(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6280108号
(P6280108)
(45)発行日　平成30年2月14日(2018.2.14)　(24)登録日　平成30年1月26日(2018.1.26)

(51)Int.Cl.　F I
G06F 12/00 (2006.01)　G06F 12/00 531Z
G06F 12/00 531R

請求項の数 17　(全 31 頁)

(21)出願番号　特願2015-517293(P2015-517293)
(86)(22)出願日　平成25年6月4日(2013.6.4)
(65)公表番号　特表2015-519674(P2015-519674A)
(43)公表日　平成27年7月9日(2015.7.9)
(86)国際出願番号　PCT/US2013/044045
(87)国際公開番号　WO2013/188169
(87)国際公開日　平成25年12月19日(2013.12.19)
審査請求日　平成28年5月20日(2016.5.20)
(31)優先権主張番号　13/517,527
(32)優先日　平成24年6月13日(2012.6.13)
(33)優先権主張国　米国(US)

(73)特許権者　505217790
カリンゴ・インコーポレーテッド
ＣＡＲＩＮＧＯ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国　テキサス州７８７３１　オースティン，ノース・キャピタル・オブ・テキサス・ハイウェイ，６８０１，ビルディング　２，スイート　２００
(74)代理人　110000028
特許業務法人明成国際特許事務所
(72)発明者　ベイカー・ドン
アメリカ合衆国　テキサス州７８７４５　オースティン，バクスター・スプリングス・ロード，８０１７

最終頁に続く

(54)【発明の名称】ストレージクラスタにおける消失符号付加および複製

(57)【特許請求の範囲】
【請求項１】
ストレージクラスタにデジタルオブジェクトを格納する方法であって、
前記ストレージクラスタのコンピュータノードでクライアントアプリケーションから、前記デジタルオブジェクトを格納するための要求を受け取り、
前記デジタルオブジェクトを前記ストレージクラスタに、複製を用いて格納するか消失符号付加を用いて格納するかを判断し、
消失符号付加を用いて前記デジタルオブジェクトを格納すると判断された場合、消失符号付加を用いて前記デジタルオブジェクトを前記ストレージクラスタの複数のコンピュータノードに書き込み、前記デジタルオブジェクトは複数のセグメントとして格納され、
消失符号付加の表示と前記ストレージクラスタ内における各前記セグメントの一意識別子とを含むマニフェストコンピュータファイルを作成し、
前記ストレージクラスタのコンピュータノードに前記マニフェストコンピュータファイルを格納し、
前記ストレージクラスタ内で前記マニフェストコンピュータファイルを一意に識別するための、前記マニフェストコンピュータファイル用の一意識別子を算出し、
前記マニフェストコンピュータファイルを識別する前記一意識別子を前記クライアントアプリケーションに返すこと
を備える、方法。
【請求項２】

前記デジタルオブジェクトの固有の特性、前記クライアントアプリケーションからの命令、または前記デジタルオブジェクトのメタデータを参照して、複製または消失符号付加を用いて前記デジタルオブジェクトを格納するか否かを判断すること
をさらに備える、請求項１に記載の方法。
【請求項３】
前記ストレージクラスタ内に前記マニフェストコンピュータファイルを複製し、消失符号付加を用いて前記マニフェストコンピュータファイルを格納しないこと
をさらに備える、請求項１に記載の方法。
【請求項４】
前記デジタルオブジェクトを前記ストレージクラスタ内に複製しないこと
をさらに備える、請求項１に記載の方法。
【請求項５】
前記ストレージクラスタのディスクに格納されている各セグメントについて、前記セグメントに関連付けられている前記ディスクに、前記デジタルオブジェクトの別のセグメントを格納する次のディスクの一意識別子を格納すること
をさらに備える、請求項１に記載の方法。
【請求項６】
前記ディスクの前記セグメントについて前記一意識別子をジャーナルエントリに格納することによって、前記セグメントに関連付けられている前記次のディスクの前記一意識別子を格納すること
をさらに備える、請求項５に記載の方法。
【請求項７】
複数のコンピュータノードを有するストレージクラスタからデジタルオブジェクトを読み出す方法であって、
前記ストレージクラスタ内にある前記コンピュータノードのうちの１つにおいて、前記デジタルオブジェクトについての一意識別子を含むクライアントアプリケーションから要求を受け取り、
複製または消失符号付加を用いて前記ストレージクラスタ内に前記デジタルオブジェクトを格納するか否かを判断し、
消失符号付加を用いて前記デジタルオブジェクトを格納すると判断された場合に、マニフェストファイルを読み出し、前記マニフェストファイルは前記一意識別子によって識別され、
前記マニフェストファイル内で発見された一意的なセグメント識別子を用いて前記ストレージクラスタ内の複数のセグメントを識別し、
前記セグメントおよび消失符号付加アルゴリズムを用いて前記デジタルオブジェクトを再構築し、
前記デジタルオブジェクトを前記クライアントアプリケーションに返すこと
とを備える、方法。
【請求項８】
前記マニフェストファイルを参照することにより、消失符号付加を用いて前記デジタルオブジェクトを格納することを判断すること
をさらに備える、請求項７に記載の方法。
【請求項９】
前記セグメントのうちの１つが前記ストレージクラスタ内に存在しないことを判断し、
他の前記セグメントおよび消失符号付加アルゴリズムを用いて、存在しない前記セグメントを再生すること
とをさらに備える、請求項７に記載の方法。
【請求項１０】
前記セグメントのうちの１つが格納されている第１のディスクを識別し、
もう１つの前記セグメントが格納されている第２のディスクについてのディスクの識別

子を読み出し、前記ディスクの識別子は、前記第１のディスクにある前記セグメントのうちの前記１つと関連付けられて格納されること
とをさらに備える、請求項７に記載の方法。
【請求項１１】
前記マニフェストファイル内で第２の消失セットを識別し、前記第２の消失セットは複数の第２の一意的なセグメント識別子を含み、
前記セグメントを用いて前記デジタルオブジェクトを再構築し、複数の第２のセグメントは、前記第２の一意的なセグメント識別子、および前記消失符号付加アルゴリズムによって識別されること
とをさらに備える、請求項７に記載の方法。
【請求項１２】
前記マニフェストファイルは、前記ストレージクラスタ内で複製され、前記マニフェストファイルは、消失符号付加を用いて前記ストレージクラスタ内には格納されない、請求項７に記載の方法。
【請求項１３】
前記デジタルオブジェクトは、前記ストレージクラスタ内で複製されない、請求項７に記載の方法。
【請求項１４】
請求項１に記載の方法はさらに、
前記デジタルオブジェクトのサイズに応じて、前記デジタルオブジェクトを、複製を用いて格納するか、または消失符号付加を用いて格納するかを判断することを備える、方法。
【請求項１５】
請求項１４に記載の方法はさらに、
前記デジタルオブジェクトのサイズが予め定められたサイズよりも大きい場合には、消失符号付加を用いて前記デジタルオブジェクトを格納することを判断することを備える、方法。
【請求項１６】
請求項７に記載の方法はさらに、
前記デジタルオブジェクトのメタデータを参照することによって、前記デジタルオブジェクトを、複製を用いて格納するか、または消失符号付加を用いて格納するかを判断することを備える、方法。
【請求項１７】
請求項１に記載の方法はさらに、
前記デジタルオブジェクトの固有の性質、前記クライアントアプリケーションからの命令、または前記デジタルオブジェクトのメタデータを参照することによって、前記デジタルオブジェクトを、複製を用いて格納するか、または消失符号付加を用いて格納するかを判断することを備える、方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は概して、消失符号付加（消失訂正符号化）に関する。さらに詳細には、本発明は、固定コンテンツのストレージクラスタにおける消失符号付加と複製との合わせ、および消失符号付加を用いた際のボリューム障害の回復に関する。
【背景技術】
【０００２】
通常、ストレージクラスタ内に備えられているストレージ（独立ノードの冗長アレイ、すなわちＲＡＩＮを用いるものなど）は、格納オブジェクトの複製または格納オブジェクトの消失符号付加のいずれかによってハードウェア障害に対して信頼性が構築されている。前者には、（例えばジャーナルおよびＲＡＭベースのインデックススキームを使用して

）同じ一意識別子が複数のレプリカにアクセスできるという利点があるが、帯域幅が大きくストレージのオーバヘッドがあるという欠点がある（希望するレプリカ数によって異なり、大きいオブジェクトであれば大量のスペースを取ることがある）。後者には、媒体を障害から保護する同様のレベルに対してストレージフットプリントが小さくオーバヘッドが少ないという利点があるが、消失（訂正消失）セットの各セグメントは異なるコンテンツであり、そのコンテンツを別々に識別してオブジェクトを読み出すか、失われた任意のセグメントを再構築しなければならないという欠点がある。ストレージクラスタが再開された場合、この識別は特に問題になることがある。消失符号付加では、さらに高い処理オーバヘッドも発生し、小さいオブジェクトを格納する際にそのフットプリントの利点を失ってしまう。

【０００３】

そのため、どちらの技術にも欠点がある。さらに、消失符号付加に適用可能ないくつかの先行技術による手法では、ストレージクラスタとは別の制御データベースを使用して特定オブジェクトのセグメントを識別し、追跡する。この手法には問題がある。なぜなら、さらに多くのオーバヘッドが生じ、この制御データベースの利用可能性の問題および複製の必要があるかどうかという問題が発生するからである。また、消失符号付加でオブジェクトを再構築できるとしても、そのオブジェクトを符号化するのに使用されるセグメントのサブセットを用いるなら（例えばディスクに障害があった場合）、どのセグメントがもう存在していないかを識別するだけでなく、残りのセグメントを配置するのにも時間がかかるおそれがある。

【０００４】

したがって、複製および消失符号付加の利点を有利にするとともに、ハードウェアに障害が発生した後の曝露を制限するためにストレージクラスタを用いる改良技術が望まれている。

【発明の概要】

【０００５】

上記のことを達成するため、本発明の目的に従って、オブジェクトの複製と消失符号付加とを両方合わせて各々の利点を利用するストレージクラスタを開示する。

１つの実施形態では、ある方法でデジタルオブジェクトをストレージクラスタに格納する。まず、ストレージクラスタは、ストレージクラスタのコンピュータノードで、デジタルオブジェクトを格納する要求をクライアントアプリケーションから受け取る。するとストレージクラスタは、デジタルオブジェクトをストレージクラスタに複製を用いて格納するか消失符号付加を用いて格納するかを判断する。この判断は、クライアントからの命令、オブジェクトの固有の特性、オブジェクトのメタデータ、クラスタの設定を参照して行われてもよいし、他の手段によって行われてもよい。消失符号付加を用いてデジタルオブジェクトを格納すると判断された場合、ストレージクラスタは、消失符号付加を用いてストレージクラスタのいくつかのコンピュータノードにデジタルオブジェクトを書き込み、デジタルオブジェクトは、複数のセグメントとして格納される。さらに、各々のセグメントに対して消失符号付加を示すものと一意識別子とをストレージクラスタ内に含んでいるマニフェストコンピュータファイルが作成される。次にストレージクラスタは、クラスタの１つ以上のノードにマニフェストコンピュータファイルを格納し、マニフェストコンピュータファイルの一意識別子をクライアントアプリケーションに返す。マニフェストは、他のデジタルオブジェクトとは区別できるものである。

【０００６】

もう１つの実施形態では、ある方法で複数のコンピュータノードを有するストレージクラスタからデジタルオブジェクトを読み出す。まず、ストレージクラスタ内のコンピュータノードの１つが、デジタルオブジェクトに対する一意識別子を含んでいるクライアントアプリケーションから要求を受け取る。ストレージクラスタは、そのように識別されたオブジェクトを、そのオブジェクトが格納されている１つのノードで発見する。オブジェクトが前述したようなマニフェストでなければ、オブジェクトはクライアントアプリケーシ

ョンに返される。オブジェクトがマニフェストであれば、次に、マニフェスト内で発見された一意的なセグメント識別子を用いて、ストレージクラスタ内の複数のセグメントを識別する。これらの一意的なセグメント識別子を用いて、本方法では、セグメントおよび消失符号付加アルゴリズムを用いてデジタルオブジェクトを再構築する。最後に、本方法は、デジタルオブジェクトをクライアントアプリケーションに返す。

【０００７】

もう１つの実施形態では、クライアントアプリケーションが、一意識別子に関連付けられているコンテンツを新バージョンのコンテンツに入れ替えることを希望している場合、本発明は、前段落に記載したように動作してオブジェクトを発見し、その後、前段落に記載したようにそのオブジェクトを書き込む。新バージョンは、前バージョンの一意識別子を保持するが、後の作成タイムスタンプを有し、これが更新プロセス中に２つのバージョンを区別する。旧バージョンは、新しい方のバージョンがクラスタにうまく書き込まれた時点で消去される。ヘルス処理モジュールは、エラー状態からクラスタデータを維持する手段として、新しい方のバージョンを存在させるためにオブジェクトの古い方のバージョンを消去してもよい。

【０００８】

もう１つの実施形態では、ある方法で、障害のあるディスクを回復する。まず、本方法は、（複数のコンピュータノードを有する）ストレージクラスタ内でノードのうちの１つに障害が発生したことを検知する。次に、本方法は、ストレージクラスタの別のディスクの永続的なストレージ領域をスキャンして、障害のあるディスクの一意識別子を発見する。この一意識別子は、ストレージクラスタのデジタルストリームに関連付けられている。その後、デジタルストリームをストレージクラスタ内に複製を用いて格納するか消失符号付加を用いて格納するかが判断される。消失符号付加を用いてデジタルストリームを格納すると判断された場合、本方法は、以前に障害のあるディスクに格納された喪失セグメントを識別する。デジタルストリームからシブリング識別子を用いて、本方法は、複数の他のセグメントをストレージクラスタ内に配置する。次に、本方法は、いくつかの他のセグメントおよび消失符号付加アルゴリズムを用いて喪失セグメントを再生する。最後に、本方法は、再生されたセグメントをストレージクラスタのコンピュータノードに格納する。

