(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2014-529804
(P2014-529804A)
(43)公表日　平成26年11月13日(2014.11.13)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 3/06	(2006.01)	G06F 3/06	301Z	
G06F 13/10	(2006.01)	G06F 3/06	301F	
		G06F 13/10	340A	

審査請求 未請求　予備審査請求 未請求　（全 70 頁）

(21)出願番号	特願2014-527263 (P2014-527263)
(86)(22)出願日	平成24年8月22日 (2012.8.22)
(85)翻訳文提出日	平成26年1月30日 (2014.1.30)
(86)国際出願番号	PCT/US2012/051872
(87)国際公開番号	WO2013/032810
(87)国際公開日	平成25年3月7日 (2013.3.7)
(31)優先権主張番号	13/219,368
(32)優先日	平成23年8月26日 (2011.8.26)
(33)優先権主張国	米国 (US)

(71)出願人　510149482
ヴイエムウェア　インコーポレイテッド
ＶＭｗａｒｅ，Ｉｎｃ．
アメリカ合衆国　９４３０４　カリフォルニア州　パロ　アルト　ヒルビュー　アベニュー　３４０１
(74)代理人　100105957
弁理士　恩田　誠
(74)代理人　100068755
弁理士　恩田　博宣
(74)代理人　100142907
弁理士　本田　淳

最終頁に続く

(54)【発明の名称】入力／出力オペレーションのためにオブジェクト・ストレージ・システムを構成すること

(57)【要約】
ストレージ・システムが、ストレージ・オブジェクトとしてプロビジョンされる論理ストレージ・ボリュームをエクスポートする。これらのストレージ・オブジェクトは、ストレージ・システムにおいて構成されているプロトコル・トラフィックに関する論理エンドポイントを通じて、ＳＣＳＩおよびＮＦＳなどの標準的なプロトコルを使用して、接続されているコンピュータ・システムによってオン・デマンドでアクセスされる。ストレージ・システムにおいて入力／出力コマンド（ＩＯ）が受信された場合には、そのＩＯから識別子が取り出され、論理ストレージ・ボリューム識別子へと変換され、その論理ストレージ・ボリューム識別子に対応する論理ストレージ・ボリュームによって参照されるストレージ・ロケーション上でＩＯが実行される。

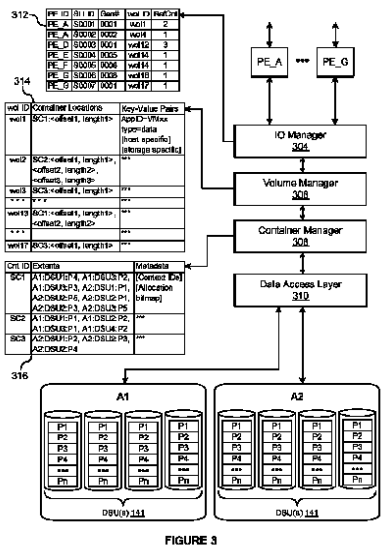


FIGURE 3

【特許請求の範囲】
【請求項１】
ストレージ・システムにおいて入力／出力コマンド（ＩＯ）を処理するための方法であって、
第１および第２の識別子を含むＩＯをストレージ・システムにおいて受信することであって、該第１の識別子が、該ＩＯを該ストレージ・システムへ導くために使用される、前記ＩＯを受信すること、
該ストレージ・システムにおいて、該ＩＯから該第２の識別子を取り出し、該第２の識別子を論理ストレージ・ボリューム識別子に変換すること、
該論理ストレージ・ボリューム識別子に対応する論理ストレージ・ボリュームによって参照されるストレージ・ロケーション上でＩＯを実行すること、を含む方法。
【請求項２】
前記ＩＯが、前記ストレージ・システムにおいて構成されているＬＵＮにおいて受信され、前記第１の識別子が、該ＬＵＮに関するワールド・ワイド・ネームである、請求項１に記載の方法。
【請求項３】
前記論理ストレージ・ボリュームまたは前記ＬＵＮにおけるエラー状況を検知し、前記ＩＯを発行したコンピュータ・システムにエラー・メッセージを送信すること
をさらに含む、請求項２に記載の方法。
【請求項４】
前記ＩＯが、前記ストレージ・システムにおいて構成されているマウント・ポイントにおいて受信され、前記第１の識別子が、前記ストレージ・システムおよび該マウント・ポイントのＩＰアドレスを含む、請求項１に記載の方法。
【請求項５】
前記論理ストレージ・ボリュームまたは前記マウント・ポイントにおけるエラー状況を検知し、前記ＩＯを発行したコンピュータ・システムにエラー・メッセージを送信すること
をさらに含む、請求項４に記載の方法。
【請求項６】
論理ストレージ・ボリューム識別子への第２の識別子のマッピングを提供するデータ構造を保持することであって、該第２の識別子が、該データ構造を使用して該論理ストレージ・ボリューム識別子に変換される、前記保持すること
をさらに含む、請求項１に記載の方法。
【請求項７】
ストレージ・システムを用意して入力／出力コマンド（ＩＯ）を実行するための方法であって、
論理ストレージ・ボリュームを用意するための要求を受信すること、
該論理ストレージ・ボリュームに関するＩＯが受信される際に経由することになる該ストレージ・システムにおいて構成されているプロトコル・エンドポイントを選択すること、
該論理ストレージ・ボリュームに関するＩＯ内に含まれることになる識別子を生成し、該要求を発行したコンピュータ・システムに該識別子を返すこと、を含む方法。
【請求項８】
プロトコル・パスを介して前記コンピュータ・システムに接続されている前記ストレージ・システムにおいて構成されている１つまたは複数のプロトコル・エンドポイントを特定することをさらに含み、
前記プロトコル・エンドポイントが、該１つまたは複数のプロトコル・エンドポイントから選択される、
請求項７に記載の方法。
【請求項９】
前記１つまたは複数のプロトコル・エンドポイントがＬＵＮである、請求項８に記載の方

法。
【請求項１０】
前記１つまたは複数のプロトコル・エンドポイントが、ネットワーク・アタッチト・ストレージ・システムのマウント・ポイントである、請求項８に記載の方法。
【請求項１１】
前記要求が、非プロトコル・パスを介して前記ストレージ・システムにおいて受信され、前記識別子が、該非プロトコル・パスを介して返される、請求項８に記載の方法。
【請求項１２】
ＩＯ内に含まれている識別子の、論理ストレージ・ボリューム識別子へのマッピングを提供するデータ構造を保持することをさらに含み、
前記要求が、論理ストレージ・ボリューム識別子を含み、前記選択されたプロトコル・エンドポイントを識別して生成された識別子を該論理ストレージ・ボリューム識別子に関連付けるエントリーが、該データ構造に加えられる、
請求項７に記載の方法。
【請求項１３】
前記データ構造が、それぞれのエントリーに関するリファレンス・カウントを保持し、該リファレンス・カウントが、前記論理ストレージ・ボリュームを用意するための要求が行われるたびにインクリメントされ、前記論理ストレージ・ボリュームに関するＩＯを受信するために選択されるプロトコル・エンドポイントが、該エントリー内で識別される同じプロトコル・エンドポイントである、請求項１２に記載の方法。
【請求項１４】
前記リファレンス・カウントが、前記論理ボリュームに関するＩＯを受信するために選択された前記プロトコル・エンドポイントから前記論理ストレージ・ボリュームを切り離すための要求が行われるたびにデクリメントされる、請求項１３に記載の方法。
【請求項１５】
前記論理ストレージ・ボリュームに関するＩＯが受信される際に経由することになる前記ストレージ・システムにおいて構成されている新たなプロトコル・エンドポイントを選択し、前記論理ストレージ・ボリュームを該新たなプロトコル・エンドポイントに結合すること、
前記論理ストレージ・ボリュームに関するＩＯ内に含まれることになる新たな識別子を生成し、前記要求を発行したコンピュータ・システムに該新たな識別子を返すこと、
をさらに含む、請求項７に記載の方法。
【請求項１６】
前記論理ストレージ・ボリュームを古いプロトコル・エンドポイントから切り離して、前記論理ストレージ・ボリュームに関する新たなＩＯが、前記新たなプロトコル・エンドポイントを通じて受信されるようにすること
をさらに含む、請求項１５に記載の方法。
【請求項１７】
ストレージ・システムを用意して、ホスト・コンピュータ上で稼働する仮想マシンによって発行される入力／出力コマンド（ＩＯ）を実行するための方法であって、
該仮想マシンに関するメタデータ論理ストレージ・ボリュームを用意するための要求を受信すること、
該メタデータ論理ストレージ・ボリュームに関するＩＯが受信される際に経由することになる該ストレージ・システムにおいて構成されているプロトコル・エンドポイントを選択すること、
該プロトコル・エンドポイントを通じて読み取りＩＯを受信し、データ論理ストレージ・ボリュームに関する識別子を含む読み取りデータを返すこと、
該仮想マシンに関する該データ論理ストレージ・ボリュームを用意するための要求を受信すること、
それぞれのデータ論理ストレージ・ボリュームに関して、論理ストレージ・ボリューム

に関するＩＯが受信される際に経由することになる該ストレージ・システムにおいて構成されているプロトコル・エンドポイントを選択すること、を含む方法。
【請求項１８】
前記メタデータ論理ストレージ・ボリュームが、前記仮想マシンに関する構成ファイルおよびログ・ファイルと、前記仮想マシンに関するスワップ・ファイルと、ディスク記述子ファイルとを含み、ファイルは、前記仮想マシンの仮想ディスクのそれぞれに関するものである、請求項１７に記載の方法。
【発明の詳細な説明】
【背景技術】
【０００１】
コンピュータ・システムが、とりわけ大規模なデータ・センターをサポートするというコンテキストにおいて、エンタープライズ・レベルへ拡張するにつれて、基礎をなすデータ・ストレージ・システムは、ストレージ・エリア・ネットワーク（ＳＡＮ）またはネットワーク・アタッチト・ストレージ（ＮＡＳ）を採用する場合が多い。従来からよく理解されているように、ＳＡＮまたはＮＡＳは、複数の技術的能力および機能上の利点を提供し、それらは基本的に、データ・ストレージ・デバイスの仮想化と、トランスペアレントな、フォルト・トレラントな、フェイルオーバな、およびフェイルセーフなコントロールを伴う物理的なデバイスの冗長性と、地理的に分散され複製されたストレージと、クライアント中心のコンピュータ・システム管理から切り離された集中化された監督およびストレージ構成管理とを含む。
【０００２】
アーキテクチャ上は、ＳＡＮストレージ・システム（たとえば、ディスク・アレイなど）内のストレージ・デバイスは、典型的にはネットワーク・スイッチ（たとえば、ファイバ・チャネル・スイッチなど）に接続され、次いで、それらのネットワーク・スイッチは、ストレージ・デバイス内のデータへのアクセスを必要とするサーバまたは「ホスト」に接続される。ＳＡＮ内のサーバ、スイッチ、およびストレージ・デバイスは、典型的には、ディスク・データ・ブロックのレベルでネットワークを介してデータを転送するスモール・コンピュータ・システム・インターフェース（ＳＣＳＩ）プロトコルを使用して通信する。対照的に、ＮＡＳデバイスは、典型的には、１つまたは複数のストレージ・ドライブを内部に含み、イーサネットなどのネットワーク・プロトコルを通じてホスト（または中間スイッチ）に接続されるデバイスである。ＮＡＳデバイスはまた、ストレージ・デバイスを含むことに加えて、ネットワーク・ファイル・システム（ＮＦＳ）またはコモン・インターネット・ファイル・システム（ＣＩＦＳ）など、ネットワークベースのファイル・システムに従って自分のストレージ・デバイスを事前にフォーマットする。したがって、ＳＡＮは、ディスク（ＬＵＮと呼ばれ、これについては、以降でさらに詳述する）をホストに公開し、次いでそれらのディスクは、フォーマットされてから、ホストによって利用されるファイル・システムに従ってマウントされる必要があるが、そうしたＳＡＮとは対照的に、ＮＡＳデバイスのネットワークベースのファイル・システム（これは、ホストのオペレーティング・システムによってサポートされる必要がある）は、ＮＡＳデバイスが、ホストのオペレーティング・システムにとってファイル・サーバとして見えるようにし、次いでホストは、そのＮＡＳデバイスを、たとえば、オペレーティング・システムによってアクセス可能なネットワーク・ドライブとして、マウントまたはマップすることができる。ストレージ・システム・ベンダーによる継続的なイノベーションおよび新製品のリリースに伴って、ＳＡＮストレージ・システムとＮＡＳストレージ・システムとの間における明確な区別が薄れ続けており、実際のストレージ・システムの実施態様は、しばしば両方の特徴を呈しており、同じシステムにおいてファイルレベル・プロトコル（ＮＡＳ）およびブロックレベル・プロトコル（ＳＡＮ）の両方を提供しているということを認識されたい。たとえば、代替ＮＡＳアーキテクチャにおいては、従来のＮＡＳデバイスではなく、ＮＡＳ「ヘッド」またはＮＡＳ「ゲートウェイ」デバイスが、ホストにネットワーク接続される。そのようなＮＡＳゲートウェイ・デバイスは、自分自身ではストレージ・

ドライブを含まず、外部のストレージ・デバイスが（たとえば、ファイバ・チャネル・インターフェースなどを介して）そのＮＡＳゲートウェイ・デバイスに接続されることを可能にする。そのようなＮＡＳゲートウェイ・デバイス（これは、従来のＮＡＳデバイスと同様の様式でホストによって知覚される）は、ファイルレベルのストレージ・アクセスのシンプルさを保持しながら、ＮＡＳベースのストレージ・アーキテクチャのキャパシティーを（たとえば、ＳＡＮによって、より伝統的にサポートされているストレージ・キャパシティー・レベルに）著しく増大させる能力を提供する。

【０００３】

図１Ａにおいて示されているストレージ・システム３０など、ＳＣＳＩおよびその他のブロック・プロトコルベースのストレージ・デバイスは、１つまたは複数のプログラムされているストレージ・プロセッサに相当するストレージ・システム・マネージャ３１を利用して、そのストレージ・デバイス内のストレージ・ユニットまたはドライブを集約し、それらを、一意に識別可能な番号をそれぞれ伴う１つまたは複数のＬＵＮ（Ｌｏｇｉｃａｌ　Ｕｎｉｔ　Ｎｕｍｂｅｒ）３４として提示する。ＬＵＮ３４は、ネットワーク２０（たとえば、ファイバ・チャネルなど）を介して物理ホスト・バス・アダプタ（ＨＢＡ）１１を通じて１つまたは複数のコンピュータ・システム１０によってアクセスされる。コンピュータ・システム１０内で、ＨＢＡ１１の上に、ストレージ・アクセス・アブストラクションが、ローレベル・デバイス・ドライバ・レイヤ１２から始まってオペレーティング・システム固有のファイル・システム・レイヤ１５で終わる一連のソフトウェア・レイヤを通じて特徴的に実装されている。デバイス・ドライバ・レイヤ１２は、ＬＵＮ３４への基本的なアクセスを可能にし、典型的には、ストレージ・システムによって使用される通信プロトコル（たとえば、ＳＣＳＩなど）に固有のものである。ＨＢＡ１１を通じて見えるＬＵＮ３４のマルチパスの統合、およびその他のデータ・アクセス・コントロールおよび管理機能をサポートするために、データ・アクセス・レイヤ１３が、デバイス・ドライバ・レイヤ１２の上に実装されることが可能である。論理ボリューム・マネージャ１４が、典型的にはデータ・アクセス・レイヤ１３と従来のオペレーティング・システム・ファイル・システム・レイヤ１５との間に実装され、ＨＢＡ１１を通じてアクセス可能なＬＵＮ３４のボリューム指向の仮想化および管理をサポートする。複数のＬＵＮ３４が集められて、１つの論理デバイスとしてファイル・システム・レイヤ１５に提示されてファイル・システム・レイヤ１５によって使用されるために論理ボリューム・マネージャ１４のコントロールのもとで１つのボリュームとしてまとめて管理されることが可能である。

【０００４】

ストレージ・システム・マネージャ３１は、ストレージ・システム３０内に存在する、図１Ａにおいてスピンドル３２と呼ばれている、物理的な、典型的にはディスク・ドライブベースのストレージ・ユニットの仮想化を実施する。論理的な観点からは、これらのスピンドル３２のそれぞれは、固定されたサイズのエクステント３３のシーケンシャル・アレイと考えられることが可能である。ストレージ・システム・マネージャ３１は、ＬＵＮ３４として知られている一組の仮想ＳＣＳＩデバイスへと分けられている連続した論理ストレージ・スペースを、コンピュータ・システム１０など、接続されているコンピュータ・システムに公開することによって、ディスク・ドライブの実際のスピンドルおよびエクステントのアドレスへのターゲットとなる読み取りおよび書き込みオペレーションの複雑さを取り去る。それぞれのＬＵＮは、そのようなＬＵＮの存在と、コンピュータ・システム１０へのそのようなＬＵＮの提示とのおかげでコンピュータ・システム１０によって使用されるために割り振られるいくらかのキャパシティーを表す。ストレージ・システム・マネージャ３１は、それぞれのそのようなＬＵＮに関する、エクステントの順序付けられたリストへのマッピングを含むメタデータを保持し、そのリストにおいては、それぞれのそのようなエクステントは、スピンドル／エクステントのペア＜スピンドル＃，エクステント＃＞として識別されることが可能であり、したがって、さまざまなスピンドル３２のうちのいずれかに位置特定されることが可能である。

【０００５】

図１Ｂは、ネットワーク２１（たとえば、イーサネット）を介してネットワーク・インターフェース・カード（ＮＩＣ）１１’を経由して１つまたは複数のコンピュータ・システム１０に接続される従来のＮＡＳまたはファイルレベル・ベースのストレージ・システム４０のブロック図である。ストレージ・システム４０は、１つまたは複数のプログラムされているストレージ・プロセッサに相当するストレージ・システム・マネージャ４１を含む。ストレージ・システム・マネージャ４１は、ストレージ・システム４０内に存在する、図１Ｂにおいてスピンドル４２と呼ばれている、物理的な、典型的にはディスク・ドライブベースのストレージ・ユニットの上にファイル・システム４５を実装する。論理的な観点からは、これらのスピンドルのそれぞれは、固定されたサイズのエクステント４３のシーケンシャル・アレイと考えられることが可能である。ファイル・システム４５は、各自のマウント・ポイントを通じてアクセスされるファイル・システム・レベル・ボリューム４４（以降では「ＦＳボリューム」と呼ばれる）へと編成されることが可能であるディレクトリおよびファイルを含むネームスペースを、コンピュータ・システム１０など、接続されているコンピュータ・システムに公開することによって、ディスク・ドライブの実際のスピンドルおよびエクステントのアドレスへのターゲットとなる読み取りおよび書き込みオペレーションの複雑さを取り去る。

【０００６】

上述のストレージ・システムにおける進歩をもってさえ、それらのストレージ・システムは、仮想化されたコンピュータ・システムの特定のニーズを満たす上で十分にスケーラブルではないということが広く認識されている。たとえば、サーバ・マシンのクラスタは、１０，０００個もの仮想マシン（ＶＭ：ｖｉｒｔｕａｌ　ｍａｃｈｉｎｅ）にサービス提供することができ、それぞれのＶＭは、複数の「仮想ディスク」および複数の「スナップショット」を使用し、それらはそれぞれ、たとえば、１つのファイルとして特定のＬＵＮまたはＦＳボリューム上に格納されることが可能である。ＶＭごとに２つの仮想ディスクおよび２つのスナップショットというスケール・ダウンされた推定においてさえ、ＶＭが物理ディスクに直接接続される（すなわち、物理ディスクごとに１つの仮想ディスクまたはスナップショット）場合、ストレージ・システムがサポートするのは、合計６０，０００個の別々のディスクに達する。加えて、このスケールでのストレージ・デバイスおよびトポロジーの管理は困難であることがわかっている。結果として、本願明細書に援用する「Ｐｒｏｖｉｄｉｎｇ　Ｍｕｌｔｉｐｌｅ　Ｃｏｎｃｕｒｒｅｎｔ　Ａｃｃｅｓｓ　ｔｏ　ａ　Ｆｉｌｅ　Ｓｙｓｔｅｍ」と題されている（特許文献１）に記載されているような、ＶＭがさらに小さな組の物理ストレージ・エンティティー（たとえば、ＬＵＮベースのＶＭＦＳクラスタ化されたファイル・システムまたはＦＳボリューム）へと多重化されるデータストアというコンセプトが開発された。

【０００７】

ＬＵＮまたはＦＳボリュームを採用している従来のストレージ・システムにおいては、複数のＶＭからのワークロードは、典型的には、単一のＬＵＮまたは単一のＦＳボリュームによってサービス提供される。結果として、１つのＶＭワークロードからのリソース需要が、同じＬＵＮまたはＦＳボリューム上の別のＶＭワークロードに提供されるサービス・レベルに影響を与えることになる。したがって、待ち時間、および１秒あたりの入力／出力処理（ＩＯ）、すなわちＩＯＰＳ（ｉｎｐｕｔ／ｏｕｔｐｕｔ　ｏｐｅｒａｔｉｏｎｓ　ｐｅｒ　ｓｅｃｏｎｄ）など、ストレージに関する効率尺度は、所与のＬＵＮまたはＦＳボリューム内のワークロードの数に応じて変わり、保証されることは不可能である。その結果として、ＬＵＮまたはＦＳボリュームを採用しているストレージ・システムに関するストレージ・ポリシーがＶＭごとに実行されることは不可能であり、サービス・レベル・アグリーメント（ＳＬＡ）保証がＶＭごとに与えられることは不可能である。加えて、スナップショット、複製、暗号化、および重複排除など、ストレージ・システム・ベンダーによって提供されるデータ・サービスは、ＶＭの仮想ディスクの粒度ではなく、ＬＵＮまたはＦＳボリュームの粒度で提供される。結果として、スナップショットは、ストレージ・システム・ベンダーによって提供されるデータ・サービスを使用してＬＵＮ全体ま

たはＦＳボリューム全体に関して作成されることが可能であるが、ＬＵＮ、または仮想ディスクが格納されているファイル・システムとは別に、ＶＭの単一の仮想ディスクに関するスナップショットが作成されることは不可能である。

【先行技術文献】

【特許文献】

【０００８】

【特許文献１】米国特許第７，８４９，０９８号明細書

【発明の概要】

【課題を解決するための手段】

【０００９】

１つまたは複数の実施形態が対象にしているストレージ・システムは、その中で実行されるワークロード同士を分離するように構成されており、それによって、ＳＬＡ保証がワークロードごとに提供されることが可能であり、ストレージ・システムのデータ・サービスがワークロードごとに提供されることが可能であり、ストレージ・システムの抜本的な再設計は必要とされない。複数の仮想マシンに関する複数の仮想ディスクを格納するストレージ・システムにおいては、ＳＬＡ保証が仮想ディスクごとに提供されることが可能であり、ストレージ・システムのデータ・サービスが仮想ディスクごとに提供されることが可能である。

【００１０】

本発明の複数の実施形態によれば、ストレージ・システムは、論理的なストレージ・キャパシティーの割り当て（本明細書においては、「ストレージ・コンテナ」と呼ばれる）から、ワークロードごとにストレージ・オブジェクトとしてプロビジョンされる論理ストレージ・ボリューム（本明細書においては、「仮想ボリューム」と呼ばれる）をエクスポートする。ＶＭに関しては、そのＶＭの仮想ディスクおよびスナップショットのそれぞれに関して仮想ボリュームが作成されることが可能である。一実施形態においては、仮想ボリュームは、ストレージ・システムにおいて構成されているプロトコル・トラフィックに関する論理エンドポイント（「プロトコル・エンドポイント」として知られている）を通じて、ＳＣＳＩおよびＮＦＳなどの標準的なプロトコルを使用して、接続されているコンピュータ・システムによってオン・デマンドでアクセスされる。

【００１１】

本発明の一実施形態による、ストレージ・システムにおいて入力／出力コマンド（ＩＯ）を処理するための方法は、第１および第２の識別子を含むＩＯをストレージ・システムにおいて受信することであって、第１の識別子が、ＩＯをストレージ・システムへ導くために使用される、ＩＯを受信すること、ＩＯから第２の識別子を取り出し、その第２の識別子を論理ストレージ・ボリューム識別子に変換すること、その論理ストレージ・ボリューム識別子に対応する論理ストレージ・ボリュームによって参照されるストレージ・ロケーション上でＩＯを実行することを含む。

【００１２】

本発明の一実施形態による、ストレージ・システムを用意してＩＯを実行するための方法は、論理ストレージ・ボリュームを用意するための要求を受信すること、論理ストレージ・ボリュームに関するＩＯが受信される際に経由することになるストレージ・システムにおいて構成されているプロトコル・エンドポイントを選択すること、論理ストレージ・ボリュームに関するＩＯ内に含まれることになる識別子を生成し、要求を発行したコンピュータ・システムにその識別子を返すことを含む。

【００１３】

本発明の一実施形態による、ストレージ・システムを用意してホスト・コンピュータ上で稼働する仮想マシンによって発行されるＩＯを実行するための方法は、仮想マシンに関するメタデータ論理ストレージ・ボリュームを用意するための要求を受信すること、メタデータ論理ストレージ・ボリュームに関するＩＯが受信される際に経由することになるストレージ・システムにおいて構成されているプロトコル・エンドポイントを選択すること

、そのプロトコル・エンドポイントを通じて読み取りＩＯを受信し、データ論理ストレージ・ボリュームに関する識別子を含む読み取りデータを返すこと、仮想マシンに関するデータ論理ストレージ・ボリュームを用意するための要求を受信すること、それぞれのデータ論理ストレージ・ボリュームに関して、論理ストレージ・ボリュームに関するＩＯが受信される際に経由することになるストレージ・システムにおいて構成されているプロトコル・エンドポイントを選択することを含む。

【００１４】

本発明の複数の実施形態はさらに、コンピュータ・システムによって実行されたときに上述の方法のうちの１つをそのコンピュータ・システムに実行させる命令を格納している非一時的なコンピュータ可読ストレージ・メディアを含む。

【図面の簡単な説明】

【００１５】

【図１Ａ】ネットワークを介して１つまたは複数のコンピュータ・システムに接続されている従来のブロック・プロトコルベースのストレージ・デバイスのブロック図。

【図１Ｂ】ネットワークを介して１つまたは複数のコンピュータ・システムに接続されている従来のＮＡＳデバイスのブロック図。

【図２Ａ】本発明の一実施形態による、仮想ボリュームを実装しているブロック・プロトコルベースのストレージ・システム・クラスタのブロック図。

【図２Ｂ】本発明の一実施形態による、仮想ボリュームを実装しているＮＡＳベースのストレージ・システム・クラスタのブロック図。

【図３】本発明の一実施形態による、仮想ボリュームを管理するための図２Ａまたは図２Ｂのストレージ・システム・クラスタのコンポーネントのブロック図。

【図４】ストレージ・コンテナを作成するための方法工程の流れ図。

【図５Ａ】ＳＡＮベースのストレージ・システム上にホストされる仮想ボリュームを実装するように構成されているコンピュータ・システムの一実施形態のブロック図。

【図５Ｂ】ＮＡＳベースのストレージ・システム上にホストされる仮想ボリュームのために構成されている図５Ａのコンピュータ・システムのブロック図。

【図５Ｃ】ＳＡＮベースのストレージ・システム上にホストされる仮想ボリュームを実装するように構成されているコンピュータ・システムの別の実施形態のブロック図。

【図５Ｄ】ＮＡＳベースのストレージ・システム上にホストされる仮想ボリュームのために構成されている図５Ｃのコンピュータ・システムのブロック図。

【図６】本発明の一実施形態による、仮想ボリュームを管理するために使用されるコンポーネントおよび通信パスを示すコンピュータ環境の簡略化されたブロック図。

【図７】図２Ａまたは図２Ｂのストレージ・システム・クラスタに対してコンピュータ・システムを認証するための方法工程の流れ図。

【図８】一実施形態による、仮想ボリュームを作成するための方法工程の流れ図。

【図９】Ａは、コンピュータ・システムにとって利用可能であるプロトコル・エンドポイントを発見するための方法工程の流れ図、Ｂは、コンピュータ・システムが帯域内パスを介して接続されるプロトコル・エンドポイントをストレージ・システムが発見するための方法工程の流れ図。

【図１０】一実施形態による、仮想ボリューム・バインド要求を発行および実行するための方法工程の流れ図。

【図１１】Ａは、一実施形態による、仮想ボリュームにＩＯを発行するための方法工程の流れ図、Ｂは、一実施形態による、仮想ボリュームにＩＯを発行するための方法工程の流れ図。

【図１２】一実施形態による、ストレージ・システムにおいてＩＯを実行するための方法工程の流れ図。

【図１３】一実施形態による、仮想ボリューム再バインド要求を発行および実行するための方法工程の流れ図。

【図１４】仮想ボリュームのライフ・サイクルの概念図。

【図１５】図２Ａのストレージ・システムを使用する一実施形態による、ＶＭをプロビジョンするための方法工程の流れ図。
【図１６】Ａは、ＶＭをパワー・オンするための方法工程の流れ図、Ｂは、ＶＭをパワー・オフするための方法工程の流れ図。
【図１７】ＶＭのｖｖｏｌのサイズを拡張するための方法工程の流れ図。
【図１８】ストレージ・コンテナ同士の間においてＶＭのｖｖｏｌを移動させるための方法工程の流れ図。
【図１９】テンプレートＶＭからＶＭをクローンするための方法工程の流れ図。
【図２０】別の実施形態による、ＶＭをプロビジョンするための方法工程の流れ図。
【図２１】サンプル・ストレージ能力プロファイルと、プロファイル選択工程を含む、ストレージ・コンテナを作成するための方法とを示す図。
【図２２】ｖｖｏｌを作成して、そのｖｖｏｌに関するストレージ能力プロファイルを定義するための方法工程を示す流れ図。
【図２３】スナップショットを作成するための方法工程を示す流れ図。
【発明を実施するための形態】
【００１６】
図２Ａおよび図２Ｂは、本発明の実施形態による、「仮想ボリューム」を実装しているストレージ・システム・クラスタのブロック図である。このストレージ・システム・クラスタは、１つまたは複数のストレージ・システム、たとえば、ストレージ・システム１３０$_1$および１３０$_2$（これらは、ディスク・アレイであることが可能であり、それぞれが、複数のデータ・ストレージ・ユニット（ＤＳＵ）を有しており、それらのＤＳＵのうちの１つは、図において１４１とラベル付けされている）と、本明細書に記載されている本発明の実施形態を可能にするためのストレージ・システム１３０のさまざまなオペレーションをコントロールするストレージ・システム・マネージャ１３１および１３２とを含む。一実施形態においては、複数のストレージ・システム１３０が、分散型ストレージ・システム・マネージャ１３５を実装することができ、分散型ストレージ・システム・マネージャ１３５は、ストレージ・システム・クラスタのオペレーションを、あたかもそれらが単一の論理ストレージ・システムであるかのようにコントロールする。分散型ストレージ・システム・マネージャ１３５の運用ドメインは、同じデータ・センター内に、または複数のデータ・センターにわたってインストールされているストレージ・システムに及ぶことができる。たとえば、そのような一実施形態においては、分散型ストレージ・システム・マネージャ１３５は、ストレージ・システム・マネージャ１３１を含むことができ、ストレージ・システム・マネージャ１３１は、ストレージ・システム・マネージャ１３２と通信する場合に「マスター」マネージャとして機能し、ストレージ・システム・マネージャ１３２は、「スレーブ」マネージャとして機能するが、分散型ストレージ・システム・マネージャを実装するためのさまざまな代替方法が実施されることが可能であるということを認識されたい。ＤＳＵは、物理ストレージ・ユニット、たとえば、回転ディスクまたはソリッド・ステート・ディスクなどのディスクまたはフラッシュ・ベースのストレージ・ユニットに相当する。複数の実施形態によれば、ストレージ・システム・クラスタは、本明細書においてさらに詳述するように、「仮想ボリューム」（ｖｖｏｌ：ｖｉｒｔｕａｌ　ｖｏｌｕｍｅ）を作成して、コンピュータ・システム１００$_1$および１００$_2$などの接続されているコンピュータ・システムに公開する。コンピュータ・システム１００内で稼働するアプリケーション（たとえば、自分の仮想ディスクにアクセスするＶＭなど）は、図２Ａの実施形態におけるＳＣＳＩ、および図２Ｂの実施形態におけるＮＦＳなど、標準的なプロトコルを使用して、ＳＣＳＩまたはＮＦＳプロトコル・トラフィックに関する論理エンドポイント（「プロトコル・エンドポイント」（ＰＥ）として知られており、ストレージ・システム１３０内で構成されている）を通じて、オン・デマンドでｖｖｏｌにアクセスする。アプリケーション関連のデータ・オペレーションに関する、コンピュータ・システム１００からストレージ・システム１３０への通信パスは、本明細書においては「帯域内」パスと呼ばれる。コンピュータ・システム１００のホスト・バス・アダプタ（

ＨＢＡ）と、ストレージ・システム１３０内で構成されているＰＥとの間における通信パス、およびコンピュータ・システム１００のネットワーク・インターフェース・カード（ＮＩＣ）と、ストレージ・システム１３０内で構成されているＰＥとの間における通信パスは、帯域内パスの例である。帯域内ではなく、かつ典型的には、管理オペレーションを実行するために使用される、コンピュータ・システム１００からストレージ・システム１３０への通信パスは、本明細書においては「帯域外」パスと呼ばれる。コンピュータ・システム１００と、ストレージ・システム１３０との間におけるイーサネット・ネットワーク接続など、帯域外パスの例は、図６において帯域内パスとは別に示されている。簡単にするために、コンピュータ・システム１００は、ストレージ・システム１３０に直接接続されているように示されている。しかしながら、それらのコンピュータ・システム１００は、複数のパス、およびスイッチのうちの１つまたは複数を通じてストレージ・システム１３０に接続されることが可能であるということを理解されたい。

