

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号

特許第7065108号

(P7065108)

(45)発行日 令和4年5月11日(2022.5.11)

(24)登録日 令和4年4月27日(2022.4.27)

(51)国際特許分類

F I

G 0 6 F 16/185(2019.01)

G 0 6 F 16/185

G 0 6 F 3/06 (2006.01)

G 0 6 F 3/06 3 0 1 N

G 0 6 F 3/06 3 0 3 Z

請求項の数 8 (全24頁)

(21)出願番号	特願2019-543300(P2019-543300)	(73)特許権者	502303739
(86)(22)出願日	平成29年11月29日(2017.11.29)		オラクル・インターナショナル・コーポ
(65)公表番号	特表2020-509481(P2020-509481		レイション
	A)		アメリカ合衆国カリフォルニア州940
(43)公表日	令和2年3月26日(2020.3.26)		65レッドウッド・シティー、オラクル
(86)国際出願番号	PCT/US2017/063637		・パークウェイ500
(87)国際公開番号	WO2018/147922	(74)代理人	110001195
(87)国際公開日	平成30年8月16日(2018.8.16)		特許業務法人深見特許事務所
審査請求日	令和2年11月27日(2020.11.27)	(72)発明者	ディッシンガー、ベンジャミン
(31)優先権主張番号	15/431,615		アメリカ合衆国、94065 カリフォ
(32)優先日	平成29年2月13日(2017.2.13)		ルニア州、レッドウッド・シティー、オ
(33)優先権主張国・地域又は機関	米国(US)	(72)発明者	ラクル・パークウェイ、500
			ブレイジング、ブラッドフォード
			アメリカ合衆国、94065 カリフォ
			ルニア州、レッドウッド・シティー、オ
			最終頁に続く

(54)【発明の名称】 データをテープボリュームコンテナ内に格納するためのシステム

(57)【特許請求の範囲】

【請求項1】

1つまたは複数のハードウェアプロセッサに動作を実行させるプログラムであって、前記動作は、

データストレージテープ内のテープブロックにアクセスするための要求を受信することと、前記データストレージテープ内の前記テープブロックとマッピングされた複数のコンテナのうちのコンテナ内のオブジェクトからデータを取得することと、

前記取得されたデータからテープアーカイブ形式のデータストリームを生成することと、前記テープアーカイブ形式の前記データストリームを前記要求に応答して送信することとを備える、プログラム。

【請求項2】

前記動作は、前記オブジェクトから前記データを取得する前に、前記取得のために前記オブジェクトをオフラインストレージシステムからオンラインストレージシステムに復元することをさらに備える、請求項1に記載のプログラム。

【請求項3】

前記動作は、

前記テープブロックにアクセスするための前記要求を受信したことに応答して、

前記テープブロックに関連する複数のテープブロックを識別することと、

各前記複数のテープブロックを対応する複数のオブジェクトにマッピングすることと、

取得のために前記複数のオブジェクトを前記オンラインストレージシステムに復元するこ

とをさらに備える、請求項 2 に記載のプログラム。

【請求項 4】

前記複数のコンテナは、前記送信を実行するデバイスからリモートにあるクラウドストレージシステム内に実装される、請求項 1 ~ 3 のいずれかに記載のプログラム。

【請求項 5】

前記動作は、前記要求を前記クラウドストレージシステムに対応する形式へと変換することをさらに備える、請求項 4 に記載のプログラム。

【請求項 6】

前記複数のコンテナは、前記送信を実行するデバイスに対してローカルにあるストレージシステム内に実装される、請求項 1 ~ 5 のいずれかに記載のプログラム。

10

【請求項 7】

前記複数のコンテナは、特定のコンテナに関連付けられたメタデータをさらに備え、前記メタデータは、前記データストレージテーブル上の前記データのための識別情報および前記データストレージテーブル上の前記データにアクセスするための命令を備える、請求項 1 ~ 6 のいずれかに記載のプログラム。

【請求項 8】

ハードウェアプロセッサを含む少なくとも 1 つのデバイスを備えるシステムであって、前記システムは、請求項 1 ~ 7 のいずれかに記載のプログラムを実行するように構成される、システム。

【発明の詳細な説明】

20

【技術分野】

【0001】

技術分野

本発明は、データストレージ管理システムに関する。特に、本発明は、テーブルブロックをオブジェクトベースストレージシステム内のオブジェクトにマッピングすることに関する。

【0002】

優先権主張

この出願は、2017年2月13日に提出された米国特許非仮出願第15/431,615号の利益と優先権を主張し、それは参照によりここに組み込まれる。

【発明の概要】

30

【0003】

背景

ストレージシステムは、さまざまな対応するストレージアーキテクチャに従いデータを格納する。ストレージアーキテクチャの例は、オブジェクトベースストレージ、テーブルベースのストレージ、ファイルシステム、ブロックストレージ、および/またはデータベースを含む。

【0004】

オブジェクトベースストレージは、データをオブジェクトとして管理する。各オブジェクトは、データおよびメタデータと関連付けられる。データは、たとえば、エンドユーザおよび/またはビジネスアプリケーションによって使用される情報を含む。例として、セールスアプリケーションのためのデータは、顧客、製品、および販売についての情報を含み得る。別の例として、人事アプリケーションのためのデータは、従業員、部門、および給料についての情報を含み得る。メタデータは、データが設定されおよび/または保存される方法、またはデータそれ自体についての他の特徴を記述する。メタデータは、ストレージシステム内のオブジェクトを操作しおよび/または管理するために使用され得る。オブジェクトベースストレージシステムでは、データは、シーケンシャル方式で読み込まれるのとは対照的に、要求されるとおりに直接的に読み込まれる。

40

【0005】

コンテナ（ここでオブジェクトコンテナとも称される）は、1つまたは複数のオブジェクトを格納する。コンテナは、たとえば、クラス、データ構造、または抽象データ種類であ

50

り得る。

【 0 0 0 6 】

テープドライブは、データをデータストレージテープ（たとえば、磁氣的テープ、または仮想テープボリューム）に読み書きするデータストレージデバイスである。テープベースストレージは、安定しており、したがって、オフラインのアーカイブ用データストレージのために一般的に使用される。テープドライブは、シーケンシャルアクセスストレージを使用する。このため、テープドライブは、データの特定の断片にアクセスするために物理的に巻き取られることを要し得る。テープボリューム上のデータは、テープアーカイブ（TAR）形式で典型的に格納される。典型的に、データストレージテープは、テープ上に何が格納されるかについての情報を含む情報領域を有する。

10

【 0 0 0 7 】

テープデータ、並びに他の種類のデータストレージでは、ブロック（ここで「テープブロック」とも称される）は、一般的に構造データに対して使用される。ブロックは、データの集合である。ブロックにされたデータは、1回に1ブロックずつ読み込まれまたは書き込まれ得、これは、データストリーミングを高速化する。磁氣的テープストレージの場合、ブロック化はまた、必要とされるストレージ量を減らす。このため、ブロック化は、磁氣的テープストレージのためにほぼ常に使用される。

【 0 0 0 8 】

このため、テープドライブは、テープドライブをテープ上の選択された位置にロードすることからの遅延といった、データをシーケンシャルに読み込むことに関連付けられる特有の欠点を有する。典型的に、オブジェクトベースストレージシステムは、データが直接的にアクセスされることができるので、これらの問題を有さない。しかし、いくつかの遅延は、より複雑なオブジェクトベースストレージシステムと関連付けられる。たとえば、マルチティアのストレージシステムが存在し、そこではデータがアクセス可能になる前に、データは、リモート位置から取得されることを要し得る。取得処理は、遅延を生み出す可能性がある。このため、テープドライブストレージの制約をオブジェクトベースストレージへと持ち込むことは、直感に反するよう見えるが、テープドライブシステムは、オブジェクトベースストレージシステムにも適用可能な長い遅延に対処する性能を有する。

20

【 0 0 0 9 】

この章で記載する手法は、請求可能な手法であり、必ずしも過去に考案または請求された手法ではない。したがって、特に明記しない限り、この節に記載されている手法のいずれかが、単にこの節に含まれているという理由だけで、先行技術として認められると想定されるべきではない。

30

【 0 0 1 0 】

図面の簡単な説明

実施形態は、添付の図面の図において限定ではなく例として示される。本開示における「ある」または「1つ」の実施形態への言及は、必ずしも同じ実施形態に対するものではなく、少なくとも1つを意味することに留意されるべきである。

【図面の簡単な説明】

【 0 0 1 1 】

40

【図1】 1つまたは複数の実施形態に従うデータ管理のためのシステムを示す。

【図2A】 1つまたは複数の実施形態に従い、テープボリュームコンテナ内のデータを読み込むためにデータ管理エンジンによって実行される動作セット例を示す。

【図2B】 1つまたは複数の実施形態に従い、テープボリュームコンテナ内のデータを読み込むためにオブジェクトベースストレージシステムによって実行される動作セット例を示す。

【図3】 1つまたは複数の実施形態に従い、テープボリュームコンテナにデータを書き込むためにデータ管理エンジンによって実行される動作セット例を示す。

【図4】 1つまたは複数の実施形態に従い、メタデータを更新するための動作セット例を示す。

50

【図 5】 1 つまたは複数の実施形態に従うシステムのブロック図を示す。

【発明を実施するための形態】

【 0 0 1 2 】

詳細な説明

以下の説明では、説明の目的で、完全な理解を提供するために多くの特定の詳細が述べられる。1 つまたは複数の実施形態は、これら特定の詳細なしに実現され得る。1 つの実施形態において記載された特徴は、異なる実施形態において記載された特徴と組み合わせられ得る。いくつかの例では、既知の構造およびデバイスは、本発明を不必要に不明瞭にすることを避けるためにブロック図形式を参照して記載される。

【 0 0 1 3 】

10

- 1 . 一般概要
- 2 . データ管理フレームワーク
- 3 . テープボリュームコンテナからのデータ取得
- A . データ管理動作
- B . ストレージシステム動作
- 4 . テープボリュームコンテナへのデータ格納
- 5 . 例示の実施形態
- A . メタデータ
- B . 復元動作
- C . 効率的なテープブロックサイズ、暗号化
- D . 物理的なテープボリュームとのインターフェース
- 6 . その他、拡張
- 7 . ハードウェア概要

