(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2017-531250
(P2017-531250A)
(43)公表日　平成29年10月19日(2017.10.19)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 12/00 (2006.01)	G06F 12/00	533J

審査請求 有　予備審査請求 未請求　（全 42 頁）

(21)出願番号	特願2017-511735 (P2017-511735)
(86)(22)出願日	平成27年7月31日 (2015.7.31)
(85)翻訳文提出日	平成29年4月14日 (2017.4.14)
(86)国際出願番号	PCT/US2015/043159
(87)国際公開番号	WO2016/032688
(87)国際公開日	平成28年3月3日 (2016.3.3)
(31)優先権主張番号	14/473,621
(32)優先日	平成26年8月29日 (2014.8.29)
(33)優先権主張国	米国 (US)

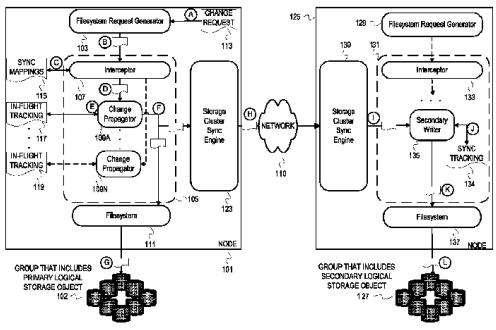
(71)出願人　303039534
ネットアップ，インコーポレイテッド
アメリカ合衆国　カリフォルニア　９４０８９，　サニーヴェール，　イースト　ジャバ　ドライブ　４９５
(74)代理人　100107766
弁理士　伊東　忠重
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100091214
弁理士　大貫　進介
(72)発明者　キンメル，ジェフリー，エス.
アメリカ合衆国　カリフォルニア州　９４０８９　サニーベール　イースト　ジャヴァ　ドライヴ　４９５

最終頁に続く

(54)【発明の名称】粒状同期／半同期アーキテクチャ

(57)【要約】
データ一貫性及び可用性が、クラスタ化された記憶環境内で記憶仮想化を使用する記憶ソリューションにおいて、論理記憶オブジェクトの粒度で提供されることができる。異なる記憶要素にわたりデータの一貫性を確保するために、異なる記憶要素にわたって同期が実行される。データに対する変更が、該変更をプライマリ論理記憶要素からセカンダリ論理記憶要素に伝搬することによって、異なるクラスタ内の記憶要素にわたり同期される。パフォーマンスを維持すると同時に最も厳しいＲＰＯを満足するために、変更要求が、プライマリ論理記憶オブジェクトをホストするファイルシステムに送信される前に傍受され、セカンダリ論理記憶オブジェクトに関連付けられた異なる管理記憶要素に伝搬される。

FIG. 1



【特許請求の範囲】
【請求項１】
記憶プロトコル入力／出力要求からファイルシステム要求を生成した後、前記記憶プロトコル入力／出力要求の対象として示される第１の論理記憶オブジェクトが同期構成データ内に示されるとおりに第２の論理記憶オブジェクトとの同期関係を有すると決定することと、
前記同期構成データ内の前記同期関係について、複数の同期関係タイプのうちいずれが示されているかを決定することと、
前記複数の同期関係タイプのうちの第１の同期関係タイプについて、
前記ファイルシステム要求を追跡するための前記ファイルシステム要求の指標を記録することと、
前記ファイルシステム要求をファイルシステムに供給することであって、前記ファイルシステムは、前記第１の論理記憶オブジェクトをホストする１つ以上の記憶要素にアクセスする、ことと、
前記ファイルシステム要求に基づいて要求を生成することであって、前記の生成された要求は前記第２の論理記憶オブジェクトを対象として示す、ことと、
前記の生成された要求を伝達のために前記第２の論理記憶オブジェクトに関連付けられたノードに供給することと、
前記記憶プロトコル入力／出力要求により示される変更が前記第１の論理記憶オブジェクトに対して成功裏に行われ、前記変更が前記第２の論理記憶オブジェクトに対して成功裏に行われる場合、前記記憶プロトコル入力／出力要求の成功を示す応答を前記記憶プロトコル入力／出力要求の要求元に供給することと、
を含む方法。
【請求項２】
前記複数の同期関係タイプのうちの第２の同期関係タイプについて、
所定の時間間隔において、前記第１の論理記憶オブジェクトを対象にする複数のファイルシステム要求を蓄積することであって、前記複数のファイルシステム要求は前記ファイルシステム要求を含む、ことと、
前記複数のファイルシステム要求の各々を前記ファイルシステムに供給することと、
前記の蓄積された複数のファイルシステム要求のうち、前記第１の論理記憶オブジェクトに対して行われる変更を示すファイルシステム要求間の依存関係を決定することと、
シーケンシング情報を記録して、決定された依存関係を保存することと、
前記第２の論理記憶オブジェクトに関連付けられた前記ノードに、前記シーケンシング情報と、変更を示す前記の蓄積された複数のファイルシステム要求とを供給することであって、該変更は前記第２の論理記憶オブジェクトに対して行われる、ことと、
をさらに含む請求項１に記載の方法。
【請求項３】
前記第１の論理記憶オブジェクトに対して行われる変更を示す前記複数のファイルシステム要求のうち冗長なファイルシステム要求を消去すること、をさらに含む請求項２に記載の方法。
【請求項４】
前記第１の論理記憶オブジェクトに対して行われる変更を示しかつ前記所定の時間間隔内に蓄積された前記複数のファイルシステム要求の総数を示すメタデータを供給すること、をさらに含む請求項２又は３に記載の方法。
【請求項５】
前記第２の論理記憶オブジェクトに関連付けられた前記ノードに供給される前記複数のファイルシステム要求の各々に前記メタデータを記録すること、をさらに含む請求項４に記載の方法。
【請求項６】
前記シーケンシング情報を記録することは、依存関係を有しかつ前記第１の論理記憶オ

ブジェクトに対して行われる変更を示す前記複数のファイルシステム要求の各々に前記シーケンシング情報を記録することを含む、請求項２乃至５のうちいずれか１項に記載の方法。

【請求項７】

前記記憶プロトコル入力／出力要求により示される変更が前記第１の論理記憶オブジェクト又は前記第２の論理記憶オブジェクトのいずれかに対して成功裏に行われない場合、前記記憶プロトコル入力／出力要求の失敗を示す応答を前記記憶プロトコル入力／出力要求の前記要求元に供給すること、をさらに含む請求項１乃至６のうちいずれか１項に記載の方法。

【請求項８】

１つ以上のコンピュータシステムにより実行されるときに、前記１つ以上のコンピュータシステムに請求項１乃至７のうちいずれか１項に記載の方法を実施させるマシン読取可能命令を含むコンピュータプログラム。

【請求項９】

記憶クラスタにわたる論理記憶オブジェクト間の同期関係を維持するコンピュータプログラムであって、当該コンピュータプログラムは、非一時的マシン読取可能媒体内に具現化されるプログラムコードを含み、前記プログラムコードは、