【００１７】

分散型ストレージ・システム・マネージャ１３５、または単一のストレージ・システム・マネージャ１３１もしくは１３２は、（たとえば、コンピュータ・システム１００などの要求に応じて、）物理的なＤＳＵの論理的な集約に相当する論理的な「ストレージ・コンテナ」からｖｖｏｌを作成することができる。一般には、１つのストレージ・コンテナは、複数のストレージ・システムにわたることができ、単一のストレージ・システム・マネージャ、または分散型ストレージ・システム・マネージャによって、多くのストレージ・コンテナが作成されることが可能である。同様に、単一のストレージ・システムは、多くのストレージ・コンテナを含むことができる。図２Ａおよび図２Ｂにおいては、分散型ストレージ・システム・マネージャ１３５によって作成されたストレージ・コンテナ１４２$_A$は、ストレージ・システム１３０$_1$およびストレージ・システム１３０$_2$にわたるものとして示されており、その一方で、ストレージ・コンテナ１４２$_B$およびストレージ・コンテナ１４２$_C$は、単一のストレージ・システム（すなわち、ストレージ・システム１３０$_1$およびストレージ・システム１３０$_2$それぞれ）の中に含まれているものとして示されている。１つのストレージ・コンテナは、複数のストレージ・システムにわたることができるため、ストレージ・システム管理者は、ストレージ・システムのいずれか１つのストレージ・キャパシティーを超えるストレージ・キャパシティーを自分の顧客にプロビジョンすることができるということを認識されたい。単一のストレージ・システム内に複数のストレージ・コンテナが作成されることが可能であるため、ストレージ・システム管理者は、単一のストレージ・システムを使用して複数の顧客にストレージをプロビジョンすることができるということをさらに認識されたい。

【００１８】

図２Ａの実施形態においては、それぞれのｖｖｏｌは、ブロック・ベースのストレージ・システムからプロビジョンされている。図２Ｂの実施形態においては、ＮＡＳベースのストレージ・システムが、ＤＳＵ１４１の上にファイル・システム１４５を実装しており、それぞれのｖｖｏｌは、このファイル・システム内の１つのファイル・オブジェクトとしてコンピュータ・システム１００に公開されている。加えて、以降でさらに詳細に説明するように、コンピュータ・システム１００上で稼働するアプリケーションは、ＰＥを通じたＩＯのためにｖｖｏｌにアクセスする。たとえば、図２Ａおよび図２Ｂにおいて破線で示されているように、ｖｖｏｌ１５１およびｖｖｏｌ１５２は、ＰＥ１６１を介してアクセス可能であり、ｖｖｏｌ１５３およびｖｖｏｌ１５５は、ＰＥ１６２を介してアクセス可能であり、ｖｖｏｌ１５４は、ＰＥ１６３およびＰＥ１６４を介してアクセス可能であり、ｖｖｏｌ１５６は、ＰＥ１６５を介してアクセス可能である。ストレージ・コンテナ１４２$_A$内のｖｖｏｌ１５３、およびストレージ・コンテナ１４２$_C$内のｖｖｏｌ１５５など、複数のストレージ・コンテナからのｖｖｏｌは、任意の所与の時点においてＰＥ１６２などの単一のＰＥを介してアクセス可能とすることができるということを認識されたい。ＰＥ１６６などのＰＥは、それらのＰＥを介してアクセス可能であるｖｖｏｌがまったくなくても、存在することができるということをさらに認識されたい。

【００１９】
図２Ａの実施形態においては、ストレージ・システム１３０は、ＬＵＮをセットアップするための知られている方法を使用して、特別なタイプのＬＵＮとしてＰＥを実装している。ＬＵＮと同様に、ストレージ・システム１３０は、ＷＷＮ（Ｗｏｒｌｄ Ｗｉｄｅ Ｎａｍｅ）として知られている一意の識別子をそれぞれのＰＥに提供する。一実施形態においては、ＰＥを作成する際に、ストレージ・システム１３０は、その特別なＬＵＮに関するサイズを指定しない。なぜなら、本明細書に記載されているＰＥは、実際のデータ・コンテナではないためである。そのような一実施形態においては、ストレージ・システム１３０は、ＰＥ関連のＬＵＮのサイズとしてゼロの値または非常に小さな値を割り振ることができ、それによって管理者は、以降でさらに論じるように、ストレージ・システムがＬＵＮ（たとえば、従来のデータＬＵＮおよびＰＥ関連のＬＵＮ）のリストを提供するように要求する場合に、ＰＥを迅速に識別することができる。同様に、ストレージ・システム１３０は、ＰＥに対するＬＵＮに関する識別番号として、２５５よりも大きな数をＬＵＮに割り振って、それらのＬＵＮがデータＬＵＮではないということを、人間にとってわかりやすい方法で示すことができる。ＰＥとＬＵＮとの間において区別を行うための別の方法として、ＰＥビットがＥｘｔｅｎｄｅｄ Ｉｎｑｕｉｒｙ Ｄａｔａ ＶＰＤページ（ページ８６ｈ）に加えられることが可能である。ＰＥビットは、ＬＵＮがＰＥである場合には１に設定され、ＬＵＮが通常のデータＬＵＮである場合には０に設定される。コンピュータ・システム１００は、ＳＣＳＩコマンドＲＥＰＯＲＴ＿ＬＵＮＳを発行することによって帯域内パスを介してＰＥを発見することと、示されているＰＥビットを調べることによって、それらのＰＥが、本明細書に記載されている実施形態によるＰＥであるか、または従来のデータＬＵＮであるかを判定することとが可能である。コンピュータ・システム１００は、ＬＵＮがＰＥであるか、または従来のＬＵＮであるかをさらに確認するために、ＬＵＮのサイズおよびＬＵＮの番号のプロパティーを任意選択で検査することができる。ＰＥ関連のＬＵＮを通常のデータＬＵＮから区別するために上述の技術のうちの任意の技術が使用されることが可能であるということを認識されたい。一実施形態においては、ＰＥビット技術が、ＰＥ関連のＬＵＮを通常のデータＬＵＮから区別するために使用される唯一の技術である。

【００２０】
図２Ｂの実施形態においては、ＰＥは、ＦＳボリュームに対してマウント・ポイントをセットアップするための知られている方法を使用してストレージ・システム１３０内に作成される。図２Ｂの実施形態において作成されるそれぞれのＰＥは、ＩＰアドレスおよびファイル・システム・パスによって一意に識別され、それらのＩＰアドレスおよびファイル・システム・パスは、従来、合わせて「マウント・ポイント」とも呼ばれている。しかしながら、従来のマウント・ポイントとは異なり、それらのＰＥは、ＦＳボリュームに関連付けられない。加えて、図２ＡのＰＥとは異なり、図２ＢのＰＥは、仮想ボリュームが所与のＰＥにバインドされていない限り、帯域内パスを介してコンピュータ・システム１００によって発見可能ではない。したがって、図２ＢのＰＥは、帯域外パスを介してストレージ・システムによって報告される。

【００２１】
図３は、一実施形態による、仮想ボリュームを管理するための図２Ａまたは図２Ｂのストレージ・システム・クラスタのコンポーネントのブロック図である。それらのコンポーネントは、一実施形態におけるストレージ・システム１３０において実行されるストレージ・システム・マネージャ１３１および１３２のソフトウェア・モジュール、または別の実施形態における分散型ストレージ・システム・マネージャ１３５のソフトウェア・モジュール、すなわち、入力／出力（Ｉ／Ｏ）マネージャ３０４、ボリューム・マネージャ３０６、コンテナ・マネージャ３０８、およびデータ・アクセス・レイヤ３１０を含む。本明細書における実施形態の説明においては、分散型ストレージ・システム・マネージャ１３５によって取られるあらゆるアクションは、実施形態に応じてストレージ・システム・マネージャ１３１またはストレージ・システム・マネージャ１３２によって取られること

が可能であるということを理解されたい。

【００２２】

図３の例においては、分散型ストレージ・システム・マネージャ１３５は、３つのストレージ・コンテナＳＣ１、ＳＣ２、およびＳＣ３をＤＳＵ１４１から作成しており、ＤＳＵ１４１のそれぞれは、Ｐ１からＰｎとラベル付けされているスピンドル・エクステントを有するように示されている。一般に、それぞれのストレージ・コンテナは、固定された物理サイズを有しており、ＤＳＵの特定のエクステントに関連付けられている。図３に示されている例においては、分散型ストレージ・システム・マネージャ１３５は、コンテナ・データベース３１６へのアクセスを有しており、コンテナ・データベース３１６は、それぞれのストレージ・コンテナに関して、そのコンテナＩＤと、物理的なレイアウトの情報と、何らかのメタデータとを格納している。コンテナ・データベース３１６は、コンテナ・マネージャ３０８によって管理および更新され、コンテナ・マネージャ３０８は、一実施形態においては、分散型ストレージ・システム・マネージャ１３５のコンポーネントである。コンテナＩＤとは、ストレージ・コンテナが作成されたときにそのストレージ・コンテナに与えられる汎用一意識別子である。物理的なレイアウトの情報は、所与のストレージ・コンテナに関連付けられているＤＳＵ１４１のスピンドル・エクステントから構成されており、＜システムＩＤ，ＤＳＵ　ＩＤ，エクステント番号＞の順序付けられたリストとして格納されている。メタデータ・セクションは、何らかの一般的なメタデータ、および何らかのストレージ・システム・ベンダー固有のメタデータを含むことができる。たとえば、メタデータ・セクションは、ストレージ・コンテナにアクセスすることを許可されているコンピュータ・システムまたはアプリケーションまたはユーザのＩＤを含むことができる。別の例として、メタデータ・セクションは、ストレージ・コンテナのどの＜システムＩＤ，ＤＳＵ　ＩＤ，エクステント番号＞のエクステントが既存のｖｖｏｌに既に割り当てられているか、およびどの＜システムＩＤ，ＤＳＵ　ＩＤ，エクステント番号＞のエクステントが空いているかを示すためのアロケーション・ビットマップを含む。一実施形態においては、ストレージ・システム管理者は、別々のビジネス・ユニットに関して別々のストレージ・コンテナを作成することができ、それによって、別々のビジネス・ユニットのｖｖｏｌ同士は、同じストレージ・コンテナからはプロビジョンされない。ｖｖｏｌ同士を分けるためのその他のポリシーが適用されることも可能である。たとえば、ストレージ・システム管理者は、クラウド・サービスの別々の顧客のｖｖｏｌ同士が別々のストレージ・コンテナからプロビジョンされるというポリシーを採用することができる。また、ｖｖｏｌ同士が、自分たちの必要とされるサービス・レベルに従ってストレージ・コンテナからグループ化されてプロビジョンされることも可能である。加えて、ストレージ・システム管理者は、ストレージ・コンテナを作成すること、削除すること、およびその他の形で管理すること、たとえば、作成されることが可能であるストレージ・コンテナの数を定義すること、および、ストレージ・コンテナごとに設定されることが可能である最大の物理サイズを設定することなどが可能である。

【００２３】

また、図３の例においては、分散型ストレージ・システム・マネージャ１３５は、（要求を行っているコンピュータ・システム１００のために）複数のｖｖｏｌを、それぞれ別々のストレージ・コンテナからプロビジョンしている。一般に、ｖｖｏｌは、固定された物理サイズを有することができ、またはシン・プロビジョニングされることが可能であり、それぞれのｖｖｏｌは、ｖｖｏｌ　ＩＤを有しており、ｖｖｏｌ　ＩＤとは、ｖｖｏｌが作成されたときにそのｖｖｏｌに与えられる汎用一意識別子である。それぞれのｖｖｏｌに関して、ｖｖｏｌデータベース３１４が、それぞれのｖｖｏｌごとに、そのｖｖｏｌ　ＩＤと、そのｖｖｏｌが作成されているストレージ・コンテナのコンテナＩＤと、そのｖｖｏｌのアドレス空間を含むそのストレージ・コンテナ内の＜オフセット，長さ＞の値の順序付けられたリストとを格納している。ｖｖｏｌデータベース３１４は、ボリューム・マネージャ３０６によって管理および更新され、ボリューム・マネージャ３０６は、一実施形態においては、分散型ストレージ・システム・マネージャ１３５のコンポーネント

である。一実施形態においては、ｖｖｏｌデータベース３１４は、ｖｖｏｌに関する少量のメタデータも格納する。このメタデータは、一組のキー／値のペアとしてｖｖｏｌデータベース３１４内に格納され、そのｖｖｏｌの存在中はいつでも帯域外パスを介してコンピュータ・システム１００によって更新されることおよびクエリーされることが可能である。格納されるキー／値のペアは、３つのカテゴリーへと分かれる。第１のカテゴリーは、よく知られているキーであり、特定のキーの定義（ひいては、それらの値の解釈）は、公に利用可能である。１例は、仮想ボリューム・タイプ（たとえば、仮想マシンの実施形態においては、ｖｖｏｌがＶＭのメタデータを含むか、またはＶＭのデータを含むか）に対応するキーである。別の例は、Ａｐｐ　ＩＤであり、これは、ｖｖｏｌ内にデータを格納したアプリケーションのＩＤである。第２のカテゴリーは、コンピュータ・システム固有のキーであり、コンピュータ・システムまたはその管理モジュールは、特定のキーおよび値を仮想ボリュームのメタデータとして格納する。第３のカテゴリーは、ストレージ・システム・ベンダー固有のキーであり、これらによって、ストレージ・システム・ベンダーは、仮想ボリュームのメタデータに関連付けられている特定のキーを格納することができる。ストレージ・システム・ベンダーがそのメタデータに関してこのキー／値の格納を使用する１つの理由は、これらのキーのすべてが、ｖｖｏｌに関する帯域外チャネルを介してストレージ・システム・ベンダーのプラグインおよびその他の拡張にとって容易に利用可能であることである。キー／値のペアに関する格納オペレーションは、仮想ボリュームの作成およびその他の工程の一部であり、したがって格納オペレーションは、適度に高速になるはずである。ストレージ・システムはまた、特定のキー上で提供される値に対する厳密な一致に基づいて仮想ボリュームの検索を可能にするように構成される。

【００２４】

ＩＯマネージャ３０４は、ＰＥとｖｖｏｌとの間における現在有効なＩＯ接続パスを記憶する接続データベース３１２を保持するソフトウェア・モジュール（また、特定の実施形態においては、分散型ストレージ・システム・マネージャ１３５のコンポーネント）である。図３に示されている例においては、７つの現在有効なＩＯセッションが示されている。それぞれの有効なセッションは、関連付けられているＰＥ　ＩＤと、セカンダリー・レベル識別子（ＳＬＬＩＤ）と、ｖｖｏｌ　ＩＤと、このＩＯセッションを通じてＩＯを実行している別々のアプリケーションの数を示すリファレンス・カウント（ＲｅｆＣｎｔ）とを有する。（たとえば、コンピュータ・システム１００による要求に応じて）分散型ストレージ・システム・マネージャ１３５によってＰＥとｖｖｏｌとの間における有効なＩＯセッションを確立する工程は、本明細書においては「バインド」工程と呼ばれる。それぞれのバインドごとに、分散型ストレージ・システム・マネージャ１３５は、（たとえば、ＩＯマネージャ３０４を介して）接続データベース３１２にエントリーを加える。その後に分散型ストレージ・システム・マネージャ１３５によってＩＯセッションを取り壊す工程は、本明細書においては「アンバインド」工程と呼ばれる。それぞれのアンバインドごとに、分散型ストレージ・システム・マネージャ１３５は、（たとえば、ＩＯマネージャ３０４を介して）ＩＯセッションのリファレンス・カウントを１ずつデクリメントする。ＩＯセッションのリファレンス・カウントがゼロである場合には、分散型ストレージ・システム・マネージャ１３５は、（たとえば、ＩＯマネージャ３０４を介して）そのＩＯ接続パスに関するエントリーを接続データベース３１２から削除することができる。前述したように、一実施形態においては、コンピュータ・システム１００は、バインド要求およびアンバインド要求を生成して、帯域外パスを介して分散型ストレージ・システム・マネージャ１３５へ送信する。あるいは、コンピュータ・システム１００は、アンバインド要求を生成して、オーバーローディングが存在しているエラー・パスによって帯域内パスを介して送信することができる。一実施形態においては、リファレンス・カウントが０から１に変わった場合、またはその逆の場合には、世代番号は、単調に増加する番号、またはランダムに生成された番号に変更される。別の実施形態においては、世代番号は、ランダムに生成された番号であり、接続データベース３１２からＲｅｆＣｎｔ列が取り除かれており、それぞれのバインドごとに、バインド要求が、既にバインドされているｖｖｏ

１に対してなされた場合でさえ、分散型ストレージ・システム・マネージャ１３５は、（たとえば、ＩＯマネージャ３０４を介して）接続データベース３１２にエントリーを加える。

【００２５】

図２Ａのストレージ・システム・クラスタにおいては、ＩＯマネージャ３０４は、接続データベース３１２を使用してＰＥを通じて受信されたコンピュータ・システム１００からのＩＯ要求（ＩＯ）を処理する。ＩＯがＰＥのうちの１つにおいて受信された場合には、ＩＯマネージャ３０４は、そのＩＯの対象であったｖｖｏｌを特定する目的で、そのＩＯ内に含まれているＰＥ　ＩＤおよびＳＬＬＩＤを識別するために、そのＩＯを解析する。次いで、接続データベース３１４にアクセスすることによって、ＩＯマネージャ３０４は、解析されたＰＥ　ＩＤおよびＳＬＬＩＤに関連付けられているｖｖｏｌ　ＩＤを取り出すことができる。図３および後続の図においては、簡単にするために、ＰＥ　ＩＤは、ＰＥ＿Ａ、ＰＥ＿Ｂなどと示されている。一実施形態においては、実際のＰＥ　ＩＤは、ＰＥのＷＷＮである。加えて、ＳＬＬＩＤは、Ｓ０００１、Ｓ０００２などと示されている。実際のＳＬＬＩＤは、接続データベース３１２内の所与のＰＥ　ＩＤに関連付けられている複数のＳＬＬＩＤのうちの任意の一意の番号として、分散型ストレージ・システム・マネージャ１３５によって生成される。ｖｖｏｌ　ＩＤを有する仮想ボリュームの論理アドレス空間と、ＤＳＵ１４１の物理ロケーションとの間におけるマッピングは、ｖｖｏｌデータベース３１４を使用してボリューム・マネージャ３０６によって、およびコンテナ・データベース３１６を使用してコンテナ・マネージャ３０８によって実行される。ＤＳＵ１４１の物理ロケーションが得られると、データ・アクセス・レイヤ３１０（一実施形態においては、やはり分散型ストレージ・システム・マネージャ１３５のコンポーネント）が、これらの物理ロケーション上でＩＯを実行する。

【００２６】

図２Ｂのストレージ・システム・クラスタにおいては、ＩＯは、ＰＥを通じて受信され、そのようなそれぞれのＩＯは、ＮＦＳハンドル（または類似のファイル・システム・ハンドル）を含み、そのＮＦＳハンドルに対して、そのＩＯは発行されている。一実施形態においては、そのようなシステムのための接続データベース３１２は、ＰＥ　ＩＤとしてストレージ・システムのＮＦＳインターフェースのＩＰアドレスを、およびＳＬＬＩＤとしてファイル・システム・パスを含む。ＳＬＬＩＤは、ファイル・システム１４５内のｖｖｏｌのロケーションに基づいて生成される。ｖｖｏｌの論理アドレス空間と、ＤＳＵ１４１の物理ロケーションとの間におけるマッピングは、ｖｖｏｌデータベース３１４を使用してボリューム・マネージャ３０６によって、およびコンテナ・データベース３１６を使用してコンテナ・マネージャ３０８によって実行される。ＤＳＵ１４１の物理ロケーションが得られると、データ・アクセス・レイヤが、これらの物理ロケーション上でＩＯを実行する。図２Ｂのストレージ・システムに関しては、コンテナ・データベース３１２は、所与のｖｖｏｌに関してコンテナ・ロケーション・エントリー内にファイル：＜オフセット，長さ＞エントリーの順序付けられたリストを含むことができる（すなわち、ｖｖｏｌは、ファイル・システム１４５内に格納されている複数のファイル・セグメントから構成されることが可能である）ということを認識されたい。

【００２７】

一実施形態においては、接続データベース３１２は、揮発性メモリ内に保持され、その一方で、ｖｖｏｌデータベース３１４およびコンテナ・データベース３１６は、ＤＳＵ１４１などの永続ストレージ内に保持される。その他の実施形態においては、データベース３１２、３１４、３１６のすべてが永続ストレージ内に保持されることが可能である。

【００２８】

図４は、ストレージ・コンテナを作成するための方法工程４１０の流れ図である。一実施形態においては、これらの工程は、ストレージ管理者のコントロールのもとで、ストレージ・システム・マネージャ１３１、ストレージ・システム・マネージャ１３２、または分散型ストレージ・システム・マネージャ１３５によって実行される。上述したように、

ストレージ・コンテナは、物理的なＤＳＵの論理的な集約に相当し、複数のストレージ・システムからの物理的なＤＳＵにわたることができる。工程４１１において、ストレージ管理者は、（分散型ストレージ・システム・マネージャ１３５などを介して、）ストレージ・コンテナの物理的なキャパシティーを設定する。クラウドまたはデータ・センター内では、この物理的なキャパシティーは、たとえば、顧客によってリースされる物理ストレージの量に相当することができる。本明細書において開示されているストレージ・コンテナによって提供される柔軟性として、別々の顧客の複数のストレージ・コンテナが、１人のストレージ管理者によって同じストレージ・システムからプロビジョンされることが可能であり、また、たとえば、いずれか１つのストレージ・デバイスの物理的なキャパシティーが、顧客によって要求されているサイズを満たすのに十分ではないケースにおいて、または１つのｖｖｏｌの物理ストレージ・フットプリントが、必然的に複数のストレージ・システムにわたることになる複製などのケースにおいて、単一の顧客のための１つのストレージ・コンテナが、複数のストレージ・システムからプロビジョンされることが可能である。工程４１２において、ストレージ管理者は、ストレージ・コンテナにアクセスするための許可レベルを設定する。マルチテナント・データ・センターにおいては、たとえば、顧客は、自分にリースされているストレージ・コンテナにアクセスすることしかできない。工程４１３において、分散型ストレージ・システム・マネージャ１３５は、ストレージ・コンテナに関する一意の識別子を生成する。次いで工程４１４において、分散型ストレージ・システム・マネージャ１３５は、（たとえば、一実施形態においては、コンテナ・マネージャ３０８を介して、）ＤＳＵ１４１の空いているスピンドル・エクステントを、工程４１１において設定された物理的なキャパシティーを満たすのに十分な量でストレージ・コンテナに割り当てる。上述したように、いずれか１つのストレージ・システムの空きスペースが、物理的なキャパシティーを満たすのに十分ではないケースにおいては、分散型ストレージ・システム・マネージャ１３５は、複数のストレージ・システムからＤＳＵ１４１のスピンドル・エクステントを割り当てることができる。パーティションが割り当てられた後に、分散型ストレージ・システム・マネージャ１３５は、（たとえば、コンテナ・マネージャ３０８を介して、）一意のコンテナＩＤと、＜システム番号，ＤＳＵ　ＩＤ，エクステント番号＞の順序付けられたリストと、ストレージ・コンテナにアクセスすることを許可されているコンピュータ・システムのコンテキストＩＤとでコンテナ・データベース３１６を更新する。

【００２９】

本明細書に記載されている実施形態によれば、ストレージ能力プロファイル、たとえば、ＳＬＡまたはサービス品質（ＱｏＳ：ｑｕａｌｉｔｙ　ｏｆ　ｓｅｒｖｉｃｅ）は、ｖｖｏｌごとに（たとえば、要求を行っているコンピュータ・システム１００のために）分散型ストレージ・システム・マネージャ１３５によって構成されることが可能である。したがって、別々のストレージ能力プロファイルを有する複数のｖｖｏｌが、同じストレージ・コンテナの一部であることが可能である。一実施形態においては、システム管理者は、ストレージ・コンテナの作成時に、新たに作成されたｖｖｏｌに関するデフォルトのストレージ能力プロファイル（または、複数の可能なストレージ能力プロファイル）を定義して、コンテナ・データベース３１６のメタデータ・セクション内に格納した。ストレージ・コンテナ内で作成されている新たなｖｖｏｌに関してストレージ能力プロファイルが明示的に指定されない場合には、その新たなｖｖｏｌは、そのストレージ・コンテナに関連付けられているデフォルトのストレージ能力プロファイルを引き継ぐことになる。

【００３０】

図５Ａは、図２Ａのストレージ・システム・クラスタ上にホストされる仮想ボリュームを実装するように構成されているコンピュータ・システムの一実施形態のブロック図である。コンピュータ・システム１０１は、従来の、典型的にはサーバクラスである、ハードウェア・プラットフォーム５００上に構築されることが可能であり、ハードウェア・プラットフォーム５００は、１つまたは複数の中央処理装置（ＣＰＵ）５０１と、メモリ５０２と、１つまたは複数のネットワーク・インターフェース・カード（ＮＩＣ）５０３と、

１つまたは複数のホスト・バス・アダプタ（ＨＢＡ）５０４とを含む。ＨＢＡ５０４は、コンピュータ・システム１０１が、ストレージ・デバイス１３０内に構成されているＰＥを通じて仮想ボリュームにＩＯを発行することを可能にする。図５Ａにおいてさらに示されているように、オペレーティング・システム５０８は、ハードウェア・プラットフォーム５００の上にインストールされており、複数のアプリケーション512_1～512_Nが、オペレーティング・システム５０８の上で実行される。オペレーティング・システム５０８の例としては、よく知られているコモディティー・オペレーティング・システム、たとえばＭｉｃｒｏｓｏｆｔ　Ｗｉｎｄｏｗｓ、Ｌｉｎｕｘなどのうちの任意のものが含まれる。

【００３１】

本明細書に記載されている実施形態によれば、それぞれのアプリケーション５１２は、自分に関連付けられている１つまたは複数のｖｖｏｌを有しており、アプリケーション５１２によるオペレーティング・システム５０８への「ＣＲＥＡＴＥ　ＤＥＶＩＣＥ」コールに従ってオペレーティング・システム５０８によって作成されたｖｖｏｌのブロック・デバイス・インスタンスにＩＯを発行する。ブロック・デバイス名と、ｖｖｏｌ　ＩＤとの間における関連付けは、ブロック・デバイス・データベース５３３内に保持される。アプリケーション512_2～512_NからのＩＯは、ファイル・システム・ドライバ５１０によって受信され、ファイル・システム・ドライバ５１０は、それらのＩＯをブロックＩＯに変換し、それらのブロックＩＯを仮想ボリューム・デバイス・ドライバ５３２に提供する。その一方で、アプリケーション512_1からのＩＯは、ファイル・システム・ドライバ５１０を迂回するように示されており、仮想ボリューム・デバイス・ドライバ５３２に直接提供され、これが意味するのは、アプリケーション512_1が、自分のブロック・デバイスに直接、ロー・ストレージ・デバイス（raw storage device）として、たとえば、データベース・ディスク、ログ・ディスク、バックアップ・アーカイブ、およびコンテンツ・リポジトリとして、「Ｐｒｏｖｉｄｉｎｇ　Ａｃｃｅｓｓ　ｔｏ　ａ　Ｒａｗ　Ｄａｔａ　Ｓｔｏｒａｇｅ　Ｕｎｉｔ　ｉｎ　ａ　Ｃｏｍｐｕｔｅｒ　Ｓｙｓｔｅｍ」と題されている米国特許第７，１５５，５５８号（その全内容を本願明細書に援用する）に記載されている様式でアクセスするということである。仮想ボリューム・デバイス・ドライバ５３２は、ブロックＩＯを受信した場合には、ブロック・デバイス・データベース５３３にアクセスして、そのＩＯ内で指定されているブロック・デバイス名と、そのブロック・デバイス名に関連付けられているｖｖｏｌへのＩＯ接続パスを定義するＰＥ　ＩＤ（ＰＥ　ＬＵＮのＷＷＮ）およびＳＬＬＩＤとの間におけるマッピングを参照する。ここで示されている例においては、「ａｒｃｈｉｖｅ」というブロック・デバイス名は、アプリケーション512_1に関して作成されたｖｖｏｌ１２のブロック・デバイス・インスタンスに対応しており、「ｆｏｏ」、「ｄｂａｓｅ」、および「ｌｏｇ」というブロック・デバイス名は、アプリケーション512_2～512_Nのうちの１つまたは複数に関してそれぞれ作成されたｖｖｏｌ１、ｖｖｏｌ１６、およびｖｖｏｌ１７のブロック・デバイス・インスタンスに対応する。ブロック・デバイス・データベース５３３内に格納されているその他の情報としては、ブロック・デバイスがアクティブであるか否かを示すそれぞれのブロック・デバイスに関するアクティブ・ビット値と、ＣＩＦ（ｃｏｍｍａｎｄｓ－ｉｎ－ｆｌｉｇｈｔ：処理中コマンド）値とが含まれる。「１」というアクティブ・ビットは、ＩＯがブロック・デバイスに発行されることが可能であるということを意味する。「０」というアクティブ・ビットは、ブロック・デバイスが非アクティブであり、ＩＯがブロック・デバイスに発行されることは不可能であるということを意味する。ＣＩＦ値は、いくつのＩＯが処理中であるか、すなわち、発行されたが完了されていないかの表示を提供する。ここで示されている例においては、「ｆｏｏ」というブロック・デバイスは、アクティブであり、いくつかの処理中コマンドを有している。「ａｒｃｈｉｖｅ」というブロック・デバイスは、非アクティブであり、さらに新しいコマンドを受け入れないであろう。しかしながら、このブロック・デバイスは、２つの処理中コマンドが完了するのを待っている。「ｄｂａｓｅ」というブロック・デバイスは、非アクティブであり、未処理のコマ

ンドはない。最後に、「ｌｏｇ」というブロック・デバイスは、アクティブであるが、アプリケーションは現在、このデバイスに対する未処理のＩＯを有していない。仮想ボリューム・デバイス・ドライバ５３２は、いつでも自分のデータベース５３３からこれらのようなデバイスを除去することを選択することができる。

【００３２】

上述のマッピングを実行することに加えて、仮想ボリューム・デバイス・ドライバ５３２は、データ・アクセス・レイヤ５４０にロー・ブロックレベルＩＯ（raw block level IO）を発行する。データ・アクセス・レイヤ５４０は、コマンド・キューイングおよびスケジューリング・ポリシーをロー・ブロックレベルＩＯに適用するデバイス・アクセス・レイヤ５３４と、プロトコルに準拠したフォーマットでロー・ブロックレベルＩＯをフォーマットして、それらのロー・ブロックレベルＩＯを、帯域内パスを介してＰＥへ転送するためにＨＢＡ５０４に送信する、ＨＢＡ５０４のためのデバイス・ドライバ５３６とを含む。ＳＣＳＩプロトコルが使用される実施形態においては、ｖｖｏｌ情報は、ＳＡＭ－５（ＳＣＳＩ　Ａｒｃｈｉｔｅｃｔｕｒｅ　Ｍｏｄｅｌ－５）において指定されているように、ＳＣＳＩ　ＬＵＮデータ・フィールド（これは、８バイト構造である）内にエンコードされる。ＰＥ　ＩＤは、最初の２バイト（これは、従来はＬＵＮ　ＩＤ用に使用されている）内にエンコードされ、ｖｖｏｌ情報、とりわけＳＬＬＩＤは、残っている６バイト（の一部）を利用して、ＳＣＳＩセカンド・レベルＬＵＮ　ＩＤ内にエンコードされる。

【００３３】

図５Ａにおいてさらに示されているように、データ・アクセス・レイヤ５４０はまた、ストレージ・システムから帯域内パスを通じて受信されるＩＯエラーを取り扱うためのエラー・ハンドリング・ユニット５４２を含む。一実施形態においては、エラー・ハンドリング・ユニット５４２によって受信されたＩＯエラーは、Ｉ／Ｏマネージャ３０４によってＰＥを通じて伝搬される。ＩＯエラー・クラスの例としては、コンピュータ・システム１０１とＰＥとの間におけるパス・エラーと、ＰＥエラーと、ｖｖｏｌエラーとが含まれる。エラー・ハンドリング・ユニット５４２は、検知されたすべてのエラーを上述のクラスへと分類する。ＰＥへのパス・エラーに出くわし、ＰＥへの別のパスが存在する場合には、データ・アクセス・レイヤ５４０は、ＰＥへの別のパスに沿ってＩＯを送信する。ＩＯエラーがＰＥエラーである場合には、エラー・ハンドリング・ユニット５４２は、ＰＥを通じてＩＯを発行しているそれぞれのブロック・デバイスに関するエラー状況を示すために、ブロック・デバイス・データベース５３３を更新する。ＩＯエラーがｖｖｏｌエラーである場合には、エラー・ハンドリング・ユニット５４２は、ｖｖｏｌに関連付けられているそれぞれのブロック・デバイスに関するエラー状況を示すために、ブロック・デバイス・データベース５３３を更新する。エラー・ハンドリング・ユニット５４２は、アラームまたはシステム・イベントを発行することもでき、それによって、エラー状況を有するブロック・デバイスへのさらなるＩＯは、拒否されることになる。

【００３４】

図５Ｂは、図２Ａのストレージ・システム・クラスタの代わりに図２Ｂのストレージ・システム・クラスタとインターフェースを取るように構成されている図５Ａのコンピュータ・システムのブロック図である。この実施形態においては、データ・アクセス・レイヤ５４０は、ＮＦＳクライアント５４５と、ＮＩＣ５０３のためのデバイス・ドライバ５４６とを含む。ＮＦＳクライアント５４５は、ブロック・デバイス名を、ＰＥ　ＩＤ（ＮＡＳストレージ・システムのＩＰアドレス）と、ブロック・デバイスに対応するＮＦＳファイル・ハンドルであるＳＬＬＩＤとにマップする。このマッピングは、図５Ｂにおいて示されているように、ブロック・デバイス・データベース５３３内に格納される。「アクティブ」および「ＣＩＦ」の列は、依然として存在するが、図５Ｂに示されているブロック・デバイス・データベース５３３においては示されていないということに留意されたい。以降で説明するように、ＮＦＳファイル・ハンドルは、ＮＡＳストレージ・システム内のファイル・オブジェクトを一意に識別し、バインド工程中に生成されることが可能である

