

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2009-533731

(P2009-533731A)

(43) 公表日 平成21年9月17日(2009.9.17)

(51) Int.Cl. F I テーマコード (参考)
G 0 6 F 1 2 / 0 0 (2006.01) G O 6 F 1 2 / 0 0 5 1 1 A 5 B O 8 2

審査請求 未請求 予備審査請求 未請求 (全 32 頁)

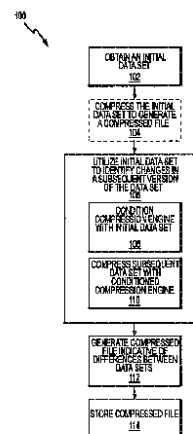
(21) 出願番号	特願2009-504506 (P2009-504506)	(71) 出願人	508299463
(86) (22) 出願日	平成19年4月9日 (2007.4.9)		データ ストレージ グループ
(85) 翻訳文提出日	平成20年10月30日 (2008.10.30)		DATA STORAGE GROUP
(86) 国際出願番号	PCT/US2007/066263		アメリカ合衆国 80503 コロラド州
(87) 国際公開番号	W02007/118243		ロングモント トレード センター ア
(87) 国際公開日	平成19年10月18日 (2007.10.18)		ベニュー 2400
(31) 優先権主張番号	60/744,477	(74) 代理人	100068755
(32) 優先日	平成18年4月7日 (2006.4.7)		弁理士 恩田 博宣
(33) 優先権主張国	米国 (US)	(74) 代理人	100105957
			弁理士 恩田 誠
		(74) 代理人	100142907
			弁理士 本田 淳
		(74) 代理人	100149641
			弁理士 池上 美穂

最終頁に続く

(54) 【発明の名称】 データ圧縮技術およびデータ格納技術

(57) 【要約】

データアーカイビングで使用するシステムおよび方法。データセット（例えば、ファイル、フォルダなど）のより早期のバージョン（102）を圧縮エンジンの辞書（108）として利用して（106）、データセットの後続のバージョンを圧縮する（110）圧縮技術が提供される。この圧縮（110）は、データセット間の変化を識別し、それらのデータセットの多くの共通の部分複製することなしに、これらの違いを格納すること（114）を可能にする。データセットの所与のバージョンに関して、新たな情報は、異なる時点で保存された個別の各セグメントから、そのバージョンを再構成するのに使用されるメタデータと共に格納される。より早期のデータセットおよび後続のデータセットの格納されたセグメントに対する1つまたは複数のリファレンスを利用して、後続のデータセットの再構成が可能となる。



【特許請求の範囲】**【請求項 1】**

データをアーカイブする / 格納する際に使用する方法であって、
第 1 のデータセットを受け取ること、
辞書ベースの圧縮エンジンを使用して前記第 1 のデータセットを圧縮し、前記第 1 のデータセットを表す第 1 の圧縮ファイルを生成すること、
前記第 1 の圧縮ファイルを格納すること、
前記第 1 の圧縮ファイルを使用して、前記第 1 のデータセットに関連する後続のデータセットにおける変化を識別すること、
を備える方法。

10

【請求項 2】

前記第 1 の圧縮ファイルを使用して前記変化を識別することは、
辞書ベースの圧縮エンジンを前記第 1 の圧縮ファイルで条件付けして、条件付けられた圧縮エンジンを定義すること、
前記条件付けられた圧縮エンジンを使用して前記後続のデータセットを圧縮し、前記後続のデータセットを表す第 2 の圧縮ファイルを生成すること、
を含む、請求項 1 に記載の方法。

【請求項 3】

前記第 2 の圧縮ファイルは、前記第 1 のデータセットと前記後続のデータセットとの間の変化部分と、前記第 1 の圧縮ファイルに対する 1 つまたは複数のリファレンス部分とからなる、請求項 2 に記載の方法。

20

【請求項 4】

前記第 2 の圧縮ファイルを格納することをさらに備える請求項 3 に記載の方法。

【請求項 5】

前記第 1 のデータセットに関連する少なくとも第 1 のハッシュを生成することを備える請求項 1 に記載の方法。

【請求項 6】

前記変化を識別することは、前記後続のデータセットの少なくとも 1 つのハッシュを、前記第 1 のデータセットの対応するハッシュと比較することをさらに含む、請求項 5 に記載の方法。

30

【請求項 7】

前記対応するハッシュ間の違いが前記比較によって識別され、当該方法はさらに、
辞書ベースの圧縮エンジンを前記第 1 の圧縮ファイルで条件付けして、条件付けられた圧縮エンジンを定義すること、
前記条件付けられた圧縮エンジンを使用して前記後続のデータセットを圧縮し、前記後続のデータセットを表す第 2 の圧縮ファイルを生成すること、
を備える、請求項 6 に記載の方法。

【請求項 8】

前記少なくとも第 1 のハッシュを生成することは、前記第 1 のデータセットのメタデータに関連する第 1 のハッシュを生成することを含む、請求項 5 に記載の方法。

40

【請求項 9】

前記第 1 のデータセットのコンテンツに関連する第 2 のハッシュを生成することをさらに備える請求項 8 に記載の方法。

【請求項 10】

前記後続のデータセットを前記圧縮エンジンの部分内にバッファリングすること、
前記後続のデータセットの前記バッファリングされた部分を、前記圧縮されたファイルが伸張された後に、前記圧縮されたファイルの同様のサイズの部分と実質的にマッチングさせること、
をさらに備える請求項 1 に記載の方法。

【請求項 11】

50

データを圧縮する方法であって、
ファイルを受け取ること、
前記ファイルの以前のバージョンが、以前に格納されていると判定すること、
前記ファイルの以前のバージョンから生成された圧縮辞書語を使用して前記ファイルを
圧縮し、圧縮ファイルを生成すること、
前記圧縮ファイルを格納すること、
を備える方法。

【請求項 1 2】

前記ファイルの前記以前に格納されたバージョンは、以前に圧縮されたファイルである
、請求項 1 1 に記載の方法。

10

【請求項 1 3】

圧縮エンジンに、前記以前に圧縮されたファイルを事前ロードすること、
受け取ったファイルを、前記圧縮エンジンの部分内にバッファリングすること、
前記受け取ったファイルの前記バッファリングされた部分を、前記以前に圧縮されたフ
ァイルが伸張された後、前記以前に圧縮されたファイルの同様のサイズの部分と実質的に
マッチングさせること、
をさらに備える請求項 1 2 に記載の方法。

【請求項 1 4】

前記判定する工程には、前記ファイルに関するバージョンリファレンスをハッシュする
ことによって得られたバージョンリファレンスハッシュを、以前に格納されたバージョン
リファレンスハッシュと合致させることに失敗すること、および前記ファイルに関する 2
次的リファレンスをハッシュすることによって得られた 2 次的リファレンスハッシュを、
以前に格納された 2 次的リファレンスハッシュと合致させることに成功することが付随す
る、請求項 1 1 に記載の方法。

20

【請求項 1 5】

前記ファイルに関連するバージョンリファレンスを生成して格納することをさらに備え
る請求項 1 1 に記載の方法。

【請求項 1 6】

前記ファイルに関連する 2 次的リファレンスを生成して格納することをさらに備える請
求項 1 5 に記載の方法。

30

【請求項 1 7】

前記バージョンリファレンスと前記 2 次的リファレンスはともにハッシュである、請求
項 1 6 に記載の方法。

【請求項 1 8】

データを圧縮する方法であって、
初期データセットを格納すること、
圧縮エンジンに、該圧縮エンジンに関する辞書を定義した前記初期データセットを事前
ロードすること、
前記事前ロード後の圧縮エンジンを使用して前記初期データセットに関連する後続のデ
ータセットを圧縮し、圧縮ファイルを生成すること、
前記圧縮ファイルを格納すること、
を備える方法。

40

【請求項 1 9】

前記圧縮ファイルは、前記初期データセットと前記後続のデータセットとの間の変化部
分と、前記初期データセットに対する 1 つまたは複数のリファレンス部分とからなる、請
求項 1 8 に記載の方法。

【請求項 2 0】

前記初期データセットと前記後続のデータセットとの間の関連付けを少なくとも 1 つの
メタデータ値に基づいて識別することをさらに備える請求項 1 8 に記載の方法。

【請求項 2 1】

50

データをアーカイブする / 格納する際に使用する方法であって、

第 1 のデータセットに関連する第 1 のシグネチャを生成することであって、

前記第 1 のデータセットのメタデータに関連する識別子ハッシュを生成すること、および

前記第 1 のデータセットのコンテンツに関連するコンテンツハッシュを生成することを含む、前記第 1 のデータセットに関連する第 1 のシグネチャを生成すること、

前記第 1 のシグネチャを格納すること、

前記第 1 のシグネチャのハッシュを、前記第 1 のデータセットに関連する後続のデータセットの対応するハッシュと比較すること、

前記比較に基づき、前記第 1 のデータセットと前記後続のデータセットとの間に変化が存在するかどうかを判定すること、
を備える方法。

10

【請求項 2 2】

前記比較することは、前記対応するハッシュの間に違いが存在するかどうかを判定することをさらに含む、請求項 2 1 に記載の方法。

【請求項 2 3】

前記対応するハッシュの間に違いが存在する場合、

前記第 1 のデータセットの前記後続のデータセットのコンテンツをハッシュして、後続バージョンコンテンツハッシュを生成すること、

前記後続バージョンコンテンツハッシュを、前記第 1 のデータセットの前記コンテンツハッシュと比較して、前記第 1 のデータセットと前記後続のデータセットとの間でコンテンツの変化が存在するかどうかを判定すること、
をさらに備える請求項 2 2 に記載の方法。

20

【請求項 2 4】

前記各コンテンツハッシュの間に違いが存在する場合、前記後続のデータセットの前記コンテンツを格納することをさらに含む請求項 2 3 に記載の方法。

【請求項 2 5】

前記後続のデータセットの前記コンテンツを格納することは、

辞書ベースの圧縮エンジンを前記第 1 のデータセットに関連するファイルで条件付けして、条件付けられた圧縮エンジンを定義すること、

30

前記条件付けられた圧縮エンジンを使用して前記後続のデータセットを圧縮し、前記第 1 のデータセットと前記後続のデータセットとの間の変化を含む前記後続のデータセットを表す圧縮ファイルを生成すること、
を含む、請求項 2 3 に記載の方法。

【請求項 2 6】

前記第 1 のシグネチャは第 1 のネットワークロケーションに格納され、前記後続のデータセットは第 2 のネットワークロケーションに位置づけられ、当該方法はさらに、

前記第 1 のネットワークロケーションから前記第 2 のネットワークロケーションに前記シグネチャを転送すること、

前記第 1 のシグネチャの少なくとも 1 つのハッシュを、前記第 2 のネットワークロケーションにおける前記後続のデータセットの対応するハッシュと比較すること、
を備える、請求項 2 1 に記載の方法。

40

【請求項 2 7】

前記ハッシュ間の違いを前記比較によって識別した場合、前記後続のデータセットの少なくとも一部分を、格納のために前記第 1 のネットワークロケーションに転送することをさらに備える請求項 2 6 に記載の方法。

【請求項 2 8】

前記転送することは、ネットワークインタフェースを介して前記第 1 のシグネチャを転送することを含む、請求項 2 6 に記載の方法。

【請求項 2 9】

50

前記識別子ハッシュを生成することは、メタデータ構成要素の第 1 および第 2 のセットにそれぞれ関連する第 1 および第 2 の識別子ハッシュを生成することをさらに含み、前記メタデータ構成要素の前記第 1 のセットと前記第 2 のセットとは異なる、請求項 2 1 に記載の方法。

【請求項 3 0】

前記後続のデータセットが前記第 1 のデータセットに関連していると識別することは、前記第 1 のデータセットの前記第 1 の識別子ハッシュと前記第 2 の識別子ハッシュのうちの少なくとも 1 つを、前記後続のデータセットの対応するハッシュとマッチングさせることを含む、請求項 2 9 に記載の方法。

【請求項 3 1】

前記第 1 の識別子ハッシュおよび前記第 2 の識別子ハッシュのうちの一方は前記第 1 のデータセットのストレージロケーションに関連付けられ、前記第 1 の識別子ハッシュおよび前記第 2 の識別子ハッシュのうちの他方は、前記第 1 のデータセットのバージョンリファレンスに関連付けられる、請求項 2 9 に記載の方法。

【請求項 3 2】

データ格納の際に使用する方法であって、

初期データセットをストレージロケーションに転送すること、

前記初期データセットに関連するシグネチャを生成することであって、

前記初期データセットのメタデータに関連する識別子ハッシュセットを生成すること、および

前記初期データセットのコンテンツに関連するコンテンツハッシュセットを生成すること

を含む、前記初期データセットに関連するシグネチャを生成すること、

前記初期データセットを転送することの後に続いて、前記初期データセットに関連する後続のデータセットのバックアップを実行することであって、

前記後続のデータセットのメタデータに関連する識別子ハッシュセットを生成すること、

前記初期データセットの識別子ハッシュセットと前記後続のデータセットの識別子ハッシュセットとを比較すること、

前記比較に基づいて、前記後続のデータセットに関する合致しない識別子ハッシュを識別すること、および

前記合致しない識別子ハッシュに関連するコンテンツの少なくとも一部分を前記ストレージロケーションに転送すること、

を含む、前記初期データセットに関連する後続のデータセットのバックアップを実行すること、

を備える方法。

【請求項 3 3】

前記初期データセットの識別子ハッシュと合致する前記後続のデータセットの識別子ハッシュに関連するコンテンツは、前記ストレージロケーションに転送されない、請求項 3 2 に記載の方法。

