

(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2009-37564  
(P2009-37564A)

(43) 公開日 平成21年2月19日(2009.2.19)

(51) Int.Cl.	F I	テーマコード (参考)
<b>G06F 12/00 (2006.01)</b>	G06F 12/00 501B	5B065
<b>G06F 3/06 (2006.01)</b>	G06F 3/06 302J	5B082

審査請求 未請求 請求項の数 7 O L (全 31 頁)

(21) 出願番号 特願2007-203512 (P2007-203512)  
(22) 出願日 平成19年8月3日 (2007.8.3)

(71) 出願人 000005223  
富士通株式会社  
神奈川県川崎市中原区上小田中4丁目1番1号  
(74) 代理人 100090516  
弁理士 松倉 秀実  
(74) 代理人 100113608  
弁理士 平川 明  
(74) 代理人 100105407  
弁理士 高田 大輔  
(74) 代理人 100089244  
弁理士 遠山 勉  
(72) 発明者 古田 智徳  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

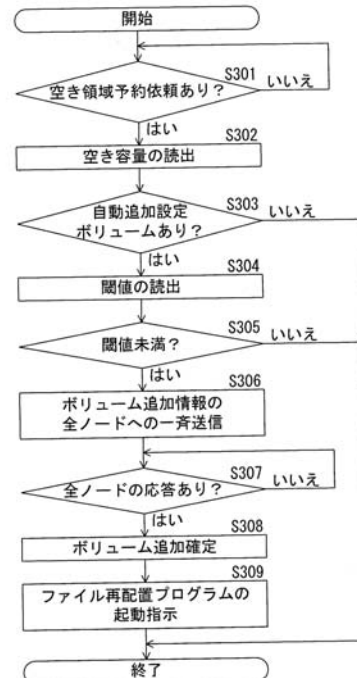
(54) 【発明の名称】 ディスク管理プログラム、ディスク管理方法及びディスク管理装置

(57) 【要約】

【課題】グローバルファイルシステムにおいてファイルの再配置が効率的に行えるようにすること。

【解決手段】領域管理用メタデータサーバマシン30は、何れかのノードマシン20～50がファイルアクセスに伴ってメタデータを問い合わせてきたときには、そのファイルが断片化状態にあるか否かを判定し(ステップS105～S108)、断片化状態にあれば、そのファイルに関する情報を断片化ファイル情報管理テーブル73に登録する(ステップS109)。また、領域管理用メタデータサーバマシン30は、この動作と並行して、操作者からボリュームの追加が指示されると(ステップS201、S202)、断片化ファイル情報管理テーブル73に情報が登録されているファイルを、既存のディスクから、追加されるディスクへ、順に移動させる(ステップS211、S309、S401～S403)。

【選択図】図13



**【特許請求の範囲】****【請求項 1】**

ファイルとディスクブロックとの関係を定義する領域情報、及び、名前空間を定義する名前情報を含むメタデータ、並びに、ファイルが記録されるディスクを複数のノードマシンに共用させるグローバルファイルシステムが実装されたクラスタシステムにおいてメタデータサーバマシンがディスクを管理するためのプログラムであって、

前記メタデータサーバマシンを、

何れかのノードマシンからファイルアクセスに伴うメタデータの問い合わせを受け付けると、そのファイルが断片化状態にあるか否かを判定する判定手段、

前記判定手段が当該ファイルが断片化状態にあると判定した場合に、そのファイルに関する情報を記憶装置内のテーブルに登録する登録手段、

任意のディスクの指定とともにボリューム追加の指示を入力装置を通じて操作者から受け付ける受付手段、及び、

前記受付手段がボリューム追加の指示を受け付けると、前記テーブルに情報が登録されているファイルを、既存ディスクから、前記受付手段で受け付けた指定により特定されるディスクへ、順に移動させる移動手段

として機能させる

ことを特徴とするディスク管理プログラム。

**【請求項 2】**

前記移動手段は、前記受付手段がボリューム追加の指示を受け付けた後、既存ディスクのボリュームが所定の状態になってから、ファイルの移動を行う

ことを特徴とする請求項 1 記載のディスク管理プログラム。

**【請求項 3】**

前記受付手段は、任意のディスクの指定とボリューム追加の指示とともに、ボリューム追加を実行すべき既存ディスクのボリュームの空き容量の条件を、入力装置を通じて操作者から受け付け、

前記移動手段は、前記受付手段がボリューム追加の指示を受け付けた後、既存ディスクのボリュームが前記受付手段が受け付けた空き容量の条件に合致した状態になってから、ファイルの移動を行う

ことを特徴とする請求項 2 記載のディスク管理プログラム。

**【請求項 4】**

前記判定手段は、当該ファイルのエクステンツ数が所定の閾値を超えている場合に、当該ファイルが断片化状態にあると判定する

ことを特徴とする請求項 1、2 又は 3 記載のディスク管理プログラム。

**【請求項 5】**

前記判定手段は、当該ファイルのアクセス数が所定の閾値を超えている場合に、当該ファイルが断片化状態にあると判定する

ことを特徴とする請求項 1、2 又は 3 記載のディスク管理プログラム。

**【請求項 6】**

ファイルとディスクブロックとの関係を定義する領域情報、及び、名前空間を定義する名前情報を含むメタデータ、並びに、ファイルが記録されるディスクを複数のノードマシンに共用させるグローバルファイルシステムが実装されたクラスタシステムにおいてメタデータサーバマシンがディスクを管理するための方法であって、

前記メタデータサーバマシンが、

何れかのノードマシンからファイルアクセスに伴うメタデータの問い合わせを受け付けると、そのファイルが断片化状態にあるか否かを判定する判定手順、

前記判定手順で当該ファイルが断片化状態にあると判定した場合に、そのファイルに関する情報を記憶装置内のテーブルに登録する登録手順、

任意のディスクの指定とともにボリューム追加の指示を入力装置を通じて操作者から受け付ける受付手順、及び、

10

20

30

40

50

前記受付手順でボリューム追加の指示を受け付けると、前記テーブルに情報が登録されているファイルを、既存ディスクから、前記受付手順で受け付けた指定により特定されるディスクへ、順に移動させる移動手順  
を実行する  
ことを特徴とするディスク管理方法。

【請求項 7】

ファイルとディスクブロックとの関係を定義する領域情報、及び、名前空間を定義する名前情報を含むメタデータ、並びに、ファイルが記録されるディスクを複数のノードマシンに共用させるグローバルファイルシステムが実装されたクラスタシステムにおいてディスクを管理するための装置であって、

10

何れかのノードマシンからファイルアクセスに伴うメタデータの問い合わせを受け付けると、そのファイルが断片化状態にあるか否かを判定する判定部、

前記判定部が当該ファイルが断片化状態にあると判定した場合に、そのファイルに関する情報をテーブルに登録する登録部、

任意のディスクの指定とともにボリューム追加の指示を操作者から受け付ける受付部、及び、

前記受付部がボリューム追加の指示を受け付けると、前記テーブルに情報が登録されているファイルを、既存ディスクから、前記受付部で受け付けた指定により特定されるディスクへ、順に移動させる移動部

を備えることを特徴とするディスク管理装置。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ディスクを管理するためのプログラム、方法及び装置に、関する。

【背景技術】

【0002】

周知のように、ストレージマシン内のディスクの記憶領域は、複数のセクタに細かく区割りされている。複数のセクタは、ファイルシステムによって、更に、連続する所定個数の単位ごとに、ブロックとして纏められ、各ブロックは、アドレスを付けて管理されている。通常、ファイルのデータ量は、1つのブロックの記憶容量より大きいため、ファイルシステムがファイルをディスクに記録する場合、ファイルは、複数のブロックに互って書き込まれる。ファイルを構成する複数の単位データが、それぞれどのブロックに記録されているかは、ファイルシステムにより管理されている（領域管理）。なお、各ファイルは、利用者の整理性の向上のため、階層的に分類されており、各階層に関するディレクトリ情報も、ファイルシステムにより管理されている（名前管理）。

30

【0003】

ファイルは、使用開始直後のディスクでは、連続したブロックに纏めて書き込まれるが、或る程度使用されたディスクでは、不連続のブロックに分散して書き込まれてしまうことが多い。これは、或る程度使用されたディスクでは、ファイルの書込と上書（すなわち割り当てブロックの増量又は減量）と消去とが繰り返されていることにより、所々に空きブロックが生じるためである。このようにファイルが不連続のブロックに書き込まれた状態は、断片化状態（フラグメンテーション）と称されており、ディスク内のファイルがこのように断片化されていると、ファイルアクセスにおける単位データのシークに時間が掛かってしまう。

40

【0004】

このようなファイルアクセス速度の低下を解消するため、従来、デフラグメンテーション機能が使用されている。デフラグメンテーション機能は、周知のように、断片化したファイルを、不連続なブロックから、連続したブロックに再配置する機能である。具体的には、デフラグメンテーション機能は、断片化したファイルの単位データを、不連続なブロックから消去して別のブロックへ一旦複製することにより、連続するブロックを確保し、

50

その後、一旦複製された単位データを、確保した連続ブロックへ移し替えるというものである。

【0005】

ところで、従来のファイルシステムは、ディスク内で名前空間（ファイルとその階層的分類（ディレクトリ）の体系）を完結させるものであったため、信頼性や拡張性に欠けていた。近年では、ファイルシステムの信頼性や拡張性を向上させるため、グローバルファイルシステムが開発されている。

【0006】

グローバルファイルシステムは、クラスタシステムを構成する複数のノードマシンに対し、同一のメタデータと同一のディスクとを共用させるファイルシステムである。メタデータは、主に、ファイルとディスクブロックとの関係を定義する領域情報、及び、名前空間を定義する名前情報からなり、各ノードマシンと同じネットワークに接続されたメタデータサーバマシンにおいて管理される。つまり、グローバルファイルシステムでは、ファイルシステムの主な三仕事である領域管理、名前管理、及び、ファイル操作制御のうち、領域管理及び名前管理を、メタデータサーバマシンが担当することとなる。一方、各ノードマシンは、ファイル操作制御の機能のみを備え、クライアントとしてメタデータを利用する。

【0007】

このようなグローバルファイルシステムにより複数のノードマシンに利用されるディスクでは、その空き容量が足りなくなってくると、前述したフラグメンテーションが発生していることが多い。そのため、空き容量の不足時には、前述したデフラグメンテーション機能によるファイルの再配置が必要となってくる。

【0008】

しかしながら、空き容量が足りないことから、断片化したファイルを構成する単位データを再配置する際に単位データを一旦複製しておくブロックも少なく、効率的にデフラグメンテーションが行えないという問題があった。

【0009】

なお、特許文献1には、ディスク内で断片化されている音声データのファイルを、視聴用マシンのディスクへ、単位データの再配置をしつつ短時間で転送する技術が、開示されている。但し、この特許文献1に係る技術は、データ転送に係るものであり、断片化されているファイルを同一ボリューム内で再配置するというものではない。

【0010】

【特許文献1】特開2006-040227号公報（段落0040）

【発明の開示】

【発明が解決しようとする課題】

【0011】

本発明は、前述したような従来技術の有する問題点に鑑みてなされたものであり、その課題は、グローバルファイルシステムにおいてファイルの再配置が効率的に行えるようにすることにある。