。あるいは、ｖｖｏｌをバインドしたいという要求に応答して、ＮＡＳストレージ・システムは、ＰＥ　ＩＤおよびＳＬＬＩＤを返し、通常の帯域内メカニズム（たとえば、ルックアップまたはｒｅａｄｄｉｒｐｌｕｓ）を使用してｖｖｏｌが開かれると、ＮＦＳファイル・ハンドルが与えられることになる。ＮＦＳクライアント５４５はまた、仮想ボリューム・デバイス・ドライバ５３２から受信されたロー・ブロックレベルＩＯをＮＦＳファイルベースのＩＯに変換する。次いで、ＮＩＣ５０３のためのデバイス・ドライバ５４６は、プロトコルに準拠したフォーマットでＮＦＳファイルベースのＩＯをフォーマットして、それらのＮＦＳファイルベースのＩＯを、帯域内パスを介してＰＥのうちの１つへ転送するために、ＮＦＳハンドルとともに、ＮＩＣ５０３へ送信する。

【００３５】

図５Ｃは、仮想ボリュームを実装するように構成されているコンピュータ・システムの別の実施形態のブロック図である。この実施形態においては、コンピュータ・システム１０２は、仮想化ソフトウェア（ここでは、ハイパーバイザ５６０として示されている）を伴って構成されている。ハイパーバイザ５６０は、ハードウェア・プラットフォーム５５０の上にインストールされており、ハードウェア・プラットフォーム５５０は、ＣＰＵ５５１、メモリ５５２、ＮＩＣ５５３、およびＨＢＡ５５４を含み、仮想マシン実行スペース５７０をサポートし、仮想マシン実行スペース５７０内では、複数の仮想マシン（ＶＭ）571_1～571_Nが、同時にインスタンス化されて実行されることが可能である。１つまたは複数の実施形態においては、ハイパーバイザ５６０および仮想マシン５７１は、ヴイエムウェア・インコーポレイティッド社（ＶＭｗａｒｅ，　Ｉｎｃ．）［米国カリフォルニア州パロアルト（Ｐａｌｏ　Ａｌｔｏ）所在］によって販売されているＶＭｗａｒｅ　ｖＳｐｈｅｒｅ（登録商標）製品を使用して実装される。それぞれの仮想マシン５７１は、仮想ハードウェア・プラットフォーム５７３を実装しており、仮想ハードウェア・プラットフォーム５７３は、アプリケーション５７９を実行することができるゲスト・オペレーティング・システム（ＯＳ）５７２のインストレーションをサポートする。ゲストＯＳ５７２の例としては、よく知られているコモディティー・オペレーティング・システム、たとえばＭｉｃｒｏｓｏｆｔ　Ｗｉｎｄｏｗｓ、Ｌｉｎｕｘなどのうちの任意のものが含まれる。それぞれのインスタンスにおいて、ゲストＯＳ５７２は、ネイティブ・ファイル・システム・レイヤ（図５Ｃにおいては示されていない）、たとえば、ＮＴＦＳまたはｅｘｔ３ＦＳタイプのファイル・システム・レイヤのいずれかを含む。これらのファイル・システム・レイヤは、仮想ハードウェア・プラットフォーム５７３とインターフェースを取って、ゲストＯＳ５７２の視点からは、データ・ストレージＨＢＡにアクセスするが、このデータ・ストレージＨＢＡは、実際には、ゲストＯＳ５７２の実行を可能にするためにディスク・ストレージ・サポートの外見（実際には、仮想ディスク、すなわち仮想ディスク575_A～575_X）を提供する仮想ハードウェア・プラットフォーム５７３によって実装されている仮想ＨＢＡ５７４である。特定の実施形態においては、仮想ディスク575_A～575_Xは、ゲストＯＳ５７２の視点からは、仮想マシンに接続するためのＳＣＳＩ標準、または、ＩＤＥ、ＡＴＡ、およびＡＴＡＰＩを含む、当技術分野における標準的な技術者に知られているその他の任意の適切なハードウェア接続インターフェース標準をサポートするように見えることが可能である。ゲストＯＳ５７２の視点からは、ファイル・システム関連のデータ転送およびコントロール・オペレーションを実施するためにそのようなゲストＯＳ５７２によって開始されたファイル・システム・コールは、最終的な実行のために仮想ディスク575_A～575_Xへ回送されるように見えるが、実際には、そのようなコールは、処理され、仮想ＨＢＡ５７４を通じて補助的な仮想マシン・モニタ（ＶＭＭ）561_1～561_Nに渡され、ＶＭＭ561_1～561_Nは、ハイパーバイザ５６０とのオペレーションを調整するために必要とされる仮想システム・サポートを実施する。とりわけ、ＨＢＡエミュレータ５６２は、データ転送およびコントロール・オペレーションがハイパーバイザ５６０によって正しく取り扱われることを機能的に可能にし、ハイパーバイザ５６０は最終的に、そのようなオペレーションを、自分のさまざまなレイヤを通じて、ストレージ・システム１３０へ接続しているＨＢＡ５５４に渡す。

【００３６】
本明細書に記載されている実施形態によれば、それぞれのＶＭ５７１は、自分に関連付けられている１つまたは複数のｖｖｏｌを有しており、ＶＭ５７１によるハイパーバイザ５６０への「ＣＲＥＡＴＥ　ＤＥＶＩＣＥ」コールに従ってハイパーバイザ５６０によって作成されたｖｖｏｌのブロック・デバイス・インスタンスにＩＯを発行する。ブロック・デバイス名と、ｖｖｏｌ　ＩＤとの間における関連付けは、ブロック・デバイス・データベース５８０内に保持される。ＶＭ５７１$_2$～５７１$_N$からのＩＯは、ＳＣＳＩ仮想化レイヤ５６３によって受信され、ＳＣＳＩ仮想化レイヤ５６３は、それらのＩＯを、仮想マシン・ファイル・システム（ＶＭＦＳ）ドライバ５６４によって理解されるファイルＩＯへと変換する。次いで、ＶＭＦＳドライバ５６４は、それらのファイルＩＯをブロックＩＯに変換し、それらのブロックＩＯを仮想ボリューム・デバイス・ドライバ５６５に提供する。その一方で、ＶＭ５７１$_1$からのＩＯは、ＶＭＦＳドライバ５６４を迂回するように示されており、仮想ボリューム・デバイス・ドライバ５６５に直接提供され、これが意味するのは、ＶＭ５７１$_1$が、自分のブロック・デバイスに直接、ロー・ストレージ・デバイスとして、たとえば、データベース・ディスク、ログ・ディスク、バックアップ・アーカイブ、およびコンテンツ・リポジトリとして、米国特許第７，１５５，５５８号に記載されている様式でアクセスするということである。

【００３７】
仮想ボリューム・デバイス・ドライバ５６５は、ブロックＩＯを受信した場合には、ブロック・デバイス・データベース５８０にアクセスして、そのＩＯ内で指定されているブロック・デバイス名と、そのブロック・デバイス名に関連付けられているｖｖｏｌへのＩＯセッションを定義するＰＥ　ＩＤおよびＳＬＬＩＤとの間におけるマッピングを参照する。ここで示されている例においては、「ｄｂａｓｅ」および「ｌｏｇ」というブロック・デバイス名は、ＶＭ５７１$_1$に関してそれぞれ作成されたｖｖｏｌ１およびｖｖｏｌ４のブロック・デバイス・インスタンスに対応しており、「ｖｍｄｋ２」、「ｖｍｄｋｎ」および「ｓｎａｐｎ」というブロック・デバイス名は、ＶＭ５７１$_2$～５７１$_N$のうちの１つまたは複数に関してそれぞれ作成されたｖｖｏｌ１２、ｖｖｏｌ１６、およびｖｖｏｌ１７のブロック・デバイス・インスタンスに対応する。ブロック・デバイス・データベース５８０内に格納されているその他の情報としては、ブロック・デバイスがアクティブであるか否かを示すそれぞれのブロック・デバイスに関するアクティブ・ビット値と、ＣＩＦ（ｃｏｍｍａｎｄｓ－ｉｎ－ｆｌｉｇｈｔ：処理中コマンド）値とが含まれる。「１」というアクティブ・ビットは、ＩＯがブロック・デバイスに発行されることが可能であるということを意味する。「０」というアクティブ・ビットは、ブロック・デバイスが非アクティブであり、ＩＯがブロック・デバイスに発行されることは不可能であるということを意味する。ＣＩＦ値は、いくつのＩＯが処理中であるか、すなわち、発行されたが完了されていないかの表示を提供する。

【００３８】
上述のマッピングを実行することに加えて、仮想ボリューム・デバイス・ドライバ５６５は、データ・アクセス・レイヤ５６６にロー・ブロックレベルＩＯを発行する。データ・アクセス・レイヤ５６６は、コマンド・キューイングおよびスケジューリング・ポリシーをロー・ブロックレベルＩＯに適用するデバイス・アクセス・レイヤ５６７と、プロトコルに準拠したフォーマットでロー・ブロックレベルＩＯをフォーマットして、それらのロー・ブロックレベルＩＯを、帯域内パスを介してＰＥへ転送するためにＨＢＡ５５４に送信する、ＨＢＡ５５４のためのデバイス・ドライバ５６８とを含む。ＳＣＳＩプロトコルが使用される実施形態においては、ｖｖｏｌ情報は、ＳＡＭ－５（ＳＣＳＩ　Ａｒｃｈｉｔｅｃｔｕｒｅ　Ｍｏｄｅｌ－５）において指定されているように、ＳＣＳＩ　ＬＵＮデータ・フィールド（これは、８バイト構造である）内にエンコードされる。ＰＥ　ＩＤは、最初の２バイト（これは、従来はＬＵＮ　ＩＤ用に使用されている）内にエンコードされ、ｖｖｏｌ情報、とりわけＳＬＬＩＤは、残っている６バイト（の一部）を利用して、ＳＣＳＩセカンド・レベルＬＵＮ　ＩＤ内にエンコードされる。図５Ｃにおいてさらに

示されているように、データ・アクセス・レイヤ５６６はまた、エラー・ハンドリング・ユニット５６９を含み、エラー・ハンドリング・ユニット５６９は、エラー・ハンドリング・ユニット５４２と同じ様式で機能する。

【００３９】

図５Ｄは、図２Ａのストレージ・システム・クラスタの代わりに図２Ｂのストレージ・システム・クラスタとインターフェースを取るように構成されている図５Ｃのコンピュータ・システムのブロック図である。この実施形態においては、データ・アクセス・レイヤ５６６は、ＮＦＳクライアント５８５と、ＮＩＣ５５３のためのデバイス・ドライバ５８６とを含む。ＮＦＳクライアント５８５は、ブロック・デバイス名を、ＰＥ　ＩＤ（ＩＰアドレス）と、ブロック・デバイスに対応するＳＬＬＩＤ（ＮＦＳファイル・ハンドル）とにマップする。このマッピングは、図５Ｄにおいて示されているように、ブロック・デバイス・データベース５８０内に格納される。「アクティブ」および「ＣＩＦ」の列は、依然として存在するが、図５Ｄに示されているブロック・デバイス・データベース５８０においては示されていないということに留意されたい。以降で説明するように、ＮＦＳファイル・ハンドルは、ＮＦＳ内のファイル・オブジェクトを一意に識別し、一実施形態においてはバインド工程中に生成される。ＮＦＳクライアント５８５はまた、仮想ボリューム・デバイス・ドライバ５６５から受信されたロー・ブロックレベルＩＯをＮＦＳファイルベースのＩＯに変換する。次いで、ＮＩＣ５５３のためのデバイス・ドライバ５８６は、プロトコルに準拠したフォーマットでＮＦＳファイルベースのＩＯをフォーマットして、それらのＮＦＳファイルベースのＩＯを、帯域内パスを介してＰＥのうちの１つへ転送するために、ＮＦＳハンドルとともに、ＮＩＣ５５３へ送信する。

【００４０】

図５Ａ～図５Ｄにおけるコンポーネントを説明するために使用されているさまざまな用語、レイヤ、および分類は、それらの機能、または本発明の趣旨もしくは範囲から逸脱することなく、別の形で参照されることが可能であるということを認識されたい。たとえば、ＶＭＭ５６１は、ＶＭ５７１とハイパーバイザ５６０との間における別個の仮想化コンポーネントとみなされることが可能である（ハイパーバイザ５６０は、そのような概念においては、それ自体が仮想化「カーネル」コンポーネントとみなされることが可能である）。なぜなら、それぞれのインスタンス化されたＶＭごとに別々のＶＭＭが存在するためである。あるいは、それぞれのＶＭＭ５６１は、そのＶＭＭ５６１の対応する仮想マシンのコンポーネントであるとみなされることが可能である。なぜなら、そのようなＶＭＭは、その仮想マシンに関するハードウェア・エミュレーション・コンポーネントを含むためである。そのような代替概念においては、たとえば、仮想ハードウェア・プラットフォーム５７３として記載されている概念レイヤは、ＶＭＭ５６１と、およびＶＭＭ５６１内へ合併されることが可能であり、それによって、仮想ホスト・バス・アダプタ５７４は、図５Ｃおよび図５Ｄから除去される（すなわち、その仮想ホスト・バス・アダプタ５７４の機能は、ホスト・バス・アダプタ・エミュレータ５６２によって実施されるためである）。

【００４１】

図６は、本発明の一実施形態による、ｖｖｏｌを管理するために使用されるコンポーネントおよび通信パスを示すコンピュータ環境の簡略化されたブロック図である。前述したように、ＩＯプロトコル・トラフィックのための通信パスは、帯域内パスと呼ばれ、図６においては破線６０１として示されており、破線６０１は、コンピュータ・システムのデータ・アクセス・レイヤ５４０を（コンピュータ・システムにおいて提供されているＨＢＡまたはＮＩＣを通じて）、ストレージ・システム１３０において構成されている１つまたは複数のＰＥと接続している。ｖｖｏｌを管理するために使用される通信パスは、帯域外パス（前に定義したように、「帯域内」ではないパス）であり、図６においては実線６０２として示されている。本明細書に記載されている実施形態によれば、ｖｖｏｌは、管理サーバ６１０において提供されているプラグイン６１２、および／またはコンピュータ・システム１０３のそれぞれにおいて提供されているプラグイン６２２を通じて管理され

ることが可能であり、図６においては、コンピュータ・システム１０３のうちの１つのみが示されている。ストレージ・デバイス側では、管理インターフェース６２５が、ストレージ・システム・マネージャ１３１によって構成されており、管理インターフェース６２６が、ストレージ・システム・マネージャ１３２によって構成されている。加えて、管理インターフェース６２４が、分散型ストレージ・システム・マネージャ１３５によって構成されている。それぞれの管理インターフェースは、プラグイン６１２、６２２と通信する。管理コマンドの発行および取り扱いを容易にするために、特別なアプリケーション・プログラミング・インターフェース（ＡＰＩ）が開発されている。一実施形態においては、プラグイン６１２、６２２の両方が、特定のストレージ・システム・ベンダーからのストレージ・ハードウェアと通信するようにカスタマイズされているということを認識されたい。したがって、管理サーバ６１０およびコンピュータ・システム１０３は、別々のストレージ・システム・ベンダーに関するストレージ・ハードウェアと通信する場合には、別々のプラグインを採用することになる。別の実施形態においては、任意のベンダーの管理インターフェースと対話する単一のプラグインが存在することができる。これは、ストレージ・システム・マネージャが、（たとえば、コンピュータ・システムおよび／または管理サーバによって発行されているという理由で、）よく知られているインターフェースに合わせてプログラムされることを必要とすることになる。

【００４２】

管理サーバ６１０はさらに、コンピュータ・システムを管理するためのシステム・マネージャ６１１を伴って構成されている。一実施形態においては、コンピュータ・システムは、仮想マシンを実行しており、システム・マネージャ６１１は、コンピュータ・システムにおいて稼働している仮想マシンを管理する。仮想マシンを管理するシステム・マネージャ６１１の１例は、ヴイエムウェア・インコーポレイティッド社（ＶＭｗａｒｅ，　Ｉｎｃ．）によって販売されているｖＳｐｈｅｒｅ（登録商標）製品である。示されているように、システム・マネージャ６１１は、コンピュータ・システム１０３からリソース使用レポートを受信するために、およびコンピュータ・システム１０３において稼働しているアプリケーション上でさまざまな管理オペレーションを開始するために、（管理サーバ６１０およびコンピュータ・システム１０３の両方において適切なハードウェア・インターフェースを通じて、）コンピュータ・システム１０３において稼働しているホスト・デーモン（ｈｏｓｔｄ）６２１と通信する。

【００４３】

図７は、認証関連のＡＰＩを使用して図２Ａまたは図２Ｂのストレージ・システム・クラスタに対してコンピュータ・システムを認証するための方法工程の流れ図である。これらの方法工程は、コンピュータ・システムが自分のセキュア・ソケット・レイヤ（ＳＳＬ）証明書をストレージ・システムに送信することによって認証を要求したときに、開始される。工程７１０において、ストレージ・システムは、認証クレデンシャル（たとえば、ユーザ名およびパスワード）を求めるプロンプトを、認証を要求しているコンピュータ・システムに発行する。工程７１２において認証クレデンシャルを受信すると、ストレージ・システムは、工程７１４において、それらの認証クレデンシャルを、格納されているクレデンシャルと比較する。正しいクレデンシャルが提供された場合には、ストレージ・システムは、認証されたコンピュータ・システムのＳＳＬ証明書をキー・ストア内に格納する（工程７１６）。正しくないクレデンシャルが提供された場合には、ストレージ・システムは、ＳＳＬ証明書を無視して、適切なエラー・メッセージを返す（工程７１８）。認証された後に、コンピュータ・システムは、ＳＳＬリンクを介してストレージ・システムに管理コマンドを発行するためにＡＰＩを呼び出すことができ、また、ＳＳＬ証明書内に含まれている一意のコンテキストＩＤが、どのコンピュータ・システムがどのストレージ・コンテナにアクセスすることができるかを定義することなどの特定のポリシーを実施するために、ストレージ・システムによって使用される。いくつかの実施形態においては、コンピュータ・システムのコンテキストＩＤが、それらのコンピュータ・システムに与えられた許可を管理する際に使用されることが可能である。たとえば、ホスト・コンピュー

タは、ｖｖｏｌを作成することを許可されることが可能であるが、そのｖｖｏｌを削除すること、もしくはそのｖｖｏｌをスナップショットすることを許可されないことが可能であり、またはホスト・コンピュータは、ｖｖｏｌのスナップショットを作成することを許可されることが可能であるが、そのｖｖｏｌをクローンすることを許可されないことが可能である。加えて、許可は、認証されたコンピュータ・システムにログインされるユーザのユーザレベル特権に従って変わることが可能である。

【００４４】

図８は、仮想ボリューム作成ＡＰＩコマンドを使用して仮想ボリュームを作成するための方法工程の流れ図である。一実施形態においては、コンピュータ・システム１０３は、最小ＩＯＰＳおよび平均待ち時間など、特定のサイズおよびストレージ能力プロファイルを有するｖｖｏｌを作成したいという要求を自分のアプリケーションのうちの１つから工程８０２において受信した場合には、帯域外パス６０２を介してストレージ・システムに仮想ボリューム作成ＡＰＩコマンドを発行する。それに応じて、コンピュータ・システム１０３は、工程８０４において、１つのストレージ・コンテナを（コンピュータ・システム１０３および要求を行っているアプリケーションがアクセスすることを許可されていて、かつ要求に対応するのに十分な空きキャパシティーを有するストレージ・コンテナの中から）選択し、プラグイン６２２を介してストレージ・システムに仮想ボリューム作成ＡＰＩコマンドを発行する。このＡＰＩコマンドは、ストレージ・コンテナＩＤと、ｖｖｏｌサイズと、ｖｖｏｌのストレージ能力プロファイルとを含む。別の実施形態においては、このＡＰＩコマンドは、一組のキー／値のペアを含み、アプリケーションは、ストレージ・システムが、その一組のキー／値のペアを、新たに作成されたｖｖｏｌとともに格納することを必要とする。別の実施形態においては、管理サーバ６１０は、帯域外パス６０２を介してストレージ・システムに仮想ボリューム作成ＡＰＩコマンドを（プラグイン６１２を介して）発行する。

【００４５】

工程８０６において、ストレージ・システム・マネージャは、管理インターフェース（たとえば、管理インターフェース６２４、６２５、または６２６）を介して、ｖｖｏｌを生成したいという要求を受信し、コンテナ・データベース３１６内の選択されたストレージ・コンテナのメタデータ・セクションにアクセスして、コンピュータ・システム１０３とアプリケーションとを含む要求コンテキストが、その選択されたストレージ・コンテナ内にｖｖｏｌを作成するのに十分な許可を有していることを確かめる。一実施形態においては、許可レベルが十分でない場合には、エラー・メッセージがコンピュータ・システム１０３に返される。許可レベルが十分である場合には、工程８１０において、一意のｖｖｏｌ　ＩＤが生成される。次いで工程８１２において、ストレージ・システム・マネージャは、コンテナ・データベース３１６のメタデータ・セクション内のアロケーション・ビットマップをスキャンして、選択されたストレージ・コンテナの空いているパーティションを特定する。ストレージ・システム・マネージャは、選択されたストレージ・コンテナの空いているパーティションを、要求されているｖｖｏｌサイズに対応するのに十分なだけ割り当て、コンテナ・データベース３１６のストレージ・コンテナのメタデータ・セクション内のアロケーション・ビットマップを更新する。ストレージ・システム・マネージャはまた、新たなｖｖｏｌエントリーでｖｖｏｌデータベース３１４を更新する。新たなｖｖｏｌエントリーは、工程８１０において生成されたｖｖｏｌ　ＩＤと、新たに割り当てられたストレージ・コンテナ・エクステントの順序付けられたリストと、キー／値のペアとして表されている新たなｖｖｏｌのメタデータとを含む。次いで工程８１４において、ストレージ・システム・マネージャは、ｖｖｏｌ　ＩＤをコンピュータ・システム１０３へ送信する。工程８１６において、コンピュータ・システム１０３は、ｖｖｏｌ　ＩＤを、ｖｖｏｌの作成を要求したアプリケーションに関連付ける。一実施形態においては、それぞれのアプリケーションに関して、１つまたは複数のｖｖｏｌ記述子ファイルが保持され、ｖｖｏｌの作成を要求したアプリケーションに関して保持されているｖｖｏｌ記述子ファイル内にｖｖｏｌ　ＩＤが書き込まれる。

【００４６】
図２Ａおよび図２Ｂにおいて示されているように、すべてのｖｖｏｌがＰＥに接続されているわけではない。ＰＥに接続されていないｖｖｏｌは、対応するアプリケーションによって発行されたＩＯに気づかない。なぜなら、そのｖｖｏｌにはＩＯセッションが確立されていないためである。ＩＯがｖｖｏｌに発行されることが可能になる前に、そのｖｖｏｌはバインド工程を経て、その結果として、そのｖｖｏｌは特定のＰＥにバインドされることになる。ｖｖｏｌがＰＥにバインドされると、そのｖｖｏｌがそのＰＥからアンバインドされるまで、ＩＯがそのｖｖｏｌに発行されることが可能である。
【００４７】
一実施形態においては、バインド要求は、コンピュータ・システム１０３によって、バインド仮想ボリュームＡＰＩを使用して、帯域外パス６０２を介してストレージ・システムに発行される。バインド要求は、（ｖｖｏｌ　ＩＤを使用して、）バインドされることになるｖｖｏｌを識別し、それに応じてストレージ・システムは、そのｖｖｏｌを、コンピュータ・システム１０３が帯域内パスを介して接続されるＰＥにバインドする。図９Ａは、コンピュータ・システムが帯域内パスを介して接続されるＰＥをそのコンピュータ・システムが発見するための方法工程の流れ図である。ＳＣＳＩプロトコルベースのストレージ・デバイスにおいて構成されているＰＥは、標準的なＳＣＳＩコマンド、ＲＥＰＯＲＴ＿ＬＵＮＳを使用して、帯域内パスを介して発見される。ＮＦＳプロトコルベースのストレージ・デバイスにおいて構成されているＰＥは、ＡＰＩを使用して、帯域外パスを介して発見される。図９Ａの方法工程は、コンピュータ・システムによって、それぞれの接続されているストレージ・システムごとに実行される。
【００４８】
工程９１０において、コンピュータ・システムは、接続されているストレージ・システムがＳＣＳＩプロトコルベースであるか、またはＮＦＳプロトコルベースであるかを判定する。ストレージ・システムがＳＣＳＩプロトコルベースである場合には、ＳＣＳＩコマンド、ＲＥＰＯＲＴ＿ＬＵＮＳが、帯域内のコンピュータ・システムによってストレージ・システムに発行される（工程９１２）。次いで工程９１３において、コンピュータ・システムは、ストレージ・システムからの応答、とりわけ、返されるＰＥ　ＩＤのそれぞれに関連付けられているＰＥビットを調べて、ＰＥ関連のＬＵＮと、従来のデータＬＵＮとの間における区別を行う。ストレージ・システムがＮＦＳプロトコルベースである場合には、利用可能なＰＥのＩＤを得るために、ＡＰＩコールが、帯域外のコンピュータ・システムによって、プラグイン６２２から管理インターフェース（たとえば、管理インターフェース６２４、６２５、または６２６）に発行される（工程９１４）。工程９１３および９１４の後に続く工程９１６において、コンピュータ・システムは、ストレージ・システムによって返されたＰＥ関連のＬＵＮのＰＥ　ＩＤ、または管理インターフェースによって返されたＰＥ　ＩＤを、バインド工程中に使用するために格納する。ＳＣＳＩプロトコルベースのストレージ・デバイスによって返されたＰＥ　ＩＤはそれぞれ、ＷＷＮを含み、ＮＦＳプロトコルベースのストレージ・デバイスによって返されたＰＥ　ＩＤはそれぞれ、ＩＰアドレスおよびマウント・ポイントを含むということを認識されたい。
【００４９】
図９Ｂは、所与のコンピュータ・システム１０３が帯域内パスを介して接続されるＰＥを、ストレージ・システム・マネージャ１３１、またはストレージ・システム・マネージャ１３２、または分散型ストレージ・システム・マネージャ１３５（以降では、「ストレージ・システム・マネージャ」と呼ばれる）が発見するための方法工程の流れ図である。そのようなＰＥをストレージ・システム・マネージャによって発見することにより、ストレージ・システムは、コンピュータ・システムからのバインド要求に応答して、要求を行っているコンピュータ・システムに、有効なＰＥ　ＩＤを返すことができ、コンピュータ・システムは、そのＰＥ　ＩＤ上に実際に接続されることが可能である。工程９５０において、ストレージ・システム・マネージャは、管理インターフェースおよびプラグイン６２２を介してコンピュータ・システム１０３に帯域外の「Ｄｉｓｃｏｖｅｒ＿Ｔｏｐｏｌ

ｏｇｙ」ＡＰＩコールを発行する。コンピュータ・システム１０３は、自分のシステムＩＤと、図９Ａの流れ図を介して自分が発見したすべてのＰＥ　ＩＤのリストとを返す。一実施形態においては、ストレージ・システム・マネージャは、管理インターフェースおよびプラグイン６１２を介して管理サーバ６１０に「Ｄｉｓｃｏｖｅｒ＿Ｔｏｐｏｌｏｇｙ」ＡＰＩコールを発行することによって、工程９５０を実行する。そのような一実施形態においては、ストレージ・システムは、複数のコンピュータ・システムＩＤと、関連付けられているＰＥ　ＩＤとを含む応答を、管理サーバ６１０が管理するそれぞれのコンピュータ・システム１０３ごとに１つ受信することになる。次いで工程９５２において、ストレージ・システム・マネージャは、工程９５０からの結果を処理する。たとえば、ストレージ・システム・マネージャは、自分の現在のコントロール下にないすべてのＰＥ　ＩＤのリストを消去する。たとえば、Ｄｉｓｃｏｖｅｒ＿Ｔｏｐｏｌｏｇｙコールを発行しているときにストレージ・システム・マネージャ１３５によって受信される特定のＰＥ　ＩＤは、同じコンピュータ・システムに接続されている別のストレージ・システムに対応している可能性がある。同様に、受信される特定のＰＥ　ＩＤは、その後にストレージ・システム管理者によって削除されたさらに古いＰＥに対応している可能性がある、といった具合である。工程９５４において、ストレージ・システム・マネージャは、処理された結果を、その後のバインド要求中に使用するためにキャッシュする。一実施形態においては、ストレージ・システム・マネージャは、図９Ｂの工程を定期的に実行して、コンピュータ・システムおよびネットワーク・トポロジーの進行中の変化で自分のキャッシュされた結果を更新する。別の実施形態においては、ストレージ・システム・マネージャは、新たなｖｖｏｌ作成要求を受信するたびに、図９Ｂの工程を実行する。さらに別の実施形態においては、ストレージ・システム・マネージャは、図７の認証工程を実行した後に、図９Ｂの工程を実行する。

【００５０】

図１０は、バインド仮想ボリュームＡＰＩを使用して仮想ボリューム・バインド要求を発行および実行するための方法工程の流れ図である。一実施形態においては、コンピュータ・システム１０３は、自分のアプリケーションのうちの１つが、まだＰＥにバインドされていないｖｖｏｌに関連付けられているブロック・デバイスへのＩＯアクセスを要求した場合には、帯域外パス６０２を介してストレージ・システムにバインド要求を発行する。別の実施形態においては、管理サーバ６１０は、ＶＭのパワー・オン、および１つのストレージ・コンテナから別のストレージ・コンテナへのｖｖｏｌの移行を含む特定のＶＭ管理オペレーションに関連したバインド要求を発行する。

【００５１】

まだＰＥにバインドされていないｖｖｏｌに関連付けられているブロック・デバイスへのＩＯアクセスをアプリケーションが要求している上述の例について続けると、コンピュータ・システム１０３は、工程１００２において、ブロック・デバイス・データベース５３３（または５８０）から、そのｖｖｏｌのｖｖｏｌ　ＩＤを特定する。次いで工程１００４において、コンピュータ・システム１０３は、そのｖｖｏｌをバインドしたいという要求を、帯域外パス６０２を通じてストレージ・システムに発行する。

【００５２】

ストレージ・システム・マネージャは、工程１００６において、管理インターフェース（たとえば、管理インターフェース６２４、６２５、または６２６）を介して、ｖｖｏｌをバインドしたいという要求を受信し、次いで、ｖｖｏｌがバインドされることになるＰＥを選択すること、選択されたＰＥに関するＳＬＬＩＤおよび世代番号を生成すること、ならびに（たとえば、ＩＯマネージャ３０４を介して）接続データベース３１２を更新することを含む工程１００８を実行する。ｖｖｏｌがバインドされることになるＰＥの選択は、接続（すなわち、コンピュータ・システム１０３への既存の帯域内接続を有するＰＥのみが、選択に利用可能である）と、利用可能なＰＥを通る現在のＩＯトラフィックなどのその他の要因とに従って行われる。一実施形態においては、ストレージ・システムは、図９Ｂの方法に従ってコンピュータ・システム１０３がそのストレージ・システムに送信

したＰＥの処理がなされてキャッシュされたリストから選択を行う。ＳＬＬＩＤの生成は、図２Ａのストレージ・システム・クラスタを採用している実施形態と、図２Ｂのストレージ・システム・クラスタを採用している実施形態との間において異なる。前者のケースにおいては、選択されたＰＥに関して一意であるＳＬＬＩＤが生成される。後者のケースにおいては、ｖｖｏｌに対応するファイル・オブジェクトへのファイル・パスが、ＳＬＬＩＤとして生成される。選択されたＰＥに関してＳＬＬＩＤおよび世代番号が生成された後に、接続データベース３１２は、ｖｖｏｌに対する新たに生成されたＩＯセッションを含むように更新される。次いで工程１０１０において、選択されたＰＥのＩＤ、生成されたＳＬＬＩＤ、および世代番号が、コンピュータ・システム１０３に返される。任意選択で、図２Ｂのストレージ・システム・クラスタを採用している実施形態においては、一意のＮＦＳファイル・ハンドルが、ｖｖｏｌに対応するファイル・オブジェクトに関して生成され、選択されたＰＥのＩＤ、生成されたＳＬＬＩＤ、および世代番号とともにコンピュータ・システム１０３に返されることが可能である。工程１０１２において、コンピュータ・システム１０３は、ストレージ・システムから返されたＰＥ　ＩＤ、ＳＬＬＩＤ（および任意選択で、ＮＦＳハンドル）、および世代番号を含めるようにブロック・デバイス・データベース５３３（または５８０）を更新する。とりわけ、ストレージ・システムから返されたＰＥ　ＩＤ、ＳＬＬＩＤ（および任意選択で、ＮＦＳハンドル）、および各組の世代番号は、新たなエントリーとしてブロック・デバイス・データベース５３３（または５８０）に加えられることになる。世代番号は、リプレイ攻撃を防ぐために使用されるということを認識されたい。したがって、リプレイ攻撃が懸念されない実施形態においては、世代番号は使用されない。

【００５３】

同じｖｖｏｌにＩＯを発行したいと望む別のアプリケーションによって開始された同じｖｖｏｌへのその後のバインド要求に対して、ストレージ・システム・マネージャは、そのｖｖｏｌを同じまたは別のＰＥにバインドすることができる。そのｖｖｏｌが同じＰＥにバインドされる場合には、ストレージ・システム・マネージャは、その同じＰＥのＩＤと、以前に生成されたＳＬＬＩＤとを返し、接続データベース３１２内に格納されているこのＩＯ接続パスのリファレンス・カウントをインクリメントする。その一方で、そのｖｖｏｌが別のＰＥにバインドされる場合には、ストレージ・システム・マネージャは、新たなＳＬＬＩＤを生成し、その別のＰＥのＩＤと、新たに生成されたＳＬＬＩＤとを返し、そのｖｖｏｌへのこの新たなＩＯ接続パスを新たなエントリーとして接続データベース３１２に加える。

