

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4667925号  
(P4667925)

(45) 発行日 平成23年4月13日(2011.4.13)

(24) 登録日 平成23年1月21日(2011.1.21)

(51) Int. Cl.	F I		
<b>G06F 3/06 (2006.01)</b>	G06F	3/06	301K
<b>G06F 12/08 (2006.01)</b>	G06F	3/06	301J
	G06F	12/08	541C
	G06F	12/08	557

請求項の数 10 (全 16 頁)

(21) 出願番号	特願2005-97433 (P2005-97433)	(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 MACHINES 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
(22) 出願日	平成17年3月30日(2005.3.30)	(74) 代理人	100108501 弁理士 上野 剛史
(65) 公開番号	特開2005-293586 (P2005-293586A)	(74) 代理人	100091568 弁理士 市位 嘉宏
(43) 公開日	平成17年10月20日(2005.10.20)	(74) 代理人	100086243 弁理士 坂口 博
審査請求日	平成17年3月30日(2005.3.30)		
審判番号	不服2007-30597 (P2007-30597/J1)		
審判請求日	平成19年11月12日(2007.11.12)		
(31) 優先権主張番号	10/815319		
(32) 優先日	平成16年3月31日(2004.3.31)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 書込み処理管理のための方法、システム、およびプログラム

(57) 【特許請求の範囲】

【請求項1】

複数のホスト装置からの記憶システムへの書込みを制御する記憶コントローラが実行する書込み制御方法であって、

ネットワークを介して接続されたホスト装置から書込み操作の宛先の位置IDを取得するステップと、

書込み操作中の宛先位置を維持する配列であるアレイ中の複数のアレイ項目のうち次に使用可能な項目を示す第1の値を有し、インデックス値として用いられるアレイ・ポインタであって、現在書き込み中および書き込み済みの位置IDを保管している複数のアレイ項目がスタックを形成するデータ構造におけるスタック・インデックス項目に保管された当該アレイ・ポインタによって、前記次に使用可能な項目を識別するステップと、

識別された前記次に使用可能な項目に、前記取得するステップで取得された書込み操作の宛先の位置IDを保管し、前記アレイ中の前記スタックの末尾の次のアレイ項目を前記スタックでの変更後次に使用可能なアレイ項目として識別するため、前記インデックス値を増分することにより、前記複数のアレイ項目がスタックを形成した前記データ構造におけるスタック・インデックス項目に保管された前記アレイ・ポインタの前記第1の値を第2の値に変更するステップと、

前記書込み操作の完了時に、前記アレイ・ポインタの前記インデックス値を減分して、前記複数のアレイ項目のスタックから、前記完了した書込み操作の宛先の前記位置IDを除去して前記複数のアレイ項目のスタックのサイズを低減するステップと

10

20

を含み、

前記低減するステップは、前記アレイ中の前記スタックの末尾の項目を前記アレイ中の低減後、次に使用可能な項目として識別するために前記アレイ・ポインタのインデックス値を減分した後、書込み操作中のインデックス値と位置IDとを維持するキャッシュ・メモリ内の前記インデックス値と、前記減分されたインデックス値とを比較し、前記インデックス値と前記減分されたインデックス値とが一致していない場合には、前記キャッシュ・メモリ内に維持されたアレイ・ポインタで識別される位置IDを前記位置IDを削除した前記アレイ項目に移動することにより、前記スタックから書込み操作が完了した前記複数のアレイ項目のうちの最後の項目を除去するステップと

を含む、方法。

10

【請求項2】

書込み操作の完了時に、前記複数のアレイ項目のスタックの前記最後の項目に保管された前記位置IDおよび前記インデックス値を前記キャッシュ・メモリのフィールドに移動し、前記移動した位置IDで前記完了した書込み操作の前記位置IDを上書きするステップと、前記アレイの前記前の項目を前記アレイの前記次に使用可能な項目として識別するために前記アレイ・ポインタのインデックス値を減分し、前記複数のアレイ項目のスタックの前記最後の項目を除去するステップと、  
をさらに有する、請求項1に記載の方法。

【請求項3】

前記低減したスタックの内容の論理冗長検査を更新するステップをさらに有する、請求項1に記載の方法。

20

【請求項4】

前記アレイは、不揮発性記憶メモリを含むデジタル・データ処理装置の論理的なデータ構造である、請求項1に記載の方法。

【請求項5】

デジタル・データ処理装置が、請求項1～4のいずれか1項に記載の方法の各ステップを実行するための装置実行可能なプログラム。

【請求項6】

複数のホスト装置からの記憶システムへの書込みを制御するシステムであって、  
書込み操作中の宛先位置を維持する配列であるアレイ中の使用可能な項目を識別する第1の値を有し、インデックス値として用いられるアレイ・ポインタを、現在書き込み中および書き込み済みの位置IDを保管している複数のアレイ項目がスタックを形成するデータ構造におけるスタック・インデックス項目に保管するように適合された記憶装置と、  
前記記憶装置に結合されたデジタル・データ処理装置であって、前記デジタル・データ処理装置が、  
ネットワークを介して接続された前記ホスト装置から書込み操作の宛先の位置IDを取得し、

30

前記アレイの複数のアレイ項目のうち、次に使用可能なアレイ項目を前記第1の値を有するアレイ・ポインタによって識別し、

識別された前記次に使用可能な項目に、取得された書込み操作の宛先の位置IDを保管し、前記アレイ中の前記スタックの末尾の次のアレイ項目を前記スタックでの変更後次に使用可能なアレイ項目として識別するため、前記インデックス値を増分することにより、前記複数のアレイ項目がスタックを形成した前記データ構造におけるスタック・インデックス項目に保管された前記アレイ・ポインタの前記第1の値を第2の値に変更し、

40

前記書込み操作の完了時に、前記アレイ・ポインタの前記インデックス値を減分して、前記複数のアレイ項目のスタックから、前記完了した書込み操作の宛先の前記位置IDを除去して前記複数のアレイ項目のスタックのサイズを低減し、

前記低減では、前記アレイ中の前記スタックの末尾の項目を前記アレイ中の低減後次に使用可能な項目として識別するために前記アレイ・ポインタのインデックス値を減分した後、書込み操作中のインデックス値と位置IDとを維持するキャッシュ・メモリ内の前記