1 . 概要

データストリームを、テープアーカイブ形式で、テープベースのストレージシステムに読み書きするためのテープ駆動コマンドは、データストレージテープを参照し得る。参照されたデータストレージテープは、データストリームがそこから読み込まれるべき、またはデータストリームがそこに書き込まれるべきデータストレージテープである。

【 0 0 1 4 】

30

1 つまたは複数の実施形態は、データストリームをデータストレージテープ内に格納するためのテープ駆動コマンドを受信したことに応答して、データストリームを、テープアーカイブ形式で、オブジェクトベースストレージシステムに格納することを含む。データ管理エンジンは、テープ駆動コマンド内で参照されたデータストレージテープを、オブジェクトベースストレージシステム内のオブジェクトのコンテナにマッピングする。データ管理システムは、データストリームを、テープアーカイブ形式で、識別されたコンテナの 1 つまたは複数のオブジェクトの組に格納する。データストリームをテープアーカイブ形式で格納するコンテナは、ここでは「テープボリュームコンテナ」とも称され得る。

【 0 0 1 5 】

実施形態では、データ管理エンジンは、データストレージテープ内のテープブロックにアクセスするためのテープ駆動コマンドを受信したことに応答して、データをオブジェクトベースストレージシステムから読み込む。システムは、データストレージテープ内の要求されたテープブロックをコンテナ内の対応するオブジェクトにマッピングする。システムは、データを、テープアーカイブ形式で、コンテナ内の対応するオブジェクトから取得する。システムは、取得されたデータから、テープアーカイブ形式のデータストリームを生成する。システムは、テープ駆動コマンドに応答して、データストリームを、テープアーカイブ形式で、送信する。

40

【 0 0 1 6 】

本明細書に記載されるおよび / または特許請求の範囲に記載される 1 つまたは複数の実施形態は、この概要の章に含まれない場合がある。

【 0 0 1 7 】

50

2. データ管理フレームワーク

図1は、1つまたは複数の実施形態に従うデータ管理フレームワーク100を示す。データ管理フレームワーク100は、データベースストレージシステムにアクセスするためのまたはそこにデータを書き込むための要求に応答して、データをオブジェクトベースストレージシステムにマッピングする。図1において示されるように、データ管理フレームワーク100は、オブジェクトベースストレージシステム110、データ管理エンジン140、およびテーブルブロッククライアント150を含む。1つまたは複数の実施形態では、データ管理フレームワーク100は、図1に示されたコンポーネントよりも多いまたは少ないコンポーネントを含み得る。図1に示されたコンポーネントは、互いにローカルにあり得、または互いからリモートにあり得る。図1に示されたコンポーネントは、ソフトウェアおよび/またはハードウェアにおいて実装され得る。各コンポーネントは、複数のアプリケーションおよび/またはマシンにわたって配置され得る。複数のコンポーネントは、1つのアプリケーションおよび/またはマシンへと組み合わせられ得る。1つのコンポーネントに対して記載された動作は、代替的に別のコンポーネントによって実行され得る。

【0018】

実施形態では、オブジェクトベースストレージシステム110は、データをコンテナ内のオブジェクトとして管理するストレージシステムである。オブジェクトベースストレージシステム110は、複数の異なるストレージユニットおよび/またはデバイスを含み得る。複数の異なるストレージユニットおよび/またはデバイスは、同じ種類であり得もしくは同じ種類でなくともよく、または同じ物理箇所に配置され得もしくは配置されなくともよい。さらに、オブジェクトベースストレージシステム110は、データ管理エンジン140およびテーブルブロッククライアント150と同じコンピューティングシステム上に実装され得、またはその上で実行し得る。代替的にまたは追加的に、オブジェクトベースストレージシステム110は、データ管理エンジン140およびテーブルブロッククライアント150とは別個のコンピューティングシステム上で実装され得、または実行され得る。オブジェクトベースストレージシステム110は、データ管理エンジン140またはテーブルブロッククライアント150に直接接続を介して、またはネットワークを介して通信可能に結合され得る。

【0019】

実施形態では、コンテナ（たとえば、コンテナ120およびコンテナ130）は、オブジェクトの組のための組織的構造である。コンテナの各インスタンスは、対応するオブジェクトの集合を含み得る。オブジェクトの組は、コンテナ識別子によって参照される特定のコンテナに属するとも言及され得る。コンテナのサイズは、コンテナ内のオブジェクトの数に依存する。コンテナは、サイズおよび複雑さにおいて変化し得る。実施形態では、コンテナは、データストレージテーブルにマッピングされる。データストレージテーブルにマッピングされるコンテナは、ここではテーブルボリュームコンテナとも称され得る。データストレージテーブルに対応するテーブルブロックは、テーブルボリュームコンテナの1つまたは複数のオブジェクトにマッピングされ得る。ポリシーは、所与のコンテナが20MBのデータを保持するべきであると特定し得る。別の例として、ポリシーは、特定のインターネットプロトコル（IP）アドレスから来るすべてのデータが特定のコンテナに格納されるべきであると特定し得る。

【0020】

上記のように、コンテナは、オブジェクト（たとえば、オブジェクト121a - 121nおよびオブジェクト131a - 131n）を含む。オブジェクトは、オブジェクトベースストレージシステム110の連続的なメモリ位置または非連続的なメモリ位置に格納され得る。例として、オブジェクトは、上述のように、データおよびメタデータといった複数の部分を含み得る。データおよびメタデータは、連続的なメモリ位置と一緒に格納され得る。代替的に、データは、メタデータとは別個の位置に格納され得る。データおよびメタデータは、同じデバイス上または異なるデバイス上に実装され得る。代替的に、データは、データティア内に格納され得、メタデータは、別個のメタデータティア内に格納され得

10

20

30

40

50

る。データティアおよびメタデータティアは、同じデバイス上または異なるデバイス上に実装され得る。さらに、オブジェクトに対応するデータそれ自体は、データティア内の連続的なメモリ位置または別個の非連続的なメモリ位置に格納され得る。

【 0 0 2 1 】

実施形態では、メタデータは、コンテナ内に何が格納されているか、およびそれをどのように見つけるかを記述する。メタデータ内の情報は、ブロック数、オフセット、どのデータが関連付けられたデータストレージテーブルに格納されているか、およびデータにアクセスするための方法についての命令を含み得る。メタデータは、サイズおよび形式といったデータについての識別情報をさらに特定し得る。

【 0 0 2 2 】

実施形態では、データ管理エンジン 1 4 0 は、1 つまたは複数のソフトウェアおよび / またはハードウェアコンポーネントを管理するために使用されるデータを含む。データ管理エンジン 1 4 0 は、いつどのようにデータがオブジェクトベースストレージシステム 1 1 0 に格納され、またはそこから取得されるべきかを決定するための機能を含み得る。データ管理エンジン 1 4 0 は、データの読み込みおよび / またはデータの書き込みを制御し得る。データ管理エンジン 1 4 0 は、テーブルブロックインターフェース 1 4 2、マッピングエンジン 1 4 4、および変換エンジン 1 4 6 を含む。

【 0 0 2 3 】

実施形態では、テーブルブロックインターフェース 1 4 2 は、テーブルブロッククライアント 1 5 0 と通信するための機能を含む。たとえば、テーブルブロックインターフェース 1 4 0 は、テーブルブロッククライアント 1 5 0 からテーブルブロック内のデータに関連する要求を受信する機能を含み得る。テーブルブロックインターフェース 1 4 2 は、データのストリーミングを、テーブルアーカイブ形式で、テーブルブロッククライアント 1 5 0 へと制御するための機能をさらに含み得る。

【 0 0 2 4 】

テーブルブロックインターフェース 1 4 2 は、データを組織化するための機能をさらに含み得る。例として、テーブルブロックインターフェースは、T A R データをテーブルブロックへと組織化し、1 つまたは複数のテーブルブロックを特定のテーブルボリュームへと組織化するための機能を含み得る。

【 0 0 2 5 】

マッピングエンジン 1 4 4 は、オブジェクトベースストレージシステム内のデータ構造をテーブルベースのストレージシステム内のデータ構造へとまたはそれからマッピングするための機能を含む。マッピングは、ハードウェア / およびまたはソフトウェアによって達成され得る。たとえば、マッピングエンジンは、テーブルブロックに対応するオブジェクトベースストレージシステム内のオブジェクトを作成するソフトウェアレイヤとして実装され得る。マッピングエンジンは、対応するテーブルブロックの識別子に基づきオブジェクトに名称を付し得る。マッピングエンジンは、データストレージテーブルをオブジェクトベースストレージシステム内の対応するテーブルボリュームコンテナにマッピングし得る。マッピングエンジンは、データストレージテーブル内のテーブルブロックをオブジェクトベースストレージシステムのテーブルボリュームコンテナ内の対応するコンテナブロックにさらにマッピングし得る。データストレージテーブル内の特定のテーブルブロックに関する要求は、テーブルボリュームコンテナ内の対応するブロック内のデータを使用してアドレス特定され得る。

【 0 0 2 6 】

変換エンジン 1 4 6 は、コマンドを 1 つの形式から別のものに変換するための機能を含む。変換エンジン 1 4 6 は、コマンドを変換するためにソフトウェアおよび / またはハードウェアを使用し得る。たとえば、データ管理エンジン 1 4 0 は、テーブルドライブからデータを取得するためのコマンドを受信する。このコマンドは、データを取得するためにオブジェクトベースストレージシステムに送られる。このため、変換エンジン 1 4 6 は、まずコマンドをオブジェクトベースストレージシステムによって理解可能な形式に変換する。実施形態では、変換エンジン 1 4 6 は、データを 1 つの形式から別のものに変換する。た

10

20

30

40

50

例えば、変換エンジン 146 は、データ T A R 形式に変換し得る。

【0027】

実施形態では、テーブルブロッククライアント 150 は、データをテーブルブロックとして送受信するための任意の種類のシステムおよび/またはデバイスである。テーブルブロッククライアントは、アーカイブエンジンを含み得る。アーカイブエンジンは、データをアーカイブするためのプログラムである。アーカイブエンジンは、ポリシーに基づき、いつデータをテーブルブロックにアーカイブすべきかを決定するソフトウェアレイヤとして実装され得る。たとえば、ポリシーは、ディスクストレージシステムに保存されたデータがテーブルドライブに、重複して、隔週ごとにアーカイブされるべきであると指令し得る。このポリシーに基づき、アーカイブエンジンは、データをテーブルブロックにアーカイブするための要求を送信する。