【０００９】

もう１つの実施形態では、余分な制御コンピュータまたは制御データベースを必要とせずにストレージクラスタ内にセグメントを再配置できる。セグメントが再配置されると、そのセグメントの上流シブリングセグメントにあるボリュームヒントは、シブリングセグメントのメタデータ内で更新される。また、ボリュームヒントは、ディスク上のジャーナルにあるシブリングセグメントのストリーム表現内で更新される。ストレージクラスタは、シブリングセグメントが位置しているディスクを発見するために、シブリングセグメントの一意識別子をブロードキャストできる。マニフェストを用いて、再配置されているセグメントの上流セグメントまたはシブリングセグメントを発見してもよい。

【００１０】

その他の実施形態では、オブジェクトとともに格納されているメタデータまたはクラスタの設定内に格納されているメタデータが、オブジェクトをいつ別の方式に変換すべきかを指定する。トリガー条件が満たされると、クラスタは、複製を用いてオブジェクトをストレージから消失符号付加方式に変換するか、１つの消失符号付加方式を別の消失符号付加方式に変換するか、消失符号付加方式を複製ストレージ方式に変換する。旧方式の元のオブジェクトは、希望すれば消去してもよい。有利には、元のオブジェクトに用いられた一意識別子は、新しいストレージ方式のオブジェクトに使用するために保持され、このようにして、オブジェクトを元々格納していたクライアントアプリケーションが、オブジェクトに備わっていた元の一意識別子を用いて今後いつでもそれを回収できるようにする。

【００１１】

もう１つの実施形態では、１つのストレージ方式（例えば複製、特定の消失符号付加など）を実装している１つのストレージクラスタから、必ずしも同じストレージ方式を実装

する必要のない第２のストレージクラスタにオブジェクトを移すことができる。移されると、オブジェクトは、第２のクラスタが用いたストレージ方式に自動的に変換される。オブジェクトの変換は、第２のストレージクラスタの初期設定によって、オブジェクトの（クラスタの設定を有効にする）ユーザメタデータによって、またはその動きを開始するプログラムからの命令によって、指定されてよい。

【００１２】

一般に、本発明のどの実施形態にも余分な制御データベースは必要ではない。本来、クラスタ内に格納されているデジタルオブジェクトは、その一意識別子を用いて書き込み，読み出しおよび管理ができ、オブジェクトが複製を用いて格納されたのか消失符号付加を用いて格納されたのかどうかは関係ない。

【図面の簡単な説明】

【００１３】

本発明およびその他の利点は、添付の図面とともに以下に記載した説明文を参照することによって最もよく理解できる。

【図１】本発明の動作環境を示す図である。

【図２】オブジェクトに対して５：７消失符号付加を用いた消失セットの一例を示す図である。

【図３】本発明の実施形態で使用できるマニフェストの一例を示す図である。

【図４Ａ】クライアントアプリケーションがどのようにファイルをストレージクラスタに書き込むかを示すフローチャートである。

【図４Ｂ】クライアントアプリケーションがどのようにファイルをストレージクラスタに書き込むかを示すフローチャートである。

【図５】クライアントアプリケーションがどのようにストレージクラスタからデジタルオブジェクトを読み出すのかを示すフローチャートである。

【図６】ストレージクラスタがボリューム障害をどのように回復できるのかを示すフローチャートである。

【図７】オブジェクトをどのように１つのフォーマットから別のフォーマットへ変換できるのかを示すフローチャートである。

【図８】ストレージクラスタ全体の管理をどのように実施できるのかを示すフローチャートである。

【図９Ａ】本発明の実施形態を実装するのに適したコンピュータシステムを示す図である。

【図９Ｂ】本発明の実施形態を実装するのに適したコンピュータシステムを示す図である。

【発明を実施するための形態】

【００１４】

先行技術で公知のように、消失符号付加は、複製のオーバヘッドを伴わずにデータオブジェクトの冗長性を提供する技術である。特定のデータオブジェクトがあるとすると、消失符号付加でオブジェクトはＫ個のデータセグメントに分割され、それらのデータセグメントからＰ個のパリティセグメントが生成され、消失セット内に合計でＭ個のセグメントがある場合、通常これをＫ：Ｍ消失符号と表記する。例えば、データオブジェクトが５個のセグメントに分割され、このセグメントが２個のパリティセグメントを生成するように使用される場合、このデータオブジェクトには５：７消失符号を使用するという。消失符号の主要特性は、消失符号付加したオブジェクトのセグメントが元のデータセグメントであろうとパリティセグメントのうちの１つであろうと、元のオブジェクトを任意のＫ個のセグメントから再構成できるという点である。したがって、各セグメントをストレージクラスタ内の異なるボリューム（および異なるノード）に分配することが有利であり、このようにしてクラスタ内で任意の２つのボリュームが損失することからデータオブジェクトを保護する。クラスタに十分なノードがあるとすると、セグメントは異なるノードに分配され、ノードの損失から保護される。ノードどうしが地理的に異なる領域に位置していれ

ば、セグメントは、その領域内で公平に分配されて、１つの地理的ロケールでクラスタの一部が損失されることから可能な限り保護する。

１つの実施形態では、本発明により、複製または消失符号付加のいずれかを用いてオブジェクトを格納できる。クラスタは、クライアントアプリケーション、オブジェクトのカテゴリ、オブジェクトのサイズ、オブジェクトのメタデータなどからの命令に応じて、オブジェクト単位で切り替わることができる。例えば、比較的大きいオブジェクトはすべて消失符号付加を用いて格納でき、比較的小さいオブジェクトはすべて複製を用いて格納できる。通常の複製は、Ｋ＝１で、Ｍが所与のオブジェクトの合計レプリカ数に等しいという消失符号付加の特別なケースと見なしてよい。また、異なるオブジェクトには異なる消失符号付加を割り当ててよい。極めて大きいオブジェクトの場合、このような消失セットを複数用いてオブジェクトを表現してよい。通常の複製を含む様々な符号化を様々なオブジェクトに割り当てるようにすることによって、本発明は、様々な処理費用およびストレージフットプリントで、データ損失からの保護を様々なレベルで可能にする。

【００１５】

第２の実施形態では、本発明は、消失符号付加によってオブジェクトのセグメントを識別し、発見するという試みで問題に対処する。別々のデータベースを用いる代わりに、マニフェストファイル（またはオブジェクト）が、特定のオブジェクトに関連付けられた各セグメントの記述を含んでいる。マニフェスト内に含まれているものは、各セグメントのクラスタ内にある一意識別子、符号化アルゴリズムを用いる各セグメントのサイズ、およびオブジェクトに対する特定の消失符号付加（５：７など）である。そのためマニフェストは、クラスタ内で通常のオブジェクトとして処理され、一意識別子を提供され、必要に応じて複製される（例えば、同じ冗長度、Ｐ＋１に複製される）。マニフェストの複製は単純である。なぜなら、クラスタは他のオブジェクトに対して既に複製を実施していて、マニフェストが比較的小サイズであるためにストレージのオーバヘッドがほとんどないからである。そのため、消失符号付加した特定のオブジェクトのセグメントには、マニフェストを介して迅速かつ容易にアクセス可能である。このマニフェストオブジェクトに対する識別子は、オブジェクトを今後回収するためにクライアントアプリケーションに返される。これによって、消失符号付加に効果的なフットプリントが提供されるとともに、単純な識別、高い利用可能性、および通常の複製の高速な開始が維持される。

【００１６】

第３の実施形態では、本発明は、ハードウェア障害が起きた後の時間を最短にしてから喪失セグメントをすべて再生することによって、ストレージクラスタ内におけるデータ損失の問題に対処する。消失符号付加したオブジェクトの各セグメントは、オブジェクトの次のセグメントを保有しているクラスタ内のボリューム識別子に関するヒントを含んでいる。このヒントは、正しいボリューム識別子である可能性があるが、保証されないことがある。ハードウェア障害（ディスク障害など）が起きた時点で、かつ特定のセグメントが喪失していることを知らされる前に、クラスタ内の各ボリュームはそのジャーナルをディスク上でスキャンして、障害が発生したボリュームに対するヒントとしてボリューム識別子を有するセグメントを発見する。そのため、クラスタの正常な整合性チェックが起きるのを待つ前に、何らかの喪失セグメントを識別してできる限り迅速に再生できる。

・ストレージクラスタの例

【００１７】

前述したように、本発明は、デジタルオブジェクト、すなわちデジタル形式で表現されたどのような種類の情報にも適用される。例えば、デジタルオブジェクトは、コンピュータファイル、一群のファイル、一群のファイル識別子、またはデータの集合もしくはデータベース情報など、電子表現の情報であってよい。他のこのようなデータの集合には、デジタル音声またはデジタル映像のストリームから得たフレームまたはクリップ、デジタル写真、スキャンした紙の文書、音声メッセージ、ＣＡＤ／ＣＡＭデザイン、ＭＲＩまたはＸ線データ、メッセージレコードまたはファイルから得たストリーム、システムの点検やステータスログから得たログエントリ、電子メールのアーカイブ、チェック画像などがあ

る。本明細書では、電子表現の情報を網羅するために、「コンピュータファイル」という用語をしばしば使用する。

【００１８】

本発明は、任意の適切なコンピュータハードウェアおよびソフトウェアを用いて実装でき、任意数のコンピュータノードを含むストレージクラスタに実装できるものである。好ましくは、各ノードは、１つのＣＰＵ（または複数のＣＰＵ）、オペレーティングシステム、他のノード（または、少なくとも１つの中央ルータ）への通信リンク、および任意数（すなわちゼロからＮ個まで）の内部ハードディスクドライブまたはソリッドステートドライブで、しばしばボリュームと呼ばれるものを含んでいる。通常、各ノードは、少なくとも１つのドライブを含み、ソリッドステートドライブとハードディスクドライブとのあらゆる組み合わせがあってよい。ストレージクラスタは通常、固定コンテンツのクラスタである。つまり、バックアップ、長期にわたるストレージ、アーカイブ保存などに使用されるものであり、通常はコンピュータファイルへ日常的にアクセスするのに使用されるものではない。しばしばＷＯＲＭ（ｗｒｉｔｅ　ｏｎｃｅ，ｒｅａｄ　ｍａｎｙ）ストレージと呼ばれるもので、つまり、一度コンピュータファイルまたはデジタルオブジェクトがクラスタに書き込まれると、変更できないということである。（もちろん、ファイルは消去で、コンピュータファイルの修正バージョンもクラスタ内に格納できる）。クラスタは、独立ノード冗長アレイ（ＲＡＩＮ）として、つまり各ノードがそれ自体のオペレーティングシステムで動作し、クラスタ内のストレージに関して独自の決定を下すものとして実装できる。ストレージクラスタは、ブレードコンピュータ、タワーコンピュータ、パーソナルコンピュータおよびサーバ上に構築できる。このようにする代わりに、単一のコンピュータボックス内にあるマルチコアプロセッサで、各コアで動作する仮想ストレージのノードをサポートしてもよい。つまり、複数のノードを有するストレージクラスタが単一のコンピュータボックス内にあってよいということである。さらに、単一の物理的なボックスの内部にあるコンピュータシステムが複数のＣＰＵを備えることができ、その場合、各ＣＰＵが１つのノードであってよく、ストレージクラスタは、単一の物理的なボックス内に実装できる。

【００１９】

図１は、本発明の動作に対する環境１００を示している。この図に含まれているものは、ストレージクラスタ１２０、クライアントアプリケーション１３０、管理コンソール１４０、任意数のコンピュータノード１０～５０、および中央ルータ１７０である。前述したように、コンピュータノードは通常、好ましくは少なくとも１つのＣＰＵと、任意数のディスクドライブ１６０、ソリッドステートドライブまたはこの両タイプを有するハイブリッドドライブとを備えている物理的なファイルサーバである。特定の一実施形態では、ストレージクラスタ１２０は、さらに論理的または物理的にサブクラスタに分割できる。例えば、ノード４０および５０を１つのサブクラスタとみなし、ノード１０、２０および３０を第２のサブクラスタとみなしてよい。１つのクラスタをサブクラスタに分割することは、１つのサブクラスタが別のサブクラスタとは地理的に異なる場所に位置している場合に有利になり得る。

【００２０】

各ノードは、Ｄｅｂｉａｎ　Ｌｉｎｕｘ（登録商標）などのオペレーティングシステムを実装し、ノード間のピアツーピア通信を管理し、ヘルス処理を実施し、ノードおよびそのボリュームに代わって独立して決定を下すためのプロセスを実行する。各ノードは管理ソフトウェアも備え、そのステータスは、インターネット上のウェブブラウザを介して見ることができる。ある特定のＲＡＩＮの実施形態では、各ノードは、標準のイーサネット（登録商標）ネットワークで１テラバイト以上のシリアルＡＴＡディスクストレージ容量を有する１Ｕサーバ（例えばｘ８６コンピュータ）である。各ノードはＩＰアドレスを有し、ＩＰベースのＬＡＮ、ＭＡＮまたはＷＡＮを用いて物理的に相互接続されてよい。そのため、各ノードは、単一のノードに通信してもよいし、ルータ１７０またはその他の同様のネットワークスイッチを用いて、１つのメッセージをストレージクラスタ内の全ノー

