(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2006-331076
(P2006-331076A)
(43)公開日　平成18年12月7日(2006.12.7)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 3/06 (2006.01)	G06F 3/06 301Z	5B065

審査請求 未請求　請求項の数 10　ＯＬ　（全 12 頁）

(21)出願番号　特願2005-153697 (P2005-153697)
(22)出願日　平成17年5月26日 (2005.5.26)

(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　110000062
特許業務法人第一国際特許事務所
(72)発明者　植田　良一
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

Ｆターム(参考)　5B065 BA01　CA30　CC03　ZA01　ZA16

(54)【発明の名称】データ記憶システム及び記憶方法

(57)【要約】
【課題】　従来技術ではディスク装置内のひとつの区画をひとつのＲＡＩＤレベルを構成する区画グループに固定的に割り当てるため、区画分割を行う際に想定した利用状況が実際の利用状況と合致しない場合、特定の区画がほぼ満杯であるにもかかわらず、他の区画はあまり利用されていない状況が発生し、ディスク全体の利用効率が悪くなる問題がある。
【解決手段】　ファイル管理プログラム（通常「ファイルシステム」と呼ばれる）が管理するファイル管理テーブルに、ファイル毎に、ファイルの記憶方法と、ディスク装置の識別子とそのディスク装置の内部での記憶位置情報の組合せの複数個と、を記憶することで、ひとつの区画に異なるＲＡＩＤレベルが混在した記憶システム及び記憶方法を実現する。
【選択図】　図８

ファイル管理テーブル

#	Filename	Size	Method	Addr0	Addr1	Addr2	Addr3
M	MngTable	50KB	R0+1	D0_0	D1_0	D2_0	D3_0
F1	nothing.txt	70KB	-	D3_5	-	-	-
F2	important.doc	60KB	R1(2)	D0_1	D2_4	-	-
F3	balance.xls	158KB	R5(3+1)	D0_4	D1_2	D2_6	D3_3
F4	speedy.jpg	110KB	R0(4)	D3_2	D2_3	D1_3	D0_7

811

【特許請求の範囲】
【請求項１】
ファイル毎に、ファイルの記憶方法と、ディスク装置の識別子とそのディスク装置の内部での記憶位置情報の組合せの複数個と、を記憶可能なファイル管理テーブルを有することを特徴とするデータ記憶システム。
【請求項２】
請求項１に記載のデータ記憶システムにおいて、
個々のディスク装置内の空き容量の記憶／更新処理部を有し、書き込みを行う際に空き容量の大きいディスク装置から優先的に選択することを特徴とするデータ記憶システム。
【請求項３】
請求項１に記載のデータ記憶システムにおいて、
個々のディスク装置内の空き容量の記憶／更新処理部と、空き容量の少ないディスク装置を利用し、かつ、空き容量の多いディスク装置を利用していないファイルの検索処理部と、当該ファイルの移動処理部を有することを特徴とするデータ記憶システム。
【請求項４】
請求項１に記載のデータ記憶システムにおいて、
次にパリティブロックを書き込むべきディスク装置を特定する情報の記憶／更新処理部を有することを特徴とするデータ記憶システム。
【請求項５】
ファイル毎に、ファイルの記憶方法と、ディスク装置の識別子とそのディスク装置の内部での記憶位置情報の組合せの複数個と、ファイル管理テーブルに記憶することを特徴とするデータ記憶方法。
【請求項６】
請求項５に記載のデータ記憶方法において、
個々のディスク装置内の空き容量を記憶／更新するステップを有し、書き込みを行う際に空き容量の大きいディスク装置を優先的に選択することを特徴とするデータ記憶方法。
【請求項７】
請求項５に記載のデータ記憶方法において、
個々のディスク装置内の空き容量を記憶／更新するステップと、空き容量の少ないディスク装置を利用し、かつ、空き容量の多いディスク装置を利用していないファイルを探し出すステップと、当該ファイルを移動するステップとを有することを特徴とするデータ記憶方法。
【請求項８】
請求項５に記載のデータ記憶方法において、
次にパリティブロックを書き込むべきディスク装置を特定する情報を記憶／更新するステップを有することを特徴とするデータ記憶方法。
【請求項９】
ファイルの記憶方法を決定するステップと、前記記憶方法に従ってデータをブロックに分割するステップと、書き込み先ディスク装置及び書き込み場所を決定するステップと、ファイル管理テーブルを更新するステップを備えた書き込み処理手順を有するデータ記憶方法。
【請求項１０】
ファイル管理テーブルから管理情報を取得するステップと、読み込み先ディスク装置を決定するするステップと、データの読み込みを実施するステップと、全データブロックの読み込み終了後、データブロックを連結するステップを備えた読み込み手順を有するデータ記憶方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、複数のディスク装置を接続可能な記憶装置に情報を記憶する方法に関するも