10

【0028】

実施形態では、アーカイブエンジンは、階層的ファイル管理システムの一部である。新たに保存されたまたは最近アクセスされたデータは、簡易的なアクセスのために、ディスクストレージといったストレージシステムに保存される。アーカイブエンジンは、データをオブジェクトベースストレージシステムに定期的にアーカイブする。例として、アーカイブエンジンは、ファイルを 5 分毎に保存する。

【0029】

実施形態では、テーブルブロッククライアント 150 は、ユーザからの要求に応答して動作し得る。例として、テーブルブロッククライアント 150 は、ユーザ入力を受信するための機能を有するユーザインターフェース (UI) を含む。たとえば、テーブルブロッククライアントは、テーブルブロッククライアントがユーザからの命令を受信し、選択されたデータをテーブルブロックから取得することを可能とするラジオボタンおよび入力可能なフィールドを有する UI を含み得る。実施形態では、テーブルブロッククライアント 150 は、ストレージシステムである。たとえば、テーブルブロッククライアント 150 は、そのデータをテーブルブロックに定期的にバックアップするローカルなファイルベースのストレージシステムである。ローカルストレージシステムは、データコンテナの名称、コンテナ内のオフセット、テーブル上のデータのための識別情報、およびテーブル上のデータにアクセスするための命令といった、データ位置情報を特定するメタデータを含み得る。

20

【0030】

3. テーブルボリュームコンテナからのデータ取得

A. データ管理動作

図 2 A は、1 つまたは複数の実施形態に従い、テーブルボリュームコンテナからデータを取得するための動作セット例を示す。図 2 A において示される 1 つまたは複数の動作は、変形され、再配置され、または全て省略され得る。従って、図 2 A において示される特定の動作の順序は、1 つまたは複数の実施形態の範囲を限定するものと解釈されるべきではない。

30

【0031】

実施形態では、データ管理エンジンは、データストレージテーブル内のテーブルブロックに対応するデータにアクセスするためのテーブル駆動要求を受信する (動作 202)。たとえば、データ管理エンジンは、アーカイブエンジンから、ユーザによって要求されたファイルのための要求を受信し得る。要求は、ユーザが開こうとしている、アーカイブエンジンによって過去にアーカイブされたファイルを特定し得る。要求は、ファイルを見つけるためのテーブルボリューム、テーブルブロック、およびオフセットを参照するメタデータをさらに含み得る。

40

【0032】

実施形態では、データ管理エンジンは、データストレージテーブルとの関係において参照された要求されたデータが、データストレージテーブル内よりもむしろコンテナオブジェクト内に格納されているかを決定する (動作 210)。要求されたデータがコンテナオブジェクト内に格納されているかを決定することは、テーブルブロックオブジェクトとコンテナオ

50

プロジェクトとの間のマッピングを使用することを含み得る。具体的には、データ管理システムは、要求において識別されたテーブルブロックを使用して、（テーブルブロック、データストレージテーブル）対（ブロック、コンテナ）のマッピングを照会する。テーブルブロックが対応するコンテナブロックとのマッピング内に見つかけられると、データ管理エンジンは、データが対応するコンテナブロック内に格納されていると決定する。データがコンテナブロック内に格納されているかを決定することは、データを格納する特定のコンテナブロックを識別することを含み得る。テーブルブロックがマッピング内に見つからない場合は、データは、いずれのコンテナブロック内にも格納されていない。

【0033】

別の例として、データ管理エンジンは、データがコンテナオブジェクト内に格納されているかをデータに対応するファイル名に基づき決定する。特定のファイル名形式またはファイル名コンテンツは、コンテナオブジェクト内に格納されるデータに対して使用され得る。要求されたファイルのファイル名が、コンテナオブジェクト内に格納されたデータに対して使用された特定のファイル名形式またはファイル名コンテンツを有する場合は、対応するデータは、コンテナオブジェクト内に格納されていると決定される。一度データがコンテナオブジェクト内に格納されていると決定されると、（テーブルブロック、データストレージテーブル）対（ブロック、コンテナ）マッピングは、データを含むコンテナオブジェクトを識別するために使用され得る。

10

【0034】

データがコンテナ内のオブジェクトにマッピングされると、データ管理エンジンは、コンテナからオブジェクトを取得する（動作220）。コンテナからオブジェクトを取得することは、オブジェクトに対する要求をオブジェクトベースストレージシステムに送信することを含み得る。複数のオブジェクトにまたがって格納されたデータは、オブジェクトベースストレージシステムから複数のオブジェクトを取得することを必要とする。複数のオブジェクトを取得することは、特定の開始点から特定の終点のオブジェクトのシーケンシャル取得を特定することを含み得る。

20

【0035】

実施形態では、データ管理エンジンは、1つまたは複数の実施形態に従い、データストリームを、テーブルアーカイブ形式で、取得されたデータから生成する（動作222）。例として、データ管理エンジンは、データをオブジェクトベースストレージシステムから取得された複数のオブジェクトから抽出し得る。データ管理エンジンは、複数のオブジェクトから取得されたデータを連結し、要求しているテーブルクライアントにストリームされるべきファイルを、ファイルアーカイブ形式で、生成し得る。

30

【0036】

実施形態では、データ管理エンジンは、テーブル駆動要求に応答して、データストリームをテーブルアーカイブ形式で送信する（動作224）。データ管理エンジンは、データストリームを、インターネットを介して無線で、および/または配線またはバスを介して物理的に送信し得る。データ管理エンジンは、データストリームをシーケンシャルにテーブルアーカイブ形式で送信し得る。データストリームの受信者は、テーブルボリュームのテーブルブロックから抽出されたデータストリームを受信するのと同様の方法で、データストリームを受信する。

40

【0037】

データがコンテナ内のオブジェクト内にマッピングされない場合は、データ管理エンジンは、データストレージテーブルからデータにアクセスすることを試みる（動作230）。データストレージテーブルからデータにアクセスするための試行は、テーブル駆動要求内で識別されたデータストレージテーブルをロードすることを含み得る。

【0038】

要求内で識別されたデータストレージテーブルが存在しない、またはそうでなければ利用不可である場合、データ管理エンジンは、エラーを生成する（動作240）。たとえば、データ管理エンジンは、そのようなデータおよび/またはデータストレージテーブルが存在し

50

ないというメッセージを送信する。

【 0 0 3 9 】

データストレージテーブルが正常にロードされると、データ管理エンジンは、格納されたテーブル駆動要求内で識別されたテーブルブロック内のデータを取得する（動作 2 3 2）。データ管理エンジンは、テーブル駆動要求内で特定されたテーブルブロックに対応するテーブルボリューム内の開始オフセットに移動する。開始オフセットに移動することは、テーブルボリュームを巻き戻しまたは早送りことを含み得る。一度オフセットにおけるデータがアクセス可能となると、データ管理エンジンは、終端オフセットに対応するテーブルブロックの終端までデータのシーケンシャル読み込みを実行し得る。

【 0 0 4 0 】

実施形態では、データ管理エンジンは、テーブルブロックから取得されたデータをストリーミングする（動作 2 3 4）。データは、シーケンシャルデータストリームとして、たとえば、無線または有線接続を介してストリームされ得る。このため、システムは、データストレージテーブル（物理または仮想）、並びにオブジェクトベースストレージシステムの両方にアクセスすることをサポートすることができる。

【 0 0 4 1 】

例として、ローカルファイルベースデータシステムのユーザが「ファイル X」というファイルを開くことを要求する。ファイルベースデータシステムは、ファイル X が 5 0 M B のサイズであり、テーブルボリュームに格納されていることを示す、ファイル X のためのメタデータを含む。ファイルベースデータシステムは、データストレージテーブル（ t p 2 0 0 0 ）内のテーブルブロック（ 5 0 0 ）内のファイル X のデータにアクセスするための要求を送信する。データ管理エンジンは、データストレージテーブル内のテーブルブロック内のファイル X にアクセスするための要求を受信する。

【 0 0 4 2 】

ファイル X にアクセスするための要求を受信したことに応答して、データ管理エンジンは、要求されたデータがコンテナ内のオブジェクトにマッピングされているかを決定する。データ管理エンジンは、受信されたテーブルブロックおよびテーブルボリューム情報に基づき、オブジェクトベースストレージシステムを検索する。データ管理エンジンは、ブロック 5 0 0 においてテーブル t p 2 0 0 0 に対応するコンテナ内のオブジェクト、すなわち「 d a t a @ t p 2 0 0 0 / b l o c k . 0 0 0 5 0 0 」を識別する。

【 0 0 4 3 】

次に、データ管理エンジンは、オブジェクトをコンテナから取得する。データ管理エンジンは、データストレージテーブル内のテーブルブロックにアクセスするための要求を受信したが、実際にはオブジェクトベースストレージシステム内のオブジェクトにアクセスする必要がある。変換エンジンは、受信された要求をオブジェクトベースストレージシステムに対応する形式に変換しなければならない。一度要求がオブジェクトベースストレージシステムによって理解可能な形式になると、データ管理エンジンは、変換された要求をオブジェクトベースストレージシステムに送信する。変換された要求を受信したことに応答して、オブジェクトベースストレージシステムは、ストリーミングのために要求されたオブジェクトをロードする。この時点で、オブジェクトは、コンテナから取得された。

【 0 0 4 4 】

コンテナからオブジェクトを取得したときに、データ管理エンジン 1 4 0 は、オブジェクトベースストレージシステムから、ブロック 5 0 0 において始まる「 d a t a @ t p 2 0 0 0 」というコンテナ内の読み込みを開始するよう準備する。データ管理エンジンは、ファイル X に対応するデータ量が読み込まれるまで、ブロックの読み込みを継続する。

【 0 0 4 5 】

データ管理エンジンは、オブジェクトベースストレージシステムにオブジェクトの読み込みを開始するように命令することによってブロック読み込みを開始する。データ管理エンジンは、データを「 d a t a @ t p 2 0 0 0 / b l o c k . 0 0 0 5 0 0 」というオブジェクトから取得する。次に、データ管理エンジンは、次のブロックにマッピングされた「