【請求項 3 4】

前記後続のデータセットの少なくとも 1 つの合致しないハッシュに関連するコンテンツに関するコンテンツハッシュを生成すること、

前記初期データセットに関して、合致するコンテンツハッシュが存在するかどうかを判定すること、

をさらに備える請求項 3 2 に記載の方法。

【請求項 3 5】

前記初期データセットに関して、前記合致するコンテンツハッシュが存在する場合、前記ストレージロケーションへの前記コンテンツの転送を差し控えること、

前記初期データセットに関して、合致するコンテンツハッシュが存在しない場合、前記

10

20

30

40

50

ストレージロケーションに前記コンテンツを転送すること、
をさらに備える請求項 34 に記載の方法。

【請求項 36】

前記初期データセットは、ネットワークを介して起源ロケーションから前記ストレージロケーションに転送される、請求項 32 に記載の方法。

【請求項 37】

前記シグネチャは、前記ストレージロケーションに生成される、請求項 36 に記載の方法。

【請求項 38】

前記ストレージロケーションから前記後続のデータセットの起源ロケーションに前記シグネチャを転送することをさらに備え、前記ハッシュの比較は、前記起源ロケーションにおいて実行される、請求項 36 に記載の方法。

10

【請求項 39】

データ格納の際に使用する方法であって、

格納のために第 1 のデータセットを受け取ること、

前記第 1 のデータセットのメタデータに関連する識別子ハッシュのセットを生成すること、

前記第 1 のデータセットのコンテンツに関連するコンテンツハッシュのセットを生成すること、

前記第 1 のデータセットに関連する後続のデータセットをバックアップする目的で、前記第 1 のデータセットに関連するデータ起源ロケーションに前記識別子ハッシュおよび前記コンテンツハッシュを供給すること、

20

前記供給された識別子ハッシュおよびコンテンツハッシュと合致しなかった前記後続のデータセットからデータを受け取ること、
を備える方法。

【請求項 40】

前記後続のデータセットからの前記データを格納することをさらに備える請求項 39 に記載の方法。

【請求項 41】

前記後続のデータセットに関する識別子ハッシュおよびコンテンツハッシュを生成することをさらに備える請求項 40 に記載の方法。

30

【請求項 42】

前記受け取る工程、および前記供給する工程は、直接データ接続を介して実行される、請求項 39 に記載の方法。

【請求項 43】

前記受け取る工程、および前記供給する工程は、ネットワーク接続を介して実行される、請求項 39 に記載の方法。

【請求項 44】

データ格納の際に使用する方法であって、

以前に格納されたデータセットのメタデータに関連する識別子ハッシュのセットを受け取ること、

40

前記識別子ハッシュを現在のデータセットの識別子ハッシュと比較することであって、前記現在のデータセットの少なくとも一部分は、前記以前に格納されたデータセットの後続のバージョンである、比較すること、

前記識別子ハッシュのセット内における識別子ハッシュと合致しない前記現在のデータセットの合致しない識別子ハッシュを識別すること、

前記合致しない識別子ハッシュに関連するコンテンツの少なくとも一部分をストレージロケーションに送ること、
を備える方法。

【請求項 45】

50

以前に格納されたデータセットのコンテンツに関連するコンテンツハッシュのセットを受け取ること、

前記合致しないハッシュの前記コンテンツに関連するコンテンツハッシュを、前記以前に格納されたデータセットのコンテンツハッシュと比較すること、
を備え、

前記送ることは、合致しない識別子ハッシュと、合致しないコンテンツハッシュとを有するコンテンツを送ることを含む、請求項 4 4 に記載の方法。

【請求項 4 6】

前記受け取る工程、および前記送る工程は、直接データ接続を介して実行される、請求項 4 4 に記載の方法。

【請求項 4 7】

前記受け取る工程、および前記送る工程は、ネットワーク接続を介して実行される、請求項 3 9 に記載の方法。

【請求項 4 8】

データ格納の際に使用する方法であって、

関連するデータセットに関する複数のカタログを複数の時点において生成することであって、複数のカタログの各々は、メタデータが格納されたセグメントに対するメタデータリファレンスと、コンテンツが格納されたセグメントに対するコンテンツリファレンスとを含む、複数のカタログを生成すること、

或る時点に関するカタログを選択し、前記或る時点に関連するファイルを識別すること、

前記或る時点に関して、少なくとも前記ファイルに関する前記メタデータリファレンスおよび前記コンテンツリファレンスを識別すること、

前記格納されたセグメントを獲得すること、

前記ファイルを再構成すること、

を備える方法。

【請求項 4 9】

前記ファイルは、少なくとも 2 つのコンテンツリファレンスを含む、請求項 4 8 に記載の方法。

【請求項 5 0】

前記コンテンツリファレンスは、異なる時点で格納されたコンテンツのリファレンスセグメントを参照する、請求項 4 9 に記載の方法。

【請求項 5 1】

仮想ファイルシステムの中の前記複数のカタログを表示することをさらに備える請求項 4 8 に記載の方法。

【請求項 5 2】

前記仮想ファイルシステムは、前記データセットのルートファイルシステムを模倣する前記ルートファイルシステムの中のファイルを表す、請求項 5 1 に記載の方法。

【請求項 5 3】

前記リファレンスは、ハッシュコードからなる、請求項 4 8 に記載の方法。

【請求項 5 4】

前記複数のカタログの少なくとも 2 つは、前記カタログ間で共通なコンテンツに関する共通のコンテンツセグメントを参照する、請求項 4 8 に記載の方法。

【請求項 5 5】

データ格納の際に使用する方法であって、

第 1 の時点において、第 1 のデータセットのメタデータの 1 つまたは複数のロケーションを識別するリファレンスと、前記第 1 のデータセットのコンテンツの 1 つまたは複数のロケーションを識別するリファレンスとを含む、前記第 1 のデータセットの第 1 のカタログを生成すること、

第 2 の時点において、前記第 1 のデータセットのメタデータの 1 つまたは複数のロケー

10

20

30

40

50

ションを識別するリファレンスと、前記第 1 のデータセットのコンテンツの 1 つまたは複数のロケーションを識別するリファレンスとを含む、前記データセットの後続のバージョンの第 2 のカタログを生成すること、
を備え、前記第 1 のカタログと前記第 2 のカタログは、共通のコンテンツに関する少なくとも 1 つのリファレンスを共有する、方法。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、デジタルデータを格納することに関する。より具体的には、本出願は、いくつかの態様が、データアーカイブにおいて応用されるデジタルデータのより効率的な格納に使用するためのユーティリティに関する。

10

【背景技術】

【0002】

(関連出願の相互参照)

本出願は、内容の全体を本明細書に援用する 2006 年 4 月 7 日の出願日を有する「Content Factoring for Long Term Digital Archiving」という名称の米国仮出願第 60/744,477 号からの優先権を主張する、2007 年 4 月 9 日の出願日を有する「Data Compression and Storage Techniques」という名称の米国出願第 11/733,086 号の優先権を主張する。

20

【0003】

組織は、長期のデータ保持要件を満たす際に新たな課題に直面しており、IT 専門家は、州および連邦政府の多くの新たな規制およびガイドラインの遵守を維持する責任を有する。これらの規制は、組織が、過去に、使用可能な仕方が必要な情報を利用可能にしておくことに苦労したために存在する。この問題を悪化させているのが、デジタル情報の継続的な爆発的増加である。ドキュメントは、コンテンツがより豊かであり、しばしば、関連する仕事を参照し、管理されるべき途方もない量の情報をもたらす。

【0004】

基礎をなすアクセスパターンをよりよく理解するため、デジタル情報の分類を最初に簡単に説明することが役立つ。すべてのデジタル情報のコレクションは、一般に、構造化されたもの、または構造化されていないものとして分類可能である。構造化された情報とは、リレーショナルデータベース内部に保持されるデータを指す。構造化されていない情報は、その他すべて、すなわちドキュメント、イメージ、映画などである。構造化されたデータと構造化されていないデータはともに、ユーザまたはアプリケーションによって活発に参照される、または将来の参照または遵守のために変更されないままに保たれる。構造化された情報、および構造化されていない情報のうち、活性の情報は、定常的に参照される、または変更されるのに対して、非活性の情報は、時々しか参照されない、または将来の何からの時点で参照される可能性を有するに過ぎない。情報が活性である、または非活性である特定の時間枠は、純粋に主観的である。

30

【0005】

デジタル情報の下位分類が、動的または固定としてデータの可変性を表す。トランザクションデータベース内のレコードなどの動的コンテンツは、頻繁に、または継続的に変化する。スキャンされるチェックイメージまたは電子メールメッセージなどの固定のコンテンツは、静的な読み取り専用の情報であり、つまり、作成され、決して変更されない。長期アーカイピングに関して、構造化された、または構造化されていない非活性の情報は、常に固定コンテンツを有すると考えられ、変化しない。

40

【0006】

時とともに、情報は、それほど頻繁にアクセスされなくなる傾向があり、アクセスパターンは、より読み取り専用になる傾向がある。固定コンテンツの読み取り専用情報は、アーカイピングの点で管理するのが比較的簡単明瞭である。もちろん、サブファイルレベル

50

においてさえ、構造化された、または構造化されていない動的情報が、静的である大きいコンテンツセグメントを含む可能性がある。このタイプの情報の例には、コンテンツが追加されているデータベースファイル、および編集されるドキュメントが含まれる。

【 0 0 0 7 】

固定であれ、動的であれ、デジタル情報のタイプにかかわらず、多くの組織は、組織のデジタルデータを固定でバックアップする。例えば、多くの組織は、すべてのデジタルデータが複製される毎週のバックアップを実行する。さらに、これらの組織の多くは、毎日のインクリメンタルバックアップを実行して、日々のデジタルデータの変更が格納されるようにする。しかし、従来のバックアップシステムは、いくつかの欠点および非効率を有する。例えば、すべてのデジタルデータが複製される毎週のバックアップ中、変更されていない固定のファイルが、複製される。理解され得るとおり、このことは、デジタル情報の不必要な冗長性ととも、処理要件および / または帯域幅要件の増大をもたらす。毎週のバックアップとインクリメンタルバックアップの両方に関する別の問題は、動的ファイルのわずかな変更が、デジタルデータの非効率な複製をもたらす可能性があることである。例えば、10MBファイルの1文字の編集により、そのファイルのコンテンツ全体がバックアップされ、カタログされることが要求される。状況は、Outlook Personal Folders (.pstファイル)などのより大きいファイルに関してはさらに悪く、これらのファイルを開く動作そのものにより、これらのファイルが変更されることになり、すると、このことが、別のバックアップを要求する。

【 0 0 0 8 】

これらの欠点および非効率の通常の結果は、大量のバックアップデータの生成であり、最も一般的なバックアップシステムにおいて、複数のデータストレージテープの生成である。これに関して、非効率なバックアップは、複数のバックアップテープの生成をもたらす。次に、これらのテープが、格納されなければならない。通常、そのようなテープは、オフラインで格納される。つまり、それらのテープは、コンピュータ化されたアクセスが即時に利用可能でない場所に格納される。したがって、バックアップテープから情報を回復することは、アーカイビング施設と連絡を取り、テープを特定し、その施設が、そのテープを探し出して、ロードするのを待つことを要求する可能性がある。

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 0 9 】

ディスクストレージの価格が下がってきているなかで、ディスクバックアップを利用して、テープバックアップの問題を軽減しようとする試みが行われてきた。しかし、これらのディスクバックアップは、依然として、データの非効率な複製を見込む大量のストレージを要求する。したがって、デジタルデータの前のバックアップと現在のデジタルデータセットとの間に生じた動的変化を識別する試みが行われてきた。これに関して、目標は、前のデジタルデータセットに対して変更されているデータ（すなわち、動的データ）のバックアップに限って作成することである。

【 0 0 1 0 】

データバックアップの合間の動的変化を識別して、それらの動的変化だけを格納しようとする1つの試みが、COS（容量最適化ストレージ）によって代表される。COSの目標は、バックアップセット間の冗長性を重複除外することである。つまり、COSの目標は、現在のデータセットを以前に格納されたデータセットと比較して、新たなデータだけを格納することである。一般に、COS処理は、デジタルデータ（例えば、第1のバックアップコピー）のセット全体をデータチャンク（例えば、256kB）に分割して、それらのデータチャンクにハッシュアルゴリズムを適用することである。当分野の技術者によって理解されたとおり、このことは、ハッシュコード / アルゴリズムに従ってデータを表すキーアドレスをもたらす。バックアップのための新たなデータセット（例えば、第2のバックアップコピー）が受け取られると、データセットは、データチャンクに再び分割され、ハッシュアルゴリズムが適用される。理論上、第1のデータセットと第2のデータセ

