

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4755642号
(P4755642)

(45) 発行日 平成23年8月24日(2011.8.24)

(24) 登録日 平成23年6月3日(2011.6.3)

(51) Int.Cl. F 1
G 0 6 F 1 2 / 0 0 (2 0 0 6 . 0 1) G 0 6 F 1 2 / 0 0 5 1 1 A

請求項の数 10 (全 24 頁)

(21) 出願番号	特願2007-510232 (P2007-510232)	(73) 特許権者	506358867
(86) (22) 出願日	平成17年4月21日 (2005.4.21)		ストアウィズ インク
(65) 公表番号	特表2007-535068 (P2007-535068A)		アメリカ合衆国 20878 メリーラン
(43) 公表日	平成19年11月29日 (2007.11.29)		ド州 ゲイザースバーグ 260 ファー
(86) 国際出願番号	PCT/IL2005/000419		ストフィールド ロード 스위트 21
(87) 国際公開番号	W02005/103878	(74) 代理人	100105050
(87) 国際公開日	平成17年11月3日 (2005.11.3)		弁理士 鷲田 公一
審査請求日	平成20年4月17日 (2008.4.17)	(72) 発明者	セーバー ギル
(31) 優先権主張番号	60/565,298		イスラエル国 48621 ロシュ ハア
(32) 優先日	平成16年4月26日 (2004.4.26)		イン レビビム ストリート 76
(33) 優先権主張国	米国 (US)	(72) 発明者	アマット ノーチェ
			イスラエル国 34334 ハイファ デ
			イスラエリ ストリート 20エー

最終頁に続く

(54) 【発明の名称】 記憶のためのファイル圧縮および圧縮ファイルの操作の方法およびシステム

(57) 【特許請求の範囲】

【請求項1】

少なくとも一つのファイルアクセスストレージプロトコルにより動作する記憶装置において記憶するために圧縮ファイルを作成し、前記圧縮ファイルに対して所与のデータ範囲でデータを書き込む方法であって、

a) 原ファイルを圧縮して圧縮データを生成するステップであって、前記原ファイルのデータはクラスタとして1個以上の固定サイズを有する第1のクラスタおよび1個または0個の前記固定サイズに満たないサイズを有する第2のクラスタを含み、

前記第1のクラスタおよび第2のクラスタが順に処理された結果、前記第1のクラスタおよび第2のクラスタのそれぞれが対応する圧縮セクションに圧縮され、前記各圧縮セクションは1個以上の固定サイズを有する固定サイズ圧縮論理ユニット(CLU)に分割されて記憶されるステップと、

b) 前記圧縮ファイルとしての前記圧縮データの記憶を促進するステップであって、前記圧縮ファイルは、前記原ファイルのクラスタに対応する圧縮セクションと、対応する前記原ファイルのサイズについての情報を保持するヘッダとを含むステップと、

c) 前記圧縮セクションに関連する少なくとも一つのレコードを有するセクションテーブルを作成するステップであって、前記レコードは、前記圧縮セクションに対応するCLUに関する情報と、前記CLUに関連する記憶場所ポインタとを少なくとも保持するステップとを、備え、

a1) 更新されるべきデータを有し元の圧縮セクションを生じさせる第1の圧縮セクシ

10

20

ヨンのシリアル番号を判定するステップと、

b 1) セクションテーブルを参照して前記元の圧縮セクションに対応する C L U およびその記憶場所を判定するステップと、

c 1) 前記元の圧縮セクションからのクラスタの復元を促進するステップと、

d 1) 前記クラスタ内の更新されるべきデータのオフセットを計算し、前記更新を促進するステップと、

e 1) 前記更新された圧縮セクションに前記更新されたクラスタを圧縮するステップと

f 1) 前記更新された圧縮セクションで前記元の圧縮セクションへの上書きを促進するステップと、

g 1) セクションテーブルを更新するステップと、

h 1) 書き込まれるべきデータの範囲が復元されたクラスタを超える場合には、書き込むべき必要なデータ全てが復元されるまでステップ b 1) 乃至 g 1) を、シリアル番号を 1 だけインクリメントした圧縮セクションに対して繰り返すステップと、

i 1) 前記圧縮ファイルへのデータの書き込み中に解放された自由 C L U のリストを扱うステップであって、前記リストは、ファイルが閉じられるまでファイルに関連する全てのセッション中に扱われるステップと、

a 2) ファイルを閉じる前に自由 C L U のリストを確認し、最も低い記憶場所ポインタを有する自由 C L U を特定するステップと、

b 2) 前記リストが空でない場合には、前記圧縮セクションに含まれる C L U の中の最も高い記憶場所ポインタを有する C L U を定め、第 1 の C L U を生じさせるステップと、

c 2) 前記第 1 の C L U から前記最も低い記憶場所ポインタを有する自由 C L U への圧縮データの移動を促進し、第 2 の C L U を生じさせるステップと、

d 2) 前記第 2 の C L U を属する圧縮セクションに割り当て、前記第 1 の C L U を自由 C L U のリストに割り当てるステップと、

e 2) 圧縮セクションに含まれる全ての C L U の記憶場所ポインタが自由 C L U のリストに含まれるどの C L U のポインタよりも低くなるまでステップ b) - d) を繰り返すステップと、

f 2) セクションテーブルを更新するステップと、を備える方法。

【請求項 2】

前記原ファイルのサイズを、対応する蓄積された前記圧縮ファイルのサイズに言及する要求に回答して報告するステップを更に備える、請求項 1 記載の方法。

【請求項 3】

少なくとも一つのクラスタの圧縮は、処理中に評価された取得可能な圧縮比が所定の基準を満たす場合にだけ与えられる、請求項 1 または請求項 2 記載の方法。

【請求項 4】

前記圧縮ファイルは、対応する前記クラスタの非圧縮データを収容する少なくとも一つの圧縮セクションを含む、請求項 1 から請求項 3 のいずれかに記載の方法。

【請求項 5】

前記圧縮セクションを記述するセクションテーブルにおけるレコードのうち少なくとも一つのレコードは、前記圧縮セクションが圧縮データを含むかを示す情報を保持する、請求項 1 から請求項 4 のいずれかに記載の方法。

【請求項 6】

前記元のセクションに対応する C L U の数 N_0 と前記更新された圧縮セクションに対応する C L U の数 N_u を比較するステップと、

a) $N_0 = N_u$ の場合、更新された圧縮セクションに対応する C L U で元の圧縮セクションに対応する全ての C L U を上書きすること、

b) $N_0 > N_u$ の場合、更新された圧縮セクションに対応する C L U で元の圧縮セクションに対応する N_u の C L U を上書きし、解放される C L U に関する自由 C L U のリストを更新すること、

10

20

30

40

50

c) $N_0 < N_u < N_0 + N_f$ (N_f は自由 C L U のリスト中の C L U の数) の場合、更新された圧縮セクションに対応する C L U で元の圧縮セクションに対応する全ての C L U を上書きし、更新された圧縮セクションに対応する残りの C L U を自由 C L U のリストに含まれる C L U に書き込むこと、

d) $N_0 + N_f < N_u$ の場合、更新された圧縮セクションに対応する C L U で元の圧縮セクションに対応する全ての C L U を上書きし、更新された圧縮セクションに対応する C L U を自由 C L U のリストに含まれる C L U に書き込み、次の自由な記憶場所に更新された圧縮セクションに対応する残りの C L U を連続的に書き込むこと、

のいずれかを促進するステップを更に備える、請求項 1 から請求項 5 のいずれかに記載の方法。

10

【請求項 7】

ストレージネットワークにおいて動作可能であり、請求項 1 から請求項 6 のいずれかの方法のステップを実行するよう構成された通信装置または記憶装置。

【請求項 8】

記憶するためにファイルを圧縮するシステムであって、

a) 原ファイルを圧縮し圧縮データを生成するよう構成されたサブシステムであって、前記原ファイルのデータはクラスタとして 1 個以上の固定サイズを有する第 1 のクラスタおよび 1 個または 0 個の前記固定サイズに満たないサイズを有する第 2 のクラスタを含み

前記第 1 のクラスタおよび第 2 のクラスタが順に処理された結果、前記第 1 のクラスタおよび第 2 のクラスタのそれぞれが対応する圧縮セクションに圧縮され、前記各圧縮セクションは 1 個以上の固定サイズを有する固定サイズ圧縮論理ユニット (C L U) に分割されて記憶されるよう構成されたサブシステムと、

20

b) 圧縮ファイルとしての圧縮データの記憶を促進するよう構成されたサブシステムであって、前記圧縮ファイルは、前記原ファイルのクラスタに対応する一つ以上の圧縮セクションと、対応する前記原ファイルのサイズについての情報を保持するヘッダとを含むサブシステムと、

c) 前記圧縮セクションを記述する少なくとも一つのレコードを有するセクションテーブルを作成するよう構成されたサブシステムであって、前記レコードは、圧縮セクションに対応する C L U と、前記 C L U に関連する記憶場所ポインタに関する情報とを保持するサブシステムと、を備え、

30

a 1) 更新されるべきデータを有し元の圧縮セクションを生じさせる第 1 の圧縮セクションのシリアル番号を判定する動作と、

b 1) セクションテーブルを参照して前記元の圧縮セクションに対応する C L U およびその記憶場所を判定する動作と、

c 1) 前記元の圧縮セクションからのクラスタの復元を促進する動作と、

d 1) 前記クラスタ内の更新されるべきデータのオフセットを計算し、前記更新を促進する動作と、

e 1) 前記更新された圧縮セクションに前記更新されたクラスタを圧縮する動作と、

f 1) 前記更新された圧縮セクションで前記元の圧縮セクションへの上書きを促進する動作と、

40

g 1) セクションテーブルを更新する動作と、

h 1) 書き込まれるべきデータの範囲が復元されたクラスタを超える場合には、書き込むべき必要なデータ全てが復元されるまで動作 b 1) 乃至 g 1) を、シリアル番号を 1 だけインクリメントした圧縮セクションに対して繰り返す動作と、

i 1) 前記圧縮ファイルへのデータの書き込み中に解放された自由 C L U のリストを扱う動作であって、前記リストは、ファイルが閉じられるまでファイルに関連する全てのセッション中に扱われる動作と、

a 2) ファイルを閉じる前に自由 C L U のリストを確認し、最も低い記憶場所ポインタを有する自由 C L U を特定する動作と、

50

b 2) 前記リストが空でない場合には、前記圧縮セクションに含まれるＣＬＵの中の最も高い記憶場所ポインタを有するＣＬＵを定め、第１のＣＬＵを生じさせる動作と、

c 2) 前記第１のＣＬＵから前記最も低い記憶場所ポインタを有する自由ＣＬＵへの圧縮データの移動を促進し、第２のＣＬＵを生じさせる動作と、

d 2) 前記第２のＣＬＵを属する圧縮セクションに割り当て、前記第１のＣＬＵを自由ＣＬＵのリストに割り当てる動作と、

e 2) 圧縮セクションに含まれる全てのＣＬＵの記憶場所ポインタが自由ＣＬＵのリストに含まれるどのＣＬＵのポインタよりも低くなるまで動作 b) - d) を繰り返す動作と

f 2) セクションテーブルを更新する動作と、を行うように構成されたサブシステムと、をさらに備えるシステム。

10

【請求項 9】

記憶された圧縮ファイルのサイズに言及する要求において介入し、前記要求に応答して当該原ファイルのサイズを報告し得るよう更に構成された請求項 8 記載のシステム。

【請求項 10】

コンピュータプログラムであって、

前記プログラムがコンピュータ上で実行され、および/または、コンピュータ可読媒体上で具現されたときに、請求項 1 から請求項 6 のいずれかにおけるすべてのステップを実行するためのコンピュータプログラムコード手段を有する、コンピュータプログラム。