【課題を解決するための手段】

【0012】

上記の課題を解決するために案出されたディスク管理プログラムは、ファイルとディスクブロックとの関係を定義する領域情報、及び、名前空間を定義する名前情報を含むメタデータ、並びに、ファイルが記録されるディスクを複数のノードマシンに共用させるグローバルファイルシステムが実装されたクラスタシステムにおいて、メタデータサーバマシンを、何れかのノードマシンからファイルアクセスに伴うメタデータの問い合わせを受け付けると、そのファイルが断片化状態にあるか否かを判定する判定手段、その判定手段が当該ファイルが断片化状態にあると判定した場合に、そのファイルに関する情報を記憶装置内のテーブルに登録する登録手段、任意のディスクの指定とともにボリューム追加の指示を入力装置を通じて操作者から受け付ける受付手段、及び、その受付手段がボリューム

10

20

30

40

50

追加の指示を受け付けると、テーブルに情報が登録されているファイルを、既存ディスクから、受付手段で受け付けた指定により特定されるディスクへ、順に移動させる移動手段として機能させることを、特徴としている。

【0013】

このように構成されると、メタデータサーバマシンは、何れかのノードマシンがファイルアクセスに伴ってメタデータを問い合わせたときには、そのファイルが断片化状態にあるか否かを判定し、断片化状態にあれば、そのファイルに関する情報をテーブルに登録するよう、動作する。また、メタデータサーバマシンは、この動作と並行して、操作者からボリュームの追加が指示されると、テーブルに情報が登録されているファイルを、既存のディスクから、追加されるディスクへ、順に移動させるよう、動作する。このような動作によれば、既存ディスク内で断片化状態にあったファイルは、追加されるディスク内へ移動されることに伴い、連続したブロックに纏めて書き込まれることとなる。つまり、ディスクの空き容量が足りなくても、ファイルが効率的に再配置されるようになる。

10

【0014】

また、上記の課題を解決するために案出されたディスク管理方法は、ファイルとディスクブロックとの関係を定義する領域情報、及び、名前空間を定義する名前情報を含むメタデータ、並びに、ファイルが記録されるディスクを複数のノードマシンに共用させるグローバルファイルシステムが実装されたクラスタシステムにおいて、メタデータサーバマシンが、何れかのノードマシンからファイルアクセスを受け付けると、そのファイルが断片化状態にあるか否かを判定する判定手順、その判定手順で当該ファイルが断片化状態にあると判定した場合に、そのファイルに関する情報を記憶装置内のテーブルに登録する登録手順、任意のディスクの指定とともにボリューム追加の指示を入力装置を通じて操作者から受け付ける受付手順、及び、その受付手順でボリューム追加の指示を受け付けると、テーブルに情報が登録されているファイルを、既存ディスクから、受付手順で受け付けた指定により特定されるディスクへ、順に移動させる移動手順を実行することを、特徴としている。

20

【0015】

従って、このディスク管理方法を実現したメタデータサーバマシンは、前述した本発明のディスク管理プログラムが動作したメタデータサーバマシンと同等に機能することとなる。

30

【0016】

また、上記の課題を解決するために案出されたディスク管理装置は、ファイルとディスクブロックとの関係を定義する領域情報、及び、名前空間を定義する名前情報を含むメタデータ、並びに、ファイルが記録されるディスクを複数のノードマシンに共用させるグローバルファイルシステムが実装されたクラスタシステムにおいてディスクを管理するため、何れかのノードマシンからファイルアクセスを受け付けると、そのファイルが断片化状態にあるか否かを判定する判定部、その判定部が当該ファイルが断片化状態にあると判定した場合に、そのファイルに関する情報をテーブルに登録する登録部、任意のディスクの指定とともにボリューム追加の指示を操作者から受け付ける受付部、及び、その受付部がボリューム追加の指示を受け付けると、テーブルに情報が登録されているファイルを、既存ディスクから、受付部で受け付けた指定により特定されるディスクへ、順に移動する移動部を備えることを、特徴としている。

40

【0017】

従って、このディスク管理装置は、前述した本発明のディスク管理プログラムが動作したメタデータサーバマシンと同等に機能することとなる。

【発明の効果】

【0018】

以上に説明したように、本発明によれば、グローバルファイルシステムにおいてファイルの再配置が効率的に行えるようになる。

【発明を実施するための最良の形態】

50

## 【 0 0 1 9 】

以下、本発明を実施するための最良の形態について、添付図面を参照しながら、説明する。

## 【 0 0 2 0 】

まず、本実施形態のコンピュータネットワークシステムのハードウェア構成とソフトウェア構成について、説明する。

## 【 0 0 2 1 】

本実施形態のコンピュータネットワークシステムは、図 1 のネットワーク構成図に示すように、ストレージマシン 1 0、名前管理用メタデータサーバマシン 2 0、領域管理用メタデータサーバマシン 3 0、デフラグ用メタデータサーバマシン 4 0、汎用サーバマシン 5 0、及び、クライアントマシン 6 0 からなる。このうち、ストレージマシン 1 0 と各サーバマシン 2 0 ~ 5 0 とは、図示せぬファイバチャンネルスイッチを介して通信自在に接続されることにより、S A N [ Storage Area Network ] を構成している。また、各サーバマシン 2 0 ~ 5 0 とクライアントマシン 6 0 とは、図示せぬルータを介して通信自在に接続されることにより、L A N [ Local Area Network ] を構成している。なお、汎用サーバマシン 5 0 は、図 1 では 1 台しか示されていないが、2 台以上存在していても良い。また、ストレージマシン 1 0 とクライアントマシン 6 0 は、図 1 ではともに 2 台しか示されていないが、何れも 1 台しか存在していても良いし、3 台以上存在していても良い。

10

## 【 0 0 2 2 】

また、本実施形態のサーバマシン 2 0 ~ 5 0 には、異機種間に単一のファイルシステム機能を提供するグローバルファイルシステムが適用されている。なお、周知のように、ファイルシステムは、主に、名前管理、領域管理、及び、ファイル操作制御を、行うものであるが、このうち、ファイル操作制御を実現する機能は、後述するように、グローバルファイルシステムのクライアントとして、全てのサーバマシン 2 0 ~ 5 0 に実装されている。一方、名前管理及び領域管理を実現する機能は、後述するように、グローバルファイルシステムのサーバとして、メタデータサーバマシン 2 0 ~ 4 0 に実装されている。

20

## 【 0 0 2 3 】

ここで、グローバルファイルシステムの名前管理及び領域管理を実現する機能は、一般的には、1 台のメタデータサーバマシンに実装されることが多いが、本実施形態では、名前管理と領域管理とは、別々のマシン 2 0、3 0 に実装されている。このように名前管理と領域管理とが別々のマシンに実装されているのは、デフラグメンテーション機能によるファイル再配置処理がファイルアクセス処理に与える影響を最小にするためである。これは、一般的なファイルアクセス処理では、ディレクトリ検索が頻繁に行われて名前管理機能に負荷が掛かるのに対し、領域管理機能には相対的に負荷が掛からないという事情、及び、ファイル再配置処理では、ファイルが記録されているディスクブロックが変更されることに伴って領域情報が頻繁に更新されて領域管理機能に負荷が掛かるのに対し、名前管理機能には相対的に負荷が掛からないという事情に基づいている。

30

## 【 0 0 2 4 】

更に、本実施形態では、領域管理機能に一般的に含まれるデフラグメンテーション機能を、領域管理用メタデータサーバマシン 3 0 とは別のマシン 4 0 に担当させている。これは、ファイル再配置処理において負荷が掛かる処理として、ファイル移動に伴うデータ複製と、領域情報の更新処理とがあり、これらを別々のサーバマシンに担当させた方が、デフラグ処理の負荷を分散できるためである。

40

## 【 0 0 2 5 】

なお、以上に説明したように、サーバマシン 2 0 ~ 5 0 は、ファイルシステムに掛かる負荷を分散したシステム ( クラスタ ) を構成しているため、ノードマシンと称することもできる。

## 【 0 0 2 6 】

ストレージマシン 1 0 は、サーバマシン 2 0 ~ 5 0 がクライアントマシン 6 0 に提供する機能や情報を記録しておくための装置であり、図 2 の構成図に示すように、ディスクユ

50

ニット10a、ストレージコントローラ10b、及び、HBA [Host Bus Adapter] 10cを、内蔵している。ディスクユニット10aは、データを記録するための1枚以上のディスク、ディスクの回転機構、及び、読み書きヘッドを内蔵するユニットである。ストレージコントローラ10bは、メタデータサーバマシン20～50からのコマンドに従ってディスクに対するデータの読み書きを制御するユニットである。HBA 10cは、ファイバチャネルプロトコルに従ってメタデータサーバマシン20～50とデータの遣り取りを行うための通信アダプタである。

#### 【0027】

名前管理用メタデータサーバマシン20は、前述した名前管理を担当するメタデータサーバマシンであり、図3の構成図に示すように、ストレージユニット20a、CPU [Central Processing Unit] 20b、メモリユニット20c、HBA 20d、及び、LANアダプタ20eを、内蔵している。ストレージユニット20aは、各種のプログラムやデータを記録しておくためのユニットである。CPU 20bは、ストレージユニット20aに記録されているプログラムに従って処理を行うユニットである。メモリユニット20cは、CPU 20bがプログラムやデータをキャッシュしたり作業領域を展開したりするユニットである。HBA 20dは、ファイバチャネルプロトコルに従ってストレージマシン10とデータの遣り取りを行うための通信アダプタである。LANアダプタ20eは、TCP/IP [Transmission Control Protocol/Internet Protocol] 及びイーサネットプロトコル（イーサネットは、米ゼロックス社及び米ヒューレットパカード社商標）に従ってクライアントマシン60とデータの遣り取りを行うための通信アダプタである。

10

20

#### 【0028】

この名前管理用メタデータサーバマシン20は、ストレージユニット20aに、アプリケーション21、オペレーティングシステム22、及び、ファイルシステム23を、記憶している。アプリケーション21は、クライアントマシン60に提供される各種機能を実現するソフトウェアであり、このアプリケーション21で実現される機能としては、例えば、ウェブサーバ、ウェブアプリケーション、ファイルサーバ、メールサーバなどがある。オペレーティングシステム22は、HBA 20dやLANアダプタ20eにおけるデータの入出力の管理や、メモリユニット20cの記憶領域の管理や、複数のタスクにウィンドウを割り当てて画面出力を多重化するウィンドウシステムや、各種設定ツールといった基本的な機能を、アプリケーション21を含む多くのアプリケーションに提供するためのソフトウェアである。ファイルシステム23は、前述したグローバルファイルシステムを実現するためのソフトウェアであり、そのグローバルファイルシステムによる機能のうち、クライアントとしてファイル操作制御機能を実現するプログラム23a、及び、名前空間（ファイルとその階層的分類（ディレクトリ）の体系）を管理する機能を実現するためのプログラム23bを、含んでいる。ファイル操作制御機能23aは、ファイルの読み出し、書き込み、複製、上書き、消去、名称変更、ディレクトリの移動、ディレクトリの作成、ディレクトリの削除と言ったファイルに関する操作をアプリケーション21に提供するプログラムである。名前管理機能23bは、メタデータのうち、名前空間（ファイル名、）を定義する名前情報（ファイルに関する情報、ディレクトリに関する情報、ディレクトリ構造に関する情報）を管理するとともに、ファイル名やディレクトリ名の検索を行うためのプログラムである。