10

20

30

40

50

data@tp2000/block.000501」というオブジェクトからデータを要求することによって次のブロックを読み込む。50MBのデータが要求されたため、データ管理エンジン140は、50MBのデータが読み込まれるまでブロックの読み込みを継続する。

【0046】

次に、データ管理エンジンは、要求に応答してデータストリームをテープアーカイブ形式で送信する。データ管理エンジンが各ブロックを読み込むと、それは、データを、テープアーカイブ形式で、要求しているファイルベースストレージシステムに返却する。データ管理エンジンは、ファイルXを構成するデータを、読み込まれた各テープブロックに対してローカルストレージシステムにシーケンシャルに書き込む。ファイルXのためのすべてのデータが正常に書き込まれた後に、ファイルは、ローカルストレージシステムから通常のファイルとして利用可能である。

10

【0047】

B. ストレージシステム動作

図2Bは、1つまたは複数の実施形態に従い、オブジェクトベースストレージシステム110から見て、テープボリュームコンテナからデータを取得するための動作セット例を示す。図2Bにおいて示される1つまたは複数の動作は、変形され、再配置され、または全て省略され得る。従って、図2Bにおいて示される特定の動作の順序は、1つまたは複数の実施形態の範囲を限定するものと解釈されるべきではない。

【0048】

20

実施形態では、オブジェクトベースストレージシステムは、コンテナ内のオブジェクト内のデータにアクセスするための要求を受信する（動作250）。データ管理エンジンは、要求をオブジェクトベースストレージシステムに転送する前に、要求をオブジェクトベースストレージシステムが理解可能な形式に変換しておき得る。オブジェクトベースストレージシステムは、要求をコンテナ内のオブジェクト内のデータにアクセスするための通常の要求として考える。これは、たとえば、特定のオブジェクトへのポインタ、およびオフセットを含み得る。

【0049】

次に、オブジェクトベースストレージシステムは、データがコンテナ内のオブジェクトからアクセスされることができるかを決定する（動作252）。オブジェクトベースストレージシステムは、要求されたデータが特定された位置に配置されているかを確認する。実施形態では、システムは、チェックサムを実行して、要求されたデータと関連付けられたメタデータ内で特定されたように、オブジェクト内のデータが要求されたデータと一致するかどうかを確認する。チェックサムが一致する場合、システムは、要求されたデータを取得することに進む。

30

【0050】

データがコンテナ内のオブジェクトからアクセスされることができる場合、オブジェクトベースストレージシステムは、データをコンテナ内のオブジェクトから送信する（動作254）。オブジェクトベースストレージシステムは、データストリームを無線でインターネットを介して、および/または配線またはバスを介して物理的に送信し得る。オブジェクトベースストレージシステムは、特定のオブジェクトのための要求に応答して、1回に1つのオブジェクトのデータを送信し得る。

40

【0051】

データがコンテナ内のオブジェクトからアクセスされることができない場合、オブジェクトベースストレージシステムは、エラーを返却する（動作256）。たとえば、オブジェクトベースストレージシステムは、データが特定された位置にないことを示す情報をデータ管理エンジンに送信する。オブジェクトベースストレージシステムは、データが破損している、またはそうでなければアクセス不可能であることを示すエラーメッセージを送信し得る。

【0052】

50

4. テープボリュームコンテナへのデータ格納

図3は、1つまたは複数の実施形態に従いテープボリュームコンテナにデータを格納するための動作セット例を示す。図3において示される1つまたは複数の動作は、変形され、再配置され、または全て省略され得る。従って、図3において示される特定の動作の順序は、1つまたは複数の実施形態の範囲を限定するものと解釈されるべきではない。

【0053】

実施形態では、データ管理エンジンは、データストリームをデータストレージテープ内のテープブロックに書き込むための命令を受信する（動作302）。たとえば、データ管理エンジンは、命令をユーザから受信し、ファイルをテープブロックに保存する。別の例として、アーカイブエンジンは、テープブロックにデータをアーカイブするための命令を送信する。変換エンジンは、受信された命令をオブジェクトベースストレージシステムに対応する形式に変換し、要求の少なくとも一部をオブジェクトベースストレージシステムに転送し得る。

10

【0054】

実施形態では、データ管理エンジンは、データストリームと関連付けられたコンテナ内に既存のオブジェクトが存在するかを決定する（動作304）。たとえば、データ管理エンジンは、要求をオブジェクトベースストレージシステムに送信して、特定されたデータがオブジェクトベースストレージシステム内に存在するかを決定する。実施形態では、データ管理エンジンに対してローカルにあるメタデータは、オブジェクトベースストレージシステム内のテープボリュームコンテナの一覧を作成する。データ管理エンジンは、ローカルメタデータをチェックして既存のオブジェクトがテープブロックに対応するかを決定する。データ管理エンジンは、チェックサムを実行し、オブジェクトに格納されたデータを検証し得る。

20

【0055】

データストリームと関連付けられたコンテナ内に既存のオブジェクトが存在する場合、データ管理エンジンは、オブジェクトデータを更新する（動作320）。実施形態では、データ管理エンジンは、データストリームをオブジェクトベースストレージシステムに、オブジェクトを上書きするための命令とともに送信する。代替的にまたは追加的に、データ管理エンジンは、オブジェクトデータを既存のデータに付け加え得る。送信する前に、データ管理エンジンは、データをTAR形式に変換し得る。

30

【0056】

データストリームと関連付けられたコンテナ内に既存のオブジェクトが存在しない場合、データ管理エンジンは、オブジェクトをデータストリームに基づき生成する（動作306）。実施形態では、データ管理エンジンは、データブロックに対応するオブジェクトを作成する。ファイルシステム内のファイルまたはファイルフォルダと関連付けられ得る従来のオブジェクトとは対照的に、オブジェクトは、データストレージテープ内の特定のテープブロックと関連付けられる。オブジェクトは、データストレージテープ内のテープブロックとして認識可能な形式の名称を付され得る。たとえば、オブジェクトの名称は、対応するテープブロックの名称を含む。テープブロックは、オブジェクト上へとマッピングされる。他の実施形態では、オブジェクトは、既存であり得、またはオブジェクトベースストレージシステムによって生成され得る。

40

【0057】

データ管理エンジンは、オブジェクトに対応するデータストリームとさらに関連付ける。データ管理エンジンは、いくつかのまたはすべてのデータストリームをオブジェクトに格納する。実施形態では、データストリーム全体は、オブジェクトに格納される。別の実施形態では、データストリームは、いくつかのデータブロックに対応するいくつかのオブジェクトに分割される。たとえば、ポリシーは、各データブロックがある量のデータ、たとえば10、15、16、または20MBを保持するべきと指令し得る。比較的大きいファイルに対して、データ管理エンジンは、データをいくつかのオブジェクトに分割しなければならない。比較的小さいファイルに対して、データ管理エンジンは、複数のファイルを

50

オブジェクトに格納する。

【 0 0 5 8 】

データ管理エンジンは、データストレージテーブルをコンテナにマッピングする（動作 3 0 8）。システムは、テーブルボリュームコンテナをオブジェクトベースストレージシステム内に作成する。代替的に、システムは、オブジェクトベースストレージシステム内の既存のコンテナを使用し得る。いずれの場合でも、データ管理エンジンは、コンテナをブロック 3 0 2 での命令において特定されたデータストレージテーブルにマッピングする。実施形態では、データ管理エンジンは、コンテナに特定されたデータストレージテーブルを参照する名称を与える。

【 0 0 5 9 】

データストレージテーブルをコンテナにマッピングした後に、データ管理エンジンは、オブジェクトをコンテナと関連付けられたストレージのために送信する（動作 3 1 0）。例として、オブジェクトは、オブジェクトベースストレージシステムに Small Computer System Interface (SCSI) を介して、オブジェクトをコンテナに格納するための命令とともに送信される。実施形態では、データ管理エンジンは、送信する前に、データを別の形式から TAR 形式に変換する。データ管理エンジンはそして、オブジェクトベースストレージシステム内のコンテナ内に格納するためのデータを、シーケンシャルデータストリームで、TAR 形式で送信する。

【 0 0 6 0 】

例として、ユーザは「ファイル Y」というファイルをローカルファイルシステムに保存する。ローカルファイルシステムは、階層的ファイル管理システムの一部であり、アーカイブエンジンは、ローカルファイルシステムを定期的に検索して、アーカイブされるべきファイルを探す。ファイルは、リモートクラウドストレージ（オブジェクトベースストレージシステム 1 1 0）にアーカイブされる。階層的ファイル管理システムは、先週編集されていないファイルをアーカイブするためのポリシーを有し、アーカイブされるファイルのローカルコピーを消去する。ローカルコピーの定期的消去は、ローカルストレージが制限されているときに必要であり得る。ファイル Y は、1 週間、アクセスされておらず、したがって、アーカイブエンジンは、ファイル Y をアーカイブされるべきファイルとして識別する。アーカイブエンジンは、命令をデータ管理エンジンに送信し、データストリームをデータストレージテーブル内のテーブルブロックに書き込む。特に、データ管理エンジンは、ファイル Y のためのデータを含むデータストリームをデータストレージテーブル「t p 2 0 0 0」にブロック 1 において書き込むための命令を受信する。

【 0 0 6 1 】

データ管理エンジンは、データストリームと関連付けられるコンテナ内に既存のオブジェクトが存在するかをチェックする。データ管理エンジンは、対応するオブジェクトがファイル Y のうちのいくらかのまたはすべての含むと示すメタデータを検索することによって、クラウドストレージシステムに対してデータストリームと関連付けられるオブジェクトを照会する。データ管理エンジンは、そのようないずれのオブジェクトも存在しないという応答を受信する。データ管理エンジンは、データストリームと関連付けられるコンテナ内に既存のオブジェクトが存在しないと決定する。

【 0 0 6 2 】

データストリームと関連付けられるコンテナ内に既存のオブジェクトが存在しないと決定したことに応答して、データ管理エンジンは、オブジェクトをデータストリームに基づき生成する。データ管理エンジンは、新たなオブジェクトを作成する。データ管理エンジンは、オブジェクトをテーブルブロック 1 にマッピングする。データ管理エンジンは、オブジェクト名称が「t p 2 0 0 0」および「1」を含むようにオブジェクトに名称を付与する。オブジェクト名称は、データストレージテーブルおよびテーブルブロック、それぞれを参照する。オブジェクトは、ファイル Y と関連付けられる。

