

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4912731号
(P4912731)

(45) 発行日 平成24年4月11日(2012.4.11)

(24) 登録日 平成24年1月27日(2012.1.27)

(51) Int.Cl. F I
G06F 13/00 (2006.01) G O 6 F 13/00 3 O 1 P
G06F 3/06 (2006.01) G O 6 F 3/06 3 O 4 B
G06F 13/14 (2006.01) G O 6 F 13/14 3 I O E

請求項の数 5 (全 8 頁)

(21) 出願番号	特願2006-115312 (P2006-115312)	(73) 特許権者	511076424
(22) 出願日	平成18年4月19日 (2006.4.19)		ヒューレット・パカード デベロップメント カンパニー エル.ピー.
(65) 公開番号	特開2006-302287 (P2006-302287A)		Hewlett-Packard Development Company, L.P.
(43) 公開日	平成18年11月2日 (2006.11.2)		アメリカ合衆国 テキサス州 77070
審査請求日	平成18年4月19日 (2006.4.19)		ヒューストン コンパック センタ ドライブ ウェスト 11445
審判番号	不服2010-8851 (P2010-8851/J1)	(74) 代理人	100087642
審判請求日	平成22年4月26日 (2010.4.26)		弁理士 古谷 聡
(31) 優先権主張番号	11/109309	(74) 代理人	100076680
(32) 優先日	平成17年4月19日 (2005.4.19)		弁理士 溝部 孝彦
(33) 優先権主張国	米国 (US)	(74) 代理人	100121061
			弁理士 西山 清春

最終頁に続く

(54) 【発明の名称】 冗長 I / O インターフェース管理

(57) 【特許請求の範囲】

【請求項 1】

冗長 I / O インターフェースマネージャであって、

第 1 の非冗長 I / O インターフェースモジュールに接続するための第 1 の接続部と、
 第 2 の非冗長 I / O インターフェースモジュールに接続するための第 2 の接続部と、
 前記非冗長 I / O インターフェースモジュールの両方に接続された外部システムとの
 通信を前記第 1 の非冗長 I / O インターフェースモジュールが停止するように、及び、前
 記第 2 の非冗長 I / O インターフェースモジュールが前記外部システムとの通信を開始す
 るように、前記非冗長 I / O インターフェースモジュールに対する設定を切り換えるため
 のコントローラ
 とを備え、

前記コントローラが、ホストシステムとインターフェースし、その結果、前記冗長 I /
 O インターフェースマネージャは、該ホストシステムの I / O インターフェースモジュ
 ールドライバにとって、該冗長 I / O インターフェースマネージャが、あたかも、非冗長 I
/ O インターフェースモジュール同士で互換性のある同じ種類の非冗長 I / O インターフ
ェースモジュールの 1 セットのうちの 1 つの非冗長 I / O インターフェースモジュール
であるかのように見え、前記ドライバは、該 1 セットのうちの 1 つの非冗長 I / O インター
フェースモジュールを駆動するよう設計されており、前記セットは、前記第 1 及び第 2 の
非冗長 I / O インターフェースモジュールを含み、

前記第 1 の非冗長 I / O インターフェースモジュールの障害の検出か又は予測に回答し

て、前記コントローラが、前記設定を切り換え、及び、

前記設定を切り換える前に、前記コントローラは、前記ドライバからのコンフィギュレーションデータを、前記第1の非冗長I/Oインターフェースモジュールと前記第2の非冗長I/Oインターフェースモジュールとの両方に対して送出して、その結果、該第1の非冗長I/Oインターフェースモジュールと該第2の非冗長I/Oインターフェースモジュールとの両方が、少なくとも幾つかの同じコンフィギュレーション変更を受けることとなるようにすることからなる、冗長I/Oインターフェースマネージャ。