30

40

#### 【0029】

領域管理用メタデータサーバマシン30は、前述した領域管理を担当するメタデータサーバマシンであり、図4の構成図に示すように、ストレージユニット30a、CPU 30b、メモリユニット30c、HBA 30d、及び、LANアダプタ30eを、内蔵している。この領域管理用メタデータサーバマシン30も、名前管理用メタデータサーバマシン20と同様に、ストレージユニット30aに、アプリケーション31、オペレーティングシステム32、及び、ファイルシステム33を、記憶している。アプリケーション31及びオペレーティングシステム32は、名前管理用メタデータサーバマシン20と同等の機能を発揮するソフトウェアである。ファイルシステム33は、前述したグローバルファイ

50

ルシステムを実現するためのソフトウェアであり、そのグローバルファイルシステムによる機能のうち、クライアントとしてファイル操作制御機能を実現するプログラム 3 3 a を、含んでいる。

【 0 0 3 0 】

また、ファイルシステム 3 3 は、領域管理機能を実現するためのプログラム 3 3 b、フラグメント情報収集プログラム 3 3 c、ボリュームマネージャ 3 3 d、及び、空き容量監視プログラム 3 3 e をも、含んでいる。領域管理機能 3 3 b は、ファイルを構成する単位データとディスクブロックのアドレスとの対応付けを管理する機能を実現するプログラムである。この領域管理機能 3 3 b は、通常のファイルとディレクトリ情報を格納するファイルとに対し、一意な識別番号としてアイノード番号を付与し、それらファイルに関する情報を、図 5 に示すようなアイノード管理テーブル 7 1 に記録して管理するようになっている。また、この領域管理機能 3 3 b は、ストレージマシン 1 0 内のディスクのボリューム（記憶領域）に関する情報を、図 6 に示すようなボリューム管理テーブル 7 2 に記録して管理するようになっている。これらテーブル 7 1、7 2 は、何れかのストレージマシン 1 0 内に格納されていても良いし、領域管理用メタデータサーバマシン 3 0 のストレージユニット 3 0 a 内に格納されていても良い。

10

【 0 0 3 1 】

アイノード管理テーブル 7 1 は、ファイル（アイノード）と同数のレコードを、有している。各レコードは、図 5 の模式的なテーブル構造図に示すように、「アイノード」、「サイズ」、「ブロック数」、「エクステント数」、「最終アクセス日時」、「アクセス数」及び「ブロック」の各フィールドを、有している。「アイノード」フィールドは、そのファイルのアイノード番号が記録されるフィールドである。「サイズ」フィールドは、そのファイルのデータ量が記録されるフィールドである。「ブロック数」フィールドは、そのファイルがディスク上で占めているブロックの数が記録されるフィールドである。「エクステント数」フィールドは、そのファイルのエクステントの数が記録されるフィールドである。なお、エクステントとは、そのファイルが記録されているディスクブロックのうち、連続しているブロックの塊を言う。「最終アクセス日時」フィールドは、そのファイルへのアクセスのうち、最後になされたアクセスの日時が記録されるフィールドである。「アクセス数」フィールドは、そのファイルへのアクセスがなされた回数が記録されるフィールドである。「ブロック」フィールドは、そのファイルが記録されているディスクブロックを特定するアドレス情報が記録されるフィールドである。

20

30

【 0 0 3 2 】

ボリューム管理テーブル 7 2 は、ストレージマシン 1 0 内のディスクと同数のレコードを、有している。各レコードは、図 6 の模式的なテーブル構造図に示すように、「ボリューム」、「ボリューム名」、「サイズ」、「空き容量」、「組込」、「自動追加」及び「再配置」の各フィールドを、有している。「ボリューム」フィールドは、そのディスクのボリューム（記憶領域）を特定するための一意な識別番号であるボリューム番号が記録されるフィールドである。「ボリューム名」フィールドは、そのボリュームの名称が記録されるフィールドである。「サイズ」フィールドは、そのボリュームの容量が記録されるフィールドである。「空き容量」フィールドは、そのボリュームのうちの空き領域の容量が記録されるフィールドである。「組込」フィールドは、そのボリュームが使用対象に組み込み済みであるか否かを特定する情報、及び、使用対象に組み込み済みである場合において使用が留保された状態にあるか否かを特定する情報が、記録されるフィールドである。なお、「組込」フィールドには、そのボリュームが使用対象に組み込まれていないことを示す「未」、使用対象に組み込まれてはいるが使用が留保された状態となっていることを示す「可」、又は、使用対象に組み込まれていて使用の留保が解除された状態にあることを示す「済」が、記録される。「自動追加」フィールドは、そのボリュームが使用留保の状態にある場合において、使用対象に組み込み済みであって使用留保が解除された全てのボリューム（以下、使用中ボリュームと表記する）の空き容量が所定の状態になったときにそのボリュームの使用留保を自動的に解除するか否かを示す情報が記録されるフィール

40

50

ドである。なお、ボリュームが自動追加（使用留保を自動解除）すべきもの設定される場合には、その自動追加の開始条件となる使用中ボリュームの空き領域の容量の閾値が、そのボリュームに係る「自動追加」フィールドに記録される。また、使用対象に組み込まれていないボリュームのレコードの「自動追加」フィールドは、空欄となる。「再配置」フィールドは、そのボリュームが自動追加（使用留保の自動解除）される場合にファイルの再配置を行うか否かを示す情報が記録されるフィールドである。なお、使用対象に組み込まれていないボリュームのレコードの「再配置」フィールドは、空欄となる。

【0033】

また、図4の領域管理機能33bは、起動後、メモリユニット30c内に、図7に示すような断片化ファイル情報管理テーブル73を新規作成する。この断片化ファイル情報管理テーブル73は、後述の処理により断片化状態と判断されたファイルに関する情報を記録しておくためのワークテーブルである。従って、起動直後は、この断片化ファイル情報管理テーブル73は、一つもレコードが記録されていない状態となっている。この断片化ファイル情報管理テーブル73の各レコードは、図7の模式的なテーブル構造図に示すように、「アイノード」、「サイズ」、「ブロック数」、「エクステント数」、「最終アクセス日時」及び「アクセス数」の各フィールドを、有している。これら各フィールドは、何れも、図5のアイノード管理テーブル71における同じ名称のフィールドと同等の値が記録されるものとなっている。

【0034】

また、図4のフラグメント情報収集プログラム33cは、何れかのサーバマシン20～50のファイル操作制御機能23a、33a、43aがファイルにアクセスした際、そのファイルが断片化状態にあるか否かを特定するためのプログラムである。このフラグメント情報収集プログラム33cに従ってCPU30bが実行する処理の内容については、図9を用いて後述する。

【0035】

また、図4のボリュームマネージャ33dは、ボリュームの使用対象への組み込み、及び、ボリュームの使用対象からの削除を行うためのツールを操作者に提供するためのプログラムである。このボリュームマネージャ33dに従ってCPU30bが実行する処理の内容については、図10乃至図12を用いて後述する。

【0036】

また、図4の空き容量監視プログラム33eは、使用中ボリュームの空き領域の容量が所定の状態になっている場合に、自動追加設定がなされているボリュームの使用留保を解除するためのプログラムである。この空き容量監視プログラム33eに従ってCPU30bが実行する処理の内容については、図13を用いて後述する。

【0037】

また、図1のデフラグ用メタデータサーバマシン40は、ファイル再配置処理を担当するメタデータサーバマシンであり、図8の構成図に示すように、ストレージユニット40a、CPU40b、メモリユニット40c、HBA40d、及び、LANアダプタ40eを、内蔵している。このデフラグ用メタデータサーバマシン40も、前述した各メタデータサーバマシン20、30と同様に、ストレージユニット40aに、アプリケーション41、オペレーティングシステム42、及び、ファイルシステム43を、記憶している。アプリケーション41及びオペレーティングシステム42は、前述した各メタデータサーバマシン20、30と同等の機能を発揮するソフトウェアである。ファイルシステム43は、前述したグローバルファイルシステムを実現するためのソフトウェアであり、そのグローバルファイルシステムによる機能のうち、クライアントとしてファイル操作制御機能を実現するプログラム43aを、含んでいる。さらに、このファイルシステム43は、ファイル再配置プログラム43bを、含んでいる。ファイル再配置プログラム43bは、ボリュームが手動で使用対象に組み込まれ、或いは、自動追加設定がなされているボリュームの使用留保が自動解除される場合に、図7の断片化ファイル情報管理テーブル73に登録されているファイルをそのボリュームへ移動するためのプログラムである。このファイル

10

20

30

40

50

再配置プログラム 4 3 b に従って CPU 4 0 b が実行する処理の内容については、図 1 4 を用いて後述する。

【 0 0 3 8 】

図 1 の汎用サーバマシン 5 0 は、クライアントマシン 6 0 に提供するとともに、前述したグローバルファイルシステムをクライアントとして利用するマシンである。従って、この汎用サーバマシン 5 0 は、図示していないが、ストレージユニット、CPU、メモリユニット、HBA、及び、LANアダプタを、内蔵しており、アプリケーション、オペレーティングシステム、ファイルシステムを、ストレージユニットに記憶している。そして、そのファイルシステムは、前述したように、ファイル操作制御機能を含んでいる。

【 0 0 3 9 】

クライアントマシン 6 0 は、例えば、ウェブブラウザ、メーラー、エミュレータのように、アプリケーション 2 1、3 1、4 1 が提供する機能を楽しむための機能が付加された LAN 通信自在なコンピュータである。

【 0 0 4 0 】

次に、本実施形態のコンピュータネットワークシステムで行われる処理について、説明する。

【 0 0 4 1 】

各サーバマシン 2 0 ~ 5 0 のファイルシステムに含まれるファイル操作制御機能 2 3 a、3 3 a、4 3 a は、周知のものであるので、簡単に説明する。このファイル操作制御機能 2 3 a、3 3 a、4 3 a は、ファイルシステムの起動とともに、又は、実行終了後直ちに、実行される。CPU は、ファイル操作制御処理の開始後、アプリケーションからファイル読み出しの指示を受け付けると、図 3 の名前管理機能 2 3 b に問い合わせ、対象となるファイルのアイノード番号を取得し、その後、そのアイノード番号を使って図 4 の領域管理機能 3 3 b に問い合わせ、対象となるファイルが記録されているブロックの情報を取得し、それらブロックが所属する対応するストレージマシン 1 0 へアクセスする。また、CPU は、ファイル操作制御処理の開始後、アプリケーションからファイル書き込みの指示を受け付けると、図 4 の領域管理機能 3 3 b に、書き込みのための空き領域の予約を依頼し、続いて、指定されたディレクトリに新規ファイルを登録するための処理を図 3 の名前管理機能 2 3 b に依頼する。その後、CPU は、予約しておいた空き領域にファイルを書き込むとともに、当該ファイルに領域が割り当てられた旨を図 4 の領域管理機能 3 3 b と図 3 の名前管理機能 2 3 b とに通知する。図 4 の領域管理機能 3 3 b は、この通知を受けると、当該ファイルに関する情報を図 5 のアイノード管理テーブル 7 1 に登録する。

【 0 0 4 2 】

図 4 のフラグメント情報収集プログラム 3 3 c は、ファイルシステムの起動とともに、又は、実行終了後直ちに、実行される。図 9 の流れ図に示すように、CPU 3 0 b は、フラグメント情報収集処理の開始後、最初のステップ S 1 0 1 において、何れかのファイルシステムのファイル操作制御機能 2 3 a、3 3 a、4 3 a からファイルアクセス（ファイルの読み出し又は書き込み）があるまで待機する。そして、ファイルアクセスがあると、CPU 3 0 b は、ステップ S 1 0 2 へ処理を進める。