ットの間の対応するデータチャンクが同一である場合、バックアップの合間に変更は、全くなかったものと想定される。したがって、第1のバックアップセットとは異なるチャンクだけが、保存されて、後続のバックアップに関するストレージ要件が低減される。C O Sの主な欠点は、バックアップセット間の冗長性を大幅に減らすのに、ますます小さいデータチャンクを利用することが望ましいことである。しかし、データチャンクのサイズが縮小されるにつれ、キーアドレスの数は、増加する。したがって、より多くのキーアドレスの格納および索引付けが、より少ない量の重複データの利益を消すように作用する。

【0011】

C O S処理の使用により、ディスクアクセス可能なデータバックアップの作成が可能になり、バックアップされたデータセットへのより容易なアクセスが可能になる。これに関して、C O Sは、仮想テーブライブラリV T Lがテープストレージデバイスをエミュレートするように、仮想テーブライブラリV T Lに組み込まれる。システムは、ユーザが、バックアップのためにオフサイトのディスクストレージセンタにデータを送信することを可能にする。しかし、このことは、データセット全体が、V T Lに伝送されることを要求し、V T Lにおいて、このデータセット全体が、格納のために最適化される（例えば、C O S）。さらに、後続の各バックアップに関して、そのデータセット全体が、オフサイトのストレージセンタに再び伝送されなければならない。理解され得るとおり、バックアップを要求する大きいデータセットを有する大きい組織に関して、データセット全体の伝送を要求するようなオフサイトのストレージシステムには、データを転送する大きい帯域幅要件、およびデータを最適化し、比較する高い処理要件がかかわる可能性がある。最後に、オフサイトV T Lを利用する組織は、組織のデータの復元に関してバックアップアプリケーションに100%依存し、やはり、ユーザは、潜在的に、不慮の削除またはディスク破損の場合に情報が利用できない可能性にさらされたままとなる。

【課題を解決するための手段】

【0012】

既存の短期データ保護ソリューションは、法外な費用がかかり、アーカイブされた情報への向上したアクセスを可能にすることにほとんど効果がない。本明細書で説明されるアーカイブ技術は、情報を管理することの長期ソリューションを提供するとともに、ディスクベースのアーカイブにおいて利用可能なソリューションを提供する。これらの技術は、既存のディスクリソースを使用して、アーカイブされた情報のコレクションへの容易なアクセスを提供する。この技術が、オープンアーキテクチャオブジェクトベースのコンテンツストアと相俟って、既存の短期データ保護手続きを全く変更することなしに、ディスクベースのシステムの有効容量の大幅な増加（例えば、20:1）を可能にする。

【0013】

さらに、コンテンツの長期ストレージをよりうまく最適化するのに、これらの新たな技術は、所与のデータセットに関して格納される冗長な情報を減らす。適応コンテンツファクタリングは、固有のデータにキーが付けられ、1回格納される、本発明者らによって開発された技術である。従来のコンテンツファクタリング技術または適応差分技術とは異なり、適応コンテンツファクタリングは、発見的方法を使用して、格納される各データ量子のサイズを最適化する。適応コンテンツファクタリングは、データ圧縮と関係するが、局所化されたコンテンツに限定されない。データセットの所与のバージョンに関して、新たな情報が、異なる時点で保存された個別の各セグメントからのバージョンを再構成するのに使用されるメタデータと一緒に格納される。このメタデータ-再構成段階は、通常のファイルシステムがI/O要求に応える際に行うことと同様である。

【0014】

本発明の第1の態様により、データを格納する方法およびシステム（ユーティリティ）が提供される。このユーティリティは、第1のデータセットを受け取ること、および辞書ベースの圧縮エンジンを使用して第1のデータセットを圧縮することが必然的に伴う。そのような圧縮により、第1のデータセットを表す第1の圧縮ファイルが生成される。次に、この第1の圧縮ファイルが格納される。次に、この第1の圧縮ファイルを利用して、第

1 のデータセットの後続のバージョンにおける変化が識別される。本明細書で利用される「データセット」には、限定なしに、個々のデータファイル、ならびに複数のデータファイルを含むフォルダ、および / または複数のフォルダを含むことが可能なドライブが含まれることが意図される。そのような事例においては、第 1 のデータセットを圧縮することにより、対応する複数の第 1 の圧縮ファイルが生成される。

【 0 0 1 5 】

一つの構成では、第 1 の圧縮ファイルを使用して変化を識別することは、辞書ベースの圧縮エンジンに第 1 の圧縮ファイルを事前ロードして、条件付き圧縮エンジンを定義することを含む。つまり、第 1 の圧縮ファイルは、圧縮エンジンにロードされることにより、圧縮エンジンのための辞書を定義する。第 1 のデータセットと後続のデータセットが実質的に同様である場合、圧縮エンジンに関する辞書として第 1 のデータセットを使用することは、圧縮率の高い第 2 のデータセットをもたらす。このため、ユーティリティは、第 1 のデータセットの後続のバージョンを、条件付き圧縮エンジンを使用して圧縮することを含む。これに関して、第 1 のデータセットの後続のバージョンを示す第 2 の圧縮ファイルが生成される。この第 2 の圧縮ファイルは、後続のデータセットと第 1 のデータセットの間の変化をやはり示す。さらに、第 2 の圧縮ファイルは、第 1 の圧縮ファイルに対する 1 つまたは複数のリファレンスを含む。第 2 の圧縮ファイルは、第 1 の圧縮ファイルよりも相当に小さくなる。より早期の 1 つまたは複数のデータセットを、辞書ベースの圧縮エンジンに関する辞書として利用することにより、複数の後続のデータセットを圧縮することが可能となる。

【 0 0 1 6 】

第 1 のデータセットの対応する部分を、第 2 のデータセットの対応する部分（例えば、対応するファイル）と同一視するため、ユーティリティは、データセットの個々の 1 つまたは複数の部分に関する識別子情報を生成することをさらに必然的に伴う。例えば、ハッシュコード情報（本明細書で「ハッシュ情報」および「ハッシュ」または「複数のハッシュ」とも呼ばれる）が、データセットの個々の部分に関して生成される。さらに、そのようなハッシュ情報が、データセットの個別の各部分の個々の構成要素に関して生成される。一つの構成において、1 つまたは複数のハッシュコードが、所与のファイルに関連するメタデータに関連し、別のハッシュコードが、ファイルのコンテンツに関して生成される。したがって、そのようなハッシュコードを利用して、圧縮の目的で、第 1 のデータセットと後続のデータセットの対応する部分を識別することが可能である。対応するハッシュコードが、後続のデータセットのいくつかの部分に関して存在する場合、通常の圧縮方法が、後続のデータセットのそれらの部分に対して利用することが可能となる。

【 0 0 1 7 】

別の態様によれば、データを圧縮するためのシステムおよび方法（ユーティリティ）が提供される。ユーティリティは、ファイルを受け取ること、およびこのファイルの前のバージョンが以前に格納されていると判定することを含む。そのような判定が行われると、ファイルを、そのファイルの前のバージョンから生成された圧縮辞書語を使用して圧縮することが可能となる。こうして、受け取られたファイルに関する圧縮ファイルが生成される。次に、この圧縮ファイルが格納される。圧縮辞書語は、ファイルの前のバージョンから、またはファイルの前のバージョンの圧縮バージョンから生成される。いずれの構成においても、ユーティリティは、ファイルの前のバージョンを圧縮エンジンに事前ロードすること、および受け取られたファイルを、圧縮エンジンを使用していくつかの部分に分けてバッファリングすることを含むことが可能である。そうすることにより、受け取られたファイルのバッファリングされた部分と、前のファイルの同様のサイズの部分との実質的なマッチングが可能となる。

【 0 0 1 8 】

ファイルの前のバージョンが以前に格納されているという判定は、任意の適切な方法で行われる。例えば、ファイルは個別に保存されて、ユーザが、バックアップ手続き中にファイルの以前に格納されたバージョンを選択することが可能である。別の構成において、

バージョンリファレンスに関連する（例えば、ファイルのメタデータに関連する）ハッシュを利用することで、ファイル間の関係を判定することが可能である。一つの構成において、第1のハッシュおよび第2のハッシュは、以前に格納されたファイルのメタデータ、および受け取られたファイルのメタデータに関連する。そのような構成において、ファイルの対応する第1のハッシュが合致する（例えば、ストレージロケーションに対応して）一方で、ファイルの第2の対応するハッシュ（例えば、バージョンリファレンス）は合致しない。これに関して、これらのファイルは、関係するが、これらのファイル間に変更があると判定される。したがって、バックアップ目的で量を減らすために、前のファイルを利用して後続のファイルを圧縮することが望ましい可能性がある。

【0019】

10

別の発明の態様によれば、データをアーカイブする際、および/または格納する際に使用するシステムおよび方法（ユーティリティ）が提供される。ユーティリティは、データセットに関する個別のシグネチャを生成することが必然的に伴う。このシグネチャが、後続のデータセットと比較されることにより、対応する部分、または同様の部分が識別される。そのため、それらのデータセット間の違いを識別することが可能となる。したがって、データセットの同様の部分は、バックアップ手続きにおいてコピーされる必要がない。そうではなく、後続のデータセットの新たな部分（例えば、違い）だけが、アーカイビング/バックアップ目的でコピーされればよい。

【0020】

20

一つの態様において、ユーティリティは、第1のデータセットに関連する第1のシグネチャを生成することを含む。第1のシグネチャを生成することは、第1のデータセットのメタデータに関連する第1のハッシュセット（例えば、ハッシュコード）を生成することを含む。さらに、第1のデータセットに関して、第1のデータセットのコンテンツに関連するコンテンツハッシュのセットが生成される。例えば、データセットの中の個別の各ファイルは、メタデータに関連する第1のハッシュ（例えば、識別子ハッシュ）と、そのファイルのコンテンツに関連する第2のハッシュ（例えば、コンテンツハッシュ）とを含む。これらが生成されると、第1のハッシュと、コンテンツハッシュとを含むシグネチャを個々に、および/または組み合わせで利用して、第1のデータセットと後続のデータセットとの間の変化が識別される。例えば、第1のデータセットの識別子ハッシュが、後続のデータセットの対応するハッシュと比較される。そのような比較に基づき、第1のデータセットと後続のデータセットの1つまたは複数の部分の間で変化が存在すると判定される。つまり、第1のデータセットと第2のデータセットの1つまたは複数の部分の間で変化が存在するかどうか判定される。

30

【0021】

一つの構成において、第2のデータセットの識別子ハッシュが、第1のデータセットの識別子ハッシュと合致しない場合、この合致しない識別子ハッシュに関連するコンテンツが、第1のデータセットのコンテンツと比較される。より詳細には、そのコンテンツをハッシュすることによって得られたコンテンツハッシュコードが、第1のデータセットに関連するコンテンツハッシュコードと比較される。これに関して、コンテンツの識別子が第1のデータセットにおける識別子と合致しない場合でさえ、そのコンテンツが第1のデータセットの中に既に存在するかどうかを判定する第2の検査が実行される。そのコンテンツハッシュコードが存在する場合、そのコンテンツは、ストレージロケーションに伝送されない、またはそれ以外で格納されない可能性がある。合致しない識別子ハッシュのコンテンツハッシュコードが、第1のデータセット内のコンテンツハッシュコードと合致しない場合、そのコンテンツは、ストレージロケーションに格納される。

40

【0022】

一つの構成において、メタデータに関連する識別子ハッシュは、第1の識別子ハッシュと、第2の識別子ハッシュとを含む。これらのハッシュのそれぞれは、メタデータのいくつかの部分に関連する。例えば、これらのハッシュの1つは、他のハッシュの下位部分であり得る。これに関して、データセットの間で、より細かい比較が行われて、データセッ

50

ト間の変化が識別される。

【 0 0 2 3 】

さらなる発明の態様において、アーカイビング目的の分散処理を可能にするためのシステムおよび方法（ユーティリティ）が提供される。これに関して、データセット全体をアーカイブロケーションに転送するのではなく、アーカイブデータセットと現在のデータセットとの間の変化の識別が、現在のデータセットのロケーション（例えば、データ起源ロケーション）において実行される。したがって、アーカイブロケーションに送られる可能性がある情報は、以前に格納されたデータセットと現在のデータセットとの間の違いだけである。

【 0 0 2 4 】

一つの態様によれば、格納のための第1のデータセットが受け取られる（例えば、アーカイブ/バックアップロケーションにおいて）。したがって、第1のデータセットのメタデータに関連する識別子ハッシュのセットが生成される。また、同様に、第1のデータセットのコンテンツに関連するコンテンツハッシュのセットも生成される。第1のデータセットに関連する現在のデータセットをバックアップすることが必要になると、これらの識別子ハッシュおよびコンテンツハッシュが、第1のデータセットに関連するデータ起源ロケーションに供給される。データ起源ロケーションにおいて、これらのハッシュを利用して、第1のデータセットと後続のデータセットとの間の変化が特定されて、これらの変化が、ストレージロケーションに転送される。これに関して、ユーティリティには、供給された識別子ハッシュおよび/またはコンテンツハッシュのいずれか、または両方と合致しなかった後続のデータセットからデータを受け取ることも必然的に伴う。そのような時点で、新たに受け取られたデータがハッシュされ、そのハッシュ情報が、後続のバックアップの目的で既存のハッシュ情報に追加される。