【 0 0 6 3 】

次に、データ管理エンジンは、データストレージテーブルをコンテナにマッピングする。デ

10

20

30

40

50

ータ管理エンジンは、コンテナ名称に「t p 2 0 0 0」を含む新たなオブジェクトに対応するコンテナを生成する。データ管理エンジンは、データストレージテープ、t p 2 0 0 0を、コンテナにマッピングする。コンテナは、テープブロッククライアントによってデータストレージテープとして認識可能である。

【0064】

次に、データ管理エンジンは、t p 2 0 0 0，ブロック1にマッピングされたオブジェクトのオブジェクトベースストレージシステムへの送信を始動する。オブジェクトは、T A R形式で、オブジェクトベースストレージシステムにストリームされる。

【0065】

データ管理エンジンは、テープブロックへのデータ送信のすべてのマーカを有するデータストリームを始動した。実施形態では、一度データストリームが始動されると、データ管理エンジンは、テープブロッククライアントへの制御を送信することができる。テープブロッククライアントは、それがデータをt p 2 0 0 0と称される標準的なテープに書き込むかのように、書き込み動作に進むことができる。

【0066】

5．例示的实施形態

A．メタデータ

1つの例では、システムは、メタデータを使用してデータ動作を管理する。システムは、ローカルのファイルベースストレージおよびリモートのクラウドストレージシステム（オブジェクトベースストレージシステム110）を含む。アーカイブエンジンは、ポリシーに基づき、所与のファイルがオブジェクトベースストレージシステムにアーカイブされるべきと定期的に決定する。ローカルおよびリモートストレージシステム間でファイルを共有することを容易にするために、2つのレイヤのメタデータは、使用される。

【0067】

ローカルストレージでは、データ管理エンジンは、ファイルを格納し、ファイルシステム内に各ファイルと関連付けられたメタデータをアーカイブする。アーカイブメタデータは、所与のファイルが配置されたデータの位置を記述する情報を含む。たとえば、アーカイブメタデータは、（データストレージテープにマップされた）コンテナの名称、そのテープボリュームコンテナ内の位置、およびオフセットを含む。オブジェクトベースストレージシステム内に格納されたデータを記述するメタデータをローカルに格納することは、データがリモートに格納されるときに特に有用である。たとえば、多くのクラウドストレージシステムでは、データを取得するのに最大4時間かかり得る。したがって、リモートに格納されたデータを記述するメタデータを、情報が即時取得のために利用可能であるローカルに格納することが望ましい。

【0068】

オブジェクトベースストレージシステムでは、コンテナメタデータは、いくつかのメタデータコンテナのうちの1つに格納される。コンテナメタデータは、どの情報がコンテナ内にあり、それをどのように見つけるかについての情報を含む。コンテナメタデータは、コンテナに格納された既存のファイルおよびブロックの数を含む。コンテナメタデータは、各ファイルのための対応するブロックとともに、ファイルのリストを含む、過去の書き込みセッションに対するマニフェストをさらに含む。コンテナメタデータはまた、チェックサムを含み得る。メタデータコンテナは、従来のテープ上の情報領域と同様に構成され、メタデータコンテナは、テープブロック上のデータについての情報、および対応するチェックサムを含む。チェックサムは、テープドライブの内部および外部CRCコードと同様であり、これは、テープブロックをスプールする前に長さチェックを可能とする。コンテナメタデータはこのようにして、テープブロッククライアントによって理解可能であるように構成される。

【0069】

たとえば、システムは、「ファイルA」というファイルをローカルのファイルベースストレージシステムへと書き込むための要求をユーザから受信する。ファイルAは、100M

10

20

30

40

50

Bのサイズである。初期的に、ファイルAのためのアーカイブメタデータは空である。

【0070】

アーカイブエンジンは、ローカルストレージシステムを検索してアーカイブされるべきファイルを探す。アーカイブエンジンは、ファイルAがアーカイブされる必要があると識別する。アーカイブエンジンは、データストリームをデータストレージテーブル内のテーブルブロックに書き込むための命令をデータ管理エンジンに送信する。命令は、データを「tp1000」というテーブルボリュームに書き込むことを特定する。アーカイブエンジンは、ローカルストレージシステムからのファイルを、各ファイルが512バイトTARヘッダを有するTAR形式へと変換することを開始する。

【0071】

データ管理エンジンは、ファイルAと一緒にアーカイブされるべきファイルのグループへとTAR形式で配置する。TARデータストリームは、所与のサイズのテーブルブロックへと分割される。ファイルは、データが16MBブロックへとグループ化されることを指令するポリシーに従い、一緒にグループ化される。

【0072】

データ管理エンジンは、データストレージテーブルをコンテナにマッピングする。データ管理エンジンは、コンテナを生成し、それを特定されたテーブルボリュームであるtp1000と関連付ける。データ管理エンジンは、従ってコンテナに「cloudlib.data@tp1000」という名称を付与する。

【0073】

データ管理エンジンが、オブジェクトをコンテナと関連付けられたストレージに対して送信するときに、データ管理エンジンは、メタデータを更新する。図4は、1つまたは複数の実施形態に従いメタデータを更新するための動作セット例を示す。図4において示される1つまたは複数の動作は、変形され、再配置され、または全て省略され得る。従って、図4において示される特定の動作の順序は、1つまたは複数の実施形態の範囲を限定するものと解釈されるべきではない。

【0074】

データ管理エンジンは、データストレージテーブルにマッピングされたメタデータコンテナを識別する(動作402)。データストレージテーブルの名称は、関連付けられたデータおよびメタデータコンテナを識別するために使用される。ファイルAは、「cloudlib.data@tp1000」というデータコンテナに関連付けられ、tp1000にマッピングされる。同様に、メタデータコンテナは、「cloudlib.meta@tp1000」と称される。

【0075】

次に、データ管理エンジンは、チェックサムを実行およびチェックして、データ保護を確実にする(動作404)。チェックサム情報は、アーカイブメタデータおよびコンテナメタデータの両方に格納される。データ管理エンジンはそして、チェックサムを比較する。

【0076】

データ管理エンジンは、メタデータマニフェストを読み込んで、データの付加を開始すべきブロック数を決定する(動作406)。tp1000に対して、メタデータマニフェストは、下記のいくつかのファイル、およびこれらのファイルのためのデータを含む仮想テーブルブロックを含む。

【0077】

file-0000001.0000000(ブロック0-409)

file-0000002.000410(ブロック410-641)

file-0000003.000642(ブロック642-965)

file-0000004.000966(ブロック966-1006)

file-0000005.001007(ブロック1007-1050)

データ管理エンジンは、メタデータマニフェストを読み込み、ファイルAに対して付加されるべきデータが、ブロック1051において開始すると決定する。

10

20

30

40

50

【 0 0 7 8 】

次に、データ管理エンジンは、コンテナと関連付けられたストレージに対してオブジェクトを送信する。データ管理エンジンは、t p 1 0 0 0 にシーケンシャルT A R形式でデータの付加を開始する。データ管理エンジンは、各テープブロックをオブジェクトベースストレージシステムに、順に、非同期送信で送信する。第1に、データ管理エンジンは、ファイルA内の第1のデータブロックをブロック1 0 5 1に送信する。

【 0 0 7 9 】

データ管理エンジンが新たなブロックを書き込むときに、それは、対応するメタデータマニフェストを更新する（動作4 0 8）。データ管理エンジンがデータをブロック1 0 5 1に正しく送信したので、データ管理エンジンは、

f i l e - 0 0 0 0 0 6 . 0 0 1 0 5 1（ブロック1 0 5 1）

をメタデータマニフェストに追加する。ファイルAは、オブジェクトベースストレージシステム内の使用のために所与の数値名称を与えられる。メタデータはまた、このファイルがファイルAと称されることを示す。

【 0 0 8 0 】

データ管理エンジンは、テープブロックの書き込みを継続し、ファイル全体が送信されるまでメタデータを更新する。1 0 2個のブロックが送信された後に、ファイルAは、完全に送信され、テープ書き込みセッションが完了される。

【 0 0 8 1 】

すべてのテープブロックが送信された後に、クローズ動作が実行される。クローズ動作は、データ管理エンジンが任意の未処理のオブジェクト送信を停止させることを引き起こす。データコンテナへの現在の書き込みセッションは、クローズされる。

【 0 0 8 2 】

次に、データ管理エンジンは、完了したメタデータマニフェストオブジェクトである「c l o u d l i b . m e t a @ t p 1 0 0 0 / f i l e - 0 0 0 0 0 6 . 0 0 1 0 5 1」を書き込み、これは、ブロック1 0 5 1 - 1 1 5 2のためのマニフェストエントリを含む。データが書き込まれると、c l o u d l i b . m e t a @ t p 1 0 0 0というメタデータコンテナは、以下の情報を含む。

6個の既存のファイル、合計1 1 5 3ブロック

過去の書き込みセッションに対するマニフェスト：

f i l e - 0 0 0 0 0 1 . 0 0 0 0 0 0（ブロック0 - 4 0 9）

f i l e - 0 0 0 0 0 2 . 0 0 0 4 1 0（ブロック4 1 0 - 6 4 1）

f i l e - 0 0 0 0 0 3 . 0 0 0 6 4 2（ブロック6 4 2 - 9 6 5）

f i l e - 0 0 0 0 0 4 . 0 0 0 9 6 6（ブロック9 6 6 - 1 0 0 6）

f i l e - 0 0 0 0 0 5 . 0 0 1 0 0 7（ブロック1 0 0 7 - 1 0 5 0）

f i l e - 0 0 0 0 0 6 . 0 0 1 0 5 1（ブロック1 0 5 1 - 1 1 5 2）

新たに更新されたメタデータマニフェストは、ファイルAがブロック1 0 5 1 - 1 1 5 2に格納されていることを示す。書き込みが正常だったので、データ管理エンジンは、アーカイブメタデータを更新して、ファイル位置を特定する（動作4 1 0）。ファイルAに対して、アーカイブメタデータは、以下を特定する。