ドにブロードキャストしてもよい（マルチキャスト）。
【００２１】
各ノードは、クライアントアプリケーションからの外部要求（例えばクライアント１３０からのＳＣＳＰ要求）、ノード間の複製要求（例えばｉｎｔｅｒＳＣＳＰ要求）、およびその他のインターノードプロトコル通信（ビッディング、情報の要求などを取り扱うための管理モジュールを備える。ヘルス処理モジュールは、各ノードのデジタルコンテンツを管理する。管理コンソール１４０は、好ましくは、任意の適切なインターネット接続を介して各ノードにアクセスできるストレージクラスタに接続されるウェブサーバである。各ノードは、クラスタ全体を見て管理するのに使用できる冗長管理コンソールを実装する。いくつかの実施形態では、全ノードが同等であるとみなされ、クラスタ内の他の全ノードに関連する情報を定期的にブロードキャストする（または「マルチキャストする」）ことで互いに通信する。
【００２２】
１つの実施形態では、ストレージクラスタは、テキサス州オースチンのＣａｒｉｎｇｏ社から市販されているコンテンツストレージソフトウェア（本明細書に記載の通りに修正）、および任意の適切なコンピュータハードウェアを用いて実装できる。この実施形態では、ストレージクラスタが固定コンテンツのコンテンツアドレスストレージを実装し、各デジタルオブジェクトは、乱数発生器を用いてそのデジタルオブジェクトに対して発生した乱数（汎用一意識別子、すなわちＵＵＩＤ）により、一意的にクラスタ内でアドレスが指定される。各デジタルオブジェクトのコンテンツは、ハッシュ関数を用いて検証できる。クライアントソフトウェアアプリケーションは、クラスタ内にデジタルオブジェクトを格納する際にＵＵＩＤを受け取り、そのＵＵＩＤをクラスタに供給することによってそのデジタルオブジェクトを回収する。ソフトウェアアプリケーションは、ＨＴＴＰ１．１規格を用いて、さらに詳細には、その規格の簡易サブセットであるＳｉｍｐｌｅ Ｃｏｎｔｅｎｔ Ｓｔｏｒａｇｅ Ｐｒｏｔｏｃｏｌ（ＳＣＳＰ）と呼ばれるものを用いて、ＣＡＳｔｏｒクラスタと通信する。この標準インターフェースを用いて、電子メール、企業コンテンツの管理、ヘルスケアアプリケーション、ウェブブラウザ、Ｗｅｂ２．０サイト、写真の共有、ソーシャルメディアサイト、セキュリティ映像、映像編集などのクライアントアプリケーションは、ＣＡＳｔｏｒストレージクラスタにアクセスして、ファイルを格納したり、ファイルを回収したり、ファイルを消去したりできる。さらに、ブラウザ、ＪＡＶＡ（登録商標）、Ｐｙｔｈｏｎ、Ｃ＋＋、およびその他のソフトウェア環境で直接ＨＴＰＰアクセスが可能である。
【００２３】
１つの実施形態では、デジタルオブジェクトが以下のようにして特定のノードに格納される。各ノードは、ＲＡＭリストにディスクインデックスを含み、その中には、オブジェクトを含むデジタルストリームが一意識別子に基づいてディスクに格納されている。例えば、インデックスの第１の列にはオブジェクトの一意識別子が記載され、第２の列にはストリームが始まるセクタが記載され、第３の列にはストリームの長さまたはストリームが終わるセクタが記載される。ストリームは、デジタルオブジェクトのほかに関連するメタデータを含んでいてよい。したがって、ノードに格納されるオブジェクトは、単純に、ディスクおよびＲＡＭインデックスに記録されたそのディスクの場所に順次書き込める。あるいは、オブジェクトは、任意の適切なストレージアルゴリズムを用いてディスクのどこに格納されてもよく、オブジェクトの場所はインデックスに再び記録される。オブジェクトが読み出されるか消去されることになっている場合、ディスク上にあるその場所は、このインデックスを調べることによって発見できる。ノードの再起動時にこのＲＡＭインデックスを簡単に構築するために、永続的なストレージに格納されているノードのジャーナルは、オブジェクトが追加されるか消去されるときは常に記録し、オブジェクトに対する一意識別子、オブジェクトが始まるセクタ、セクタ内でのオブジェクトの長さやバイト、および後述するその他の情報を含む。したがって、ノードが再起動された時、ジャーナル内の情報は読み出され、ＲＡＭにディスクインデックスを作成するために使用される。ジ

ャーナルを使用する代わりにインデックスを構築するもう１つの技術が、再起動時にディスク全体を読み出して必要な情報を集めることだが、これにはさらに多くの時間がかかる。

【００２４】

オブジェクトは格納でき、重複しているものは消去できる。これは、上記に引用した「Ｔｗｏ Ｌｅｖｅｌ Ａｄｄｒｅｓｓｉｎｇ ｉｎ Ｓｔｏｒａｇｅ Ｃｌｕｓｔｅｒｓ」および「Ｅｌｉｍｉｎａｔｉｏｎ ｏｆ Ｄｕｐｌｉｃａｔｅｓ ｉｎ Ｓｔｏｒａｇｅ Ｃｌｕｓｔｅｒｓ」に記載の通りである。

・消失セットの例

【００２５】

図２は、オブジェクトに対して５：７消失符号付加を用いた消失セット２００の一例を示している。図示したように、元のオブジェクトのデータは、５つのデータセグメント（ｋ１～ｋ５）２１０～２１８に分離され、これらのセグメントから２つのパリティセグメント（ｐ１およびｐ２）２２０および２２２が生成される。１つの実施形態では、ストライプ（ｓｔ１～ｓｔ９）２３１～２３９にデータが書き込まれ、パリティが生成される。例えば、第１のストライプ２３１は、元のデータ２５１～２５５で構成され、これらのデータからパリティデータ２５６および２５７が生成される。どの残りのデータも最後の残りのストライプ（ｒｅｍ）２４０として形成され、ハッシュメタデータは、各セグメントの末尾にあるセクション２７０に格納できる。

【００２６】

先に述べたように、クラスタ内に格納されるオブジェクト（またはストリーム）は、所与のサイズである複数の消失セットに分割でき、このサイズはパフォーマンスに応じて決定される。極めて大きいオブジェクトは、例えば複数の消失セットに分割されてよい。消失セット内では、Ｋ個のデータセグメントおよびＰ個のパリティセグメントは、各Ｋ個のデータセグメント内に順に書き込まれている固定サイズのデータブロックを備えるストライプを用いて書き込まれ、次に、パリティブロックを生成して各Ｐ個のパリティセグメント書き込まれる。（Ｋ＋Ｐセグメント全体にわたる）各ストライプは、消失符号付加ユニットの役割を果たす。セグメント内の最後のストライプ（例えば残りのストライプ２４０）は、容易に計算できる小さめのブロックサイズであってよい。ストライプは通常、入力データが使い果たされるまで、あるいは消失セットの所与のサイズが一杯になるまで、次のデータが新たな消失セットを開始している状態で書き込まれる。

【００２７】

特定の実施形態では、単一の書き込み動作から得られたデータが、消失セットのＫ個のセグメントすべてにわたって固定サイズブロック（例えば３２ｋバイト）でストライプに書き込まれる。換言すれば、ブロック１、Ｋ＋１、２Ｋ＋１などは、第１のセグメント２１０に書き込まれ、ブロック２、Ｋ＋２、２Ｋ＋２などは、第２のセグメント２１２に書き込まれる、というふうになる。Ｐ個のパリティブロックは、データの各ストライプ生成物の一部として同時に生成され、各ストライプの末尾に増えていくようにして書き込まれる。ストライピングによって、予測可能なメモリのオーバヘッドと同時にパリティセグメントを生成できるとともに、チャンク形式で符号化したストリームを書き込むことも可能になり、普通はこれによって書き込み動作の開始時にコンテンツの長さは提供されない。もちろん、元のデータオブジェクトは、消失符号付加を実施するためにストライプに分割される必要はないが、こうするといくらか効果的になる。というのも、パリティセグメントは各ストライプに対して生成でき、その後次のストライプに続くからである。また、ストライピングが用いられなければ、第１のデータセグメント２１０は、データオブジェクト内に第１のデータブロックを含み、第２のデータセグメント２１２は、次のデータブロックセットを含む、というふうになる。セット内の各セグメントの最後にあるのは、各セグメントに対するメタデータ２７０であり、このメタデータは、そのセグメントに対するＭＤ５ハッシュ値を含んでいる。例えば、各ハッシュ値は１６バイトであり、セパレータなしで書き込まれる。

【００２８】
消失セットの各セグメントは、不変のストリームであり、クラスタの任意の他のストリームと同じように、各セグメントがそれ自体の一意識別子を有する。各消失セグメントは、補足的ヘッダ２８０も含み、このヘッダは、そのマニフェストの一意識別子など、消失セット内の他のセグメントに関するメタデータを含むほか、全データおよびパリティセグメントの一意識別子も順に含む。各セグメント自体の一意識別子は、一連のシブリング内でその場所を識別する。各セグメントに対するヘッダは、セグメントが位置している所のボリューム識別子や、消失セットの次のセグメントに対して可能性のあるボリューム識別子などのシステムのメタデータも含む（ボリュームヒント）。例えば、セグメントｋ４は、セグメントｋ５が位置している所のボリューム識別子を含み、セグメントｐ２は、セグメントｋ１が位置している所のボリューム識別子を含む、というふうになっている。好ましくは、（単一のオブジェクトを表す）より大きいストリームの一部である各消失セットは、すべての消失セットに対して同じ消失符号付加、例えば５：７の消失符号付加を有する。

【００２９】
書き込み動作の過程で、図示した７つのセグメントの各々は、クラスタの７つの異なるノードを用いて平行して書き込まれてよく、万一ノードまたはボリュームに障害が発生した際にはこのようにしてデータを保護する。セグメントがボリュームに書き込まれると、そのセグメントは、クラスタ内の任意の他のストリームと同じように処理されてよい（ボリュームおよびそのコンテンツ、その回復プロセス、ならびに複製する必要がないという点を除く）。

・マニフェストの例

【００３０】
図３は、本発明の実施形態で使用できるマニフェスト３００の一例を示している。先に述べたように、消失符号付加を用いてデータオブジェクトが任意数のセグメントに符号化されると（用いた符号化によって異なる）、それらのセグメントに対する一意識別子は、後で回収するためにマニフェスト内に格納される（マニフェスト自体はクラスタ内に格納されたオブジェクトであり、それ自体の一意識別子を有する）。

【００３１】
各マニフェスト内に含まれているのは、メタデータセクション３１０、少なくとも１つの消失セット３４０および任意数の他の消失セット３８０である。もちろん、マニフェスト内にある消失セットの数は、データオブジェクトのサイズ、各セグメントのサイズ、用いた符号化スキームによって異なる。マニフェストのセクション３１２は、使用した消失符号付加アルゴリズムの名称およびそのバージョン番号を提供する。セクション３１４には特定の消失符号付加を示し、セクション３１６にはセグメント内の各データブロックのサイズ（ストライプ幅）を示している。このメタデータセクションにある他の情報には、オブジェクトが書き込まれた時間、その一意識別子、その所有権、およびアプリケーションで指定されたメタデータがある。

【００３２】
第１の消失セット（または唯一の消失セット）の表現には、そのサイズ３４２、この消失セット３４４に使用された消失符号付加、セグメント幅３４６（単位はバイト）、および合計セグメントサイズ３４８（単位はバイト）が含まれている。符号化およびセグメント幅のよう情報は、柔軟性を提供するために各消失セット内にもメタデータセクション３１０にもあってよいし、このようなデータのみが１つの領域にあってもよい。

【００３３】
第１の消失セットは、その消失セット内にある各々のセグメントに対する一意識別子も含んでいる。例えば、セクション３５０は、第１の消失セットにあるセグメントに対する一意識別子を示している。前述したように、マニフェストは、１つ以上の消失セットを含んでいてよい。その状態では、第２の消失セット３８０の表現が、任意数の他の消失セットの他の表現と同じように含まれてよい。第２の消失セットは、（第１のセットとは異な

っていてよい）サイズ、符号化、および第１の消失セットと同様の情報を示している。第２の消失セットは、セグメント幅にもセグメントサイズにも第１のセットと同じ符号化を用いてよいが、これは一般的な要件ではない。

【００３４】

どのような追加の消失セットでも、その消失セット内に含まれるすべてのセグメントに対して一意識別子を含んでいる。好ましくは、マニフェスト自体は複製されるが消失符号付加されない。これがスペースの効率を損なうことはない。なぜなら、マニフェストは、そのマニフェストの対象になっているデータオブジェクトのスペースのうちの小部分しか使用しないからである。このほか、消失セットの消失符号付加と同時に起こるディスク障害から同じように保護するために、Ｋ：Ｍ（Ｐ＝Ｍ－Ｋ）符号化を用いて符号化したデータオブジェクトに対するマニフェストは、複製の合計数が少なくともＰ＋１である必要がある。

・クラスタへのデジタルオブジェクトの書き込み

【００３５】

図４Ａおよび図４Ｂは、クライアントアプリケーションがどのようにファイル（デジタルオブジェクト）をストレージクラスタに書き込むかを示すフローチャートである。ステップ４０４では、任意の適切なクライアントアプリケーション１３０がデジタルオブジェクト（任意のコンピュータファイル、デジタル画像、デジタル映像、ヘルス記録など）をストレージクラスタ１２０に格納しようとする。クライアントアプリケーションは、クラスタ内にあるノード１０～５０のうちの１つのＩＰアドレスを発見するか取得し、ストレージプロセスを開始するプライマリアクセスノード（ＰＡＮ）としてそのノードを選ぶ。通常、クライアントアプリケーションは、次の書き込み要求のために使用する最後のノードにアクセスする。クライアントアプリケーションは、ＰＡＮに要求を送ってデジタルオブジェクトを格納する。１つの実施形態では、この要求はＨＴＴＰ　ＰＯＳＴ要求であり、これは、ヘッダ、バイト単位でのデジタルオブジェクトの長さ、およびオブジェクト自体を含んでいる。この要求に対するＰＡＮからクライアントへの応答は、次のうちのいずれかになる。はい、ＰＡＮはオブジェクトの格納を簡易化できます、いいえ、オブジェクトを格納するためにはこのノードの方がいいです、あるいは、いいえ、このクラスタ内にそのオブジェクトを格納できるノードはありません。

【００３６】

ＰＡＮがオブジェクトの格納を簡易化すると仮定すると、クライアントは、格納するデジタルオブジェクトをその時点で動かしてしいても、ストレージ用にセカンダリアクセスノード（ＳＡＮ）が選択されるまでオブジェクトを動かすのを待つことができる。オブジェクトが最初に動かされていなければ、クライアントは、例えばオブジェクトのサイズ、長期または短期のストレージが望ましいかどうか、オブジェクトが今後頻繁にアクセスされるかどうか（これらはすべて、ストレージノードの選択時に役に立つことができる）、また、任意選択でオブジェクトに対するファイル名などのオブジェクトメタデータを動かすことができる。いくつかのクライアントには、クライアントがクラスタ内に格納するオブジェクトに対して階層式のファイル名または任意のファイル名を使用することが必要であり、このような状況では、このようなファイル名からハッシュ値を引き出して、一意識別子として使用してよい。

【００３７】

しかし、本発明にさらに関連することは、複製または消失符号付加を用いてオブジェクトを格納するかどうかを決定するのに使用できるオブジェクトメタデータである。書き込み要求があるとき、あるいはオブジェクトメタデータでは、クライアントアプリケーションはこのオブジェクトを、複製を用いて格納すべきか消失符号付加を用いてすべきかを指定できる（特定の消失符号付加を指定してもよい）。実際、クライアントは、書き込まれるすべてのオブジェクトを複製または消失符号付加を用いて格納するように指定してよい。クライアントからの特定の命令がない場合、本発明は、多様な情報のうちのいずれかを用いて、オブジェクトに対して複製を選択するか消失符号付加を選択するかを決定してよ