記憶プロトコル入力／出力要求からのファイルシステム要求が生成された後、前記記憶プロトコル入力／出力要求の対象として示される第１の論理記憶オブジェクトが同期構成データ内に示されるとおりに第２の論理記憶オブジェクトとの同期関係を有すると決定し、

前記同期構成データ内の前記同期関係について、複数の同期関係タイプのうちいずれが示されているかを決定し、

前記複数の同期関係タイプのうちの第１の同期関係タイプについて、

前記ファイルシステム要求を追跡するための前記ファイルシステム要求の指標を記録し、

前記ファイルシステム要求をファイルシステムに供給することであって、前記ファイルシステムは、前記第１の論理記憶オブジェクトをホストする１つ以上の記憶要素にアクセスし、

前記ファイルシステム要求に基づいて要求を生成することであって、前記の生成された要求は前記第２の論理記憶オブジェクトを対象として示し、

前記の生成された要求を伝達のために前記第２の論理記憶オブジェクトに関連付けられたノードに供給し、

前記記憶プロトコル入力／出力要求により示される変更が前記第１の論理記憶オブジェクトに対して成功裏に行われ、前記変更が前記第２の論理記憶オブジェクトに対して成功裏に行われる場合、前記記憶プロトコル入力／出力要求の成功を示す応答を前記記憶プロトコル入力／出力要求の要求元に供給する

ように構成される、コンピュータプログラム。

【請求項１０】

前記複数の同期関係タイプのうちの第２の同期関係タイプについて、

所定の時間間隔において、前記第１の論理記憶オブジェクトを対象にする複数のファイルシステム要求を蓄積することであって、前記複数のファイルシステム要求は前記ファイルシステム要求を含み、

前記複数のファイルシステム要求の各々を前記ファイルシステムに供給し、

前記の蓄積された複数のファイルシステム要求のうち、前記第１の論理記憶オブジェクトに対して行われる変更を示すファイルシステム要求間の依存関係を決定し、

シーケンシング情報を記録して、決定された依存関係を保存し、

前記第２の論理記憶オブジェクトに関連付けられた前記ノードに、前記シーケンシング情報と、変更を示す前記の蓄積された複数のファイルシステム要求とを供給することであって、該変更は前記第２の論理記憶オブジェクトに対して行われる

ように構成されたプログラムコード、をさらに含む請求項９に記載のコンピュータプログラム。
【請求項１１】
前記第１の論理記憶オブジェクトに対して行われる変更を示す前記複数のファイルシステム要求のうち冗長なファイルシステム要求を消去するプログラムコード、をさらに含む請求項１０に記載のコンピュータプログラム。
【請求項１２】
前記第１の論理記憶オブジェクトに対して行われる変更を示しかつ前記所定の時間間隔内に蓄積された前記複数のファイルシステム要求の総数を示すメタデータを供給するプログラムコード、をさらに含む請求項１０又は１１に記載のコンピュータプログラム。
【請求項１３】
前記第２の論理記憶オブジェクトに関連付けられた前記ノードに供給されることになる前記複数のファイルシステム要求の各々に前記メタデータを記録するプログラムコード、をさらに含む請求項１２に記載のコンピュータプログラム。
【請求項１４】
前記シーケンシング情報を記録する前記プログラムコードは、依存関係を有しかつ前記第１の論理記憶オブジェクトに対して行われる変更を示す前記複数のファイルシステム要求の各々に前記シーケンシング情報を記録するプログラムコードを含む、請求項１０乃至１３のうちいずれか１項に記載のコンピュータプログラム。
【請求項１５】
前記記憶プロトコル入力／出力要求により示される変更が前記第１の論理記憶オブジェクト又は前記第２の論理記憶オブジェクトのいずれかに対して成功裏に行われない場合、前記記憶プロトコル入力／出力要求の失敗を示す応答を前記記憶プロトコル入力／出力要求の前記要求元に供給するプログラムコード、をさらに含む請求項９乃至１４のうちいずれか１項に記載のコンピュータプログラム。
【請求項１６】
装置であって、
プロセッサと、
プログラムコードを記憶させた非一時的マシン読取可能媒体と、
を含み、
前記プログラムコードは前記プロセッサにより実行可能であって、当該装置に、
記憶プロトコル入力／出力要求からのファイルシステム要求が生成された後、前記記憶プロトコル入力／出力要求の対象として示される第１の論理記憶オブジェクトが同期構成データ内に示されるとおりに第２の論理記憶オブジェクトとの同期関係を有すると決定し、
前記同期構成データ内の前記同期関係について、複数の同期関係タイプのうちいずれが示されているかを決定し、
前記複数の同期関係タイプのうちの第１の同期関係タイプについて、
前記ファイルシステム要求を追跡するための前記ファイルシステム要求の指標を記録し、
前記ファイルシステム要求をファイルシステムに供給することであって、前記ファイルシステムは、前記第１の論理記憶オブジェクトをホストする１つ以上の記憶要素にアクセスし、
前記ファイルシステム要求に基づいて要求を生成することであって、前記の生成された要求は前記第２の論理記憶オブジェクトを対象として示し、
前記の生成された要求を伝達のために前記第２の論理記憶オブジェクトに関連付けられたノードに供給し、
前記記憶プロトコル入力／出力要求により示される変更が前記第１の論理記憶オブジェクトに対して成功裏に行われ、前記変更が前記第２の論理記憶オブジェクトに対して成功裏に行われる場合、前記記憶プロトコル入力／出力要求の成功を示す応答を前記記憶プ

ロトコル入力／出力要求の要求元に供給する
ことを実行させる、装置。
【請求項１７】
前記プログラムコードは、当該装置に、
前記複数の同期関係タイプのうちの第２の同期関係タイプについて、
所定の時間間隔において、前記第１の論理記憶オブジェクトを対象にする複数のファイルシステム要求を蓄積することであって、前記複数のファイルシステム要求は前記ファイルシステム要求を含み、
前記複数のファイルシステム要求の各々を前記ファイルシステムに供給し、
前記の蓄積された複数のファイルシステム要求のうち、前記第１の論理記憶オブジェクトに対して行われる変更を示すファイルシステム要求間の依存関係を決定し、
シーケンシング情報を記録して、決定された依存関係を保存し、
前記第２の論理記憶オブジェクトに関連付けられた前記ノードに、前記シーケンシング情報と、変更を示す前記の蓄積された複数のファイルシステム要求とを供給することであって、該変更は前記第２の論理記憶オブジェクトに対して行われる
ことを実行させるための、前記プロセッサにより実行可能なプログラムコードをさらに含む、請求項１６に記載の装置。
【請求項１８】
前記プログラムコードは、前記第１の論理記憶オブジェクトに対して行われる変更を示す前記複数のファイルシステム要求のうち冗長なファイルシステム要求を消去することを当該装置に実行させるための、前記プロセッサにより実行可能なプログラムコードをさらに含む、請求項１７に記載の装置。
【請求項１９】
前記プログラムコードは、前記第１の論理記憶オブジェクトに対して行われる変更を示しかつ前記所定の時間間隔内に蓄積された前記複数のファイルシステム要求の総数を示すメタデータを供給することを当該装置に実行させるための、前記プロセッサにより実行可能なプログラムコードをさらに含む、請求項１７又は１８に記載の装置。
【請求項２０】
前記プログラムコードは、前記記憶プロトコル入力／出力要求により示される変更が前記第１の論理記憶オブジェクト又は前記第２の論理記憶オブジェクトのいずれかに対して成功裏に行われない場合、前記記憶プロトコル入力／出力要求の失敗を示す応答を前記記憶プロトコル入力／出力要求の前記要求元に供給することを当該装置に実行させるための、前記プロセッサにより実行可能なプログラムコードをさらに含む、請求項１６乃至１９のうちいずれか１項に記載の装置。