50

インデックス値と、前記減分されたインデックス値とを比較し、前記インデックス値と前記減分されたインデックス値とが一致していない場合には、前記キャッシュ・メモリ内に維持されたアレイ・ポインタで識別される位置IDを前記位置IDを削除した前記アレイ項目に移動することにより、前記スタックから書込み操作が完了した前記複数のアレイ項目のうちの最後の項目を除去する

システム。

【請求項7】

前記アレイ・ポインタがインデックス値で指定され、前記変更が、前記アレイの次の項目を前記アレイの前記次の使用可能な項目として識別するために前記アレイ・ポインタのインデックス値を増分することを含む、請求項6に記載のシステム。

10

【請求項8】

前記デジタル・データ処理装置が、書込み操作の完了時に、前記複数のアレイ項目のスタックの前記最後の項目に保管された前記位置IDおよび前記インデックス値を前記キャッシュ・メモリのフィールドに移動し、前記移動した位置IDで前記完了した書込み操作の前記位置IDを上書きし、前記アレイの前記前の項目を前記アレイの前記次の使用可能な項目として識別するために前記アレイ・ポインタのインデックス値を減分し、前記複数のアレイ項目のスタックの前記最後の項目を除去するようにさらに構成される、請求項6に記載のシステム。

【請求項9】

前記デジタル・データ処理装置が、前記低減したスタックの内容の論理冗長検査を更新するようにさらに構成される、請求項6に記載のシステム。

20

【請求項10】

前記記憶装置が不揮発性記憶メモリである、請求項6に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、記憶装置への書込み処理を管理するための方法、システム、およびプログラムに関する。

【背景技術】

【0002】

コンピューティング・システムは、データを処理し、アプリケーション・プログラムを実行するための1つまたは複数のホスト・コンピュータ(「ホスト」)と、データを保管するための直接アクセス記憶装置(DASD)と、ホストとDASDとの間のデータの転送を制御するための記憶コントローラとを含む場合が多い。制御ユニットまたは記憶ディレクタとも呼ばれる記憶コントローラは、さもなければ直接アクセス記憶装置(DASD)と呼ばれる、ループ・アーキテクチャ内に接続された多数のハード・ディスク・ドライブからなる場合が多い記憶空間へのアクセスを管理する。ホストは、記憶コントローラにより記憶空間に入出力(I/O)要求を伝達することができる。

30

【0003】

ディスク状の記憶媒体は複数の物理的部分に細分することができ、そのそれぞれは「パイ状」になっており、「セクタ」と呼ばれることが多い。その円形の媒体をパイ・スライスに分割することは、ドライブの読取り/書込みヘッドによるデータの位置特定を容易にする。また、ディスクまたはディスクは、概して、いくつかの同心円またはトラックに分割される。各トラックは、概して、読取りまたは書込み操作のためにアドレス可能な最小記憶単位を表す、いくつかのクラスタに分割される。概して、1つのクラスタは256または512バイトの長さである。したがって、データは、そのデータが保管されているセクタ、同心トラック、およびクラスタのうちの1つまたは複数を選択することによって位置を特定することができる。

40

【0004】

書込み操作において、データを記憶媒体に書き込むことができる。電源障害または装置

50

の誤動作によって書込み操作が中断された場合、記憶媒体に書き込まれたデータは破壊されるか、さもなければ不完全なものになる可能性がある。したがって、記憶コントローラは、データが書き込まれている宛先位置のリストを頻繁に維持する。書込み操作中の場合の宛先位置のリストは、電力が失われた場合でもリスト情報が維持されるように、不揮発性記憶（NVS）メモリ内に保持される場合が多い。

**【0005】**

特定の宛先位置への書込み操作が完了すると、その特定の書込み操作の宛先位置はリストから除去することができる。しかし、完了前に書込み操作が中断された場合、書込み操作中の場合の宛先位置のリストを検査して、不完全なまたは破壊されたデータを収容している可能性のある書込み宛先位置を識別することができる。

10

**【0006】**

図1は、典型的な従来技術の書込み操作中の宛先位置のリスト10の一例を示している。図1の例では、リスト10は複数の項目12a、12b、・・・12nを有し、それぞれは書込み操作中の宛先位置を保管することができる。図1の例では、宛先位置は、たとえば、リスト項目12cによって示す通り、トラック識別子（「トラックID」）によって識別される。リスト項目12a、12b、・・・12nのうちの1つへの各トラックIDの入力後、リスト10の内容について論理的冗長検査を実行することができ、その検査結果は項目14に保管することができる。

**【0007】**

書込み操作の完了時に、その書込み操作の宛先位置のトラックIDは、たとえば、リスト項目12bによって示す通り、リスト10から除去することができる。図1の例では、リスト項目12b、12d、12e、および12nは、トラックIDを保管するために使用可能であるものとして示されている。この場合も、リスト項目12a、12b、・・・12nのうちの1つからのトラックIDの除去後、リスト10の内容について論理的冗長検査を実行することができ、その検査結果は論理的冗長検査項目14に保管することができる。

20

**【0008】**

リスト10の使用可能リスト項目の迅速な識別を容易にするために、リスト10の使用可能な項目のリストが維持される場合が多い。図1の例では、たとえば、項目12b、12d、12eなどの使用可能リスト項目は、それぞれ、リスト項目12a、12b、・・・12nのうちの次の使用可能リスト項目の位置を識別するポインタを収容している。このようなポインタは、たとえば、アドレス・オフセットの形を取ることができる。しかし、たとえば、項目12nなど、リスト10の最後の使用可能な項目は、その最後の項目の後に次の使用可能な項目がまったくないので、ポインタを収容していない可能性がある。書込み操作の開始の前の初期設定時には、リスト10のリスト項目12a、12b、・・・12nのそれぞれが使用可能である可能性がある。このため、リスト10が初期設定されると、リスト10は、最後の項目12nを除くリスト項目12a、12b、・・・のそれぞれのためのポインタを収容している可能性がある。

30

**【0009】**

動作に際しては、書込み操作中という状態がその完了前に中断された場合、書込み操作中の場合、即ち未完の書込み操作のトラックIDのリスト10をトラバースすることができる。このように、書込み操作中の宛先位置のトラックIDは、不完全なまたは破壊されたデータを収容している可能性があり、識別することができる。