【００５４】

アンバインド仮想ボリュームＡＰＩを使用して、仮想ボリューム・アンバインド要求が発行されることが可能である。アンバインド要求は、ｖｖｏｌがそれまでにバインドされた際に経由したＩＯ接続パスのＰＥ　ＩＤおよびＳＬＬＩＤを含む。しかしながら、アンバインド要求の処理は、助言的なものである。ストレージ・システム・マネージャが、すぐに、または遅れてｖｖｏｌをＰＥからアンバインドするのは自由である。アンバインド要求は、ＰＥ　ＩＤおよびＳＬＬＩＤを含むエントリーのリファレンス・カウントをデクリメントするように接続データベース３１２を更新することによって、処理される。リファレンス・カウントがゼロまでデクリメントされた場合には、そのエントリーは、削除されることが可能である。このケースにおいては、そのｖｖｏｌは、引き続き存在するが、その所与のＰＥ　ＩＤおよびＳＬＬＩＤを使用したＩＯにそれ以上利用することはできないということに留意されたい。

【００５５】

ＶＭの仮想ディスクを実装しているｖｖｏｌのケースにおいては、このｖｖｏｌに関するリファレンス・カウントは、少なくとも１となる。ＶＭがパワー・オフされ、それに関連してアンバインド要求が発行された場合には、リファレンス・カウントは、１だけデクリメントされることになる。リファレンス・カウントがゼロである場合には、ｖｖｏｌエントリーは、接続データベース３１２から除去されることが可能である。一般に、エント

リーを接続データベース３１２から除去することは有益である。なぜなら、Ｉ／Ｏマネージャ３０４は、より少ないデータを管理し、ＳＬＬＩＤを再利用することもできるためである。そのような利点は、ストレージ・システムによって格納されているｖｖｏｌの総数が多い（たとえば、数百万個程度のｖｖｏｌ）が、アプリケーションによってアクティブにアクセスされているｖｖｏｌの総数が少ない（たとえば、数万個のＶＭ）場合に、顕著になる。加えて、ｖｖｏｌがいずれのＰＥにもバインドされていない場合には、ストレージ・システムは、そのｖｖｏｌをＤＳＵ１４１内のどこに格納するかを選択する際に、より大きな柔軟性を有する。たとえば、ストレージ・システムは、いくつかのＤＳＵ１４１が、より高速なデータ・アクセスを提供し、その他のＤＳＵ１４１が、（たとえば、ストレージ・コストを節約するために）より低速なデータ・アクセスを提供する、非対称的な、階層的なＤＳＵ１４１を伴って実装されることが可能である。１実施態様においては、ｖｖｏｌがいずれのＰＥにもバインドされていない場合には（これは、接続データベース３１２内のそのｖｖｏｌのエントリーのリファレンス・カウントをチェックすることによって判定されることが可能である）、ストレージ・システムは、そのｖｖｏｌを、より低速なおよび／またはより安価なタイプの物理ストレージに移行させることができる。次いで、そのｖｖｏｌがＰＥにバインドされると、ストレージ・システムは、そのｖｖｏｌを、より高速なタイプの物理ストレージに移行させることができる。そのような移行は、ｖｖｏｌデータベース３１４内の所与のｖｖｏｌを構成するコンテナ・ロケーションの順序付けられたリストの１つまたは複数の要素を変更すること、およびコンテナ・データベース３１６のメタデータ・セクション内の対応するエクステント・アロケーション・ビットマップを更新することによって達成されることが可能であるということを認識されたい。

【００５６】

ｖｖｏｌをＰＥにバインドすることおよびアンバインドすることによって、ストレージ・システム・マネージャは、ｖｖｏｌの活性を決定することができる。ストレージ・システム・マネージャは、この情報を利用して、ＩＯサービスを提供しない（パッシブな）ｖｖｏｌおよびＩＯサービスを提供する（アクティブな）ｖｖｏｌに関してストレージ・システム・ベンダー固有の最適化を実行することができる。たとえば、ストレージ・システム・マネージャは、ｖｖｏｌが、特定のしきい値の時間を超えてパッシブな状態にとどまっている場合には、そのｖｖｏｌを、待ち時間が少ない（高いコストの）ＳＳＤから、待ち時間が中程度の（低いコストの）ハード・ドライブへ移動させるように構成されることが可能である。

【００５７】

図１１Ａおよび図１１Ｂは、一実施形態による、仮想ボリュームにＩＯを発行するための方法工程の流れ図である。図１１Ａは、アプリケーションからのＩＯを直接ロー・ブロック・デバイスに発行するための方法工程１１００の流れ図であり、図１１Ｂは、アプリケーションからのＩＯを、ファイル・システム・ドライバを通じて発行するための方法工程１１２０の流れ図である。

【００５８】

方法１１００は、工程１１０２において開始し、工程１１０２では、図５Ａ～図５Ｂにおいて示されているアプリケーション５１２、または図５Ｃ～図５Ｄにおいて示されているＶＭ５７１などのアプリケーションが、ロー・ブロック・デバイスにＩＯを発行する。工程１１０４において、仮想ボリューム・デバイス・ドライバ５３２または５６５が、アプリケーションによって発行されたＩＯからロー・ブロックレベルＩＯを生成する。工程１１０６において、ロー・ブロック・デバイスの名前が、仮想ボリューム・デバイス・ドライバ５３２または５６５によってＰＥ　ＩＤおよびＳＬＬＩＤに（そしてまた、図２Ｂのストレージ・デバイスを採用している実施形態においては、ＮＦＳクライアント５４５または５８５によってＮＦＳハンドルに）変換される。工程１１０８において、データ・アクセス・レイヤ５４０または５６６が、ＰＥ　ＩＤおよびＳＬＬＩＤを（そしてまた、図２Ｂのストレージ・デバイスを採用している実施形態においては、ＮＦＳハンドルを）ロー・ブロックレベルＩＯへとエンコードすることを実行する。次いで工程１１１０にお

いて、ＨＢＡ／ＮＩＣが、ロー・ブロックレベルＩＯを発行する。

【００５９】

図５Ａ～図５Ｂにおいて示されているアプリケーション５１２など、ＶＭ以外のアプリケーションに関しては、方法１１２０は、工程１１２１において開始する。工程１１２１においては、アプリケーションが、ｖｖｏｌベースのブロック・デバイス上に格納されているファイルにＩＯを発行する。次いで工程１１２２において、ファイル・システム・ドライバ、たとえばファイル・システム・ドライバ５１０が、ファイルＩＯからブロックレベルＩＯを生成する。工程１１２２の後には、工程１１２６、１１２８、および１１３０（これらは、工程１１０６、１１０８、および１１１０と同じである）が実行される。

【００６０】

図５Ｃ～図５Ｄにおいて示されているＶＭ５７１など、ＶＭアプリケーションに関しては、方法１１２０は、工程１１２３において開始する。工程１１２３においては、ＶＭが、自分の仮想ディスクにＩＯを発行する。次いで工程１１２４において、このＩＯは、たとえばＳＣＳＩ仮想化レイヤ５６３によって、ファイルＩＯに変換される。次いで、ファイル・システム・ドライバ、たとえばＶＭＦＳドライバ５６４が、工程１１２５において、ファイルＩＯからブロックレベルＩＯを生成する。工程１１２５の後には、工程１１２６、１１２８、および１１３０（これらは、工程１１０６、１１０８、および１１１０と同じである）が実行される。

【００６１】

図１２は、一実施形態による、ストレージ・システムにおいてＩＯを実行するための方法工程の流れ図である。工程１２１０において、コンピュータ・システムによって発行されたＩＯが、ストレージ・システムにおいて構成されているＰＥのうちの１つを通じて受信される。そのＩＯは、工程１２１２においてＩＯマネージャ３０４によって解析される。工程１２１２の後には、ストレージ・システム・クラスタが、図２Ａにおいて示されているタイプのものである場合には、ＩＯマネージャ３０４によって工程１２１４ａが実行され、ストレージ・システム・クラスタが、図２Ｂにおいて示されているタイプのものである場合には、ＩＯマネージャ３０４によって工程１２１４ｂが実行される。工程１２１４ａにおいては、ＩＯマネージャ３０４は、解析されたＩＯからＳＬＬＩＤを抽出し、接続データベース３１２にアクセスして、ＰＥ　ＩＤと、抽出されたＳＬＬＩＤとに対応するｖｖｏｌ　ＩＤを特定する。工程１２１４ｂにおいては、ＩＯマネージャ３０４は、解析されたＩＯからＮＦＳハンドルを抽出し、ＰＥ　ＩＤと、ＳＬＬＩＤとしてのＮＦＳハンドルとを使用して、ｖｖｏｌを識別する。工程１２１４ａおよび１２１４ｂの後には、工程１２１６が実行される。工程１２１６においては、ＩＯが実行されることになる物理ストレージ・ロケーションを得るために、ｖｖｏｌデータベース３１４およびコンテナ・データベース３１６が、それぞれボリューム・マネージャ３０６およびコンテナ・マネージャ３０８によってアクセスされる。次いで工程１２１８において、データ・アクセス・レイヤ３１０が、工程１２１６において得られた物理ストレージ・ロケーション上でＩＯを実行する。

【００６２】

いくつかの状況においては、アプリケーション（アプリケーション５１２またはＶＭ５７１）、管理サーバ６１０、および／またはストレージ・システム・マネージャは、特定のＰＥに対するあるｖｖｏｌのバインディングが、問題（そのＰＥが、あまりにも多くのバインディングでオーバーロード状態になっている場合など）を経験していると判定する場合がある。そのような問題を解決するための方法として、バインドされているｖｖｏｌは、ＩＯコマンドがそのｖｖｏｌへ導かれている間でさえ、ストレージ・システム・マネージャによって別のＰＥに再バインドされることが可能である。図１３は、再バインドＡＰＩを使用した、一実施形態による、ｖｖｏｌ再バインド要求を発行および実行するための方法工程１３００の流れ図である。

【００６３】

示されているように、方法１３００は、工程１３０２において開始し、工程１３０２で

は、ストレージ・システム・マネージャは、ｖｖｏｌが、そのｖｖｏｌが現在バインドされている第１のＰＥとは異なる第２のＰＥにバインドされるべきであると判定する。工程１３０４において、ストレージ・システム・マネージャは、ｖｖｏｌを再バインドしたいという要求を、ｖｖｏｌにＩＯを発行するアプリケーションを実行しているコンピュータ・システム（たとえば、コンピュータ・システム１０３）に、帯域外パスを介して発行する。工程１３０６において、コンピュータ・システム１０３は、ストレージ・システム・マネージャから再バインド要求を受信し、それに応じて、ｖｖｏｌを新たなＰＥにバインドしたいという要求を発行する。工程１３０８において、ストレージ・システム・マネージャは、再バインド要求を受信し、それに応じて、ｖｖｏｌを新たなＰＥにバインドする。工程１３１０において、ストレージ・システム・マネージャは、図１０に関連して上述したように、ｖｖｏｌが現在やはりバインドされている新たなＰＥのＩＤと、ｖｖｏｌにアクセスするためのＳＬＬＩＤとをコンピュータ・システムに送信する。

【００６４】

工程１３１２において、コンピュータ・システムは、ストレージ・システム・マネージャから新たなＰＥ　ＩＤおよびＳＬＬＩＤを受信する。ブロック・デバイス・データベース５３３または５８０において、新たなＰＥ接続のアクティブ・ビットが、はじめは１に設定され、これが意味するのは、新たなＰＥを介したｖｖｏｌのための新たなＩＯセッションが確立されたということである。コンピュータ・システムはまた、第１のＰＥ接続のアクティブ・ビットを０に設定し、これが意味するのは、このＰＥ接続を通じてそのｖｖｏｌにそれ以上ＩＯが発行されることは不可能であるということである。認識されたいこととして、このＰＥ接続は、非アクティブ化された際にすぐにアンバインドされるべきではない。なぜなら、処理中である、すなわち、発行されたが完了されていない可能性がある、このＰＥ接続を通じたそのｖｖｏｌへのＩＯが存在する可能性があるためである。したがって、工程１３１４において、コンピュータ・システムは、ブロック・デバイス・データベース５３３または５８０にアクセスして、第１のＰＥ接続を通じてそのｖｖｏｌに発行されたすべての「処理中コマンド」（ＣＩＦ）が完了されているか、すなわち、ＣＩＦ＝０であるかを確かめる。コンピュータ・システムは、工程１３１８を実行する前に、ＣＩＦがゼロになるのを待つ。その間に、そのｖｖｏｌへのさらなるＩＯが、新たなＰＥを通じて発行される。なぜなら、新たなＰＥ接続のアクティブ・ビットが既に１に設定されているためである。ＣＩＦがゼロに達しない場合には、工程１３１８が実行され、工程１３１８では、第１のＰＥ接続をアンバインドしたいという要求が、ストレージ・システム・マネージャに発行される。次いで工程１３２０において、ストレージ・システム・マネージャは、そのｖｖｏｌを第１のＰＥからアンバインドする。また、コンピュータ・システムは、工程１３２４において、新たなＰＥを通じてそのｖｖｏｌにさらなるＩＯをすべて発行する。

【００６５】

図１４は、一実施形態による、仮想ボリュームのライフ・サイクルの概念図である。図１４において示されているすべてのコマンド、すなわち、作成、スナップショット、クローン、バインド、アンバインド、拡張、および削除は、ｖｖｏｌ管理コマンド・セットを形成しており、図６に関連して上述したプラグイン６１２、６２２を通じてアクセス可能である。示されているように、「ｖｖｏｌを作成する」、「ｖｖｏｌをスナップショットする」、または「ｖｖｏｌをクローンする」というコマンドのうちのいずれかの結果としてｖｖｏｌが生成された場合には、その生成されたｖｖｏｌは、「パッシブな」状態にとどまり、パッシブな状態では、そのｖｖｏｌは、特定のＰＥにバインドされておらず、したがってＩＯを受信することはできない。加えて、ｖｖｏｌがパッシブな状態にあるときに、「ｖｖｏｌをスナップショットする」、「ｖｖｏｌをクローンする」、または「ｖｖｏｌを拡張する」というコマンドのうちのいずれかが実行された場合には、オリジナルのｖｖｏｌおよび（もしあれば）新たに作成されたｖｖｏｌは、パッシブな状態にとどまる。やはり示されているように、パッシブな状態にあるｖｖｏｌがＰＥにバインドされた場合には、そのｖｖｏｌは、「アクティブな」状態に入る。逆に、アクティブなｖｖｏｌが

ＰＥからアンバインドされた場合には、そのｖｖｏｌがいずれのさらなるＰＥにもバインドされていないと仮定すると、そのｖｖｏｌは、パッシブな状態に入る。ｖｖｏｌがアクティブな状態にあるときに、「ｖｖｏｌをスナップショットする」、「ｖｖｏｌをクローンする」、「ｖｖｏｌを拡張する」、または「ｖｖｏｌを再バインドする」というコマンドのうちのいずれかが実行された場合には、オリジナルのｖｖｏｌは、アクティブな状態にとどまり、（もしあれば）新たに作成されたｖｖｏｌは、パッシブな状態にとどまる。

【００６６】

上述したように、１つのＶＭは、複数の仮想ディスクを有することができ、それぞれの仮想ディスクごとに別々のｖｖｏｌが作成される。ＶＭはまた、そのＶＭの構成について記述するメタデータ・ファイルを有する。それらのメタデータ・ファイルとしては、ＶＭ構成ファイル、ＶＭログ・ファイル、ディスク記述子ファイル、すなわち、ＶＭのための仮想ディスクのそれぞれに関するファイル、ＶＭスワップ・ファイルなどが含まれる。仮想ディスクに関するディスク記述子ファイルは、仮想ディスクに関連する情報、たとえば、その仮想ディスクのｖｖｏｌ　ＩＤ、その仮想ディスクのサイズ、その仮想ディスクがシン・プロビジョニングされているかどうか、および、その仮想ディスクに関して作成された１つまたは複数のスナップショットのＩＤなどを含む。ＶＭスワップ・ファイルは、ストレージ・システム上におけるＶＭのスワップ・スペースを提供する。一実施形態においては、これらのＶＭ構成ファイルは、ｖｖｏｌ内に格納され、このｖｖｏｌは、本明細書においてはメタデータｖｖｏｌと呼ばれる。

【００６７】

図１５は、一実施形態による、ＶＭをプロビジョンするための方法工程の流れ図である。この実施形態においては、管理サーバ６１０と、ＶＭをホストしているコンピュータ・システム、たとえば、図５Ｃにおいて示されているコンピュータ・システム１０２（以降では、「ホスト・コンピュータ」と呼ばれる）と、図２Ａのストレージ・システム・クラスタ、とりわけストレージ・システム・マネージャ１３１、１３２、または１３５とが使用される。示されているように、ストレージ・システム・マネージャは、工程１５０２において、ＶＭをプロビジョンしたいという要求を受信する。これは、ＶＭ管理者が、管理サーバ６１０への適切なユーザ・インターフェースを使用して、特定のサイズおよびストレージ能力プロファイルを有するＶＭをプロビジョンするためのコマンドを管理サーバ６１０に発行するときに生成される要求であることが可能である。それに応じて、工程１５０４において、管理サーバ６１０は、図８に関連して上述した様式で、ＶＭのメタデータを含めるためのｖｖｏｌ（以降では、「メタデータｖｖｏｌ」と呼ばれる）を作成するための方法を開始し、それに従ってストレージ・システム・マネージャは、工程１５０８において、メタデータｖｖｏｌを作成し、そのメタデータｖｖｏｌのｖｖｏｌ　ＩＤを管理サーバ６１０に返す。工程１５１４において、管理サーバ６１０は、ＶＭをホストしているコンピュータ・システムへ戻るメタデータｖｖｏｌのｖｖｏｌ　ＩＤを登録する。工程１５１６において、ホスト・コンピュータは、図１０に関連して上述した様式で、メタデータｖｖｏｌをＰＥにバインドするための方法を開始し、それに従ってストレージ・システム・マネージャは、工程１５１８において、メタデータｖｖｏｌをＰＥにバインドし、ＰＥ　ＩＤおよびＳＬＬＩＤをホスト・コンピュータに返す。

【００６８】

工程１５２２において、ホスト・コンピュータは、ホスト・コンピュータのオペレーティング・システムへの「ＣＲＥＡＴＥ　ＤＥＶＩＣＥ」コールを使用して、メタデータｖｖｏｌのブロック・デバイス・インスタンスを作成する。次いで工程１５２４において、ホスト・コンピュータは、ブロック・デバイスの上にファイル・システム（たとえば、ＶＭＦＳ）を作成し、それに応答して、ファイル・システムＩＤ（ＦＳＩＤ）が返される。ホスト・コンピュータは、工程１５２６において、返されたＦＳＩＤを有するファイル・システムをマウントし、ＶＭのメタデータを、そのファイル・システムに関連付けられているネームスペース内に格納する。メタデータの例としては、ＶＭログ・ファイル、ディスク記述子ファイル、すなわち、ＶＭのための仮想ディスクのそれぞれに関するファイル

、およびＶＭスワップ・ファイルが含まれる。

【００６９】

工程１５２８において、ホスト・コンピュータは、図８に関連して上述した様式で、ＶＭの仮想ディスクのそれぞれに関するｖｖｏｌ（そのようなそれぞれのｖｖｏｌは、本明細書においては「データｖｖｏｌ」と呼ばれる）を作成するための方法を開始し、それに従ってストレージ・システム・マネージャは、工程１５３０において、データｖｖｏｌを作成し、そのデータｖｖｏｌのｖｖｏｌ　ＩＤをホスト・コンピュータに返す。工程１５３２において、ホスト・コンピュータは、データｖｖｏｌのＩＤを、仮想ディスクに関するディスク記述子ファイル内に格納する。この方法は、ＶＭの仮想ディスクのうちのすべてに関してデータｖｖｏｌが作成された後にメタデータｖｖｏｌがアンバインドされること（図示せず）に伴って、終了する。

【００７０】

図１６Ａは、図１５に関連して説明した様式でＶＭがプロビジョンされた後にＶＭをパワー・オンするための方法工程の流れ図である。図１６Ｂは、ＶＭがパワー・オンされた後にＶＭをパワー・オフするための方法工程の流れ図である。これらの２つの方法は、ＶＭのためのホスト・コンピュータによって実行される。

【００７１】

工程１６０８においてＶＭパワー・オン・コマンドを受信すると、そのＶＭに対応するメタデータｖｖｏｌのＩＤが、工程１６１０において取り出される。次いで工程１６１２において、メタデータｖｖｏｌは、図１０に関連して上述したようなバインド工程を経る。工程１６１４において、ファイル・システムがメタデータｖｖｏｌ上にマウントされ、それによって、工程１６１６において、データｖｖｏｌに関するメタデータ・ファイル、とりわけディスク記述子ファイルを読み取ることができ、データｖｖｏｌ　ＩＤを得ることができる。次いで工程１６１８において、データｖｖｏｌは、図１０に関連して上述したように、１つずつバインド工程を経る。

【００７２】

工程１６２０においてＶＭパワー・オフ・コマンドを受信すると、そのＶＭのデータｖｖｏｌが、ブロック・デバイス・データベース（たとえば、図５Ｃのブロック・デバイス・データベース５８０）において非アクティブとしてマークされ、ホスト・コンピュータは、それらのデータｖｖｏｌのそれぞれに関連付けられているＣＩＦがゼロに達するのを待つ（工程１６２２）。それぞれのデータｖｖｏｌに関連付けられているＣＩＦがゼロに達した際に、ホスト・コンピュータは、工程１６２４において、そのデータｖｖｏｌをアンバインドするようストレージ・システムに要求する。すべてのデータｖｖｏｌに関連付けられているＣＩＦがゼロに達した後に、工程１６２６において、メタデータｖｖｏｌが、ブロック・デバイス・データベースにおいて非アクティブとしてマークされる。次いで工程１６２８において、メタデータｖｖｏｌに関連付けられているＣＩＦがゼロに達したときに、ホスト・コンピュータは、工程１６３０において、メタデータｖｖｏｌがアンバインドされるよう要求する。

【００７３】

図１７および図１８は、ＶＭを再プロビジョンするための方法工程の流れ図である。ここで示されている例においては、図１７は、ＶＭのｖｖｏｌ、とりわけＶＭの仮想ディスクに関するデータｖｖｏｌのサイズを拡張するための、ホスト・コンピュータ上で実行される方法工程の流れ図であり、図１８は、ストレージ・コンテナ同士の間においてＶＭのｖｖｏｌを移動させるための、ストレージ・システムにおいて実行される方法工程の流れ図である。

【００７４】

ＶＭの仮想ディスクに関するデータｖｖｏｌのサイズを拡張するための方法が、工程１７０８において開始し、工程１７０８では、ホスト・コンピュータが、ＶＭがパワー・オンされているかどうかを判定する。ホスト・コンピュータは、工程１７０８において、ＶＭがパワー・オンされていないと判定した場合には、工程１７１０において、そのＶＭに

対応するメタデータｖｖｏｌのＩＤを取り出す。次いで工程１７１２において、メタデータｖｖｏｌに関するバインド工程が、ホスト・コンピュータによって開始される。バインドの後に、工程１７１４において、ホスト・コンピュータが、ファイル・システムをメタデータｖｖｏｌ上にマウントし、仮想ディスクに対応するデータｖｖｏｌのＩＤを、仮想ディスクに関するディスク記述子ファイル（これは、メタデータｖｖｏｌ上にマウントされたファイル・システム内のファイルである）から取り出す。次いで工程１７１６において、ホスト・コンピュータは、拡張ｖｖｏｌ　ＡＰＩコールを工程１７１６においてストレージ・システムへ送信し、その拡張ｖｖｏｌ　ＡＰＩコールは、データｖｖｏｌのＩＤと、データｖｖｏｌの新たなサイズとを含む。

【００７５】

ＶＭがパワー・オンされている場合には、ホスト・コンピュータは、工程１７１５において、拡張されることになるＶＭの仮想ディスクのデータｖｖｏｌのＩＤを取り出す。図１６Ａの方法から認識されたいこととして、このＩＤは、ＶＭの仮想ディスクに関連付けられているディスク記述子ファイルから入手されることが可能である。次いで工程１７１６において、ホスト・コンピュータは、拡張ｖｖｏｌ　ＡＰＩコールを工程１７１６においてストレージ・システムへ送信し、その拡張ｖｖｏｌ　ＡＰＩコールは、データｖｖｏｌのＩＤと、データｖｖｏｌの新たなサイズとを含む。

【００７６】

拡張ｖｖｏｌ　ＡＰＩコールの結果、ｖｖｏｌデータベースおよびコンテナ・データベース（たとえば、図３のｖｖｏｌデータベース３１４およびコンテナ・データベース３１６）は、ｖｖｏｌの増大されたアドレス空間を反映するようにストレージ・システムにおいて更新される。拡張ｖｖｏｌ　ＡＰＩコールが完了したという肯定応答を受信すると、ホスト・コンピュータは、工程１７１８において、ＶＭの仮想ディスクに関するディスク記述子ファイルを新たなサイズで更新する。次いで工程１７２０において、ホスト・コンピュータは、ＶＭがパワー・オンされているかどうかを判定する。ＶＭがパワー・オンされていない場合には、ホスト・コンピュータは、工程１７２２において、ファイル・システムをアンマウントし、メタデータｖｖｏｌをアンバインドしたいという要求をストレージ・システムへ送信する。その一方で、ＶＭがパワー・オンされている場合には、この方法は終了する。

【００７７】

現在ＰＥにバインドされているＶＭのｖｖｏｌをソース・ストレージ・コンテナから宛先ストレージ・コンテナへ移動させるための方法（この場合、ソース・ストレージ・コンテナおよび宛先ストレージ・コンテナの両方が、同じストレージ・システム・マネージャの範囲内にある）が、工程１８１０において開始し、工程１８１０では、ソース・ストレージ・コンテナおよび宛先ストレージ・コンテナ（それぞれ、ＳＣ１およびＳＣ２）のコンテナＩＤと、移動されることになるｖｖｏｌのｖｖｏｌ　ＩＤとが受信される。次いで工程１８１２において、ｖｖｏｌデータベース（たとえば、図３のｖｖｏｌデータベース３１４）、およびコンテナ・データベース（たとえば、図３のコンテナ・データベース３１６）のエクステント・アロケーション・ビットマップが、下記のように更新される。はじめに、ストレージ・システム・マネージャが、ＳＣ１内のｖｖｏｌエクステントをコンテナ・データベース３１６内のＳＣ１のエントリーから除去し、次いで、コンテナ・データベース３１６内のＳＣ２のエントリーを修正することによって、これらのエクステントをＳＣ２に割り振る。一実施形態においては、ストレージ・システムは、ＳＣ１における（ｖｖｏｌストレージ・エクステントの除去に起因する）ストレージ・キャパシティーの喪失を、新たなスピンドル・エクステントをＳＣ１に割り振ることによって補うこと、およびＳＣ２における（ｖｖｏｌストレージ・エクステントの追加に起因する）ストレージ・キャパシティーの増大を、いくつかの使用されていないスピンドル・エクステントをＳＣ２から除去することによって調整することが可能である。工程１８１４において、ストレージ・システム・マネージャは、現在バインドされているＰＥがｖｖｏｌの新たなロケーションにＩＯを最適にサービス提供することができるかどうかを判定する。現在のＰＥ

がｖｖｏｌの新たなロケーションにＩＯをサービス提供することができない場合の１例は、ストレージ管理者が、ストレージ・システム・マネージャを、別々のＰＥを別々の顧客ひいては別々のストレージ・コンテナからのｖｖｏｌに割り振るように静的に構成している場合である。現在のＰＥがｖｖｏｌにＩＯをサービス提供することができない場合には、ｖｖｏｌは、工程１８１５において、図１３に関連して上述した再バインド工程（および接続データベース、たとえば、図３の接続データベース３１２に対する関連する変更）を経る。工程１８１５の後には、工程１８１６が実行され、工程１８１６では、移動が首尾よく完了した旨の肯定応答が、ホスト・コンピュータに返される。工程１８１４において、現在のＰＥがｖｖｏｌの新たなロケーションにＩＯをサービス提供することができるとストレージ・システム・マネージャが判定した場合には、工程１８１５は迂回され、次いで工程１８１６が実行される。

【００７８】

互換性がないストレージ・コンテナ同士の間において、たとえば、別々の製造業者のストレージ・デバイス内に作成されたストレージ・コンテナ同士の間においてｖｖｏｌが移動される場合には、コンテナ・データベース３１６、ｖｖｏｌデータベース３１４、および接続データベース３１２に対する変更に加えて、ストレージ・コンテナ同士の間においてデータの移動が実行される。一実施形態においては、２００８年5月２９日に出願された「Ｏｆｆｌｏａｄｉｎｇ　Ｓｔｏｒａｇｅ　Ｏｐｅｒａｔｉｏｎｓ　ｔｏ　Ｓｔｏｒａｇｅ　Ｈａｒｄｗａｒｅ」と題されている米国特許出願第１２／１２９，３２３号（その全内容を本願明細書に援用する）に記載されているデータ移動技術が採用される。

【００７９】

図１９は、テンプレートＶＭからＶＭをクローンするための、ホスト・コンピュータおよびストレージ・システムにおいて実行される方法工程の流れ図である。この方法は、工程１９０８において開始し、工程１９０８では、ホスト・コンピュータが、新たなＶＭに関するメタデータｖｖｏｌを作成したいという要求をストレージ・システムへ送信する。１９１０において、ストレージ・システムは、図８に関連して上述した方法に従って新たなＶＭに関するメタデータｖｖｏｌを作成し、新たなメタデータｖｖｏｌ　ＩＤをホスト・コンピュータに返す。次いで工程１９１４において、クローンｖｖｏｌ　ＡＰＩコールが、テンプレートＶＭに属するすべてのデータｖｖｏｌ　ＩＤに関して、ホスト・コンピュータから帯域外パス６０１を介してストレージ・システムに発行される。工程１９１８において、ストレージ・システム・マネージャが、テンプレートＶＭのデータｖｖｏｌと、新たなＶＭのデータｖｖｏｌとに互換性があるか否かをチェックする。別々の製造業者のストレージ・システム内に作成されたストレージ・コンテナ同士の間においてクローニングが行われる場合には、データｖｖｏｌ同士に互換性がない可能性があるということを認識されたい。互換性がある場合には、工程１９１９が実行される。工程１９１９において、ストレージ・システム・マネージャは、新たなデータｖｖｏｌ　ＩＤを生成すること、コンテナ・データベース３１６内のアロケーション・ビットマップを更新すること、および新たなｖｖｏｌエントリーをｖｖｏｌデータベース３１４に加えることによって、新たなデータｖｖｏｌを作成し、テンプレートＶＭのデータｖｖｏｌ内に格納されているコンテンツを新たなＶＭのデータｖｖｏｌにコピーする。工程１９２０において、ストレージ・システム・マネージャは、新たなデータｖｖｏｌ　ＩＤをホスト・コンピュータに返す。新たなデータｖｖｏｌ　ＩＤを受信することは、データｖｖｏｌのクローニングがエラーなく完了した旨の確認をホスト・コンピュータに提供する。次いで工程１９２５において、ホスト・コンピュータは、メタデータ・ファイル、とりわけディスク記述子ファイルを、新たに生成されたデータｖｖｏｌ　ＩＤで更新するために、新たなＶＭのメタデータｖｖｏｌにＩＯを発行する。ホスト・コンピュータによってストレージ・システムに発行されたＩＯは、工程１９２６においてストレージ・システムによって実行され、その結果として、新たなＶＭのディスク記述子ファイルが、新たに生成されたデータｖｖｏｌ　ＩＤで更新される。

【００８０】

工程１９１８において、テンプレートＶＭのデータｖｖｏｌと、新たなＶＭのデータｖｖｏｌとに互換性がないとストレージ・システム・マネージャが判定した場合には、エラー・メッセージが、ホスト・コンピュータに返される。このエラー・メッセージを受信すると、ホスト・コンピュータは、工程１９２１において、新たなデータｖｖｏｌを作成するために、作成ｖｖｏｌ　ＡＰＩコールをストレージ・システムに発行する。工程１９２２において、ストレージ・システム・マネージャは、新たなデータｖｖｏｌ　ＩＤを生成すること、コンテナ・データベース３１６内のアロケーション・ビットマップを更新すること、および新たなｖｖｏｌエントリーをｖｖｏｌデータベース３１４に加えることによって、新たなデータｖｖｏｌを作成し、新たなデータｖｖｏｌ　ＩＤをホスト・コンピュータに返す。工程１９２３において、ホスト・コンピュータは、２００９年１月２１日に出願された「Ｄａｔａ　Ｍｏｖｅｒ　ｆｏｒ　Ｃｏｍｐｕｔｅｒ　Ｓｙｓｔｅｍ」と題されている米国特許出願第１２／３５６，６９４号（その全内容を本願明細書に援用する）に記載されている技術に従って、データの移動を実行する（工程１９２３）。工程１９２３の後には、工程１９２５および１９２６が、上述のように実行される。