【 0 0 4 3 】

ステップ S 1 0 2 では、CPU 3 0 b は、図 5 のアイノード管理テーブル 7 1 内のレコードのうち、ファイルアクセス時にファイル操作制御機能 2 3 a、3 3 a、4 3 a から通知されたアイノード番号を含むレコードを特定し、その特定したレコードの「最終アクセス日時」フィールドの値をその時点の時刻で更新する。また、CPU 3 0 b は、当該レコードの「アクセス数」フィールドの値を 1 つ増やす。

【 0 0 4 4 】

次のステップ S 1 0 3 では、CPU 3 0 b は、図 5 のアイノード管理テーブル 7 1 における当該ファイルのレコードの「ブロック」フィールドにおいて、ブロック数を計数する。続いて、CPU 3 0 b は、計数して得たブロック数と当該レコードの「ブロック数」フィールドの値とが一致するか否かを、判別する。そして、計数して得たブロック数と当該

10

20

30

40

50

レコードの「ブロック数」フィールドの値とが一致した場合、CPU30bは、書き込みによってブロック数が増加していないとして、ステップS103からS105へ処理を分岐させる。一方、計数して得たブロック数と当該レコードの「ブロック数」フィールドの値とが一致しなかった場合、CPU30bは、書き込みによってブロック数が増加したとして、ステップS104へ処理を進める。

【0045】

ステップS104では、CPU30bは、図5のアイノード管理テーブル71における当該ファイルのレコードの「ブロック数」フィールドの値を更新する。また、CPU30bは、当該レコードの「ブロック」フィールドにおいて、エクステント数を計数し、当該レコードの「エクステント数」フィールドの値を更新する。その後、CPU30bは、ステップS105へ処理を進める。

10

【0046】

ステップS105では、CPU30bは、図5のアイノード管理テーブル71における当該ファイルのレコードの「エクステント数」フィールドから値を読み出す。

【0047】

次のステップS106では、CPU30bは、ステップS105で読み出したエクステント数が所定の閾値を超えているか否かを、判別する。そして、エクステント数が所定の閾値を超えていた場合、CPU30bは、ステップS109へ処理を進める。一方、エクステント数が所定の閾値を超えていなかった場合、CPU30bは、ステップS106からS107へ処理を分岐させる。

20

【0048】

ステップS107では、CPU30bは、図5のアイノード管理テーブル71における当該ファイルのレコードの「アクセス数」フィールドから値を読み出す。

【0049】

次のステップS108では、CPU30bは、ステップS107で読み出したアクセス数が所定の閾値を超えているか否かを、判別する。そして、アクセス数が所定の閾値を超えていた場合、CPU30bは、ステップS109へ処理を進める。一方、アクセス数が所定の閾値を超えていなかった場合、CPU30bは、ステップS108から処理を分岐させ、図9に係るフラグメント情報収集処理を終了する。

【0050】

ステップS109では、CPU30bは、図5のアイノード管理テーブル71における当該ファイルのレコードを読み出し、図7の断片化ファイル情報管理テーブル73へ追加登録する。その後、CPU30bは、図9に係るフラグメント情報収集処理を終了する。

30

【0051】

このフラグメント情報収集プログラム33cによれば、ファイルアクセスがある度に、そのファイルが断片化された状態にあるか否かが、領域管理用メタデータサーバマシン30によって判別され(ステップS106、S108)、断片化ファイルであった場合には、そのファイルの情報が、図7の断片化ファイル情報管理テーブル73に登録される(ステップS109)。

【0052】

なお、ステップS105乃至S108を実行するCPU30bは、前述した判定手段及び前述した判定部に相当し、ステップS105乃至S108は、前述した判定手順に相当している。また、ステップS109を実行するCPU30bは、前述した登録手段及び前述した登録部に相当し、ステップS109は、前述した登録手順に相当している。

40

【0053】

このフラグメント情報収集処理と並行して、領域管理用メタデータサーバマシン30では、図示せぬ入力ユニットを介して操作者から起動指示を受け付けることにより、ボリュームマネージャ33dが実行されるようになっている。図10及び図11の流れ図に示すように、CPU30bは、処理開始後、最初のステップS201において、ボリューム設定画面を図示せぬ表示ユニットに表示する。図12の画面例に示すように、ボリューム設

50

定画面 7 4 は、図 6 のボリューム管理テーブル 7 2 において「組込」フィールドの値が「済」又は「可」であるボリュームの名称が列挙される構成ボリューム欄 7 4 a、及び、「組込」フィールドの値が「未」であるボリュームの名称が列挙される追加候補ボリューム欄 7 4 b を、含んでいる。また、ボリューム設定画面 7 4 は、追加候補ボリューム欄 7 4 b において選択されたボリュームを使用対象へ組み込むことを指示するための追加ボタン 7 4 c、及び、構成ボリューム欄 7 4 a において選択されたボリュームを使用対象から取り外すことを指示するための削除ボタン 7 4 d を、含んでいる。更に、ボリューム設定画面 7 4 は、ボリュームを使用対象へ組み込む際の付随的な設定事項として、そのボリュームを直ちに使用対象に組み込んで使用状態にするか、それとも、そのボリュームを使用対象に組み込んで使用を留保状態にしておき使用中ボリュームの空き領域の容量が所定値以下になったときにそのボリュームの使用留保を解除するかを、決定するためのラジオボタン 7 4 e、7 4 e を、含んでいる。また、ボリューム設定画面 7 4 は、使用中ボリュームの空き容量に応じてボリュームの自動追加（使用留保の解除）を行うよう設定する場合において、そのボリュームの自動追加が行われる条件を入力するためのテキストボックス 7 4 f を、含んでいる。また、ボリューム設定画面 7 4 は、ボリュームの追加が行われる場合にファイルの再配置を同時に行うことを指定するためのチェックボックス 7 4 g も、含んでいる。更に、ボリューム設定画面 7 4 は、各ボタン 7 4 c、7 4 d、ラジオボタン 7 4 e、テキストボックス 7 4 f、チェックボックス 7 4 g を使って設定した内容で、ボリュームの登録を行うときにクリックすべき実行ボタン 7 4 h を、含んでいる。また、ボリューム設定画面 7 4 は、これら設定行為を取り止めるときにクリックすべき中止ボタン 7 4 i を、含んでいる。CPU 3 0 b は、この図 1 2 に示すようなボリューム設定画面 7 4 を図示せぬ表示ユニットに表示すると、図 1 0 のステップ S 2 0 2 へ処理を進める。

#### 【0054】

ステップ S 2 0 2 では、CPU 3 0 b は、ボリューム設定画面 7 4 の実行ボタン 7 4 h 及び中止ボタン 7 4 i のうち、何れかのボタンがクリックされるまで、待機する。そして、何れかのボタンがクリックされた場合、CPU 3 0 b は、ステップ S 2 0 3 へ処理を進める。

#### 【0055】

ステップ S 2 0 3 では、CPU 3 0 b は、クリックされたボタンが中止ボタン 7 4 i であるか否かを、判別する。そして、クリックされたボタンが中止ボタン 7 4 i であった場合、CPU 3 0 b は、図 1 0 及び図 1 1 に係る処理を終了する。一方、クリックされたボタンが実行ボタン 7 4 h であった場合、CPU 3 0 b は、ステップ S 2 0 3 からステップ S 2 0 4 へ処理を分岐させる。

#### 【0056】

ステップ S 2 0 4 では、CPU 3 0 b は、ボリュームの追加（使用対象への組み込み）が選択されていたかボリュームの削除が選択されていたかを、判別する。そして、ボリュームの削除が選択されていた場合、CPU 3 0 b は、ステップ S 2 0 4 からステップ S 2 0 5 へ処理を分岐させる。

#### 【0057】

ステップ S 2 0 5 では、CPU 3 0 b は、実行ボタン 7 4 h がクリックされた時点で指定されていたボリュームを使用対象から取り外す処理を行う。この処理では、図 6 のボリューム管理テーブル 7 2 における当該ボリュームのレコードの「組込」フィールドの値が「未」へ切り替えられて、当該ボリュームの登録が抹消される。その後、CPU 3 0 b は、図 1 0 及び図 1 1 に係る処理を終了する。

#### 【0058】

一方、ステップ S 2 0 4 において、ボリュームの追加（使用対象への組み込み）が選択されていた場合、CPU 3 0 b は、ステップ S 2 0 6 へ処理を進める。

#### 【0059】

ステップ S 2 0 6 では、CPU 3 0 b は、実行ボタン 7 4 h がクリックされた時点で、ボリュームの追加（使用対象への組み込みと使用状態への移行）を直ちに行うことを指定

するラジオボタン74eが選択されていたか否かを、判別する。そして、使用中ボリュームの空き領域の容量が所定値以下になったときにそのボリュームの追加（使用留保の解除）を行うことを指定するラジオボタン74eが選択されていた場合、CPU30bは、ステップS206からステップS207へ処理を分岐させる。

【0060】

ステップS207では、CPU30bは、実行ボタン74hがクリックされた時点で指定されていたボリュームを使用対象へ組み込む処理を行う。この処理では、図6のボリューム管理テーブル72における当該ボリュームのレコードの「組込」フィールドの値が「可」へ切り替えられて、当該ボリュームが登録される。その後、CPU30bは、図10及び図11に係る処理を終了する。

10

【0061】

一方、ステップS206において、ボリュームの追加（使用対象への組み込みと使用状態への移行）を直ちに行うことを指定するラジオボタン74eが選択されていた場合、CPU30bは、ステップS208へ処理を進める。

【0062】

ステップS208では、CPU30bは、ボリューム追加情報を全てのサーバマシン（クラスタノード）20～50へ一斉送信する。なお、ボリューム追加情報は、使用中ボリュームの容量が増加したことを通知するための情報である。

【0063】

次のステップS209では、CPU30bは、全てのサーバマシン20～50から応答があるまで、待機する。そして、全てのサーバマシン20～50から応答があると、CPU30bは、ステップS210へ処理を進める。

20

【0064】

ステップS210では、CPU30bは、指定されたボリュームの追加（使用対象への組み込みと使用状態への移行）を確定する処理を行う。この処理では、図6のボリューム管理テーブル72における当該ボリュームのレコードの「組込」フィールドの値が「済」へ切り替えられて、当該ボリュームが登録される。

【0065】

次のステップS211では、CPU30bは、図8のデフラグ用メタデータサーバマシン40のファイルシステム43に対し、ファイル再配置プログラム43bの起動を指示し、併せて、追加したボリュームのボリューム番号を引数として引き渡す。その後、CPU30bは、図10及び図11に係る処理を終了する。

30

【0066】

このボリュームマネージャ33dによれば、本実施形態のコンピュータネットワークシステムの管理者は、図12に示すようなボリューム設定画面74を通じて、使用対象に対するボリュームの組み込み及び削除を行うことができる。また、ボリュームの組み込みを行う際には、そのボリュームを直ちに使用対象に組み込んで使用状態にするか、それとも、そのボリュームを使用対象に組み込んで使用を留保状態にしておき使用中ボリュームの空き領域の容量が所定値以下になったときにそのボリュームの使用留保を解除するかを、ラジオボタン74e、74eを使って、選択することができる。更に、管理者は、使用中ボリュームが所定の状態になった後でボリュームの追加（使用留保の解除）が行われるよう指定する場合に、ボリューム追加が行われるべき使用中ボリュームの条件と、ファイル再配置を行うか否かとを、テキストボックス74f及びチェックボックス74gを使って、指定することができる。