【 0 0 2 5 】

別の態様によれば、以前に格納されたデータセットのメタデータに関連する識別子ハッシュのセットが受け取られるユーティリティが提供される。これらの識別子ハッシュは、現在のデータセットの識別子ハッシュと比較される。このデータセットの少なくとも一部分が、以前に格納されたデータセットの後続のバージョンを形成する。これらの識別子ハッシュを比較することにより、現在のデータセットの合致しない識別子ハッシュを識別することが可能になる。したがって、合致しない識別子ハッシュに関連するコンテンツの一部分または全部が、ストレージロケーションに送られる。

【 0 0 2 6 】

さらなる構成において、ユーティリティは、以前に格納されたデータセットのコンテンツに関連するコンテンツハッシュのセットを受け取ることをさらに含む。そのような構成において、現在のデータセットの合致しないハッシュのコンテンツに関連するコンテンツハッシュが、以前に格納されたデータセットのコンテンツハッシュと比較される。したがって、そのような構成において、識別子ハッシュも、コンテンツハッシュも、以前に格納されたデータセットのハッシュに対応しない場合、その合致しないコンテンツが、ストレージロケーションに送られる。

【 0 0 2 7 】

前述の2つの態様において、送信する/供給する工程、および/または受け取る工程は、例えば、コンピュータとストレージロケーション（例えば、直接に接続されたストレージ、取り外し可能なハードドライブ、または他のポータブルストレージデバイス）との間の直接接続によって、あるいはネットワーク接続によって実行される。ネットワーク接続によって実行される場合、そのようなネットワーク接続には、ワイドエリアネットワーク、インターネット、直接接続されたストレージネットワークおよび/またはピアコンピュータが含まれる。

【 0 0 2 8 】

さらなる態様において、データセットの複数の異なるバージョン（例えば、連続したバージョン）を格納するための、およびそのようなバージョンへのアクセスを提供するため

10

20

30

40

50

のシステムおよび方法が提供される。このユーティリティには、異なる時点で異なるデータセットのカatalogを生成することが含まれる。各Catalogは、或る特定の時点における関連するデータセットを再構成するのに必要とされる情報を含む。つまり、或る時点に関する或る特定のデータセットの完全なコピーを生成するのではなく、このユーティリティは、所与のデータセットを再構成するのに要求されるデータのロケーションに対するリファレンスを有するCatalogを生成する。

【0029】

一つの構成において、Catalogは、異なるデータストリーム（例えば、ファイルの構成要素）に関する様々なハッシュコードを含む。これらのハッシュコードは、Catalog内で所与のファイルの構成要素を識別し、探し出すことができるようにする。したがって、これらの構成要素は、Catalogが生成された際に存在していた形態でファイルを形成するように再構成される。別の言い方をすると、所与のファイルのデータを格納するのではなく、Catalogは、そのファイルに関連するデータのロケーションに対するリファレンスを格納して、ファイルの構成要素を複製することが、必ずしも必要でないようにする。さらに、所与のCatalogの格納されたリファレンスは、異なる時点で保存されている所与のファイルの異なるセグメントを参照可能である。

【0030】

例示的な実施形態が図に示される。本明細書で開示される実施形態および図は、限定するものではなく、例示的であるものと考えられることが意図されている。

【発明を実施するための最良の形態】

【0031】

次に、添付の図面を参照して、本発明の様々な特徴を説明する。本発明は、ここでは、主として、電子データのアーカイビング/バックアップストレージに関連して説明されるものの、本発明は、本明細書に含まれる発明の目的を実現することが所望される他の応用例にも適用可能である。つまり、本発明は、任意のデータストレージ環境において利用することが可能である。これに関して、アーカイビングのための使用の以下の説明は、例示および説明の目的で提示される。さらに、この説明は、本発明を、本明細書で開示される形態に限定することを意図していない。したがって、以下の教示、および当該技術分野の技能および知識に相応する変種および変形は、本発明の範囲に含まれる。本明細書で説明される実施形態は、本発明を実施する当業者が、本発明を、そのような実施形態において、または他の実施形態において、本発明の特定の応用例または用法によって要求される様々な変形とともに利用することができるようにすることをさらに意図している。

【0032】

アーカイビングの目的でバックアップ工程および復元工程だけを厳密に使用することは、ほとんどの規制された環境に関して容認できない。ディスクベースのバックアップ環境に関して、従来の方法を使用することは、一般に、法外な費用がかかる。より高い利用可能性に対処し、ディスクストレージの費用を最小限に抑える2つの一般的な方法は、HBDL（ハードウェアベースのディスクライブラリ）またはVTL（仮想テープライブラリ）を組み込むことである。いずれのソリューションも、データ冗長性問題に対処せず、これらのソリューションは、全体的なTCO（総所有コスト）を低減することにほとんど効果がない。

【0033】

IT組織によって採用される代替のアプローチは、ボリュームシャドウコピーサービスなどのブロックレベルスナップショット技術、または類似したハードウェアベンダによって提供されるスナップショット技術を使用することである。この場合、所与の回復ポイントに関して、変更されたブロックが記録される。しかし、これらのシステムは、通常、指定された数のスナップショットの後、またはボリューム容量閾値に達すると、リセットされる（ロールオーバーする）。すべての場合において、ブロックが再使用された後、削除された情報は、もはや利用できない。さらに、スナップショット技術は、長期アーカイビングに適したデータを編成する能力を欠いている。

【 0 0 3 4 】

図 1 は、アーカイピング要件に対処するように毎月のボリュームイメージを保持することを含む通常の 4 週間ローテーションスキームを使用して、2 年間にわたって 1 テラバイトのボリュームを管理するのに要求される容量を示す。この例は、50 % の複合年間データ成長レートをモデル化する。バックアップされるべきデータの全体的な量が、50 % 増加する一方で、既存のバックアップ技術に基づいて 1 年の期間にわたってこのデータをバックアップするのに要求されるデータリソースは、元のコンテンツ / データのほぼ 20 倍である。やはり図示されるのが、本明細書で提供される技術（例えば、適応コンテンツファクタリング技術）に基づくディスクベースのアーカイピング方法を使用して達せられることが可能な、元のコンテンツ / データに対する、ほぼ線形のスケーリングである。バックエンドストレージ要件は、より多くの回復ポイント、およびアーカイブされた情報への向上したニアラインアクセスを提供しながら、ほぼ 20 分の 1 に低減される（有効容量比として表された軸を参照）。TCO は、低位ないし中位のディスクストレージ上に展開された場合の、従来のテープベースのバックアップシステムの TCO に近づく。

10

【 0 0 3 5 】

本明細書で開示されるアーカイブ技術は、リファレンス / 格納された情報へのオンライン / 動的アクセスも可能にし得る長期データ保持戦略として特徴付けられる。この技術は、適応コンテンツファクタリングを利用して、ディスクベースのストレージシステムの有効容量を増加させて、デジタルアーカイピングに関する TCO を大幅に低減する。従来のバックアップおよび回復とは異なり、管理されるすべてのデータは、オンラインであり、利用可能である。さらに、アーカイブ内のすべてのデータが、有効期限が切れるまで、アクセス可能なままである。統合された探索 - アーカイブコレクション管理フィーチャが、アーカイブされた情報の全体的な編成および管理を向上させる。

20

【 0 0 3 6 】

コンテンツの長期ストレージをよりうまく最適化するのに、この新たなアーカイピング技術は、所与のデータセットに関して格納される冗長な情報を減らす。冗長な情報が減らされると、データの連続するバージョンを格納するのに要求されるストレージリソースが、より少なくなる。これに関して、適応コンテンツファクタリングは、固有のデータにキーが付けられ、1 回、格納される技術である。従来のコンテンツファクタリング技術または適応差分技術とは異なり、適応コンテンツファクタリングは、発見的方法を使用して、格納されるデータの各量子のサイズを最適化する。適応コンテンツファクタリングは、データ圧縮と関係するが、局所化されたコンテンツに限定されない。データセットの所与のバージョンに関して、新たな情報は、異なる時点において保存された個別の各セグメントからのバージョンを再構成するのに使用されるメタデータと一緒に格納される。このメタデータ - 再構成段階は、I/O 要求に応じる際に通常のファイルシステムが行うことと同様である。

30

【 0 0 3 7 】

図 2 は、適応コンテンツファクタリングの背後の基本的概念を示す。 T_0 において、データセット V_0 （ファイル、ボリューム、またはデータベース）がセグメント化され、個々の要素にキーが付けられ、このデータセットを再構成するのに使用されるセグメントおよび工程を記述するメタデータと一緒に格納される。 T_1 および T_2 において、このデータセットは、データセットが、それぞれ、 V_1 および V_2 になるように更新される。しかし、データセット V_1 および V_2 の新たなバージョン全体を格納するのではなく、これらのデータセットの更新部分を表す変更だけが、バージョン V_1 および V_2 を再構成するのに使用されるメタデータと一緒に格納される。

40

【 0 0 3 8 】

本明細書でさらに説明されるとおり、新規な方法が、初期データセット V_0 と後続のデータセット V_1 との間の変化（例えば、データブロック 3' および 10）を識別して、データチャンクのより大きいセット（例えば、ファイル、ディレクトリなど）が、そのファイル、またはそのディレクトリの前のバージョンと比較されて、後続のバージョンにおけ

50

る変化だけがアーカイブされるようにすることが可能であるようにすることを提供する。これに関して、変化していない元のデータセット V_0 (例えば、ベースラインバージョン) の部分 (例えば、データブロック 1、2、および 4 ~ 9) が、不必要に複製されない。そうではなく、変化のセットを含むファイルまたはディレクトリを再現する際、そのファイル/ディレクトリのベースラインバージョンが利用され、記録された変化 (例えば、3' および 10) 、つまり、デルタが、回復された後続のバージョンに組み込まれる。これに関して、時刻 T_1 においてデータセット V_1 をバックアップする際、データセット V_1 を事実上、バックアップするのに初期データセット V_0 に対する変更だけが保存されればよい。

【0039】

10

データセットの後続のバージョン (例えば、 V_0 および V_1) の間の変化を識別するため、本発明は、新規な圧縮技術を利用する。理解されるとおり、データ圧縮は、データストリームの中のパターンの識別によって機能する。データ圧縮アルゴリズムが、同一の情報を表すより効率的な方法を選択する。基本的に、可能な限り多くの冗長性を取り除くためにデータにアルゴリズムが適用される。圧縮スキームの効率および有効性は、圧縮されたデータに対する圧縮されていないデータのサイズの比である、圧縮比によって測定される。2対1の圧縮比 (標準の圧縮アルゴリズムにおいて比較的一般的である) は、元のデータのサイズの半分である圧縮されたデータを意味する。

【0040】

20

様々な圧縮アルゴリズム/エンジンが、データを圧縮するために異なる方法を利用する。しかし、いくつかの損失のない圧縮アルゴリズムは、辞書ベースの圧縮アルゴリズムである。辞書ベースのアルゴリズムは、圧縮されているテキストの中で以前に見られたストリングの辞書を自動的に構築することが可能であるという洞察を中心に構築されている。これに関して、圧縮中に生成される辞書 (例えば、もたらされる圧縮されたファイル) は、展開器が、圧縮器と同一の仕方で辞書を構築することができ、正しく符号化された場合、テキストにおける同一の箇所で圧縮器辞書が有するのと全く同一のストリングを有することになるので、圧縮されたテキストと一緒に伝送されなくてもよい。そのような構成において、辞書は、初期の圧縮に関連して、生成される。

【0041】

30

本発明者らは、辞書を、圧縮中に生成する代わりに、データセットを圧縮する目的で圧縮器に与えることが可能であることを認識した。特に、本発明者らは、図2に示される第1の時刻 T_0 に関連する元のデータセット V_0 を辞書として利用することで、その後の時刻 T_1 における後続の対応するデータセット V_1 を圧縮可能であることを認識した。これに関して、圧縮器は、元のデータセット V_0 を辞書として利用し、後続のデータセット V_1 の中のデータの大きいストリングは、最初のセットにおけるストリングと完全に重複するものであり得る。例えば、図2に示されるとおり、時刻 T_1 における V_1 の実際の格納は、時刻 T_0 における V_0 のデータブロックに対応するいくつかのブロックを組み込む可能性がある。つまり、第2のデータセット V_1 におけるブロックのいくつかは、データセットの間で変化していない。したがって、変化していないデータブロックを格納する (例えば、そのデータブロックを複製する) のではなく、 V_0 からの対応するデータブロックを参照する識別子が格納される。そのような識別子は、非常に小さい、例えば、10バイトのオーダである。例えば、識別子は、ベースラインの辞書ブロックを参照することが可能である。データのブロック、例えば、3' に変更があった事例において、圧縮器は、3' の変更を、ベースライン V_0 との違い、ならびにブロック3における変更を含むエントリに圧縮するように動作する。さらに、追加のテキストが、後続のバージョン (例えば、ブロック10') に加えられた場合、このテキストが、後続のバージョン T_1 の中に保存される。