40

**【発明の開示】****【発明が解決しようとする課題】****【0010】**

書込み操作中の場合の宛先位置IDのリストが、アレイの次の使用可能な項目を識別するアレイ・ポインタを有するアレイ内に維持される、書込み処理を管理するための方法、システム、およびプログラムが提供される。

**【課題を解決するための手段】**

50

## 【 0 0 1 1 】

本発明によれば、書込み操作の宛先の位置IDを取得するステップと、アレイの使用可能な項目に前記位置IDを保管するステップであって、前記項目が第1の値を有するアレイ・ポインタによって識別されるステップと、他の使用可能なアレイ項目を識別するために前記アレイ・ポインタの前記第1の値を第2の値に変更するステップとを有する方法が提供される。

## 【 0 0 1 2 】

一実施形態では、そのアレイは可変サイズのスタックを含む。書込み操作中の宛先位置IDを追加すると、スタックのサイズが増加する。完了した書込み操作の宛先位置IDを除去すると、スタックのサイズが低減する。スタック・インデックスは、書込み操作の宛先位置IDが追加されると増分することができ、書込み操作の宛先位置IDがスタックから除去されると減分することができる。

10

## 【 0 0 1 3 】

次に図面を参照するが、図面全体を通して同様の参照番号は対応する部分を表している。

## 【 発明を実施するための最良の形態 】

## 【 0 0 1 4 】

以下の説明では、本明細書の一部を形成し、いくつかの実施形態を図示する添付図面を参照する。他の諸実施形態を使用することができ、この説明の範囲を逸脱せずに構造上および操作上の変更を行うことができることは言うまでもないことである。

20

## 【 0 0 1 5 】

図2は、図示した諸実施形態の諸態様が実現されるコンピューティング・アーキテクチャを示している。1つまたは複数のホスト102a、102b、・・・102nは、記憶コントローラ106を介して、DASDまたは当技術分野で知られている任意のその他の記憶システムなどの記憶システム104とデータ通信状態にある。ホスト102は、サーバ、メインフレーム、ワークステーション、パーソナル・コンピュータ、ハンドヘルド・コンピュータ、ラップトップ、電話装置、ネットワーク・アプライアンスなど、当技術分野で知られている任意のコンピューティング・デバイスにすることができる。記憶コントローラ106とホスト・システム（複数も可）102はネットワーク108を介して通信するが、このネットワークはストレージ・エリア・ネットワーク（Storage Area Network : SAN）、ローカル・エリア・ネットワーク（LAN）、イントラネット、インターネット、広域ネットワーク（WAN）などを含むことができる。記憶システム104は、ハード・ディスク・ドライブ、テープ・カートリッジ・ライブラリ、光ディスク、または当技術分野で知られている任意の適切な不揮発性記憶媒体で構成することができる。記憶システム104は、JBOD（Just a Bunch of Disks、単なるディスクの束）、DASD、RAID（Redundant Array of Independent Disks）アレイ、仮想化装置（virtualization device）などの記憶装置のアレイとして配置することができる。記憶コントローラ106は、IBMエンタープライズ・ストレージ・サーバ（Enterprise Storage Server : ESS、IBMは登録商標）または当技術分野で知られている任意のその他の記憶コントローラなど、当技術分野で知られている任意の記憶コントローラまたはサーバを含むことができる。特定の実現例では、記憶コントローラ104内の記憶空間は、複数の論理装置（LD）110a、110b、・・・110nとして構成される。

30

40

## 【 0 0 1 6 】

記憶コントローラ106は、可用性改善のための冗長性を提供するために、複数のハードウェア・コンポーネントからなる2つの個別のクラスタ120a、120bを含む。図示した実施形態はクラスタを有する記憶コントローラに関して記載されているが、本発明は、1つまたは複数のプロセッサを使用する様々なシステムに適用可能であることは認識されている。

## 【 0 0 1 7 】

図示した実施形態の各クラスタ120a、120bは、個別の電力境界上で維持するこ

50

とができ、プロセッサ複合体 1 2 2 a、1 2 2 b と、キャッシュ 1 2 4 a、1 2 4 b と、不揮発性記憶装置 ( N V S ) 1 2 6 a、1 2 6 b とを含む。N V S 1 2 6 a、1 2 6 b は、バッテリー・バックアップ付き R A M、あるいはキャッシュにデータをバックアップするために使用される任意のその他のタイプの不揮発性または揮発性バックアップ・キャッシュを含むことができる。ホスト 1 0 2 a、1 0 2 b、・・・1 0 2 n は、ターゲット論理装置 ( L D ) 1 1 0 a、1 1 0 b、・・・1 1 0 n が割り当てられているクラスタ 1 2 0 a、1 2 0 b に対し、書込みデータを含む、ターゲット論理装置 ( L D ) 1 1 0 a、1 1 0 b、・・・1 1 0 n に向けられたアプリケーション I / O 要求を提示することができる。一方のクラスタ 1 2 0 a、1 2 0 b 内の N V S 1 2 6 a、1 2 6 b は、もう一方のクラスタ 1 2 0 b、1 2 0 a 内のキャッシュ 1 2 4 b、1 2 4 a 内の書込みデータをバックアップするために使用され、たとえば、N V S 1 2 6 a はキャッシュ 1 2 4 b 内の書込みデータをバックアップする。

10

#### 【 0 0 1 8 】

図 3 は、書込み操作中の場合の宛先位置のリストを維持することができる論理構造 3 0 0 の一例を示している。この構造 3 0 0 は、複数項目 3 0 4 a、3 0 4 b、・・・3 0 4 n のアレイ 3 0 2 を含み、その項目のそれぞれは書込み操作中の宛先位置を保管することができる。図示した実施形態では、書込み操作の宛先位置はトラック I D で識別される。シリンダ、セクタ、クラスタ、アドレスなどを含む、その他の宛先位置 I D を使用できることは認識されている。