【発明の詳細な説明】

20

【技術分野】

【0001】

本発明は、記憶データへのネットワークアクセスをサポートするネットワークインフラストラクチャデバイスに関し、特に、ファイルシステムプロトコルによるファイルシステムでの圧縮データの圧縮および操作を促進する方法および装置に関する。

【背景技術】

【0002】

現在のビジネス環境では、全てのタイプのビジネスデータがビジネスの成功にとって益々重要になってきている。ビジネスにより生成される、情報資産の共有、管理および保護の方法を定めたデータの著しい増加および複雑性が、情報ストレージの需要を後押しして

30

【0003】

典型的には、どの技術も構造も全ての組織の全ての要求に取り組むことはできない。主なストレージ技術は、EMC による白書（非特許文献 1）に記載され、基本的には、接続のタイプ（直接取付型ストレージ（DAS）、IP またはチャネルネットワーク）によって、また、データがアクセスされる方法によって識別される。データアクセスの方法と関連して、ブロックアクセス、ファイルアクセスおよびオブジェクトアクセスといったストレージデバイスの構造の三つの基本的なタイプを考える。

【0004】

ブロックアクセス構造では、サーバ/クライアントとストレージデバイスとの間の通信は、ブロックで実行され、つまり、情報はディスクからブロック毎に直接的に引き出される。オペレーションシステムは、各情報がディスク上のどこにあるかを把握している一方、ストレージデバイスは通常、デバイス上でデータを整理するために使用されるファイルシステムを認識していない。読み出しまたは書き込みが必要となると、ディスク上のどこにデータの各ブロックが存在し、どのようにしてひとまとめにするかを知っているプロセッサによってディスクのデータは直接的にアクセスされる。ブロックアクセスストレージ技術の例として DAS（直接取付型ストレージ）、SAN（ストレージ領域ネットワーク）およびブロックストレージオーバー IP（例えば、FCIP、iFCP、iSCSI 等）が挙げられる。

40

【0005】

50

ファイルアクセスでは、サーバまたはクライアントは物理的な場所ではなく名前でファイルを要求することが必要である。その結果、ストレージデバイスは、通常、ファイルシステムを作成、維持および更新するためにファイルをデータのブロックにマッピングすることを担う。ファイルサーバは、ファイルを読み出しまたは書き込む要求を受信し、「密かに」ブロックアクセスを行う。ファイルアクセスストレージ技術の例として、NAS (NFS、CIFS、HTTP等のプロトコルによるネットワーク接続ストレージ) およびMPFS (マルチパスファイルサービング) が挙げられる。ファイルアクセスストレージは、例えば、汎用ファイル、ウェブアプリケーション、エンジニアリングアプリケーション (例えば、CAD、CAM、ソフトウェア開発等)、画像および3Dデータ処理、マルチメディアストリーミング等のために実行される。

10

【0006】

オブジェクトアクセスは、ブロック、ファイルおよびストレージのトポロジーに関する全ての詳細をアプリケーションから隠すことによってデータアクセスを更に簡略化する。オブジェクトアクセスは、コンテンツ管理アプリケーションで一体化されるAPI上で実行される。オブジェクトアクセスストレージ技術の例としてCAS (コンテンツアドレスストレージ) が挙げられる。

【0007】

ストレージデバイスのより効率的な使用は、データを記憶する前に圧縮することで実現される。データ圧縮技法は、ストレージデバイスの容量および送信時間をそれぞれ減少させるよう記憶されまたは送信されるべきデータ量を減少させるために使用される。圧縮は、例えば、非特許文献2に記載されている標準的な圧縮アルゴリズム等の様々な圧縮アルゴリズムを用いて実現される。透過的に圧縮を実行すること、つまり、データが既存のアプリケーションを変更することなく使用されることが重要である。いずれの場合においても、元のデータが再構成され、アプリケーションにアクセス可能となるよう対応する解凍技法を提供することが必要である。圧縮データを更新する場合、特に、データの比較的小部分を更新する場合、ブロックまたはファイル全体を解凍し再圧縮することは一般的に非効率的である。

20

【0008】

記憶および記憶データのアクセスの最適化の様々な実行は、例えば、以下の特許公報に開示される。

30

【0009】

特許文献1 (Franaszek) は、メモリの断片化を減少する一方でメモリの再編の必要性を回避するように可変長のオブジェクトを記憶するシステムおよび方法を開示する。可変長オブジェクトの残余は、最良適合または第一適合ベースで別の可変長オブジェクトからの残余とストレージの固定サイズのブロックを共有するよう割り当てられる (ブロックを共有する二つの残余はルームメイトと称される)。一方の残余がブロックの一方の端部に記憶され、他方の残余はブロックの他方の端部に記憶される。ストレージのブロックを共有する可変長オブジェクトは、同じ集団 (cohort) から選択される。それにより、オブジェクト間では何らかの関連性がある。この関連性は、例えば、オブジェクトが同じページからまたは多数のページにわたって何らかの線形順序にあることを含む。ある集団の可変長オブジェクトに関する情報、例えば、オブジェクトがルームメイトを有するか否かは、メモリに記憶される。

40

【0010】

特許文献2 (Yoshida et al.) は、圧縮データを記憶する方法および装置を開示する。ここでは、圧縮ファイルは、圧縮管理テーブルの位置を示す情報を有するヘッダと、圧縮されたコードと、元のレコードそれぞれの圧縮されたコードの記憶場所を示す情報を保持する圧縮管理表とよりなる。

【0011】

特許文献3 (Morris et al.) は、バックアップサブシステムにおけるストレージ要件を減少させ、セグメント化された圧縮フォーマットにおいてサーバ上で基本ファイルが維

50

持される送信帯域幅に対する負荷を減少させる方法および手段を開示する。ファイルがクライアントで変更されると、ファイルはサーバに送信され、基本ファイル全体を解凍することなく、差分関数を用いてファイルのセグメント化された圧縮基本バージョンと比較される。圧縮基本ファイルとファイルの変更バージョンとの間の差であるデルタファイルが作成され、バックアップシステムの一部である記憶媒体に記憶される。

【 0 0 1 2 】

特許文献 4 (Bolan et al.) は、システムストレージ階層において区別可能な場所で圧縮データがシステム管理されるリソースとなるよう、システムの時系列パラメータに基づいてデータの圧縮および解凍を制御するシステムを開示する。プロセッサレジスタはキャッシュにより支持され、キャッシュはメインストレージにより支持され、メインストレージは解凍ディスクストレージにより支持され、解凍ディスクストレージは圧縮ディスクストレージ、テープ等によって支持される。データは、解凍形態から圧縮形態に移行し、システムの時系列パラメータに基づいてデータライフサイクルに応じてシステム制御下でストレージ階層を通じて移され、または、任意には、要求に応じて、データは最初に作成され記憶され、データはシステム制御下で後に圧縮され、データがアクセスされると、要求に応じてセグメント毎に解凍され、その幾らか後に、データは次に参照されるまでシステム制御下で再び圧縮される。大きいデータオブジェクトはセグメント化され、圧縮はより使用頻度が低いデータに適用される。

10

【 0 0 1 3 】

特許文献 5 (Obara et al.) は、ディスクストレージシステムを開示する。ここでは、キャッシュメモリに記憶されるべきデータは、データが所属するトラックブロックと関連付けられた二つのキャッシュブロックをそれぞれが有する複数のデータブロックに分割されて圧縮され、それにより、読みやすい方法でディスクストレージシステムのキャッシュメモリに複数の圧縮レコードが記憶される。圧縮後のそれぞれのデータブロックは、一つのまたは複数のキャッシュブロックに記憶される。データブロックに対するイントラックアドレスから各キャッシュブロックを取り出す情報は、キャッシュメモリに対する取り出し情報の一部として記憶される。レコードにおけるそれぞれのデータブロックが読み出されると、圧縮データブロックを記憶するキャッシュブロックは、データブロックのイントラックアドレスおよび取り出し情報に基づいて判定される。

20

【 0 0 1 4 】

特許文献 6 (Faulkner et al.) は、仮想メモリシステムを用いたデータ圧縮および解凍の組み合わせを開示する。データが圧縮フォーマットでストレージデバイスに記憶されるいわゆる、埋め込み型システムを含む幾つかのコンピュータシステムが記載されている。中央処理ユニット (CPU) によるデータに対する要求に応答して、仮想メモリシステムはまず、要求されたデータが、解凍データが記憶されている部分でもあり CPU がアクセス可能であるメインメモリの部分に存在するか否かを判定する。要求されたデータがメインメモリの解凍部分に存在しないがストレージデバイスにおいて圧縮フォーマットで存在する場合、データは、デマンドページング操作を通じてメインメモリの解凍部分に転送される。デマンドページング操作中、圧縮データは解凍される。同様に、データがメインメモリの解凍部分からページアウトされ、当該データは保存されなくてはならない場合、当該データは圧縮データのストレージデバイスに記憶される前に圧縮される。

30

40

【 0 0 1 5 】

特許文献 7 (Coward et al.) は、圧縮ファイルの記憶および取り出し方法を開示する。この方法は、コンパクトディスク読み出し専用メモリから直接的に圧縮ファイルを取り出すためにファイル割り付けテーブルを動的に生成することを伴う。