【発明の詳細な説明】
【技術分野】
【０００１】
関連出願
本出願は、“GRANULAR SYNC/SEMI-SYNC ARCHITECTURE”と題され2014年8月29日に申請された米国特許出願第14/473,621号に対する優先を主張し、上記出願は本明細書において参照により援用される。
【０００２】
本開示の態様は、概して分散記憶の分野に関し、より詳細には分散ストレージにわたりデータを同期するアーキテクチャに関する。
【背景技術】
【０００３】
企業は、顧客データを維持するか又は企業独自データを維持するかにかかわらず、常に利用可能な又は高度に利用可能なデータと、そのデータの保護とを求める。こうした需要をサポートするために、データはしばしば、複数のサイトにおける複数の記憶システムに

わたり存在し、複数のサイトはしばしば、大きく距離が離れている。上記サイトが長距離離れている理由の１つは、１つの大惨事がデータ可用性に影響するのを回避するためである。可用性要件を定義するのに使用されるメトリクスには、復旧ポイント目標（ＲＰＯ）及び復旧時間目標（ＲＴＯ）が含まれる。企業は、ＲＴＯを、企業が企業データへのアクセスの欠如を許容する最大量の時間として指定する。企業は、ＲＰＯを、中断に起因して失われる可能性のある、時間の観点からのデータの量として指定する。例えば、企業は、ＲＴＯを１５秒として指定することができる。換言すると、企業は、サービス中断又は失敗の時間からそのシステムの完全復旧の時間まで、最大で１５秒を受け入れることになる。ＲＰＯについて、企業は、５秒を指定することができる。このことは、企業が、失敗又は中断の前の５秒内に書き込まれたデータ（例えば、新しい書き込み、更新等）を超えるいかなるデータも失うことを受け入れないことを意味する。

【０００４】

記憶システムにわたる企業の可用性及び保護の需要をサポートするための記憶機構は、様々な名称を付与されており、例えば、スナップショット（snapshotting）、ミラーリング、クローニング、及びレプリケーション（replicating）などである。上記記憶機構の各々は、ストレージ機構及び／又はストレージ製品のプロバイダによってさらに変動することがある。こうした変動にかかわらず、各記憶機構は、企業のデータについての一貫性のあるビューを提供する。

【図面の簡単な説明】

【０００５】

本開示の態様は、添付図面を参照することによって、より理解され得る。

【図１】変更要求に応答して異なるクラスタ内のプライマリ論理記憶オブジェクトとセカンダリ論理記憶オブジェクトとの間のデータ交換を調整する例示的な記憶クラスタ同期エンジンを表す。

【図２】変更要求に応答して異なるクラスタ内のプライマリ論理記憶オブジェクトとセカンダリ論理記憶オブジェクトとの間のデータ交換を調整する例示的な記憶クラスタ同期エンジンを表す。

【図３】完全同期関係においてセカンダリ論理記憶オブジェクトとして構成され、及び半同期関係においてプライマリ論理記憶オブジェクトとして構成された論理記憶オブジェクトの、例示的な動作を表す。

【図４】完全同期関係においてセカンダリ論理記憶オブジェクトとして構成され、及び半同期関係においてプライマリ論理記憶オブジェクトとして構成された論理記憶オブジェクトの、例示的な動作を表す。