い。例えば、オブジェクトのサイズを用いてもよいし（一定サイズを上回るオブジェクトが消失符号付加を用いて格納される）、オブジェクトの種類（画像ファイル、テキスト情報、ヘルス記録など）を用いてもよいし、オブジェクトの所有権、および予想された寿命を用いてもよい。書き込み後にオブジェクトをいつ変換するか、また変換するのかどうかを決定するのに用いられる補足的メタデータについて、以下で考察する。

【００３８】

ステップ４０８では、セカンダリアクセスノード（ＳＡＮ）が選択されると、ＳＡＮは、現在のオブジェクトを格納するために複製を用いるのか消失符号付加を用いるのかを決定する。前述したように、ＳＡＮは、この決定を下すために、クライアントアプリケーションからの命令を用いてもよいし、オブジェクトメタデータに基づく任意の適切な規定を用いてもよい。特定の一実施形態では、デジタルオブジェクトのサイズは基準として用いられる。つまり具体的には、１０メガバイト未満のサイズであるオブジェクトは複製されるが、それよりも大きいオブジェクトはいずれも消失符号付加を用いて格納される。一般に、複製を用いるか消失符号付加を用いるかを決定するのに使用できる情報は、クラスタの管理者が設定するストレージクラスタの何らかの規定または設定、オブジェクト自体またはオブジェクトのメタデータの何らかの固有の特性、およびクライアントアプリケーションからの問い合わせの性質または何らかの命令を含む。複製が選定されたとすると、ステップ４１２においてＳＡＮは、任意数の書き込みビッドを要求し、デジタルオブジェクトを書き込むためのクラスタ内のノードからその書き込みビッドを受け取る。

【００３９】

ＳＡＮのビッドが最低であれば、クライアントアプリケーションに「継続する」というメッセージを送り返して応答する。それに応じてクライアントはデジタルオブジェクトをＳＡＮに送信し、ＳＡＮは、デジタルオブジェクトを格納し、一意識別子を計算し、この識別子をクライアントアプリケーションに返す。一方、ＳＡＮがビッドを失った場合、ＳＡＮは最低ビッドを有するノードにクライアントアプリケーションをリダイレクトし、このノードが要求を取り扱う。するとクライアントアプリケーションは、同じ書き込み要求をこのノードに送信し、ノードは、「継続する」というメッセージをクライアントアプリケーションに送り返して応答する。これに応じてクライアントは、デジタルオブジェクトをノードに送信し、ノードはデジタルオブジェクトを格納し、一意識別子を計算し、この識別子をクライアントアプリケーションに返す。書き込みビッドの計算は、特許明細書第１２／８３９，０７１号、発明の名称「Ａｄａｐｔｉｖｅ　Ｐｏｗｅｒ　Ｃｏｎｓｅｒｖａｔｉｏｎ」に記載されているように実施でき、この明細書を参照することにより本願に組み込む。このようにする代わりに、デジタルオブジェクトを、前述した発明の名称が「Ｔｗｏ　Ｌｅｖｅｌ　Ａｄｄｒｅｓｓｉｎｇ　ｉｎ　Ｓｔｏｒａｇｅ　Ｃｌｕｓｔｅｒｓ」である明細書に記載のように書き込んでもよい。オブジェクトが書き込まれた後は、直ちに様々なノードに必要なだけ複製されてもよいし、クラスタが定期的な整合性チェックを待ってオブジェクトを複製してもよい。

【００４０】

一方、クラスタ内のオブジェクトを格納するのに消失符号付加が選定された場合、ステップ４１６でＳＡＮは、ストレージクラスタ内の全ノードから受けた書き込みビッドの要求を発行する。ＳＡＮが、用いるべき特定の消失符号付加（Ｋ：Ｍ、クライアントの命令、オブジェクトメタデータ、またはクラスタシステムの設定や定数に基づく）を決定すると、デジタルオブジェクトのデータおよびパリティセグメントを格納するのに用いるべきＭ個のノードを選択する。好ましくは、ＳＡＮは、最低コストでのノードのビッディングを選定するが、パフォーマンスを最良にしたり、消費電力を最低にしたり、他の基準を採用するなどの他の技術を用いてもよい。

【００４１】

１つの実施形態では、リスク軽減のために、物理的に離れている様々なサブクラスタ内でノードを選定でき、１つのサブクラスタが失われれば、全体のオブジェクトを残りのサブクラスタ内のセグメントから再生できるようにする。例えば、所与の３つのサブクラス

タがあるとすると、４：６符号化で符号化されたオブジェクトは、３つのサブクラスタの各々に２つのセグメントが格納されるように分配されたセグメントを有する。いずれか１つのサブクラスタが失われることで４つのセグメントが残り、これはオブジェクトを再構築するのに十分なものである。異なる数のサブクラスタに対して同様のスキームが可能である。

【００４２】

次に、ステップ４２０で、オブジェクトの第１のデータセグメントを格納する第１のノードが指定され、このノードは、様々なノード上のデータおよびパリティセグメントの中でオブジェクトを消失符号付加する準備をするためのいくつかのステップを実施する。例えば、ＳＡＮは、消失セット内のすべてのデータおよびパリティセグメントに対して（例えば乱数生成器を用いて）一意識別子を選定し、チャンク形式で符号化したＰＯＳＴＳをＭ個のノードの受信先に対して設定し、書き込み開始時に最大セグメントサイズを決定し、これによってこの消失セットのサイズを制限する。最大セグメントサイズは、ストレージクラスタの設定を参照して決定されてよい。各ノードは、ＳＡＮへの応答時にそのセグメントに対するボリュームの情報を返す。

【００４３】

次に、ステップ４２４でクライアントアプリケーションは、データオブジェクトをＳＡＮに転送し始め、ＳＡＮは、ストライプにある様々なノードの各データセグメントにデータを書き込み、適切な消失符号付加アルゴリズムを用いてパリティセグメントに対してデータを計算する。１つの実施形態では、良好に動作するためのＺｆｅｃアルゴリズムを発見した。例えば、図２を参照し、５：７符号化を仮定すると、受信したオブジェクトの第１の３２ｋのブロックは、第１のノード（ＳＡＮ）でデータセグメント２１０に書き込まれ、第２の３２ｋのブロックは、第２のノードでデータセグメント２１２に書き込まれ、というふうになる。第５のデータブロックが第５のノードに書き込まれた後、２つのパリティブロックが計算され、クラスタの選択された最後の２つのノードにあるセグメント２２０および２２２に格納される。デジタルオブジェクトは、引き続きクライアントアプリケーションから読み出されるとともに、ストレージクラスタの１つ１つのストライプごとにＭ個の選択されたノードに書き込まれる。これは、オブジェクトの末尾に達するかセグメントの末尾に達するまで続けられ、この動作はテスト４２８である。一杯になった書き込み対象のデータブロックがなければ、図示したように残りのストライプ２４０に書き込める。セグメントの末尾に達するか（ただし、オブジェクトに残っているデータは依然として書き込まれる必要がある）、オブジェクト全体がＭ個のセグメントに書き込まれると、制御は図４Ｂのステップ４３２に移る。

【００４４】

次に、ステップ４３２においてＳＡＮは、任意選択として、各セグメントのデータに基づいて各セグメントに対してハッシュ値を計算し、これらの値２７０（例えば）を選択したすべてのノードにトレーリングデータとして送信し、このノードがセグメントをそのボリュームに書き込んでいる。各ノードはＳＡＮから受け取ったこのハッシュ値を、自身のボリュームのうちの１つに格納されているセグメントについてディスク上で計算するハッシュ値と比較する。

【００４５】

次に、ステップ４３６では、消失符号付加のボリュームヒントが決定され、各セグメントに対して格納される。例えば、セグメント２１６が格納されているボリューム識別子は、セグメント２１４に対するシステムのメタデータの中に書き込まれ、このようにして、各セグメントが次のセグメントに対して、可能性のあるボリューム識別子をリング状に格納するようにする。好ましくは、ボリュームヒントは、各セグメントがＰＯＳＴ要求を用いて書き込まれる際にＳＡＮから転送される。このほか、ボリュームヒントは、各ボリュームのジャーナルの中に書き込まれてもよい。換言すれば、消失セットが任意のいくつかのノードに書き込まれると、所与のセグメントが格納されている特定のボリュームに対するジャーナルのエントリが更新されて、オブジェクトの次のセグメントが格納されている

ボリューム識別子を含める。

【００４６】

ステップ４４０では、ストレージクラスタに書き込むためにデジタルオブジェクトから来るデータがまだあるかどうかを判断する。ある場合は、ステップ４４４で新たな消失セットが開始される。この状況では、ＳＡＮは、クラスタ全体から書き込みビッドを要求して次のM個のノードを決定し、このノードが、そのボリュームに対するデータおよびパリティセグメントの書き込みを承諾する。すると、制御は、ステップ４１６に移って、このデジタルオブジェクトに対して次の消失セットを書き込む。

【００４７】

書き込むデータがこれ以上ない場合、ステップ４４８において、この消失セットに対するマニフェストが書き込まれる。図３に示したように、マニフェストは、メタデータセクションおよび消失セットの各々に対するセクションを含む。例えば、一意識別子（ハッシュ値、乱数など）が各消失セットの各セグメントに対して計算され、セクション３５０に格納される。この一意識別子は、ステップ４１６またはその後のステップ（乱数の場合）で計算されるか、あるいはステップ４２８またはその後のステップ（ハッシュ値の場合）で計算されてよい。マニフェストは、クラスタのいずれかのノードに書き込まれ、ストレージクラスタに書き込まれた任意のストリームのように処理される。換言すれば、一意識別子がマニフェストに対して計算され、マニフェストは、クラスタ内の様々なノードに対して複製される。好ましくは、マニフェストは、クラスタ内で合計Ｐ＋１回複製される。最後に、マニフェストに対する一意識別子はクライアントアプリケーションに返され、その結果クライアントアプリケーションは、後に読み出し操作を実施する際にそのデジタルオブジェクトにアクセスできる。

・クラスタからのデジタルオブジェクトの読み出し

【００４８】

図５は、クライアントアプリケーションがどのようにストレージクラスタからデジタルオブジェクトを読み出すのかを示すフローチャートである。有利には、クライアントアプリケーションは、ストレージクラスタがデジタルオブジェクトを格納するのに用いている技術がどちらであるか（複製か消失符号付加か）を認識している必要はない。単純に（以前にストレージクラスタによって提供されていた）デジタルオブジェクトに対して一意識別子を用いることによって、クライアントアプリケーションは、任意の外部データベースまたは制御システムに頼る必要なく、クラスタからオブジェクトを回収できる。実際、クライアントアプリケーションは、一意識別子がクラスタ内で複製されたオブジェクトを表しているのか、クラスタ内のオブジェクトを格納するために消失符号付加が用いられたことを示しているマニフェストを表しているのかを認識することはない。

【００４９】

ステップ５０４では、クライアントアプリケーションは、一意識別子に識別された特定のデジタルオブジェクトを返すためのストレージクラスタの要求を作成できる。これは、ＳＣＳＰ　ＧＥＴ要求または同様のＨＴＴＰプロトコルを用いて実装できる。クライアントアプリケーションは、ストレージクラスタの任意のノード（これがプライマリアクセスノードになる）に識別子を供給する。次に、ステップ５０８においてＰＡＮは、その特定の一意識別子を有するオブジェクトを探しているクラスタ内の全ノードにメッセージをブロードキャストする。この時点で、一意識別子が実際のオブジェクトを表しているのかマニフェストを表しているのかということもＰＡＮにとっては明白である。

【００５０】

実際のオブジェクトもマニフェストも両方ストレージクラスタ内で複製されるため、ＰＡＮは、そのブロードキャスト要求に対していくつかの応答を受け取る。１つの実施形態では、オブジェクトのコピー（またはマニフェストのコピー）を有する各ノードは、読み出しビッド（デジタルオブジェクトを読み出すためのコスト）を計算し、ＰＡＮは、読み出しビッドが最低であるノードを選定し、クライアントアプリケーションをそのノードにリダイレクトし、その後そのノードはセカンダリアクセスノード（ＳＡＮ）になる。ＳＡ

Ｎは、発見されたオブジェクト（実際のオブジェクトかオブジェクトのマニフェストかのいずれか）のシステムメタデータを見ることによって、複製が用いられたのか消失符号付加が用いられたのかを認識する。

【００５１】

したがって、ステップ５１０でＳＡＮは、（複製が用いられたために）実際のデジタルオブジェクトを保有しているのか、（消失符号付加が用いられたために）実際のデジタルオブジェクトに対するマニフェストを保有しているのかを判断する。複製が用いられた場合、ステップ５１２でＳＡＮは、単純に要求しているクライアントアプリケーションにデジタルオブジェクトを返し、その方法は終了する。その代わりに、消失符号付加が用いられ、ＳＡＮがマニフェストを保有している場合、ステップ５１６でＳＡＮは、クラスタのセグメントを要求するプロセスを開始して、要求されたデジタルオブジェクトを再集結する。マニフェストのメタデータを用いて、ＳＡＮは、用いられた消失符号付加アルゴリズムおよび特定の消失符号付加（例えば５：７）を認識する。好ましくは、デジタルオブジェクトを再結集するのに必要なのは第１のＫ個のデータセグメントのみであるため、ＳＡＮは、マニフェストのセクション３５０で発見された一意識別子を用いて、第１のＫ個のデータセグメントのみに対する要求をブロードキャストする。これが成功すれば、制御はステップ５２８に移る。

【００５２】

しかしながら、この第１のＫ個のデータセグメントのうちのいずれかが喪失している場合（ステップ５２０）、必要とされている任意のパリティセグメントに対して要求がブロードキャストされる。例えば、元のデータセグメントのうちの２つが喪失している場合、マニフェストからの一意識別子を用いて、パリティセグメントのうちの２つに対して要求をブロードキャストしなければならない。必要数のパリティセグメントが発見された場合、ステップ５２４で、喪失データセグメント（または複数のセグメント）は、適切な消失符号付加アルゴリズムおよび発見されたパリティセグメントを用いて再生される。１つの実施形態では、喪失セグメントのハッシュ値は、計算されて元のハッシュ値と比較されてよい。あるいは、セグメント１から（Ｋ－１）までのブロックを備える入力値として生成されたブロックを用いることによって各ストライプに対するデータを確認し、ブロックＫを生成してそのブロックを元のブロックと比較することも可能である。しかし、Ｋ個のセグメントを発見できない場合は、クライアントアプリケーションにエラーメッセージが返される。