【請求項2】

前記冗長I/Oインターフェースマネージャが、I/Oブリッジチップに接続され、該I/Oブリッジチップを介して、データが、前記ドライバと前記コントローラとの間において転送されることからなる、請求項1に記載の冗長I/Oインターフェースマネージャ

10

【請求項3】

非冗長I/Oインターフェースモジュール同士で互換性のある同じ種類の複数の非冗長I/Oインターフェースモジュールのうち1つの非冗長I/Oインターフェースモジュールが行うものとして、I/Oインターフェースモジュールドライバからの通信に冗長I/Oインターフェースマネージャが応答し、ここで、前記ドライバは、該1つの非冗長I/Oインターフェースモジュールと共に動作するよう設計されており、

前記応答中、前記ドライバと外部システムとの間の通信データを、第1の非冗長I/Oインターフェースモジュールと前記外部システムの第1のポートとを含む第1の経路を通して、且つ、第2の非冗長I/Oインターフェースモジュールと前記外部システムの第2のポートとを含む第2の経路を通さずに、前記冗長I/Oインターフェースマネージャが導き、ここで、前記ドライバは、非冗長I/Oインターフェースモジュールの1セットのうち1つの非冗長I/Oインターフェースモジュールと共に動作するよう設計されており、該セットは、前記第1及び第2の非冗長I/Oインターフェースモジュールを含み、及び、

20

続いて、前記互換性のある同じ種類の非冗長I/Oインターフェースモジュールが行うものとして前記ドライバに**応答し続ける間に、前記第1の経路から前記第2の経路に通信経路を前記冗長I/Oインターフェースマネージャが切り換える**

ことを含む、方法であって、

30

前記冗長I/Oインターフェースマネージャが、前記第1の非冗長I/Oインターフェースモジュールの障害を検出するか又は予測し、及び、

前記検出するか又は予測することに**応答して、前記切り換えを実施する前に、前記冗長I/Oインターフェースマネージャが、更に、**

前記第1の非冗長I/Oインターフェースモジュールに対して意図されたコンフィギュレーションデータを受け取り、及び、

前記コンフィギュレーションデータのコピーを、前記第1の非冗長I/Oインターフェースモジュールと前記第2の非冗長I/Oインターフェースモジュールとの両方に対して提供し、その結果、それらのコンフィギュレーション状態が**コヒーレントのままとなる**ようにすることからなる、方法。

40

【請求項4】

前記外部システムがネットワークであることからなる、請求項3に記載の方法。

【請求項5】

前記外部システムがマルチポートディスクアレイであることからなる、請求項3に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータに関し、特に、コンピュータのためのI/O(「入力/出力」)サブシステムに関する。本明細書内において、「従来技術」とラベル付けられた関連技

50

術は、従来技術であると認められており、「従来技術」とラベル付けられていない関連技術は、従来技術であると認めていない。

【背景技術】

【0002】

現代社会におけるコンピュータの普及は、汎用コンピュータが、往々にしてサードパーティの構成要素である、標準規格品としての構成要素を使用して組み立てられ、保守され、及びアップグレードされることを可能にするインターフェース規格が遵守されていることに部分的に起因する。欠陥構成要素によるダウンタイム（又は故障による停止時間）が非常にコストがかかる用途において使用される、高可用性（又は高稼働率）コンピュータは、構成要素が典型的には高可用性基準を満たすために特別に設計されなければならないため、標準規格から汎用コンピュータが受けるのと同じほどには恩恵を受けてこなかった。例えば、ネットワーク及びディスクアレイ I/O インターフェースカードのような幾つかの構成要素を、一方に障害が発生した場合には他方が動作を大幅に中断させることなく引き継ぐことができるように、冗長なグループで構成することができる。

10

【発明の開示】

【発明が解決しようとする課題】

【0003】

前記特別な設計は、冗長動作のために設計された特別なハードウェアだけでなく、特別なソフトウェア（例えば冗長構成要素を管理するために設計されたオペレーティングシステム及びドライバ）も頻繁に含む。これらはその結果として、大量の技術設計リソースと長期間にわたる設計及び開発スケジュールとを要する（これらは急速に展開する市場において問題があることである）。