【図５】１つ又は複数のクラスタのノードにわたり論理記憶オブジェクト粒度において完全同期及び半同期の双方を提供する一例示的なアーキテクチャを表す。

【図６】変更要求の受信を扱うことと完全同期関係におけるプライマリエンドポイントを対象にする変更要求を扱うこととの例示的な動作のフローチャートを表す。

【図７】論理記憶オブジェクト粒度半同期動作についての例示的な動作のフローチャートを表す。

【図８】クローズされた変更設定ログを処理する例示的な動作のフローチャートを表す。

【図９】同期関係におけるプライマリエンドポイントの記憶要素モジュールからの応答を扱う例示的な動作のフローチャートを表す。

【図１０】クラスタベースの同期エンジンが伝搬器とカウンターパートの同期エンジンとからの要求を処理する例示的な動作のフローチャートを表す。

【図１１】伝搬器インスタンスがセカンダリエンドポイントに対する変更要求への応答を扱う例示的な動作のフローチャートを表す。

【図１２】セカンダリライタが複製要求を扱う例示的な動作のフローチャートを表す。

【図１３】セカンダリライタインスタンスが下層の記憶要素モジュールからの応答を扱う例示的な動作のフローチャートを表す。

【図１４】記憶クラスタベース粒状完全同期及び半同期伝搬エンジンを備えた一例示的な

コンピュータシステムを表す。
【発明を実施するための形態】
【０００６】
下記の説明は、本開示の手法を具現化する例示的なシステム、方法、手法、命令シーケンス、及びコンピュータプログラム製品を含む。しかしながら、説明される本開示の態様は、こうした特定の詳細なしに実施され得ることが理解される。例えば、例示がディスク及びディスクアレイを参照するが、開示の態様はそのように限定されない。開示の態様は、ソリッドステート記憶装置、光学記憶装置、個々の記憶装置の連合体、異なるタイプの記憶装置の組み合わせ等を使用する記憶システムに実装され得る。さらに、多くの例示が、論理記憶オブジェクトのペアを使用して動作を例示する。開示の態様は、プライマリ（primary）及びセカンダリ（secondary）論理記憶オブジェクトペアに限定されず、論理記憶オブジェクトのグループに適用されることができる。例えば、プライマリ論理記憶オブジェクトを複数のセカンダリ論理記憶オブジェクトと同期するようにシステムが構成されることができる。良く知られる命令インスタンス、プロトコル、構造、及び手法は、説明を分かりにくくしないように詳細には図示されていない。
【０００７】
用語
本説明は、用語「記憶要素」を使用して、データをホストし及び／又はデータに対するアクセスを管理する記憶システム内の任意のエンティティを参照する。本明細書において参照される記憶要素は、管理記憶要素及びホスト記憶要素として分類されることができる。管理記憶要素とホスト記憶要素との間の区別は、記憶要素の基本機能性（primary functionality）から生じる。管理記憶要素は、ホスト記憶要素に対するアクセスを主に管理する。管理記憶要素は、他の装置（例えば、クライアント）からの要求を処理し、動作（例えば、スナップショット動作）を実行するための要求を発する（originate）ことができる。要求が別の装置からであるか又は管理記憶要素から発せられるかにかかわらず、管理記憶要素は、要求をホスト記憶要素に伝達する。管理記憶要素の例には、ファイルサーバ及び記憶コントローラが含まれる。ホスト記憶要素は、管理記憶要素の観点から要求を最終的に満たす動作を主に実行する。ホスト記憶要素は、管理記憶要素からの要求により指定される場所の読み出し又は該場所への書き込みを実行する。この読み出し又は書き込みは、一ディスク又は複数ディスクに対して実行されてよい。仮想化の複数レイヤの場合、読み出し又は書き込みは、管理記憶要素の観点から１つ又は複数のディスクであるように見えるものに対して実行されてよい。ホスト記憶要素の例には、ディスクドライブ、光学ドライブ、記憶アレイ、及びテープドライブが含まれる。
【０００８】
用語の管理記憶要素及びホスト記憶用は記憶要素の基本機能性に基づいて使用され、なぜならば、機能性が要素間で排他的ではないからである。例えば、記憶コントローラが、キャッシュ内にローカルに記憶されたデータを有して、アクセス要求の扱いを促進してもよい。記憶コントローラがアクセス要求を満たすことができるとしても、記憶コントローラの基本機能性は、ローカルメモリからデータを読み出すことやローカルメモリにデータを書き込むことではない。同様に、ホスト記憶要素が、ディスクに対するアクセスを管理するハードウェアを含むことができる。例えば、ＲＡＩＤ（redundant array of independent disks）コントローラとディスクのアレイとが、単一の筐体内に収納されることができる。ＲＡＩＤコントローラはディスクのアレイに対するアクセスを管理するが、上記単一筐体内に収納されるコンポーネントの基本機能性は、管理記憶要素から受信される要求を満たすことである。
【０００９】
本説明は、用語の完全同期（full synchronization）（「完全同期（full sync）」）及び半同期（semi-synchronization）（「半同期（semi sync）」）をさらに使用する。上記用語は、異なるタイプの同期構成を参照する。「完全同期」構成は、本明細書において使用されるとき、変更要求が実行されたことを確認するリプライの送信を、該変更が

プライマリ論理記憶オブジェクトとセカンダリ論理記憶オブジェクトとにわたり同期されるまで遅延させる構成を参照する。「半同期」構成は、本明細書において使用されるとき、変更要求が実行されたことを確認するリプライが、プライマリ論理記憶オブジェクトに対して該変更が実行された後、（１つ以上の）セカンダリ論理記憶オブジェクトとの同期が依然として進行中であり得る間に送信されることを可能にする構成を参照する。

【００１０】

本説明は、用語「要求」を使用して、何らかが行われることを要求するソフトウェアエンティティ又はハードウェアエンティティ間における通信を参照し、様々なプロトコルにおいて使用される名称、データフィールド等のバリエーションを回避する。要求は、データが読み出されること、データが書き込まれることの要求、又は何らかの他のデータ処理要求を示すことができる。要求は、動作のタイプ（例えば、読み出し、書き込み）、要求の対象（例えば、論理記憶オブジェクト識別子）、及び要求元（requestor）の識別子を示すことができる。さらなる情報が、統制するプロトコルに依存して要求の中に示されてもよい。しかし、本説明は、こうしたさらなる情報の詳細を掘り下げない。さらに、複数のプロトコルが、プロトコルスタックといわれるものを形成することができる。プロトコルスタックは、要求が通過し又は横断する一連の処理モジュールとみなされることができる。プロトコルスタックの各レイヤにおいて、ヘッダ及び／又はトレーラが要求に追加され又は要求から削除されてもよい。本説明において、少なくともいくつかのスタック処理は、説明にさらなる複雑さを加えるのを回避するために、説明されない。本説明は、要求を、関連付けられたヘッダ又はトレーラにかかわらない、かつヘッダ及び／又はトレーラ内の値に対するとり得る修正にかかわらない要求として、参照することになる。

【００１１】

導入

クラスタリングは、一般に、ハードウェア要素を一緒にグループ化することを参照し、個々のハードウェア要素から得ることができないハードウェア要素（例えば、ディスクドライブ、記憶アレイ、ファイルサーバ、記憶コントローラ等）のグループ（「クラスタ」）の恩恵を得る。クラスタリングは、様々な記憶機構に対して使用することができ、その例には、ロードバランシング、フェイルオーバーサポート、Ｉ／Ｏ帯域幅の増大、及びデータ可用性が含まれる。こうした記憶の態様をサポートし、一貫性のある記憶のビューを提供するために、サポート記憶要素間でデータが同期される。データを要求元に供給し、要求元のためにデータを修正するのに、いずれの記憶要素が（例えば、構成によって）最初に及び／又は優先的に使用されるかに依存して、種々のハードウェア記憶要素がプライマリ記憶要素及びセカンダリ記憶要素としばしばいわれる。さらに、記憶要素の一クラスタがプライマリクラスタとして指名されることができ、記憶要素の一クラスタがセカンダリクラスタとして指名されることができる。

【００１２】

多くの記憶システム機能性が、記憶仮想化の特徴として展開される。しばしば、記憶仮想化ソフトウェア／ツールが、記憶システムを構成する実際のハードウェア要素を分かりにくくする。ゆえに、要求元（ときに、本明細書においてクライアントといわれる）は、しばしば、論理記憶オブジェクト又は論理記憶コンテナから読み出し及び書き込みをし、その例には、論理ユニット番号（ＬＵＮ）、ファイル、仮想マシンディスク（ＶＭＤＫ）、仮想ボリューム、及び論理パーティションが含まれる。仮想化の任意数のレイヤが、実際の記憶システムハードウェア要素と、アクセス要求を送信するクライアントとを分離することができる。各記憶システムハードウェア要素は、多数の論理記憶オブジェクト及び／又は論理記憶オブジェクトの多数の部分をホストすることができる。さらに、クライアントのために要求を扱う記憶コントローラが、物理記憶アレイであるように見える仮想記憶アレイと通信してもよい。ゆえに、記憶アレイであるかのように提示される論理記憶オブジェクトが、複数の論理記憶オブジェクトをホストするものとして提示されることができる。