【 0 0 1 6 】

特許文献 8 (Pham et al.) は、クライアントコンピュータシステムのファイルシステムとネットワークデータストレージとの間のデータの確実なアクセスおよび転送をサポートするセキュアネットワークファイルアクセス機器を開示する。クライアントコンピュータシステム上に設けられ、確実なネットワークファイルアクセス機器によってモニタリン

50

グされるエージェントは、ネットワークデータストレージに発信されるファイルシステム要求に対してクライアントコンピュータシステムの認証を確実にする。セキュアネットワークファイルアクセス機器は、クライアントコンピュータシステムとネットワークデータストレージとの間のネットワークインフラストラクチャに設けられ、適格なアクセスポリシーを適用し、ファイルシステム要求に対して選択的に通過させる。セキュアネットワークファイルアクセス機器は、暗号鍵ストレージを保持し、暗号鍵を対応するシステムファイルと関連付けて、セキュアネットワークファイルアクセス機器を通じてネットワークデータストレージに転送されまたはそこから読み出されるファイルデータを暗号化および復号化する。

【 0 0 1 7 】

10

特許文献 9 (Benveniste et al.) は、情報を記憶する方法およびシステムを開示し、メインメモリ圧縮情報をメモリ圧縮ディスクに記憶することを含み、ページはメインメモリ圧縮情報を解凍することなく個別に記憶され取り出される。

【 0 0 1 8 】

特許文献 1 0 (Negishi et al.) は、ネットワーク上で N A S サーバとクライアントとの間に介在し、所定のクライアントから送られる N A S サーバに対する操作要求を一時的に保持する、フロントエンドサーバを開示する。フロントエンドサーバは、N A S サーバで記憶されるデータファイル間の相関に関する情報を保持し、当該情報に基づいてクライアントから受信する操作要求を最適化し、N A S サーバに操作要求を送信する。

【 特許文献 1 】 米国特許第 5,761,536 号明細書

20

【 特許文献 2 】 米国特許第 5,813,011 号明細書

【 特許文献 3 】 米国特許第 5,813,017 号明細書

【 特許文献 4 】 米国特許第 6,092,071 号明細書

【 特許文献 5 】 米国特許第 6,115,787 号明細書

【 特許文献 6 】 米国特許第 6,349,375 号明細書

【 特許文献 7 】 米国特許第 6,584,520 号明細書

【 特許文献 8 】 米国特許第 6,678,828 号明細書

【 特許文献 9 】 米国特許出願公開第 2004/030,813 号明細書

【 特許文献 1 0 】 米国特許出願公開第 2005/021,657 号明細書

【 非特許文献 1 】 "Leveraging Networked storage for your business", 2003, www.emc.com/pdf/products/networkedstorage/leveraging_net_storage.pdf

30

【 非特許文献 2 】 J. Ziv and A. Lempel, "A Universal Algorithm For Sequential Data Compression," IEEE Transactions on Information Theory, IT-23, pp. 337-343 (1997)

【 発明の開示 】

【 0 0 1 9 】

記憶および取り出し能力を損なうことなく、且つ、ユーザが圧縮 / 解凍操作並びに圧縮データの記憶場所を認識する必要性を有することなく、ファイルアクセスストレージシステムに使用する新しい圧縮ストレージのシステムおよび方法を提供することがこの技術において必要である。本発明は、幾つかの態様において、圧縮記憶ファイルにおけるデータへのランダムアクセスを促進し、ファイル全体を解凍することなく圧縮データの操作を可能にする新規の解決策を提供することを目的とする。

40

【 0 0 2 0 】

本発明のある態様では、ファイルアクセスストレージに使用する、圧縮ファイルの作成、読み出しおよび書き込みの方法およびシステムが提供され、上記方法およびシステムは、圧縮ファイルのデフラグを維持しながら圧縮データへの直接アクセスを促進する。

【 0 0 2 1 】

本発明のある態様によると、上記方法は、

原ファイルを圧縮して圧縮データを生成するステップであって、原ファイルのデータ (クラスタ) の少なくとも一つの固定サイズ部分は、順に処理されて、少なくとも一つの固

50

定サイズ圧縮論理ユニット（ＣＬＵ）に分割される対応する圧縮セクションとされる、ステップと、

圧縮ファイルとしての圧縮データの記憶を促進するステップであって、圧縮ファイルは、原ファイルのクラスタに対応する圧縮セクションと、固有のファイル記述子を有するヘッダとを含む、ステップと、

圧縮セクションを記述する少なくとも一つのレコードを有するセクションテーブルを作成するステップであって、上記レコードは圧縮セクションに対応するＣＬＵに関する情報と上記ＣＬＵに関連する記憶場所ポインタとを少なくとも保持する、ステップと、を備える。

【 0 0 2 2 】

上述のように圧縮されるファイルに保存されるデータの読み出しのために、上記方法は更に、

読み出すべきデータを有する第 1 の圧縮セクションのシリアル番号を判定するステップと、

セクションテーブルを参照して上記圧縮セクションに対応するＣＬＵおよびその記憶場所を判定するステップと、

上記圧縮セクションからクラスタを復元するステップと、

読み出すべきデータの範囲が復元されたクラスタのサイズを超える場合には、読み出すべきデータ全てが復元されるまでステップ b) および c) を、シリアル番号を 1 だけインクリメントした圧縮セクションに対して繰り返すステップと、を備える。

【 0 0 2 3 】

上述のように圧縮されるファイルへのデータの書き込みのために、上記方法は更に、

更新されるべき（元のセクション）データを有する第 1 の圧縮セクションのシリアル番号を判定するステップと、

セクションテーブルを参照して上記元の圧縮セクションに対応するＣＬＵおよびその記憶場所を判定するステップと、

上記元の圧縮セクションからクラスタを復元するステップと、

上記クラスタ内の更新するデータのオフセットを計算し、要求されるデータ範囲の更新を促進するステップと、

更新された圧縮セクションに更新されたクラスタを圧縮するステップと、

上記更新された圧縮セクションで上記元の圧縮セクションの上書きを促進するステップと、

セクションテーブルを更新するステップと、

書き込むべきデータの範囲が復元されたクラスタのサイズを超える場合には、書き込むべき必要なデータ全てが復元されるまでステップ b) 乃至 g) を、シリアル番号を 1 だけインクリメントした圧縮セクションに対して繰り返すステップと、を備える。

【 0 0 2 4 】

本発明の更なる態様によると、この方法は、圧縮ファイルへのデータの書き込み中に解放される自由ＣＬＵのリストを扱うステップを含み、上記リストは、ファイルが閉じられるまでファイルに関連する全てのセッション中に扱われる。当該方法は、更に、上記元の N_0 と更新された N_u の圧縮セクションに必要なＣＬＵの数を比較し、

・ $N_0 = N_u$ の場合、更新された圧縮セクションに対応するＣＬＵで元の圧縮セクションに対応する全てのＣＬＵを上書きすること、

・ $N_0 > N_u$ の場合、更新された圧縮セクションに対応するＣＬＵで元の圧縮セクションに対応する N_u 個のＣＬＵを上書きし、解放されるＣＬＵに関する自由ＣＬＵのリストを更新すること、

・ $N_0 < N_u < N_0 + N_f$ (N_f は自由ＣＬＵリスト中のＣＬＵの数) の場合、更新された圧縮セクションに対応するＣＬＵで元の圧縮セクションに対応する全てのＣＬＵを上書きし、更新された圧縮セクションに対応する残りのＣＬＵを自由ＣＬＵのリストに含まれるＣＬＵに書き込むこと、

10

20

30

40

50

・ $N_0 + N_f < N_u$ の場合、更新された圧縮セクションに対応する C L U で元の圧縮セクションに対応する全ての C L U を上書きし、更新された圧縮セクションに対応する C L U を自由 C L U のリストに含まれる C L U に書き込み、次の自由な記憶場所に更新された圧縮セクションに対応する残りの C L U を連続的に書き込むこと、

のいずれかを促進することを含む。

【 0 0 2 5 】

本発明の更なる態様によると、この方法は、ファイルを閉じる前に自由 C L U のリストを確認するステップを更に含み、空でない場合には、

・ 圧縮セクションに含まれる C L U の中で最も高い記憶場所ポインタを有する C L U (第 1 の C L U) を定め、

・ 上記第 1 の C L U からより低い記憶場所ポインタを有する自由 C L U (第 2 の C L U) への圧縮データの移動を促進し、

・ 上記第 2 の C L U を関連する圧縮セクションに割り当て、上記第 1 の C L U を自由 C L U のリストに割り当て、

・ 圧縮セクションに含まれる全ての C L U の記憶場所ポインタが自由 C L U のリストに含まれるどの C L U のポインタよりも低くなるまでステップ b) - d) を繰り返す。

【 0 0 2 6 】

本発明の更なる態様によると、記憶するためにファイルを圧縮するシステムは、

原ファイルを圧縮して圧縮データを生成するサブシステムであって、原ファイルのデータ (クラスタ) の少なくとも一つの固定サイズ部分は、順に処理されて、少なくとも一つの固定サイズ圧縮論理ユニット (C L U) に分割される対応する圧縮セクションとされる、サブシステムと、

圧縮ファイルとしての圧縮データの記憶を促進するサブシステムであって、圧縮ファイルは原ファイルのクラスタに対応する圧縮セクションおよび固有のファイル記述子を有するヘッダとを含む、サブシステムと、

全ての圧縮セクションのレコードを有するセクションテーブルを作成するステップであって、上記レコードのそれぞれは、圧縮セクションに対応する C L U に関する情報と上記 C L U に関連する記憶場所ポインタとを保持する、サブシステムと、を備える。

【 0 0 2 7 】

本発明によるシステムは好適にプログラムされたコンピュータでもよいことは理解されるべきである。同様にして、本発明は、本発明の方法を実行するためにコンピュータによって読み出されるコンピュータプログラムに関する。本発明は、更に、本発明の方法を実行するために機械によって実行可能な命令のプログラムを明白に具現化する機械読み出し可能メモリに関する。

【 発明を実施するための最良の形態 】

【 0 0 2 8 】

本発明を理解し、それが実際にどのようにして実行されるかを知るために、添付の図面を参照して非制限的な例のみにより好ましい実施の形態を説明する。

【 0 0 2 9 】

以下の詳細な説明では、本発明の完全な理解を提供するために多数の特定の詳細を記載する。しかしながら、当業者には本発明がこれらの特定の詳細を有することなく実行され得ることが理解されるであろう。加えて、周知の方法、手順、構成要素および回路は、本発明の説明を不明瞭にしないよう詳細には説明しない。