#### 【 0 0 1 9 】

記載した諸実施形態の一態様によれば、この構造 3 0 0 は、書込み操作中の宛先位置を保管することができるアレイ 3 0 4 の次の使用可能な項目を識別するアレイ・ポインタ 3 0 6 をさらに含む。図示した実施形態では、複数項目 3 0 4 a、3 0 4 b、・・・3 0 4 n のアレイ全体の一部分 3 0 4 は、複数の隣接リスト項目 3 0 4 a、3 0 4 b、・・・3 0 4 i のスタックを提供し、そこではトラック I D などの宛先位置が各スタック項目 3 0 4 a、3 0 4 b、・・・3 0 4 i に保管される。アレイ・ポインタ 3 0 6 は、アレイ 3 0 2 の次の使用可能な項目を指し示し、したがって、保管されたトラック I D のスタック 3 0 4 の末尾の後の最初の項目を示すスタック・インデックスを有する。図 3 の例では、スタック 3 0 4 の最後の項目は項目 3 0 4 i であり、最初の使用可能な項目は複数項目 3 0 4 a、3 0 4 b、・・・3 0 4 n のアレイのうちの項目 3 0 4 j である。

20

30

#### 【 0 0 2 0 】

このような論理構造の一応用例では、書込み操作中の宛先位置を決定するための複数のアレイ項目のトラバーサルは、空のまたは無効な項目 3 0 4 j、・・・3 0 4 n を含む複数項目 3 0 4 a、3 0 4 b、・・・3 0 4 n のアレイ 3 0 2 全体ではなく、実際にトラック I D を収容している複数項目 3 0 4 a、3 0 4 b、・・・3 0 4 i のスタックに限定することができる。このような論理構造 3 0 0 の他の応用例では、アレイ 3 0 2 に対する変更後の複数のアレイ項目の論理的冗長検査は、空のまたは無効な項目 3 0 4 j、・・・3 0 4 n を含む複数項目 3 0 4 a、3 0 4 b、・・・3 0 4 n のアレイ全体ではなく、実際に有効なトラック I D を収容している複数項目 3 0 4 a、3 0 4 b、・・・3 0 4 i のスタックに限定することができる。この論理的冗長検査は項目 3 1 0 に保管することができる。スタックと、複数項目のアレイと、使用可能な項目へのアレイ・ポインタがその他の形式で配置できることは認識されている。

40

#### 【 0 0 2 1 】

図 5 は、書込み操作中の場合、宛先位置のリストを維持する、図 3 の構造 3 0 0 などの論理構造を初期設定する際の記憶コントローラ 1 0 6 の動作を示している。図 5 の実施形態では、論理構造のためのアレイ・ポインタが初期設定される ( ブロック 4 0 2 )。図示した実施形態では、図 4 に示す通り、スタック 3 0 2 をゼロに初期設定することにより、これを容易に達成することができる。この例では、スタック・インデックス 3 0 6 は、空である最初のアレイ項目 3 0 4 a を指し示すように、リセットされる。したがって、論理構造 3 0 0 が図 4 に示す通りに初期設定されると、スタック 3 0 4 はいかなる項目も持た

50

なくなる。

【 0 0 2 2 】

加えて、論理的冗長検査を初期設定することができる（図5のブロック404）。この場合も、図示した実施形態では、検査310の内容をゼロに設定することにより、図4の論理的冗長検査310を容易に初期設定することができる。

【 0 0 2 3 】

図6は、データが記憶装置104などの記憶装置に書き込まれる書込み操作中の宛先の位置IDを、図3の論理構造300などの論理構造に保管する際の記憶コントローラ106の動作を示している。図6の実施形態では、記憶コントローラ106は、書込み操作から宛先の位置IDを取得する（ブロック502）。図示した実施形態では、書込み操作のための宛先位置IDは、図7に示すデータ構造602a、602b、・・・602nのうちの1つなどのデータ構造へのポインタの形で記憶コントローラ106に提供される。各書込み操作中は、関連データ構造602a、602b、・・・602nをキャッシュ・メモリ124a、124b内に維持する。各データ構造はいくつかのフィールドを有し、そのそれぞれは書込み操作に使用される様々なパラメータの1つを保管する。このようなパラメータの1つは、書込み操作によって記憶装置104に書き込まれるデータの宛先の位置を識別するトラックIDである。トラックIDは、関連データ構造のフィールド604に保管される。したがって、データ構造602a、602b、・・・602nのうちの1つへのポインタの受領に回答して、記憶コントローラ106は、書込み操作で書き込まれるデータの宛先のトラックIDを、示されたデータ構造から取得する（ブロック502）。書込み操作の宛先の位置IDが、書込み操作によって直接、宛先位置IDが提供されることを含む様々な方法で取得可能であることは認識されている。

【 0 0 2 4 】

記憶コントローラ106は、アレイ・ポインタによって識別される使用可能アレイ項目に取得した位置IDを保管する（ブロック504）。図3の実施形態では、アレイ・ポインタは、アレイ302のうちの次の使用可能な項目を指し示すスタック・インデックス306によって実現される。図3の例では、スタック項目304a～304iのそれぞれは、すでに関連書込み操作のためのトラックIDで充填されている。このため、スタック・インデックス306は、次の使用可能アレイ項目304jを指し示す。したがって、記憶コントローラ106は、スタック・インデックス306によって識別された使用可能アレイ項目304jに取得したトラックIDを保管する（ブロック504）。書込み操作の宛先位置IDで充填されると、アレイ項目304jはスタック304の一部になる。

【 0 0 2 5 】

また、アレイ・ポインタの値は、位置IDが当初、そこから取得されたデータ構造に保管される（ブロック506）。図3の例では、スタック・インデックス306の値は、アレイ項目304jを指し示すオフセットである。このオフセット値は、アレイ項目304jに保管されたトラックIDがそこから取得されたデータ構造602a、602b、・・・602nのフィールド606に保管される（ブロック506）。

【 0 0 2 6 】

さらに、アレイ・ポインタの内容は、使用可能なアレイ項目を識別するために変更される（ブロック508）。図3の例では、スタック・インデックス306は、アレイ項目304jがトラックIDで充填された後にアレイ項目304kになる次の使用可能アレイ項目を指し示すように増分される。アレイ項目304kは、アレイ302内のアレイ項目304jに直接隣接している。