【００１３】

概説

データ一貫性及び可用性が、クラスタ化された記憶環境内で記憶仮想化を使用する記憶ソリューションにおいて、論理記憶オブジェクトの粒度で提供されることができる。可用性について、データは、前述されたとおり、異なるサイトにおける異なる記憶要素上で維持される。異なる記憶要素にわたりデータの一貫性を確保するために、異なる記憶要素にわたって同期が実行される。論理記憶オブジェクトの粒度において、離れたサイトにおける異なる記憶要素にわたりデータが効率的に同期されることができ、なぜならば、少なくとも部分的に、同期されるデータの量がより小さくなり、データを運ぶネットワーク内での否定的インシデントの影響をより受けにくくなるからである。データに対する変更が、該変更をプライマリ論理記憶オブジェクト（すなわち、変更要求の中に指定された論理記憶オブジェクト）に関連付けられたノードからセカンダリ論理記憶オブジェクト（すなわち、同期についてプライマリ論理記憶オブジェクトに関連付けられた論理記憶オブジェクト）に伝搬することによって、異なるクラスタ内の記憶要素にわたり同期される。パフォーマンスを維持すると同時に最も厳しいＲＰＯ（例えば、ＲＰＯ＝０）及びＲＴＯを満足するために、ファイルシステム要求が、プライマリ論理記憶オブジェクトをホストするファイルシステム（「プライマリファイルシステム」）に送信される前に傍受され（intercepted）、セカンダリ論理記憶オブジェクトに関連付けられたノードのファイルシステム（「セカンダリファイルシステム」）に伝搬される。論理記憶オブジェクトは、少なくともいくらかの関連クラスタ内で排他的である不変識別子を有して、クラスタにわたり論理記憶オブジェクトの有効な識別を可能にする。ファイルシステム要求が記憶プロトコル固有要求から生成された後、それがプライマリファイルシステムに送信される前に該ファイルシステム要求を傍受することは、記憶プロトコル固有及び／又はアプリケーション固有の動作を伝搬動作に負わせることを回避し、このことは、伝搬される変更要求のサイズと処理動作の数とをさらに低減させる。ファイルシステムと直接インターフェースをとる同期をサポートする動作を扱うエンティティを有することで、ファイルシステム応答の効率的な伝達についてファイルシステムのメカニズムを活用する。

【００１４】

例示

図１～４に表される例示は、システムに関して圧倒的な量の情報を提示するのを回避する試みとして、異なる度合の例示的な詳細を表す。あらゆるとり得るデータ構造とあらゆるとり得る機能性のモジュール化とが提示されるわけではなく、なぜならば、それらは多数であり、本開示の態様の理解に必要ではないからである。例えば、複数のデータ構造として提示されるデータ構造が、様々なとり得る索引／アクセススキームとデータの配置とを用いて、別様に編成されることができる。同様に、例示の中の個々のモジュール／エンジン／ユニットとして提示される機能性もまた、プラットフォーム（オペレーティングシステム及び／又はハードウェア）、アプリケーションエコシステム、インターフェース、プログラマの選好、プログラミング言語等のうち任意のものに従って、別様に編成されることができる。さらに、いくつかの機能性が、同様に圧倒的な量の情報を提示するのを回避する試みとして、本説明の中で後から説明される。例えば、管理エンティティからのスナップショット要求、又は半同期構成が、プライマリ管理記憶要素における複数のアクセス要求のシーケンシング（sequencing）につながることがある。シーケンシングは、前の方の例示の中では論じられない。

【００１５】

図１及び図２は、変更要求に応答して異なるクラスタ内のプライマリ論理記憶オブジェクトとセカンダリ論理記憶オブジェクトとの間のデータ変更を調整する例示的な記憶クラスタ同期エンジンを表す。図１は、プライマリ管理記憶要素からセカンダリ管理記憶要素への変更要求の伝搬を表す。管理記憶要素は、簡潔さのため、以降はノードといわれる。図１において、第１のクラスタがプライマリノード１０１を含み、第２のクラスタがセカンダリノード１２５を含む。図の簡素化のため、及び描画空間制約に起因して、クラスタ全体は表されていない。プライマリノード１０１は、論理記憶オブジェクトのグループ１

０２をホストするホスト記憶要素に通信可能に結合される。グループ１０２は、プライマリ論理記憶オブジェクトを含む。プライマリノード１０１は、ファイルシステム要求生成器１０３、変更伝搬エンジン１０５、記憶クラスタ同期エンジン１２３、及びファイルシステム１１１を含む。ファイルシステム要求生成器１０３は、記憶プロトコルベースの要求からファイルシステム要求を生成する。変更伝搬エンジン１０５は、傍受器（interceptor）１０７、変更伝搬器（change propagator）１０９Ａ、及び変更伝搬器１０９Ｎを含む。プライマリノード１０１の中のこれらモジュールは、同期マッピング１１５、飛行中追跡データ（in-flight tracking data）１１７、及び飛行中追跡データ１１９として図１に表されるデータにアクセスする。同期マッピング１１５は、論理記憶オブジェクト間における同期構成を示す（本明細書において、同期関係（synchronization relationships）又は同期関係（sync relationships）ともいわれる）。例えば、プライマリ論理記憶オブジェクトが、１つのセカンダリ論理記憶オブジェクトとの完全同期関係と、別のセカンダリ論理記憶オブジェクトとの半同期関係とを有することができる。飛行中追跡データは、要求の進捗又は状態を、対応する変更伝搬器の観点から追跡する。換言すると、各変更伝搬器インスタンスは、飛行中追跡データを、同期関係を有する対応した論理記憶オブジェクトについて維持する。

【００１６】

セカンダリノード１２５は、プライマリノード１０１と同じモジュール／エンジンのすべてを含むことができる。図１において、モジュールのうちいくつかは、繰り返しを低減させるよう表されていない。セカンダリノード１２５は、ファイルシステム要求生成器１２９、記憶クラスタ同期エンジン１３９、変更伝搬エンジン１３１、及びファイルシステム１３７を含むものとして表されている。変更伝搬エンジン１３１は、傍受器１３３及びセカンダリライタ１３５を含む。セカンダリノード１２５のセカンダリライタ１３５は、同期追跡データ（sync tracking data）１３４として図１に表されるデータにアクセスする。同期追跡データ１３４は、要求の進捗又は状態を、セカンダリライタ１３５の観点から示す。同期追跡データ１３４は、必ずしも変更伝搬エンジン１３１内に含まれない。同期追跡データ１３４は、本説明のためにセカンダリライタ１３５の近くに単に表されている。セカンダリノード１２５は、論理記憶オブジェクトのグループ１２７をホストするホスト記憶要素に通信可能に結合される。グループ１２７は、セカンダリ記憶オブジェクトを含む。