【 0 0 3 0 】

特に記載しない限り、以下の説明から明らかとなるように、明細書中に用いる「処理」、「演算」、「計算」、「判定」等といった用語は、コンピュータまたは演算システム、あるいは、演算システムのレジスタおよび/またはメモリ内の物理的な、例えば電子的な数量によって表されるデータを、演算システムのメモリ、レジスタもしくは他の情報記憶装置または表示装置内の物理的な量として同様に表される他のデータに操作および/または変換するプロセッサまたは同様の演算装置の作用および/または処理を意味する。

【 0 0 3 1 】

本発明の実施の形態は、操作を実行するプロセッサ、コンピュータ、装置、システム、サブシステム、モジュール、ユニット、デバイス（単独であるいは複数で）といった用語を用いる。これは、所望の目的のために特別に構成されてもよく、あるいは、コンピュータに記憶されるコンピュータプログラムによって選択的に作動または再設定される汎用コンピュータを有してもよい。このようなコンピュータプログラムは、フロッピー（登録商標）ディスク、光学ディスク、CD-ROM、光磁気ディスク、読み取り専用メモリ（ROM）、電気コンピュータ読み取り可能記憶媒体にランダムアクセスメモリ（RAM）、電氣的にプログラム可能な読み取り専用メモリ（EPROM）電氣的に消去可能且つプログラム可能な読み取り専用メモリ（EEPROM）、磁気もしくは光学カードを含む全てのタイプのディスク、または、電子指示を記憶することに適し、コンピュータシステムバスに接続することができる全ての他のタイプの媒体等のコンピュータ読み取り可能記憶媒体に記憶されるが、これらに制限されない。

10

【 0 0 3 2 】

本願記載の処理 / デバイス（または上述の同様の用語）および表示装置は、特に記載しない限り、如何なる特定のコンピュータまたは他の装置にも本質的に関連しない。様々な汎用システムは、本願記載の教示に応じたプログラムに使用され、または、所望の方法を実行するためにより特殊な装置を構成することが便利となる。これらの様々なシステムに対する所望の構成を以下に記載する。更に、本発明の実施の形態は、特定のプログラミング言語を参照して説明されない。様々なプログラミング言語が本願記載の本発明の教示を実行するために使用され得ることが理解されるであろう。

20

【 0 0 3 3 】

これを念頭において、技術において公知の典型的なNASストレージネットワーク構造の概略図を例示する図1に注目する。クライアント11および / またはサーバ12からのファイルは、ファイルストレージデバイス14にIPネットワーク13を介して転送される。ファイルストレージデバイスは、特殊NASファイルサーバ、汎用ファイルサーバ、SANストレージ装置、ストリームストレージ等でもよい。IPネットワークは、ローカルエリアネットワーク（LAN）、広域ネットワーク（WAN）、その組み合わせ等でもよい。

【 0 0 3 4 】

図2aを参照するに、本発明のある実施の形態によるNASストレージネットワーク構造の概略図を例示する。サーバ12および / またはクライアント11とファイルストレージデバイス14との間の通信はプラットフォーム20を介する。プラットフォーム20は、一つ以上のクライアント / サーバを接続するクライアント / サーバインタフェース21と、一つ以上のファイルストレージデバイスとのブリッジ / プロキシ接続のためのファイルストレージインタフェース22とを備える。プラットフォーム20は、選択されたトランザクションではプロキシとして機能し、それにより、制御トランザクションに対するスループット、および、データトランザクションに対するプロキシを保持する。プラットフォーム20は、全ての物理的インタフェース（例えば、イーサネット（登録商標）、ファイバチャネル等）をサポートし、例えば、冗長性、ミラーリング、スナップショット、フェールオーバー、ロールバック、管理等のストレージデバイス特徴を保存する。プラットフォームは、既存のネットワークインフラストラクチャと継ぎ目なく一体化される。以下に説明する圧縮およびアクセス機能を実施するために、典型的には、プラットフォームが、ファイル自体より上の全てのレベルにおいて変更を必要としない。ユーザは、圧縮および解凍の操作並びに圧縮データの記憶場所を考慮する必要がない。

30

40

【 0 0 3 5 】

記憶前の圧縮されるべきファイルに対する「書き込み」操作中、クライアント11および / またはサーバ12からのファイルは、ネットワーク13を介してプラットフォーム20に流れ、プラットフォームによって圧縮され、ファイルストレージデバイス14にて書き込まれる。異なる種類のデータ（例えば、テキスト、画像、音声等）を含むファイルは、異

50

なるアルゴリズムによって圧縮される（例えば、<http://www.data-compression.com>参照）。「読み出し」操作は逆方向に実行され、つまり、必要なファイルはプラットフォームによって取り出され、解凍（要求されるデータ範囲に応じて部分的にあるいは全体的に）され、適当なクライアント/サーバに送られる。以下に、図3乃至図9を参照して本発明のある態様による圧縮/解凍操作を更に説明する。図2bは、クライアント/サーバから受信されるファイルがネットワーク13を介して転送される前にプラットフォームで圧縮され書き込まれる場合の本発明の他の実施の形態によるストレージ構造を例示する。

【0036】

ある実施の形態では、プラットフォーム20は、暗号化、認証等のセキュリティ機能も提供する。

10

【0037】

プラットフォーム20は、透過的な方法によりクライアント/サーバとファイルストレージデバイスとの間で、選択されたトランザクション（典型的には、ファイルのコピー、ファイルの削除、ファイル名を変える、スナップショット等の制御関連のトランザクション）を転送する一方で、データ関連のトランザクション（例えば、開く、閉じる、読み出し、書き込み、作成等）、および、例えば、ディレクトリリストコマンド等の幾つかの制御関連のトランザクションに介入するよう構成される。

【0038】

本発明のある実施の形態では、プラットフォームは、プラットフォーム20を介さずにファイルストレージデバイス14に先験的に記憶される圧縮されていないファイルを圧縮するよう構成されてもよい。プラットフォームは、予め定められた基準のファイルに従ってストレージデバイスから全てのまたは選択されたファイルを抽出し、それらを本発明に従って圧縮し、圧縮された形態でストレージデバイスに書き換えてもよい。

20

【0039】

本発明のある実施の形態では、プラットフォーム20は、予め定められた基準（例えば、サイズ、IPアドレス、ディレクトリ、ファイルタイプ）に従って選択通過したファイルだけを圧縮するよう構成されてもよい。

【0040】

原ファイル（または、その関連部分）は、ファイルストレージデバイス14への書き込み中または書き込み前にプラットフォーム20によって圧縮される。同様にして、ファイル（またはその関連部分）は、ファイルストレージデバイス14からの読み出し中または読み出し後にプラットフォーム20によって解凍される。本発明の幾つかの実施の形態では、ファイルは圧縮された状態でプラットフォーム20とファイルストレージデバイス14との間を転送され、例えば、原ファイルはプラットフォーム内で圧縮されて書き込まれ、結果として生ずる圧縮ファイルがファイルストレージデバイスに更に転送される、および/または、圧縮ファイルがプラットフォーム内での更なる読み出し/書き込み操作のためにファイルストレージデバイスから受信される。

30

【0041】

本発明は、図1および図2を参照して説明した特定のネットワーク構造によって制限されないことに注意する。当業者には、本発明がNAS、あるいは、IPまたはチャンネルベースの通信を有するファイルアクセスストレージサブネットワークに基づく全てのネットワークに同様に適用可能であることが容易に理解されるであろう。プラットフォーム20（またはその一部）の機能は、幾つかのプラットフォーム間で分散されている、または、他のストレージネットワーク要素（例えば、ファイルサーバ、企業スイッチ等）内で一体化されている、スタンドアロン型サーバ（図2参照）で実行されてもよい。

40

【0042】

図3は、本発明のある実施の形態によるプラットフォーム20の概略的な機能ブロック図である。

【0043】

入力/出力ブロック31の主な機能は、ソケットハンドリングであり、同ブロックはI

50

P通信の完全性に対して責任を担う。I/Oブロック31は、新しいTCPパケットを受信し、それをアンパッキングし、セッションマネージャ32に移行する。このブロック31は、セッションマネージャからもデータを取得し、それをTCP構成にパッキングし、要求アドレスに送る。

【0044】

セッションは、「ファイルを開く」といった要求によって開始し、同じIPアドレス(ユーザ)から受信する「ファイルを閉じる」といった要求によって終了する。セッションマネージャ32は、例えば、ソースIPアドレス等の全てのセッションのプライベートデータ、使用中の全てのファイルインスタンス、セッションカウンタ、セッションステータス、使用中のバッファに対する全てのインスタンス等を保持する。セッションマネージャは、「ファイルブロック」も扱い、切断時に全ての関連するリソースを解放する。セッションマネージャ32は、受信したパケットのフィルタ処理のためにディスパッチャ33にセッションタスクをロードする。ディスパッチャは、CIFS/NFSマネージャ34にCIFS/NFSパケットを送り、全ての他のパケットを手付かずのままI/Oブロックに(セッションマネージャを介して)送り返す。CIFS/NFSマネージャ34は、これに応じてCIFSおよびNFSTRANZAKUSIONを統合し、データ関連のTRANZAKUSION(例えば、開く、読み出す、書き込む、閉じる等)をファイルマネージャ35に要求し、本発明のある実施の形態による圧縮/解凍操作を圧縮/解凍ブロック37に要求する。一般的には、圧縮アルゴリズムは、圧縮効率性と性能パラメータとのトレードオフによって特徴付けられる幾つかの圧縮レベルを有する。圧縮ブロック37は、最適圧縮レベルを選択し、入力/出力ブロック31(および/またはCPU利用)によって現在扱われるソケットの数に圧縮比を調節する。選択された圧縮レベルの情報は、データの圧縮部分に保持される。ファイルマネージャ35は、ファイルの完全性と操作とに対する責任を担う。ファイルマネージャ35は、ファイルに関連する全ての要求を組み合わせ、ファイルマニピュレーションの共有を可能にする。圧縮/解凍ブロック37は、バッファから読み出し、解凍することができ、更に、CLUSを圧縮し書き込むことができる。メモリバッファリソースは、CIFS/NFSマネージャに接続されるバッファマネージャ36によって管理される。完全性マネージャ38は、I/Oブロック、セッションマネージャ、バッファマネージャおよびファイルマネージャに接続され、プラットフォームにおける全ての処理の同期および全体的な制御に対する責任を担う。