【 0 0 2 7 】

また、論理構造300の内容の論理的冗長検査は更新することができる（ブロック510）。図示した実施形態では、スタック・インデックス306の内容とともに、スタック304の内容について論理的冗長検査を実行することができ、これは図3の例では（アレイ項目304jがトラックIDで充填された後に）アレイ項目304a～304jの内容になる。更新された論理的冗長検査は論理的冗長検査310に保管することができる（ブ

10

20

30

40

50

ロック 510)。したがって、応用例によっては、アレイ項目 304k ~ 304n などの空のアレイ項目は、トラバースする必要がなく、事実上、論理的冗長検査中に無視することができる。

#### 【0028】

図 8 は、完了した書込み操作のための宛先位置 ID、すなわち、その書込み操作のデータのすべてが記憶装置 104 内の適切な宛先位置に正常に書き込まれた後の書込み操作のための宛先位置 ID を、図 3 の論理構造 300 などの論理構造から除去する際の記憶コントローラ 106 の動作を示している。書込み操作の完了の通知を受領すると（ブロック 702）、記憶コントローラ 106 は、スタック 304 内で完了した書込み操作の位置 ID を突き止め、スタック 304 からその位置 ID を除去する（ブロック 704）。 10

#### 【0029】

スタック 304 から位置 ID を除去すると、ブランクのまたは空の項目を残すことができる。一実施形態では、スタック 304 のサイズを低減する（ブロック 706）ために、スタック 304 内の空のまたはブランクのアレイ項目を除去することができる。このようなサイズ低減は、論理的冗長検査更新中またはエラー回復中にスタックのトラバースを容易にすることができる。

#### 【0030】

スタック・サイズの低減後、論理構造 300 の内容の論理的冗長検査を更新することができる（ブロック 710）。この場合も、スタック・インデックス 306 の内容とともに、スタック 304 の内容について論理的冗長検査を実行することができ、これは図 3 の例では（アレイ項目 304j が次の使用可能な項目であるときに）アレイ項目 304a ~ 304j の内容になる。更新された論理的冗長検査は論理的冗長検査 310 に保管することができる（ブロック 710）。したがって、応用例によっては、書込み操作中のための位置 ID を収容しておらず、スタック 304 の一部ではないアレイ項目 304j ~ 304n などのアレイ項目は、トラバースする必要がなく、事実上、論理的冗長検査中に無視することができる。 20

#### 【0031】

図示した実施形態では、記憶コントローラは、完了した書込み操作のデータ構造 602a、602b、・・・602n のうちの特定のデータ構造へのポインタを受領する（図 9 のブロック 802）ことにより、特定の書込み操作の完了が通知される。次に、記憶コントローラ 106 は、特定のデータ構造のフィールド 606 に保管されたスタック・インデックスの値を取得し（ブロック 804）、それにより、完了した書込み操作のためのトラック ID が保管される特定のスタック項目の位置を特定することができる。次に、完了した書込み操作のためのデータ構造から取得したスタック・インデックスの値によって識別されたスタック項目に保管されたトラック ID は、後述の通り、スタック 304 から除去することができる。 30

#### 【0032】

書込み操作の完了の通知が多種多様な方法で取得できることは認識されている。さらに、宛先位置 ID を収容しているアレイ項目の位置が様々な方法で決定できることも認識されている。たとえば、書込み操作は、完了した書込み操作のトラック ID を直接提供する場合もあれば、そのトラック ID 用のアレイ項目位置を記憶コントローラ 106 に直接提供する場合もある。 40

#### 【0033】

図示した実施形態では、記憶コントローラ 106 は、最後の項目をスタック 304 から除去する（ブロック 806）ことにより、スタック 304 のサイズを低減する。図示した例では、これは、スタック 304 の前のアレイ項目を指し示すようにスタック・インデックス 306 を減分することによって達成される。図 3 の例では、スタック・インデックス 306 を減分すると、スタック・インデックス 306 はアレイ項目 304j ではなくアレイ項目 304i を指し示すようになる。したがって、スタック 304 の最後のアレイ項目は前のアレイ項目 304h になる。アレイ項目 304i はもはやスタック 304 の一部で 50

はなくなるが、その代わりに、スタック・インデックス306が示す通り、次の使用可能なアレイ項目として示されることになる。

【0034】

また、除去した最後の項目が、スタック304から除去すべき完了した書込み操作の宛先位置IDを収容しているかどうかの判定(ブロック808)が行われる。収容している場合、ブロック810に示す通り、スタック304から最後のアレイ項目を除去すると、完了した書込み操作の宛先位置IDが事実上、スタック304からすでに除去されている。図示した実施形態では、除去した最後の項目が完了した書込み操作の宛先位置IDを収容しているかどうかの判定(ブロック808)は、完了した書込み操作によって供給されたポインタによって識別されたデータ構造のフィールド606から取得した(ブロック804)スタック・インデックスの値を、減分したスタック・インデックスの値と比較することによって、行うことができる。一致している場合、すなわち、2つのインデックス値が同じである場合、除去した最後の項目が、その最後の項目をスタック304から除去することによってスタック304から除去される完了した書込み操作の宛先位置IDを収容していると判定される。

10

【0035】

図3に示した例では、スタック304の最後の項目は、上記の通り、アレイ項目304iであるものとして示されている。スタック・インデックス306を減分すると、スタック・インデックス306は、アレイ項目304jの代わりに、新しい位置IDを書き込むために使用可能な次の項目としてアレイ項目304iを指し示すことになる。このため、スタック304の最後の項目は前のアレイ項目304hになる。完了した書込み操作によって供給されたポインタによって識別されたデータ構造のフィールド606から取得した(ブロック804)スタック・インデックスの値が同じアレイ項目304iを指し示す場合、正しい位置IDはすでにスタック304から除去されていることになる。換言すれば、一致している場合、すなわち、2つのインデックス値が同じである場合、除去した最後の項目が、その最後の項目をスタック304から除去することによってスタック304から除去される完了した書込み操作の宛先位置IDを収容していると判定される。