【００１７】

異なるノードにわたり機能性において様々なバリエーションがあり得るが、同じ名称を有するモジュールの機能性は一般に本例示において同じである。ファイルシステム要求生成器１０３、１２９は、ファイルシステム生成器１０３、１２９に渡される記憶プロトコル入力／出力（Ｉ／Ｏ）要求に基づいてファイルシステム要求を生成する。ファイルシステム生成器１０３、１２９は、記憶プロトコルＩ／Ｏ要求を、ネットワークスタック、スモールコンピュータシステムインターフェース（ＳＣＳＩ）スタック、インターネットＳＣＳＩ（ｉＳＣＳＩ）モジュール等から受信することができる。記憶プロトコルＩ／Ｏ要求の例には、ストレージエリアネットワーク（ＳＡＮ）要求及びネットワークアタッチトストレージ（ＮＡＳ）要求が含まれる。ファイルシステム生成器１０３、１２９は、そのノード上に実装されるファイルシステムに基づいてファイルシステム要求を生成する。傍受器１０７、１３３は、ファイルシステム要求生成器１０３、１２９からの要求を傍受する。傍受は、別様に実装されることができる。呼び出し元に提示されるインターフェースを変更することなく下層の機能性が変わるように、アプリケーションプログラミングインターフェースが修正されることができる。別の例として、監視処理が、実行キューを監視し、指定されたアドレスが実行キュー内で発生するときに呼び出しをリダイレクトすることができる。ファイルシステム１１１、１３７は、ファイルシステム要求に従って下層のホスト記憶要素にアクセスする。記憶クラスタ同期エンジン１２３、１３９は、ネットワーク１１０を介して実装されるプロトコルに従って通信を処理する。例として、エンジン１２３、１３９により実装されるプロトコルは、ファイバチャネル（ＦＣ）、ファイバチ

ャネルオーバーイーサネット（登録商標）（ＦＣｏＥ）、インターネットファイバチャネルプロトコル（ｉＦＣＰ）、及びトンネリングプロトコルのうち任意の１つ以上であり得る。特定のプロトコルにかかわらず、エンジン１２３、１３９は、マシン間の距離及びホップにかかわらずマシン間の直接接続として認識できるアクティブな接続をサポートするプロトコルを実装する。

【００１８】

図１は、文字Ａ～Ｎによって識別される一連の段階（stages）を備えた例示的な動作を表す。文字によって示唆される動作の順序付けは本例示に限られ、請求項の範囲を限定するのに使用されるべきではない。段階Ａにおいて、プライマリノード１０１が、変更要求１１３を受信する。変更要求１１３は、クライアントから発せられ、クライアントは、管理ノード（例えば、クラスタマネージャ）、ユーザノード（例えば、顧客のサーバ）等にあり得る。ファイルシステム要求生成器１０３が、変更要求１１３を処理し、変更要求に基づいてファイルシステム要求を生成し、ファイルシステム要求１１３をファイルシステム１１１に渡すコードを起動する。ファイルシステム要求１１３を生成することの一部として、ファイルシステム要求生成器１０３は、変更要求の中に対象として示される論理記憶オブジェクト識別子を、論理記憶オブジェクトのファイルシステム場所情報（例えば、ｉｎｏｄｅ識別子、オフセット等）に変換する。しかし、ファイルシステム要求生成器１０３は、ファイルシステム要求と共に進むように、上記論理記憶オブジェクト識別子をさらに示す。論理記憶オブジェクト識別子は、異なる仕方でファイルシステム要求と共に進むことができる。例えば、ファイルシステム生成器は、論理オブジェクト識別子をファイルシステム要求のメタデータに書き込むことができる。別の例として、ファイルシステム生成器は、データ構造を作成し、それをファイルシステムに関連付ける。ファイルシステム１１１がファイルシステム要求を受信することに代わって、傍受器１０７が段階Ｂにおいてファイルシステム要求を受信する。様々な形式における要求（例えば、記憶プロトコルＩ／Ｏ要求、ファイルシステム要求等）はラベル１１３ではもはや識別されず、なぜならば、要求されている変更が要求の形式にかかわらず同じであるからである。

【００１９】

段階Ｃにおいて、傍受器１０７は、同期マッピング１１５にアクセスして、ファイルシステム要求に関連する何らかの同期関係を決定する。ファイルシステム要求は、（ファイルシステム場所情報の観点で）ファイルシステム要求の対象であるグループ１０２の中の論理記憶オブジェクトを示す。傍受器１０７は、同期マッピング１１５にアクセスして、ファイルシステム要求対象について定義されたいくらかの同期関係を決定する。対象は、単一の同期関係を有してもよく、複数の同期関係を有してもよく、あるいは同期関係を有さなくてもよい。対象が同期関係を有さない場合、ファイルシステム要求はファイルシステム１１１に渡されることになる。本例示について、同期マッピング１１５は、対象がグループ１２７の中の論理記憶オブジェクトとの完全同期関係を有することを示す。ファイルシステム要求の対象が同期関係を有するため、ファイルシステム要求の対象はプライマリ論理記憶オブジェクトとみなされることができる。前述されたとおり、論理記憶オブジェクトは、少なくとも互いに関連付けられたクラスタにわたり排他的である不変識別子によって識別される。同期マッピングは、１つ以上のデータ構造内に示されることがあり、同期マッピングは、論理オブジェクト（例えば、ファイル、ＬＵＮ等）及び下層のファイルシステムに依存して、論理オブジェクト又はファイルシステム要求対象の複数レベル又は複数レイヤにわたり同期関係をマッピングする。例えば、論理オブジェクトはファイルであり得る。論理オブジェクト識別子は、最初、ファイル識別子又はファイルハンドルであることになる。ファイルシステムは、ファイルハンドルを影響されるデータブロックに向けて書き込み要求を解決する。ファイルシステムは、例えば、任意数のｉｎｏｄｅレベルをとおして解決してもよい。同期関係が存在するとき、同期マッピングは、プライマリノードにおけるより高いレベルの識別子（すなわち、論理オブジェクト識別子）をセカンダリノードにおけるより高いレベルの識別子にマッピングするだけでなく、さらに、同期マッピングは、より低いレベルの識別子（すなわち、ファイルシステム場所情報）をマッ

ピングする。この例の場合、より低いレベルの識別子は、ｉｎｏｄｅ識別子であることになる。対象にされているファイルの一部のためのプライマリノードｉｎｏｄｅ識別子は、対象にされているファイルの一部のためのセカンダリノード上のｉｎｏｄｅ識別子にマッピングすることになる。