【0045】

当業者には、本発明が図3の構成によって制限されず、同等のおよび/または変更された機能が別の方法で統合あるいは分割され得ることは容易に理解されるであろう。

【0046】

図4は、本発明のある実施の形態による原ファイルおよび圧縮ファイルの概略図である。圧縮されていない原ファイル41は、略等しい所定のサイズを有するデータ部分43(以降クラスタと称する)にセグメント化される。これらのクラスタは、ファイルに対する入力/出力TRANZAKUSION中に圧縮/解凍操作の原子要素として機能する。クラスタへの原ファイルのセグメント化は、圧縮処理中に「オンザフライで」提供され、所定のサイズを有する次の部分43それぞれは圧縮を施されるクラスタを構成する。本発明のある実施の形態では、セグメント化は、圧縮前に行われてもよい。原ファイルの最後の部分のサイズは、クラスタの所定のサイズ以下でもよく、いずれの場合でも当該部分は完全なクラスタのサイズを有するかのよう扱われる。クラスタのサイズは設定可能であり、クラスタがより大きいと処理オーバーヘッドがより低くなり圧縮比がより高くなり、クラスタがより小さいとアクセスがより効率的になるが処理オーバーヘッドがより高くなる。更に、クラスタのサイズは、各ファイルセッションの圧縮/解凍がメモリにおいて利用可能なクラスタを少なくとも一つ必要とし、性能が同時セッションの数を定めるため、利用可能なクラスタ利用可能なメモリと要求される性能に依存する。クラスタの数は、(原ファイルのサイズをクラスタのサイズで割算した)整数値、あるいは、余りがある場合には、それに1を加算した値に等しい。

【 0 0 4 7 】

代替的には、本発明のある他の実施の形態では、クラスタのサイズは、例えば、データのタイプ（例えば、テキスト、画像、音声、合成等）による所定の基準に応じて変化する。例えば、各データのタイプは、所定のサイズのクラスタを有し、圧縮中プラットフォームは原ファイルの圧縮部分で優位を占めるデータのタイプに応じて適切なサイズのクラスタを選択してもよい。

【 0 0 4 8 】

各ファイル内クラスタ 4 3（例えば、図 4 では 4 3 A - 4 3 C）は、対応する圧縮セクション 4 6（例えば、図 4 では 4 6 A - 4 6 C）に圧縮される。同じサイズを有するクラスタは、必然的に結果として、各クラスタにおけるデータの性質および圧縮アルゴリズムに依存して異なるサイズの圧縮セクションを生成する。クラスタ圧縮の比が所定の値未満である場合、圧縮ファイルにおける対応する圧縮セクションは当該クラスタからの圧縮されていないデータを有する場合もある。例えば、所与のクラスタにおける原データが元のクラスタサイズの少なくとも X %（例えば、95%）に圧縮されると、無視できる圧縮比により、対応するセクションは圧縮データの代わりに原クラスタデータを受けることになる。

【 0 0 4 9 】

本発明のある実施の形態では、圧縮処理は、適応能力を含み、内容に応じて各クラスタに最適圧縮アルゴリズム（例えば、優位を占める音声、テキスト、画像等のデータを有するクラスタに最適な異なる圧縮アルゴリズム）を提供する。

【 0 0 5 0 】

本発明のある実施の形態では、各圧縮ファイル 4 4 はヘッダ 4 5 と、幾つかの圧縮セクション 4 6 と、セクションテーブル 4 7 とを有する。圧縮ファイルのヘッダ 4 5 は、固有のファイル記述子、原ファイル 4 1 のサイズ、および、ファイルがプラットフォーム 2 0 によって処理されたか否かを示す署名（所定の値未満の取得可能な圧縮比によりプラットフォームによって圧縮されていないファイルについても同様）を有する。

【 0 0 5 1 】

圧縮ファイル内の圧縮セクションの数は、クラスタの数に等しい。本発明のある実施の形態では、圧縮セクション 4 6 におけるデータは所定のサイズを全て有する圧縮論理ユニット（CLU）4 8 に記憶される（例えば、図 4 では、圧縮論理ユニット 4 8 A 0 - 4 8 A 2 はクラスタ 4 3 A に対応する圧縮セクション 4 6 A に対応する）。所定の CLU サイズは設定可能であり、CLU がより大きいとオーバーヘッドが低くなり、CLU がより小さいと分解能がより高くなる。更に、本発明のある実施の形態では、CLU サイズは最大および/または最適 C I F S / N F S パケット長に調節されてもよい。

【 0 0 5 2 】

圧縮セクション内の CLU の数は、（圧縮セクションのサイズを CLU のサイズで割算した）整数値、あるいは、余りがある場合には、それに 1 を加算した値に等しい。圧縮セクションにおける最後の CLU は、部分的に満たされている（例えば、図 4 において 4 8 - A 2、4 8 - C 1）。このような CLU は満たされた CLU と同様に扱われる。本発明のある実施の形態では、最後の圧縮セクション（例えば、図 4 において 4 8 - C 1 によって例示される）における最後の CLU は特別な方法で扱われる、即ち、部分的に満たされた場合に（以下に図 9 を参照して更に説明する）、正確な圧縮サイズに切られる。

【 0 0 5 3 】

CLU は、メモリにおいてセグメントの仮想シーケンスによって形成される圧縮ファイルの仮想部分として考えられる。CLU と割り当てられたメモリセグメントとの間の関係は、以下に図 1 1 を参照して更に説明される。

【 0 0 5 4 】

セクションテーブル 4 7 は、全ての圧縮セクション 4 6 のレコードを有し、圧縮セクションそれぞれに対応する CLU をどこで見つけるかを特定する。レコードは、各圧縮セクション（以降、セクションレコードと称する）に対して、セクションが圧縮されたか否か

10

20

30

40

50

を示す署名、圧縮セクションの全体的なサイズ、セクションに含まれる全てのCLUに関連するポインタのリストを有する。任意には、レコードは、対応するクラスタの圧縮中に使用される圧縮アルゴリズムの表示、および、クラスタのサイズ（所定の基準に対して可変）を有する。好ましくは、セクションテーブル47は、圧縮ファイルの終わりに配置される。なぜなら、その長さはファイルのコンテンツが更新されると変化するためである（更に例示するように、セクションテーブルの長さは、圧縮セクションの数、従って、クラスタの数に比例する）。

【0055】

図5は、例示的なファイルのセクションテーブルの典型的な構成を非制限的な例により例示する。

10

【0056】

この例示的なファイル50（更なる例でも参照する）は、元のサイズが3MB+413ビットであり、所定のクラスタサイズが1Mであり、CLUサイズが60Kである。従って、原ファイルは4個のクラスタを含む（1MBのクラスタが3個と、部分的に満たされているが完全なクラスタとして扱われるクラスタが1個）。

【0057】

ある圧縮セクションのレコード51は、署名52と、セクション53のサイズと、幾つかのエントリ54とを有する。セクションレコードの各エントリ54は、圧縮セクションに含まれるCLUの一つに関する情報を有する。セクションテーブルは、物理的な場所と論理CLU#との関係を有する。

20

【0058】

例示的なファイル50のクラスタは、例えば、301123、432111、120423、および、10342バイトのそれぞれのサイズを有する圧縮セクションに圧縮される。CLUの長さ60Kであることは61440バイトを意味し、セクション#0は5個の割り付けられたCLU（[301123/61440]+1）を有し、セクション#1は8個の割り付けられたCLU（[432111/61440]+1）を有し、セクション#2は2個の割り付けられたCLU（[120423/61440]+1）を有し、セクション#3は1個の割り付けられたCLU（[10342/61440]+1）を有する。合計で、圧縮ファイルは16個のCLU（合計サイズは15×61440バイト+10342バイト）、固定長ヘッダ（例えば、署名に対する4バイト、ファイルID（固有記述子）に対する16バイト、および元のサイズに関する情報に対する4バイト）、および、4個のセクションレコードを有するセクションテーブルを有する。

30

【0059】

例示的なファイル50は新しい圧縮ファイルとして作成された場合、CLUは順に割り付けられる。例えば、

最初に、ポインタ1、2、3、4、5を有する5個のCLUがセクション0に割り付けられ、

次に、ポインタ6、7、8、9、10、11、12、13を有する8個のCLUがセクション1に割り付けられ、

次に、ポインタ14、15を有する2個のCLUがセクション2に割り付けられ、

40

次に、ポインタ16を有する1個のCLUがセクション3に割り付けられる。

【0060】

ファイル内のCLUの配分は更新後に変更されてもよい（以下に図8乃至図11を参照して更に説明する）。例えば、

ポインタ1、4、5、6、9を有するCLUがセクション0に割り付けられ、

ポインタ2、3、7、10、11、12、15、14を有するCLUがセクション1に割り付けられ、

ポインタ8、13を有するCLUがセクション2に割り付けられ、

ポインタ16を有するCLUがセクション3に割り付けられる。

【0061】

50

(本例では、更新は圧縮セクションのサイズに対して影響を及ぼさない)。

【0062】

ファイルが新しい圧縮ファイルとして作成されると、CLUの仮想(論理)シーケンスがCLUに対応するディスクセグメントの物理的シーケンスと同じである。更新されたファイルでは、CLUの仮想(論理)シーケンスは、CLUに対応するディスクセグメントの物理的シーケンスと異なる場合もある。例えば、上述の例では、第1のクラスタの第2のCLUは物理的セグメント#2に最初に位置していたが、更新後、物理的セグメント#4に位置する。各CLUはメモリにおけるセグメントに割り当てられ、対応するセグメントは、CLUの長さを加算しセグメントシリアル番号を乗算したヘッダ45の長さのオフセットに書き込まれる。例えば、上述の例示的なファイルでは、第1のクラスタの第2のCLUが物理的セグメント#2に位置する場合、記憶場所メモリにおいて2×61440バイトを加算したヘッダのオフセット24バイトに書き込まれる。更新後に当該CLUが物理的セグメント#4に位置すると、そのオフセットはヘッダに4×61440バイトを加算した24バイトとなる。

10

【0063】

本発明のある実施の形態では、各セクションレコードのエントリの数は一定であり、クラスタを記憶するに必要となるCLUの最大数に対応し、従って、各セクションレコードのサイズはセクションに設けられるCLUの実際の数に関わらず一定であり、使用中でないエントリには特別なマークが設けられる。セクションレコードにおけるエントリの数はクラスタのサイズをCLUのサイズで分割し、1を加算した整数値に等しい。

20

【0064】

クラスタの所定のサイズが1MBであり、CLUの所定のサイズが60Kの例示的な例では、圧縮セクションの各レコードはそれぞれが4バイトを有する17個のエントリ(1MB/60K+1の整数値)を有する。圧縮セクション#0の例示するセクションレコード50は、対応するCLUの物理的な場所に関する情報、および、12個の空のエントリ(例えば、-1とマーキングされる)を含む5個のエントリを有する。セクションレコードのサイズは72バイトである(圧縮セクションおよび署名のサイズに関する情報に対して4バイト+17エントリ×4バイト)。セクションテーブルの全体的なサイズは288バイト(各セクションレコードに対して4圧縮セクション×72バイト)である。