【 0 0 8 3 】

媒体種類：c l

v s n：t p 1 0 0 0

位置：1 0 5 1

オフセット：5 1 2

種類「c l」は、ファイルがクラウド内のテープボリュームコンテナに格納されていることを示す。V s n t p 1 0 0 0は、ファイルがデータストレージテープt p 1 0 0 0にマッピングされたコンテナ内に格納されていることを示す。位置1 0 5 1、およびオフセット5 1 2は、ファイルAがそのクラウドコンテナ内のブロック1 0 5 1、オフセット5 1 2で始まることを示す。

10

20

30

40

50

【0084】

メタデータはまた、読み込み動作のために使用される。今やファイルAがテープボリュームコンテナに書き込まれたので、ファイルAは、以下のようにメタデータを使用して読み込みのために取得されることができる。

【0085】

第1に、システムは、ユーザがファイルAをローカルストレージシステムから読み込むことを試みたことを示す情報を受信する。ユーザがファイルAを読み込もうとしたという通知を受信したときに、データ管理エンジンは、ファイルAのためのアーカイブメタデータをチェックする。アーカイブメタデータは、ファイルAがクラウド内のテープボリュームコンテナに格納されていると意味する「c1」という符号を含む。データ管理エンジンは、チェックサムを実行して、ファイルAが特定された位置に格納されていることを確かめる。

10

【0086】

オブジェクトをコンテナから取得するために、データ管理エンジンは、「1051に位置決める」とのコマンドをオブジェクトベースストレージシステムに送信する。これは、オブジェクトベースストレージシステムに、データブロック1051にマッピングされたオブジェクトからの取得を始動するように指示する。オブジェクトベースストレージシステムは、ブロック1051にマッピングされたオブジェクトの読み込み開始を準備するために、その内部状態を設定する。

【0087】

次に、データ管理エンジンは、「cloudlib.data@tp.1000block.001051」から開始してオブジェクトの読み込みを開始する。各ブロックがロードされると、データ管理エンジンは、チェックサムを計算し、チェックサムをロード時間中に読み込まれるマニフェストからの対応するチェックサムに対してチェックする。

20

【0088】

検証後に、データ管理エンジンは、各ブロックからのデータをシーケンシャルに送信する。ブロックを読み込むと、データ管理エンジンは、次のシーケンシャルブロックに進む。ブロック1051に対するデータを送信した後に、データ管理エンジンは、「cloudlib.data@tp.1000block.001052」に対するデータを返却する。その後、データ管理エンジンは、「cloudlib.data@tp.1000/block.001053」というオブジェクトからのデータを返却する。データ管理エンジンは、すべてのブロックが送信されるまでシーケンシャルにブロックの送信を継続する。

30

【0089】

各ブロックを読み込んだ後に、データ管理エンジンは、予期されたTARヘッダを、アーカイブメタデータに従い検証する。正常である場合、データ管理エンジンは、データをローカルストレージシステムに各テープブロックの読み込みに対してシーケンシャルに書き込む。100MBであるファイルAのために、7個のテープブロックが読み込まれる。ファイルAのデータは、第1のブロックのオフセット512から開始し、最終ブロックのオフセット4194816で終了する。

40

【0090】

ファイルAのすべてのデータがローカルストレージに正常に書き込まれた後に、データ管理エンジンは、ファイルをローカルストレージシステム内でオンラインとしてアーカイブメタデータ内に印す。アーカイブメタデータ内で、媒体種類は「fi」に変更され、ファイルがローカルに格納されたファイルであることを示す。ファイルは、今や、ローカルファイルシステムを通してローカルファイルとして使用するために利用可能である。

【0091】

B. 復元動作

例では、動作は、復元動作をさらに含み、オブジェクトは、取得のためにオフラインストレージシステムからオンラインストレージシステムに復元される。

50

【 0 0 9 2 】

実施形態では、復元動作は、データを読み込むときに実行される。データをオブジェクトベースストレージシステムから取得するときに、データは、オフラインまたはオンライン（復元された）のいずれかであり得る。データがオフラインである場合は、データは、それが取得可能となる前に、復元されなければならない。この処理は、数時間もかかり得る。たとえば、クラウドストレージシステムでは、クラウド内に格納されたいくらかのデータを有することが一般的であり、そこではデータは、直接アクセス可能である。その他のデータは、オフラインストレージセンタ内に格納され、そこではデータが直接アクセス可能になる前に、データは、クラウドに復元されなければならない。

【 0 0 9 3 】

テーブルブロックにアクセスするための要求を受信するときに、データ管理エンジンは、要求されたテーブルブロックがオンラインかオフラインかを決定する。データ管理エンジンは、（章 5 A で説明されたように）ローカルメタデータをチェックして、要求されたデータブロックと関連付けられたオブジェクトの状態を決定し得る。データがオンラインである場合、復元動作は必要とされない。データがオフラインである場合は、復元動作が必要とされる。

【 0 0 9 4 】

実施形態では、要求されたデータがオフラインである場合、システムは、取得動作のためにオブジェクトをオフラインストレージシステムからオンラインストレージシステムに復元する。このステップは、データ管理エンジンからの要求に回答して、オブジェクトベースストレージシステムによって実行され得る。たとえば、データ管理エンジンは、要求を、オブジェクトベースストレージシステムに送信して、データブロックを復元する。要求に回答して、オブジェクトベースストレージシステムは、要求されたデータブロックにマッピングされたオブジェクトを、リモートストレージシステムからオンラインストレージにコピーする。

【 0 0 9 5 】

システムは、いくつかのデータブロックについて一度に復元動作を実行し得る。たとえば、データ管理エンジンは、復元動作を実行する前に、それがデータを取得するためのある数の要求を受信するまで待機する。データ管理エンジンはそして、データブロックのグループに対して復元動作を同時に始める。

【 0 0 9 6 】

実施形態では、データ管理エンジンは、テーブルブロックにアクセスするための要求を受信し、そして、要求されたテーブルブロックに関連付けられた 1 つまたは複数のテーブルブロックを識別する。たとえば、データ管理エンジンは、テーブルブロック B 上のファイル B にアクセスするための要求を受信するデータ管理エンジンは、テーブルブロック C 上のファイル C がファイル B の更新されたバージョンであると決定する。データ管理エンジンはまた、テーブルブロック B およびテーブルブロック C を復元することによって、ファイル C をファイル B とともに、復元する。

【 0 0 9 7 】

復元のためのテーブルブロックを識別することに続いて、データ管理エンジンは、テーブルブロックの各々を対応するオブジェクトにマッピングする。データ管理エンジンは、取得のために、要求されたテーブルブロックと関連付けられたオブジェクトをオンラインストレージシステムに復元する。データ管理エンジンは、同時に復元されるべきオブジェクトを単一のコマンドのように要求し得る。データ管理エンジンは、互いにかかなり近接して復元されるべきオブジェクトを、連続して実行されたオブジェクト毎に 1 つのコマンドのように要求し得る。

【 0 0 9 8 】

たとえば、データ管理エンジンは、10 個のファイルが、オブジェクトベースストレージから取得されなければならないと決定する。10 個のファイルの各々は、異なるテーブルブロックと関連付けられる。データ管理エンジンは、10 個のファイルの各々と関連付けら

10

20

30

40

50

れた 10 個のテープブロックが取得されるべきと識別する。アーカイブエンジンはそして、10 個のテープブロックの各々がそれぞれマッピングされた 10 個のオブジェクトを決定する。データ管理エンジンは、10 個の要求されたブロックが復元されるのを待ち、そしてブロックの読み込みを開始する。

【0099】

たとえば、復元処理は、4 時間かかる。この遅延のために、複数の関連付けられた復元動作を一度に実行することが有利である。そうでなければ、各オフラインオブジェクトが要求されたときに、復元が始動され、各オブジェクトに対して別個の 4 時間の遅延がもたらされるだろう。データ管理エンジンにとって一緒に実行されるべきいくつかの復元動作を要求することがより効率的である。

10

【0100】

C. 効率的なテープブロックサイズ、暗号化

実施形態では、データ管理エンジンは、効率性の考慮に基づき、データを特定のサイズのテープブロックへとグループ分けする。たとえば、多くのクラウドストレージシステムでは、小さいデータブロックについて複数の動作を実行するよりもむしろ、大きいデータブロックを送受信する方が、コスト対効果が高い。いくつかのクラウドストレージシステムは、10 MB よりも小さいファイルを送信することに対してペナルティを課す。

【0101】

実施形態では、データ管理エンジンは、16 MB のオブジェクトサイズを使用する。このオブジェクトサイズは、比較的小さいファイルを送信することにより被るペナルティを回避するということが分かる。オブジェクトサイズは、十分なストリーミング性能を達成するために依然として十分に小さい。各オブジェクトが 16 MB に可能な限り近くなることを確実にするために、データ管理エンジンは、読み込みまたは書き込み動作を実行する前に、データを 16 MB ブロックへと組み立てる。

20

【0102】

たとえば、アーカイブエンジンは、10 個のファイルがアーカイブされるべきと決定する。10 個の選択されたファイルの各々は、4 MB のサイズである。データ管理エンジンは、10 個のファイルを一緒にアーカイブされるべきファイルのグループへと TAR 形式で配置する。アーカイブエンジンは、データブロックが 16 MB のデータを含むように、4 個のファイルをデータブロックへとグループ分けする。

30

【0103】

実施形態では、データ管理エンジンは、データをデータオブジェクトベースストレージシステムに送信する前に暗号化する。今日既知のまたは後に開発された任意の技法は、データを暗号化するために使用され得る。任意の数のコンポーネントは、データを暗号化することに関わり得る。データの暗号化は、任意の特定の技法またはコンポーネントに限定されない。たとえば、データブロックは、参照によって組み込まれる米国特許第 8892905 B2 号(Method and Apparatus for Performing Selective Encryption/Decryption in a Data Storage System)において説明されるように選択的に暗号化され得る。

【0104】

D. 物理的なテープボリュームとのインターフェース

40

実施形態では、システムは、物理テープボリュームからオブジェクトベースストレージシステムへのまたはオブジェクトベースストレージシステムからの、データの物理的インポートまたはエクスポート可能とする。オブジェクトおよびコンテナがテープブロックおよびテープボリュームにマッピングされているので、その内部のデータは、物理テープボリュームからおよび物理テープボリュームへと容易に送信されることができる。