20

【0036】

これに対して、除去した最後の項目が完了した書込み操作の宛先位置IDを収容していないと判定された場合(ブロック808)、除去したアレイ項目は書込み操作中のための有効な位置IDを収容している。したがって、スタック304から除去されたアレイ項目からの書込み操作の宛先位置IDは、完了した書込み操作の位置IDを収容していたスタック項目に移動する(ブロック812)。除去したアレイ項目から、完了した書込み操作の位置IDを収容しているアレイ項目に、有効な位置IDを移動することにより、完了した書込み操作の位置IDは上書きされ、事実上、スタック304から除去される。

30

【0037】

上記の通り、図示した実施形態では、除去した最後の項目が完了した書込み操作の宛先位置IDを収容しているかどうかの判定(ブロック808)は、完了した書込み操作によって供給されたポインタによって識別されたデータ構造のフィールド606から取得した(ブロック804)スタック・インデックスの値を、減分したスタック・インデックスの値と比較することによって、行うことができる。一致していない場合、すなわち、2つのインデックス値が同じではない場合、除去した最後の項目が、依然として完了していない書込み操作の有効な宛先位置IDを収容しており、したがって、その最後の項目をスタック304から除去することによってスタック304から除去してはならないと判定される。したがって、スタック304から除去されたアレイ項目からの書込み操作の宛先位置IDは、完了した書込み操作によって供給されたポインタによって識別されたデータ構造のフィールド606から取得した(ブロック804)スタック・インデックスの値によって識別される通り、完了した書込み操作の位置IDを収容していたスタック項目に移動する(ブロック812)。

40

【0038】

50

書込み操作の宛先位置IDが他のスタック項目に移動するので、その宛先位置IDの書込み操作中のためのデータ構造は、宛先位置IDの移動先である異なるスタック項目の位置を示すように更新される(ブロック814)。図示した実施形態では、スタック304から除去されたアレイ項目からの書込み操作の宛先位置IDは、完了した書込み操作によって供給されたポインタによって識別されたデータ構造のフィールド606から取得した(ブロック804)スタック・インデックスの値によって識別される通り、完了した書込み操作の位置IDを収容していたスタック項目に移動する(ブロック812)。このため、取得したスタック・インデックス値は、移動した宛先位置IDに関する書込み操作中のデータ構造のフィールド606に保管される。

【0039】

このように、図示した実施形態では、書込み操作中のための位置IDがスタック304に追加されるたびに、アレイ項目がスタック304に追加される。逆に、書込み操作が完了し、完了した書込み操作の位置IDがスタックから除去されるたびに、アレイ項目がスタック304から除去される。その結果、一応用例では、論理的冗長検査を効率よく実行することができる。加えて、他の応用例では、エラー回復操作中に、アレイ302のトラバーサルをスタック304のアレイ項目に限定することができる。スタックの一部ではないアレイ項目は、書込み操作中のための宛先位置IDを収容しておらず、一応用例では無視することができる。スタックの一部ではないアレイ項目が無視されるので、スタック外のアレイ項目内にある完了した書込み操作のための宛先位置は消去する必要がない。

【0040】

追加の実現例の詳細

書込み処理管理のための上述の技法は、ソフトウェア、ファームウェア、ハードウェア、またはそれらの任意の組合せを生産するための標準的なプログラミングおよび/またはエンジニアリング技法を使用して、方法、装置、または製品として実現することができる。本明細書で使用する「製品」という用語は、ハードウェア・ロジック(たとえば、集積回路チップ、プログラマブル・ゲート・アレイ(PGA)、特定用途向け集積回路(ASIC)など)または磁気記憶媒体(たとえば、ハード・ディスク・ドライブ、フレキシブル・ディスク、テープなど)、光学記憶装置(CD-ROM、光ディスクなど)、揮発性および不揮発性メモリ・デバイス(たとえば、EEPROM、ROM、PROM、RAM、DRAM、SRAM、ファームウェア、プログラマブル・ロジックなど)などのコンピュータ可読媒体に実現されたコードまたはロジックを指す。コンピュータ可読媒体内のコードは、プロセッサ複合体によってアクセスされ、実行される。好ましい諸実施形態が実現されるコードは、伝送媒体により、またはネットワークを介したファイル・サーバから、さらにアクセス可能である。このような場合、そのコードが実現される製品は、ネットワーク伝送回線、ワイヤレス伝送媒体、空間を伝搬する信号、電波、赤外線信号などの伝送媒体を有することができる。したがって、「製品」は、そのコードが実施される媒体を有することができる。さらに、「製品」は、そのコードが実施され、処理され、実行される、ハードウェア・コンポーネントとソフトウェア・コンポーネントの組合せを有することができる。当然のことながら、当業者であれば、本発明の範囲を逸脱せずに、この構成に対して多くの変更が可能であること、ならびに製品が当技術分野で知られている任意の情報担持媒体を有することができることを認識するであろう。

【0041】

上述の実現例では、特定の動作は、プロセッサ複合体122aおよび122bによって実行されるものとして記載されている。代替実現例では、プロセッサ複合体によって実行されるものとして記載されている特定の動作は、様々なコンピュータ、プロセッサ、およびコントローラによって実行することができる。

【0042】

上述の実現例では、特定の動作は、インデックスまたはカウンタを増分および減分するものとして記載されている。カウンタの「カウント」は正である場合もあれば負である場合もあり、カウントを「増分」または「減分」とすると、その特定の応用例に応じて、実際

10

20

30

40

50

のカウン트의負または正の程度がより大きくなる場合もあれば小さくなる場合もあることは認識されている。

【0043】

図5、図6、図8、および図9の図示したロジックは、特定の順序で発生する特定のイベントを示している。代替実現例では、特定の動作は、異なる順序で実行するか、変更するか、または除去することができる。その上、いくつかの動作を上述のロジックに追加することができ、このような動作は依然として上述の実現例に適合するものである。さらに、本明細書に記載した動作は順次発生する場合もあれば、特定の動作が並行して処理される場合もある。さらに、動作は単一処理装置によってまたは複数の分散処理装置によって実行することができる。

10

【0044】

変数Nは、上述の諸要素のうちのいくつかに関する任意の整数変数を示すために使用されており、異なる事例で使用する場合は同じ整数値を示す場合もあれば、異なる整数値を示す場合もある。