40

【0067】

なお、ステップS201及びS202を実行するCPU30bは、前述した受付手段及び前述した受付部に相当し、ステップS201及びS202は、前述した受付手順に相当している。

【0068】

また、領域管理用メタデータサーバマシン30では、空き容量監視プログラム33eが

50

、ファイルシステムの起動とともに、又は、実行終了後直ちに、実行されるようになって  
いる。図 1 3 の流れ図に示すように、CPU 3 0 b は、空き容量監視処理の開始後、最初  
のステップ S 3 0 1 では、何れかのサーバマシン 2 0 ~ 5 0 のファイル操作制御機能 2 3  
a、3 3 a、4 3 a から、書き込みのための空き領域の予約の依頼があるまで、待機する  
。そして、空き領域の予約の依頼を何れかのサーバマシン 2 0 ~ 5 0 から受け付けると、  
CPU 3 0 b は、ステップ S 3 0 2 へ処理を進める。

【 0 0 6 9 】

ステップ S 3 0 2 では、CPU 3 0 b は、使用中ボリュームの中から空き領域を確保し  
て、他の予約を受け付けない排他状態になるとともに、使用中ボリュームの全ての空き領  
域の容量を読み出す。なお、空き領域の容量は、例えば、図 6 のボリューム管理テーブル  
7 2 における「組込」フィールドに「済」を記録する全てのレコードの「空き容量」フ  
ィールドの値の総計により、取得しても良い。

10

【 0 0 7 0 】

次のステップ S 3 0 3 では、CPU 3 0 b は、自動追加（使用留保の自動解除）が設定  
されているボリュームが存在するか否かを、判別する。具体的には、CPU 3 0 b は、図  
6 のボリューム管理テーブル 7 2 のレコードのうち、「組込」フィールドの値が「可」で  
あるとともに「自動追加」フィールドの値が「不可」でないレコードが検出できるか否か  
を、判別する。そして、自動追加が設定されているボリュームが存在していなかった場合  
、CPU 3 0 b は、ステップ S 3 0 3 から処理を分岐させ、図 1 3 に係る空き容量監視を  
終了する。一方、自動追加が設定されているボリュームが存在していた場合、CPU 3 0  
b は、ステップ S 3 0 4 へ処理を進める。

20

【 0 0 7 1 】

ステップ S 3 0 4 では、CPU 3 0 b は、自動追加（使用留保の自動解除）が設定され  
ているボリュームについて条件として定義されている閾値を読み出す。すなわち、CPU  
3 0 b は、ステップ S 3 0 3 で検出されたレコードの「自動追加」フィールドの値を読み  
出す。

【 0 0 7 2 】

次のステップ S 3 0 5 では、CPU 3 0 b は、ステップ S 3 0 2 で読み出した空き容量  
が、ステップ S 3 0 4 で読み出した閾値未満であるか否かを、判別する。そして、空き容  
量が閾値以上であった場合、CPU 3 0 b は、ステップ S 3 0 5 から処理を分岐させ、図  
1 3 に係る空き容量監視を終了する。一方、空き領域が閾値未満であった場合、CPU 3  
0 b は、ステップ S 3 0 6 へ処理を進める。

30

【 0 0 7 3 】

ステップ S 3 0 6 では、CPU 3 0 b は、ボリューム追加情報を全てのサーバマシン（  
クラスタノード）2 0 ~ 5 0 へ一斉送信する。なお、ボリューム追加情報は、前述したよ  
うに、使用中ボリュームの容量が増加したことを通知するための情報である。

【 0 0 7 4 】

次のステップ S 3 0 7 では、CPU 3 0 b は、全てのサーバマシン 2 0 ~ 5 0 から応答  
があるまで、待機する。そして、全てのサーバマシン 2 0 ~ 5 0 から応答があると、C P  
U 3 0 b は、ステップ S 3 0 8 へ処理を進める。

40

【 0 0 7 5 】

ステップ S 3 0 8 では、CPU 3 0 b は、指定されたボリュームの追加（使用留保の解  
除）を確定する処理を行う。この処理では、図 6 のボリューム管理テーブル 7 2 における  
当該ボリュームのレコードの「組込」フィールドの値が「済」へ切り替えられる。

【 0 0 7 6 】

次のステップ S 3 0 9 では、CPU 3 0 b は、図 8 のデフラグ用メタデータサーバマシ  
ン 4 0 のファイルシステム 4 3 に対し、ファイル再配置プログラム 4 3 b の起動を指示し  
、併せて、追加したボリュームのボリューム番号を引数として引き渡す。その後、CPU  
3 0 b は、図 1 3 に係る空き容量監視を終了する。

【 0 0 7 7 】

50

この空き容量監視プログラム 33e によれば、何れかのサーバマシン 20 ~ 50 のファイルシステムがアプリケーション 21、31、41 からファイルの書き込み要求を受けると、自動追加設定（使用留保）がなされているボリュームが探され（ステップ S303）、そのようなボリュームが探し出されれば、そのボリュームについて定義されている条件に基づいて、使用中ボリュームの空き容量が所定の状態になっているか否かが判別され（ステップ S302、S304、S305）、使用中ボリュームの空き容量が所定の状態になっている場合に、自動追加設定がなされているボリュームが使用中ボリュームに追加される（ステップ S306 ~ S309）。

#### 【0078】

以上の二つの処理のうちの何れかによる指示、すなわち、図 4 の領域管理用メタデータサーバマシン 30 のボリュームマネージャ 33d による処理（図 10 及び図 11）のステップ S211 による指示、又は、空き容量監視プログラムによる処理（図 13）のステップ S309 による指示に基づいて、図 8 のデフラグ用メタデータサーバマシン 40 の CPU 40b は、ファイル再配置プログラム 43b を実行する。図 14 の流れ図に示すように、CPU 40b は、ファイル再配置処理の開始後、最初のステップ S401 において、ボリューム追加に続いてファイルの再配置を行うよう設定されているか否かを、判別する。すなわち、CPU 40b は、図 6 のボリューム管理テーブル 72 において、起動指示とともに引数として受け取ったボリューム番号のボリュームのレコードの「再配置」フィールドから値を読み出し、その値が「要」であるか「不要」であるかを、判別する。そして、ファイルの再配置を行うよう設定されていなかった場合、CPU 40b は、ステップ S401 から処理を分岐させ、図 14 に係るファイル再配置処理を終了する。一方、ファイルの再配置を行うよう設定されていた場合、CPU 40b は、ステップ S402 へ処理を進める。

#### 【0079】

ステップ S402 では、CPU 40b は、図 7 の断片化ファイル情報管理テーブル 73 に登録されているファイルを、一つずつ、順に、追加されたボリュームへ移動する処理を行う。具体的には、CPU 40b は、一つ以上のファイルの中から処理対象として選択された一つのファイルについて、まず、そのファイルを構成する単位データが記録されているディスクブロックからその単位データをメモリユニット 40c へ複製する処理を、実行する。続いて、CPU 40b は、メモリユニット 40c に複製した単位データを、追加されたボリュームにおけるアドレスの若いブロックから古いブロックへと順に、記録していく。その後、CPU 40b は、処理対象のファイルを構成する単位データが元々記録されていたボリューム（ディスク）から、そのファイルのデータを削除する処理を行う。CPU 40b は、図 7 の断片化ファイル情報管理テーブル 73 に登録されている全てのファイルについて、上記の処理を終えると、図 7 の断片化ファイル情報管理テーブル 73 から、これらファイルに係るレコードを削除し、その後、ステップ S403 へ処理を進める。

#### 【0080】

ステップ S403 では、CPU 40b は、図 3 の領域管理用メタデータサーバマシン 30 のファイルシステム 33 内の領域管理機能 33b に対し、ステップ S402 で再配置されたファイルについての領域情報の更新を依頼する。なお、領域管理機能 33b は、この依頼を受けて、図 5 のアイノード管理テーブル 71 における該当ファイルに関する情報を更新する。CPU 40b は、このような領域情報の更新を領域管理機能 33b に依頼すると、図 14 に係るファイル再配置処理を終了する。

#### 【0081】

このファイル再配置プログラム 43b によれば、領域管理用メタデータサーバマシン 30 からの依頼（ステップ S211 又は S309）があると、既存ディスク内にある断片化ファイルが、順に、追加されたディスクへと、移し替えられる。

#### 【0082】

このように、既存ディスク内で断片化状態にあったファイルが、追加されるディスク内へ移し替えられると、当該ファイルは、連続したブロックに纏めて書き込まれることとな

10

20

30

40

50

る。このため、ディスクの空き容量が足りなくても、ファイルが効率的に再配置されるようになる。

【0083】

なお、ステップS206乃至S211、S301乃至S309、及び、S401乃至S403を実行するCPU30bは、前述した移動手段及び前述した移動部に相当し、ステップS206乃至S211、S301乃至S309、及び、S401乃至S403は、前述した移動手順に相当している。

