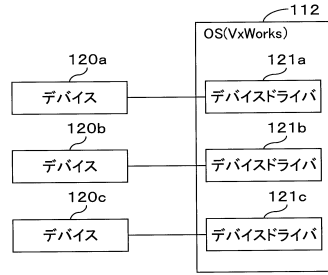
30

40

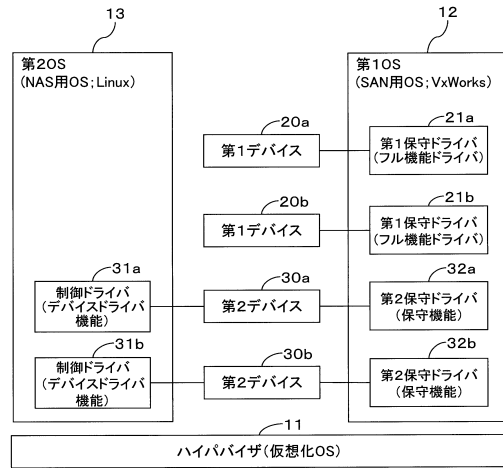
【図1】



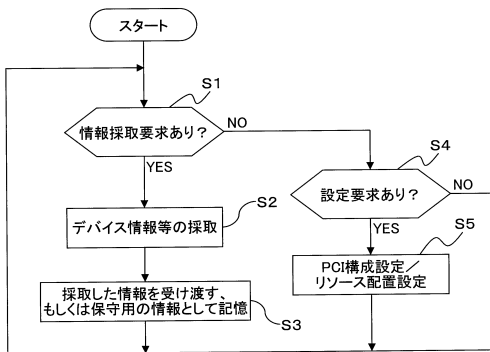
【図2】



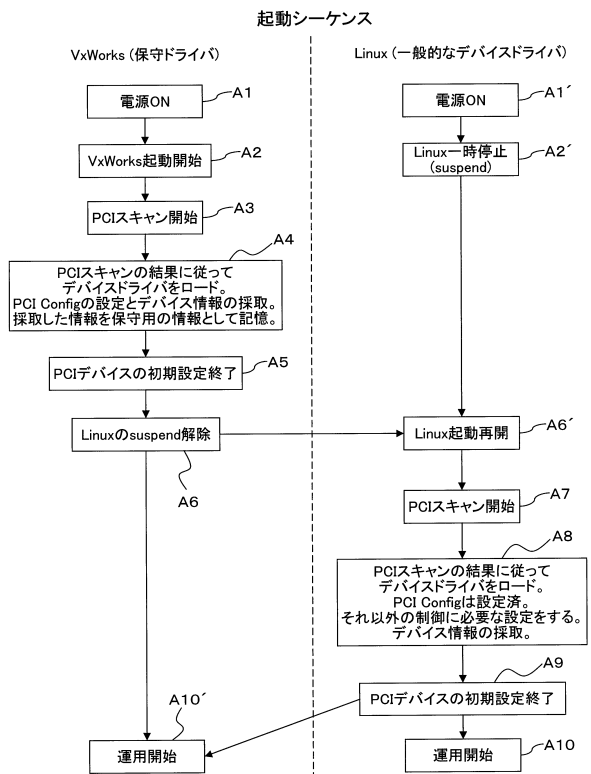
【図3】



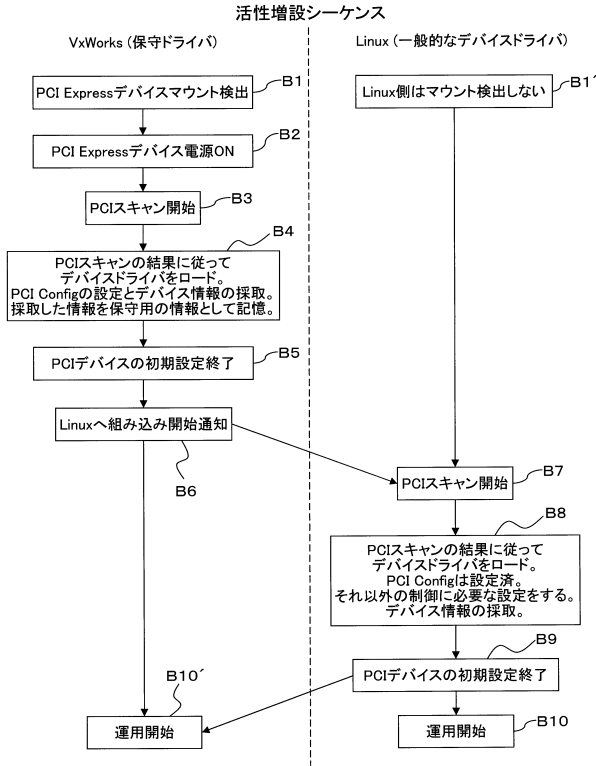
【図4】



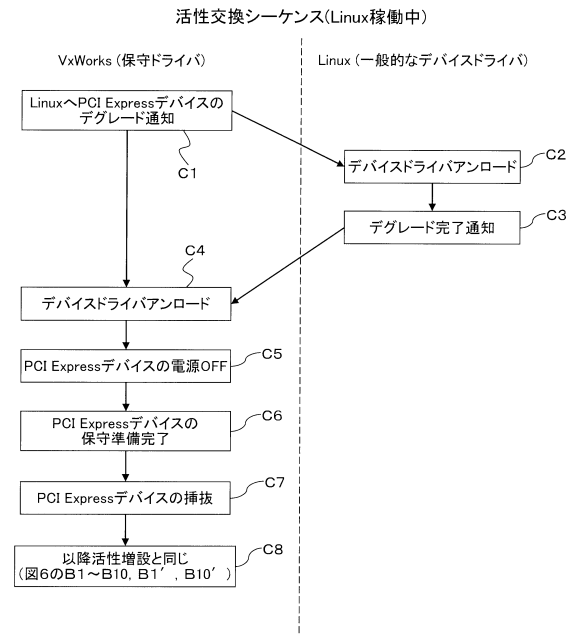