【0045】

図10は、図2に示すホストおよび記憶コントローラ・クラスタなどのネットワーク・コンポーネントからなるコンピュータ・アーキテクチャ900の一実現例を示している。アーキテクチャ900は、プロセッサ902（たとえば、マイクロプロセッサ）と、メモリ904（たとえば、揮発性メモリ・デバイス）と、記憶装置906（たとえば、磁気ディスク・ドライブ、光ディスク・ドライブ、テープ・ドライブなどの不揮発性記憶装置）とを含むことができる。記憶装置906は、内部記憶装置あるいは付加記憶装置またはネットワーク・アクセス可能記憶装置を含むことができる。記憶装置906内のプログラムは、当技術分野で知られている方法で、メモリ904にロードされ、プロセッサ902によって実行される。このアーキテクチャは、ネットワークとの通信を可能にするためにネットワーク・カード908をさらに含む。入力装置910は、プロセッサ902にユーザ入力を提供するために使用され、キーボード、マウス、ペンスタイラス、マイクロホン、タッチ検知表示画面、あるいは当技術分野で知られている任意のその他の起動または入力メカニズムを含むことができる。出力装置912は、プロセッサ902から、または表示モニタ、プリンタ、記憶装置などのその他のコンポーネントから伝送される情報を表現することができる。

20

30

【0046】

様々な実現例に関する上記の記述は、例示および説明のために提示されたものである。網羅的にするか、または開示された正確な形式に範囲を限定するためのものではない。上記の教示を考慮すると、多くの変更および変形が可能である。

【図面の簡単な説明】

【0047】

【図1】書込み操作中即ち未完の書込み操作を追跡するための従来技術の論理構造の概略図である。

【図2】図示した諸実施形態の諸態様を実現可能なコンピューティング環境の一例を示す図である。

40

【図3】図示した諸実施形態の一態様により未完の書込み操作を追跡するための論理構造の概略図である。

【図4】図示した諸実施形態の一態様により未完の書込み操作を追跡するための論理構造の概略図である。

【図5】図3の構造などの論理構造を初期設定する際の記憶コントローラの動作を示す図である。

【図6】データが記憶装置に書き込まれている未完の書込み操作の宛先の位置IDを図3の論理構造などの論理構造に保管する際の記憶コントローラの動作を示す図である。

【図7】書込み操作で使用される様々なパラメータを保管するデータ構造の一例を示す図である。

50

【図 8】完了した書込み操作のための宛先位置 ID を図 3 の論理構造などの論理構造から除去する際の記憶コントローラの動作を示す図である。

【図 9】完了した書込み操作のための宛先位置 ID を論理構造から除去する際の記憶コントローラの動作のより詳細な一例を示す図である。

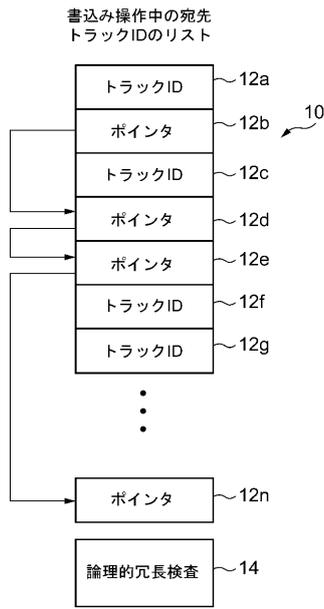
【図 10】ホスト、記憶コントローラ、クラスタ、および任意のその他のコンピューティング・デバイスなど、ネットワーク環境内のコンピューティング・コンポーネントのアーキテクチャを示す図である。

【符号の説明】

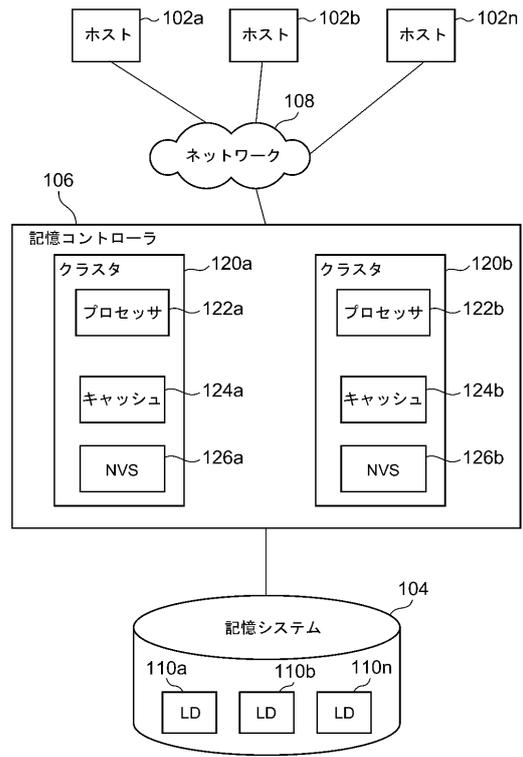
【 0 0 4 8 】

1 0 2 a	ホスト	10
1 0 2 b	ホスト	
1 0 2 n	ホスト	
1 0 4	記憶システム	
1 0 6	記憶コントローラ	
1 0 8	ネットワーク	
1 1 0 a	論理装置 ( L D )	
1 1 0 b	論理装置 ( L D )	
1 1 0 n	論理装置 ( L D )	
1 2 0 a	クラスタ	
1 2 0 b	クラスタ	20
1 2 2 a	プロセッサ	
1 2 2 b	プロセッサ	
1 2 4 a	キャッシュ	
1 2 4 b	キャッシュ	
1 2 6 a	不揮発性記憶装置 ( N V S )	
1 2 6 b	不揮発性記憶装置 ( N V S )	