【図面の簡単な説明】

【0084】

【図1】本実施形態のコンピュータネットワークシステムのネットワーク構成図

10

【図2】ストレージマシンの構成図

【図3】名前管理用メタデータサーバマシンの構成図

【図4】領域管理用メタデータサーバマシンの構成図

【図5】アイノード管理テーブルの模式的なテーブル構造図

【図6】ボリューム管理テーブルの模式的なテーブル構造図

【図7】断片化ファイル情報管理テーブルの模式的なテーブル構造図

【図8】デフラグ用メタデータサーバマシンの構成図

【図9】フラグメント情報収集プログラムによる処理の流れを示す図

【図10】ボリュームマネージャによる処理の流れを示す図

【図11】ボリュームマネージャによる処理の流れを示す図

20

【図12】ボリューム設定画面の画面例を示す図

【図13】空き容量監視プログラムによる処理の流れを示す図

【図14】ファイル再配置プログラムによる処理の流れを示す図

【符号の説明】

【0085】

10 ストレージマシン

10a ディスクユニット

10b ストレージコントローラ

20 名前管理用メタデータサーバマシン

20a ストレージユニット

30

20b CPU

23 ファイルシステム

23a ファイル操作制御機能プログラム

23b 名前管理機能プログラム

30 領域管理用メタデータサーバマシン

30a ストレージユニット

30b CPU

33 ファイルシステム

33a ファイル操作制御機能プログラム

33b 領域管理機能プログラム

40

33c フラグメント情報収集プログラム

33d ボリュームマネージャ

33e 空き容量監視プログラム

40 デフラグ用メタデータサーバマシン

40a ストレージユニット

40b CPU

43 ファイルシステム

43a ファイル操作制御機能プログラム

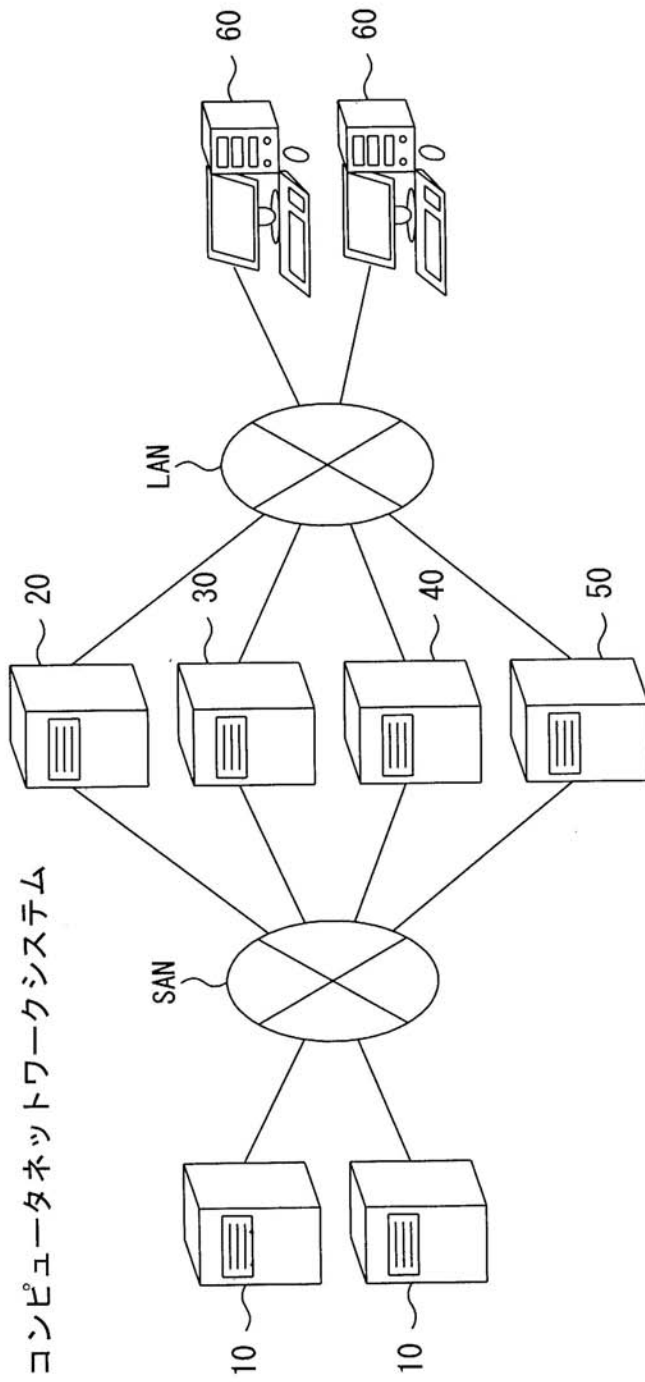
43b ファイル再配置プログラム

50 汎用サーバマシン

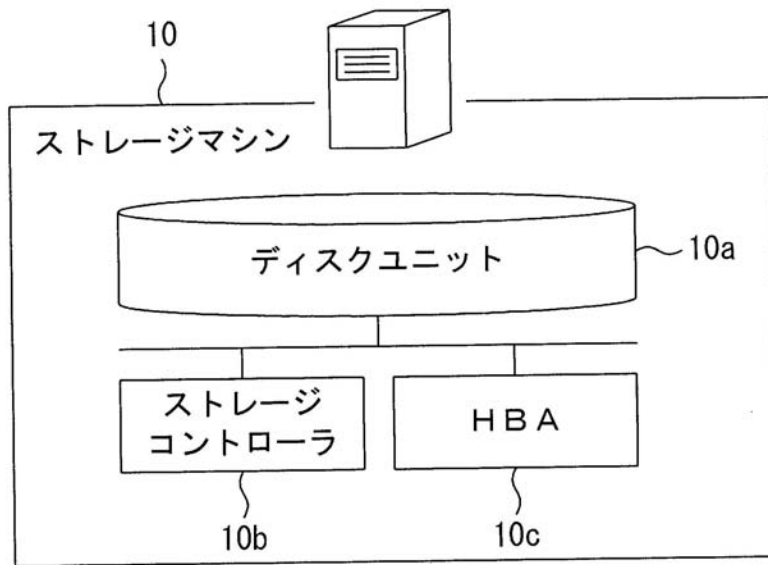
50

- 6 0 クライアントマシン
- 7 1 アイノード管理テーブル
- 7 2 ボリューム管理テーブル
- 7 3 断片化ファイル情報管理テーブル
- 7 4 ボリューム設定画面

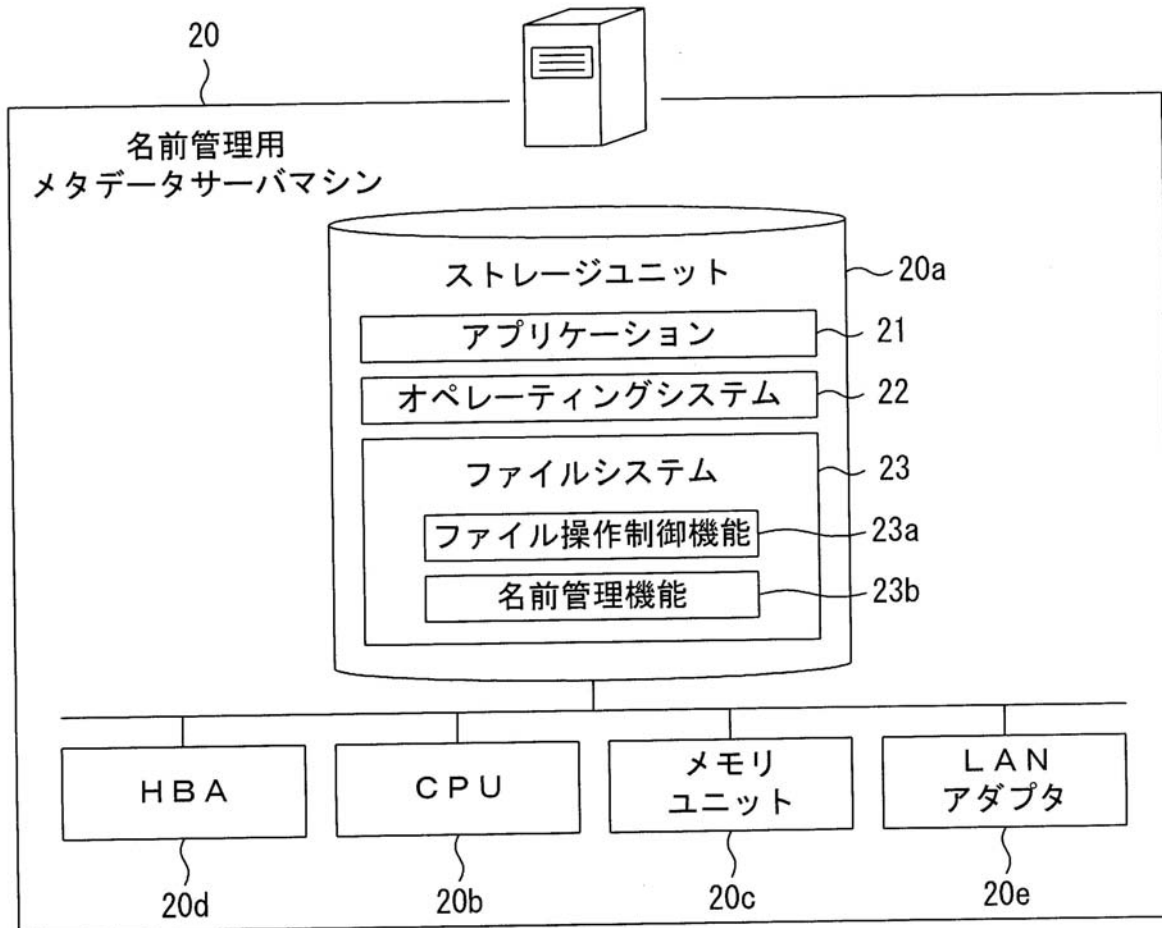
【 図 1 】



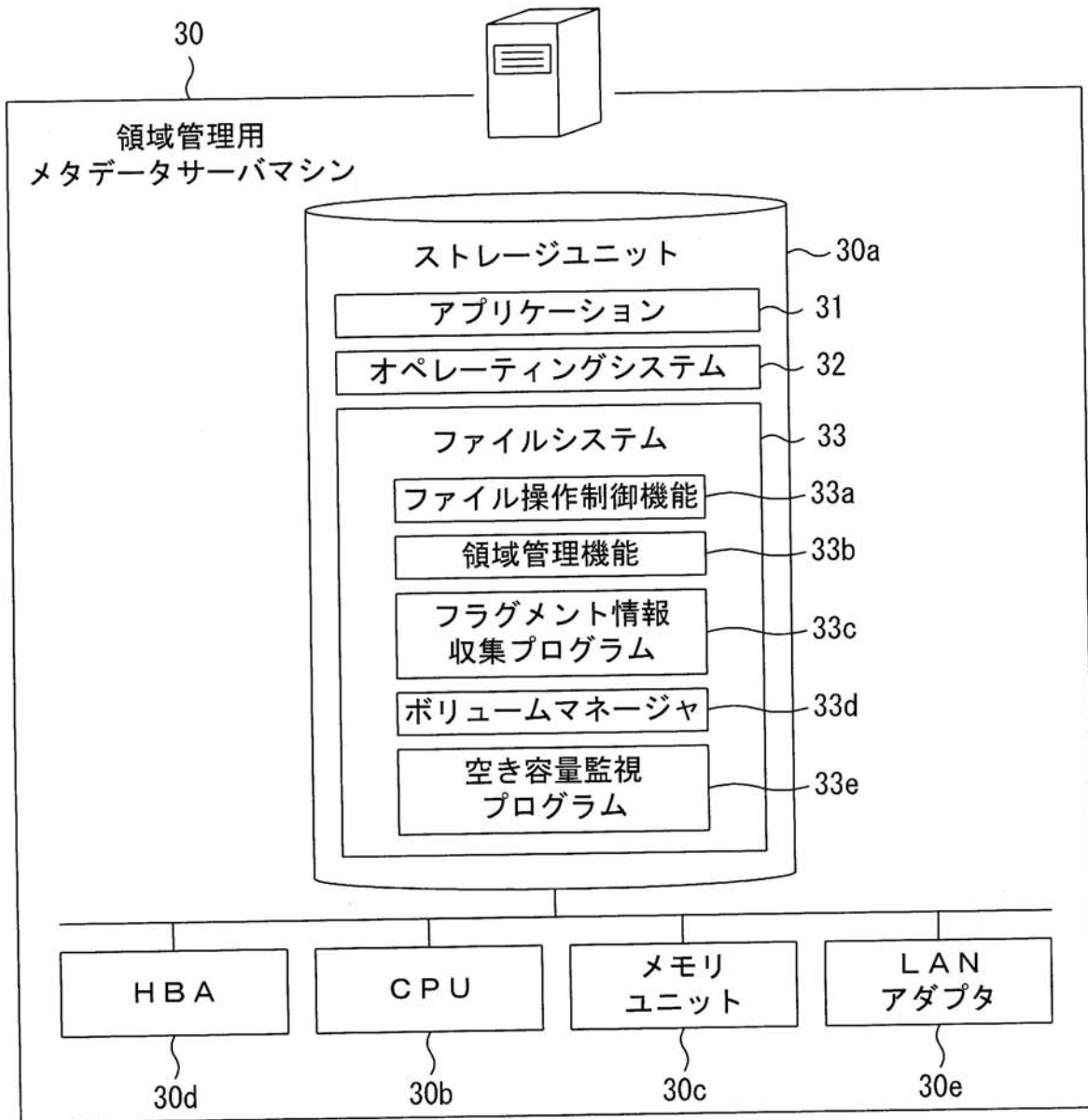
【図2】



【図3】



【図4】



71: アイノード管理テーブル

アイノード	サイズ	ブロック数	エクステンント数	最終アクセス日時	アクセス数	ブロック
∴	∴	∴	∴	∴	∴	∴
1226	79275349	9678	645	2006/11/17 21:19:35	162	略
1227	4173312	510	2	2006/11/19 03:58:02	938	略
1228	1459200	179	8	2006/08/14 15:47:25	3	略
∴	∴	∴	∴	∴	∴	∴