【図5】



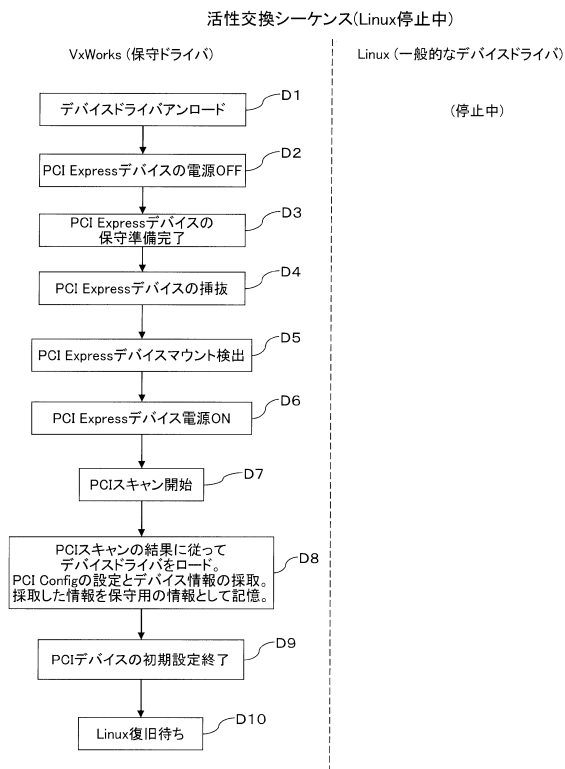
【図6】



【図7】



【図8】



フロントページの続き

- (56)参考文献 特開2001-022598(JP,A)
特開2008-305034(JP,A)
米国特許出願公開第2010/0058335(US,A1)
特開2009-258798(JP,A)
特開2012-009029(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F3/06-3/08
G06F9/46、12/00
G06F13/10-13/14