【0065】

本発明のある実施の形態では、圧縮データはセクションテーブル47とは別個に記憶されてもよい。セクションテーブルは、圧縮データと同じ記憶スペース内の別のファイルに、または、例えば、プラットフォーム20の、異なる、場合によっては物理的に遠隔な記憶スペースに記憶されてもよい。プラットフォーム20は、読み出し/書き込み操作中の圧縮データと対応するセクションテーブルとの間の関連付けを促進するように構成されなくてはならない。

30

【0066】

図6乃至図11は、本発明のある実施の形態による圧縮ファイルに対して実行される入力/出力操作を例示する。プラットフォーム20が、原ファイル(例えば、DIR、STAT等)のサイズに関するコマンドに介入し、対応する圧縮ファイルのヘッダのサイズを維持し、上記データを要求があると供給する。従って、例えば、ファイルサイズがX(原ファイルの状態)で圧縮状態(ディスクに記憶された状態)でY(<X)を有するファイルを考える。特定の特徴により、ヘッダに記憶されるファイルのサイズは、X(原ファイルのサイズ)であり、DIR、STAT等のシステムコマンドを考える限り完全に透過性を維持する。

40

【0067】

本発明のある実施の形態による圧縮された特定のファイルを開くための入来要求を受信すると(ユーザはファイルが圧縮されていることに気付かない場合もある)、プラットフォーム20は当該要求をNASシステム(ストレージデバイス14)に転送し、ファイル管理に対する鍵として機能する「ハンドル」応答(または、ファイルが見つからない場合に

50

は「ヌル」)を受信する。「ハンドル」の受信に続いて、プラットフォーム20はファイルID(固有のファイル記述子)を含むヘッダ45および対応する原ファイルのサイズを読み出す。プラットフォーム20がファイルに関連する同時のセッションが存在するか否かをファイルID毎に確認する。「ノー」である場合、プラットフォームは、固有のファイル記述子と原ファイルのサイズとを含むファイルブロックを生成する。ファイルが既に使用中の場合、プラットフォームは、既存のファイルブロックに追加的なセッションを追加する。「ハンドル」は、ファイル操作に対する以下の要求とプラットフォームに送られるようユーザに戻される。

【0068】

ファイルを開く操作は、圧縮ファイルのセクションテーブル47を読み出し、ファイルに対応する全てのCLUの情報を得ることを含む。ファイルが開かれた瞬間から閉じられるまで、プラットフォームはファイルのCLU構成およびファイル内の全てのバイトのオフセットを認識している。

【0069】

図6を参照するに、本発明のある実施の形態による圧縮ファイルを作成する一般化されたフローチャートを例示する。この処理は、入来作成要求によって開始される。プラットフォーム20は、ファイルストレージデバイス14に発出要求を生成し(60)、確認後、ファイルストレージデバイスで圧縮ファイルのヘッダに書き込み始める(61)。図4に記載したように、ヘッダは、ファイル記述子、未圧縮の原ファイルのサイズ、および、ファイルがプラットフォーム20によって処理されたことを示す署名を含む。次のステップ(62)では、プラットフォームは、原ファイルの最初の固定サイズの部分(クラスタ)をサイズXを有する圧縮セクションに処理する。(圧縮は、任意の適当な市販のまたは特殊なアルゴリズムの補助で提供される)。プラットフォームは、第1のCLUに対する最初の自由記憶場所を定め、ファイルストレージデバイスでの記憶について当該CLUおよび後続するCLUへの圧縮セクションの連続的な書き込み(63)を開始して扱い、セクションテーブルにレコードされるべき処理中に占有されるCLUのポインタを準備する(64)。プラットフォームは、ファイル全体のデータが圧縮形態で書き込まれるまで次のクラスタに対して処理を繰り返し(65)、セクションテーブルが作成される(66)。本発明のある実施の形態では、セクションテーブルは、圧縮ファイル外で記憶されてもよい。

【0070】

図7を参照するに、本発明のある実施の形態による圧縮ファイルに対する読み出し操作の一般化されたフローチャートが示される。

【0071】

読み出し操作は、入力パラメータ(例えば、ファイルハンドル、シーク番号(データオフセット)、および、データ長Y)および出力パラメータ(例えば、ターゲットバッファアドレス)を有する入来読み出し要求で開始される(70)。入来読み出し要求は、オフセット(原ファイルにおける)と、読み出すデータの範囲Yとを識別する。プラットフォーム20は、読み出すべき第1のクラスタ(以降、開始クラスタと称する)のシリアル番号を(オフセットをクラスタで割算した)整数値、あるいは、余りがある場合には、それに1を加算した値として計算する(71)。読み出すべきクラスタの数は、(読み出すべきデータの範囲をクラスタのサイズで割算した)整数値に1を加算した値として定義される。その結果、プラットフォームは、読み出すべきクラスタに一对一に対応して圧縮セクションを定め、発出読み出し要求を生成する(72)。要求は、圧縮ファイル(ヘッダおよびセクションテーブル)のメタデータに基づき、読み出すべき圧縮セクションに対応するCLUを指す。本発明のある実施の形態では、圧縮ファイルの終わりに配置されるセクションテーブルのオフセットは、圧縮ファイルのサイズ-クラスタの数×セクションレコードの固定サイズにより簡単に計算される。

【0072】

他の実施の形態では、プラットフォームは、圧縮データと別のファイルに保存されている対応するメタデータとの間の関連付けを促進するよう構成される。

【 0 0 7 3 】

本発明のある実施の形態では、発出読み出し要求は、読み出すべきデータの範囲全てを特定して送られてもよい。あるいは、図7に示すように、読み出し要求全体がステップ毎に扱われ、読み出し操作に対して、プラットフォームはクラスタのサイズに略等しいバッファを維持してもよい。第1の発出読み出し要求は、開始クラスタの圧縮セクションに含まれるCLUに対するポインタを有する。開始クラスタに対応する圧縮セクション全体は読み出され(73)、プラットフォームによってターゲットバッファに解凍される(74)。次のステップでは、プラットフォームは、クラスタ内の要求されるオフセットを計算し(75)、要求されるデータをコピーし(76)、アプリケーションに送る。コピーするデータの必要な長さは次の通りに計算される。

長さ = 最小 { データ範囲 Y ; (クラスタのサイズ - オフセット mod クラスタのサイズ) }

【 0 0 7 4 】

データ範囲 Y がクラスタのサイズを超えると、操作が繰り返される(77)。

【 0 0 7 5 】

例えば、例示的なファイル50を参照するに、要求は、オフセット1MB + 1340から20バイトの長さのファイルデータを読み出すことである。読み出しは、第2のクラスタから始まり、従って、要求されるデータが二番目の圧縮セクションから始まる圧縮ファイルに含まれる。セクションテーブルのオフセットは、圧縮ファイルからクラスタの数(4) × セクションレコードのサイズ(72バイト)として定義される。セクションテーブルにおける二番目の圧縮セクションのレコードは、ポインタ2、3、7、10、11、12、15、14を有するCLUを含む。従って、これらのCLUは、プラットフォーム20における一時的なバッファに読み出され、プラットフォームにおいて1MBのバッファに解凍される。バッファオフセット1340からの20バイトはターゲット(ユーザの)バッファに移動される。コピーするデータの必要な長さは、20バイトである(20バイトと(1MB - 1340バイト)の最小値に等しい)。他の要求は同じオフセットから2MBの長さのファイルデータを読み出すことである場合、操作は、三番目、四番目の圧縮セクションに対して同様に繰り返され、開始クラスタからコピーするデータの必要な長さは1MB - 1340バイトとなる(2MBと(1MB - 1340バイト)の最小値に等しい)。

【 0 0 7 6 】

図8を参照するに、本発明のある実施の形態による圧縮ファイルに対する書き込み操作の一般化されたフローチャートが示される。入来書き込み要求(80)は、オフセット(原ファイルにおける)と書き込むべきデータの範囲Yを識別する。プラットフォーム20は、更新されるべき(上書きすべき)一番目のクラスタのシリアル番号を(オフセットをクラスタのサイズで割算した)整数値、あるいは、余りがある場合にはそれに1を加算した値として計算する(81)。上書きすべきクラスタの数は、(書き込むべきデータの範囲をクラスタのサイズで割算した)整数値あるいは、余りがある場合にはそれに1を加算した値として定義する。その結果、プラットフォームは上書きする圧縮セクションを定め、図7を参照して説明した方法と同様に発出読み出し要求を生成する。開始クラスタに対応する圧縮セクション全体が読み出され(82)、プラットフォームによってバッファに解凍(83)された後、プラットフォームは、図7を参照して説明したように、クラスタ内の要求されるオフセットを計算(84)し、要求されるデータ範囲を更新する(上書きする)(85)。続いて、プラットフォームは、更新されたクラスタを圧縮し(86)、セクションテーブルを更新し、圧縮ファイルに新しい圧縮セクションを書き込む(87)ことを要求する。データ範囲Yがクラスタのサイズを超える場合、操作は、後続するクラスタに対して繰り返される(88)。処理が終わると、プラットフォームはセクションテーブルを更新する(89)。

【 0 0 7 7 】

上述した通り、本発明のある実施の形態では、要求されるデータの記憶場所は、直接的にアクセスされ、従って、読み出し/更新(および同様の)操作は、ファイル全体でなく

10

20

30

40

50

必要なデータ範囲を含むクラスタだけを復元することを必要とする。

【0078】

典型的には、ファイルの更新により、割り付けられたストレージで不使用スペースが生ずるため、断片化を生ずる。図9および図10は、本発明のある実施の形態によるCLU管理の断片化扱いアルゴリズムを示す。図9は、本発明のある実施の形態による圧縮セクション(図8のステップ87)に対する書き込み/更新操作中のCLU管理のアルゴリズムを示す。更新された圧縮セクションを書き込む前に、プラットフォームは、更新された圧縮セクションと古い圧縮セクションに必要なCLUの数を比較する(91)。CLUの数が変化していない場合、プラットフォーム20は、古い圧縮セクションに対応する全てのCLUに対して更新された圧縮セクションを順に書き込むことを要求する(92)。要求されるCLUの新しい数が古い数未満である場合、圧縮セクションは、古い圧縮セクションに対応するCLUの部分に順に書き込まれる。解放されたCLUに関する情報は、ファイルが閉じられるまでプラットフォーム20によって扱われる自由CLUの特別なリスト(キュー)に更新される(93)。要求されるCLUの数が古い数よりも大きい場合、圧縮セクションは古い圧縮セクションに対応する全てのCLU順に書き込まれ(94)、続いて、自由CLUのキューから取得したCLUに対して書き込まれる(95)。更なるCLUが必要な場合には、プラットフォームはファイル(#n)に割り付けられた最後のCLUを定め、(n+1)の数から始まるCLUに対して順に書き込むことを要求し(96)、割り付けられたCLUのリストはこれに応じて更新される(97)。