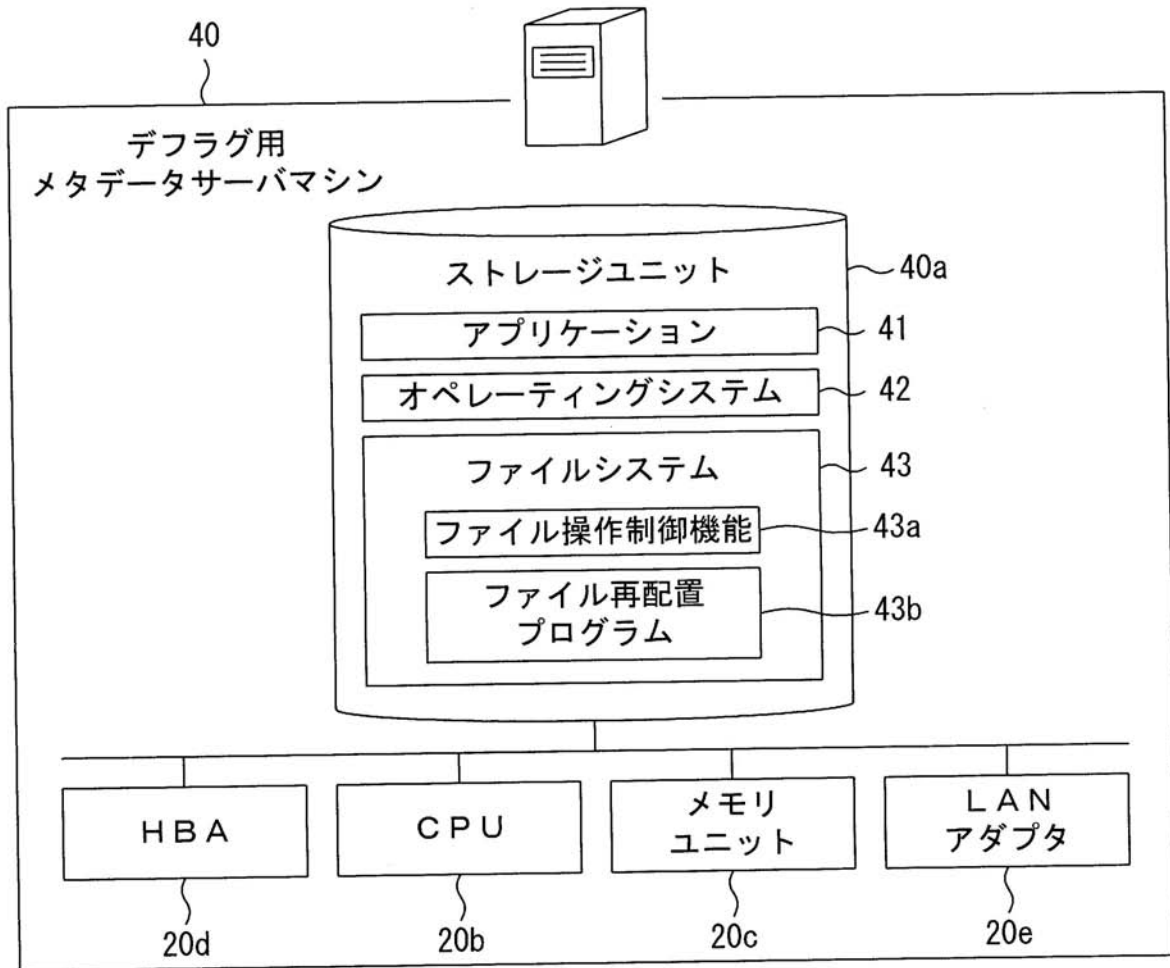
72:ポリューム管理テーブル

ポリューム	ポリューム名	サイズ	空き容量	組込	自動追加	再配置
1	/dev/sfdsk/gfs/dsk/volume01	256GB	3GB	済	不可	不要
2	/dev/sfdsk/gfs/dsk/volume02	256GB	47GB	済	80GB	不要
3	/dev/sfdsk/gfs/dsk/volume03	512GB	18GB	済	60GB	要
4	/dev/sfdsk/gfs/dsk/volume04	512GB	512GB	可	40GB	要
5	/dev/sfdsk/gfs/dsk/volume05	512GB	512GB	未	-	-
6	/dev/sfdsk/gfs/dsk/volume06	512GB	512GB	未	-	-
7	/dev/sfdsk/gfs/dsk/volume07	512GB	512GB	未	-	-

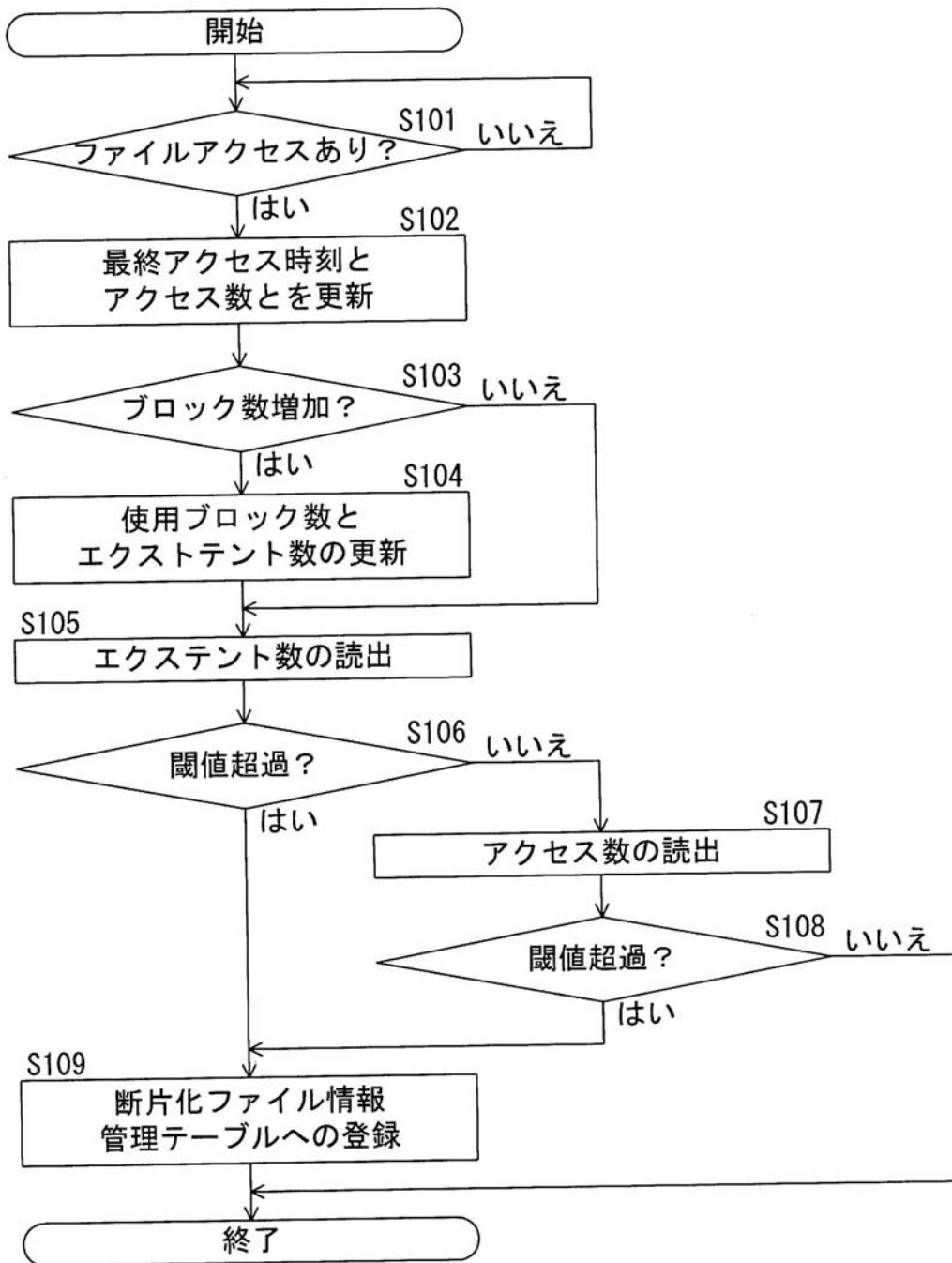
73: 断片化ファイル情報管理テーブル

アイノード	サイズ	ブロック数	エクステンション数	最終アクセス日時	アクセス数
1226	79275349	9678	645	2006/11/17 21:19:35	162
2212	9377808384	1144752	39474	2006/11/19 18:02:11	238
9386	527945728	64446	3222	2006/11/15 05:31:48	58
∴	∴	∴	∴	∴	∴