【００８１】

図２０は、別の実施形態による、ＶＭをプロビジョンするための方法工程の流れ図である。この実施形態においては、管理サーバ６１０と、ＶＭをホストしているコンピュータ・システム、たとえば、図５Ｄにおいて示されているコンピュータ・システム１０２（以降では、「ホスト・コンピュータ」と呼ばれる）と、図２Ｂのストレージ・システム・クラスタ、とりわけストレージ・システム・マネージャ１３１、またはストレージ・システム・マネージャ１３２、またはストレージ・システム・マネージャ１３５とが使用される。示されているように、ＶＭをプロビジョンしたいという要求が、工程２００２において受信される。これは、ＶＭ管理者が、管理サーバ６１０への適切なユーザ・インターフェースを使用して、特定のサイズおよびストレージ能力プロファイルを有するＶＭをプロビジョンするためのコマンドを管理サーバ６１０に発行するときに生成される要求であることが可能である。それに応じて、工程２００４において、管理サーバ６１０は、図８に関連して上述した様式で、ＶＭのメタデータを含めるためのｖｖｏｌ、とりわけメタデータｖｖｏｌを作成するための方法を開始し、それに従ってストレージ・システム・マネージャは、工程２００８において、メタデータｖｖｏｌ（これは、ＮＡＳデバイス内のファイルである）を作成し、メタデータｖｖｏｌ　ＩＤを管理サーバ６１０に返す。工程２０２０において、管理サーバ６１０は、ホスト・コンピュータへ戻るメタデータｖｖｏｌのｖｖｏｌ　ＩＤを登録する。工程２０２２において、ホスト・コンピュータは、メタデータｖｖｏｌ　ＩＤに関するバインド要求をストレージ・システムに発行し、それに応答して、ストレージ・システムは、工程２０２３において、ＩＰアドレスおよびディレクトリ・パスをそれぞれＰＥ　ＩＤおよびＳＬＬＩＤとして返す。工程２０２４において、ホスト・コンピュータは、指定されたＩＰアドレスおよびディレクトリ・パスにおいてディレクトリをマウントし、そのマウントされたディレクトリ内にメタデータ・ファイルを格納する。ＮＦＳを使用する実施形態においては、ＮＦＳクライアント５４５または５８５が、そのようなディレクトリにＮＦＳ要求を発行するために、所与のＩＰアドレスおよびディレクトリ・パスをＮＦＳハンドルへと変換することができる。

【００８２】

工程２０２６において、ホスト・コンピュータは、図８に関連して上述した様式で、ＶＭの仮想ディスクのそれぞれに関するデータｖｖｏｌを作成するための方法を開始し、それに従ってストレージ・システム・マネージャは、工程２０３０において、データｖｖｏｌを作成し、そのデータｖｖｏｌのｖｖｏｌ　ＩＤをホスト・コンピュータに返す。工程２０３２において、ホスト・コンピュータは、データｖｖｏｌのＩＤを、仮想ディスクに関するディスク記述子ファイル内に格納する。この方法は、ＶＭの仮想ディスクのうちのすべてに関してデータｖｖｏｌが作成された後にメタデータｖｖｏｌがアンバインドされること（図示せず）に伴って、終了する。

【００８３】

図８に関連して上述したように、新たなｖｖｏｌがストレージ・コンテナから作成され、その新たなｖｖｏｌに関してストレージ能力プロファイルが明示的に指定されない場合には、その新たなｖｖｏｌは、ストレージ・コンテナに関連付けられているストレージ能力プロファイルを引き継ぐことになる。ストレージ・コンテナに関連付けられているストレージ能力プロファイルは、いくつかの別々のプロファイルのうちの１つから選択されることが可能である。たとえば、図２１において示されているように、それらの別々のプロファイルとしては、プロダクション（ｐｒｏｄ）プロファイル２１０１、デベロップメント（ｄｅｖ）プロファイル２１０２、およびテスト・プロファイル２１０３（ここでは「プロファイル２１００」と総称される）が含まれる。その他の多くのプロファイルが定義されることも可能であるということを認識されたい。示されているように、特定のプロファイルのそれぞれのプロファイル・エントリーは、固定タイプまたは可変タイプのものであり、１つの名前と、それに関連付けられている１つまたは複数の値とを有している。固定タイプのプロファイル・エントリーは、固定された数の選択可能なアイテムを有している。たとえば、プロファイル・エントリー「複製」は、真または偽であるように設定されることが可能である。対照的に、可変タイプのプロファイル・エントリーは、事前に定義された選択肢を有していない。その代わりに、可変タイプのプロファイル・エントリーに関しては、デフォルトの値およびある範囲の値が設定され、ユーザは、その範囲内にある任意の値を選択することができる。値がまったく指定されない場合には、デフォルトの値が使用される。図２１に示されている例示的なプロファイル２１００においては、可変タイプのプロファイル・エントリーは、コンマによって区切られている３つの数字を有している。第１の数字は、指定された範囲の下限であり、第２の数字は、指定された範囲の上限である。第３の数字は、デフォルトの値である。したがって、プロダクション・プロファイル２１０１において定義されているストレージ能力プロファイルを引き継ぐｖｖｏｌは、複製されることになり（複製。値＝真）、複製に関する目標復旧時間（ＲＴＯ：ｒｅｃｏｖｅｒｙ　ｔｉｍｅ　ｏｂｊｅｃｔｉｖｅ）は、０．１時間から２４時間の範囲内で定義されることが可能であり、デフォルトは１時間である。加えて、このｖｖｏｌに関しては、スナップショットが可能である（スナップショット。値＝真）。保持されるスナップショットの数は、１から１００の範囲内であり、デフォルトは１であり、スナップショットの頻度は、１時間に１回から２４時間に１回の範囲内であり、デフォルトは１時間に１回である。「スナップ引き継ぎ」の列は、派生ｖｖｏｌである新たなｖｖｏｌを作成するために所与のｖｖｏｌがスナップショットされる場合に所与のプロファイル属性（およびその値）が派生ｖｖｏｌに伝搬されるべきかどうかを示す。プロダクション・プロファイル２１０１の例においては、最初の２つのプロファイル・エントリー（複製およびＲＴＯ）のみが、プロダクション・プロファイル２１０１を有する所与のｖｖｏｌのスナップショットｖｖｏｌに伝搬されることが可能である。スナップショットｖｖｏｌのその他のすべての属性の値は、そのプロファイルにおいて指定されているデフォルトの値に設定されることになる。言い換えれば、所与のｖｖｏｌに関するこれらのその他の属性のあらゆるカスタマイゼーション（たとえば、スナップショット頻度のデフォルトではない値）は、それらの対応する「スナップ引き継ぎ」の列が偽であるため、スナップショットｖｖｏｌに伝搬されない。このプロファイルは、どの属性値が所与のｖｖｏｌのそれぞれクローンおよびレプリカに伝搬されるかをコントロールする「クローン引き継ぎ」（図示せず）および「レプリカ引き継ぎ」（図示せず）などのその他の列も含む。

【００８４】

図４の方法に従ってストレージ・コンテナが作成される場合には、そのストレージ・コンテナから作成されるｖｖｏｌに関して定義することができるストレージ能力プロファイルのタイプが設定されることが可能である。図２１における流れ図は、図４において示されているストレージ・コンテナを作成するための方法を、工程４１２と工程４１３との間に工程２１１０が挿入された状態で示している。工程２１１０において、ストレージ管理者は、作成されているストレージ・コンテナに関するプロファイル２１００のうちの１つまたは複数を選択する。たとえば、１人の顧客のために作成された１つのストレージ・コ

ンテナが、プロダクション・プロファイル２１０１およびデベロップメント・プロファイル２１０２に関連付けられることが可能であり、それによって、プロダクション・タイプのものであるｖｖｏｌは、場合によってデフォルトの値または顧客によって指定された値を伴うプロダクション・プロファイル２１０１において定義されているストレージ能力プロファイルを引き継ぐことになり、デベロップメント・タイプのものであるｖｖｏｌは、場合によってデフォルトの値または顧客によって指定された値を伴うデベロップメント・プロファイル２１０２において定義されているストレージ能力プロファイルを引き継ぐことになる。

【００８５】

図２２は、ｖｖｏｌを作成して、そのｖｖｏｌに関するストレージ能力プロファイルを定義するための、ストレージ・システム・マネージャ１３１、１３２、または１３５によって実行される方法工程を示す流れ図である。図２２の方法工程、とりわけ工程２２１０、２２１２、２２１８、および２２２０はそれぞれ、図８において示されている工程８０６、８１０、８１２、および８１４に対応する。加えて、図２２の方法工程は、工程２２１４、２２１５、および２２１６を含み、これらの工程は、作成されているｖｖｏｌに関するストレージ能力プロファイルを定義する。

【００８６】

工程２２１４において、ストレージ・システム・マネージャは、ストレージ能力プロファイルにおいて使用されることになる値が、ｖｖｏｌを作成したいという要求内で指定されているかどうかを判定する。ストレージ能力プロファイルにおいて使用されることになる値が指定されていない場合には、ストレージ・システム・マネージャは、工程２２１５において、ｖｖｏｌのストレージ・コンテナに関連付けられているストレージ能力プロファイルを、デフォルトの値を伴うｖｖｏｌのストレージ能力プロファイルとして採用する。ストレージ能力プロファイルにおいて使用されることになる値が指定されている場合には、ストレージ・システム・マネージャは、工程２２１６において、ｖｖｏｌのストレージ・コンテナに関連付けられているストレージ能力プロファイルを、デフォルトの値の代わりに指定されている値を伴うｖｖｏｌのストレージ能力プロファイルとして採用する。

【００８７】

一実施形態においては、ｖｖｏｌのストレージ能力プロファイルは、キー／値のペアとしてｖｖｏｌデータベース３１４内に格納される。いったんｖｖｏｌのストレージ能力プロファイルが定義されて、キー／値のペアとしてｖｖｏｌデータベース３１４内に格納されると、図２１の例示的なプロファイルにおいて示されているように複製およびスナップショット関連の属性および値がこのプロファイルの一部である限り、ストレージ・システムは、ホスト・コンピュータによって発行されるさらなる命令を伴わずに、そのｖｖｏｌに関する複製およびスナップショットを実行することができる。

【００８８】

図２３は、親ｖｖｏｌからスナップショットを作成するための、ストレージ・システム・マネージャ１３１、１３２、または１３５によって実行される方法工程を示す流れ図である。一実施形態においては、所与のｖｖｏｌのストレージ能力プロファイル内のスナップショット定義に従ってスナップショットをスケジュールするためのスナップショット・トラッキング・データ構造が採用される。スナップショットのためのスケジュールされた時刻に達すると、ストレージ・システム・マネージャは、工程２３１０において、スナップショット・トラッキング・データ構造からｖｖｏｌ　ＩＤを取り出す。次いで工程２３１２において、ストレージ・システム・マネージャは、スナップショットに関する一意のｖｖｏｌ　ＩＤを生成する。ストレージ・システム・マネージャは、工程２３１５において、親ｖｖｏｌ（すなわち、スナップショット・トラッキング・データ構造から取り出されたｖｖｏｌ　ＩＤを有するｖｖｏｌ）のストレージ能力プロファイルを、スナップショットｖｖｏｌのストレージ能力プロファイルとして採用する。これは、ストレージ・システムによって駆動される自動化されたプロファイル駆動型のスナップショット工程であるため、ユーザには、スナップショットｖｖｏｌのストレージ能力プロファイルにおいて使

用されることになるカスタムの値を指定するための機会はないということに留意されたい。工程２３１８において、ストレージ・システム・マネージャは、コンテナ・データベース３１６内のアロケーション・ビットマップを更新すること、およびスナップショットｖｖｏｌに関する新たなｖｖｏｌエントリーをｖｖｏｌデータベース３１４に加えることによって、親ｖｖｏｌのストレージ・コンテナ内にスナップショットｖｖｏｌを作成する。次いで工程２３２０において、ストレージ・システム・マネージャは、親ｖｖｏｌに関する次なるスナップショットを生成するための時刻をスケジュールすることによって、スナップショット・トラッキング・データ構造を更新する。ストレージ・システム・マネージャは、スナップショット・トラッキング・データ構造を保持することと、スケジュールされたスナップショットを命じるストレージ能力プロファイルを有するすべてのｖｖｏｌに関して図２３の方法工程を実行することとを同時に行わなければならないということを認識されたい。

【００８９】

上述の様式でスナップショットが作成された後に、ｖｖｏｌデータベース３１４内に格納されているキー／値のペアは、スナップショットｖｖｏｌがタイプ＝スナップショットのものであるということを示すように更新される。また、スナップショットに関して世代番号が保持され、その世代番号が、スナップショットがとられるたびにインクリメントされるか、または日にち＋時刻に等しくなるように設定される実施形態においては、世代番号は、キー／値のペアとして格納される。スナップショットｖｖｏｌの親ｖｖｏｌ　ＩＤも、キー／値のペアとしてスナップショットｖｖｏｌエントリー内に格納される。結果として、ホスト・コンピュータは、特定のｖｖｏｌ　ＩＤに対応するスナップショットを求めてｖｖｏｌデータベース３１４にクエリーを行うことができる。ホスト・コンピュータは、特定のｖｖｏｌ　ＩＤおよび特定の世代番号に対応するスナップショットを求めてｖｖｏｌデータベースにクエリーを発行することも可能である。

【００９０】

本明細書に記載されているさまざまな実施形態は、コンピュータ・システム内に格納されているデータを含むさまざまなコンピュータ実施オペレーションを採用することができる。たとえば、これらのオペレーションは、通常、必須ではないが、物理量の物理的な操作を必要とする場合があり、これらの量は、電気信号または磁気信号の形態を取ることができ、そうした形態では、それらの量、またはそれらの量の表示は、格納されること、転送されること、結合されること、比較されること、またはその他の形で操作されることが可能である。さらに、そのような操作はしばしば、作り出すこと、識別すること、判定すること、または比較することなどの用語で呼ばれる。１つまたは複数の実施形態の一部を形成する、本明細書に記載されているあらゆるオペレーションは、有用なマシン・オペレーションであることが可能である。加えて、１つまたは複数の実施形態はまた、これらのオペレーションを実行するためのデバイスまたは装置に関する。その装置は、特定の必要とされる目的のために特別に構築されることが可能であり、または、コンピュータ内に格納されているコンピュータ・プログラムによって選択的にアクティブ化または構成される汎用コンピュータであることが可能である。とりわけ、さまざまな汎用マシンが、本明細書における教示に従って書かれたコンピュータ・プログラムとともに使用されることが可能であり、または、必要とされるオペレーションを実行するためのさらに専門化された装置を構築することが、より好都合である可能性がある。

【００９１】

本明細書に記載されているさまざまな実施形態は、ハンドヘルド・デバイス、マイクロプロセッサ・システム、マイクロプロセッサベースのまたはプログラム可能な家庭用電化製品、ミニコンピュータ、メインフレーム・コンピュータなどを含むその他のコンピュータ・システム構成とともに実施されることが可能である。

【００９２】

１つまたは複数の実施形態は、１つもしくは複数のコンピュータ・プログラムとして、または１つもしくは複数のコンピュータ可読メディアにおいて具体化される１つもしくは

複数のコンピュータ・プログラム・モジュールとして実装されることが可能である。コンピュータ可読メディアという用語は、その後にコンピュータ・システムに入力されることが可能であるデータを格納することができる任意のデータ・ストレージ・デバイスを指し、コンピュータ可読メディアは、コンピュータ・プログラムがコンピュータによって読み取られることを可能にする様式でそれらのコンピュータ・プログラムを具体化するための任意の既存のまたはその後に開発されるテクノロジーに基づくことができる。コンピュータ可読メディアの例としては、ハード・ドライブ、ネットワーク・アタッチト・ストレージ（ＮＡＳ）、読み取り専用メモリ、ランダムアクセス・メモリ（たとえば、フラッシュ・メモリ・デバイス）、ＣＤ（Ｃｏｍｐａｃｔ　Ｄｉｓｃ）、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、またはＣＤ－ＲＷ、ＤＶＤ（Ｄｉｇｉｔａｌ　Ｖｅｒｓａｔｉｌｅ　Ｄｉｓｃ）、磁気テープ、ならびにその他の光学式および非光学式のデータ・ストレージ・デバイスが含まれる。コンピュータ可読メディアは、ネットワークに結合されたコンピュータ・システムを介して分散されることも可能であり、それによってコンピュータ可読コードは、分散された様式で格納および実行される。

【００９３】

１つまたは複数の実施形態について、理解を明確にするためにいくらか詳細に説明したが、特許請求の範囲の範疇内で特定の変更および修正が行われることが可能であるということは明らかであろう。たとえば、ＳＣＳＩが、ＳＡＮデバイスのためのプロトコルとして採用され、ＮＦＳが、ＮＡＳデバイスのためのプロトコルとして使用される。ファイバ・チャネルなど、ＳＣＳＩプロトコルに対する任意の代替手段が使用されることが可能であり、ＣＩＦＳ（Ｃｏｍｍｏｎ　Ｉｎｔｅｒｎｅｔ　Ｆｉｌｅ　Ｓｙｓｔｅｍ）プロトコルなど、ＮＦＳプロトコルに対する任意の代替手段が使用されることが可能である。したがって、記載されている実施形態は、限定的なものではなく例示的なものとみなされるべきであり、特許請求の範囲の範疇は、本明細書において与えられている詳細に限定されるものではなく、特許請求の範囲の範疇および均等物の中で修正されることが可能である。特許請求の範囲において、要素および／または工程は、特許請求の範囲において明示的に記載されていない限り、オペレーションのいかなる特定の順序も意味するものではない。

【００９４】

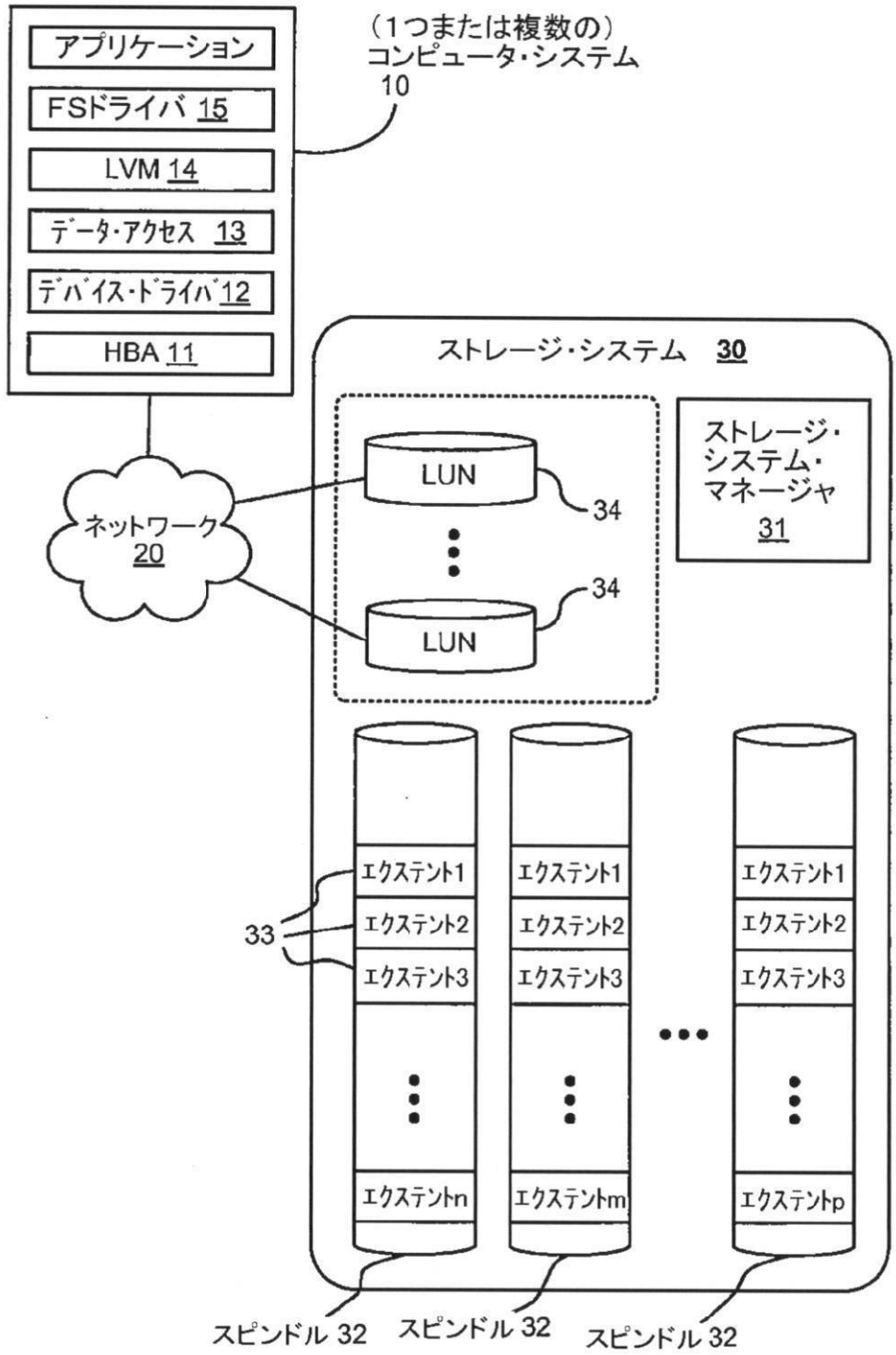
加えて、記載されている仮想化方法は一般に、仮想マシンが、特定のハードウェア・システムと整合するインターフェースを提示すると想定しているが、記載されているそれらの方法は、いかなる特定のハードウェア・システムにも直接対応しない仮想化に関連して使用されることが可能である。ホストされる実施形態、ホストされない実施形態として、またはそれらの両者の間における区別をあいまいにする傾向がある実施形態として実施される、さまざまな実施形態による仮想化システムが、すべて想定されている。さらに、さまざまな仮想化オペレーションは、全体的にまたは部分的にハードウェアで実施されることが可能である。たとえば、ハードウェアの実施態様は、非ディスク・データをセキュアにするためのストレージ・アクセス要求の修正のためのルックアップ・テーブルを採用することができる。

【００９５】

仮想化の度合いにかかわらず、多くの変形、修正、追加、および改良が可能である。したがって仮想化ソフトウェアは、仮想化機能を実行するホスト、コンソール、またはゲスト・オペレーティング・システムのコンポーネントを含むことができる。単一のインスタンスとして本明細書に記載されているコンポーネント、オペレーション、または構造のために、複数のインスタンスが提供されることが可能である。最後に、さまざまなコンポーネント、オペレーション、およびデータストア同士の間における境界は、いくらか任意のものであり、特定のオペレーションは、特定の例示的な構成のコンテキストにおいて示されている。機能のその他の割り当ても想定されており、本明細書に記載されている実施形態の範囲内に収まることができる。一般に、例示的な構成において別々のコンポーネントとして提示されている構造および機能は、結合された構造またはコンポーネントとして実装されることが可能である。同様に、単一のコンポーネントとして提示されている構造お

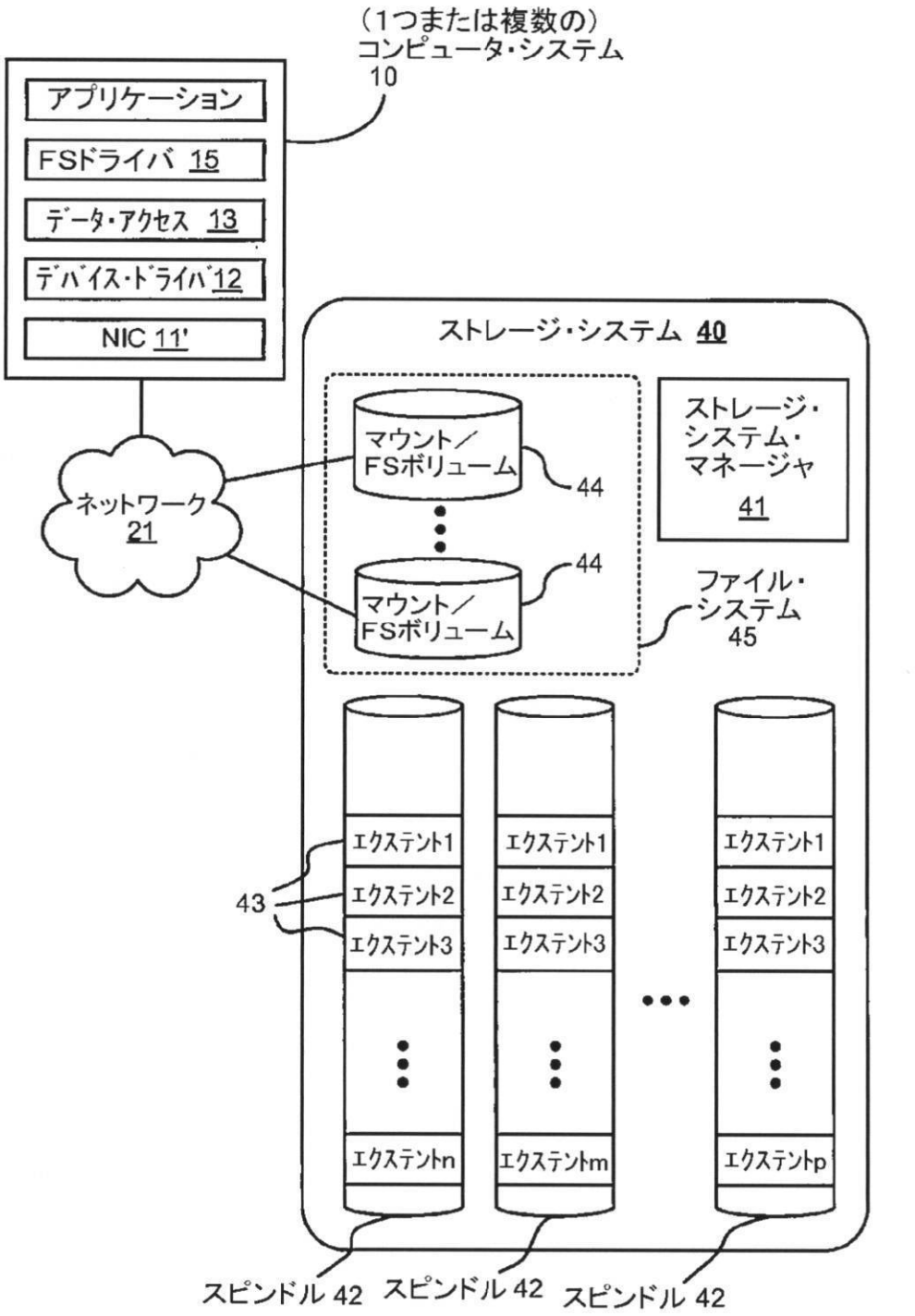
よび機能は、別々のコンポーネントとして実装されることが可能である。これらおよびその他の変形、修正、追加、および改良は、添付の（１つまたは複数の）特許請求の範囲の範疇内に収まることができる。

【図１Ａ】


アプリケーション
FSドライバ 15
LVM 14
データ・アクセス 13
デバイス・ドライバ 12
HBA 11
（1つまたは複数の）
コンピュータ・システム
10
ストレージ・システム 30
ネットワーク
20
LUN
34
LUN
34
ストレージ・
システム・
マネージャ
31
エクステント1
エクステント2
エクステント3
エクステントn
エクステント1
エクステント2
エクステント3
エクステントm
エクステント1
エクステント2
エクステント3
エクステントp
33
スピンドル 32
スピンドル 32
スピンドル 32

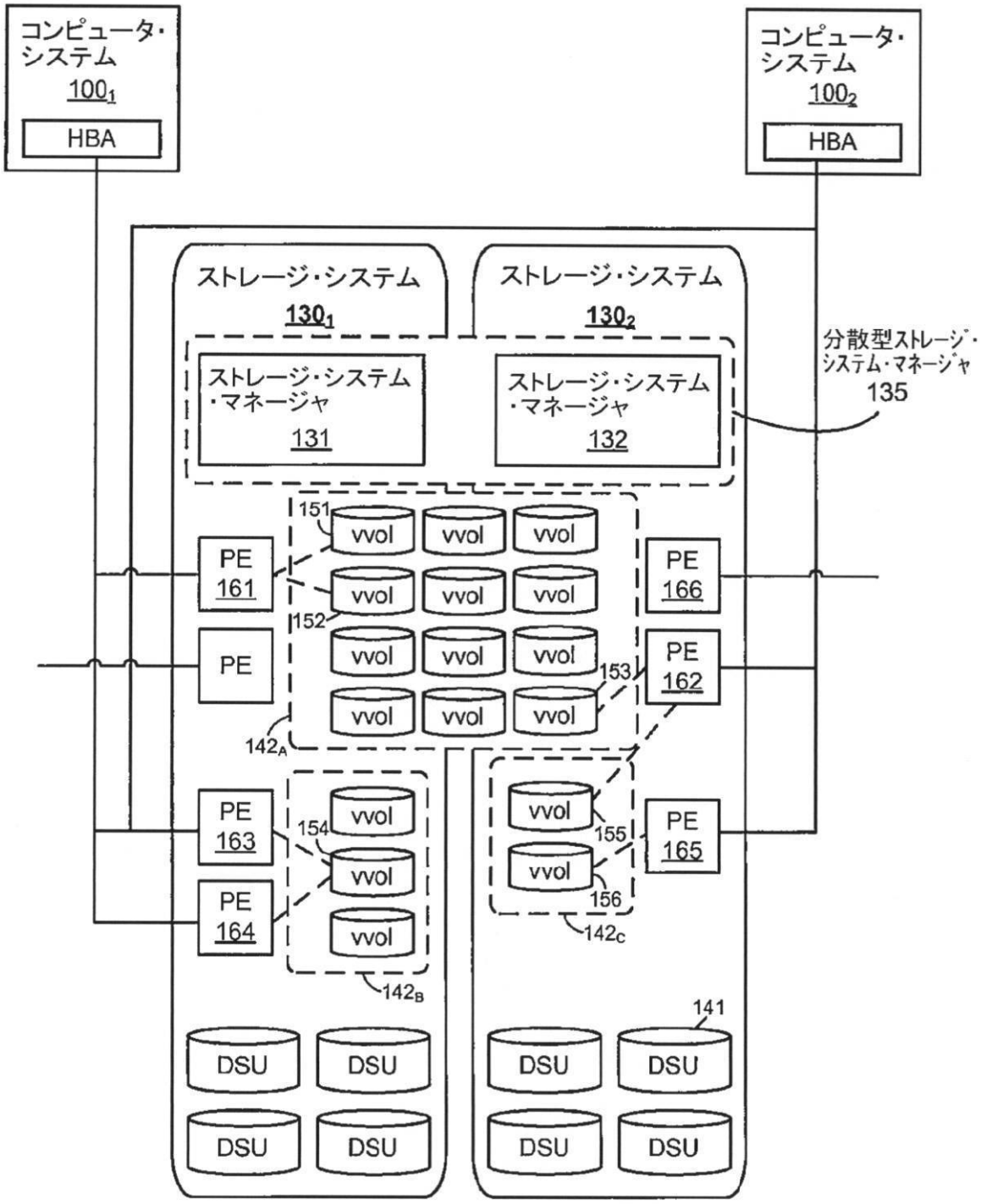
（従来技術）

【図１Ｂ】

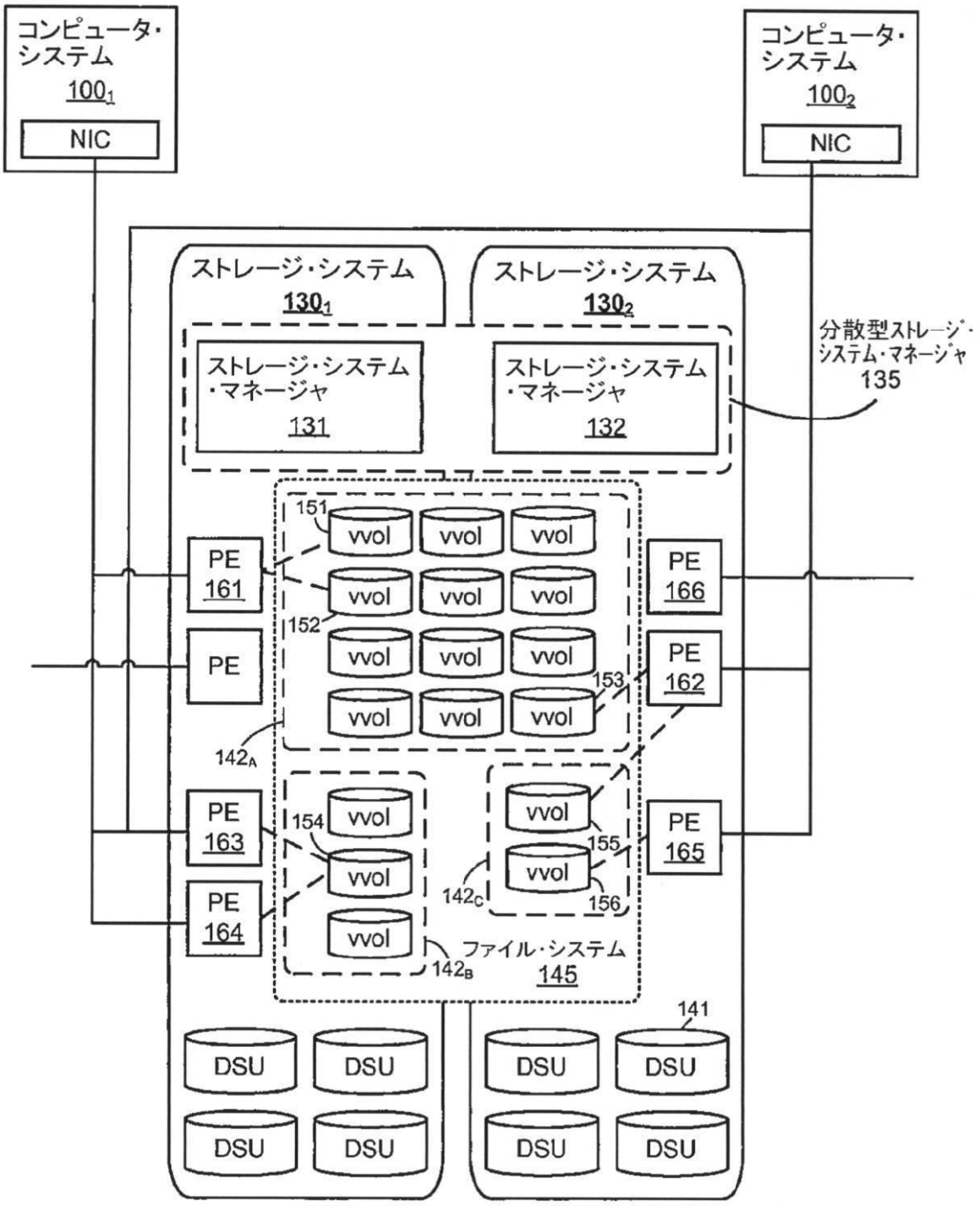

（1つまたは複数の）
コンピュータ・システム
10
アプリケーション
FSドライバ 15
データ・アクセス 13
デバイス・ドライバ 12
NIC 11'
ネットワーク
21
ストレージ・システム 40
マウント／
FSボリューム
44
マウント／
FSボリューム
44
ストレージ・
システム・
マネージャ
41
ファイル・
システム
45
43
エクステント1
エクステント2
エクステント3
エクステントn
エクステント1
エクステント2
エクステント3
エクステントm
エクステント1
エクステント2
エクステント3
エクステントp
スピンドル 42
スピンドル 42
スピンドル 42