【００５３】

Ｋ個のセグメントが発見されるか生成されると仮定すると、ステップ５２８においてＳＡＮは、（マニフェストを用いて）得られる別の消失セットがあるかどうか判断する。ない場合、制御はステップ５３２に移る。ある場合、制御はステップ５１６に移り、ＳＡＮは、マニフェストの第２の消失セットに対する対応するセクションで発見された一意識別子を用いて、第２の消失セットの必要なセグメントを要求するプロセスを開始する。ステップ５３２および５３６においてＳＡＮは、それが回収した各消失セットに対してデータおよび／またはパリティセグメントを結集して、元のデータオブジェクトを再構築する。例えば、必要なＫ個のセグメントがクラスタ内のノードで発見された時、ＳＡＮは、これらのセグメントからメモリにデータを読み出し、適切な消失符号付加アルゴリズムに適用して、ＳＡＮで元のデジタルオブジェクトを再構築する。好ましくは、オブジェクトの各ストライプは回収されるか再構築されるため、このデータは、ＨＴＴＰを介してバイトごとにクライアントアプリケーションにフィードバックされる。消失セットが２つ以上あるとすると、ＳＡＮは、次の消失セットを用いてデジタルオブジェクトの次の部分を再構築し、次の消失セットのバイトをクライアントアプリケーションにフィードバックする。その代わりに、ＳＡＮは、それ自体のメモリ内のオブジェクト全体を集結してからオブジェクトをクライアントに送り返してもよい。

・ボリューム障害からの回復

【００５４】

図６は、ストレージクラスタがボリューム障害からどのように回復できるのかを示すフローチャートである。前述したように、ストレージクラスタは、任意数のコンピュータノードを含み、各ノードは、任意数のハードディスクまたはボリュームと呼ばれるソリッドステートディスクを有する。ストレージクラスタは通常、オブジェクトの様々なレプリカを様々なノードに格納し（複製が用いられた場合）、オブジェクトの多様なデータおよびパリティセグメントを様々なノードに格納することで（消失符号付加が用いられた場合）、データ冗長性に達する。その結果、ノードのディスクに障害が発生すると、任意数のオブジェクトの多くのレプリカおよびセグメントが損失し、それによってストレージクラスタ全体のデータ冗長性といわれるものが劣化する。さらに、ストレージクラスタの質は、どれだけのボリュームを損失できる余地があるのかというだけでなく、ボリュームに障害が発生した際にクラスタがどれだけ速く喪失データを回復できるのかということによって判断される。したがって、図６は複製（レプリカ）および消失符号付加（セグメント）を用いて格納されたオブジェクトの混合物が存在する場合に障害が発生したボリュームの回復を扱うだけでなく、喪失セグメントの回復もできるだけ迅速に回復する技術を示している。ノード全体に障害が発生すれば、障害が発生したノードの各ボリュームに対して以下の技術が実施される。

【００５５】

ステップ６０４では、クラスタのノードは、そのディスクのうちの１つに障害が発生したことを認識する。１つの事例では、ノードが正常な業務過程でボリュームと通信すると、そのノードは、ボリュームから定期的に連絡を受けるよう待機する。連絡がない場合、ノードは、喪失ボリュームを探しているそのすべてのボリュームにメッセージをブロードキャストする。応答がない場合、ノードは、ボリュームに障害が発生したと仮定する。別の事例では、ストレージクラスタは、メンテナンスやクラスタの移動などの理由で全体的にシャットダウンされることがある。クラスタがバックアップされた場合、ボリュームに障害が発生する可能性があるが、ボリュームから事前に連絡がないため、ノードはそれを認識しないおそれがある。この状況では、ノードのモジュールのヘルス処理が補助できる。ヘルス処理モジュールは、各ボリュームで全ストリームの整合性を定期的にチェックし、特定のストリームのボリュームヒント（ボリュームの一意識別子）を検知した際に、そのボリュームを探す。発見されなければ、ノードは、ボリュームに障害が発生したと再度仮定する。ボリュームに障害が発生したことを検知する他の技術を用いてもよい。

【００５６】

次に、ステップ６０８では、障害が発生したボリュームの一意識別子が取得されると、クラスタ内の各ノードは、機能しているボリュームすべてをスキャンするように誘導されて、喪失ボリュームに対するボリュームヒントを含むストリームを識別する。好ましくは、喪失ボリュームを識別したノードは、ブロードキャストメッセージ（ボリューム識別子を含む）を他の全ノードに送信し、喪失ボリュームに対するヒントを有するストリームを検索するよう要求する。また、ノードは、効率のために平行して検索を実施する。１つの実施形態では、各ボリュームがディスクに記録したジャーナルがスキャンされ、各ストリームの表現が分析されて、そこに含まれているボリュームヒントを判断する。消失符号付加したオブジェクトのセグメントを表しているジャーナル内の各ストリーム表現は、次のセグメントに対するボリューム識別子を含んでいるため、喪失ボリューム対するボリューム識別子を含んでいるそのように識別されたストリームはいずれも、喪失ボリュームにあったセグメントも指している。例えば、図２のセグメント２２２のストリーム表現がボリュームヒントを含んでいて、そのヒントが喪失ボリュームに対するボリューム識別子である場合、それはつまり、セグメント２１０がそのボリュームにあって再生される必要があるということである。このほか、ジャーナル内の複製されたストリームの表現は、そのストリームのすべてのレプリカに対するボリューム識別子を指すボリュームヒントを含む。各ストリームに対して、ジャーナルは通常、ストリームが複製されたオブジェクトを表しているのか消失符号付加したオブジェクトを表しているのかを示す種類の情報を含んでいる。

【００５７】
もう１つの実施形態では、これらのボリュームヒントは、セグメントのシステムメタデータ２８０（または複製されたストリームのメタデータ）に格納できる。例えば、セグメント２１６に対するシステムメタデータは、セグメント２１８が格納されているボリューム識別子を指すボリュームヒントを含んでいる。各ノードがそのボリュームをスキャンしてディスク上の各ストリームのシステムメタデータを探せる可能性があるが、この技術では遅くなってしまう。この場合、ボリュームヒントはこのシステムメタデータから読み出せる。この場合もまた、障害が発生したボリュームを指す特定セグメントにあるボリュームヒントは、次のセグメントが喪失していることを指している。セグメントが喪失していることをノードが識別すると、そのノードは、前のセグメントのメタデータ２８０を見てすべてのシブリングセグメントに対する一意識別子を回収することによって、その喪失セグメントに対する一意識別子を判断できる。喪失セグメントを再生するのに必要な任意のセグメントを発見するために、シブリングセグメントのこれらの一意識別子を使用してよい。

【００５８】
各ノードがそのジャーナル（またはディスク上のストリーム）をスキャンし終えると、各ノードは、障害が発生したボリュームにあった喪失ストリームのリストを有する。ストレージクラスタは、複製および消失符号付加を用いて格納されたオブジェクトを含んでいるため、これらの喪失ストリームのいくつかは、複製されたオブジェクトを表し、ストリームのいくつかは、消失符号付加したオブジェクトの喪失セグメントを表している。

【００５９】
複製されたストリーム（ある場合）が喪失している場合、ステップ６１２において各ノードは、他のノードからビッドを要求することによって喪失ストリームを複製してストリームを複製し、その後、選定したノードに制御を転送する。消失符号付加したオブジェクトの少なくとも１つのセグメントが所与のノードから喪失していると仮定すると、ステップ６１６では（ステップ６０８で得られた一意識別子を用いて）、所与のノードが、喪失セグメントを再生するのに必要なＫ個のセグメントを供給できるノードにビッドするように他のノードに要求する。勝利するビッドが受け取られ、ノードがＫ個のセグメントを供給できると識別されると、所与のノードが１つのノードに喪失セグメントを再生して格納するよう要求する。

【００６０】
したがって、ステップ６２０では、所与のノードは、特定のノードが喪失セグメントを再生するように、クラスタ内のノードからのビッドを要求する。このノードが選定されると、そのノードは、ステップ６１６において識別されたＫ個のセグメントを用いて喪失セグメントを再生する。この再生は、適切な消失符号付加アルゴリズムを用いて実施されてもよい。１つの実施形態では、喪失セグメントの再生で、Ｋ個のセグメントから得たストライプのデータが消費され、目的のストライプは、喪失セグメントを再生するために計算され書き込まれる。

【００６１】
ステップ６２４においてノードは、そのボリュームのうちの１つにセグメントを格納する。所与のノードに識別された喪失セグメントがまだある場合、制御はステップ６１６に移り、前述したようにノードは再び喪失セグメントに対するビッドを要求する。クラスタ内の各ノードは、ステップ６０８においてそのボリュームをスキャンして喪失ストリームを探しているため、各ノードは、ステップ６１２から６２８を平行して実施していて、各ノードが喪失ストリームを識別したと仮定する。

・消失符号付加されたセグメントの再配置

【００６２】
セグメントがストレージクラスタに書き込まれ、一意識別子を提供されると、そのセグメントは、複製されたストリームを含むクラスタ内の他のストリームのように管理されてよい。換言すれば、ヘルス処理モジュールは、セグメントを１つのボリュームから別のボ

リュームへ移すのがよいか、あるいは１つのノードから別のノードへ移すのがよいかを、消失セット内の他のセグメントとは関係なく、かつ移されているセグメントの利用可能性を失うことなく、考慮することができる。例えば、セグメント２１８を別のボリュームに移すとすると、セグメント２１６内のボリュームヒントは更新されて、セグメント２１８に対する新たなボリュームを指す。セグメント２１８が再配置されると、システムは、そのシブリングセグメント（およびその一意識別子）をすべて認識する。なぜなら、セグメント２１８のメタデータ２８０は、すべてのシブリングセグメントの一意識別子を順に含んでいるからである。上流のセグメント、つまりセグメント２１６は、その一意識別子を用いてクラスタ内から回収でき、かつセグメント２１８に対する新たなボリューム識別子がわかると、この新たなボリューム識別子は、セグメント２１８の新たな場所に対するボリュームヒントとしてセグメント２１６のメタデータセクション２８０の中に書き込むことができる。その代わりに、そのジャーナルでのセグメント２１６のストリーム表現は、更新されて新たなボリューム識別子を含んでもよい。

【００６３】

再配置されたセグメントに対するボリュームヒントのこの更新は、再配置が行われる際に実施されてもよいし、ヘルス処理モジュールによって後に実施されてもよい。この更新の利点は、セグメントをクラスタ内に再配置でき、利用可能性の損失がなく、クラスタ内のセグメントを追跡するための余分なコンピュータ制御やデータベース制御の必要がない点である。

・消失符号付加したオブジェクトを複製に変換、およびその逆

【００６４】

本発明の１つの実施形態では、クラスタ内に格納されたデジタルオブジェクトを１つの方式から別の方式へ変換できる。例えば、５：７の消失符号付加を用いて格納されたオブジェクトを６：１０の暗号化に変換でき、消失符号付加したオブジェクトを、複製を用いたストレージに変換でき、複製を用いて格納したオブジェクトを、消失符号付加を用いるストレージに変換できる。オブジェクトを別の方式に変換するかどうか、またそれをいつするかというのは、オブジェクトメタデータ、ストレージクラスタの初期設定、またはこの両方を合わせたものによって指定できる。

【００６５】

前述したように、クライアントアプリケーションからオブジェクトを提供されたユーザメタデータは、オブジェクトをどのように格納すべきかということに関する情報を提供できるとともに、直前に記載したように、オブジェクトを１つの方式から別の方式にいつ変換すべきかということも指定できる。例えば、ユーザメタデータは、特定の時間枠内に、あるいは先の特定の時間に、オブジェクトを別の方式に変換すべきであることを指定できる。あるいは、ストレージクラスタの設定および規定で、オブジェクトを１つまたは複数の特定の時間に変換すべきだということ、一定サイズのオブジェクトを定期的にまたは特定の時間に変換すべきだということ、あるいは一定割合のオブジェクトを変換すべきだということも指定できる。クラスタは、クラスタの設定を変更したり、クラスタ内の１つまたは複数のオブジェクトに対していつどのように変換を起こすべきかを指定したり、といった管理者からの手動入力を受けることもできる。変換を実施するために特別な変換モジュールを使用してもよいし、あるいはそのような機能性をクラスタのヘルス処理モジュールに組み入れてもよい。

【００６６】

図７は、ストレージクラスタ内のオブジェクトを１つのストレージフォーマットから別のストレージフォーマットへ変換できるという１つの実施形態を示すフローチャートである。この図では単一のオブジェクトを扱っているが、この技術を用いてクラスタ内の任意数のオブジェクトを変換してよい。クラスタ内のオブジェクトに対する一意識別子は、クライアントアプリケーションが元々備えていた同じ一意識別子を用いてオブジェクトを回収できるように、同じままである。乱数を一意識別子として有するオブジェクトの場合、この乱数は、変換後もオブジェクトに対する識別子のままである。クライアントアプリケ

ーションがオブジェクトに一意的な名称を提供する場合、クラスタは、その名称のハッシュ値を一意識別子として使用でき、このハッシュ値は、変換後も識別子のままである。

【００６７】

オブジェクトのバージョンの概念により、オブジェクトを変換した新たな一意識別子に対して元の一意識別子を保持しやすくなる。各オブジェクトは、それがいつ作成されたのか、オブジェクトがいつ変換されるか、それは元のオブジェクトと同じ一意識別子を有するのか、新たなオブジェクトは元のオブジェクトよりも後にタイムスタンプを与えられたのか、を示すタイムスタンプを含んでいる。このように、両一意識別子は、クラスタ内に同時に存在していてよい。ただし、クラスタは、タイムスタンプを参照することによってどのオブジェクトが現在有効なオブジェクトなのかを認識する。ツインのオブジェクトよりも先にタイムスタンプを有しているオブジェクトは、必要ではないためいつ消去されてもよい。