20

【課題を解決するための手段】

【0004】

本発明は、特許請求の範囲内において画定されるものとして、マルチパスターゲット（例えばネットワーク及びマルチパスディスクアレイ）に対する、標準規格品としての I/O インターフェースモジュール（例えば I/O インターフェースカード）の、冗長構成を管理するための冗長 I/O インターフェースマネージャ（Redundant I/O Interface Manager: R I M）を提供し、一方で、単一 I/O インターフェースカードが存在していることを I/O インターフェースカードドライバに対して見えさせる（又はわからせる）。本発明により、I/O インターフェースカードのための特別なドライバの必要性を不要にし、冗長動作のために設計されていないストックドライバを使用することができる。標準規格品としての I/O インターフェースカード及びドライバを使用することができるため、製造、保守、及びアップグレードにおいて、大幅なコスト削減を達成することができる。更に、本発明は、高い信頼性/可用性のコンピュータを設計するために必要なリソースを低減させ、より高速な開発時間を提供し、従って、よりタイムリーなリリーススケジュールを達成させる。本発明の、これらの及び他の特徴及び利点は、添付図面に関連して下記の説明から明らかである。

30

【発明の効果】

【0005】

特別な設計を必要とせずに、冗長性を持たせた I/O インターフェースによって障害発生時に動作を正常動作に切り換えることができる。

40

【実施例】

【0006】

本発明によるコンピューティングシステム A P 1 が、コンピュータ 1 1 及びディスクアレイ 1 3 を含む図 1 内に示されている。ディスクアレイ 1 3 は、ポート 1 5 と 1 7 とにおける 2 つの独立した接続を提供する。典型的な構成において、該 2 つの接続は 2 つの異なるコンピュータに対するものである。該 2 つの接続は、この場合には、コンピュータシステム 1 1 の、2 つの異なるディスクアレイ I/O インターフェースカード 2 1 及び 2 3 に対するものである。他の実施形態において、ターゲットは、ネットワークであり、I/O

50

インターフェースカードは、ネットワークI/Oインターフェースカードである。より一般的には、I/Oインターフェースカードは、2つか又はそれよりも多くの利用可能な接続部を有する、他のタイプのデバイスに、接続することができる。

【0007】

コンピュータシステム11は、プロセッサ25及び27、メモリ29、入力-出力(I/O)ブリッジ31、冗長I/Oインターフェースマネージャ33、及びI/Oインターフェースカード21及び23、並びに、他の構成要素を備える。プロセッサ25及び27、メモリ29、及びI/Oブリッジ31は、バス35として概略的に示された通信構造を介して通信可能に接続される。I/Oブリッジ31は、I/Oブリッジに対するPCIバスインターフェース43を介して、冗長I/Oインターフェースマネージャ33のシステムポート41に結合される。I/Oインターフェースカード21と23とは、バスインターフェース48と49とによって、冗長I/Oインターフェースマネージャ33のI/Oポート45と47とにそれぞれ結合される。代替の実施形態において、PCIとは別のI/O通信プロトコル及び技術が使用される。冗長I/Oインターフェースマネージャ33のコントローラ50は、そのポート41、45、及び47との間の相互作用を管理する。

10

【0008】

メモリ29は、ランダムアクセスメモリと内部ハードディスクとの両方を含む。メモリ29は、データ51と、オペレーティングシステム53、アプリケーション55、及びI/Oドライバ57を含むプログラムとを格納する。I/Oブリッジ31が、いくつかの接続59を有することに留意されたい。すなわち、図1内において、接続されるデバイスが図示されていないが、それらは他のI/Oデバイスを含むことができ、該他のI/Oデバイスのうちのいくつかは、冗長構成になっているが、他のものはそうっていない。