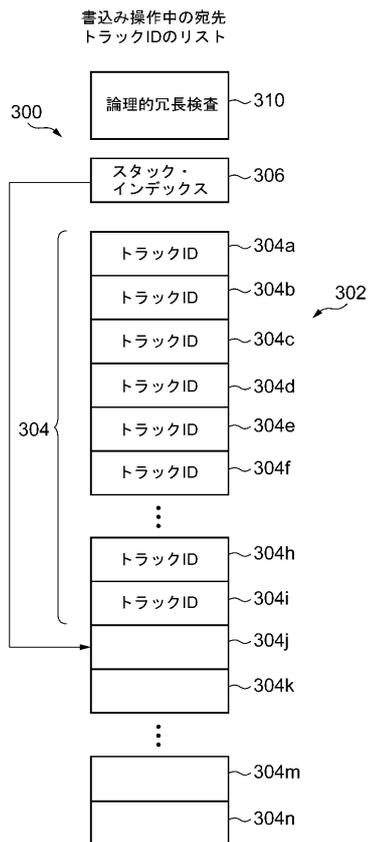
【図1】



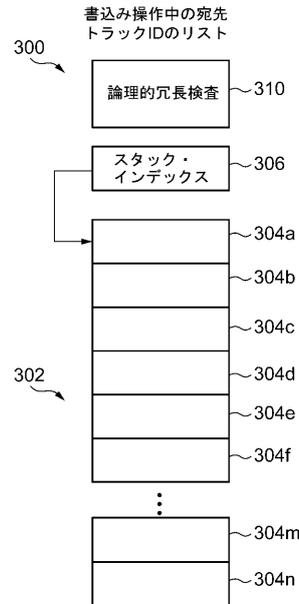
【図2】



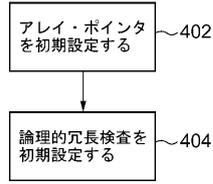
【図3】



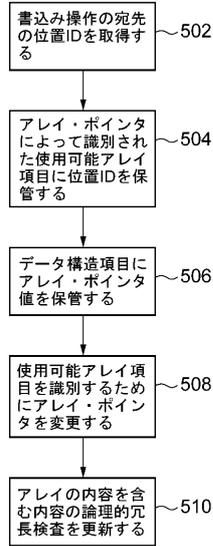
【図4】



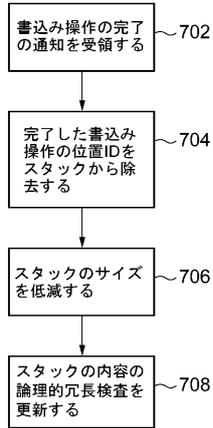
【図5】



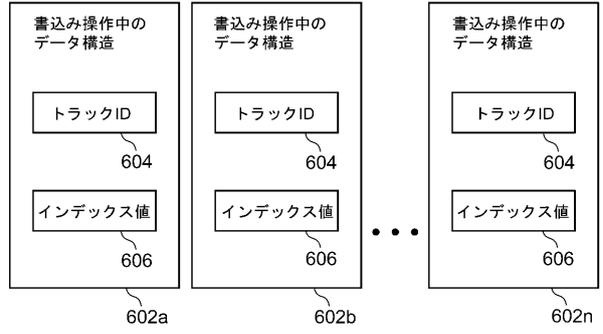
【図6】



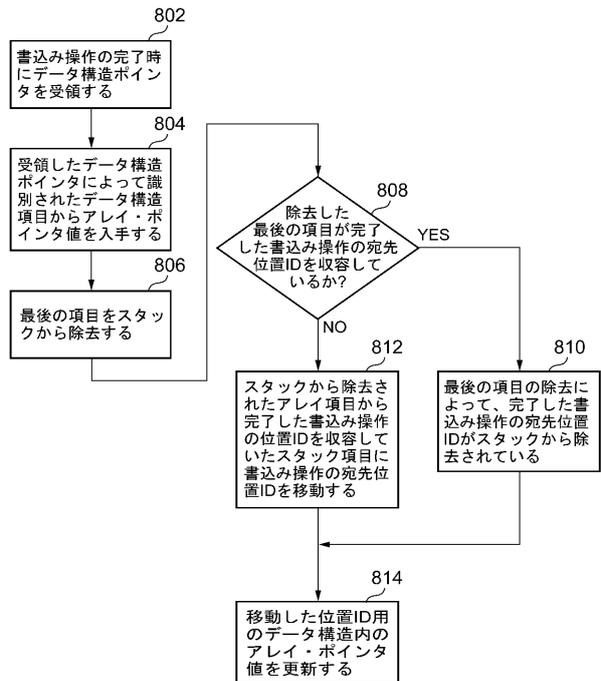
【図8】



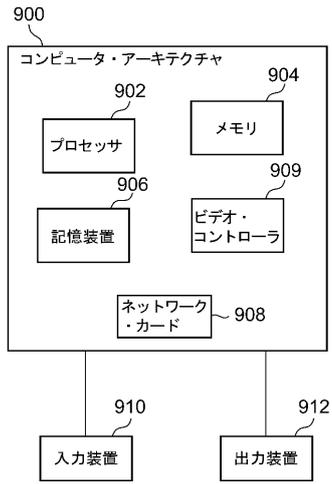
【図7】



【図9】



【図10】



## フロントページの続き

- (72)発明者 ジョン・ジェイ・ウォルフガング  
アメリカ合衆国 2 7 1 0 6 ノースカロライナ州ウィンストンセーレム サガモア・レーン 4 1  
1 3
- (72)発明者 フィリップ・マシュー・ドウトマス  
アメリカ合衆国 8 5 7 4 1 アリゾナ州トゥーソン ノース・スターシャイン・ドライブ 6 7 0  
0
- (72)発明者 ケネス・フェアクロー・デイ・ザ・サード  
アメリカ合衆国 8 5 7 4 8 アリゾナ州トゥーソン ノース・レイジー・ジェイ・ウェイ 7 3 0
- (72)発明者 ケネス・ウェイン・ボイド  
アメリカ合衆国 8 5 7 4 8 アリゾナ州トゥーソン ノース・カミーノ・コードン 8 5 5

## 合議体

審判長 江口 能弘  
審判官 安久 司郎  
審判官 安島 智也

- (56)参考文献 特開 2 0 0 2 - 2 7 8 7 0 7 ( J P , A )  
特開昭 6 2 - 2 8 4 4 3 7 ( J P , A )  
特開昭 6 2 - 8 9 2 7 3 ( J P , A )  
特開平 1 1 - 1 5 4 0 5 8 ( J P , A )  
特開平 1 1 - 1 1 0 2 6 6 ( J P , A )  
特開平 5 - 1 3 4 9 1 9 ( J P , A )  
再公表特許第 2 0 0 3 / 0 9 6 1 9 2 ( J P , A 1 )

## (58)調査した分野(Int.Cl. , DB名)

G06F 3/06  
G06F 12/08