【0105】

たとえば、8 TB のデータがオブジェクトベースストレージシステムにアーカイブされるとする。ネットワークを介してよりもむしろ、そのような大きい量のデータをオブジェクトベースストレージシステムに物理的に送信する方が、はるかにより効率的である。データストレージテープをオブジェクトベースストレージシステムへとマッピングすることは

50

、システムが物理テープボリュームを占有し、テープボリュームをオブジェクトベースストレージシステム内のコンテナにマッピングすることを可能とする。システムは、データをオブジェクトベースストレージシステムにより効率的に移動することができる。同様に、システムは、物理テープボリュームをテープボリュームコンテナから生成することができる。

【 0 1 0 6 】

6 . その他、拡張

実施形態は、ハードウェアプロセッサを含み、ここで説明されるおよび / または下記の請求項のいずれか 1 項に記載される任意の動作を実行するように構成される 1 つまたは複数のデバイスを有するシステムに向けられる。

【 0 1 0 7 】

実施形態では、非一時的コンピュータ可読記憶媒体は、1 つまたは複数のハードウェアプロセッサによって実行されると、ここで説明されるおよび / または請求項のいずれか 1 項に記載される動作の任意を実行させる命令を備える。

【 0 1 0 8 】

本明細書で説明される特徴および機能の任意の組み合わせは、1 つまたは複数の実施形態に従い使用され得る。上述の明細書において、実施形態は、実装ごとに異なり得る多数の特定の詳細を参照して説明されてきた。したがって、明細書と図面は、限定的な意味ではなく、例示的な意味で考えられるべきである。本発明の範囲の唯一かつ排他的な指標および本出願人が本発明の範囲とすることを意図するものは、本出願から発行される特許請求の範囲の文言通りおよび均等の範囲であり、そのような特許請求の範囲は、特定の形式において後の訂正を含む。

【 0 1 0 9 】

7 . ハードウェア概要

1 つの実施形態に従って、ここで説明される技法は、1 つまたは複数の特定用途のコンピューティングデバイスによって実装される。特定用途コンピューティングデバイスは、技法を実行するために有線であり得、または技法を実行するために永続的にプログラムされた 1 つまたは複数の特定用途向け集積回路 (A S I C)、フィールドプログラマブルゲートアレイ (F P G A)、またはネットワーク処理ユニット (N P U) といったデジタル電子デバイスを含み得、またはファームウェア、メモリ、他のストレージ、または組み合わせにおけるプログラム命令に従って技法を実行するようにプログラムされた 1 つまたは複数の汎用ハードウェアプロセッサを含み得る。そのような特定用途コンピューティングデバイスは、カスタム有線ロジック、A S I C、F P G A、または N P U をカスタムプログラミングと組み合わせ、技法を実現し得る。特定用途コンピューティングデバイスは、デスクトップコンピュータシステム、ポータブルコンピュータシステム、ハンドヘルドデバイス、ネットワークデバイス、または技法を実装するための有線および / またはプログラムロジックを組み込んだ任意の他のデバイスであり得る。

【 0 1 1 0 】

たとえば、図 5 は、本発明の実施形態がその上に実装され得るコンピュータシステム 5 0 0 を示すブロックダイアグラムである。コンピュータシステム 5 0 0 は、情報を通信するためのバス 5 0 2 または他の通信機構、および情報を処理するためにバス 5 0 2 と結合されたハードウェアプロセッサ 5 0 4 を含む。ハードウェアプロセッサ 5 0 4 は、たとえば、汎用マイクロプロセッサであり得る。コンピュータシステム 5 0 0 はまた、プロセッサ 5 0 4 によって実行される情報および命令を格納するためにバス 5 0 2 に結合されたランダムアクセスメモリ (R A M) または他の動的ストレージデバイスなどのメインメモリ 5 0 6 を含む。メインメモリ 5 0 6 はまた、プロセッサ 5 0 4 によって実行される命令の実行中に一時変数または他の中間情報を格納するために使用され得る。そのような命令は、プロセッサ 5 0 4 がアクセス可能な非一時的記憶媒体に格納されると、コンピュータシステム 5 0 0 を、命令において特定された動作を実行するようにカスタマイズされた特定用途マシンにする。

10

20

30

40

50

【 0 1 1 1 】

コンピュータシステム 5 0 0 は、プロセッサ 5 0 4 に対する静的情報および命令を格納するためにバス 5 0 2 に結合された読み出し専用メモリ (R O M) 5 0 8 または他の静的ストレージデバイスをさらに含む。磁気ディスクまたは光ディスクなどの記憶装置 5 1 0 が提供され、情報および命令を格納するためにバス 5 0 2 に結合される。

【 0 1 1 2 】

コンピュータシステム 5 0 0 は、情報をコンピュータユーザに表示するために、バス 5 0 2 を介して陰極線管 (C R T) などのディスプレイ 5 1 2 に結合され得る。英数字および他のキーを含む入力デバイス 5 1 4 は、情報およびコマンド選択をプロセッサ 5 0 4 に伝達するためにバス 5 0 2 に結合される。別の種類のユーザ入力デバイスは、方向情報およびコマンド選択をプロセッサ 5 0 4 に伝達し、ディスプレイ 5 1 2 上のカーソル移動を制御するためのマウス、トラックボール、またはカーソル方向キーなどのカーソル制御 5 1 6 である。この入力デバイスは典型的に、第 1 軸 (x など) および第 2 軸 (y など) の 2 つの軸において 2 つの自由度があり、デバイスが平面内の位置を特定可能にする。

10

【 0 1 1 3 】

コンピュータシステム 5 0 0 は、カスタマイズされた有線ロジック、1 つまたは複数の A S I C または F P G A、ファームウェアおよび / またはプログラムロジックを使用して本明細書に記載の技法を実装し、これらはコンピュータシステムと組み合わせてコンピュータシステム 5 0 0 を特定用途マシンにするまたはプログラムする。一実施形態によれば、本明細書の技法は、プロセッサ 5 0 4 がメインメモリ 5 0 6 に含まれる 1 つまたは複数の命令の 1 つまたは複数のシーケンスを実行したことに応答して、コンピュータシステム 5 0 0 によって実行される。そのような命令は、ストレージデバイス 5 1 0 などの別の記憶媒体からメインメモリ 5 0 6 に読み込まれ得る。メインメモリ 5 0 6 に含まれる命令のシーケンスの実行により、プロセッサ 5 0 4 は本明細書に記載の処理ステップを実行する。代替実施形態では、ソフトウェア命令の代わりに、またはそれと組み合わせて、有線回路を使用することができる。

20

【 0 1 1 4 】

本明細書で使用される「記憶媒体」という用語は、マシンを特定の方法で動作させるデータおよび / または命令を記憶する任意の非一時的媒体を指す。そのような記憶媒体は、不揮発性媒体および / または揮発性媒体を含み得る。不揮発性媒体は、たとえば、ストレージデバイス 5 1 0 などの光ディスクまたは磁気ディスクが含まれる。揮発性媒体は、メインメモリ 5 0 6 などの動的メモリを含む。記憶媒体の一般的な形態は、たとえば、フロッピー (登録商標) ディスク、フレキシブルディスク、ハードディスク、ソリッドステートドライブ、磁気テープ、または他の磁気データ記憶媒体、C D - R O M、その他の光学データ記憶媒体、穴のパターンを備えた物理媒体、R A M、P R O M、および E P R O M、フラッシュ E P R O M、N V R A M、その他のメモリチップまたはカートリッジ、連想メモリ (C A M)、および T e r n a r y C o n t e n t - A d d r e s s a b l e M e m o r y (T C A M) を含む。

30

【 0 1 1 5 】

記憶媒体は、伝送媒体と別のものであるが、これと組み合わせて使用されることができる。伝送メディアは、記憶媒体間の情報の送信に関与する。たとえば、伝送媒体は、バス 5 0 2 を構成するワイヤを含む同軸ケーブル、銅線、光ファイバーを含む。伝送媒体は、電波および赤外線データ通信中に生成される音波などの音響波または光波の形態を取ることができる。

40

【 0 1 1 6 】

さまざまな形態の媒体が、実行のためにプロセッサ 5 0 4 に 1 つまたは複数の命令の 1 つまたは複数のシーケンスを運ぶことに関与し得る。たとえば、命令は、初期的にリモートディスクの磁気ディスクまたはソリッドステートドライブで実行されることができる。リモートコンピュータは、命令を動的メモリにロードし、モデムを使用して電話回線で命令を送信することができる。コンピュータシステム 5 0 0 に対してローカルなモデムは、電

50

話線でデータを受信し、赤外線送信機を使用してデータを赤外線信号に変換することができる。赤外線検出器は、赤外線信号で運ばれるデータを受信でき、適切な回路は、データをバス502に配置することができる。バス502は、データをメインメモリ506に運び、プロセッサ504は、そこから命令を取り出して実行する。メインメモリ506によって受信された命令は、プロセッサ504による実行の前または後のいずれかに、ストレージデバイス510に任意に格納され得る。

【0117】

コンピュータシステム500はまた、バス502に結合された通信インターフェース518を含む。通信インターフェース518は、ローカルネットワーク522に接続されたネットワークリンク520に結合する双方向データ通信を提供する。例えば、通信インターフェース518は、統合サービスデジタルネットワーク（ISDN）カード、ケーブルモデム、衛星モデム、または対応する種類の電話回線へのデータ通信接続を提供するモデムであり得る。別の例として、通信インターフェース518は、互換性のあるLANへのデータ通信接続を提供するローカルエリアネットワーク（LAN）カードであり得る。無線はまた、実装され得る。そのような実装では、通信インターフェース518は、さまざまな種類の情報を表すデジタルデータストリームを運ぶ電気信号、電磁信号、または光信号を送受信する。