20

【0009】

I/Oインターフェースカード21と23とは、それらが同じ製造業者によるものであることから名目上同一であり、同一のドライバが提供される。I/Oドライバ57は、I/Oインターフェースカード21と23との両方のために使用されるドライバの一例だけを含む。初期化時には、冗長I/Oインターフェースマネージャ33は、「アクティブな」カードとしてカードのうちの一方を(例えばカード21を)選択し、「スペアの」カードとして他方を(例えばカード23を)選択する。ディスクアレイ13との通信は、現在アクティブなカードにより単独で行われる。冗長I/Oインターフェースマネージャ33は、オペレーティングシステム53にとって単一のI/Oインターフェースカードとして見える、I/Oインターフェースカードに対するプロキシ(又は代理)としての機能を果たす。冗長動作をサポートするために、ドライバソフトウェアを修正する必要が無い。

30

【0010】

通常動作中、RIMコントローラ50は、書き込まれているトランザクションIDとアドレス空間とに基づいてコンフィギュレーションデータ(又は設定データ、又は構成データ)を認識することができる。RIMコントローラ50は、アクティブなI/OインターフェースカードとスペアのI/Oインターフェースカードとの両方によってコンフィギュレーションデータが受け取られるようにI/Oインターフェースカードが見えることを意図された該コンフィギュレーションデータを自動的にミラーリングする。従って、スペアは、アクティブなカードと同じコンフィギュレーション(又は設定、又は構成)において維持される。切換えが生じる時には、スペアはドライバによって期待された状態にある。

40

【0011】

現在アクティブなカードに障害が発生したイベントにおいて、RIMコントローラ50は、スペアカードへの切換えを管理する。今までアクティブであったカードによる通信は終了させられて、次いでスペアによってアクティブにされる(又はスペアがアクティブにされる)。RIMコントローラ50は、切換えの行うためにかかる時間中に起こり得るタイムアウトを除いて、OS53にとって不可視であるような手法で切換えを管理する。典型的には、タイムアウトのイベントにおいて、データの消失が起こらないように、通信のリトライが引き起こされる。アクティブI/OインターフェースカードとスペアI/Oイ

50

ンターフェースカードとの両方に障害が発生した時のみ、P C Iバスエラーが発生する。

【0012】

ネットワークA P 1の状況において実施されるような本発明の方法M 1が、図2において、フローチャートに示されている。システム11は、方法セグメントS 11において起動される。方法セグメントS 12において、R I M 33は、その2つのスロット内におけるI / Oインターフェースカードの存在をチェックし、少なくとも1つのI / Oインターフェースカードが存在する場合には、「存在 (presence)」フラグを設定する。方法セグメントS 13において、2つのカードが存在していると仮定すると、R I M 33は、I / Oインターフェースカードのうち的一方を (例えばカード21を)、一次I / Oインターフェースカードとなるように選択し、他方を (例えばカード23を)、二次I / Oインターフェースカードとなるように選択する。前記一次カードは、デフォルトで「アクティブ」であるが、二次I / Oインターフェースカードは、デフォルトで「スペア」である。

10

【0013】

方法セグメントS 14において、システムファームウェアは、I / Oバスを1つずつ調べて、I / Oインターフェースカードを探す。カード21と23とを読み出す代わりに、I / Oインターフェースカードのプロキシとしての機能を果たすR I M 33内において設定された存在フラグを読み出す。存在フラグが設定されていると仮定すると、方法セグメントS 15において、システムファームウェアは、検出する「カード」を初期化することを試みる。このことには、I / Oアドレスを設定すること、モードビットを設定すること、マイクロコードを提供すること等を含めることができる。方法セグメントS 16において、R I M 33は、2つのI / Oインターフェースカード21と23との間の全てのセットアップトランザクションをミラーリングする。この時点で、I / Oインターフェースカード21及び23は同一にセットアップされる。動作中、オペレーティングシステム53が新たなコンフィギュレーションデータを送出する場合には、R I M 33はまた、それをI / Oインターフェースカード21と23との両方に対してミラーリングし、その結果、それらのコンフィギュレーション状態がコヒーレントなまま (又は整合がとれたまま) となる。方法セグメントS 17において、ファームウェアは、立ち上がる時にオペレーティングシステム53に対しアドレスマップを提示する。ここでもまた、オペレーティングシステム53及びドライバ57にとって、冗長ペア21と23とが単一アドレスを有する単一I / Oインターフェースカードとして見える。