のである。
【背景技術】
【０００２】
複数のディスク装置を利用して、情報を多重化し同じ情報を複数のディスクに記憶することで耐障害性および読み出し速度を高めたり、情報を分割して記憶することでアクセス速度を高めたりするＲＡＩＤ技術が知られている（例えば非特許文献１参照）。
【０００３】
ＲＡＩＤにはいくつかの種類があり、通常ＲＡＩＤレベルと呼ぶ番号で区別する。例えば、「ＲＡＩＤ１」は複数のディスク装置に同じデータを重複して書き込むことで、いずれかのディスク装置が障害により利用不可となっても別のディスクからデータを読み出し可能とする。これによりデータを失う可能性を低減している。さらに、全てのディスク装置に同時に読み出し指示を送り早く反応したディスク装置からデータを読み出すことで、読み出し性能の向上も可能とする。また、「ＲＡＩＤ５」は書き込むデータを特定の長さのデータブロックに分割後、いくつかのデータブロック毎にパリティブロック生成し、このデータブロックとそれに対応するパリティブロックを異なるディスク装置に記憶する。読み出し処理は複数のディスク装置に同時に指示を出し、データブロックを連結させることで完了するため高速に処理可能である。一台のディスク装置の障害によりひとつのデータブロックが読み出せなくなっても、パリティブロックから復元可能なので一台までの耐障害性も実現する。
このようなＲＡＩＤ技術が誕生してしばらくの間、ＲＡＩＤを構成するディスク装置は同じ容量のディスク装置を組み合わせることが想定されていた。容量の異なるディスク装置を組み合わせても、容量の最も小さいディスク装置の容量しか有効に利用できなかった。
【０００４】
この問題を解決する、すなわち、容量の異なるディスク装置を組み合わせた場合でも全ディスク装置の全領域を有効に利用する方法が特許文献１に記載されている。本文献では、容量の最も小さいディスク装置のサイズで記憶領域を分割し（分割後の個々の記憶領域を「区画」と呼ぶ）、それらを集めてひとつのＲＡＩＤを構成した後、残りの区画で別のＲＡＩＤを構成するという作業を繰り返し、この構成情報をテーブルで管理することで従来利用できなかった領域の有効利用を図っている。
【０００５】
また、特許文献２には、クライアントからの要求を受信して、各ネットワークストレージに転送し、各ネットワークストレージから受信した応答をを結合して、クライアントに送信することにより、専用のネットワークストレージ、集中管理型サーバ、分散ディレクトリを用いずに、汎用のネットワークストレージとネットワークストレージアクセス用プロトコルの処理によりネットワークストレージ仮想化を実現する技術が開示されている。
【０００６】
上記に代表されるディスク装置の利用方法に関する技術とは別に、ディスク装置に記憶される情報の扱い方に関する技術として「情報ライフサイクル管理」技術がある。情報ライフサイクル管理技術とは「その種別によって異なり、時間の経過と共に変化する個々の情報のビジネス上の価値に応じて、その記憶場所や記憶方法を変化させる管理手法」を指す。例えば、ある文書の作成段階では頻繁に参照や更新が行われるため高速にアクセスできる記憶方法で格納することが要求されるが、その後、作成作業が終了し、さらに時間が経過するとアクセス頻度が下がるため低速ではあるが安価な記憶方法で格納可能となる、といった状況が考えられる。
【特許文献１】特開平８－２４９１３２号公報
【特許文献２】特開２００４－５４７２１号公報
【非特許文献１】David A. Patterson, Garth A. Gibson and Randy H. Katz, ”A Case for Redundant Arrays of Inexpensive Disks (RAID)”（http：//www-2.cs.cmu.edu/～garth/RAIDpaper/Patterson88.pdf）, ACM SIGMOD Conference

1988: pp 109-116.

【発明の開示】

【発明が解決しようとする課題】

【０００７】

従来技術ではディスク装置内のひとつの区画をひとつのＲＡＩＤレベルを構成する区画グループに固定的に割り当てるため、記憶領域を分割して区画を作成する際に想定した利用状況が実際の利用状況と合致しない場合、特定の区画がほぼ満杯であるにもかかわらず、他の区画はあまり利用されていない状況が発生し、ディスク全体の利用効率が悪くなる問題がある。このような問題への対応として、区画サイズの再設定（拡大／縮小）を可能とする技術も存在するが、サイズ縮小に対応していなかったり、縮小前に未利用ブロックを区画の特定の位置（区画の末尾など）に集める処理が必要で、この処理に長い時間がかかるなど実運用時の制限が多く、利用しにくいものであった。

また、あるファイルの価値の変化に応じてＲＡＩＤレベルを変更する際にファイルの移動およびそれに伴う物理パスの変更を余儀なくされるため、論理パスから物理パスへの変換機構が必要となり、ファイル管理が複雑になるという問題があった。

【課題を解決するための手段】

【０００８】

このような問題を解決するために、本発明では、ファイル管理プログラム（通常「ファイルシステム」と呼ばれる）が管理するファイル管理テーブルに、ファイル毎に、ファイルの記憶方法と、ディスク装置の識別子とそのディスク装置の内部での記憶位置情報の組合せの複数個と、を記憶することで、ひとつの区画に異なるＲＡＩＤレベルが混在した記憶システム及び記憶方法を実現する。

【発明の効果】

【０００９】

本発明のデータ記憶システム及び記憶方法によると、区画サイズの変更なしにディスク装置の利用効率を上げることができるという利点がある。また別の効果として、情報の価値の変化に応じて記憶方法を変化させる際に、情報へのパスを変更する必要がないため、論理パスから物理パスへの変換が不要となり、情報管理が単純になるという利点がある。