40

【0042】

データセットの後続のバージョンの間で非常にわずかな変更が行われた場合、非常に大きい圧縮比が達成される。これらの圧縮比は、50対1、100対1、200対1、また

50

はそれより大きいものとなる。つまり、10ページのテキストドキュメント内で単一の文字が変更された事例において、元のバージョンと後続のバージョンとの間の圧縮は、その1つのわずかな変更を除いて、ほとんど完全である。理解されるとおり、圧縮アルゴリズムのための起源の辞書として元のデータセットを利用することは、後続のデータセットの間の変化を容易に識別して、ベースラインデータセット V_0 からの後続の変化を格納するのに、非常に少しのストレージしか要求しないものになる。したがって、データセットの後続のバージョンを再現することになった際に、データセットの所望されるバージョンに関する辞書識別子が識別される。これに関して、変化が全く存在しない場合、辞書識別子は、ベースラインデータセット V_0 の元のブロックをポイントする。変化が存在する（例えば、3'または6'）場合、識別子は、元のベースラインデータセットおよびデルタデータセットをポイントする。そのような構成は、限られたストレージリソースを利用して、データセットの複数の後続のバージョンを保存することを可能にする。

10

【0043】

この方法は、データセットの後続のバージョンのバックアップの間でわずかな変化しか存在しない場合、特にうまく機能する。しかし、以前にバックアップされたデータセットと比べてデータセットに相当な変化が生じている場合においてさえ、データのサイズ的大幅な縮小が、依然として達せられる。例えば、元のデータセットが、10ページのテキストドキュメントに相当し、後続の対応するドキュメントが、15の新たなページを組み込む（すなわち、合計で25ページ）場合、最初の10ページは、ほぼ完璧な圧縮（例えば、200対1）を実現する一方で、新たなテキストの15ページは、例えば、2対1という、より普通の圧縮オーダで圧縮され得る。しかし、さらなる後続のバックアップ（例えば、第3のバージョン）が、バージョン1および2の新たなテキストをベースラインリファレンスとして利用することができる。代替として、圧縮が、或る所定の圧縮比閾値に達しない場合、変化が、データの元のバージョンを、その後、ベースライン値となる後続のデータバージョンで置き換えることが妥当とされるだけ十分に大きいと判定される。

20

【0044】

図3は、ベースラインデータセットを利用して、データセットの後続のバージョンが圧縮される工程100を示す。図示されるとおり、初期データセットが取得される（102）。このことには、この初期データセットを受け取り、格納すること、および/または、例えば、標準の圧縮技術を利用して、このデータセットを圧縮すること（104）が含まれる。これに関して、この初期データセットを表す圧縮ファイルが生成され、後の時点で、この初期データセットを利用することにより（106）、後続のデータセットにおける違いが識別される。そのような利用には、元のデータ（圧縮された、または圧縮されていない）で辞書ベースの圧縮エンジンを条件付けすること（108）、および元のデータセットを辞書として利用する圧縮エンジンを利用して、後続のデータセットを圧縮すること（110）が含まれる。このことにより、初期データセットと後続のデータセットとの間の変化を示す圧縮ファイルが生成される（112）。さらに、そのような圧縮ファイルは、圧縮辞書（例えば、元のデータセットおよび/または初期の圧縮ファイル）に対するリファレンスを含み得る。したがって、後続のデータセットを示す圧縮ファイルが、例えば、データ復元のために後でアクセス可能な、或る時点のアーカイブとして格納される（114）。後続の対応するデータセットの圧縮のための辞書としてベースラインデータセットを使用することにより、以下のアプリケーションのいくつかは、或る程度、容易になる。以下のアプリケーションは、それらの態様自体が新規である。

30

40

【0045】

前述した圧縮技術を少なくとも或る程度、活用することが可能なアーカイビングサービスを提供するのに、初期データセットが、最初にカタログされなければならない。そのようなカタログは、データセットの様々な構成要素のロケーションのマップを形成し、後の時点におけるデータセットの再構成を可能にする。これに関して、データのセットが、そのデータのベースラインバージョンを生成するように初めてバックアップされる際に、そのデータが、1つまたは複数の知られているハッシュアルゴリズムを使用してハッシュさ

50

れる。これに関して、初期カタログ工程は、主として、既存の工程と同様である。しかし、ハッシングを利用する他のアーカイピング工程とは異なり、このアプリケーションは、データセットの異なる部分に関して複数のハッシュを利用する。さらに、このアプリケーションは、共通の構成要素に関して2つ以上のハッシュを使用することが可能である。

【0046】

例えば、データセットは、各々ハッシュされる異なる3つのデータストリームに分割される。これらのデータストリームは、異なるファイル、異なるフォルダ、および/または異なるデータセットに関するドライブ名/フォルダ名/ファイル名、および/またはサーバIDを含むベースラインリファレンスを含む。つまり、ベースラインリファレンスは、大きいデータセット/ブロックデータの識別と関係する。第2のハッシュが、ベースラインリファレンスのそれぞれに関するメタデータ（例えば、バージョンリファレンス）に対して実行される。この実施形態において、ベースラインリファレンス（例えば、ストレージロケーション）と関係する第1のハッシュは、第2のハッシュを形成するのに利用されるメタデータのサブセットである。これに関して、データセットの各ファイルに関連するメタデータは、いくつかの異なる特性を含み得る。例えば、そのような各バージョンリファレンスに関して12～15のプロパティが存在する。これらのプロパティには、名前、パス、サーバおよびボリューム、最終変更時刻、ファイルリファレンスID、ファイルサイズ、ファイル属性、オブジェクトID、セキュリティID、および最終アーカイブ時刻が含まれる。最後に、各ベースラインリファレンスに関して、生のデータ、またはデータのBlob（バイナリラージオブジェクト）が存在する。一般に、データのそのようなBlobは、ファイルコンテンツおよび/またはセキュリティ情報を含む。データセットをこれら3つの構成要素に分離して、これらの構成要素のそれぞれをハッシュすることにより、各データセットに対して、後続のバージョンに関する変化を識別する複数の検査を実行することが可能である。

【0047】

第1のハッシュ

ベースラインリファレンス - B r e f

1次フィールド

パス/フォルダ/ファイル名

ボリュームコンテキスト

修飾子

最終アーカイブ時刻

第2のハッシュ

バージョンリファレンス - V r e f (1 2 ~ 1 5 のプロパティ)

1次フィールド（変化標識）

パス/フォルダ/ファイル名

リファレンスコンテキスト（1つまたは3つのフィールド）

ファイル最終変更時刻（2つのフィールド）

ファイルリファレンスID

ファイルサイズ（2つのフィールド）

2次フィールド（変化標識）

ファイル属性

ファイルオブジェクトID

ファイルセキュリティID

修飾子

最終アーカイブ時刻

第3のハッシュ（データの大半）

B l o b (個々のデータストリーム)

1次データストリーム

セキュリティデータストリーム

残りのデータストリーム（オブジェクトIDストリームを除く）

別の構成において、合成ハッシュが、2つ以上のハッシュコードから作られる。つまり、V R e f 識別子、B R e f 識別子、およびB l o b 識別子は、2つのハッシュコードから構成される。例えば、高周波数（強い）ハッシュアルゴリズムが、低周波数（弱い）ハッシュアルゴリズムと共に利用される。弱ハッシュコードは、強ハッシュがどれだけ良好であるかを示し、確度の高いハッシュコード衝突（すなわち、合致するハッシュ）に関する1次標識である。代替として、さらに強い（より多くのバイト数の）ハッシュコードも利用可能であるが、さらに強いハッシュコードを生成するのに要求される処理時間が問題となる可能性がある。合成ハッシュコードは、以下のとおり表される。すなわち、

b a = " 0 1 1 5 4 9 4 3 b 7 a 6 e e 0 e 1 b 3 d b 1 d d f 0 9 9 6 e 9 2 4
b 6 0 3 2 1 d "

| 強いハッシュ構成要素 | 弱 |

| 高周波数 | 低 |

これに関して、より大きい単一のハッシュコードと比べて、要求する処理リソースが合算して、より少ない2つのハッシュコードが、スタックされる。これによりもたらされるコードは、データセットの一部分/ファイルに関するさらなる情報を提供する。

【 0 0 4 8 】

一般に、図4に示されるとおり、初期データセットが、そのデータセットに関連するシグネチャを作成する（122）ために、ハッシュされて異なるプロパティになる。このシグネチャは、データセットの個々の部分（例えば、ファイル）に関するいくつかの異なるハッシュコードを含む。さらに、データセットの各部分は、互いに索引付け可能な複数のハッシュ（例えば、ハッシュ1～3）を含む。例えば、データセットの各部分に関するハッシュは、メタデータ（例えば、ベースラインリファレンスおよび/またはバージョンリファレンス）に関連する識別子ハッシュ、およびデータセットのその部分のコンテンツに関連するコンテンツハッシュを含む。バックアップが実行可能となるように後続のデータセットが獲得されると（124）、その後続のデータセットがハッシュされて、シグネチャハッシュコードと比較するためのハッシュコードが生成される。

【 0 0 4 9 】

しかし、すべてのデータをハッシュすることとは異なり、一般に、データB l o b と比べて少ない量のデータを含む、後続のデータセットのメタデータおよびベースラインリファレンス、または識別子構成要素は、初期ベースライン格納以来、変更された、または追加されたファイル（例えば、合致しないハッシュ）を識別するために（128）、最初にハッシュされる（126）。これに関して、変更されているとして識別された合致しないハッシュのコンテンツ（例えば、ファイルのB l o b ）は、次に、ハッシュされて（130）、ベースラインデータセットの格納されたバージョンと比較される（132）。理解されたとおり、ファイルの名前が、第1のバックアップと第2のバックアップとの間で変わる場合がある。しかし、ファイルのテキストに全く変更が行われないことも珍しくない。そのような場合、バージョンリファレンス間のハッシュは、第1のバックアップと第2のバックアップとの間の変更時間における変化を示す。したがって、初期データセットに関連するコンテンツハッシュを識別し、これらのハッシュを、後続のデータセットのコンテンツハッシュと比較することが望ましい可能性がある。理解されたとおり、バックアップ間のドキュメントのテキストに全く変更が行われていない場合、コンテンツハッシュと、関連するデータ（例えば、B l o b ）とは、同一である。これに関して、名前が変更されたファイルに関連するデータを保存する（例えば、以前に保存されたデータを複製する）必要は全くない。したがって、新たなファイル名は、元のファイルのベースラインB l o b に対するリファレンスを共有することが可能である。同様に、同一のコンテンツを有するファイルが、同一のサーバの異なるボリューム上、または異なるサーバ上に存在可能である。例えば、ワークグループ内の多くのシステムは、M i c r o s o f t W o r d （登録商標）に関するアプリケーションファイルの同一のコピー、またはM i c r o s o f t W i n d o w s （登録商標）オペレーティングシステムを構成するファイルを含む

10

20

30

40

50

。したがって、これらのファイルの各ファイルのファイルコンテンツは、同一である。これに関して、別のサーバ上で見られる同一のファイルに関連するデータを再保存する必要は全くない。したがって、ファイルは、別のボリューム、または別のサーバからの元のファイルのベースライン Blob に対するリファレンスを共有する。データセットのベースラインバージョンからのデータセットの後続のバージョンにおいて合致しないコンテンツが存在する場合、後続の Blob が、格納され (1 3 4)、および / または圧縮されて、格納される (1 3 4)。

【 0 0 5 0 】

重要なこととして、図 4 の工程 1 2 0 は、分散可能である。これに関して、格納されたデータに関連するハッシュコードは、データの起源ロケーションに供給される。つまり、初期データセットは、オフサイトのロケーションにおいて格納される。ハッシュコードをデータ起源ロケーションに供給することにより、何が新たなコンテンツであるかという判定が、データの起源ロケーションで行われる。したがって、新たなデータだけが、ストレージロケーションに転送されればよい。理解されるとおり、これにより、バックアップデータをオフサイトのストレージロケーションに転送するための帯域幅要件が低減される。

【 0 0 5 1 】

図 5 は、本発明のいくつかの態様に従ってデータをアーカイブするための工程の一実施形態を示す。最初、元のデータセットが、受け取られる (1)。このデータセットは、限定なしに、サーバ、データベース、またはファイルシステムから受け取られるデータを含む。このデータは、通常、このデータをバックアップする、またはアーカイブする目的で受け取られる。受け取られたデータ内の各アイテム / オブジェクト (例えば、ファイル、フォルダ、またはデータの任意のブロック) が処理され (2)、バージョンリファレンス (「 V r e f 」) が計算される (3)。前述したとおり、V r e f は、オブジェクトのメタデータ 3 a と関係する多数のフィールドを含む。これらのフィールドには、1 次フィールドおよび 2 次フィールドが含まれる。これらのフィールドは、データセットの第 1 のインスタンスのアーカイビング (すなわち、バックアップ) と、後続のインスタンスのアーカイビング (すなわち、バックアップ) との間の変化を識別するのに利用される。