【００２０】

段階Ｄにおいて、傍受器１０７は、ファイルシステム要求と対象に関する同期関係の指標とを変更伝搬器１０９Ａに渡す。プライマリノード１０１が、変更要求１１３の中に示されるのと同じプライマリ論理記憶オブジェクトを対象にする変更要求をまだ受信していなかった場合、傍受器１０７は、変更伝搬器１０９Ａをインスタンス化するコードを起動してもよい。必要ではないが、変更伝搬器は、この例示においてプライマリ論理記憶オブジェクトごとにインスタンス化される。傍受器１０７は、プライマリ論理記憶オブジェクトの同期関係を変更伝搬器に対して様々な仕方で示すことができる。例えば、傍受器１０７は、パラメータ値としてのプライマリ論理記憶オブジェクト識別子とパラメータ値としてのセカンダリ論理記憶オブジェクト識別子とを用いて、変更伝搬器をインスタンス化する関数を呼び出すことができる。別の例として、傍受器１０７は、すでにインスタンス化された変更伝搬器１０９Ａに対して、ローカルメモリに記憶されたファイルシステム要求への参照と共に、プロセス間通信を送信することができる。プライマリ論理記憶オブジェクトごとの変更伝搬器のインスタンス化を例示するために、変更伝搬器１０９Ｎは、飛行中追跡データ１１９に対する破線と共に表されている。破線が使用されて、変更伝搬器１０９Ｎが異なるファイルシステム要求のための飛行中追跡データ１１９にアクセスしている可能性があることを示す。

【００２１】

段階Ｅにおいて、変更伝搬器１０９Ａは、同期関係のセカンダリ論理記憶オブジェクトを対象にするファイルシステム要求を作成し、飛行中追跡データ１１７を更新する。変更伝搬器１０９Ａがちょうどインスタンス化されたところである場合、追跡データのための構造がまだ存在しない可能性があり、あるいは空の構造が存在する可能性がある。変更伝搬器１０９Ａは、飛行中追跡データ１１７を更新して、プライマリ論理記憶オブジェクトを対象にするファイルシステム要求が飛行中である（すなわち、送信されることになるか又は送信されている）ことを示す。変更伝搬器１０９Ａは、飛行中追跡データ１１７を更新して、セカンダリ論理記憶オブジェクトを対象にするファイルシステム要求が飛行中であることをさらに示す。変更伝搬器１０９Ａは、それから（又は同時に）、プライマリ論理記憶オブジェクトとの完全同期関係を有するセカンダリ論理記憶オブジェクトの識別子を備えた要求を作成する。変更伝搬器１０９Ａは、異なる要求元を有するファイルシステム要求を同様に作成する。変更伝搬器１０９Ａは、変更伝搬器１０９Ａを要求元として示す。変更伝搬器１０９Ａは、任意の関連付けられたクラスタ内で変更伝搬器１０９Ａを排他的に識別する様々なデータを用いて識別されることができ、例えば、変更伝搬器１０９Ａのプロセス／スレッド識別子とプライマリノード１０１のネットワークアドレスとの組み合わせなどである。変更伝搬器１０９Ａは、プライマリ論理記憶オブジェクト識別子を要求元の指標にさらに組み込むことができる。変更伝搬器１０９Ａから送信されるプライマリ論理記憶オブジェクトを対象にするファイルシステム要求は、プライマリ変更要求といわれる。変更伝搬器１０９Ａから送信されるセカンダリ論理記憶オブジェクトを対象にするファイルシステム要求は、セカンダリ変更要求といわれる。

【００２２】

段階Ｆにおいて、変更伝搬器１０９Ａは、サービス提供のためのファイルシステム要求を送信する。プライマリ論理記憶オブジェクトがセカンダリ論理記憶オブジェクトとの完全同期関係を有するため、プライマリノード１０１は、変更がプライマリ及びセカンダリ双方の論理記憶オブジェクトにおいて行われるまで、変更要求１１３に応答しないことになる。したがって、変更伝搬器１０９Ａは、プライマリ及びセカンダリ変更要求を任意の順序で送信することができる。変更伝搬器１０９Ａは、プライマリ変更要求をファイルシステム１１１に送信する。変更伝搬器１０９Ａは、セカンダリ変更要求を記憶クラスタ同

期エンジン１２３に送信する。変更要求が変更伝搬器１０９Ａから渡された後、動作のタイミングは、ネットワーク条件、ノード能力における差等に依存して変動する可能性がある。

【００２３】

段階Ｇにおいて、ファイルシステム１１１は、ホスト記憶要素にアクセスする。段階Ｈにおいて、記憶クラスタ同期エンジン１２３は、ネットワーク１１０を横断する記憶クラスタ同期エンジン１２３と記憶クラスタ同期エンジン１３９との間の接続のプロトコルに従って、セカンダリ変更要求を処理する。記憶クラスタ同期エンジン１２３は、接続プロトコルに従って新しい要求を構築し、この新しい要求をセカンダリ変更要求からの関連情報（例えば、セカンダリ論理記憶オブジェクト識別子、書き込まれるべきデータ等）で埋めることができる。記憶クラスタ同期エンジン１２３は、セカンダリ変更要求を、接続プロトコルに準拠したヘッダでカプセル化してもよい。この例示について、プライマリノードにおける同期マッピングは、プライマリノードとセカンダリノードとの間で論理オブジェクト識別子（例えば、ファイルハンドル）をマッピングし、さらに、ファイルシステム場所情報（例えば、ｉｎｏｄｅ識別子）をマッピングする。セカンダリ変更要求は、変更要求によって影響されるデータブロックのセカンダリノードファイルシステム場所情報を用いて構築される。いくつかの場合、ファイルシステム場所情報同期マッピングは、論理オブジェクト識別子同期マッピングとは別個であることになる。また、ファイルシステム場所情報同期マッピングは、セカンダリノードにおいて維持されてもよい。こうした場合、セカンダリ変更要求は、対象にされた論理オブジェクトの指標とプライマリノードのファイルシステム場所情報とを用いて構築される。受信されたとき、セカンダリノードは、同期マッピングにアクセスし、セカンダリノードファイルシステム場所情報に対するプライマリノードファイルシステム場所情報を解決することになる。

【００２４】

段階Ｉにおいて、記憶クラスタ同期エンジン１３９が、接続プロトコルに従って受信した要求を処理し、セカンダリ変更要求をセカンダリライタ１３５に渡す。記憶クラスタ同期エンジン１３９は、受信した要求からセカンダリ変更要求を再構築し、あるいは受信した要求からセカンダリ変更要求を抽出することができる。セカンダリ変更要求がまだ受信されていなかった場合、記憶クラスタ同期エンジン１３９は、セカンダリライタ１３５をインスタンス化するコードを起動してもよい。記憶クラスタ同期エンジン１３９は、記憶クラスタ同期エンジン１３９により受信されるすべてのセカンダリ変更要求を扱うセカンダリライタをインスタンス化し、あるいはプライマリ論理記憶オブジェクト及びセカンダリ論理記憶オブジェクトのペアごとにこれらをインスタンス化することができる。

【００２５】

図１は、ファイルシステム要求生成器１２９から及び傍受器１３３からの破線を表している。ファイルシステム要求生成器１２９からの破線は、ファイルシステム要求生成器１２９が他の変更要求を受信し、処理し、傍受器１３３に渡している可能性を示す。傍受器１３３から省略記号への破線は、傍受器１３３が変更要求を傍受し、セカンダリノード１２５の変更伝搬器に渡している可能性を例示しており、該変更伝搬器は図示されていない。こうした可能性は、セカンダリノード１２５がセカンダリ変更要求を扱うことに限定されないことを示すために例示される。