【発明を実施するための最良の形態】

【００１０】

本発明は、ファイルシステムが管理するファイル管理テーブルにファイルの記憶方法および、記憶方法に応じたファイル記憶位置情報を持つものである。

【実施例１】

【００１１】

本発明は、図１に示すように、ネットワーク（１１１）に接続され、ＣＰＵ（１０１）、メモリ（１０２）、表示装置（１０３）、入力装置（１０４）、通信装置（１０５）、複数のディスク装置（１３１、１３２、１３３、１３４）を備える記憶装置（１０６）から構成される計算機上のファイル管理プログラム（１２１）により実現される。ファイル管理プログラム（１２１）はファイル管理テーブル（１４１）、ディスク管理情報（１４２）、記憶方法管理テーブル（１４３）を利用する。本実施例では4台のディスク装置を備える例を使って説明を行うが本発明は2台以上のディスク装置が備わっていれば適用可能である。ディスク装置の台数は実現できる記憶方法の種別に影響する。例えば、３つのデータブロックに対して１つのパリティブロックを生成するＲＡＩＤ５（３Ｄ＋１Ｐ）と呼ばれる記憶方法には少なくとも４台のディスク装置が必要となるため、３台のディスク装置しか備わっていない環境ではＲＡＩＤ５（３Ｄ＋１Ｐ）には対応できない。本発明ではディスク装置の台数以下で実現可能な全ての記憶方法を実現可能である。

【００１２】

図２にファイル管理テーブル（１４１）の例を示す。２０１には個々のファイルを識別するＩＤを格納する。例えば「M」は当該情報がユーザのファイルではなく、管理情報であることを意味する。管理情報にはファイル管理テーブル（１４１）、ディスク管理情報

（１４２）、記憶方法管理テーブル（１４３）が含まれる。「Ｆ１」「Ｆ２」などは当該情報がユーザのファイルであることを意味する。２０２にはファイル名を格納する。２０３にはファイルサイズを格納する。２０４には当該ファイルの記憶方法を格納する。例えば「Ｒ０＋１」は「ＲＡＩＤ０（２）＋１（２）」と呼ばれる、「ＲＡＩＤ０（２）」と「ＲＡＩＤ１（２）」を組み合わせた記憶方法を意味する。本記憶方法はデータを２分割して、並列に書き込みを行うと当時に、２台のディスク装置に同じ情報を書き込む方法である。
また「－」は多重化や分割などをせずに１台のディスク装置にのみデータを記憶する単純な記憶方法を意味する。２０５、２０６、２０７、２０８にはファイル記憶開始位置情報を格納する。例えば「Ｄ０＿０」はディスク装置「Ｄ０」のブロック番号「０」を意味する。

【００１３】
ファイル記憶位置情報は記憶方法により異なる意味付けが行われる。例えば記憶方法が「Ｒ０＋１」の場合、２０５には二重化の第一（正）のデータの、二分割した一方の側（先）の記憶開始位置を格納し、もう一方の側（後）の記憶開始位置を２０６に格納する。２０７には二重化の第二（副）のデータの、二分割した一方の側（先）の記憶開始位置を格納し、もう一方の側（後）の記憶開始位置を２０８に格納する。一方、記憶方法が「－」の場合、すなわち多重化も分割も行わない場合、２０５に記憶開始位置を格納するだけで、２０６、２０７、２０８は利用しない。また、記憶方法が「Ｒ５（３＋１）」すなわち前記「ＲＡＩＤ５（３Ｄ＋１Ｐ）」の場合、２０５には３分割したデータの第一のブロックの記憶開始位置を、２０６には３分割したデータの第二のブロックの記憶開始位置を、２０７には３分割したデータの第三のブロックの記憶開始位置を、２０８にはパリティブロックの記憶開始位置を、ぞれぞれ格納する。

【００１４】
また、ＲＡＩＤ１（２）と呼ばれる二重化を行う記憶方法では、２０５に正データの記憶開始位置を、２０６に副データの記憶開始位置を格納し、２０７、２０８は利用しない。また、ＲＡＩＤ０（４）と呼ばれる、４分割を行う記憶方法では、４分割したそれぞれのデータを先頭から順に２０５、２０６、２０７、２０８にそれぞれ格納する。本実施例ではディスク装置４台の例を説明しているため２０５から２０８までの４列を記憶位置情報に利用しているが、ｎ台の場合にはｎ列が必要となる。将来のディスク装置の増設に備えて実際の台数以上の列を持つファイル管理テーブルを当初から作成しておくことも可能である。

【００１５】
図３にディスク管理情報（１４２）の例を示す。ディスク管理情報はディスク空き容量管理テーブル（３０１）と、ディスク属性管理テーブル（３０２）からなる。ディスク空き容量管理テーブルはディスク装置ＩＤ（３１１）と空き容量（３１２）からなる。例えば、３２１はディスク装置「Ｄ０」の空き容量が８ブロックであることを意味する。ディスク属性管理テーブルは属性名（３３１）とその値（３３２）からなる。３４１は次にパリティブロックを書き込むディスク装置が「Ｄ３」であることを意味する。３４２はディスク装置のブロックサイズが３２ＫＢであることを意味する。