【００６８】

ステップ７０４では、１つのオブジェクトに対して（または任意数のオブジェクトに対して）関連する変換情報がストレージクラスタ内に格納される。前述したように、各オブジェクトは、オブジェクトをどのように変換すべきか、オブジェクトをいつ変換すべきかなどを指定するユーザメタデータを含むクライアントアプリケーションから受け取られてよい。このユーザメタデータは、オブジェクトがクラスタに書き込まれる際に各オブジェクトと一緒に格納される。このメタデータは、消失符号付加したオブジェクトのマニフェストに格納される。

【００６９】

また、ストレージクラスタの設定および規定は、クラスタ内のオブジェクトに対する初期変換設定を指定できる。これらの設定および規定は、指定されたクラスタのオブジェクトに格納されてもよいし、起動中にノードを備えている情報に含まれてもよいし、ノードが動作しているネットワーク上で指定されたソースによって提供されてもよい。例えば、設定が、全オブジェクトを特定の日付ごとに１つの消失符号付加方式から別の消失符号付加方式へ変換することを必要としてもよいし、オブジェクトが一定の年月に達すると、ある期間にわたって全オブジェクトに対して複製から消失符号付加に変換することを必要としてもよいし、一定サイズを超えるオブジェクトが、特定の年月または日付ごとに（またはある期間にわたって徐々に）消失符号付加に変換することなどを必要としてもよい。さらに、管理者が設定またはコマンドを入力して、１つまたは複数のオブジェクトまたはオブジェクトにいつどのように変換を起こすべきかを指定してもよい。

【００７０】

ステップ７０８では、オブジェクトを１つの方式から別の方式へ変換すべきであることを指摘する特定のオブジェクトに対してトリガー条件を検知する。このトリガー条件は多くの様々な方法で検知できる。例えば、クラスタ内のオブジェクトに対して反復過程にあるヘルス処理モジュールは、オブジェクトに触れたときにその特定のオブジェクトに対するオブジェクトメタデータを見直す。条件（例えば「この特定の消失符号付加方式を用いて特定の日付ごとに、または特定の日付で消失符号付加に変換する」）が合えば、オブジェクトは、後述するように変換される。あるいは、何らかの理由でオブジェクトが触れられるかアクセスされた時はいつでも、そのユーザメタデータは、トリガー条件が満たされているかどうかを見るために見直される。また、ストレージクラスタ自体は、定期的にそのクラスタの設定および規定を見直して、時間または日付が過ぎたかどうかを判断して、オブジェクトまたはオブジェクトクラスタの設定に応じて変換すべきであることを指摘する。もちろん、クラスタの管理者からの何らかの手動入力が即座に作用し、トリガー条件を示してもよい。

【００７１】

ステップ７１２は、（複製を用いて現在格納されている）オブジェクトを消失符号付加に変換すべきであるとトリガー条件が示したときの結果である。オブジェクトに対する一意識別子が（オブジェクトメタデータ、クラスタの設定、管理者の入力などから）得られ

、クラスタは、オブジェクトのレプリカが存在しているノードを判断する。ステップ７１６では、このノードは、そのディスクのうちの１つからメモリへオブジェクトを読み出す。次に、ステップ７２０でノードは、ユーザメタデータ、システムメタデータ、クラスタの設定、または管理者の入力から判断された特定の消失符号付加方式を用いて、オブジェクトをクラスタに書き込む。このステップは、図４Ａおよび図４Ｂ、具体的にはステップ４１６～４４８を参照して上記に考察したように実施されてよい。消失符号付加を用いて書き込まれたこの新たなオブジェクトには、そのマニフェストに対する一意識別子が供給され、この一意識別子は、ステップ７１６で読み出された元のレプリカに用いられた一意識別子と同じである。ステップ７２４では、元のオブジェクトおよび任意のレプリカが、即座にまたは後にヘルス処理モジュールによって消去されてよい。ヘルス処理モジュールは、新たに変換されたオブジェクトよりもタイムスタンプが早い一意識別子を有する任意のレプリカを消去してよいことを決定する。

【００７２】

ステップ７３２は、（消失符号付加を用いて現在格納されている）オブジェクトを、複製を用いたストレージに変換すべきであるとトリガー条件が示したときの結果である。オブジェクトに対する一意識別子が（オブジェクトメタデータ、クラスタの設定、管理者の入力などから）得られ、クラスタは、消失符号付加したオブジェクトに対するマニフェストが存在しているノードを判断する。ステップ７３６では、このノードは、クラスタからメモリへオブジェクトを読み出す。このステップは、図５、具体的にはステップ５１６～５３２を参照して上記に考察したように実施されてよい。次に、ステップ７４０でこのノードは、オブジェクトを連続的なストリームとして（消失符号付加したセグメントとしてではなく）クラスタのノードに書き込む。この書き込みは、例えば、クラスタ全体への書き込みビッドに対する要求をブロードキャストし、その後ビッドが勝っているノードにストリームを書き込むことによって実施できる。あるいは、他の技術を用いてオブジェクトを書き込むための特定のノードまたはディスクを選定してもよい。単一のストリームとして書き込まれたこの新たなオブジェクトには、ステップ７３６で読み出された元のマニフェストに用いられた一意識別子と同じ一意識別子が供給される。ステップ７４４では、元のオブジェクトは、即座にまたは後にヘルス処理モジュールによって消去されてよい。ヘルス処理モジュールは、新たに変換されたオブジェクトよりもタイムスタンプが早い一意識別子を有する任意のマニフェスト（およびそれに関連するセグメント）を消去してよいことを決定する。ステップ７４８では、新たに書き込まれたオブジェクトは、複製されたクラスタ内に任意数のレプリカを作成でき、この複製は、即座に起きてもよいし、ヘルス処理モジュールがこのオブジェクトに対して反復している時間に起きてもよい。

【００７３】

ステップ７５２は、（古い消失符号付加を用いて現在格納されている）オブジェクトを、新たな消失符号付加に変換すべきであるとトリガー条件が示したときの結果である。オブジェクトに対する一意識別子は、オブジェクトメタデータから得られ、クラスタは、消失符号付加したオブジェクトに対するマニフェストが存在しているノードを判断する。ステップ７５６では、このノードは、クラスタからメモリへオブジェクトを読み出す。このステップは、図５、具体的にはステップ５１６～５３２を参照して上記に考察したように実施されてよい。次に、ステップ７６０においてノードは、ユーザメタデータ、システムメタデータ、クラスタの設定、または管理者の入力から判断された新たな消失符号付加方式（フォーマット）を用いて、オブジェクトをクラスタに書き込む。このステップは、図４Ａおよび図４Ｂ、具体的にはステップ４１６から４４８を参照して上記に考察したように実施されてよい。新たな消失符号付加を用いて書き込まれたこのオブジェクトには、そのマニフェストに対する一意識別子が供給され、この一意識別子は、ステップ７５６で読み出された元のオブジェクトに用いられた一意識別子と同じである。ステップ７６４では、元のオブジェクトは、即座にまたは後にヘルス処理モジュールによって消去されてよい。ヘルス処理モジュールは、新たに変換されたオブジェクトよりもタイムスタンプが早い一意識別子を有する任意のマニフェストを消去してよいことを決定する。

・クラスタ全体のデジタルオブジェクトの管理
【００７４】
本発明のもう１つの実施形態では、１つのストレージクラスタから別のストレージクラスタにデジタルオブジェクトを移せるとともに、オブジェクトを新たなクラスタが求める方式に変換するか、オブジェクトのユーザメタデータに求められる方式に変換することができる。例えば、５：７の消失符号付加を用いて第１のクラスタに格納されたオブジェクトを、第２のクラスタに移る際に６：１０の暗号化に変換してもよいし、消失符号付加したオブジェクトを、第２のクラスタに移った際に複製を用いてストレージに変換してもよいし、複製を用いて第１のクラスタに格納したオブジェクトを、移った際に消失符号付加を用いてストレージに変換してもよい。オブジェクトを異なる方式に変換するかどうかは、ユーザメタデータ、ストレージクラスタの初期設定、この両方を合わせたもの、外部ソフトウェア製品からの命令、またはクラスタ管理者によって指定されてよい。有利には、第１のクラスタにあるオブジェクトの一意識別子は、第２のクラスタ内で使用されるためにも保有される。
【００７５】
図８は、１つのストレージクラスタ内のオブジェクトを第２のストレージクラスタに移して、１つのストレージフォーマットから別のストレージフォーマットに変換できるという１つの実施形態を示すフローチャートである。この図では単一のオブジェクトを扱っているが、この技術を用いてクラスタ内の任意数のオブジェクトを移してよい。好ましくは、第１のクラスタ内のオブジェクトに対する一意識別子は、クライアントアプリケーションが同じ一意識別子を用いて第２のクラスタからオブジェクトを回収できるように、同じままである。例えば、一意識別子は、クライアントアプリケーションによって供給されたオブジェクトの名称の乱数またはハッシュ値であってよい。
【００７６】
ステップ８０４では、ソースクラスタからターゲットクラスタにオブジェクトをコピーする（移す）ための命令が生成される。オブジェクトは、バックアップ目的で（元のオブジェクトをソースクラスタに残して）ターゲットクラスタにコピーされてもよいし、オブジェクトは、単にターゲットクラスタに移されて元のオブジェクトが消去されてもよい。命令は、任意の外部ソフトウェア製品、クライアントアプリケーションからのものであってもよいし、クラスタ自体の中からのものであってもよいし、クラスタ管理者からのものであってもよい。１つの実施形態では、Ｃａｒｉｎｇｏ社から市販されている「Ｃｏｎｔｅｎｔ　Ｒｏｕｔｅｒ」ソフトウェア製品を使用して、ソースクラスタからターゲットクラスタにオブジェクトを複製するための命令を生成する。命令は、ソースクラスタ内にある一意識別子など、複製されるオブジェクトの識別を含んでいる。
【００７７】
オブジェクトは、多くの様々な方法でソースクラスタからコピーされてよい。例えば、ソースクラスタは、オブジェクトを読み出した後にそれをターゲットクラスタに「押し出し」てもよいし、あるいは、ターゲットクラスタがオブジェクトをソースクラスタから「押し出し」てもよい。１つの実施形態では、ステップ８０８で、ターゲットノードがまずターゲットクラスタで選択されてターゲットクラスタ内へのオブジェクトの書き込みを実施する。ターゲットノードは、ビッドプロセス、またはその他の技術を用いてランダムに選択されてよい。選択されると、ターゲットノードは、コピーされるオブジェクトに対する一意識別子を提供され、ソースクラスタに対する情報に接触する。例えば、ターゲットノードは、ソースクラスタ全体に対する通信アドレスや、クラスタ内の中央ノードまたは調整ノードのアドレスや、好ましくは、ソースクラスタ内の任意のノードのＩＰアドレスを提供されてよい。
【００７８】
ステップ８１２では、ターゲットクラスタ内で任意の関連する変換情報が識別される。例えば、コピーされたオブジェクトをどのようにターゲットクラスタ内に格納すべきか（すなわち複製または消失符号付加を用いて）を指定する任意の初期設定または規定が識別

される。関連する初期設定がなければ、変換情報はコピーされるオブジェクト内に含まれるユーザメタデータから取得されてよい。その代わりに、オブジェクトをコピーする命令は、変換情報を含んでいてよい。

【００７９】

ステップ８１６においてオブジェクトは、ソースクラスタからターゲットクラスタにコピーされる。ターゲットノードは、提供されたＩＰアドレスを用いてソースクラスタのいずれかのノードに接触することによってオブジェクトのコピーを開始し、オブジェクトに対する一意識別子を提供する。するとオブジェクトは、ソースクラスタからターゲットノードのメモリに通信されてよい。このステップは、例えば、図５を参照して上記に説明したように実施されてよく、この場合ターゲットノードは、クライアントアプリケーションとして作用する。ターゲットノードがオブジェクトを受け取ると（あるいはターゲットノードが受け取られた際にオブジェクトが）、ターゲットノードは、上記で決定された適切な変換情報を用いてターゲットクラスタにメモを書き込む。換言すれば、オブジェクトは、連続ストリーム（複製）として書き込まれるか、消失符号付加を用いて書き込まれる。例えば、この書き込みステップは、図４Ａおよび図４Ｂを参照して上記に記載したように実施されてよく、この場合ターゲットノードは、セカンダリアクセスノードとして作用する。複製の場合、ターゲットノードは、ターゲットクラスタ内の他のノードからビッドを求めてもよいし、またはそれ自体のディスクのうちの１つにオブジェクトを書き込んでもよい。消失符号付加の場合、セグメントは、ターゲットクラスタ内の様々なノードに書き込まれる。好ましくは、ターゲットクラスタ内のコピーされたオブジェクトは、ソースクラスタ内にあったものと同じ一意識別子を保有している。オブジェクトがターゲットクラスタにコピーされると、オブジェクトはソースクラスタ内で保有されてもよいし、その後に消去されてよい。

・コンピュータシステムの実施形態

【００８０】

図９Ａおよび図９Ｂは、本発明の実施形態を実装するのに適したコンピュータシステム９００を示している。図９Ａは、コンピュータシステムの１つの可能な物理的形態の例を示している。もちろん、コンピュータシステムは、例としては、集積回路、プリント基板、（携帯電話やＰＤＡなどの）小型携帯用デバイス、パーソナルコンピュータまたはスーパーコンピュータなど、多くの物理的形態をとることができる。コンピュータシステム９００は、モニタ９０２、ディスプレイ９０４、筐体９０６、ディスクドライブ９０８、キーボード９１０およびマウス９１２を備える。ディスク９１４は、コンピュータシステム９００との間のデータ転送に用いられるコンピュータ可読媒体である。

【００８１】

図９Ｂは、コンピュータシステム９００のブロック図の一例を示している。システムバス９２０に連結されているのは、さまざまな種類のサブシステムである。（１つまたは複数の）プロセッサ９２２（中央処理装置、すなわちＣＰＵとも呼ばれる）は、メモリ９２４を含むストレージ装置に接続されている。メモリ９２４は、ランダムアクセスメモリ（ＲＡＭ）およびリードオンリメモリ（ＲＯＭ）を備えている。先行技術で公知のように、ＲＯＭは、ＣＰＵに対して一方向にデータおよび命令を転送するように作動し、ＲＡＭは通常、双方向にデータおよび命令を転送するのに用いられる。この類のメモリはいずれも、後述する任意の適切なコンピュータ可読媒体を備えていてよい。固定ディスク９２６も、ＣＰＵ９２２に双方向に接続され、付加的なデータストレージ容量を提供し、後述するコンピュータ可読媒体のいずれかを備えていてもよい。固定ディスク９２６は、プログラムやデータなどを格納するのに使用されてよく、通常、主要なストレージ装置よりも低速の二次ストレージ媒体（ハードディスクなど）である。固定ディスク９２６内に保持されている情報は、適切な場合には、標準的な方法で仮想メモリとしてメモリ９２４内に組み込まれてよいことは理解できるであろう。リムーバブルディスク９１４は、後述するコンピュータ可読媒体いずれの形態であってもよい。