（従来技術）

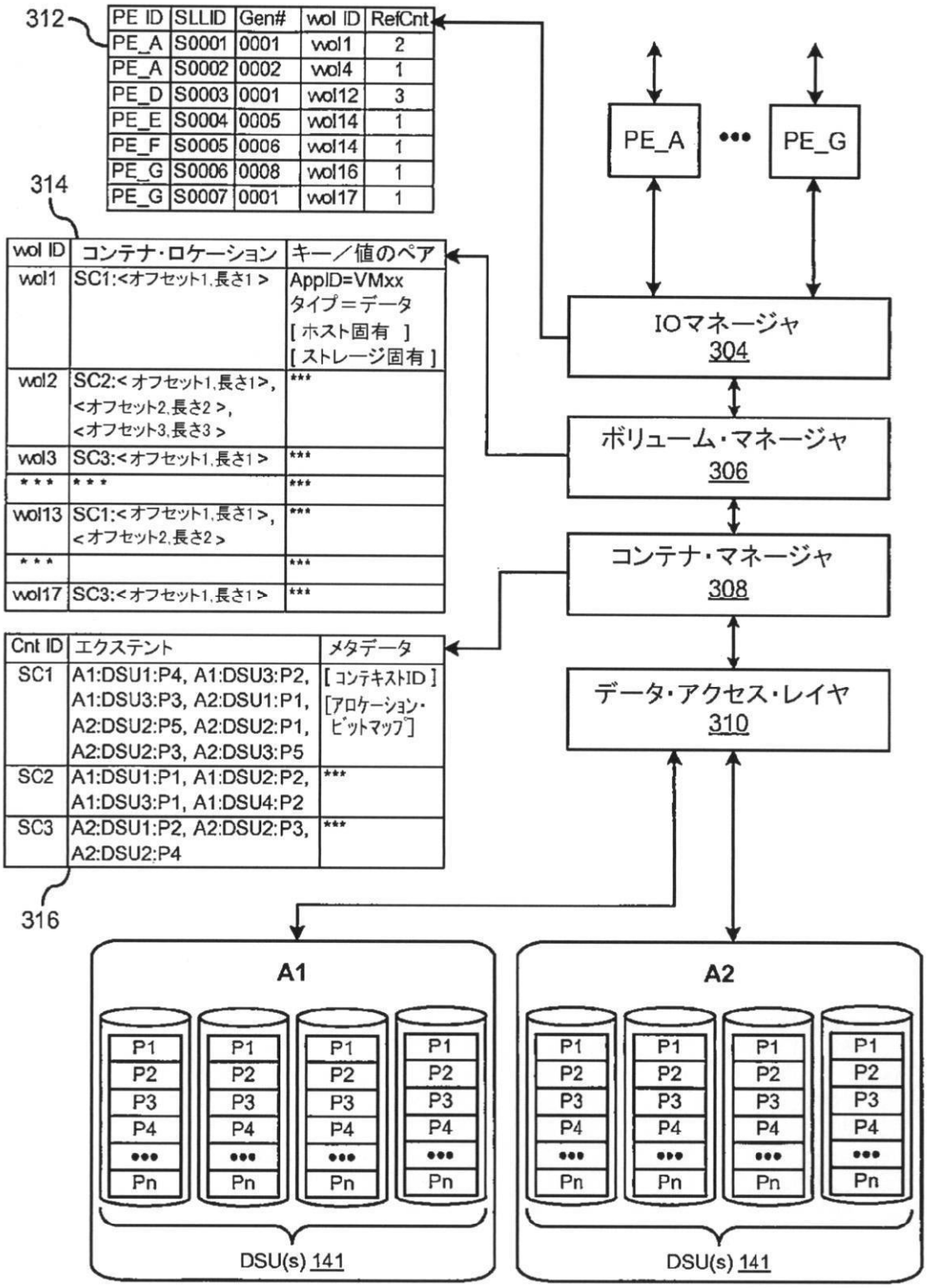
【図２Ａ】


コンピュータ・システム
100₁
HBA
コンピュータ・システム
100₂
HBA
ストレージ・システム
130₁
ストレージ・システム
130₂
ストレージ・システム・マネージャ
131
ストレージ・システム・マネージャ
132
分散型ストレージ・システム・マネージャ
135
151
152
153
154
155
156
vvol
PE
161
PE
166
PE
162
PE
163
PE
164
PE
165
142A
142B
142C
141
DSU

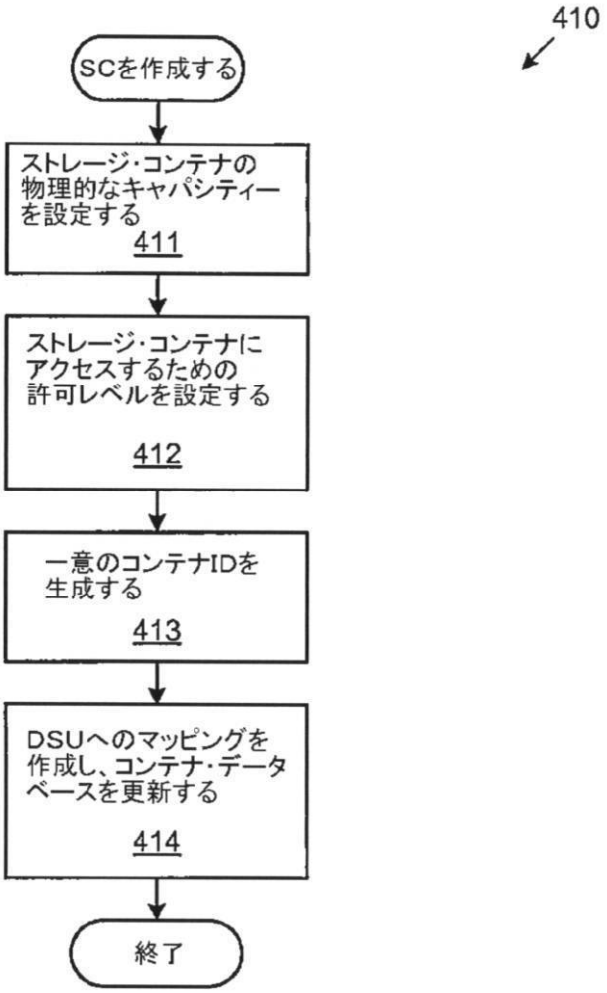
【図２Ｂ】


コンピュータ・システム
100₁
NIC
コンピュータ・システム
100₂
NIC
ストレージ・システム
130₁
ストレージ・システム
130₂
ストレージ・システム・マネージャ
131
ストレージ・システム・マネージャ
132
分散型ストレージ・システム・マネージャ
135
151
152
153
vvol
PE
161
PE
166
PE
162
PE
163
PE
164
PE
165
142A
142B
142C
154
155
156
ファイル・システム
145
141
DSU

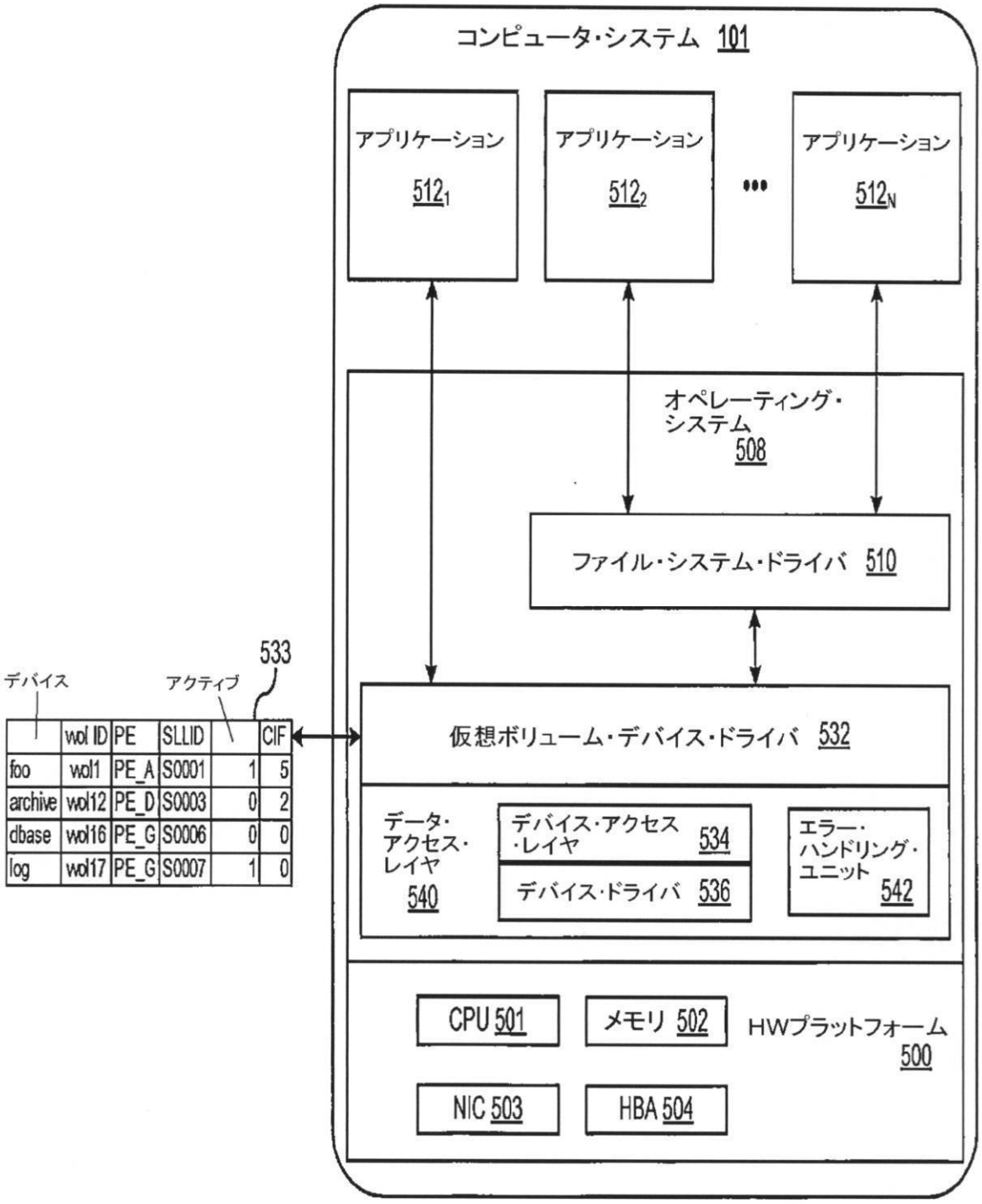
【図３】


312
PE ID | SLLID | Gen# | wol ID | RefCnt
PE_A | S0001 | 0001 | wol1 | 2
PE_A | S0002 | 0002 | wol4 | 1
PE_D | S0003 | 0001 | wol12 | 3
PE_E | S0004 | 0005 | wol14 | 1
PE_F | S0005 | 0006 | wol14 | 1
PE_G | S0006 | 0008 | wol16 | 1
PE_G | S0007 | 0001 | wol17 | 1
314
wol ID | コンテナ・ロケーション | キー／値のペア
wol1 | SC1:<オフセット1,長さ1> | AppID=VMxx タイプ＝データ [ホスト固有] [ストレージ固有]
wol2 | SC2:<オフセット1,長さ1>, <オフセット2,長さ2>, <オフセット3,長さ3> | ***
wol3 | SC3:<オフセット1,長さ1> | ***
*** | *** | ***
wol13 | SC1:<オフセット1,長さ1>, <オフセット2,長さ2> | ***
*** | | ***
wol17 | SC3:<オフセット1,長さ1> | ***
Cnt ID | エクステント | メタデータ
SC1 | A1:DSU1:P4, A1:DSU3:P2, A1:DSU3:P3, A2:DSU1:P1, A2:DSU2:P5, A2:DSU2:P1, A2:DSU2:P3, A2:DSU3:P5 | [コンテキストID] [アロケーション・ビットマップ]
SC2 | A1:DSU1:P1, A1:DSU2:P2, A1:DSU3:P1, A1:DSU4:P2 | ***
SC3 | A2:DSU1:P2, A2:DSU2:P3, A2:DSU2:P4 | ***
316
PE_A
PE_G
IOマネージャ 304
ボリューム・マネージャ 306
コンテナ・マネージャ 308
データ・アクセス・レイヤ 310
A1
A2
P1
P2
P3
P4
Pn
DSU(s) 141
DSU(s) 141

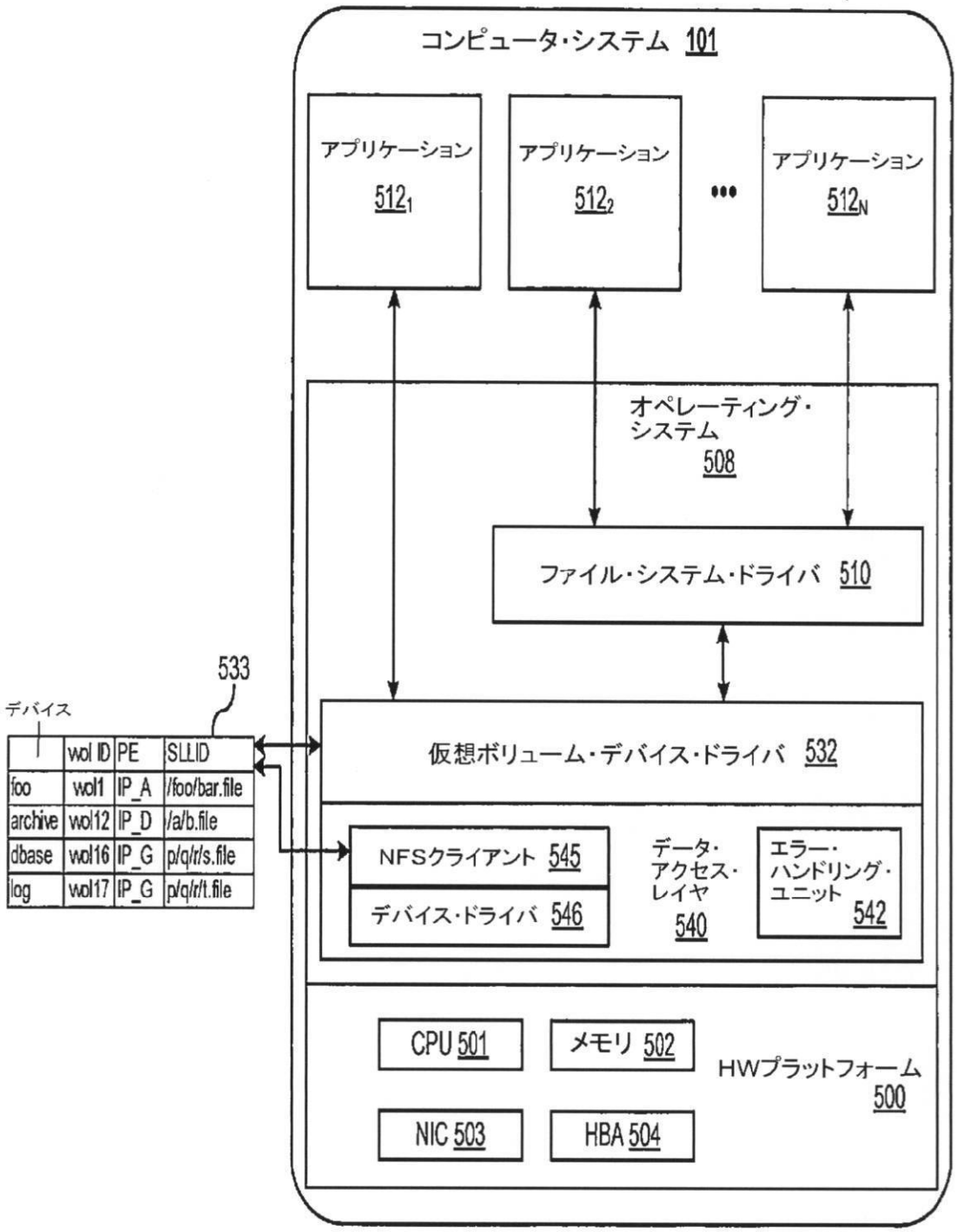
【図４】


410
SCを作成する
ストレージ・コンテナの物理的なキャパシティーを設定する
411
ストレージ・コンテナにアクセスするための許可レベルを設定する
412
一意のコンテナIDを生成する
413
DSUへのマッピングを作成し、コンテナ・データベースを更新する
414
終了

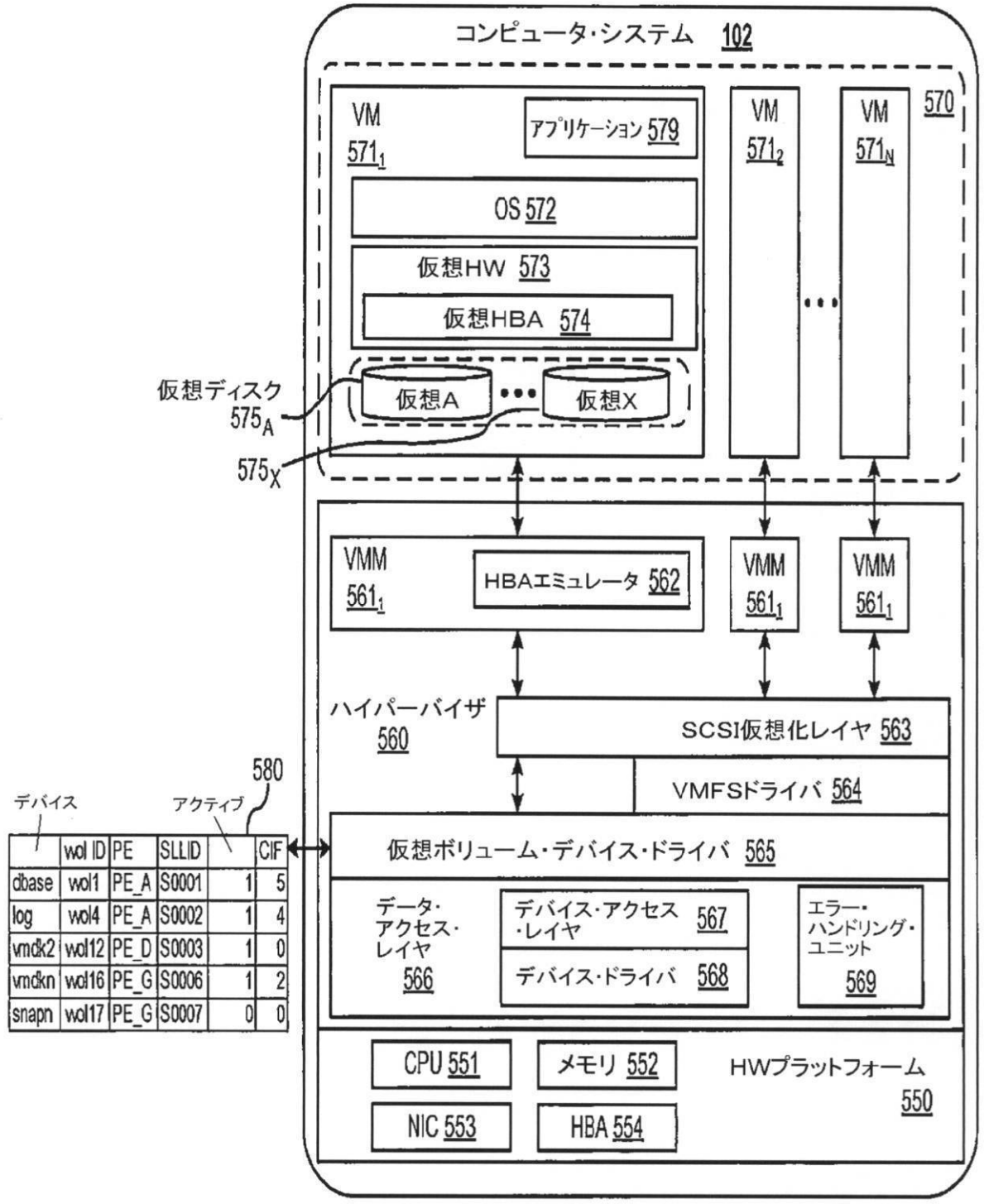
【図５Ａ】


コンピュータ・システム 101
アプリケーション 512₁
アプリケーション 512₂
アプリケーション 512N
オペレーティング・システム 508
ファイル・システム・ドライバ 510
533
デバイス
アクティブ
vol ID PE SLLID CIF
foo vol1 PE_A S0001 1 5
archive vol12 PE_D S0003 0 2
dbase vol16 PE_G S0006 0 0
log vol17 PE_G S0007 1 0
仮想ボリューム・デバイス・ドライバ 532
データ・アクセス・レイヤ 540
デバイス・アクセス・レイヤ 534
デバイス・ドライバ 536
エラー・ハンドリング・ユニット 542
CPU 501
メモリ 502
HWプラットフォーム 500
NIC 503
HBA 504

【図５Ｂ】

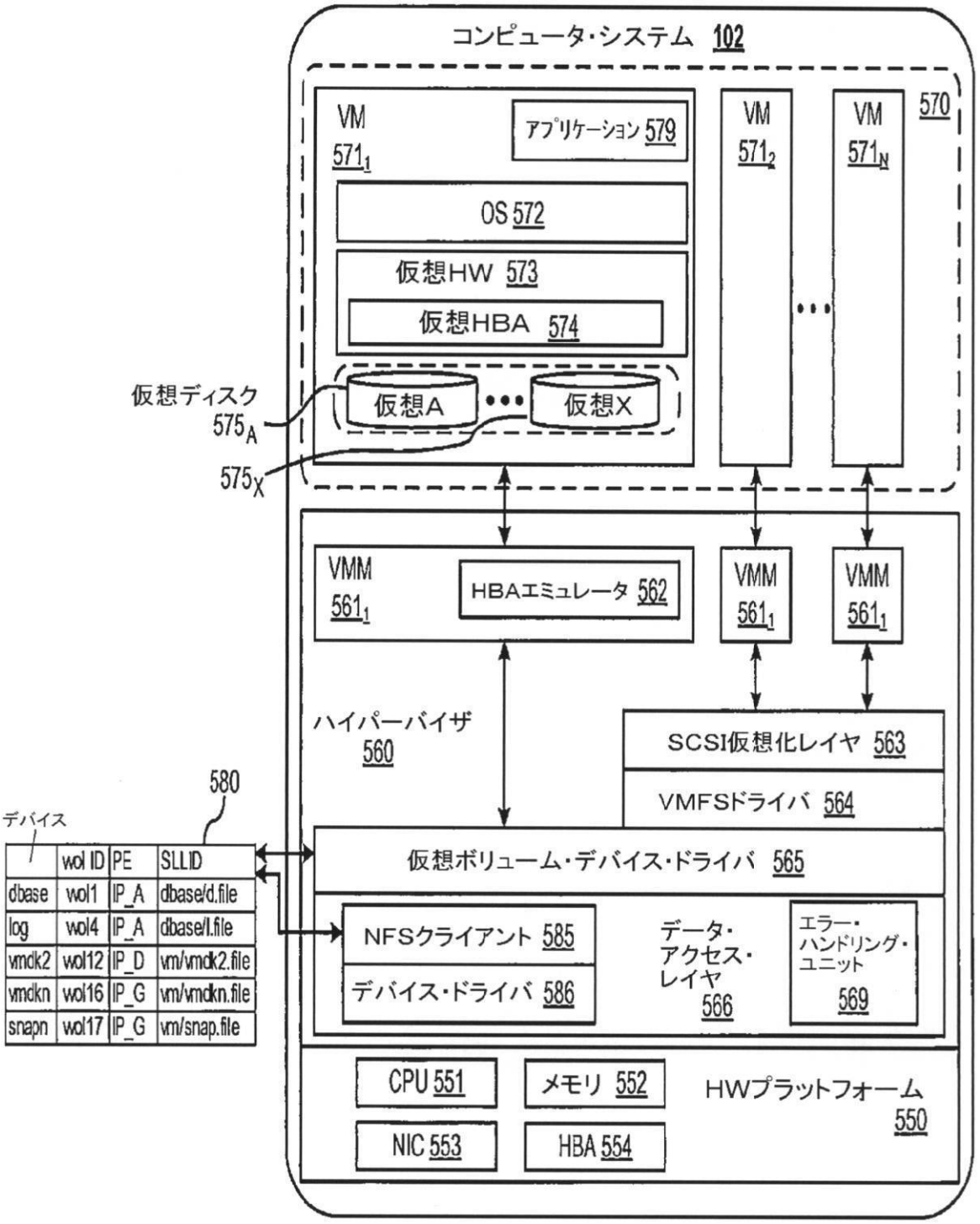

コンピュータ・システム 101
アプリケーション 512₁
アプリケーション 512₂
アプリケーション 512N
オペレーティング・システム 508
ファイル・システム・ドライバ 510
533
デバイス
vol ID
PE
SLLID
foo
vol1
IP_A
/foo/bar.file
archive
vol12
IP_D
/a/b.file
dbase
vol16
IP_G
p/q/r/s.file
log
vol17
IP_G
p/q/r/t.file
仮想ボリューム・デバイス・ドライバ 532
NFSクライアント 545
デバイス・ドライバ 546
データ・アクセス・レイヤ 540
エラー・ハンドリング・ユニット 542
CPU 501
メモリ 502
NIC 503
HBA 504
HWプラットフォーム 500

【図５Ｃ】

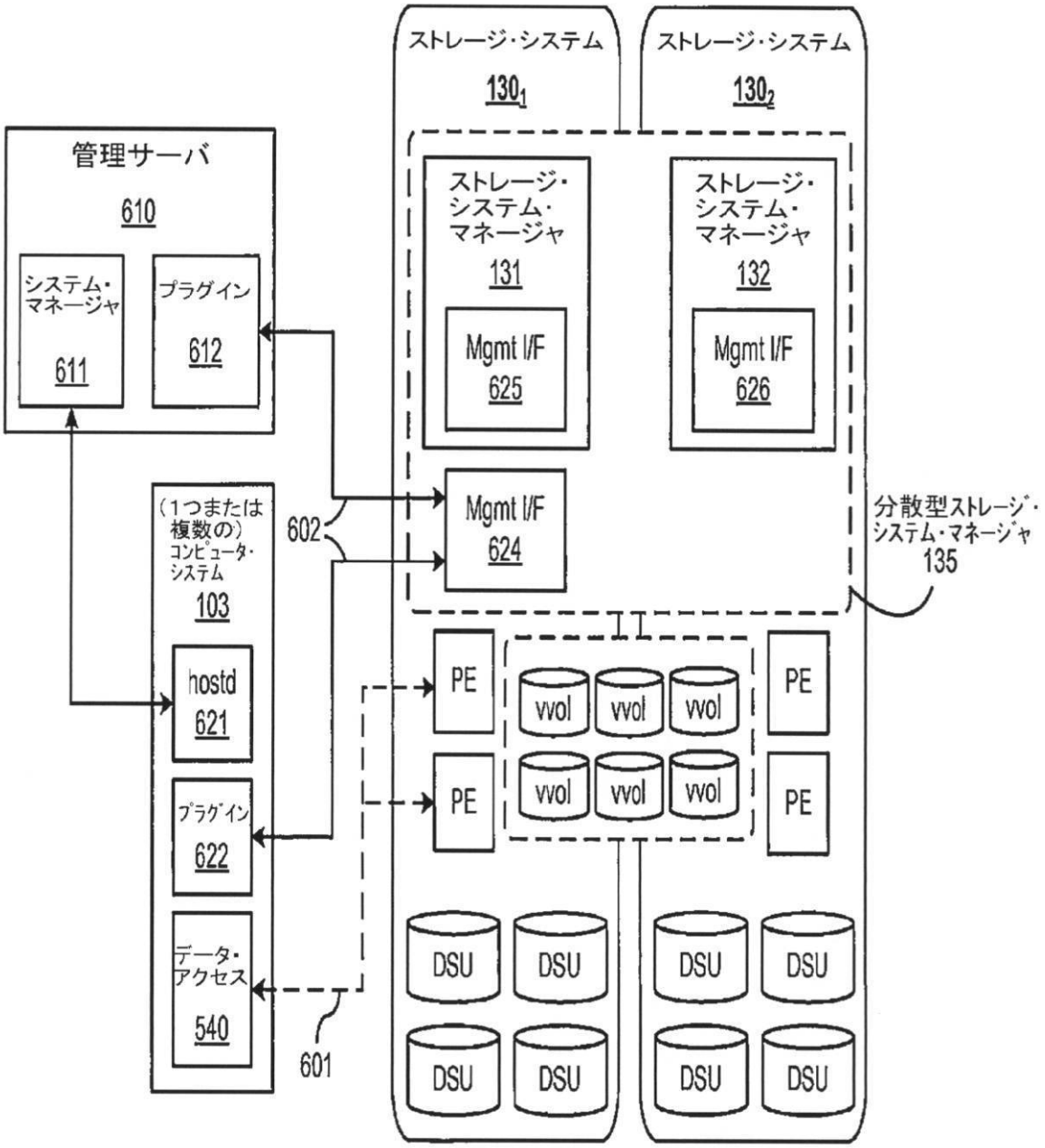


	wol ID	PE	SLLID		CIF
dbase	wol1	PE_A	S0001	1	5
log	wol4	PE_A	S0002	1	4
vmdk2	wol12	PE_D	S0003	1	0
vmdkn	wol16	PE_G	S0006	1	2
snapn	wol17	PE_G	S0007	0	0

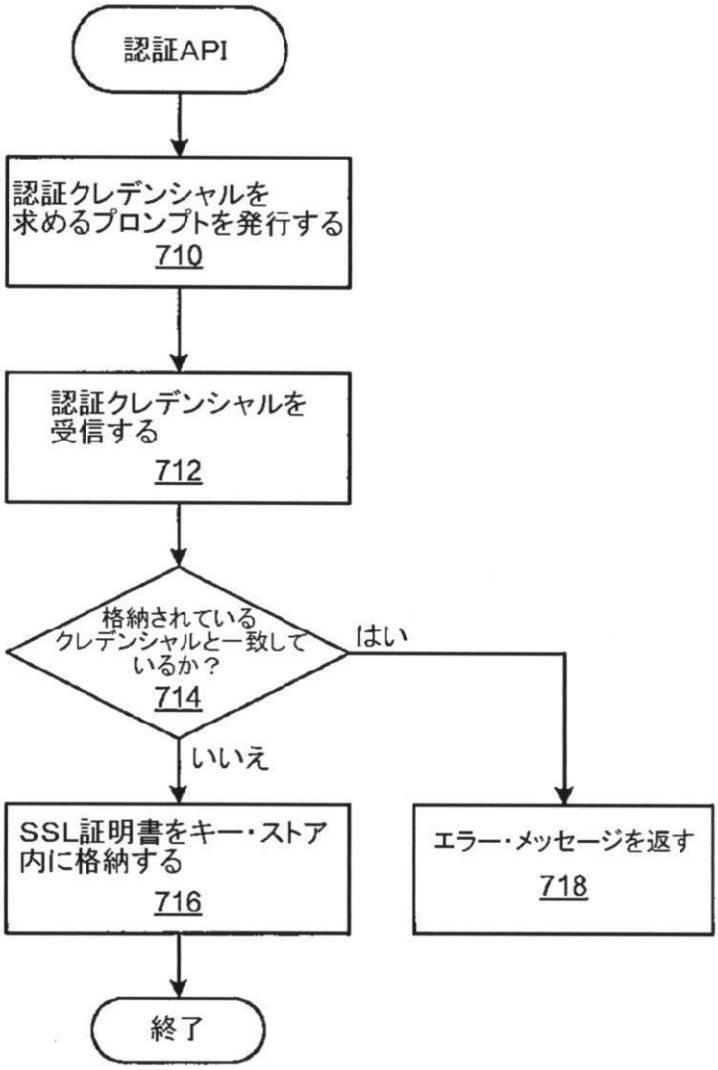
【図５Ｄ】


コンピュータ・システム 102
VM 571₁
アプリケーション 579
OS 572
仮想HW 573
仮想HBA 574
仮想A
仮想X
VM 571₂
VM 571N
570
仮想ディスク 575A
575X
VMM 561₁
HBAエミュレータ 562
ハイパーバイザ 560
SCSI仮想化レイヤ 563
VMFSドライバ 564
仮想ボリューム・デバイス・ドライバ 565
NFSクライアント 585
デバイス・ドライバ 586
データ・アクセス・レイヤ 566
エラー・ハンドリング・ユニット 569
CPU 551
メモリ 552
NIC 553
HBA 554
HWプラットフォーム 550
580
デバイス
vol ID
PE
SLLID
dbase vol1 IP_A dbase/d.file
log vol4 IP_A dbase/l.file
vmdk2 vol12 IP_D vm/vmdk2.file
vmdkn vol16 IP_G vm/vmdkn.file
snapn vol17 IP_G vm/snap.file