【００１６】
図２の状態に対応するディスク装置の状態の例を図４に示す。ディスク装置（１３１、１３２、１３３、１３４）の中のテーブルはディスクに記憶されたデータを意味する（１行が１ブロック分に相当）。４１１はディスク装置内の位置（ブロック番号）を意味する。ブロック番号は０から通し番号がふられている。４１２はデータが格納される領域を表す。ここではどのファイルのデータが記憶されているかを表現するためにファイルＩＤ「Ｆ１」とファイルのブロック番号「０」を記述した。パリティブロックは「Ｐ０」のように番号の前に「Ｐ」を付けて表現する。４１３は次のデータが格納されるブロック番号を意味する。「Ｅ」は最後のブロックであることを、空白は未使用であることを意味する。例えば、４２１はディスク装置「Ｄ３」のブロック番号「５」にファイル「Ｆ１」の０番

目のデータブロックが記憶されており、次のデータブロックが「６」であることを意味する。この例では、ディスク装置「Ｄ３」のブロック「６」にはファイル「Ｆ１」の１番目のデータブロックが記憶され、さらに、ブロック「８」にはファイル「Ｆ１」の最後のデータブロックが記憶されている。

【００１７】

図５に記憶方法管理テーブル（１４３）の例を示す。５０１には記憶方法を、５０２には多重度を、５０３にはデータ分割数を、５０４にはパリティ生成数を、それぞれ記憶する。例えば、５１１は記憶方法「Ｒ５（３＋１）」が、多重度１、分割数３、パリティ生成数１であることを意味する。

【００１８】

図６にデータを書き込む際の手順を示す。はじめにファイルの記憶方法を決定する（ステップ６０１）。これには、ユーザが対話的に指定する方法、上位アプリケーションプログラムが決定し本プログラムに通知する方法、ファイル種別／サイズなどの属性情報からあらかじめ設定したルールに則って決定する方法、初めての書き込み時はあらかじめ決めた記憶方法で書き込み、その後に上位アプリケーションから記憶方法を変更する方法など様々な方法が考えられる。本発明は記憶方法の決定方法に依存せず有効に働くため、ここでは前記の方法などにより記憶方法が決定されたと仮定して話を進める。

次にディスク装置に十分な空き領域があるかどうかを調べる（ステップ６１１）。同じサイズのファイルを書き込む場合でも記憶方法によって必要となるディスク容量が異なる可能性があることを考慮する必要がある。例えば、６０ＫＢのファイルを記憶方法ＲＡＩＤ０（２）で書き込む場合、２つのディスク装置にひとつずつ空きブロックがあれば書き込み可能であるが、同じファイルを記憶方法ＲＡＩＤ１（２）で書き込む場合、２つのディスク装置に２つずつの空きブロックが必要となる。十分な空き領域が存在しない場合、書き込みエラーとなる。

次にデータを一定の長さのブロックに分割する（ステップ６０２）。ブロックサイズは前記３４２の値（３２ＫＢ）を利用する。

次にパリティが必要な記憶方法かどうかを判定する（ステップ６０３）。判定には記憶方法管理テーブル（１４３、図５）のパリティ生成数欄（５０４）を利用する。パリティが必要な記憶方法である場合、記憶方法およびデータブロック数によって決まる数のパリティブロックを生成する（ステップ６０４）。例えば、記憶方法がＲＡＩＤ５（３Ｄ＋１Ｐ）である場合、データブロック３つに対して１つのパリティブロックが必要なので、ブロックサイズが３２ＫＢの場合、ファイルサイズ９６ＫＢ（３２ＫＢ×３）毎にひとつのパリティブロックを生成する。

次に書き込み先ディスク装置とその位置を決定する（ステップ６０５）。本発明では、空き領域が十分にあるディスク装置の台数以下で実現可能な記憶方法に対応可能である。よって、なるべく多種の記憶方法に対応するためにはディスク装置の空き領域が均等になるように、空き領域の多いディスク装置から優先的に書き込み先ディスク装置を決定する。例えば、４台のディスク装置に空き領域が存在する場合、ＲＡＩＤ５（３Ｄ＋１Ｐ）で記憶することは可能であるが、４台の内１台の空き領域がなくなると、ＲＡＩＤ５（３Ｄ＋１Ｐ）で記憶することができなくなる。空き領域はディスク空き容量管理テーブル（３０１）で管理する。ディスク装置内の書き込み位置は空きブロックの中から選択する。連続するブロックでなくても構わない。

次に前ステップで決定したディスク装置に書き込みを行う（ステップ６０６）。記憶方法によってはデータブロックだけでなくパリティブロックの書き込みも行う。また、記憶方法によっては同じデータブロックを複数のディスク装置に書き込む事もある。

最後に、関連する管理テーブルの内容を更新する（ステップ６０７）。本実施例ではファイル管理テーブル（１４１）、ディスク空き容量管理テーブル（３０１）、ディスク属性管理テーブル（３０２）が更新対象となる。

【００１９】

前記書き込み手順に従って実際に書き込みを行う例を以下に示す。ここではファイル管

理テーブル（１４１）が図２の状態、ディスク管理情報（１４２）が図３の状態、ディスク装置の利用状況が図４の状態である時に、図７に示す書き込み要求＃２から＃４が来た場合を想定する。

図７はユーザまたは上位アプリケーションからの書き込み要求を表すテーブルで、書き込み要求番号＃１は既に処理されているものとする。７０１には書き込み要求番号、７０２にはファイル名を含むパス、７０３にはファイルのサイズ、７０４には記憶方法、７０５にはデータブロックを格納する。例えば、７１１は書き込み要求＃２が「サイズ６０ＫＢのファイル『important.doc』をディレクトリ『/』に記憶方法『Ｒ１（２）』すなわち『ＲＡＩＤ１（２）』と呼ばれる、2台のディスク装置に二重化してデータを書き込む」要求であることを表す。