20

30

【0014】

方法セグメントS 18において、通常動作中、R I M 33は、ブリッジ31を介してオペレーティングシステム53からの読み出し / 書き込み動作を受け入れる。R I M 33は、トランザクションをそれが完了するまで保持する。方法セグメントS 19において、R I M 33は、アクティブなI / Oインターフェースカード (例えばカード21) に対する動作を進行させる。ディスクアレイ13に関連した要求された転送が成功した場合には、R I M 33は、方法セグメント20において読み出し / 書き込み動作を完了する。

【0015】

トランザクションが成功しなかった場合には、R I M 33は、方法セグメントS 21において切り換えを実行する。新たにアクティブとなったI / Oインターフェースカード (例えばカード23) によって、ディスクアレイ13とのトランザクションが成功した場合には、R I M 33は、方法セグメントS 20において読み出し / 書き込み動作を完了する。もしもそうでなく、オペレーティングシステム53から見て、読み出し / 書き込み動作がタイムアウトした場合には、方法M 1は通常、リトライのために方法セグメントS 18に戻ることになる。おそらくリトライは成功するであろう。しかしながら、両方のカードに障害が発生している場合には、トランザクションを完了させることができない。この場合を、非冗長コンフィギュレーションにおける単一I / Oインターフェースカードの障害と同様の手法において処理することができる。

40

【0016】

50

方法 M 1 において、アクティブなカードの障害が検出された時に切換えが生じる。しかしながら、他の状況において同様に切換えを生じさせることも可能である。例えば、障害の予測に応答して、例えば R I M 3 3 がアクティブカードに関連するトランザクションにおいて過度なエラーを検出した時には、切換えを生じさせることができる。I / O インターフェイスカード間のデューティサイクルのバランスをとることに役立つように切換えを実行することもできる。代替の実施形態において、冗長 I / O インターフェイスカードは、オペレーティングシステムにとって可視であるが、特定の I / O インターフェイスカードドライバにとっては可視ではなく、この代替の実施形態において、OS が切換えを要求することができる。

【 0 0 1 7 】

本発明は、任意の数のプロセッサ及び任意のメモリアーキテクチャを有するシステムを提供する。冗長性には、2つか又はそれよりも多くの I / O インターフェイスモジュールを関与させることができる。3つか又はそれよりも多くの I / O インターフェイスモジュールの構成を有するいくつかの実施形態において、本発明は、2つ以上のアクティブな I / O インターフェイスモジュールを提供する。例示された実施形態において、両方の I / O インターフェイスカードに対して1つのドライバだけが用いられているが、本発明は、冗長インターフェイスモジュールが同一ドライバを使用する必要がないように、異なるドライバをやり繰り返すことができる冗長管理ソフトウェアを更に提供する。例示された実施形態において、I / O インターフェイスモジュールを「カード」として説明することができるが、本発明は他の形態要素を有するモジュールを提供する。添付の特許請求の範囲内によって範囲が画定される本発明によって、記載された実施形態に対するこれらの及び他の変形形態と修正とが提供される。