10

【0079】

本発明のある実施の形態では、最後の圧縮セクション(図4の48-C1によって示される)における最後のCLUは、特別な方法で扱われ、即ち、部分的に満たされている場合に正確な圧縮サイズに切られる。セクションテーブルは、ヘッダ長+(N-1)×CLUサイズ+S_Lのオフセットに対して書き込まれ、このとき、Nは割り付けられたCLUの合計であり、S_Lは最後のCLUにおける圧縮データのサイズである。

20

【0080】

図10は、本発明のある実施の形態による、ファイルを閉じる操作中のCLU管理のアルゴリズムを示す。

【0081】

ファイルを閉じる(102)前に、プラットフォームは、自由CLUのリストが空か否かを確認する(101)。リスト中にまだCLUが存在する場合、プラットフォーム20は使用中のCLUの中から最も高い記憶場所ポインタを有するCLUを定める。上記CLUに含まれる圧縮データは、より低いポインタで自由CLUに転送され(103)、空になったCLUは自由CLUのリストに追加される。当該処理は、使用中のCLUの全てのポインタが自由CLUのリストにあるどのCLUのポインタよりも低くなるまで繰り返される(104)。セクションテーブルは、これに応じて更新される(105)。このような更新は、上記CLUの書き換え毎に、書き換え処理全体の終わり、または、他の所定の基準に応じて行われる。処理が終わると、ファイルは閉じられ、自由CLUは解放される(106)。上述の処理に対する自由CLUの選択は、異なるアルゴリズムに従って提供されてもよい。例えば、本発明のある実施の形態では、最も高い記憶場所ポインタを有するCLUからの上記圧縮データは、最も低い記憶場所ポインタを有する自由CLUに転送されてもよい。

30

40

【0082】

図11a乃至図11cを参照するに、本発明のある実施の形態によるCLUと割り当てられたディスクメモリセグメントとの間の関係が示される。図11aは、新しい圧縮ファイルとして作成されたときの図5に示す例示的なファイル50を示す。CLUの仮想(論理)シーケンスは、CLUに対応するディスクセグメントの物理的シーケンスと同じである(CLU内の数に対応するディスクメモリセグメントへのポインタを示す)。図11bは、図5を参照して説明した更新された例示的なファイルのように圧縮セクションのサイズが変更されていない更新された圧縮ファイル内のCLUの新しい配分を示す。CLUの

50

仮想（論理）シーケンスは、ファイルのデフラグ構成（de-fragmented structure）を維持しつつ、ＣＬＵに対応するディスクセグメントの物理的なシーケンスとは異なる。図 1 1 c は、更新された例示的な圧縮ファイル 5 0 内のＣＬＵのデフラグ配分を示し、二番目の圧縮セクションのサイズは 4 3 2 1 1 1 から 2 0 0 1 0 0 バイトへの更新後に変更されている。例えば、更新オフセットが 1 MB + 3 1 4 バイトの場合、第 1 の圧縮セクションは更新中に影響を及ぼされない。二番目の圧縮セクションの新しいサイズは、4 C L U（「2 0 0 1 0 0 / 6 1 4 4 0」+ 1）の割り付けだけが必要とする。ここで、図 1 1 b に示すように、更新前には、第 2 の圧縮セクションは 8 個の C L U（番号 2、3、7、1 0、1 1、1 2、1 5 および 1 6）を収容している。図 9 を参照して説明したように、プラットフォーム 2 0 は、圧縮セクション（本例では 2、3、7、1 0）から最初の 4 個の C L U に対して更新された二番目の圧縮セクションを書き込み、ポインタ 1 1、1 2、1 5 および 1 6 を有する C L U を自由 C L U のリストに送る。三番目および四番目の圧縮セクションもこの特定の更新中は影響を及ぼされない。図 1 0 を参照して説明したように、プラットフォーム 2 0 は、ファイルを閉じる前に自由 C L U のリストが空か否かを確認する。本例では、リストは、記憶場所ポインタ 1 1、1 2、1 5 および 1 6 を有する C L U を含む。図 1 0 を参照して説明したように、プラットフォームは、ポインタ 1 3 を有する C L U からポインタ 1 1 を有する C L U に圧縮データを書き換え、ポインタ 1 6 を有する C L U からポインタ 1 2 を有する C L U に圧縮データを書き換え、ポインタ 1 3 乃至 1 6 を有する C L U を解放する。従って、更新されたファイルは、デフラグのない 1 2 個の割り付けられた C L U を有する。

【 0 0 8 3 】

本発明は、図 6 乃至図 1 1 を参照して説明したプラットフォーム 1 2 とストレージデバイス 1 4 との間の特定の入来 / 発出通信に限定されないことに注意する。当業者には、本発明がネットワーク構造およびプラットフォーム 2 0 の機能の実行方法によって全ての他の通信にも同様に適用可能であることが理解されるであろう。例えば、圧縮ファイルが内部プラットフォームリソースに書き込まれおよび / または読み出される場合、発出要求を生成する必要はない。同様にして、プラットフォーム機能がストレージリソースと一体化されている場合には発出要求の必要性はない。追加的な要件（例えば、確実なアクセス、データ保全性等）は、プラットフォームとファイルストレージデバイスとの間の通信並びにネットワークリソース全体をより複雑にする。

【 0 0 8 4 】

本発明は、本願に記載するあるいは図面に例示する説明にその適用が制限されないことは理解されるであろう。本発明の他の実施の形態も可能であり、様々な方法で実行され実施されてもよい。従って、本願で用いる表現および用語が、説明目的であり、制限的として解釈されてはならない。当業者には、本開示が基づく概念が、本発明の幾つかの目的を実行するための他の構成、方法およびシステムを設計する基礎となることは理解されるであろう。

【 0 0 8 5 】

当業者には、添付の特許請求の範囲に定められる範囲から逸脱することなく上述の説明したような本発明の実施の形態に様々な変更および変化が行われてもよいことは容易に理解されるであろう。

【 図面の簡単な説明 】

【 0 0 8 6 】

【 図 1 】 当該技術において公知の典型的な N A S ストレージネットワーク構造の概略的なブロック図

【 図 2 a 】 本発明のある実施の形態による N A S ストレージネットワーク構造の概略的なブロック図

【 図 2 b 】 本発明のある実施の形態による N A S ストレージネットワーク構造の概略的なブロック図

【 図 3 】 本発明のある実施の形態によるシステム機能構造の概略的なブロック図

10

20

30

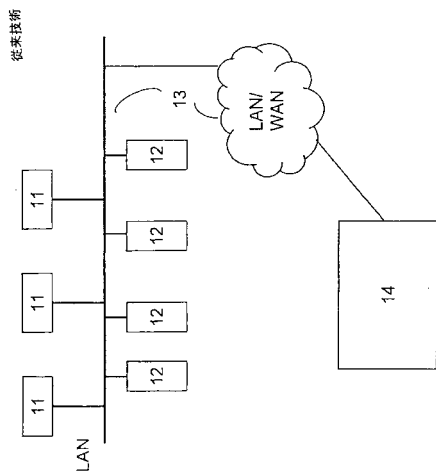
40

50

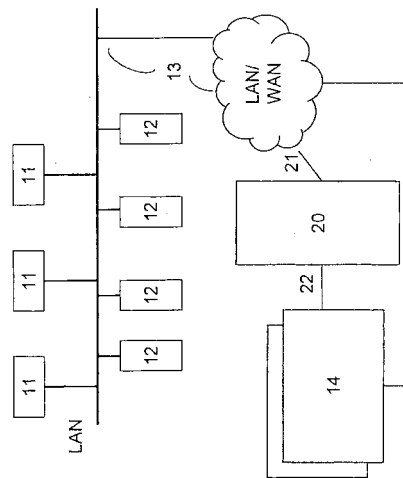
- 【図4】本発明のある実施の形態による原ファイルおよび圧縮ファイルの略図
- 【図5】本発明のある実施の形態によるセクションテーブルの例示的な構成を示す図
- 【図6】本発明のある実施の形態による圧縮ファイルを作成する操作の一般化されたフローチャート
- 【図7】本発明のある実施の形態による圧縮ファイルに対する読み出し操作の一般化されたフローチャート
- 【図8】本発明のある実施の形態による圧縮ファイルに対する書き込み操作の一般化されたフローチャート
- 【図9】本発明のある実施の形態による圧縮セクションに対する書き込み操作のシーケンスを例示する一般化されたフローチャート
- 【図10】ファイルに対する閉じる操作中のCLU管理の一般化されたフローチャート
- 【図11a】本発明のある実施の形態によるCLUと割り当てられたディスクメモリセグメントとの間の関係を示す略図
- 【図11b】本発明のある実施の形態によるCLUと割り当てられたディスクメモリセグメントとの間の関係を示す略図
- 【図11c】本発明のある実施の形態によるCLUと割り当てられたディスクメモリセグメントとの間の関係を示す略図