【００２０】

図７の書き込み要求＃２を処理する様子を説明する。はじめに当該書き込み要求を処理するのに十分な空き領域（２台のディスクに２ブロック以上の空き）があるかどうか調べる。次に、このファイルの先頭から３２ＫＢ分の「Ｆ２＿０」と残りのデータと未使用領域を含む「Ｆ２＿１」の２つのデータブロックに分割する。「Ｒ１（２）」にはパリティは不要なのでパリティブロックは生成しない。「Ｒ１（２）」で記憶するためには２台のディスク装置が必要なので、ディスク装置の空き容量テーブルから空き容量の多い順にディスク「Ｄ０」と「Ｄ２」を選択して、書き込みを行う。書き込み後のディスクの様子を図９に示す。ディスクＤ０のブロック１から（９１１）と、ディスクＤ２のブロック４から（９１２）の２箇所に同じデータ（Ｆ２＿０とＦ２＿１）が書き込まれたのが分かる。最後に、ファイル管理テーブルに当該ファイルの情報を登録し（図８の８１１）、ディスク管理情報を更新する（ディスクＤ０とＤ２の空き容量を２減らす）。ファイル管理テーブルの当該ファイルの行（８１１）にはデータの記録場所「Ｄ０＿１」と「Ｄ２＿４」が記録される。「Ｄ０＿１」はディスク装置「Ｄ０」のブロック「１」を意味する。

【００２１】

同様に図７の書き込み要求＃３を処理する様子を説明する。はじめに十分な空き領域があることを確認し、次に１５８ＫＢのファイルを３２ＫＢ毎に５つのデータブロックに分割する（先頭から順にＦ３＿０、Ｆ３＿１、Ｆ３＿２、Ｆ３＿３、Ｆ３＿４と呼ぶ）。記憶方法「Ｒ５（３＋１）」は３つのデータブロック毎に１つのパリティブロックを必要とするので、先頭から３つのデータブロック（Ｆ３＿０、Ｆ３＿１、Ｆ３＿２）に対して、パリティブロックを生成する（Ｆ３＿Ｐ１と呼ぶ）。さらに、残りの２つのデータブロック（Ｆ３＿３、Ｆ３＿４）とあらかじめ決めた特定のデータ列（全て０など）からなるダミーデータブロックに対して、パリティブロックを生成する（Ｆ３＿Ｐ２と呼ぶ）。次に、４つのディスク装置に分散してデータブロックおよびパリティブロックの書き込みを行う。パリティブロックを最初に書き込むディスクは「Next Parity Disk」の値「Ｄ３」とする。これは特定のディスクにパリティブロックが集中しないようにするために利用する。

【００２２】

書き込み後のディスクの様子を図１０に示す。ディスクＤ０のブロック４から（１０１１）、ディスクＤ１のブロック２から（１０１２）、ディスクＤ２のブロック６から（１０１３）、ディスクＤ３のブロック３から（１０１４）、の４箇所にデータおよびパリティブロックが書き込まれたのが分かる。最後に、ファイル管理テーブルに当該ファイルの情報を登録し（図８のファイル識別ＩＤ＃Ｆ３）、ディスク管理情報を更新する（各ディスクの空き容量を利用したブロック分だけ減らし、ディスク属性情報の「Next Parity Disk」を「Ｄ１」に変更する）。

この例では記憶方式ＲＡＩＤ５のパリティを分散させるアルゴリズムとして「Left Symmetric」と呼ばれる、ブロックの並びを順次左に回転させるものを適用した。すなわち、データの先頭から順に３つのデータブロックとそれに対応するパリティブロックを組み合わせて４つのブロックからなる「ブロックグループ」を作り、ブロックグループ毎にパリティブロックの位置が左へひとつずつずれるように４ブロックの並び全体を左回転させ

て書き込みを行う。例えば、最初のブロックグループに対応するパリティブロックを最初にディスクD3に書き込んだら、次の３つのデータブロックに対応するパリティブロックがディスクD2になるように、４ブロック全体を左にずらす。

【００２３】

このアルゴリズムを適用した書き込みの例を図１３に示す。テーブル１３０１に、合計７つのデータブロックからなるファイル「Ｆ７」をＲＡＩＤ５（３Ｄ＋１Ｐ）で記憶するために先頭から3ブロック毎にパリティブロックを生成した様子を示す。１３１１はブロックグループ番号を、１３１２はデータブロックを、１３１３はパリティブロックを意味する。これを書き込む際にパリティブロックの位置がずれるように配置した例をテーブル１３０２に示す。列１３２１はディスクＤ０に書き込むブロックを、列１３２２はディスクＤ１に書き込むブロックを、列１３２３はディスクＤ２に書き込むブロックを、列１３２４はディスクＤ３に書き込むブロックを表す。パリティブロックが順次ずれて記憶される様子が分かる。