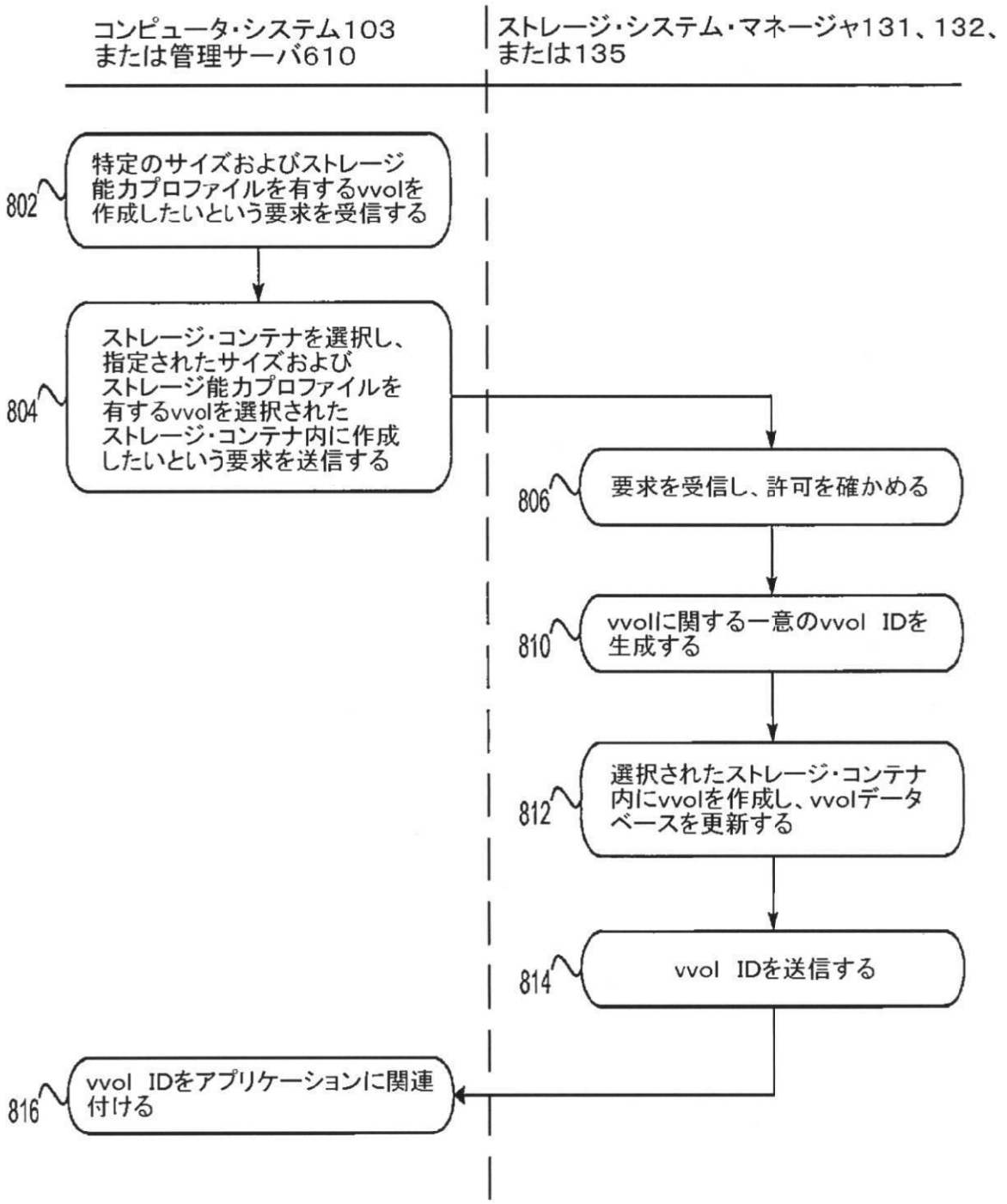
【図６】


ストレージ・システム
130₁
ストレージ・システム
130₂
管理サーバ
610
システム・マネージャ
611
プラグイン
612
ストレージ・システム・マネージャ
131
Mgmt I/F
625
ストレージ・システム・マネージャ
132
Mgmt I/F
626
Mgmt I/F
624
分散型ストレージ・システム・マネージャ
135
602
（1つまたは複数の）コンピュータ・システム
103
hostd
621
プラグイン
622
データ・アクセス
540
601
PE
PE
PE
PE
vvol
vvol
vvol
vvol
vvol
vvol
DSU
DSU
DSU
DSU
DSU
DSU
DSU
DSU

【図７】


認証API
認証クレデンシャルを求めるプロンプトを発行する
710
認証クレデンシャルを受信する
712
格納されているクレデンシャルと一致しているか？
714
はい
いいえ
SSL証明書をキー・ストア内に格納する
716
エラー・メッセージを返す
718
終了

【図８】


コンピュータ・システム103
または管理サーバ610
ストレージ・システム・マネージャ131、132、
または135
802
特定のサイズおよびストレージ
能力プロファイルを有するvvolを
作成したいという要求を受信する
804
ストレージ・コンテナを選択し、
指定されたサイズおよび
ストレージ能力プロファイルを
有するvvolを選択された
ストレージ・コンテナ内に作成
したいという要求を送信する
806
要求を受信し、許可を確かめる
810
vvolに関する一意のvvol IDを
生成する
812
選択されたストレージ・コンテナ
内にvvolを作成し、vvolデータ
ベースを更新する
814
vvol IDを送信する
816
vvol IDをアプリケーションに関連
付ける

【図９】

図9A

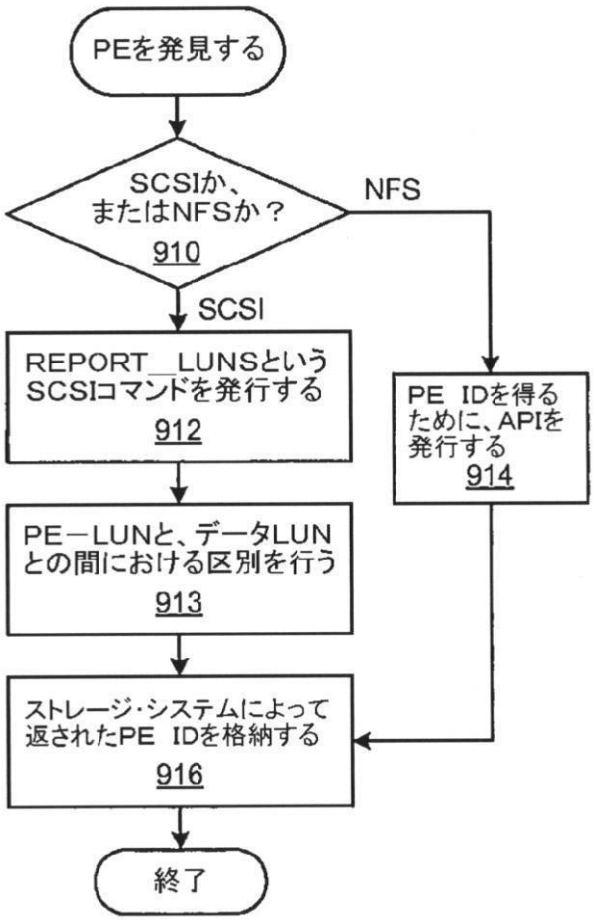
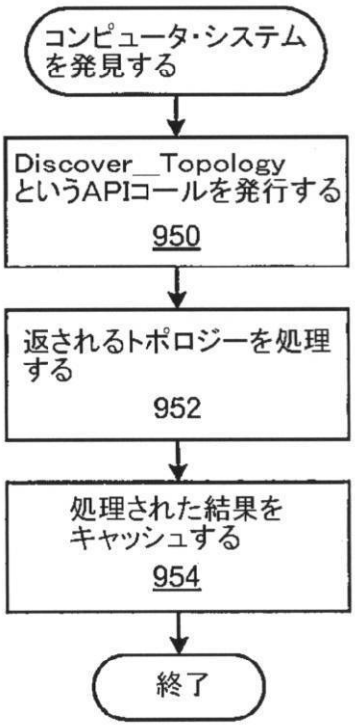
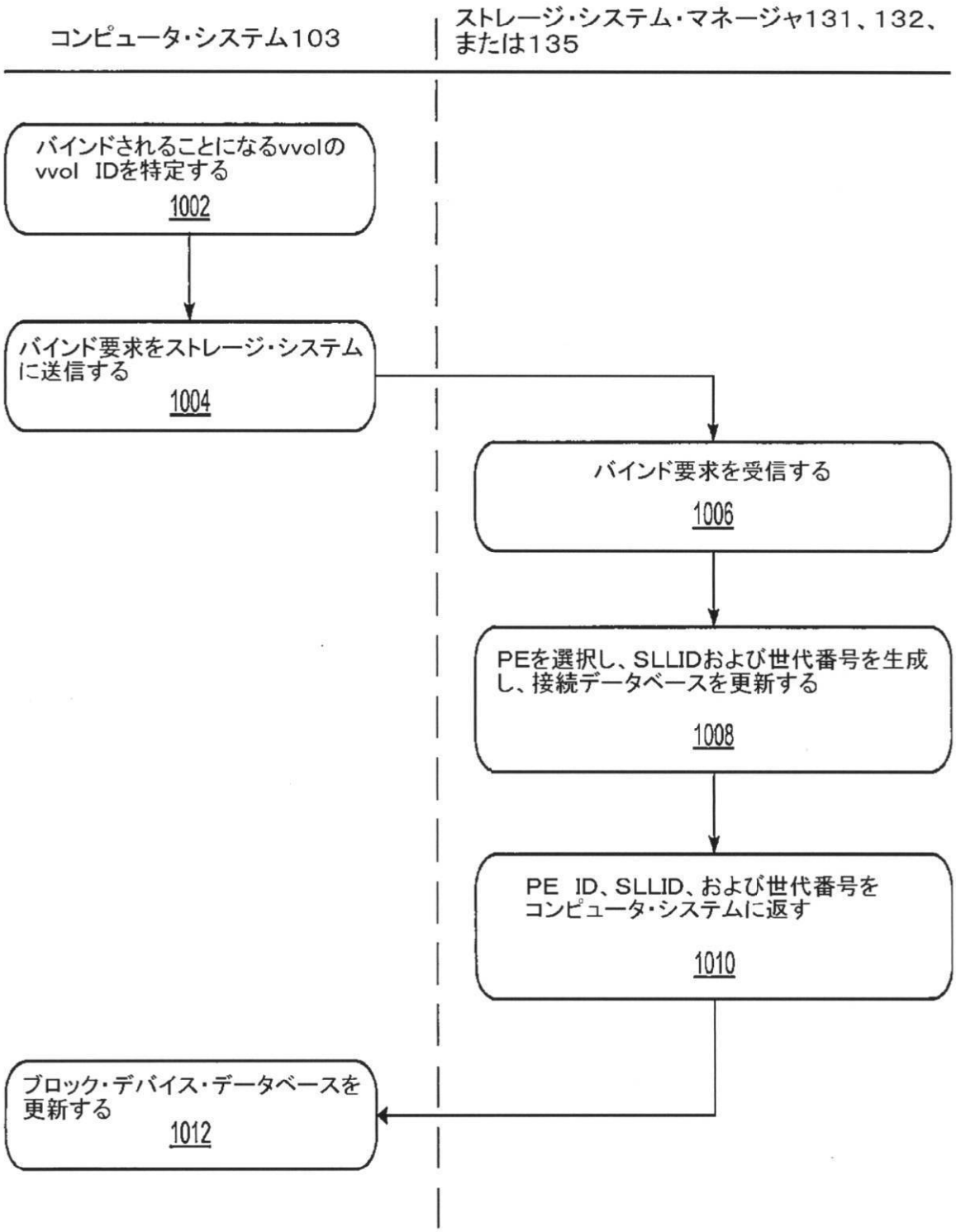

PEを発見する
SCSIか、
またはNFSか？
910
NFS
SCSI
REPORT_LUNSという
SCSIコマンドを発行する
912
PE IDを得る
ために、APIを
発行する
914
PE－LUNと、データLUN
との間における区別を行う
913
ストレージ・システムによって
返されたPE IDを格納する
916
終了

図9B


コンピュータ・システム
を発見する
Discover_Topology
というAPIコールを発行する
950
返されるトポロジーを処理
する
952
処理された結果を
キャッシュする
954
終了

【図１０】


コンピュータ・システム103
ストレージ・システム・マネージャ131、132、または135
バインドされることになるvvolのvvol IDを特定する
1002
バインド要求をストレージ・システムに送信する
1004
バインド要求を受信する
1006
PEを選択し、SLLIDおよび世代番号を生成し、接続データベースを更新する
1008
PE ID、SLLID、および世代番号をコンピュータ・システムに返す
1010
ブロック・デバイス・データベースを更新する
1012

【図１１】

図11A

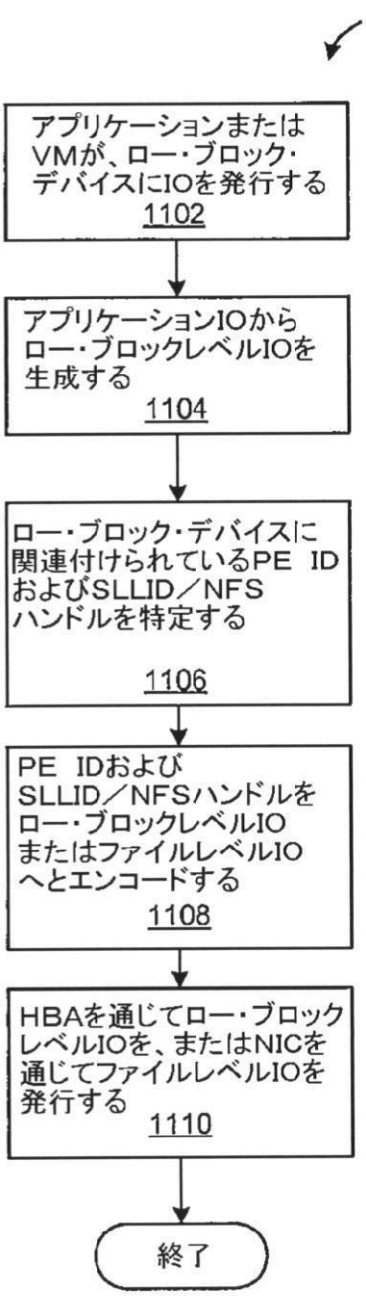
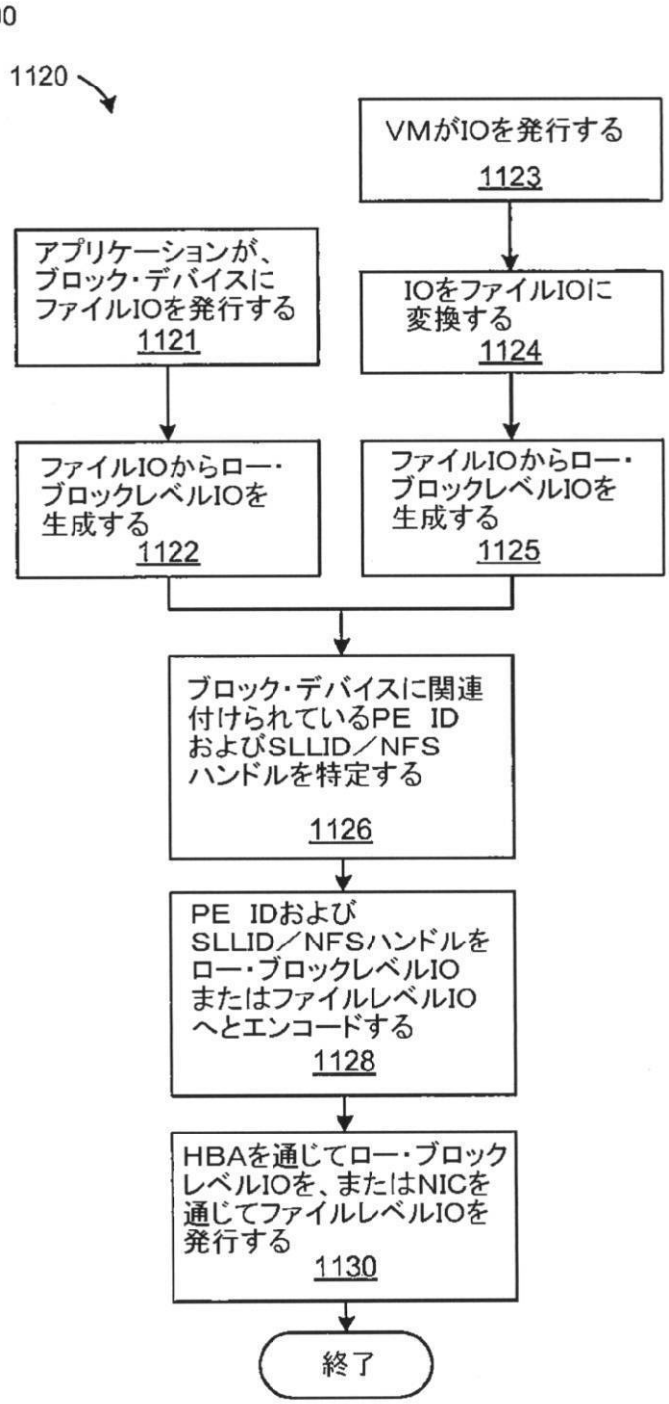
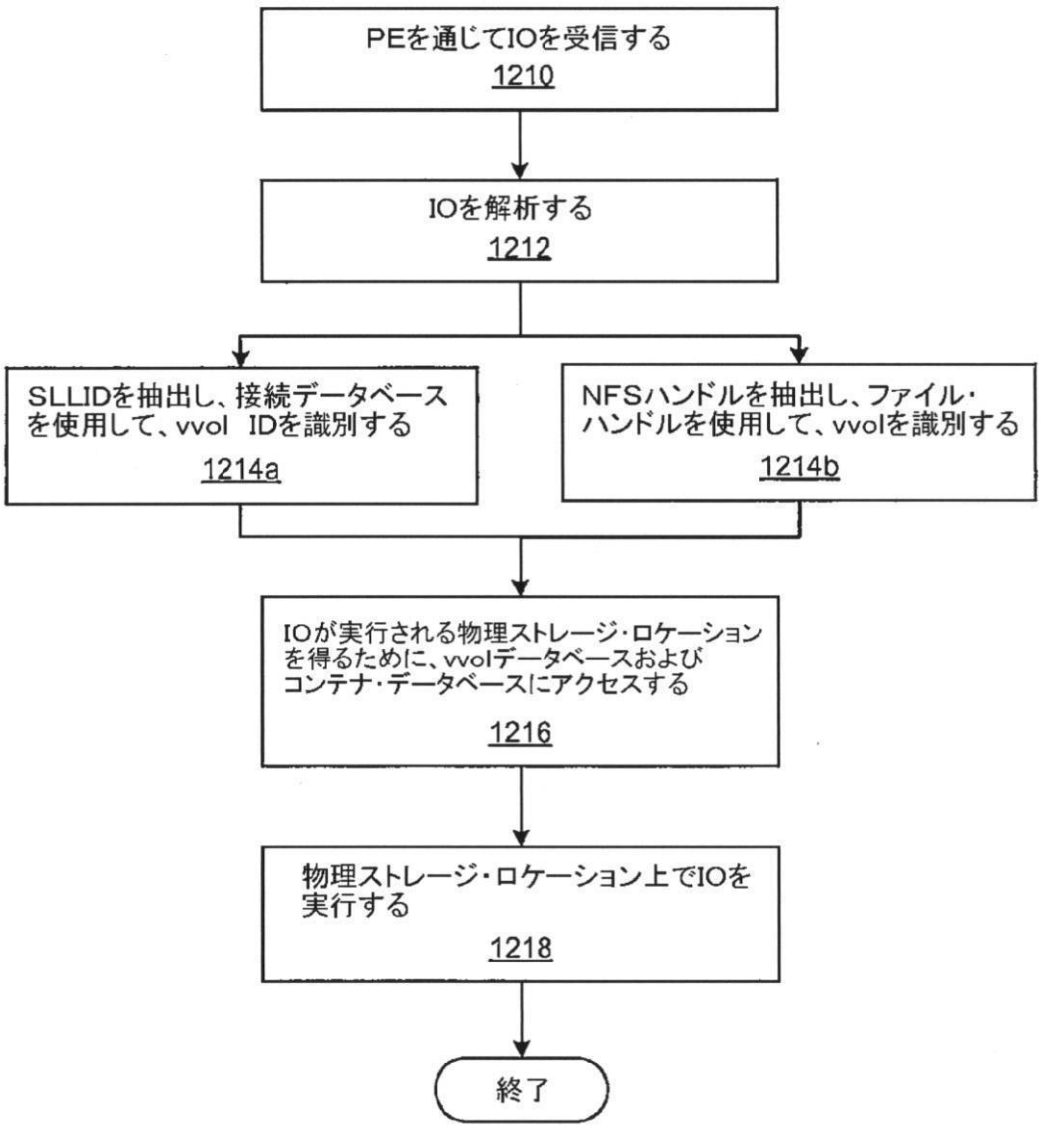


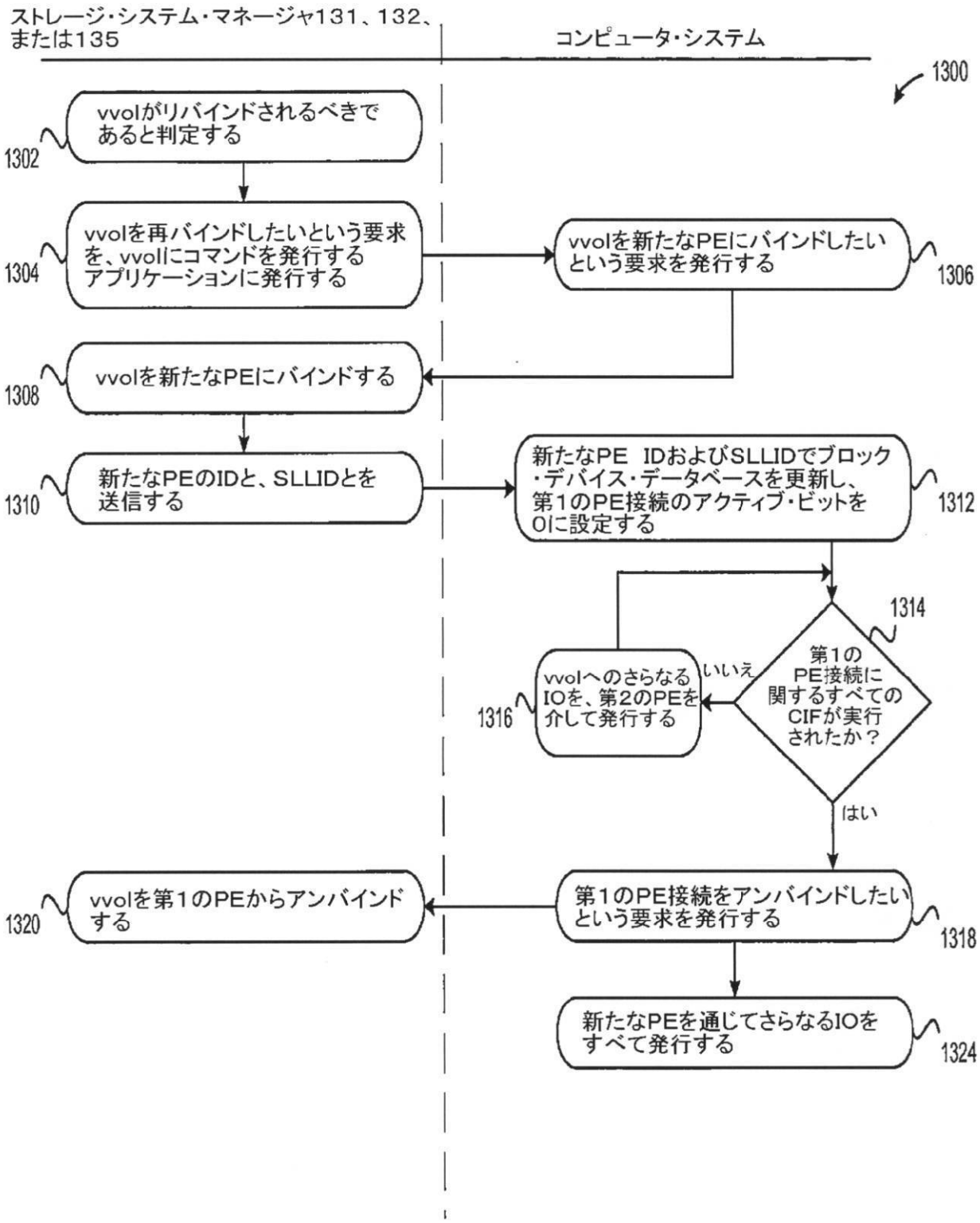
図11B



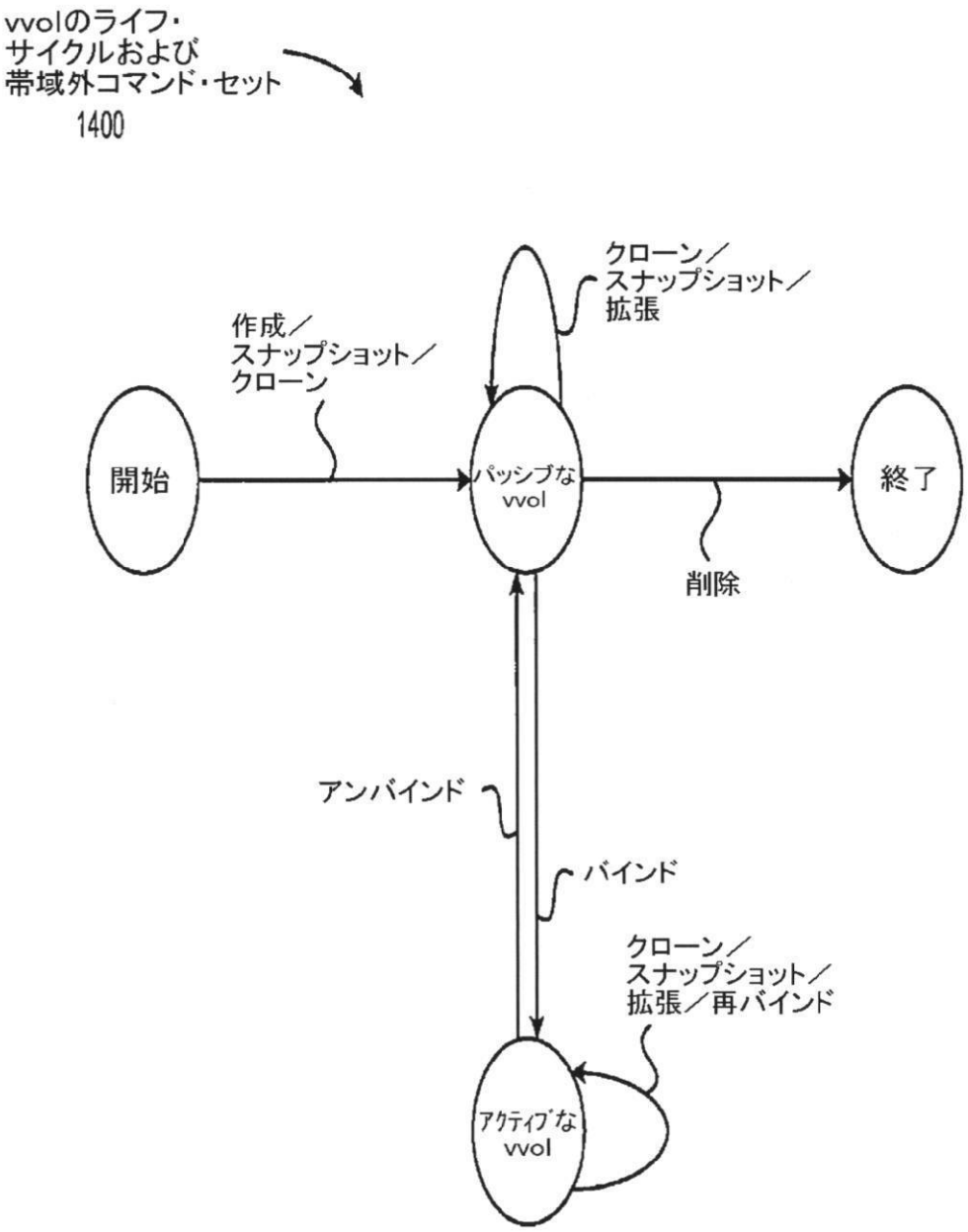
【図１２】


PEを通じてIOを受信する
1210
IOを解析する
1212
SLLIDを抽出し、接続データベースを使用して、vvol IDを識別する
1214a
NFSハンドルを抽出し、ファイル・ハンドルを使用して、vvolを識別する
1214b
IOが実行される物理ストレージ・ロケーションを得るために、vvolデータベースおよびコンテナ・データベースにアクセスする
1216
物理ストレージ・ロケーション上でIOを実行する
1218
終了

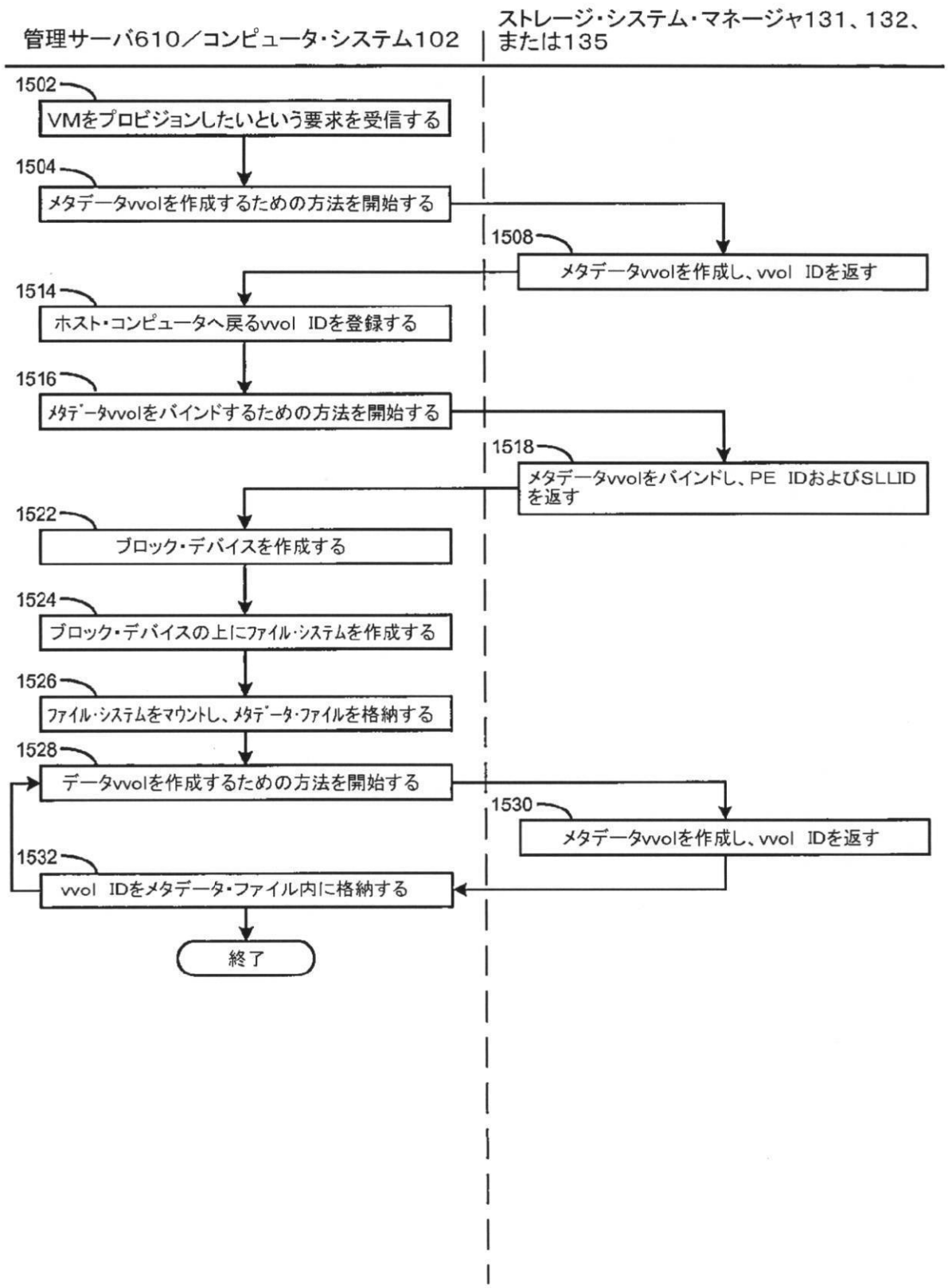
【図１３】


ストレージ・システム・マネージャ131、132、または135
コンピュータ・システム
1300
vvolがリバインドされるべきであると判定する
1302
vvolを再バインドしたいという要求を、vvolにコマンドを発行するアプリケーションに発行する
1304
vvolを新たなPEにバインドしたいという要求を発行する
1306
vvolを新たなPEにバインドする
1308
新たなPEのIDと、SLLIDとを送信する
1310
新たなPE IDおよびSLLIDでブロック・デバイス・データベースを更新し、第1のPE接続のアクティブ・ビットを0に設定する
1312
1314
第1のPE接続に関するすべてのCIFが実行されたか？
いいえ
vvolへのさらなるIOを、第2のPEを介して発行する
1316
はい
第1のPE接続をアンバインドしたいという要求を発行する
1318
vvolを第1のPEからアンバインドする
1320
新たなPEを通じてさらなるIOをすべて発行する
1324

【図１４】


vvolのライフ・
サイクルおよび
帯域外コマンド・セット
1400
クローン／
スナップショット／
拡張
作成／
スナップショット／
クローン
開始
パッシブな
vvol
終了
削除
アンバインド
バインド
クローン／
スナップショット／
拡張／再バインド
ｱｸﾃｨﾌﾞな
vvol