【００８２】

ＣＰＵ９２２は、ディスプレイ９０４、キーボード９１０、マウス９１２およびスピーカー９３０などの様々な入出力装置にも接続されている。一般に、入出力装置は、ビデオディスプレイ、トラックボール、マウス、キーボード、マイクロフォン、タッチセンサ式ディスプレイ、トランスデューサカードリーダ、磁気または紙テープ読み出し装置、タブレット、スタイラス、音声または手書き認識装置、生体認証読み出し装置、または他のコンピュータのいずれかであってよい。任意選択として、ＣＰＵ９２２は、ネットワークインターフェース９４０を用いて、別のコンピュータネットワークまたは電気通信ネットワークに接続されてもよい。このようなネットワークインターフェースを用いると、ＣＰＵは、前述した方法のステップを実施する過程でネットワークから情報を受信したり、ネットワークに情報を出力したりすることができることが予想される。さらに、本発明の方法の実施形態は、ＣＰＵ９２２で単独で実行してもよいし、あるいは、インターネットなどのネットワークを介して実行し、リモートＣＰＵに処理の一部を分担させるようにしてもよい。

【００８３】

このほか、さらに、本発明の実施形態は、様々なコンピュータ実装動作を実施するためのコンピュータコードを有するコンピュータ可読媒体を備えるコンピュータストレージ製品に関する。媒体およびコンピュータコードは、本発明の目的のために特別に設計し構成されたものでもよいし、コンピュータソフトウェア分野の当業者に公知で利用可能な種類のものでもよい。コンピュータ可読媒体の例には、以下のものがあるがこれに限定されるものではない：ハードディスク、フロッピー（登録商標）ディスク、および磁気テープなどの磁気媒体；ＣＤ－ＲＯＭおよびホログラフィック装置などの光学媒体；フロプティカルディスクなどの光磁気媒体、ならびに、特定用途向けの集積回路（ＡＳＩＣ）、プログラム可能な論理装置（ＰＬＤ）ならびにＲＯＭおよびＲＡＭ装置などのプログラムコードを格納して実行するように特別に構成されたハードウェア装置。コンピュータコードの例には、コンパイラにより生成されるものなどのマシンコードや、インタープリタを用いるコンピュータによって実行される上位コードを含むファイルなどがある。

【００８４】

以上に記載した本発明について、明確に理解する目的で詳細に説明してきたが、添付の特許請求の範囲内でいくらかの変更および修正を加えてもよいことは明らかであろう。したがって、記載した実施形態は例示に過ぎず本発明を限定するものではない。本発明は、本明細書に記載した詳細に限定されるものではなく、以下の特許請求の範囲およびそれと同等の全範囲内で規定されるものである。

適用例１：ストレージクラスタにデジタルオブジェクトを格納する方法であって、

前記ストレージクラスタのコンピュータノードでクライアントアプリケーションから、前記デジタルオブジェクトを格納するための要求を受け取り、

前記デジタルオブジェクトを前記ストレージクラスタに、複製を用いて格納するか消失符号付加を用いて格納するかを判断し、

消失符号付加を用いて前記デジタルオブジェクトを格納すると判断された場合、消失符号付加を用いて前記デジタルオブジェクトを前記ストレージクラスタの複数のコンピュータノードに書き込み、前記デジタルオブジェクトは複数のセグメントとして格納され、

消失符号付加の表示と前記ストレージクラスタ内における各前記セグメントの一意識別子とを含むマニフェストコンピュータファイルを作成し、

前記ストレージクラスタのコンピュータノードに前記マニフェストコンピュータファイルを格納し、

前記マニフェストコンピュータファイルの一意識別子を前記クライアントアプリケーションに返すこと

を備える、方法。

適用例２：前記デジタルオブジェクトの固有の特性、前記クライアントアプリケーションからの命令、または前記デジタルオブジェクトのメタデータを参照して、複製または消失符号付加を用いて前記デジタルオブジェクトを格納するか否かを判断すること

をさらに備える、適用例１に記載の方法。
適用例３：前記ストレージクラスタ内に前記マニフェストコンピュータファイルを複製し、消失符号付加を用いて前記マニフェストコンピュータファイルを格納しないこと
をさらに備える、適用例１に記載の方法。
適用例４：前記デジタルオブジェクトを前記ストレージクラスタ内に複製しないこと
をさらに備える、適用例１に記載の方法。
適用例５：前記ストレージクラスタのディスクに格納されている各セグメントについて、前記セグメントに関連付けられている前記ディスクに、前記デジタルオブジェクトの別のセグメントを格納する次のディスクの一意識別子を格納すること
をさらに備える、適用例１に記載の方法。
適用例６：前記ディスクの前記セグメントについて前記一意識別子をジャーナルエントリに格納することによって、前記セグメントに関連付けられている前記次のディスクの前記一意識別子を格納すること
をさらに備える、適用例５に記載の方法。
適用例７：複数のコンピュータノードを有するストレージクラスタからデジタルオブジェクトを読み出す方法であって、
前記ストレージクラスタ内にある前記コンピュータノードのうちの１つにおいて、前記デジタルオブジェクトについての一意識別子を含むクライアントアプリケーションから要求を受け取り、
複製または消失符号付加を用いて前記ストレージクラスタ内に前記デジタルオブジェクトを格納するか否かを判断し、
消失符号付加を用いて前記デジタルオブジェクトを格納すると判断された場合に、前記コンピュータノードのうちの１つに格納されているマニフェストを読み出し、前記マニフェストは前記一意識別子によって識別され、
前記マニフェスト内で発見された一意的なセグメント識別子を用いて前記ストレージクラスタ内の複数のセグメントを識別し、
前記セグメントおよび消失符号付加アルゴリズムを用いて前記デジタルオブジェクトを再構築し、
前記デジタルオブジェクトを前記クライアントアプリケーションに返すこと
とを備える、方法。
適用例８：前記マニフェストを参照することにより、消失符号付加を用いて前記デジタルオブジェクトを格納することを判断すること
をさらに備える、適用例７に記載の方法。
適用例９：前記セグメントのうちの１つが前記ストレージクラスタ内に存在しないことを判断し、
他の前記セグメントおよび消失符号付加アルゴリズムを用いて、存在しない前記セグメントを再生すること
とをさらに備える、適用例７に記載の方法。
適用例１０：前記セグメントのうちの１つが格納されている第１のディスクを識別し、
もう１つの前記セグメントが格納されている第２のディスクについてのディスクの識別子を読み出し、前記ディスクの識別子は、前記第１のディスクにある前記セグメントのうちの前記１つと関連付けられて格納されること
とをさらに備える、適用例７に記載の方法。
適用例１１：前記マニフェスト内で第２の消失セットを識別し、前記第２の消失セットは複数の第２の一意的なセグメント識別子を含み、
前記セグメントを用いて前記デジタルオブジェクトを再構築し、複数の第２のセグメントは、前記第２の一意的なセグメント識別子、および前記消失符号付加アルゴリズムによって識別されること
とをさらに備える、適用例７に記載の方法。
適用例１２：前記マニフェストは、前記ストレージクラスタ内で複製され、前記マニフ

ェストは、消失符号付加を用いて前記ストレージクラスタ内には格納されない、適用例７に記載の方法。
適用例１３：前記デジタルオブジェクトは、前記ストレージクラスタ内で複製されない、適用例７に記載の方法。
適用例１４：障害のあるディスクから回復する方法であって、
複数のコンピュータノードを有するストレージクラスタ内で、前記ノードのうちの１つの第１のディスクに障害があることを検知し、
前記ストレージクラスタの第２のディスクの永続的なストレージ領域をスキャンして前記障害のあるディスクの一意識別子を発見し、前記一意識別子は前記ストレージクラスタのデジタルストリーム関連付けられており、
複製または消失符号付加を用いて、前記デジタルストリームを前記ストレージクラスタ内に格納するか否かを判断し、
消失符号付加を用いて前記デジタルストリームを格納すると判断された場合、前記障害のあるディスクに以前格納された喪失セグメントを識別し、
前記ストレージクラスタ内において複数の他のセグメントを見つけ出し、前記複数のセグメントは前記デジタルストリームを含み、
前記複数の他のセグメントおよび消失符号付加アルゴリズムを用いて、前記障害のあるディスクに以前格納された前記喪失セグメントを再生し、
前記再生されたセグメントを前記ストレージクラスタのコンピュータノードに格納すること
を備える、方法。
適用例１５：前記デジタルストリームのメタデータセクションをスキャンして、前記複数の他のセグメントに対する一意識別子を発見すること
をさらに備える、適用例１４に記載の方法。
適用例１６：前記第２のディスクの前記永続的なストレージ領域をスキャンすることは、前記デジタルストリームに対するジャーナルエントリをスキャンすることを含むこと
をさらに備える、適用例１４に記載の方法。
適用例１７：前記ジャーナルエントリを参照することにより、消失符号付加を用いて前記デジタルストリームを格納するか否かを判断すること
をさらに備える、適用例１６に記載の方法。
適用例１８：前記喪失セグメントを含む消失符号付加を用いて、ストレージクラスタの外部から前記ストレージクラスタ内に格納されているデジタルオブジェクトについての要求を受け取る前に、前記障害のあるディスクに以前格納された前記喪失セグメントを識別すること
をさら備える、適用例１４に記載の方法。
適用例１９：前記喪失セグメントは、前記ストレージクラスタ内では複製されない、適用例１４に記載の方法。
適用例２０：セグメントをストレージクラスタ内で再配置する方法であって、
コンピュータノードの第１のディスク上にあるセグメントをストレージクラスタ内で識別し、前記セグメントは、前記ストレージクラスタ内で格納されているデジタルオブジェクトを表す複数のセグメントのうちの１つであり、
前記ストレージクラスタの前記第１のディスクから第２のディスクへ前記セグメントを再配置し、前記第２のディスクは一意的なディスク識別子によって識別され、
前記セグメントのメタデータから前記複数のセグメントのシブリングセグメントについての一意識別子を回収し、前記シブリングセグメントは、そのメタデータ内に前記第１のディスクに対する一意的なディスク識別子を含み、
前記一意識別子を用いて前記シブリングセグメントを前記ストレージクラスタ内で見つけ出し、
前記シブリングセグメントの前記メタデータ内で、前記第１のディスクについての前記一意的なディスク識別子を前記第２のディスクについての一意的なディスク識別子に入れ

替え、前記シブリングセグメントの前記メタデータは、前記セグメントが再配置された先のディスクを示すこと
とをさらに備える、方法。
適用例２１：ストレージクラスタ内でデジタルオブジェクトを変換する方法であって、
前記ストレージクラスタ内の前記デジタルオブジェクトを、ビットの連続ストリームとしてコンピュータノードの単一のディスク上に格納し、前記デジタルオブジェクトは、前記ストレージクラスタ内で一意識別子を有し、
前記格納の後に、前記デジタルオブジェクトを消失符号付加ストレージ方式に変換する要求を示す前記ストレージクラスタのメタデータを識別し、
前記ストレージクラスタのコンピュータノードを用いて前記単一のディスクから前記デジタルオブジェクトを読み出し、
前記消失符号付加ストレージ方式を用いて、前記デジタルオブジェクトを前記ストレージクラスタの複数のディスクに書き込み、
前記消失符号付加ストレージ方式で書き込まれた前記デジタルオブジェクトについての前記一意識別子を保持し、これにより、クライアントアプリケーションは、前記一意識別子を用いて前記消失符号付加ストレージ方式で書き込まれた前記デジタルオブジェクトを検索可能であること
を備える、方法。
適用例２２：ストレージクラスタ内でデジタルオブジェクトを変換する方法であって、
前記ストレージクラスタ内の前記デジタルオブジェクトを第１の消失符号付加ストレージ方式で格納し、前記デジタルオブジェクトは、前記ストレージクラスタ内で一意識別子を有し
前記格納の後に、前記デジタルオブジェクトを第２の消失符号付加ストレージ方式に変換する要求を示す前記ストレージクラスタのメタデータを識別し、
前記ストレージクラスタのコンピュータノードを用いて前記ストレージクラスタから前記デジタルオブジェクトを読み出し、
前記第２の消失符号付加ストレージ方式を用いて、前記デジタルオブジェクトを前記ストレージクラスタに書き込み、
前記第２の消失符号付加ストレージ方式で書き込まれた前記デジタルオブジェクトについての前記一意識別子を保持し、クライアントアプリケーションは、前記一意識別子を用いて前記第２の消失符号付加ストレージ方式で書き込まれた前記デジタルオブジェクトを検索可能であること
を含備える、方法。
適用例２３：ストレージクラスタ内でデジタルオブジェクトを変換する方法であって、
前記ストレージクラスタ内の前記デジタルオブジェクトを消失符号付加ストレージ方式で格納し、前記デジタルオブジェクトが、前記ストレージクラスタ内で一意識別子を有し、
前記格納の後に、複製を用いて前記デジタルオブジェクトをストレージ方式に変換する要求を示す前記ストレージクラスタのメタデータを識別し、
前記ストレージクラスタのコンピュータノードを用いて前記ストレージクラスタから前記デジタルオブジェクトを読み出し、
前記ストレージクラスタのコンピュータノードの単一のディスクに前記デジタルオブジェクトをビットの連続ストリームとして書き込み、
前記ビットの連続ストリームとして書き込まれた前記デジタルオブジェクトについての前記一意識別子を保持し、クライアントアプリケーションは、前記一意識別子を用いて前記ビットの連続ストリームとして書き込まれた前記デジタルオブジェクトを検索可能であること
とを備える、方法。
適用例２４：ソースストレージクラスタからターゲットストレージクラスタにデジタルオブジェクトをコピーする方法であって、