【００２４】

同様に図７の書き込み要求＃４を処理する様子を説明する。はじめに十分な空き領域（４つのディスク装置にひとつずつの空きブロック）があることを確認し、次に１１０ＫＢのファイルを３２ＫＢ毎に４つのデータブロックに分割する（先頭から順にＦ４＿０、Ｆ４＿１、Ｆ４＿２、Ｆ４＿３と呼ぶ）。記憶方法「Ｒ０（４）」はデータを多重化せずに４分割して書き込む方法で、パリティは必要ない。書き込み後のディスクの様子を図１１に示す。ディスクＤ０のブロック７（１１１１）と、ディスクＤ１のブロック３（１１１２）、ディスクＤ２のブロック３（１１１３）、ディスクＤ３のブロック２（１１１４）の合計４箇所に分散してデータが書き込まれたのが分かる。最後に、ファイル管理テーブルに当該ファイルの情報を登録し（図８のファイル識別ＩＤ＃Ｆ４）、ディスク管理情報を更新する（各ディスク空き容量を１減らす）。

【００２５】

図１２にデータを読み込む際の手順を示す。はじめにファイルの管理情報を取得する（ステップ１２０１）。これにより、ファイルの記憶方法およびその記憶場所が分かる。次に読み込みを行うディスク装置を決定し（ステップ１２０２）、読み出し指示をディスク装置に送る（ステップ１２０３）。データを多重化して記憶している場合には記憶している全てのディスク装置に読み出し指示を送って最短で読み出しを行うことができるディスクからのみ読み込んでも良い。次に必要となる全データブロックが正しく読み出せたかどうか調べる（ステップ１２０４）。特定のディスク装置の故障などの原因でデータの一部が正常に読み出せない場合、パリティブロックから欠損したデータブロックの復旧を試みる（ステップ１２０５）。パリティブロックのない記憶方法である場合、または、パリティからデータブロックを復旧できない場合はエラーとなる（ステップ１２０６）。最後に全てのデータブロックが揃ったらそれらを連結して（ステップ１２０７）読み出し処理を終了する。

【００２６】

次に、記憶方法（ＲＡＩＤレベル）を変更する際の手順を説明する。はじめに、記憶方法変更対象となるファイルを図１２の読み込み手順で読み込む。次に、新たな記憶方法を指定して図６の書き込み手順で書き込みを行う。最後に、旧記憶方法で利用していた領域を解放する。多重度を２から１に変更する場合には、単純に空き領域の少ないディスク装置上のコピーを削除し、ファイル管理テーブルを更新する処理を行っても良い。また、逆に多重度を１から２に変更する場合には、当該ファイルが未使用のディスク装置の中から空き領域最大のディスク装置を選択してコピーを作成し、ファイル管理テーブルを更新する処理を行っても良い。ＲＡＩＤレベルの変更による記憶方法の変更は前記情報ライフサイクル管理を行う際にも効果的である。

【００２７】

次に、ファイルを削除する際の手順を説明する。はじめに、ファイル管理テーブルから削除対象ファイルの記憶位置を探し出し、当該ブロックをその連鎖をたどりながら「未使

用」領域とする。多重化して記憶している場合には、存在する存在する全てのコピーに対して本処理を行う。最後にファイル管理テーブルから当該ファイルの行を消去する。

【００２８】

本実施例では説明を簡単にするために、全てのファイルをルートディレクトリ（/）直下に配置したが、本発明の適用はルートディレクトリ直下のファイルに制限されるものではない。また、本実施例ではファイル管理テーブル自体を各ディスク装置の先頭ブロックに１ブロック分だけ存在する例を示したが、これに制限されるものではない。

本実施例ではファイル管理テーブル（１４１）にディスク装置内の物理ブロック番号を直接記憶したが、現在の多くのファイルシステムが実装しているように、ここに論理ブロック番号を記憶し、論理ブロック番号を介して物理ブロック番号に到達する方法も実現可能である。これにより、ファイル管理テーブルを変更することなく、ディスク上のデータブロックの位置を変更することが可能となる。

また、ディスク装置内のデータのつながりをたどるのに利用している情報４１３にブロック番号だけでなく、ディスク装置のＩＤを記憶する方法も実現可能である。これにより、ディスク装置をまたぐような、より自由度の高いブロック間のつながりが構築可能となる。

また、本実施例ではディスク装置として物理ディスクを想定して説明したが、本実施例のディスク装置として、複数の物理ディスクを束ねてひとつの仮想的なディスク装置として扱う技術を適用しても構わない。

また、ディスク装置のインターフェイスとしてＳＣＳＩなど直接接続型のインターフェイスだけでなく、ネットワークを介して接続するインターフェイスを適用しても構わない。その場合には、ディスク装置ＩＤとしてネットワークアドレスやＵＲＬなどが利用される可能性がある。

また、本実施例では説明を簡単化するために、書き込み対象のファイルのデータが全て揃った後、書き込みを行う例を説明したが、書き込み対象のファイルの先頭から記憶方法に応じた分のデータが確定した段階で、その分の書き込みを行う方法も適用可能である。この方法では、データ書き込みの最小単位のデータが揃った時点で一旦ひとつのファイルとして書き込みを行い、その後のデータの増加に応じて、追加書き込みを行いデータを末尾に連結する。追加書き込みを行う際には、直前に書き込んだデータブロックの次ブロックとして追加書き込みを行ったブロックのブロック番号を記憶する。例えば、記憶方法がＲＡＩＤ１（２）である場合、先頭のブロックサイズ（３２ＫＢ）分のデータが確定した段階で、その分の書き込み（二重化するため２ブロック分の書き込み）を実施し、データの増加に応じて順次書き込みを実施するようにする。記憶方法がＲＡＩＤ５（３Ｄ＋１Ｐ）である場合、先頭から３ブロック分（９６ＫＢ）が確定した段階で、パリティブロックを生成し、１ブロックグループ分の書き込みを行い、データの増加に応じて順次書き込みを実施する。直前の書き込み後のファイルサイズからパリティブロックの位置は計算可能である。