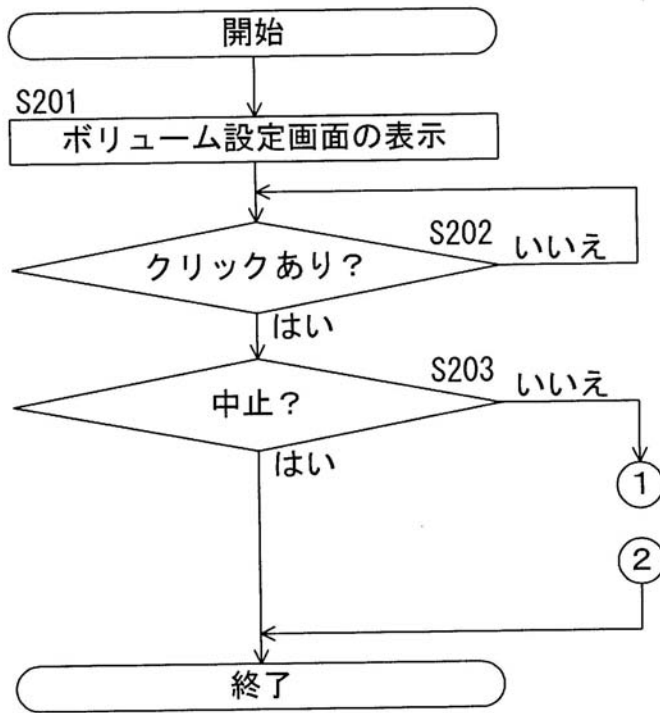
【 図 8 】



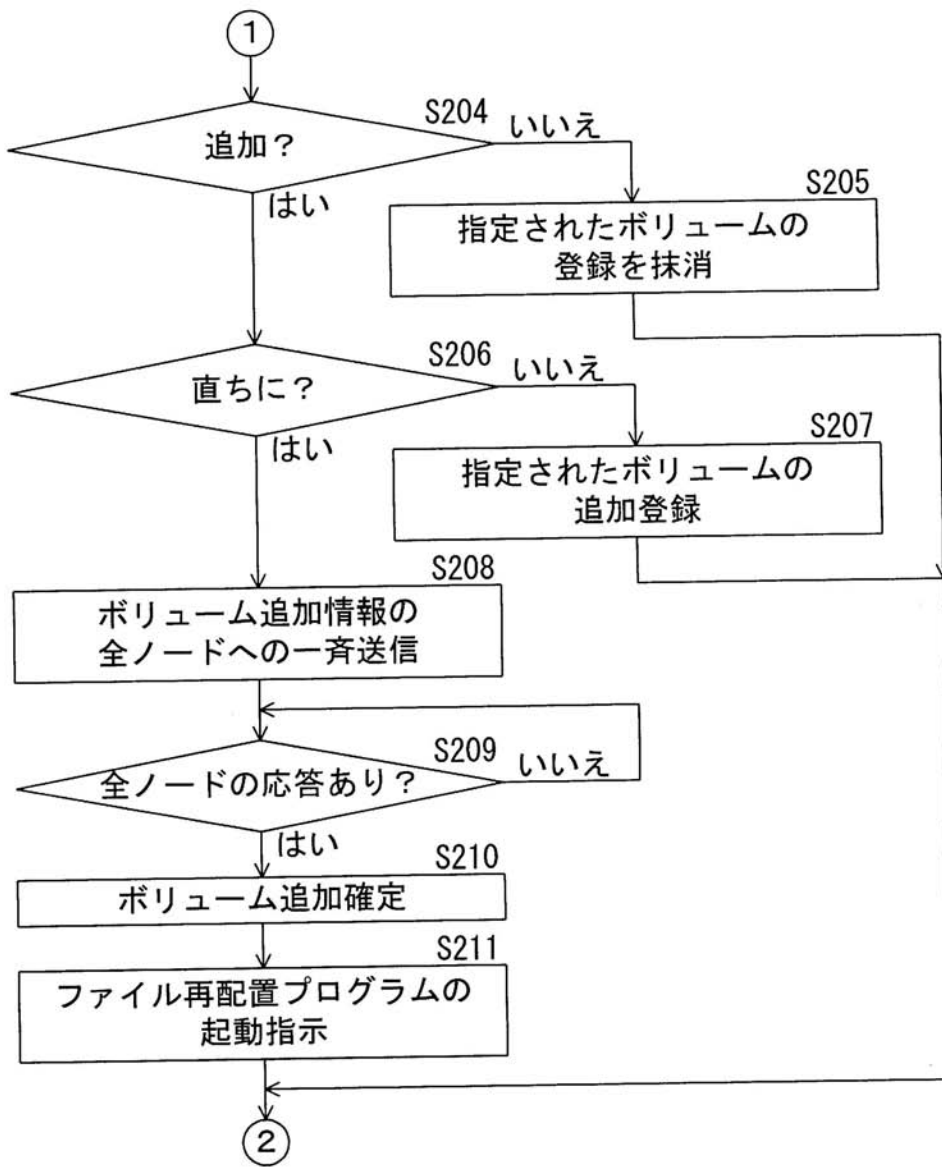
【図9】



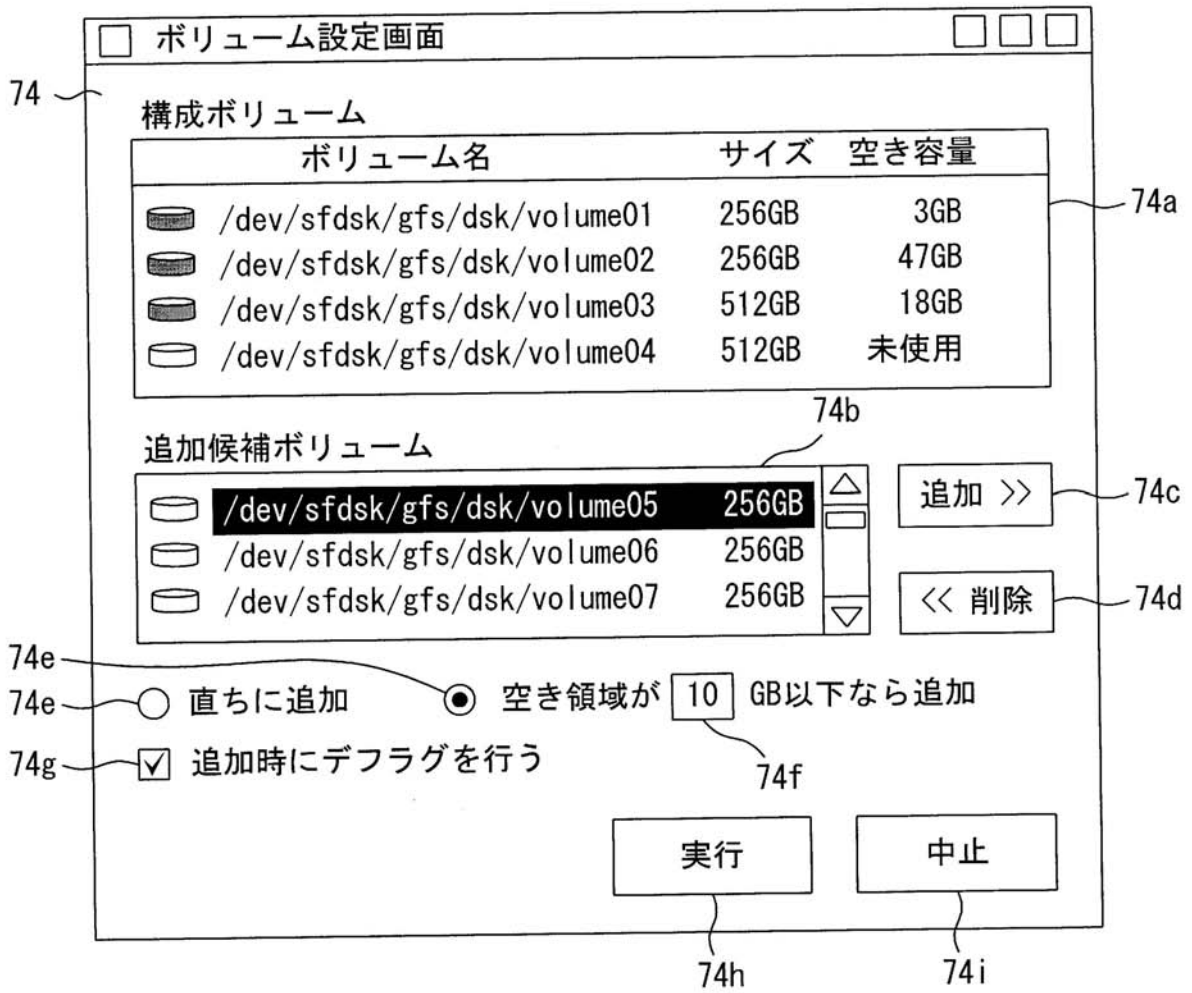
【図10】



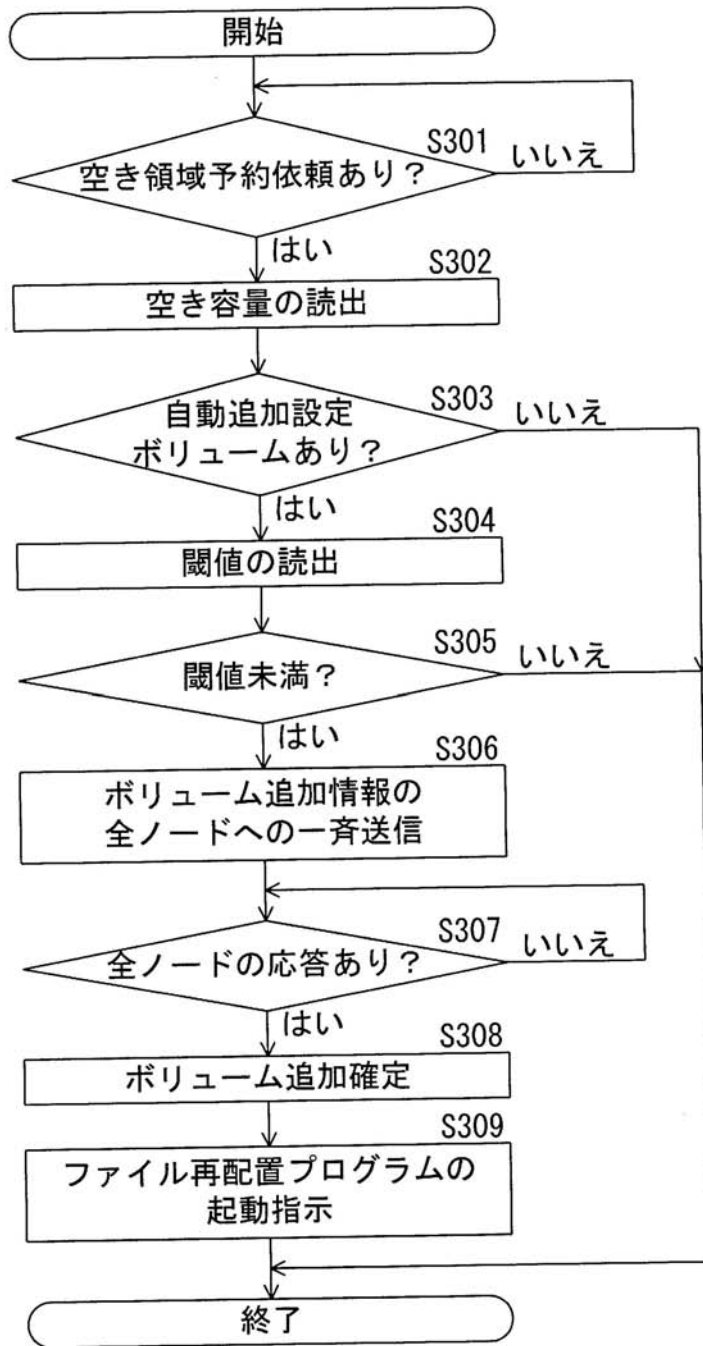
【図 1 1】



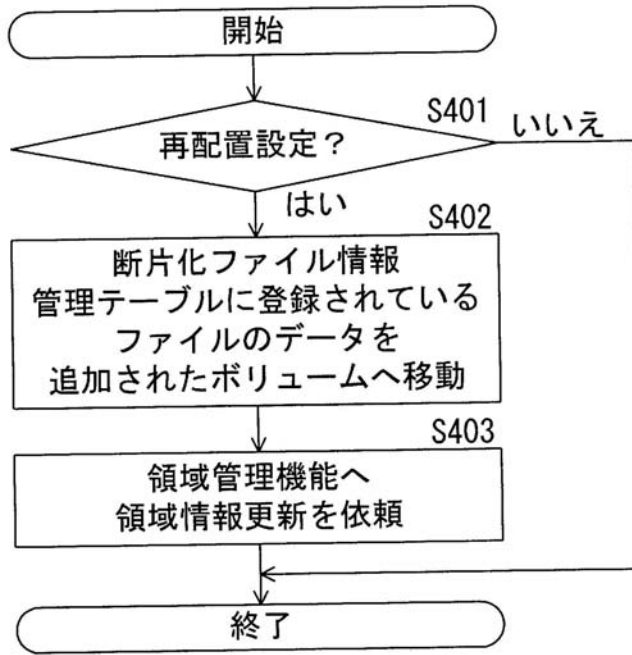
【図 1 2】



【図13】



【 図 1 4 】



フロントページの続き

Fターム(参考) 5B065 BA01 ZA02 ZA16  
5B082 CA11