前記ターゲットクラスタのターゲットノードにいて、前記ソースクラスタから前記ターゲットクラスタに前記デジタルオブジェクトをコピーするための命令を受け取り、前記命令は、前記ソースクラスタのソースノードのアドレスを含み、

前記デジタルオブジェクトを前記ターゲットクラスタに格納する先のターゲット変換方式を決定し、

前記ソースクラスタから前記デジタルオブジェクトを読み出し、前記デジタルオブジェクトは、ソース変換方式で格納され、前記ソースクラスタ内に一意識別子を有し、

前記ターゲット変換方式を用いて前記デジタルオブジェクトを前記ターゲットクラスタに格納し、前記一意識別子を用いて前記デジタルオブジェクトは前記一意識別子を用いて格納されること

を備える、方法。

適用例２５：前記ターゲット変換方式は、前記ソース変換方式とは異なる、適用例２４に記載の方法。

適用例２６：前記ソース変換方式は、複製または消失符号付加である、適用例２４に記載の方法。

適用例２７：前記デジタルオブジェクトのメタデータを参照するか前記ターゲットクラスタの設定を参照して、前記ターゲット変換方式を決定することを更に備える、適用例２４に記載の方法。

適用例２８：前記ターゲットクラスタは、前記ソースクラスタとは異なるストレージ方式を実装する、適用例２４に記載の方法。

【図１】

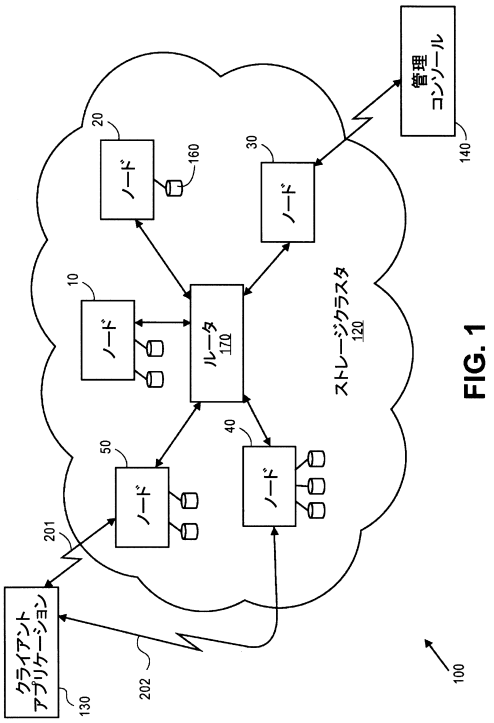

FIG. 1

【図２】

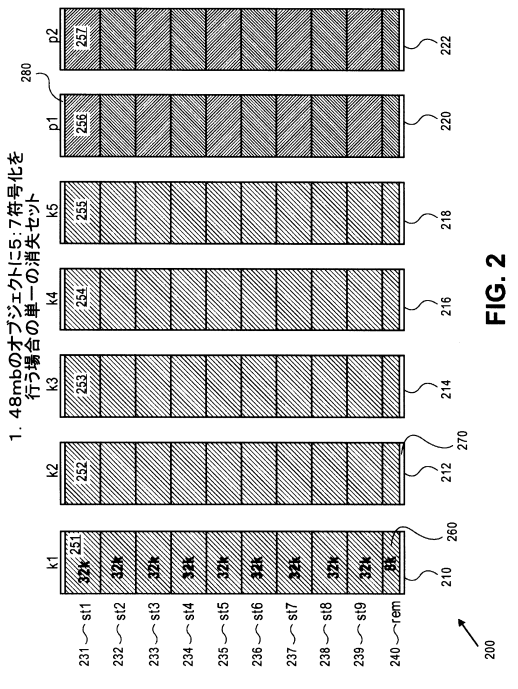

FIG. 2

【図３】

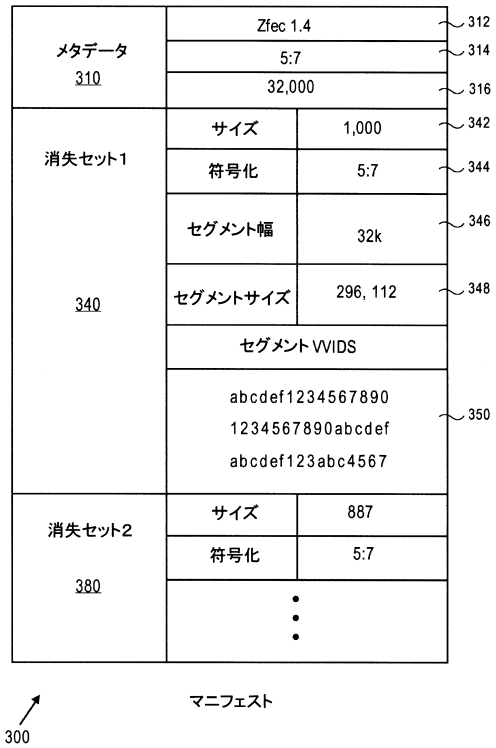

FIG. 3

【図４Ａ】

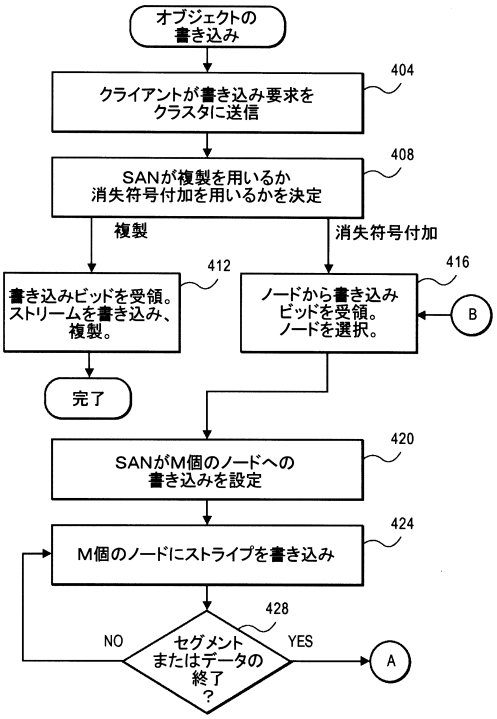

FIG. 4A

【図４Ｂ】

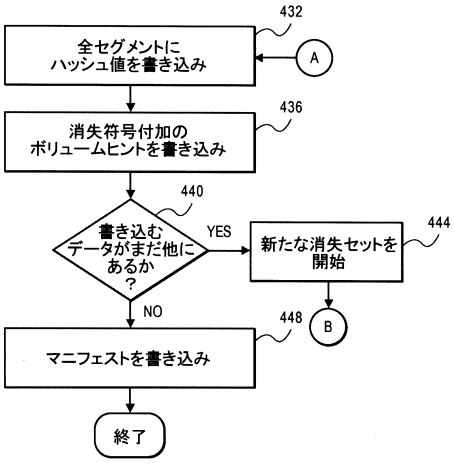

FIG. 4B

【図５】

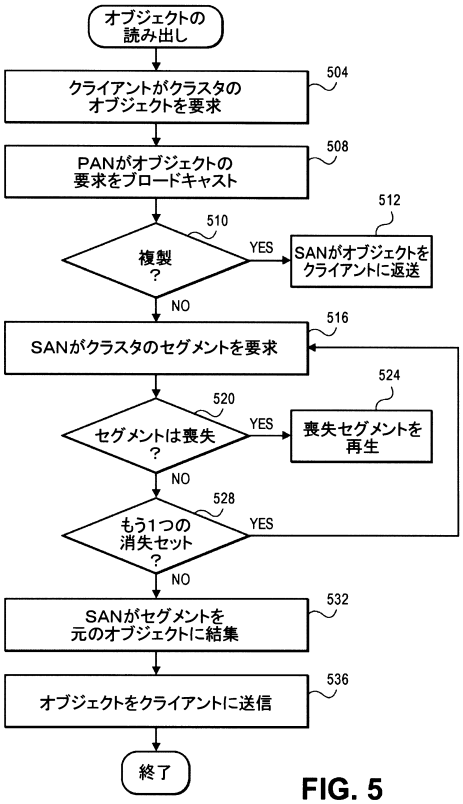

FIG. 5

【図６】

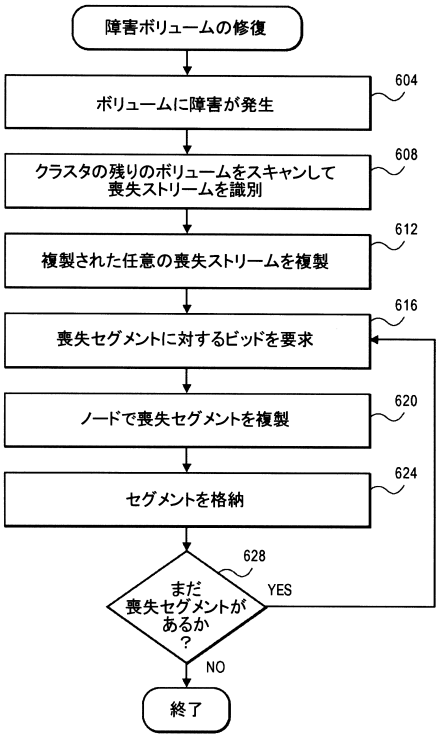

FIG. 6

【図７】

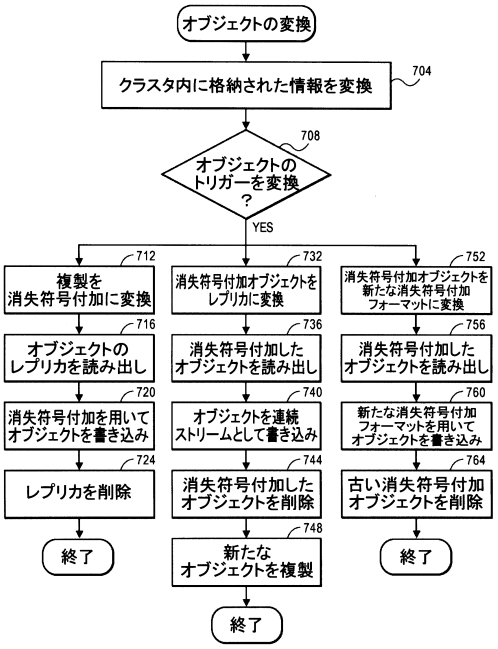

FIG. 7

【図８】

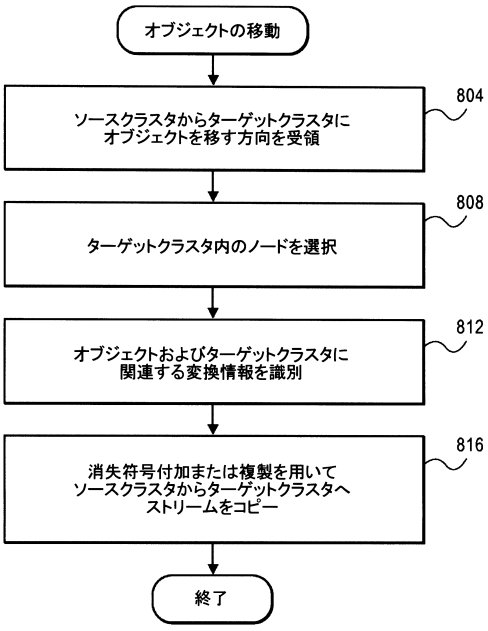

FIG. 8

【図９Ａ】

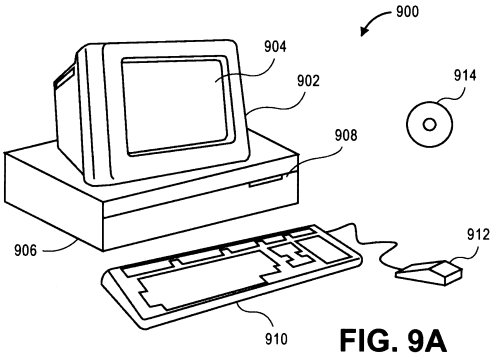

FIG. 9A

【図９Ｂ】

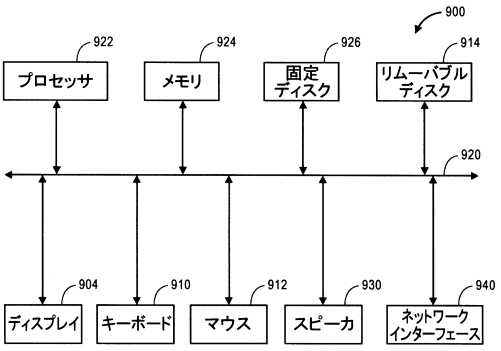

FIG. 9B

フロントページの続き

(72)発明者 カーペンティア・ポール・アール．・エム．
ベルギー国 ボエコート Ｂ－２５３０，ユーベルストラット ４０
(72)発明者 クレイガー・アンドリュー
アメリカ合衆国 テキサス州７８７０３ オースティン，ウェスト・３０番・ストリート，１５０５
(72)発明者 ピアース・アーロン
アメリカ合衆国 テキサス州７８７４９ オースティン，ヘイデン・レーン，９１０８
(72)発明者 リング・ジョナサン
アメリカ合衆国 テキサス州７８７４６ オースティン，トロ・キャニオン・ロード，３９００
(72)発明者 タービン・ラッセル
アメリカ合衆国 テキサス州７８７５１ オースティン，スピードウェイ，４４０４
(72)発明者 ヨークリー・デビッド
アメリカ合衆国 テキサス州７８７３７ オースティン，コーク・レーン，１５０

審査官 桜井 茂行

(56)参考文献 特開２０１０－０４４７８９（ＪＰ，Ａ）
国際公開第２００４／０４６９７１（WO，Ａ１）
特開２００９－１１６４１４（ＪＰ，Ａ）
米国特許出願公開第２００９／０１１９３９５（ＵＳ，Ａ１）
特開平０７－１４１１２３（ＪＰ，Ａ）
米国特許第５７６１４０２（ＵＳ，Ａ）
米国特許出願公開第２０１２／００６００７２（ＵＳ，Ａ１）
米国特許出願公開第２０１１／００２９８４０（ＵＳ，Ａ１）
米国特許出願公開第２０１２／００４７１１１（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １２／００
Ｇ０６Ｆ １７／３０
Ｇ０６Ｆ ３／０６－３／０８
Ｇ０６Ｆ １１／００