また、本実施例では書き込みを行う際に、各ディスク装置の空き領域が均等化するように書き込みを行うが、多数のファイルの削除が実行された場合、空き領域に偏りが生じる場合がある。このような場合に、空き領域を均等化する処理がディスク領域の有効利用につながると考えられる。また、新たなディスク装置を増設した際にもこの処理が有効に働く。

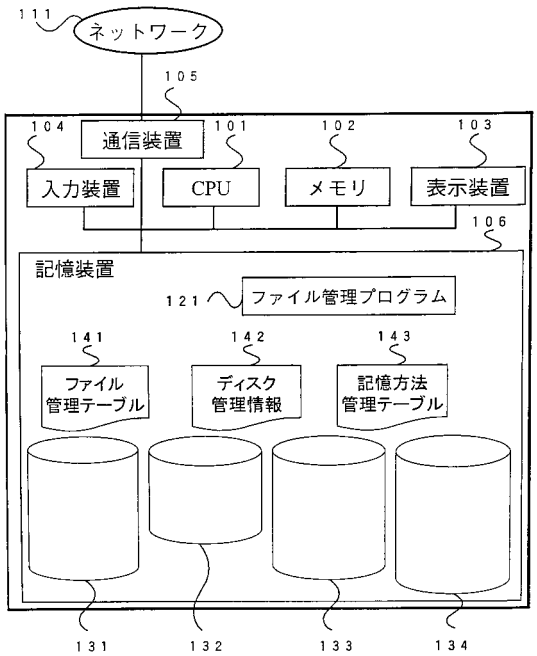
【００２９】

次に、ディスク装置の空き領域を均等化する処理の手順を説明する。はじめに、移動可能なファイルを検索する。空き領域最少のディスクを利用している、かつ、空き領域最大のディスクを利用していない、ファイルが移動対象として適当である。次に、空き領域最少のディスク上の当該ファイルの利用しているブロックを空き領域最大のディスク上にコピーする。次に、当該ファイルのファイル管理テーブルをコピー先を指すように更新する。最後に、当該ファイルが利用していた空き領域最少のディスク内の領域を解放する。

【図面の簡単な説明】

【００３０】
【図１】機器構成を示した図。
【図２】ファイル管理テーブルの構造を示した図（その１）。
【図３】ディスク管理情報の構造を示した図。
【図４】ディスク装置内のブロックの利用状況を示した図（その１）。
【図５】記憶方法管理テーブルの構造を示した図。
【図６】書き込み処理手順を示した図。
【図７】ファイル書き込み要求を示した図。
【図８】ファイル管理テーブルの構造を示した図。
【図９】ディスク装置内のブロックの利用状況を示した図（その２）。
【図１０】ディスク装置内のブロックの利用状況を示した図（その３）。
【図１１】ディスク装置内のブロックの利用状況を示した図（その４）。
【図１２】ファイル読み込み手順を示した図。
【図１３】パリティブロックの書き込み位置を説明するための図。
【符号の説明】
【００３１】
２０１　ファイル識別ＩＤ
２０２　ファイル名
２０３　ファイルサイズ
２０４　ファイル記憶方法
２０５　ファイル記憶開始位置情報
４１１　ディスク内ブロック番号
４１２　データ記憶領域
４１３　次データのブロック番号

【図１】



【図２】

ファイル管理テーブル（141）

201 #	202 Filename	203 Size	204 Method	205 Addr0	206 Addr1	207 Addr2	208 Addr3
M	MngTable	50KB	R0+1	D0_0	D1_0	D2_0	D3_0
F1	nothing.txt	70KB	-	D3_5	-	-	-

【図３】

ディスク管理情報(142)

301 311 312

Disk ID	Free Size
D0	8
D1	5
D2	7
D3	6

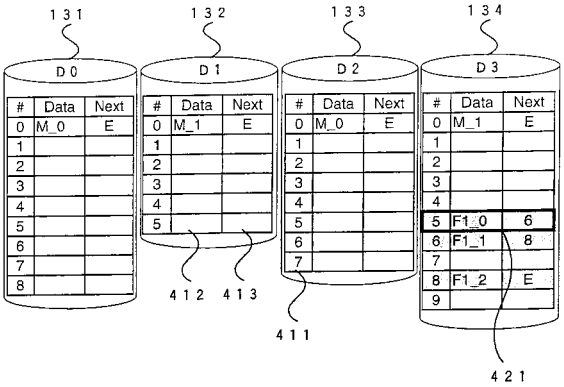
321

302 331 332

Attribute Name	Value
Next Parity Disk	D3
Block size	32KB

341 342

【図４】



【図５】

記憶方法管理テーブル(143)