【図１５】


管理サーバ610／コンピュータ・システム102
ストレージ・システム・マネージャ131、132、または135
1502
VMをプロビジョンしたいという要求を受信する
1504
メタデータvvolを作成するための方法を開始する
1508
メタデータvvolを作成し、vvol IDを返す
1514
ホスト・コンピュータへ戻るvvol IDを登録する
1516
メタデータvvolをバインドするための方法を開始する
1518
メタデータvvolをバインドし、PE IDおよびSLLIDを返す
1522
ブロック・デバイスを作成する
1524
ブロック・デバイスの上にファイル・システムを作成する
1526
ファイル・システムをマウントし、メタデータ・ファイルを格納する
1528
データvvolを作成するための方法を開始する
1530
メタデータvvolを作成し、vvol IDを返す
1532
vvol IDをメタデータ・ファイル内に格納する
終了

【図１６】

図16A

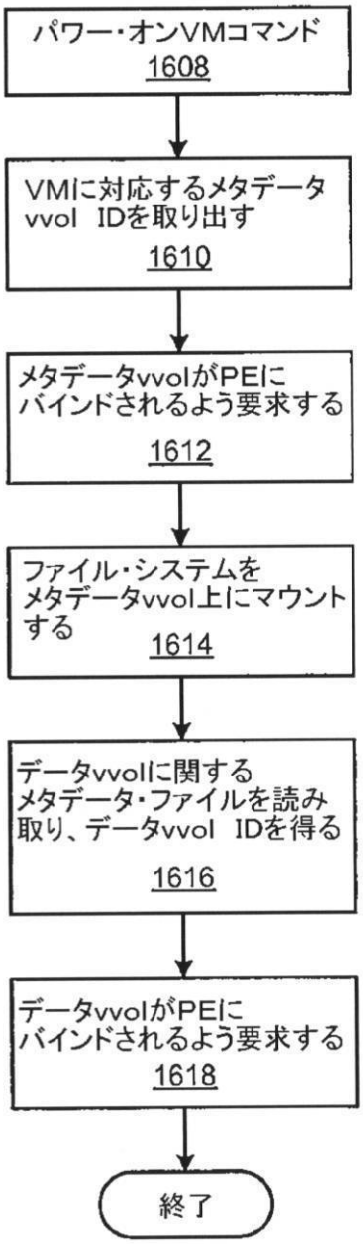
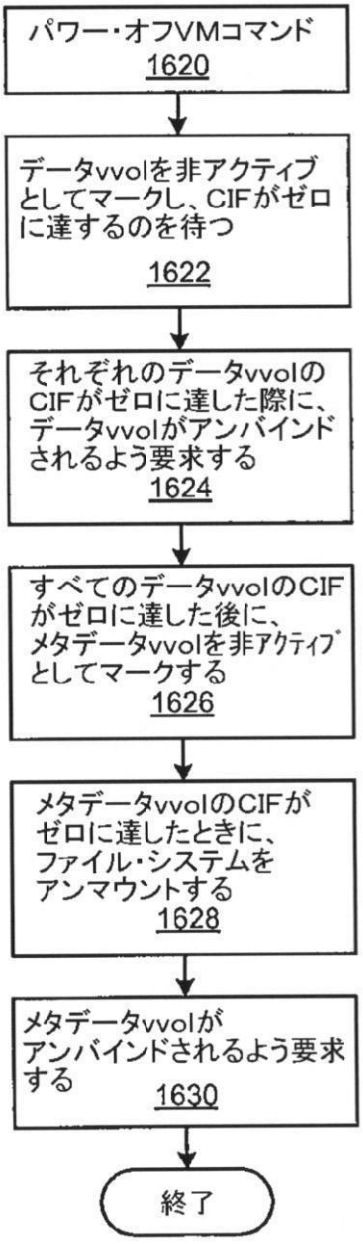
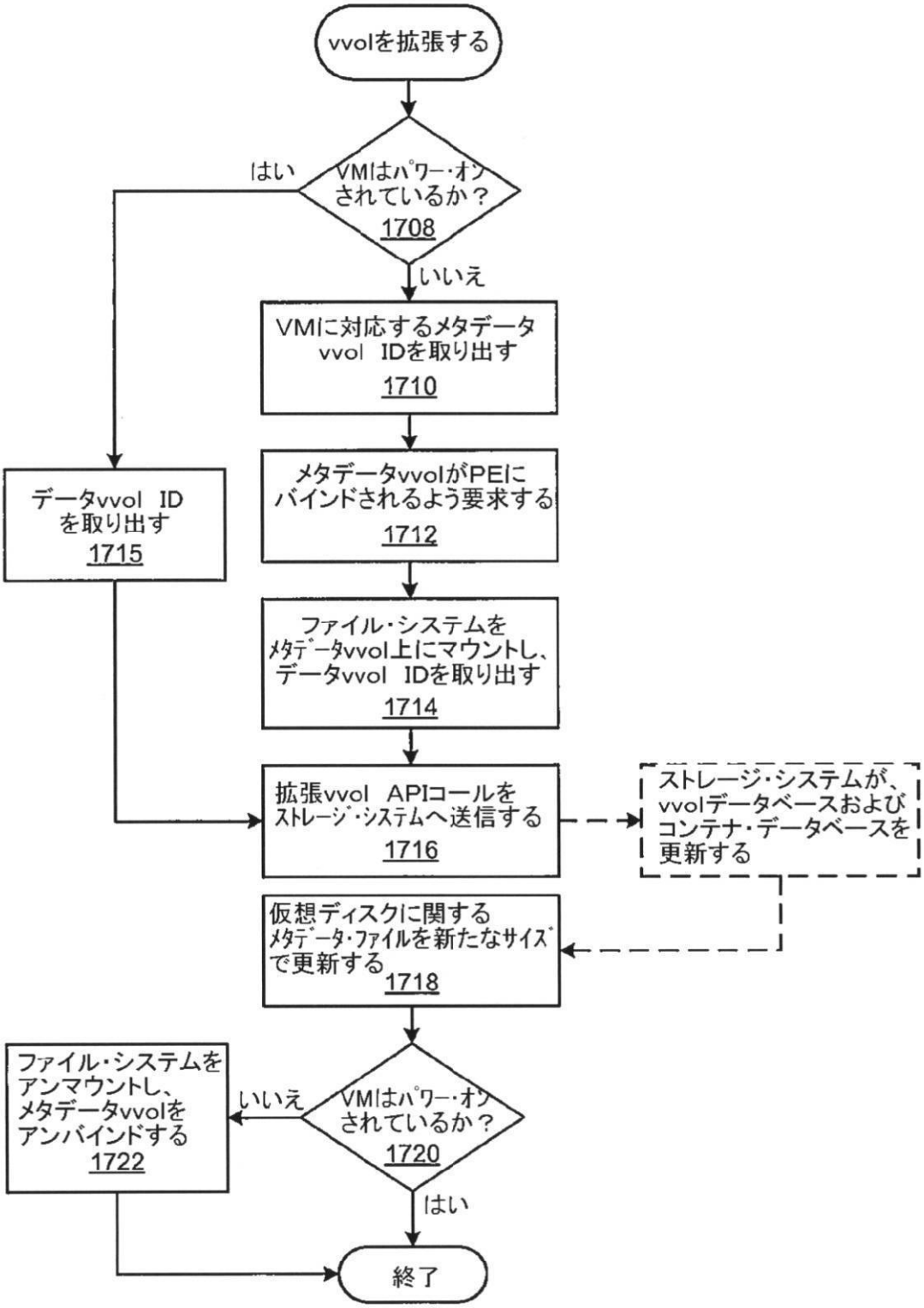


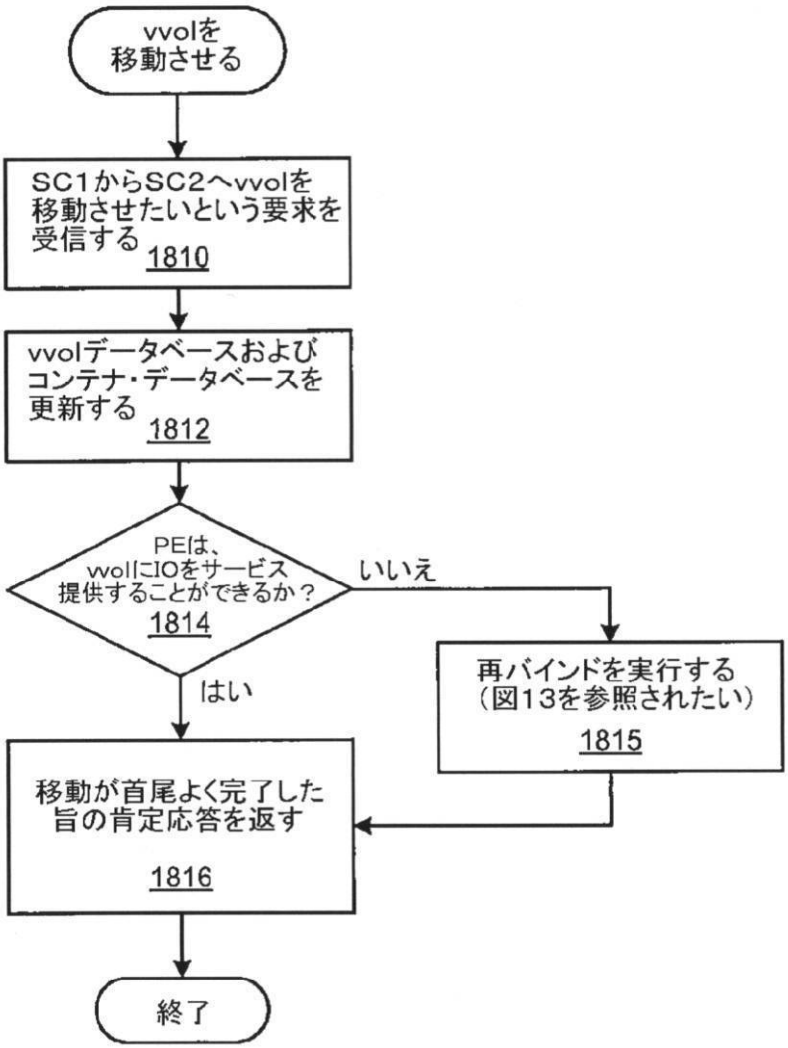
図16B



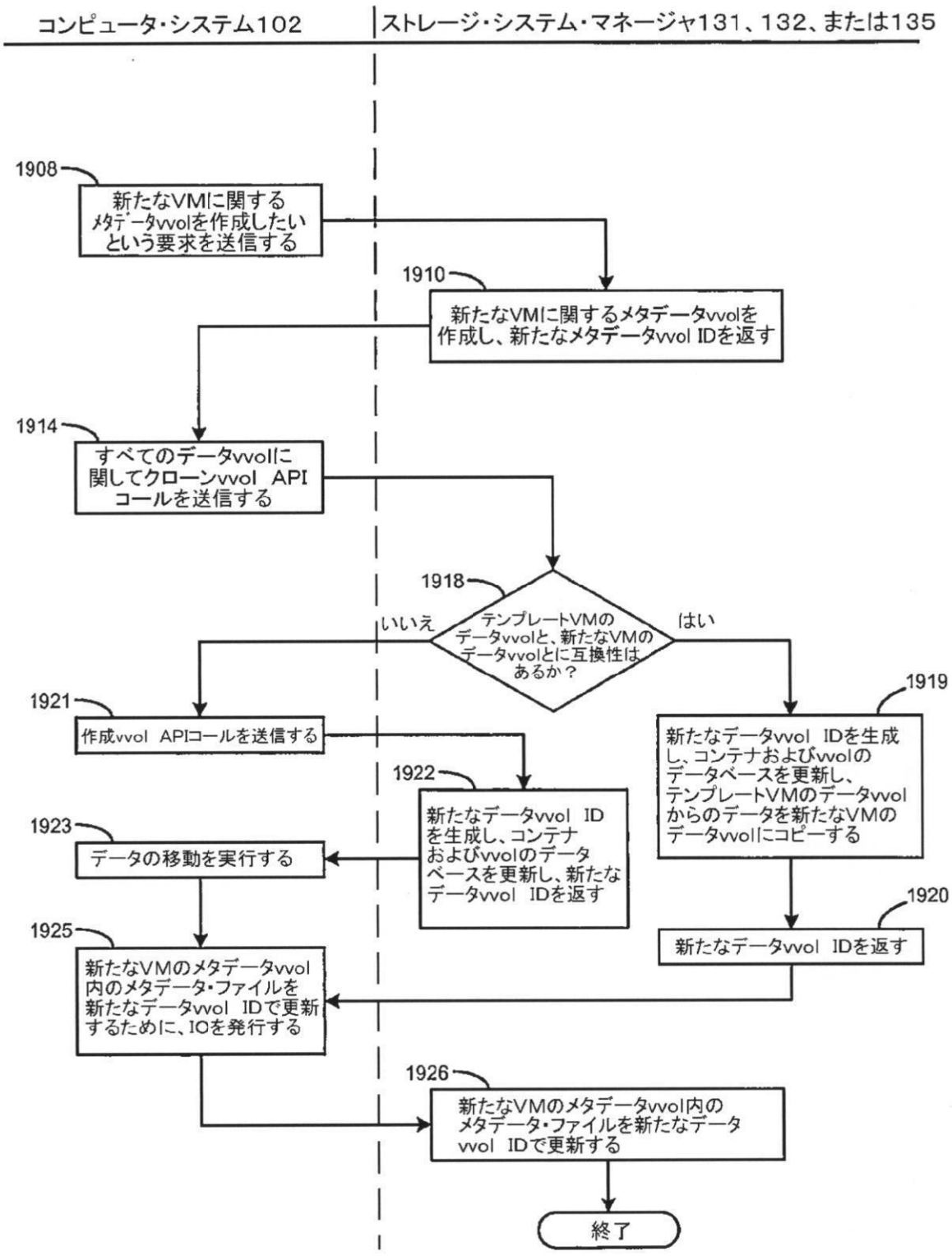
【図１７】


vvolを拡張する
はい
VMはパワー・オフされているか？
1708
いいえ
VMに対応するメタデータvvol IDを取り出す
1710
メタデータvvolがPEにバインドされるよう要求する
1712
ファイル・システムをメタデータvvol上にマウントし、データvvol IDを取り出す
1714
データvvol IDを取り出す
1715
拡張vvol APIコールをストレージ・システムへ送信する
1716
ストレージ・システムが、vvolデータベースおよびコンテナ・データベースを更新する
仮想ディスクに関するメタデータ・ファイルを新たなサイズで更新する
1718
VMはパワー・オフされているか？
1720
いいえ
ファイル・システムをアンマウントし、メタデータvvolをアンバインドする
1722
はい
終了

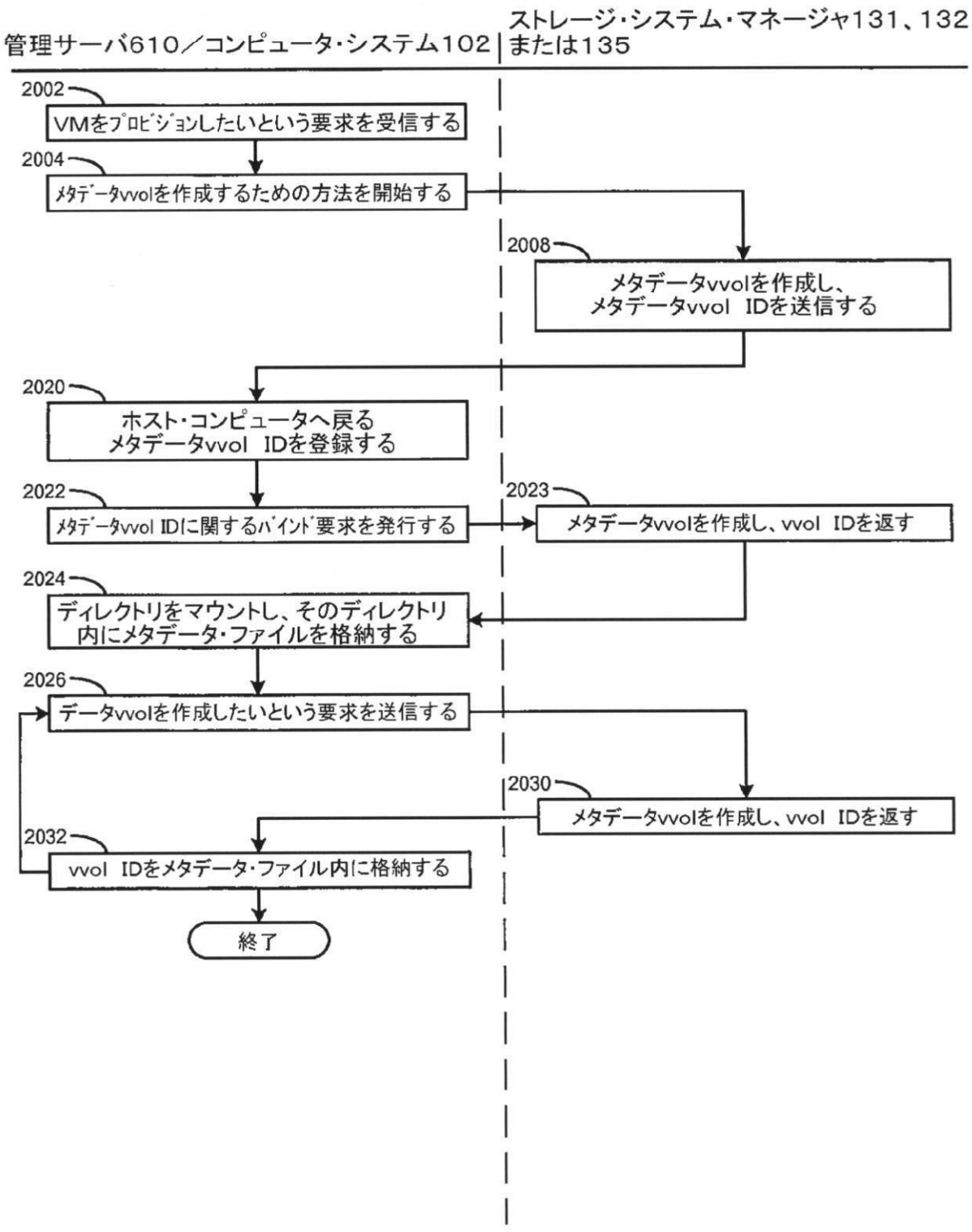
【図１８】


vvolを
移動させる
SC1からSC2へvvolを
移動させたいという要求を
受信する
1810
vvolデータベースおよび
コンテナ・データベースを
更新する
1812
PEは、
vvolにIOをサービス
提供することができるか？
1814
いいえ
はい
再バインドを実行する
（図13を参照されたい）
1815
移動が首尾よく完了した
旨の肯定応答を返す
1816
終了

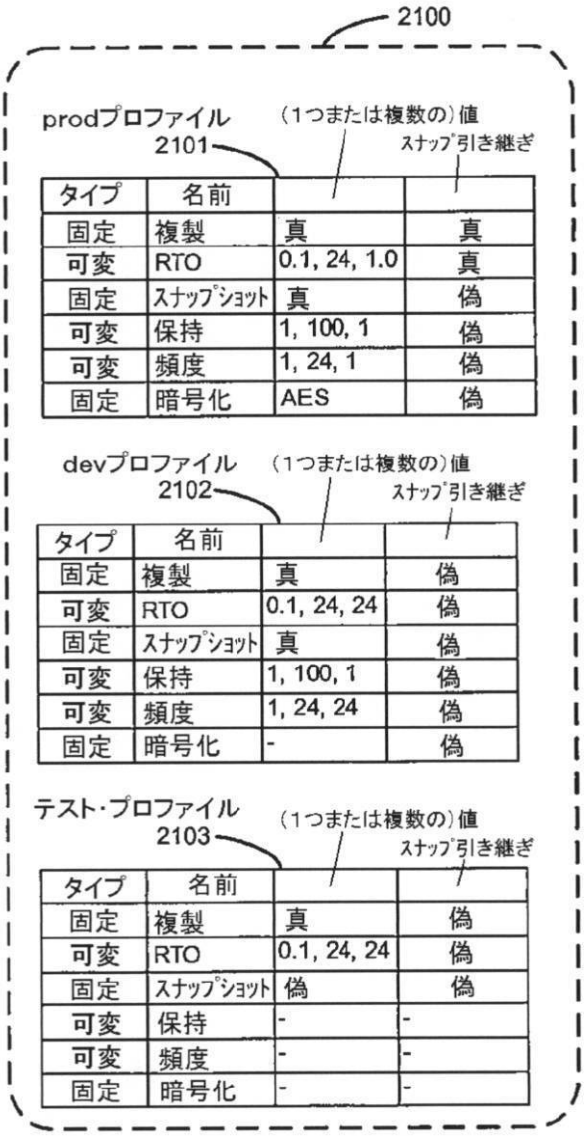
【図１９】


コンピュータ・システム102
ストレージ・システム・マネージャ131、132、または135
1908
新たなVMに関するメタデータvvolを作成したいという要求を送信する
1910
新たなVMに関するメタデータvvolを作成し、新たなメタデータvvol IDを返す
1914
すべてのデータvvolに関してクローンvvol API コールを送信する
1918
テンプレートVMのデータvvolと、新たなVMのデータvvolとに互換性はあるか？
いいえ
はい
1921
作成vvol APIコールを送信する
1922
新たなデータvvol IDを生成し、コンテナおよびvvolのデータベースを更新し、新たなデータvvol IDを返す
1923
データの移動を実行する
1919
新たなデータvvol IDを生成し、コンテナおよびvvolのデータベースを更新し、テンプレートVMのデータvvolからのデータを新たなVMのデータvvolにコピーする
1920
新たなデータvvol IDを返す
1925
新たなVMのメタデータvvol内のメタデータ・ファイルを新たなデータvvol IDで更新するために、IOを発行する
1926
新たなVMのメタデータvvol内のメタデータ・ファイルを新たなデータvvol IDで更新する
終了

【図２０】


管理サーバ610／コンピュータ・システム102
ストレージ・システム・マネージャ131、132、または135
2002
VMをプロビジョンしたいという要求を受信する
2004
メタデータvvolを作成するための方法を開始する
2008
メタデータvvolを作成し、
メタデータvvol IDを送信する
2020
ホスト・コンピュータへ戻る
メタデータvvol IDを登録する
2022
メタデータvvol IDに関するバインド要求を発行する
2023
メタデータvvolを作成し、vvol IDを返す
2024
ディレクトリをマウントし、そのディレクトリ
内にメタデータ・ファイルを格納する
2026
データvvolを作成したいという要求を送信する
2030
メタデータvvolを作成し、vvol IDを返す
2032
vvol IDをメタデータ・ファイル内に格納する
終了

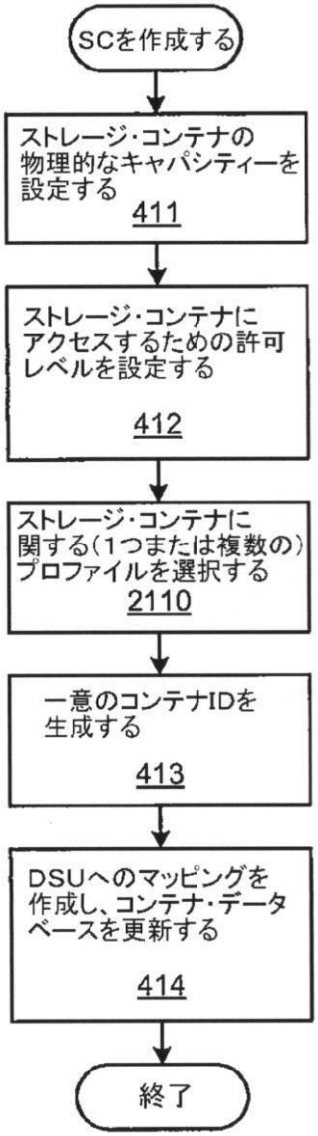
【図２１】



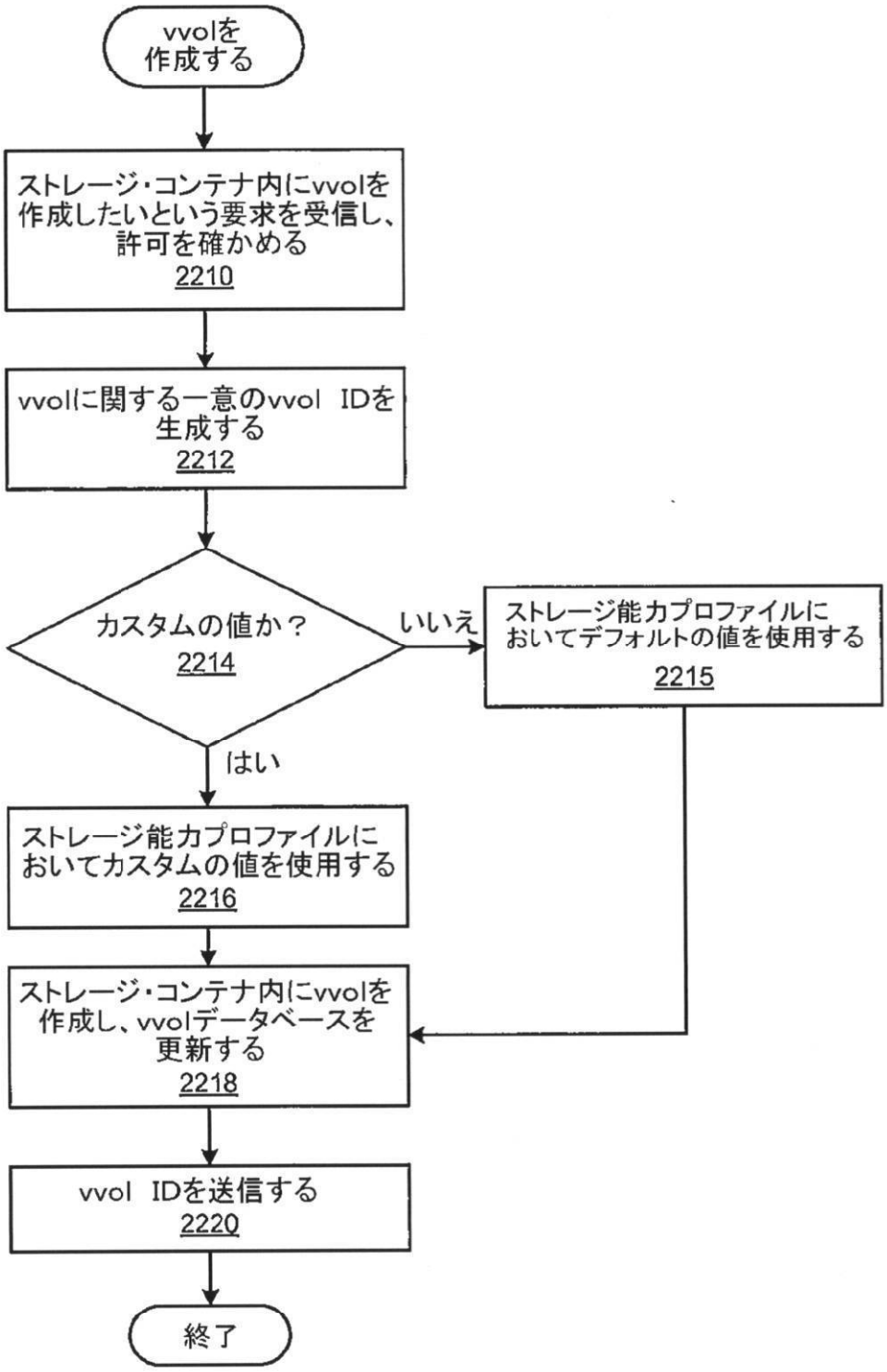
タイプ	名前	（1つまたは複数の）値	スナップ引き継ぎ
固定	複製	真	真
可変	RTO	0.1, 24, 1.0	真
固定	スナップショット	真	偽
可変	保持	1, 100, 1	偽
可変	頻度	1, 24, 1	偽
固定	暗号化	AES	偽

タイプ	名前	（1つまたは複数の）値	スナップ引き継ぎ
固定	複製	真	偽
可変	RTO	0.1, 24, 24	偽
固定	スナップショット	真	偽
可変	保持	1, 100, 1	偽
可変	頻度	1, 24, 24	偽
固定	暗号化	-	偽

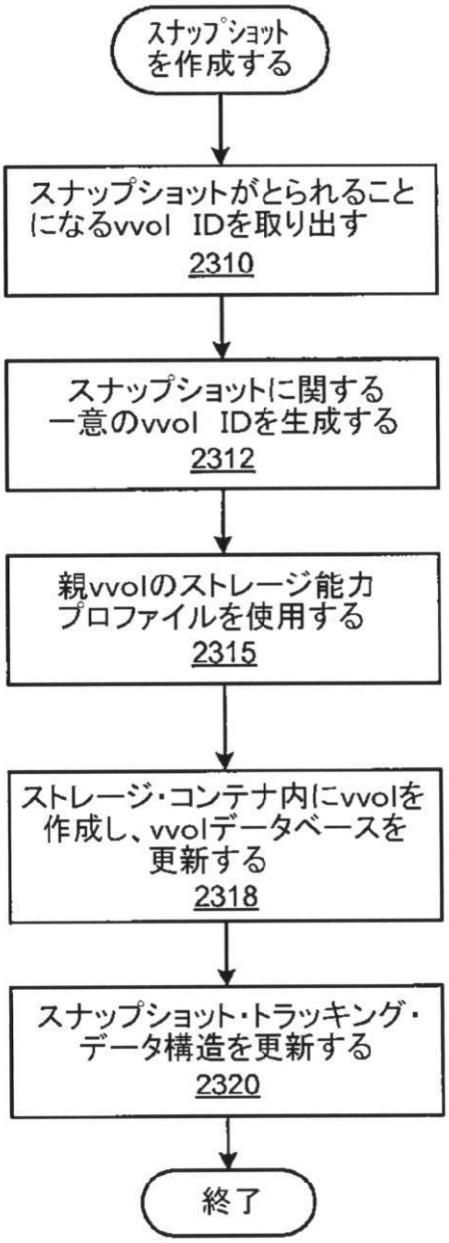
タイプ	名前	（1つまたは複数の）値	スナップ引き継ぎ
固定	複製	真	偽
可変	RTO	0.1, 24, 24	偽
固定	スナップショット	偽	偽
可変	保持	-	-
可変	頻度	-	-
固定	暗号化	-	-



【図２２】


vvolを
作成する
ストレージ・コンテナ内にvvolを
作成したいという要求を受信し、
許可を確かめる
2210
vvolに関する一意のvvol IDを
生成する
2212
カスタムの値か？
2214
いいえ
ストレージ能力プロファイルに
おいてデフォルトの値を使用する
2215
はい
ストレージ能力プロファイルに
おいてカスタムの値を使用する
2216
ストレージ・コンテナ内にvvolを
作成し、vvolデータベースを
更新する
2218
vvol IDを送信する
2220
終了

【図２３】


スナップショット
を作成する
スナップショットがとられることになるvvol IDを取り出す
2310
スナップショットに関する一意のvvol IDを生成する
2312
親vvolのストレージ能力プロファイルを使用する
2315
ストレージ・コンテナ内にvvolを作成し、vvolデータベースを更新する
2318
スナップショット・トラッキング・データ構造を更新する
2320
終了

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/051872

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F3/06 G06F9/00 H04L29/06
ADD.

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)
G06F H04L

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)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2011/070605 A1 (HITACHI LTD [JP]; SUGINO SHOJI [JP]; URABE KIICHIRO [JP]; TANINAKA HIR) 16 June 2011 (2011-06-16) paragraph [0011] paragraphs [0014] - [0024] figure 1 paragraphs [0025] - [0038] figure 2 paragraph [0039] figure 3 paragraphs [0042] - [0044] figure 4 paragraphs [0045] - [0055] figures 5,6 paragraph [0062] figure 8 ----- -/--	1-18

[X] Further documents are listed in the continuation of Box C. [X] 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

"E" earlier application or patent but published on or after the international filing date

"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 reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"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

"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

"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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 November 2012	04/12/2012

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	De Ceulaer, Bart

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/051872

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 01/80013 A1 (STOREAGE NETWORKING TECHNOLOGI [IL]; NAHUM NELSON [IL]) 25 October 2001 (2001-10-25) abstract page 10, line 10 - page 12, line 5 figure 1 -----	1-18
A	US 2010/153947 A1 (HARUMA TOSHIYUKI [JP]) 17 June 2010 (2010-06-17) paragraphs [0076] - [0084] figure 2 paragraph [0090] figure 3G -----	17,18

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2012/051872

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2011070605 A1	16-06-2011	US 2011258377 A1	20-10-2011
		WO 2011070605 A1	16-06-2011
WO 0180013 A1	25-10-2001	AU 4679901 A	30-10-2001
		CA 2405405 A1	25-10-2001
		EP 1282861 A1	12-02-2003
		US 2003236945 A1	25-12-2003
		WO 0180013 A1	25-10-2001
US 2010153947 A1	17-06-2010	JP 4705982 B2	22-06-2011
		JP 2010140273 A	24-06-2010
		US 2010153947 A1	17-06-2010

Form PCT/ISA/210 (patent family annex) (April 2005)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,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,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN

（特許庁注：以下のものは登録商標）
１．イーサネット
２．Ｌｉｎｕｘ

(72)発明者 バト、ラジェシュ
アメリカ合衆国 ９４３０４ カリフォルニア州 パロ アルト ヒルビュー アベニュー ３４０１
(72)発明者 アチャリヤ、サンジェイ
アメリカ合衆国 ９４３０４ カリフォルニア州 パロ アルト ヒルビュー アベニュー ３４０１
(72)発明者 バガーニ、サティヤム ビー.
アメリカ合衆国 ９５１１７ カリフォルニア州 サンノゼ ダフォディル ウェイ ８９３
(72)発明者 シー、チャオーチャン
アメリカ合衆国 ９４３０４ カリフォルニア州 パロ アルト ヒルビュー アベニュー ３４０１