【 0 0 5 2 】

このことは、最初、オブジェクトデータが、アーカイブシステム内に存在するかどうかを判定することを可能にする。V r e f が計算されると (3)、V r e f はオブジェクトストア 4 a に割り当てられる (4)。割り当てが行われると、共通コンテンツオブジェクトストアとの比較 (5) が実行されて、その V r e f に関連するオブジェクトが既に存在する (すなわち、前のアーカイブ動作から) かどうか判定される (6)。この判定は、リファレンスルックアサイドテーブル 7 を利用して実行される。リファレンスルックアサイドテーブル 7 は、V r e f ハッシュコードおよび B r e f ハッシュコードを含むテーブルである。いずれにしても、新たに受信されたデータからのオブジェクトの V r e f が、以前にアーカイブされたオブジェクトの V r e f と均等である場合、そのオブジェクトが、既に存在する可能性があるという判定が行われる。マッチが全く見出されない場合、処理は、本明細書で説明されるとおり進む。リファレンスルックアサイドテーブル 7 内でマッチが全く見出されない場合、オブジェクトの存在は、オブジェクトストアを探索することによってさらに判定される。マッチが見つかった場合、V r e f は、リファレンスルックアサイドテーブルにロードされる。

【 0 0 5 3 】

マッチが全く識別されない (例えば、オブジェクトが、新たなデータ、またはより早期のバックアップ以来、変更されているデータを表す) 場合、データをアーカイブすることに関するストレージポリシーが選択される (8)。図示される実施形態では、汎用ポリシーが選択される。理解されるとおり、異なるデータタイプに関して異なるポリシーが選択される。例えば、知られていないデータに関して汎用ポリシーが選択される。これに対して、データの 1 つまたは複数の構成要素が知られているデータセットに関して、その特定のデータセットのニーズによりよく合致するポリシーを選択することが好ましい可能性が

10

20

30

40

50

ある。ポリシーが選択されると(9)、工程は、継続し、ベースラインリファレンス(「B r e f」)9が、データソースの以前に合致しなかった各オブジェクト10aに関して計算される。V r e fデータのサブセットを利用して、ベースラインデータまたはB r e fデータが計算される。具体的には、前段で概説されるメタデータを利用して、ベースラインリファレンスオブジェクトに関するハッシュが計算される。

【0054】

オブジェクトに関してB r e f 9が計算されると、B r e fは、ストアに割り当てられる(11)。この割り当て(11)は、対応するV r e fに関して行われた同一の割り当て(11)に基づく。通常、計算されるB r e fは、最新のB r e fである。しかしながら、メタデータは、第1の時点と第2の時点(例えば、第1のアーカイビング工程と第2のアーカイビング工程)に関して同一である一方で、オブジェクトデータは、変化する可能性がある。そのような場合、現在のB r e fが、最新のB r e fであるかどうかの判定(12)が、最終アーカイブ時刻修飾子を使用してオブジェクトストアの中の他のB r e fデータと比較することによって行われる。これにより、異なるアーカイビング工程の対応するオブジェクトの間で変更があったかどうかを確かめる冗長性検査が可能になる。

【0055】

次に、現在のB r e fが、オブジェクトストア内に既に存在するかどうかの判定(13)が行われる。この場合も、リファレンスルックアサイドテーブル7が、この判定のために利用される。これに関して、現在のB r e fデータのハッシュが、リファレンスルックアサイドテーブル7内の既存のハッシュと比較される。

【0056】

オブジェクトが既に存在する場合、オブジェクトは、0と1の一連のバイナリデータを含むB l o b(すなわち、バイナリラージオブジェクト)に解決される(14)。B r e fを利用して、V r e fが調べられ、次に、V r e fを利用して、データの関連するB l o bが調べられる。一部の事例において、データのB l o bは、ルートベースラインB l o bである、さらなるB l o bを参照することが可能である。一部の事例において、共通のデータのB l o bが、多くのオブジェクトに関して存在する。例えば、多数の別々のコンピュータのオペレーティングシステムは、同一のファイルの多くを有して、実質的に同一であり得る。したがって、そのような別々のコンピュータのバックアップが実行される際、それらの共通のファイルに関する、もたらされるB l o bは、同一であることが可能である。したがって、異なるオブジェクトのV r e fおよびB r e fは、同一のB l o bを参照することが可能である。

【0057】

ベースラインB l o bが探し出されると、このB l o bは、圧縮アルゴリズムのための辞書としてロードされる(15)。B l o bは、辞書にロードされると(15)、個々のデータチャンクに分割される。例えば、ベースラインB l o bは、30KBのデータチャンクに分割されるか、または操作者の選択に基づく他の任意のサイズのデータチャンクに分割される。これらの個々のチャンクは、圧縮アルゴリズムに事前に条件付けをするように圧縮器にロードされる。

【0058】

事前に条件付けされることが可能である限り、複数の既知の圧縮技術のいずれも利用することができる。この事例では、圧縮アルゴリズムは、B l o bデータのいくつかの部分または全体で事前に条件付けられる。この時点まで、処理されたすべてのデータは、メタデータであった。しかし、この時点で、受け取られたオブジェクトは、ベースラインB l o bで事前に条件付けされた圧縮アルゴリズムを使用して圧縮される(16)につれ、ハッシュされる。オブジェクトが、B r e fを有する場合、新たなオブジェクトとベースラインオブジェクトとの間の変化が、デルタB l o b 17と呼ばれるアイテムのもたらされる圧縮によって特定される。オブジェクトが、B r e fを有する場合、対応するデルタB l o bは、しばしば、ベースラインB l o bのサイズの数分の1に過ぎず、100:1という圧縮比が、珍しくない。

10

20

30

40

50

【 0 0 5 9 】

変化を識別する工程は、デルタ B l o b 工程と呼ばれる。デルタ B l o b 工程の出力は、ベースラインデータセットと新たなデータセットとの間の違いを表すバイナリデータセットであり、あるいはベースラインが存在しない場合、この出力が、将来の参照の目的でベースラインになることが可能である。いずれの場合でも、デルタ B l o b またはベースライン B l o b は、受け取られたデータのハッシュによって表され、現在、存在していない場合、オブジェクトストア 5 にコピー / 格納される (1 8)。なお、任意で、B r e f、および対応する V r e f の、最終アーカイブ時刻修飾子によって特定される、より古いバージョン、およびベースライン B l o b またはデルタ B l o b データが、オブジェクトストア内のスペースを空けるようにリサイクルされる。

10

【 0 0 6 0 】

理解されるとおり、前述したアーカイピングシステムは、完全に自己完結型であり、外部ストレージ要件を全く有さない。このため、オブジェクトストア 5 全体が、オフサイトの格納の目的で、単一の取り外し可能な媒体ユニット上でホストされる。すべての索引、すべてのリファレンス、およびすべてのコンテンツが、個々のアイテムとして単一のファイル構造内に保持されるため、さらに格納されたアイテムのいずれも、更新される必要がないので、オブジェクトストアを代替のロケーション、または遠隔ロケーションに複製するファシリティは、全く使用されない。この独特なストレージレイアウトは、任意の所与のディスク破損の影響を隔離する障害許容型の構造を提供する。さらに、アイテムの参照の完全性が、検証を可能にし、障害を隔離する。後続のアーカイピングジョブは、検出された破損を自己回復するのに使用可能である。取り外し可能な媒体に関して、ベースオブジェクトストアレイアウトおよびツリー深度が定義されると、同一の構造が、独立した時点をもたぐ媒体の連続的なローテーションを提供するような仕方で、任意の数の取り外し可能な媒体上に複製される。この工程は、テープ媒体ローテーションと同様であるが、共通のコンテンツが考慮に入れられるので、はるかに効率的である。この構造は、2 0 : 1 以上で均等の媒体ユニットに関する要件を緩和する。

20

【 0 0 6 1 】

図 7 および図 8 は、オブジェクトストアからのデータの再構成を示す。前述したとおり、この工程は、データのリアルタイムの、つまり、動的な、または「オンザフライ」の再構成を可能にする。そのような動的再構成を提供するのに、アーカイブされたデータは、データを再構成しようと試みるユーザがアクセスすることができる仮想ファイルシステムの中で表される。データを再構成するのに、所望されるオブジェクトまたはファイルのアドレスが、知らなければならない。そのアドレスがどのように知られるようになるかについて、後段で説明する。

30

【 0 0 6 2 】

最初に、システム内のすべてのデータが、オブジェクトストア内に格納され、図 6 に示されるとおり、仮想ファイルシステムの中で表される。図 6 は、この仮想ファイルシステムを使用して、この実施形態では、ウェブクライアントネットワークを使用して、アーカイブされたデータにアクセスすることを示す。しかし、アーカイブされたデータへのアクセスは、アーカイピングが所望されるシステムに接続されたスタンドアロンのユニットを介することも可能である。V F S (仮想ファイルシステム) のいくつかの態様は、両方のシステムに適用可能である。ウェブクライアントネットワークの場合、アーカイブされたデータへのアクセスは、W i n d o w s W e b C l i e n t サービスリダイレクタを使用して W e b D A V を介して達せられる。このリダイレクタは、U N C (ユニバーサルネームコンベンション) パスを使用する、アーカイブされたデータへのアクセスを可能にする。この事例では、アーカイブされたデータを閲覧することへの入口点は、U N C パス、\\ v o y a g e r \ O b j e c t S t o r e を介してである。さらに、W e b C l i e n t リダイレクタは、ドライブ名を U N C パスにマップすることをサポートする。例えば、ドライブ名 L : が、\\ v o y a g e r \ O b j e c t S t o r e に割り当てられる。ドライブ名マッピングは、階層の任意のレベルに割り当てられることが可能であることに

40

50

留意されたい。例えば、X : が、`\\voyager\ObjectStore\Important Documents`に直接にマッピングされることも可能である。

【0063】

図6は、VFS階層におけるオブジェクトストアエントリを示す。この例では、オブジェクトストアインスタンスは、`ObjectStore`と呼ばれる。オブジェクトストアは、複数のリソースからプールされた、アーカイブされたデータ（例えば、複数のソースからの共通のコンテンツ）と、或る特定の/個別のデータセットまたはカタログをより厳密に定義するアーカイブをともに含む。つまり、個々のデータセットには、それらのデータセット独自のアーカイブ（例えば、重要なドキュメント）内で索引が付けられる。これに関して、知られているデータセットに関連するデータを再構成しようと試みる際、オブジェクトストアの索引全体を探索するのではなく、そのデータセットのアーカイブが探索される。これにより、所望される情報に関して、大域的な索引を探索するのではなく、個別のアーカイブを探索することが可能になる。これにより、索引に関する格納要件、探索に関する計算要件、ならびにコアメモリ要件が低減される。

【0064】

データセットが、システムの中に移動されるたびに、そのデータセットの現在の状態、または時点カタログが作成され、システム内に記録される。理解されるとおり、このことに必然的に伴うのは、データセットの生のデータを格納するのではなく、データに関連する情報（例えば、メタデータ）を格納することだけである（例えば、そのデータが、システム内に既に存在するものと想定して）。いずれにしても、データセットがシステム内に格納された時点が保存される。これにより、時点カタログ（例えば、図6のアーカイブされたUTCエントリ）の生成がもたらされる。或る特定の時点に関するデータセットを表す各カタログは、或る特定のデータセットに関するすべてのメタデータの正確な表現を含む。しかし、或る特定の時点に関するデータセットに関連するすべての生のデータが、コピーされなければならない訳ではない。前述したとおり、前の時点と、現時点との間で変化したファイルだけが、システムの中にコピーされる。変化していないファイルに関して、時点カタログに関するメタデータは、前のカタログのデータに対する適切なりファレンスと一緒に格納される。

【0065】

すべての情報、時点は、格納されなくてもよいので、多数のカタログが、多数の時点に関して生成され、保存される。つまり、限られた数の完全なバックアップデータセット（例えば、より新しいバックアップデータセットで定期的に置き換えられる）を提供し、データセットのそれぞれが、共通のデータの冗長なコピーを含むシステムではなく、比較的小さいカタログの使用が、データが再構成される時点の量を増加させることを可能にする。つまり、カタログは、ユーザに利用可能であるバックアップデータセットの細分性を大幅に高めることを可能にする。

【0066】

つまり、各時点に関するデータを保存するのではなく、カタログは、所与の時点に関するデータを再現するためのコードを保存する。具体的には、或る時点に関するカタログは、仮想ファイルシステムによって、所与の時点に関するデータセットのレプリカを再現するのに使用される、各レコード（ファイル）に関する1つまたは複数のハッシュコードを含む。以下は、カタログの中の単一のレコードの例示的なサンプルであり、`ca`、`sa`、`oa`、`ba`、および`aa`に関するエントリは、異なるデータストリームを表すハッシュコードである。例えば、`<ca>`は、レコードに関する`Vref`であり、或る特定のバージョンを識別するのに使用されるすべてのメタデータを組み込む。`<sa>`は、セキュリティストリームに対する`Blob`アドレス（ハッシュ）である。`<oa>`は、オブジェクト識別ストリームに対する`Blob`アドレスである。`<ba>`は、1次`Blob`アドレスである。`<aa>`は、代替の（または2次）`Blob`アドレスである。