【0118】

ネットワークリンク520は、典型的に、1つまたは複数のネットワークを介して他のデータデバイスへのデータ通信を提供する。たとえば、ネットワークリンク520は、ローカルネットワーク522を介してホストコンピュータ524またはインターネットサービスプロバイダー（ISP）526によって運営されるデータ機器への接続を提供することができる。ISP526は、今日一般に「インターネット」528と呼ばれる世界規模の packets データ通信ネットワークを通じてデータ通信サービスを次いで提供する。ローカルネットワーク522およびインターネット528は両方とも、デジタルデータストリームを運ぶ電気信号、電磁信号または光信号を使用する。コンピュータシステム500へとまたはそこへとデジタルデータを搬送する、さまざまなネットワークを介した信号、およびネットワークリンク520および通信インターフェース518を介した信号は、伝送媒体の例示的な形態である。

【0119】

コンピュータシステム500は、ネットワーク（複数可）、ネットワークリンク520、および通信インターフェース518を通じて、メッセージを送信し、プログラムコードを含むデータを受信することができる。インターネットの例では、サーバ530は、インターネット528、ISP526、ローカルネットワーク522、および通信インターフェース518を介して、アプリケーションプログラムに要求されたコードを送信する。

【0120】

受信されたコードは、プロセッサ504が受信されたときにそれによって実行され、および/または後で実行するためにストレージデバイス510または他の不揮発性ストレージに格納され得る。

【0121】

上述の明細書では、本発明の実施形態は、実装ごとに異なり得る多数の特定の詳細を参照して説明されてきた。したがって、明細書と図面は、限定的な意味ではなく、例示的な意味で考えられるべきである。本発明の範囲の唯一かつ排他的な指標および本出願人が本発明の範囲とすることを意図するものは、本出願から発行される特許請求の範囲の文言通りおよび均等の範囲であり、そのような特許請求の範囲は、特定の形式において後の訂正を含む。

10

20

30

40

【図面】

【図 1】

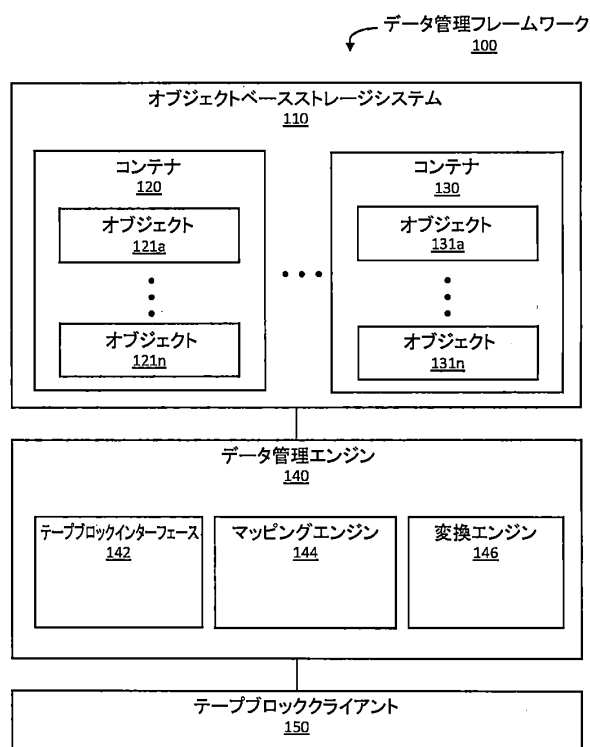


FIG. 1

【図 2 A】

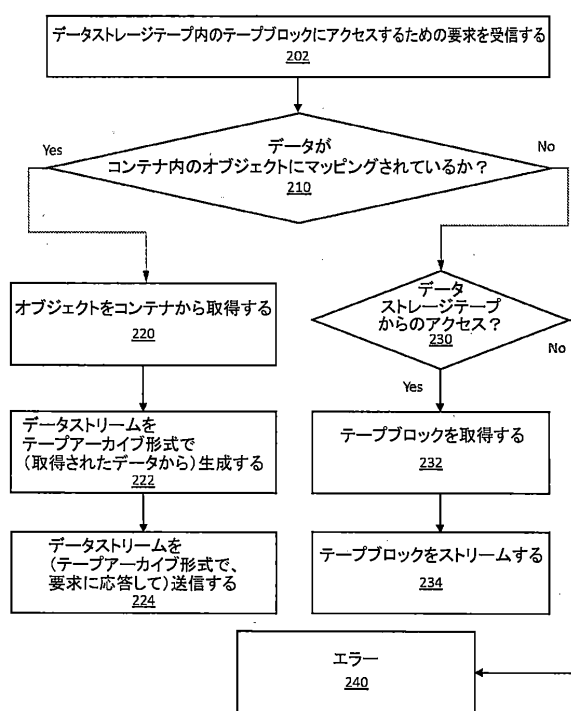


FIG. 2A

【図 2 B】

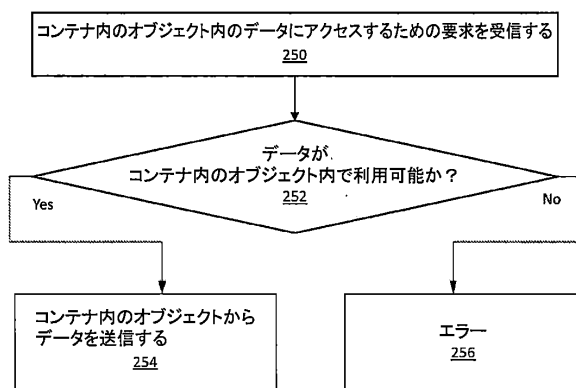


FIG. 2B

【図 3】

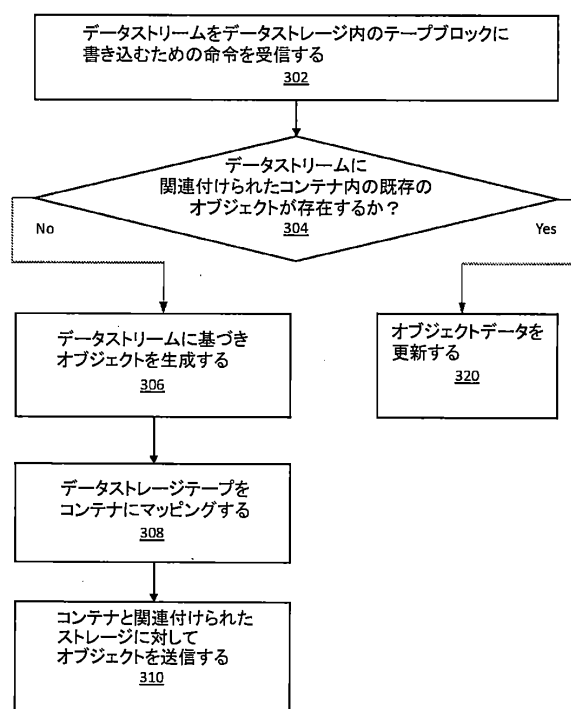


FIG. 3

10

20

30

40

50

【 図 4 】

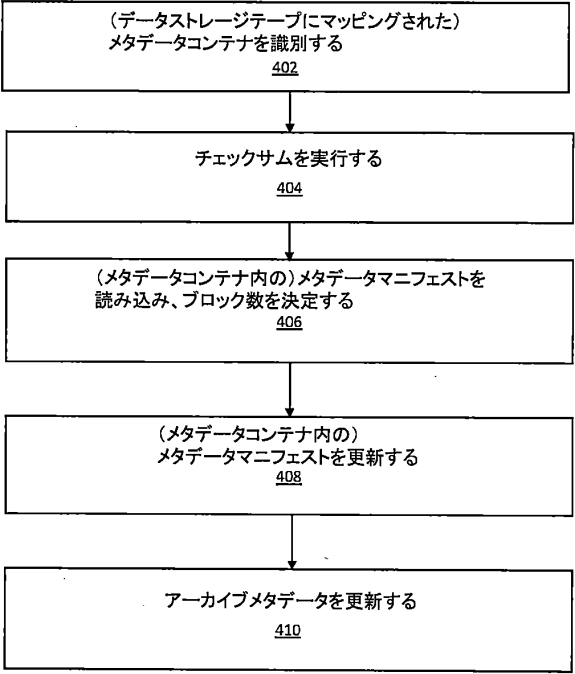
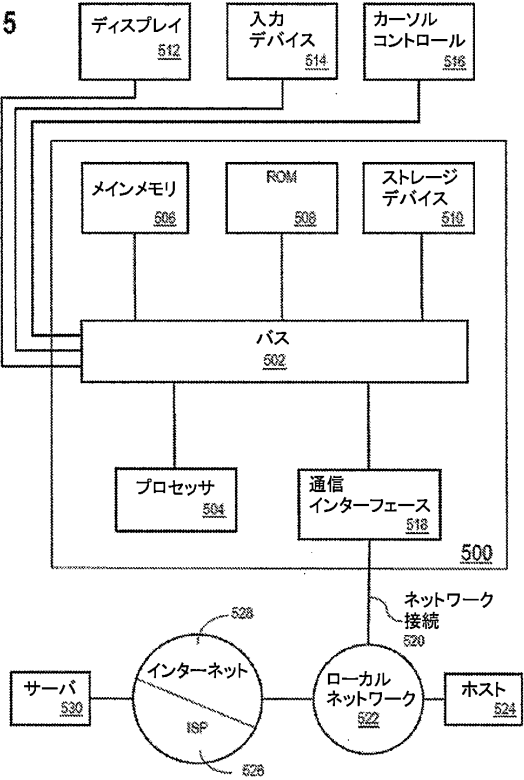


FIG. 4

【 図 5 】

FIG. 5



10

20

30

40

50

フロントページの続き

- ラクル・パークウェイ、５００
- (72)発明者 アンドレ，ジェフリー
アメリカ合衆国、９４０６５ カリフォルニア州、レッドウッド・シティー、オラクル・パークウ
エイ、５００
- (72)発明者 メジャー，デイビッド
アメリカ合衆国、９４０６５ カリフォルニア州、レッドウッド・シティー、オラクル・パークウ
エイ、５００
- 審査官 鹿野 博嗣
- (56)参考文献 米国特許出願公開第２０１４／０１８９２６９（ＵＳ，Ａ１）
特表２００７－５２７５７２（ＪＰ，Ａ）
特表２０１６－５０６５７５（ＪＰ，Ａ）
特表２０１３－５４１０８３（ＪＰ，Ａ）
- (58)調査した分野 (Int.Cl.，ＤＢ名)
 Ｇ０６Ｆ １６／１８５
 Ｇ０６Ｆ ３／０６