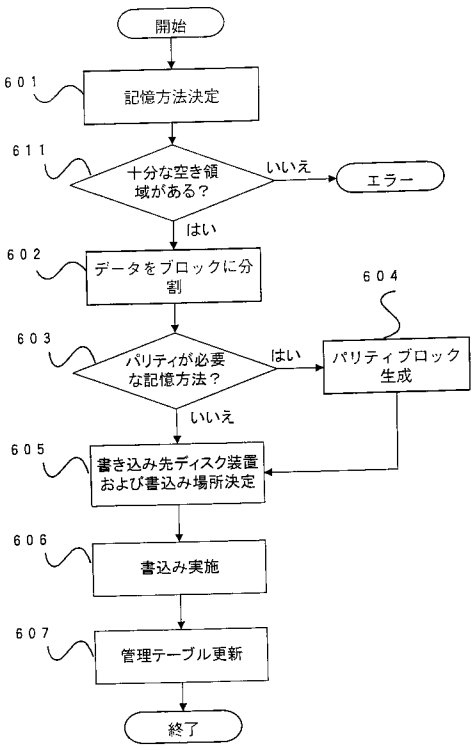
501 502 503 504

Method	Redundancy	Striping	Parity
-	1	1	0
R0+1	2	2	0
R0(4)	1	4	0
R1(2)	2	1	0
R5(3+1)	1	3	1

511

【図６】

書込み処理手順



【図７】

ファイル書き込み要求

701 702 703 704 705

#	File Path	Size	Method	Data Block				
1	/nothing.txt	70KB	-	F1_0	F1_1	F1_2	-	-
2	/important.doc	60KB	R1(2)	F2_0	F2_1	-	-	-
3	/balance.xls	158KB	R5(3+1)	F3_0	F3_1	F3_2	F3_3	F3_4
4	/speedy.jpg	110KB	R0(4)	F4_0	F4_1	F4_2	F4_3	-
5	...							

711

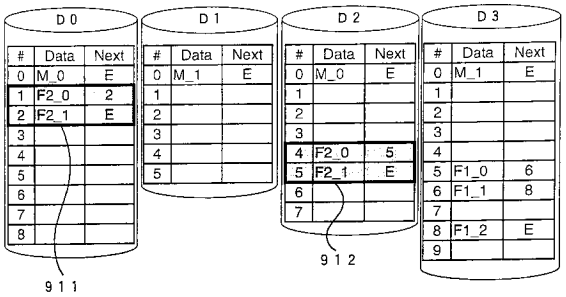
【図８】

ファイル管理テーブル

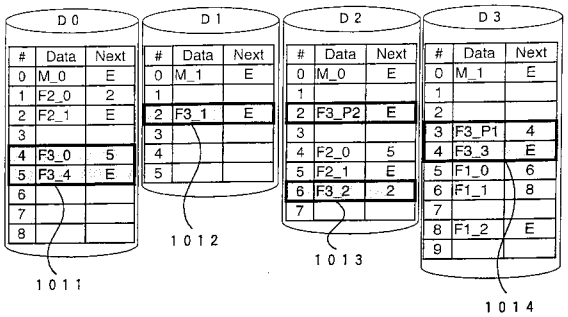
#	Filename	Size	Method	Addr0	Addr1	Addr2	Addr3
M	MngTable	50KB	R0+1	D0_0	D1_0	D2_0	D3_0
F1	nothing.txt	70KB	-	D3_5	-	-	-
F2	important.doc	60KB	R1(2)	D0_1	D2_4	-	-
F3	balance.xls	158KB	R5(3+1)	D0_4	D1_2	D2_6	D3_3
F4	speedy.jpg	110KB	R0(4)	D3_2	D2_3	D1_3	D0_7

811

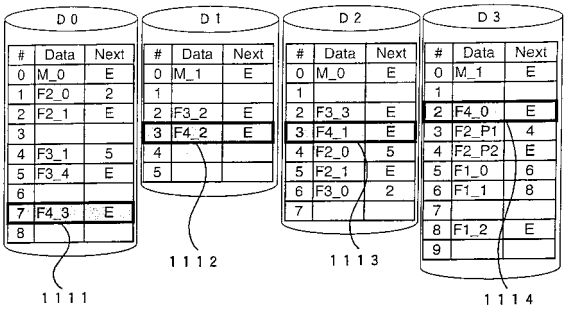
【図９】



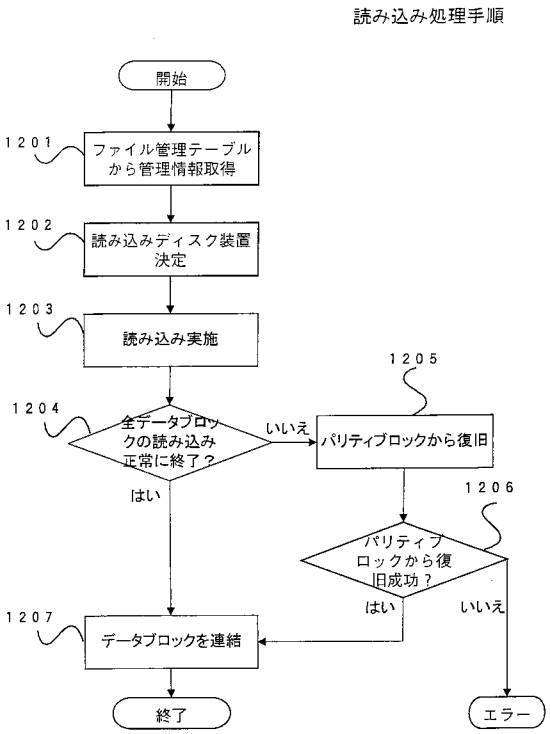
【図１０】



【図１１】



【図１２】



【図１３】