【0067】

`<ref ty = "2" nm = "build.properties.sample`

```
e" in = "\ Little Test" on = " 3 1 6 2 6 4 5 3 0 3" ct =
" 1 2 8 1 0 5 3 9 1 2 8 7 9 6 8 7 5 0" at = " 1 2 8 1 8 6 3 6 4 5 7 1 7 1 8
7 5 0" mt = " 1 2 7 4 8 3 0 6 0 7 9 0 0 0 0 0 0 0" sz = " 1 6 4 4" f
l = " 1 2 8" id = " 5 6 2 9 4 9 9 5 3 4 4 8 8 2 9 4" ca = " 1 d 1 6 4 9
c b 2 b 3 9 8 1 6 d 6 9 9 6 4 c 1 c 9 5 a 4 a 6 a d 7 9 a 4 1 6 8 7" sa = "
3 a f 4 e c 9 5 8 1 8 b d c 0 6 a 6 f 1 0 5 1 2 3 c 2 7 3 7 b e 6 e a 2 8 8 d f
" oa = "" ba = " 0 1 1 5 4 9 4 3 b 7 a 6 e e 0 e 1 b 3 d b 1 d d f 0 9 9
6 e 9 2 4 b 6 0 3 2 1 d" aa = "" op = " 1" / >
```

上記のとおり、カタログのこの部分は、所与のファイルのメタデータおよびコンテンツを探し出して、再現することを可能にするレコードを形成する。

10

【0068】

図6を再び参照すると、カタログは、元のデータセットを表し、ボリューム、フォルダ、およびファイルを含む階層形態になっている。この階層におけるエントリのそれぞれは、エントリのプロパティを記述するメタデータを含む。さらに、フォルダレコードおよびファイルレコードは、Vrefアドレスを含み、タイムスタンプをアーカイブする。この階層は、バックアップされるデータセットの階層を模倣する。例えば、階層は、個々のユーザを含む。選択された特定のユーザ、例えば、マイク(Mike)に関して、そのユーザのコンピュータ、サーバなどのコンテンツが、そのユーザのコンピュータ、サーバなど同一の仕方で格納される。

【0069】

20

この階層は、前述したとおり、格納されたデータの任意のセットに遠隔でアクセスするのに使用可能であり、本明細書で説明されるアーカイビングシステム以外にも応用されるVFS(仮想ファイルシステム)の一部として提示される。ユーザは、VFS階層にアクセスして、オブジェクトストアの適切なアーカイブからデータを再構成することができる。これに関して、ユーザは、ユーザのスクリーン上で、図6に示される表現を見ることができる。ユーザは、或る特定のアーカイブまでVFSの中を移動し、所望される時点カタログを選択して、そのフォルダを展開することができる。その時点で、その時点カタログより下の階層が提供されて、ユーザが、その時点カタログ内の所望されるドキュメントまで移動することができるようにする。つまり、ユーザは、ユーザが再構成したいと望むドキュメントを探し出すまで、ユーザの標準のストレージインタフェースを模倣するVFSの中を移動することができる。特に、特定の時点が、ユーザによって全く選択されなくてもよい。例えば、探索エンジンが、各時点アーカイブを、そのアーカイブの中の所望されるデータに関して探索する能力を有してもよい。重要なこととして、VFSにアクセスするのに、特化されたクライアントアプリケーションは、全く要求されない。これに関して、許可されたユーザが、ユーザ自身のコンピュータ上で所望されるファイルにアクセスする場合と同様に、アーカイブされたデータセットにアクセスするためにユーザの標準のオペレーティングシステムを利用することができる。

30

【0070】

前述したとおり、図6は、アーカイブされたデータの表現である。この場合、データは、ファイルシステムの完全に関連可能なバージョンをユーザが利用できるように複数回のアーカイビング実行によって保っているWindowsファイルシステムからである。注目すべきこととして、VFSにおける遷移は、アーカイビング時点階層が、止まり、ソースからのデータの表現が始まるVFS階層において生じる。この例では、遷移またはピボットは、「Archived UTC - 2006.04.03 - 23.57.01.125」と名付けられている。階層においてこのポイントより下のフォルダは、アーカイビングタスクのためのファイル/フォルダ基準として指定されるルートファイルシステムを表す。「voyager」上のユーザ(U\$)は、Userというラベル、Uというドライブ名を有し、voyagerと名付けられたシステムからのファイルボリュームである。しかし、他のファイルシステム(例えば、非Windowsシステム)が表現されることも可能である。或る特定の時点に関してアーカイブ内で或るファイルレベルに到達す

40

50

ると、ユーザは、或る特定のファイルを選択することができる。すると、この選択により、バージョンリファレンスアドレス (V r e f) が与えられ、アーカイブ時刻を利用して、その特定のファイルの再構成が始められる。

【 0 0 7 1 】

B l o b アドレスを V r e f と共に格納することの重要性は、そうすることにより、そのアドレスが、B l o b であるか、またはデルタ B l o b であるかにかかわらず、V r e f が、オブジェクトストア 5 内の実際のコンテンツを参照することが可能になることである。アドレスがデルタ B l o b である場合、そのデルタ B l o b は、ベースライン B l o b をさらに参照することが可能である。したがって、情報は、所望されるデータを再構成しようとして獲得される。この時点で、ベースライン B l o b、および存在する場合、デルタ B l o b が識別されており、つまり、データが、この時点で再構成される。

10

【 0 0 7 2 】

ユーザは、或る特定の期間からの (例えば、或る特定の V r e f に関する) データを再構成するために、アーカイブ時刻を指定することができる (32)。理解されるとおり、実際のアーカイブ時刻は、ユーザによって与えられる、所望される期間と同一ではない可能性がある。いずれにしても、システムは、最も妥当な再構成時刻 (例えば、所望される時刻より前、または所望される時刻のすぐ後に実行されたバックアップからのデータ) を特定する (34)。初期 V r e f がデルタ B l o b を有するかどうかについての初期判定 (36) が行われる。デルタ B l o b が、この V r e f に関して存在する場合、そのデルタ B l o b がオブジェクトストアから獲得される (38)。また、対応するベースライン B l o b がオブジェクトストアから獲得される (40)。デルタ B l o b が存在しない場合、ベースライン B l o b だけが獲得される。V r e f が、圧縮されていないオブジェクト (例えば、個別のファイル) を参照する場合、その圧縮されていないオブジェクトが、後続の読み取りのために獲得される (44)。

20

【 0 0 7 3 】

B l o b (または圧縮されていないオブジェクト) は、獲得されると、圧縮されていないデータの出力を生成するように再構成される。図 8 を参照すると、この工程において、V r e f (すなわち、デルタ B l o b またはベースライン B l o b を参照する) が、獲得された B l o b からの個々のチャンクまたはバッファの中で再構成される。そのようなバッファの長さは、固定長であっても、可変長であってもよく、このことは、ユーザによって指定可能である。V r e f が、前述したとおり獲得されたデルタ B l o b を参照する事例において、そのデルタ B l o b が、次に、展開されて、V r e f データが再構成される。オブジェクト (例えば、デルタ B l o b) が読み取られ (52)、バッファ 54 が一杯になるまで展開される。このことが、オブジェクト全体が展開されるまで、繰り返し反復される。デルタ B l o b の各展開に関して、そのデルタ B l o b の一部分が、バッファを埋めるのにベースラインの参照される部分を要求する可能性がある。これに関して、デルタ B l o b のその特定の部分を展開する展開情報を提供する新たな辞書 (例えば、ベースライン B l o b の部分) が要求されるかどうかの判定 (56) が行われる。つまり、必要な場合、システムは、デルタ B l o b の現在の部分を展開する (62) ように展開アルゴリズムに事前に条件付けをする (60) 開かれたベースライン B l o b の一部分を獲得する (58)。

30

40

【 0 0 7 4 】

V r e f アドレスおよびアーカイブ時刻という 2 つのデータを所与として、これら 2 つのデータが、取り込まれ、正確な V r e f およびアーカイブ時刻マッチ、または 1 つ前の V r e f アーカイブ時刻に関してオブジェクトストアを探索するのに利用される。図 7 を参照すると、例えば、再構成されるべき所望されるファイルが、より早期のバックアップ以来、変更されていない場合、V r e f アドレスは、そのファイルに関するデータが格納された実際の時刻を表す、より早期の V r e f 時刻を参照する。このレベルまで解決されると、V r e f の属性が読み取られて、その V r e f が、デルタ V r e f であるか、またはベースライン V r e f であるかが判定されるべきである。

50

【 0 0 7 5 】

デルタ B l o b が存在せず、ベースライン B l o b だけが存在する場合 (6 4)、工程は、オブジェクトストアからの V r e f に基づいて、ベースライン B l o b を獲得し (6 6)、バッファを埋めるようにベースライン B l o b を展開する (6 8)。バッファが、展開されたデータで埋められると、データのこのバッファが、要求側ユーザに戻される。一つの構成では、オブジェクトは、圧縮されていないデータであることが可能である。この事例において、データセットが、圧縮されていない形態で存在し得る。そのような事例において、バッファは、展開工程を要求することなしに埋められる (7 0)。バッファを埋めること、および戻すことは、例えば、ファイルの終わりに達するまで、繰り返される。データセットから複数のファイル (例えば、複数の V r e f) を取得可能であることが理解される。さらに、データセット全体を取得することも可能である。

10

【 0 0 7 6 】

適応コンテンツファクタリング技術に関する一つの応用例は、従来のディスクベースのバックアップから情報を取り入れることである。ほとんどの場合において、相当な量の情報が 2 つの完全なバックアップデータセットの間で共通である。共通のデータを除外することにより、所与のストレージデバイスの有効容量が、機能の損失なしに、アーカイピングシステムの向上した性能を伴って、大幅に増加する。このことにより、長期のディスクベースのアーカイピングが、経済的に実現可能となる。そのようなアーカイピングは、ローカルで実行することも、ネットワークを介して実行することも可能である。例えば、図 9 を参照すると、当業者には理解されるとおり、ネットワーク帯域幅が低下するにつれ、所与のデータセットの共通のコンテンツを識別して、遠隔サーバから中央アーカイブに変更だけを送信することが有利である。これに関して、コンテンツが既に格納されているかどうかを判定するのに使用される索引が、ネットワーク 8 0 を介して効率的に格納され、分散されることから、前述した新規なアプローチは特にうまく機能する。所与のデータセット、または同様のデータセットに固有のコンテンツ索引を作成して、保持することにより、索引の対応するサイズが、局所化されたコンテンツに縮小される。例えば、索引におけるエントリが、1 アイテム当たり 8 バイトである場合、データセットは、5 0 0 0 0 のアイテムを含む。索引の対応するサイズは、4 0 0 0 0 0 バイトに過ぎない。このことは、アーカイブされた数百万の離散的なアイテムに対する一体構造の索引を使用する他のシステムとは対照的である。このため、より小さい分散型の索引が、ローカルで、またはネ

20

30

【 0 0 7 7 】

説明される技術は、共通のコンテンツに対する索引のローカルでキャッシュ可能なネットワークを提供する。つまり、複数のサーバ / コンピュータ 8 2 が、共通のストレージファシリティ 8 4 を共有することが可能である。このコンテンツは、共通のコンテンツが共有されてストレージ要件を低減するように、アーカイピング機器 8 8 によって処理される。必要なカタログは、共通ストレージファシリティ 8 4 において、または 2 次ストレージ 8 6 において格納される。個々のサーバ / コンピュータをバックアップ可能にするのに、この技術は、データセットごとに分散型の索引を使用する。つまり、識別子とコンテンツハッシュの固有のセットが、特定のサーバ / コンピュータに提供される。一般に、索引内の情報は、データセット内の所与のアイテムに対するハッシュ (例えば、V r e f) に対応する。しかし、理解されるとおり、データセットが、異なるホストシステムからのアイテムに対応する場合でさえ、異なるデータセット間のデータセット内のアイテムに共通する可能性があるファイルセキュリティ情報またはオブジェクトセキュリティ情報などの、非常によく参照されるコンテンツ索引または B l o b 索引を格納して、これらの B l o b が既に格納されていることを迅速に識別することも望ましい。これに関して、この技術は、元のデータセットコンテンツを、一連の B l o b アドレスと、その後続く 0 で埋めら