10

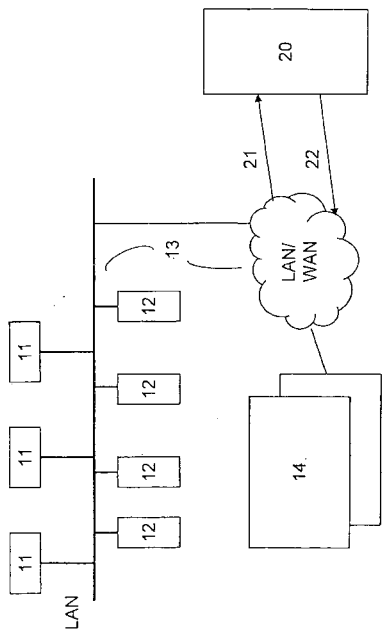
【図1】



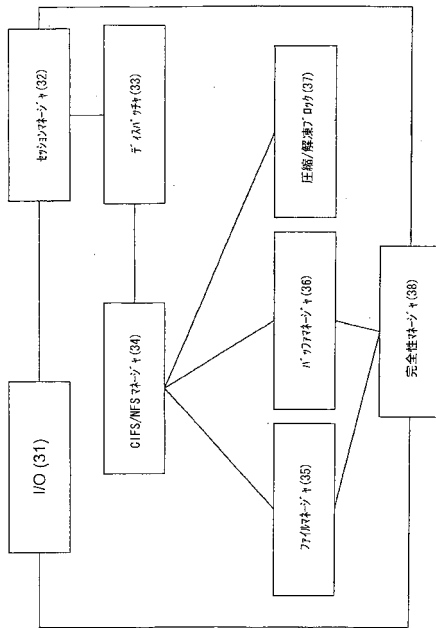
【図2a】



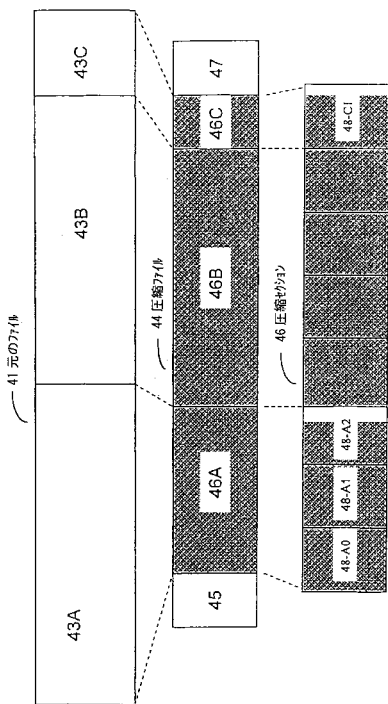
【図2b】



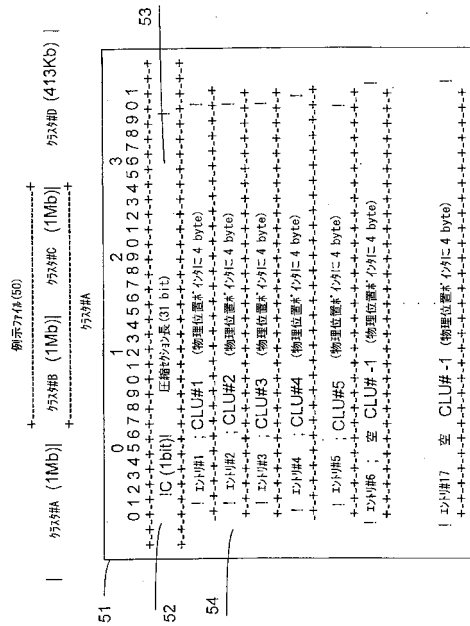
【図3】



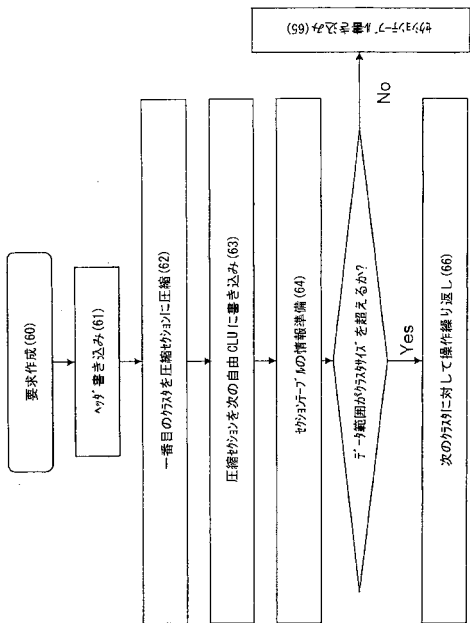
【図4】



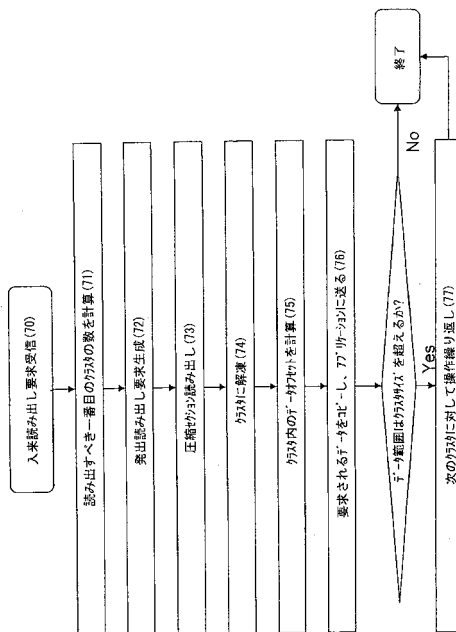
【図5】



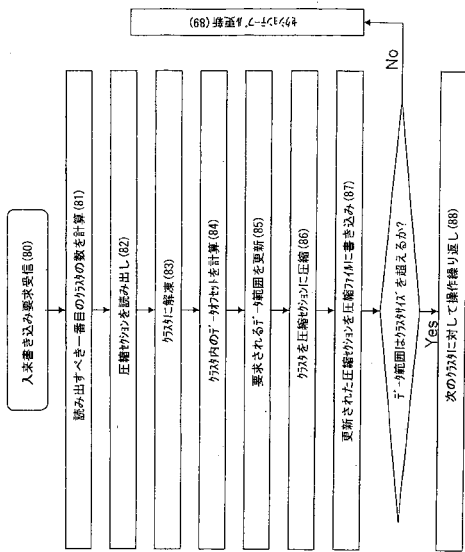
【図6】



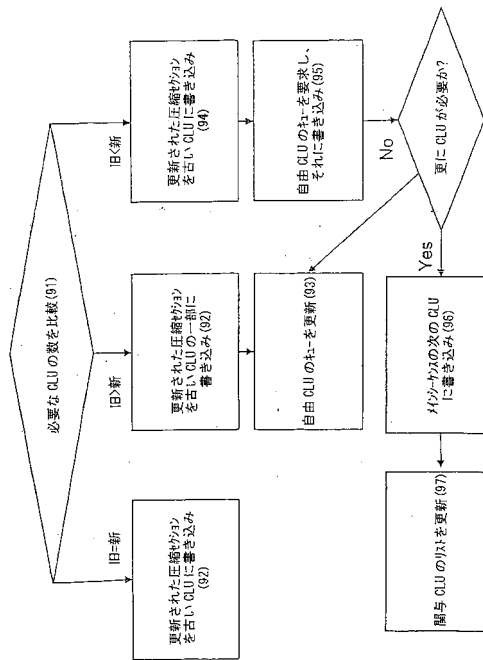
【図7】



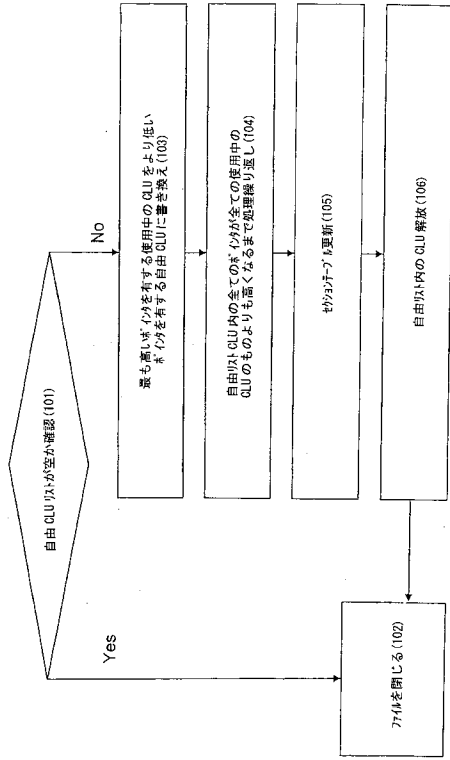
【図8】



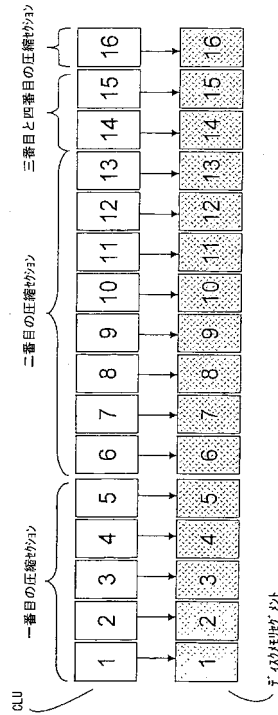
【図9】



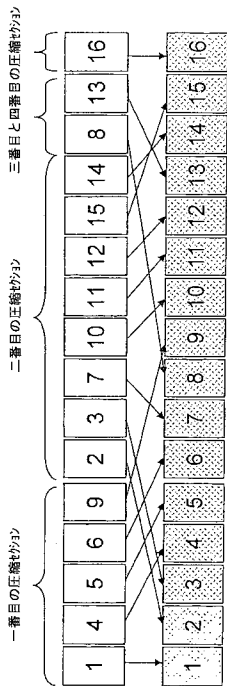
【図10】



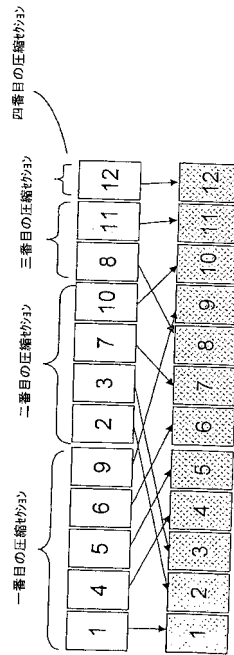
【図11a】



【図11b】



【図11c】



フロントページの続き

- (72)発明者 ケデム ナダブ
イスラエル国 69620 テル アビブ シャシャ アルゴブ ストリート 19
- (72)発明者 コーエン ヤコブ
イスラエル国 44833 エロン モルヘ ギヴァト ハブラハ 10

審査官 桜井 茂行

- (56)参考文献 特開平08-314689(JP,A)
米国特許第06577734(US,B1)
KAMPF F.A., "Performance as a function of compression", IBM SYSTEMS JOURNAL, 米国, IBM, 1998年, Volume 42, Number 6, Page 1-9
Pajarola et al, "An Image Compression Method for Spatial Search", IEEE Transactions on image processing, 米国, IEEE, 2000年, Volume 9, Number 3, Page 357-365
Wee-Keong et al, "Block-Oriented Compression Techniques for Large Statistical Database", IEEE Transactions on knowledge and data engineering, 米国, IEEE, 1997年, Volume 9, Number 2, Page 314-328

- (58)調査した分野(Int.Cl., DB名)
G06F 12/00