【 図面の簡単な説明 】

【 0 0 1 8 】

【 図 1 】 本発明によって提供される多くの実現可能なコンピュータシステムのうちの1つの概略ブロック図である。

【 図 2 】 本発明によって提供される多くの方法のうちの1つのフローチャートである。

【 符号の説明 】

【 0 0 1 9 】

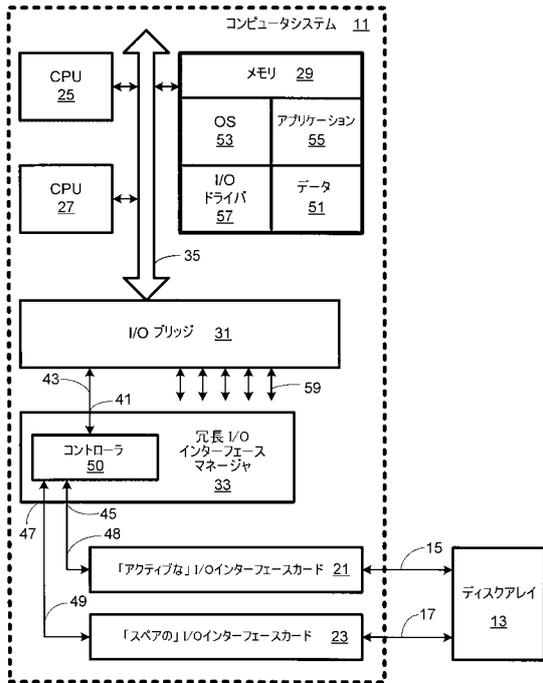
- 1 3 外部システム
- 2 1 第1の I / O インターフェイスモジュール
- 2 3 第2の I / O インターフェイスモジュール
- 3 3 冗長 I / O インターフェイスマネージャ
- 5 7 I / O インターフェイスモジュールドライバ

10

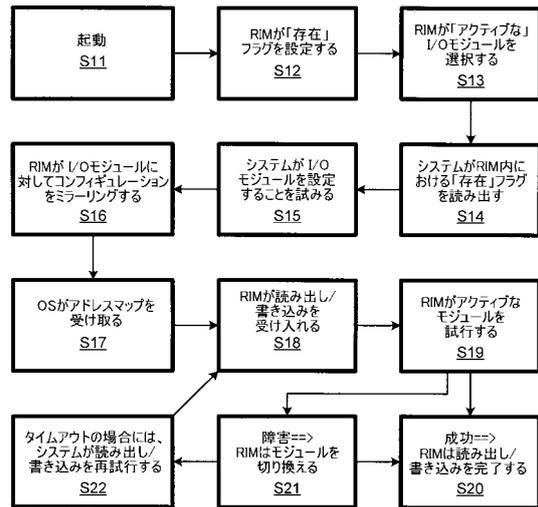
20

30

【図 1】



【図 2】



フロントページの続き

- (72)発明者 ケン・ジー・ボマランスキー
アメリカ合衆国カリフォルニア州 パロアルト ハノーバー・ストリート 3000 ヒューレ
ット・パッカー・カンパニー内
- (72)発明者 アンドリュー・エイチ・バール
アメリカ合衆国カリフォルニア州 パロアルト ハノーバー・ストリート 3000 ヒューレ
ット・パッカー・カンパニー内
- (72)発明者 デール・ジェイ・シドラ
アメリカ合衆国カリフォルニア州 パロアルト ハノーバー・ストリート 3000 ヒューレ
ット・パッカー・カンパニー内

合議体

審判長 清水 稔
審判官 安島 智也
審判官 稲葉 和生

- (56)参考文献 特開2000-284982(JP,A)
特開2003-330817(JP,A)
国際公開第02/8914(WO,A1)
特開2002-324042(JP,A)
特開平8-314853(JP,A)
米国特許第(US,B1)6272113
Prem Mohan Mendhikar、How to Configure NIC T
eaming Drivers Using INETCFG、2002年10月1日、<UR
L:http://support.novell.com/techcenter/arti
cles/ana20021003.html>

(58)調査した分野(Int.Cl., DB名)

G06F 13/00, G06F 3/06, G06F 13/14