40

50

れたバイトアレイで置き換えることによって、B l o bに対する代替の索引を使用する。B l o bアドレスと、これに加えられた、0で埋められたアレイは、元のコンテンツの各セグメントの論理サイズと完全に合致するようになっている。当業者には理解されたとおり、0で埋められたアレイは、いくつかのデータ圧縮アルゴリズムによって極めて容易に圧縮可能である。本発明は、データセットをまず、離散的なオブジェクトデータストリームに分割し、各オブジェクトデータストリームを、後段で説明されるM3工程、または類似した工程を使用して以前にアーカイブされた、または同時にアーカイブされるコンテンツ（またはB l o b）に対するストリームアドレスで置き換え、次に、再マップされたデータストリームの残りの部分を0で埋めることによって、任意の知られているファイルフォーマットで機能する。最後に、この再マップされたストリームが、圧縮され、これにより、0で埋められたアレイにおける冗長性が基本的に取り除かれる。もたらされるファイルが、データストリームコンテンツの再マッピングを除いて、オリジナルと区別ができないことが望ましい。これに関して、ビットフラグが、元のファイルのメタデータ内で使用されて、ストリームデータが置き換えられていることを示して、データストリームが再マップされていると、元のデータを作成した元のプログラムを判定することができる。本発明は、ヘッダチェックサムにこだわらずに、ストリームヘッダの中で予約フラグを設定する。起元プログラムは、データセットをカタログすることができるが、データストリームが読み取られると、チェックサムが検査される。予約フラグが設定されているため、チェックサム試験は、不合格となり、アプリケーションが、再マップされたストリームを間違っ

て読み取ることが防止される。図10は、この工程を示す。ストリームアドレスの特定は、データセット内部に格納されたメタデータを使用する完全な工程を使用し、ストリームB l o bアドレスを特定する逆ルックアップを含むこと、またはストリームデータに対してハッシュアルゴリズムを使用して、一意のストリームB l o bアドレスを計算することが可能である。アンマップ工程は、動作の順序を単に逆にして、各B l o bアドレスと0で埋められたアレイが、元のコンテンツで置き換えられ、予約フラグが設定解除されるようにする。このアンマップ再構成工程の結果は、元のデータセットと同一のコピーである。

10

20

30

40

【0078】

以上の説明は、例示および説明の目的で提示されてきた。さらに、この説明は、本発明を、本明細書で開示される形態に限定することを意図していない。いくつかの例示的な態様および実施形態が、以上に説明されてきたが、これらの態様および実施形態の、いくつかの変種、変形、置換、追加、および部分的組み合わせが、当業者には認識される。したがって、添付の特許請求の範囲は、本発明の思想および範囲に含まれる、すべてのそのような変種、変形、置換、追加、および部分的組み合わせを含むように解釈されることが意図される。

【図面の簡単な説明】

【0079】

【図1】データセットの長期格納要件を示す図。

【図2】バージョン間のデータセットに対する変更を示す図。

【図3】関連するデータセット間の違いを識別するための工程を示す図。

【図4】データセットに関するシグネチャを生成するための工程を示す図。

【図5】データを格納するための工程を示す図。

【図6】複数のアーカイブカタログのアクセス可能なカタログを示す図。

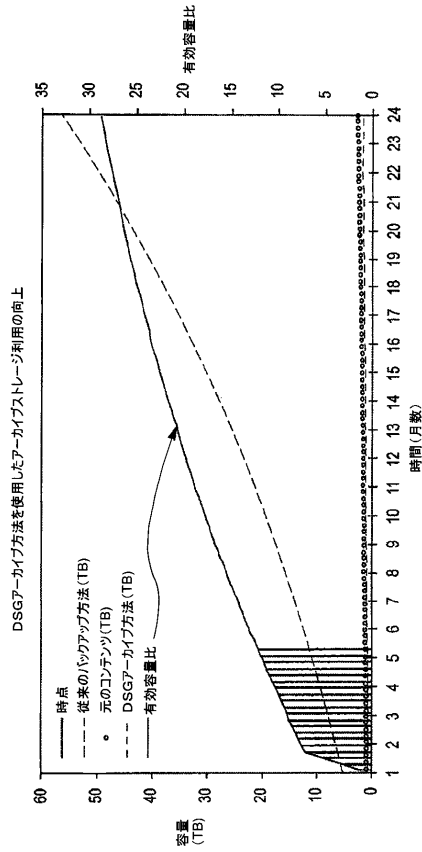
【図7】データを取得するための工程を示す図。

【図8】データを再構成するための工程を示す図。

【図9】ネットワークを介するデータの格納を示す図。

【図10】コンテンツデータと一緒にメタデータを格納することの一実施形態を示す図。

【図 1】



【図 2】

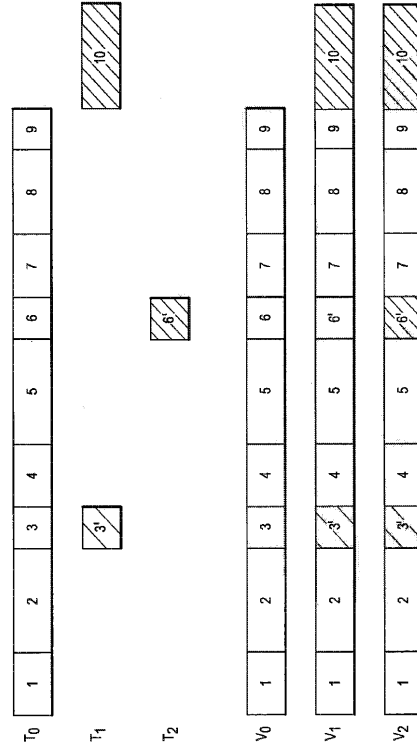
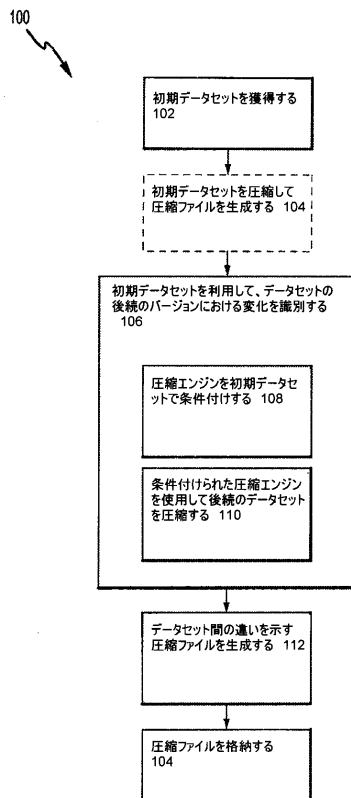
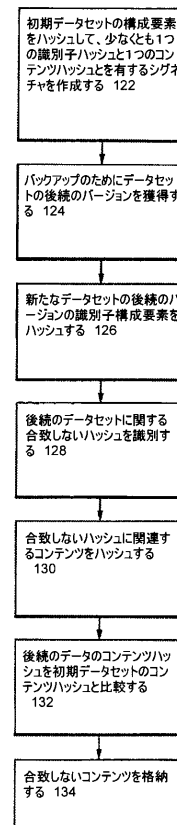


FIG.2

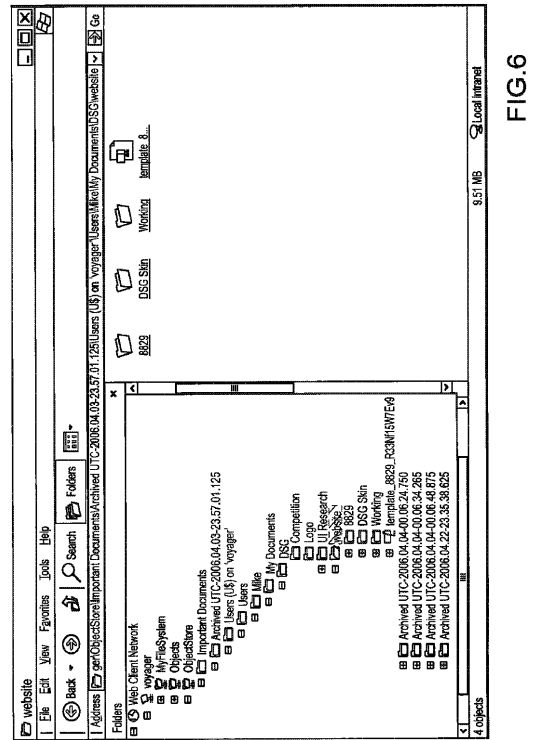
【図 3】



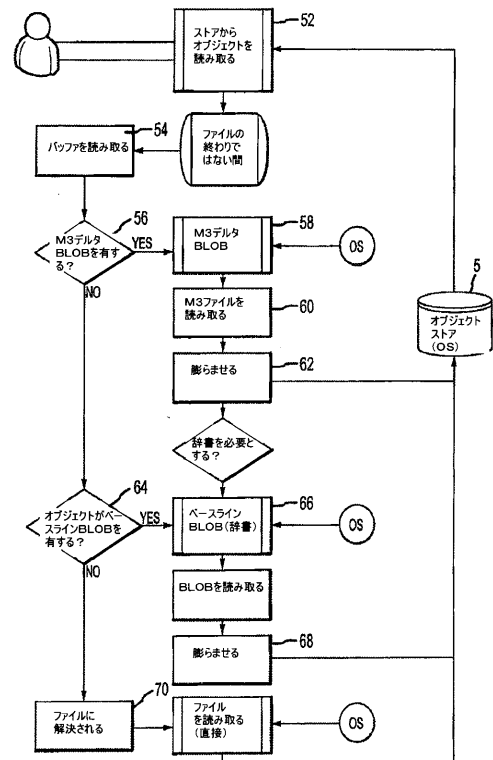
【図 4】



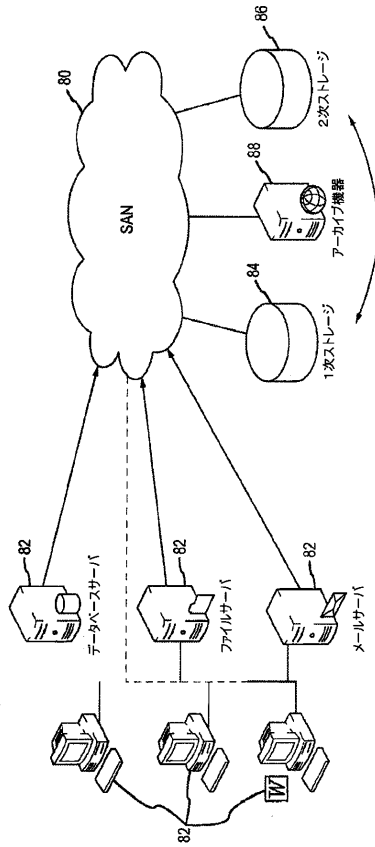
【 図 6 】



【 図 8 】



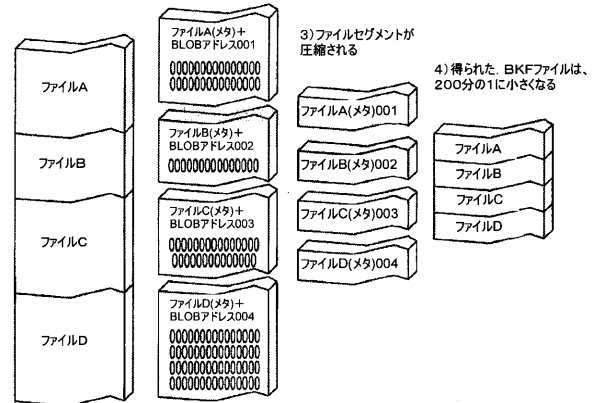
【図 9】



【図 10】

2) 再マップする: ファイルストリームデータが、BLOBアドレス、および0で埋められたアレイで置き換えられる

1) ファイルセグメントを示す元の、BKFファイル



【 国際調査報告 】

60900090015



INTERNATIONAL SEARCH REPORT

International application No.

PCT/US07/66263

A. CLASSIFICATION OF SUBJECT MATTER IPC: G06F 7/00(2006.01), 17/00(2006.01) USPC: 707/100 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) U.S.: 707/100 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) US PG Pub, US Pat, USOCR, FPRS, EPO, JPO, DERWENT, IBM_TDB		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,574,906 A (MORRIS) 12 November 1996 (12.11.1996), entire document, especially col.'s 9-12.	1-20
Y		25
Y	US 6,374,250 B2 (AJTAI et al) 16 April 2002 (16.04.2002), col.'s 5 and 9.	5-9
X	US 6,779,003 B1 (MIDGLEY et al) 17 August 2004 (17.08.2004), entire document, especially col.'s 3-5, 14, and 15.	21-55
Y		8,9,14,17
Y	USPA 2005/0216788 A1 (MANI-MEITAV et al) 29 September 2005 (29.09.2005), paragraphs 246-259.	51, 52
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents:		
"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention	
"E" earlier application or patent published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone	
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reasons (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art	
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family	
"P" document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search 02 September 2008 (02.09.2008)	Date of mailing of the international search report 29 SEP 2008	
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450 Alexandria, Virginia 22313-1450 Facsimile No. (571) 273-3201	Authorized officer Mohammad Ali <i>J. Roberts for</i> Telephone No. 571-272-4105	

Form PCT/ISA/210 (second sheet) (April 2007)

17. 2. 2009

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW

(72)発明者 ドッド、ブライアン

アメリカ合衆国 8 0 5 0 4 コロラド州 ロングモント メドウ ベイル ロード 2 1 7 8

(72)発明者 ムーア、マイケル

アメリカ合衆国 8 0 0 2 6 コロラド州 ラファイエット ロッジウッド レーン 3 8 8

Fターム(参考) 5B082 GA01