【００２６】

段階Ｊにおいて、セカンダリライタ１３５が、同期追跡データ１３４を更新する。セカンダリライタ１３５は、対象にされたセカンダリ論理記憶オブジェクトと要求元（すなわち、変更伝搬器１０９Ａ）とセカンダリ変更要求の状態とを少なくとも含むセカンダリ変更要求の指標を記録する。この時点で、セカンダリライタ１３５は状態を飛行中として記録し、なぜならば、セカンダリ変更要求は送信されているか又は送信されることになるからである。段階Ｋにおいて、セカンダリライタ１３５は、セカンダリ変更要求をファイルシステム１３７に送信する。

【００２７】

段階Ｌにおいて、ファイルシステム１３７が、セカンダリ変更要求に従ってホスト記憶要素にアクセスする。

【００２８】

図２は、図１の同期マッピングにおいて定義される完全同期関係に従って処理されるプライマリ及びセカンダリ変更要求に対する応答を表す。図２は、段階ラベルＡ～Ｌを用いて例示的な動作を表す。段階Ａ～Ｊは、セカンダリノード１２５の前にプライマリ論理記憶オブジェクトのホスト記憶要素からの応答が応答する場合として表されている。しかしながら、その順序付けは必要ではない。いくつかの場合、プライマリ論理記憶オブジェクトのホスト記憶要素がプライマリノード１０１に応答することができる前に、セカンダリノード１２５が変更伝搬器１０９Ａに応答することができる可能性がある。応答のタイミングにかかわらず、プライマリ及びセカンダリ双方の論理記憶オブジェクトにおける変更が変更伝搬器１０９Ａによって確認されるまで、要求元に対する応答は提供されない。図１からのいくつかの要素が、図２を簡素化するために除去されている。

【００２９】

段階Ａ～Ｃは、プライマリ論理記憶オブジェクトのホスト記憶要素から変更伝搬器１０９Ａに進む応答と、飛行中追跡データ１１７の対応する更新とを例示する。段階Ａにおいて、プライマリ論理記憶オブジェクトをホストするホスト記憶要素が、ファイルシステム１１１に応答を供給する。ファイルシステム１１１は、段階Ｂにおいてこの応答を変更伝搬器１０９Ａに転送する。段階Ｃにおいて、変更伝搬器１０９Ａは、飛行中追跡データ１１７を更新して、プライマリ変更要求がプライマリ論理記憶オブジェクトにおいて実行されたことを示す。

【００３０】

段階Ｄ～Ｊは、セカンダリ論理記憶オブジェクトのホスト記憶要素から変更伝搬器１０９Ａに進む応答と、飛行中追跡データ１１７の対応する更新とを例示する。段階Ｄにおいて、セカンダリ論理記憶オブジェクトをホストするホスト記憶要素が、応答をファイルシステム１３７に供給する。ファイルシステム１３７は、段階Ｅにおいてこの応答をセカンダリライタ１３５に転送する。段階Ｆにおいて、セカンダリライタ１３５は、同期追跡データを更新して、セカンダリ論理記憶オブジェクトに対する更新を反映する。例えば、セカンダリライタ１３５は、セカンダリ論理記憶オブジェクト識別子と転送される応答の要求元との組み合わせを使用して、同期追跡データ１３４をホストする構造の中のエントリをルックアップする。セカンダリライタ１３５は、上記エントリの中に値又はフラグを設定して、セカンダリ論理記憶オブジェクトに対して変更が完了したことを示す。セカンダリライタ１３５は、それから、応答を記憶クラスタ同期エンジン１３９に転送する。記憶クラスタ同期エンジン１３９は、セカンダリ変更要求に対する応答（「セカンダリ応答」）がプライマリノード１０１に対して送信されるべきかを決定する。記憶クラスタ同期エンジン１３９は、接続プロトコルに従ってセカンダリ応答を処理し、段階Ｈにおいてネットワーク１１０を介して接続を通じてセカンダリ応答を送信する。段階Ｉにおいて、記憶クラスタ同期エンジン１２３が、接続プロトコルに従ってセカンダリ応答を処理し、セカンダリ応答を変更伝搬器１０９Ａに転送する。セカンダリ応答の処理の一部として、記憶クラスタ同期エンジン１２３は、変更伝搬器１０９Ａのプロセス／スレッド識別子を組み込んでいる要求元識別子に基づいて、上記セカンダリ応答が変更伝搬器Ａに送信されるべきかを決定することができる。段階Ｊにおいて、変更伝搬器１０９Ａは、飛行中追跡データ１１７を更新して、セカンダリ変更要求がセカンダリ論理記憶オブジェクトにおいて実行されたことを示す。

【００３１】

最初の変更要求１１３に対応するすべての未処理の変更要求が完了したと決定した後、変更伝搬器１０９Ａは、応答をファイルシステム要求生成器１０３に供給する。変更伝搬器１０９Ａが飛行中追跡データ１１７を更新するたび、変更伝搬器１０９Ａは、例えば、エントリを読み出して、エントリ内に示されるすべての要求が完了したか又は依然として飛行中であるかを決定することができる。この例示について、ファイルシステム要求生成

器１０３は、変更要求１１３に対応する要求元を示すデータを維持する。要求がファイルシステム要求生成器１０３によって最初に受信されるとき、この要求は、要求元に対応する要求識別子でタグ付けされることができる。この要求識別子は、要求及び対応する応答と共に進むことができる。要求識別子は、要求元のアイデンティティと要求とを示して、該要求を同じ要求元からの他の要求から区別する。変更伝搬エンジン１０５は、変更要求１１３の要求元を示し及び変更要求１１３それ自体を示すデータを、さらに（又は代わって）維持するようにプログラムされることができる。段階Ｌにおいて、ファイルシステム要求生成器１０３は、変更応答２１３を形成し、変更応答２１３を対応する要求元に供給する。

【００３２】

論理記憶オブジェクト間でとり得る同期関係の組み合わせのさらなる一例示として、図３～４は、完全同期関係においてセカンダリ論理記憶オブジェクトとして構成され及び半同期関係においてプライマリ論理記憶オブジェクトとして構成された論理記憶オブジェクトについての例示的な動作を表す。論理記憶オブジェクトの異なる観点を提供するために、図３～４は、ホスト記憶要素のクラスタの文脈において論理記憶オブジェクトを表す。論理記憶オブジェクトは、ホスト記憶要素（例えば、記憶アレイ）上に破線を用いて表される。論理記憶オブジェクトがこのように表されて、論理記憶オブジェクトが複数のホスト記憶要素に及ぶことと、単一のホスト記憶要素内にホストされることとの可能性を例示する。ホスト記憶要素が、ホスト記憶要素の集合（例えば、ディスクアレイ）である場合、論理記憶オブジェクトは、ディスクアレイ内の複数ディスクに及ぶ可能性がある。図３は、ノード３０１に関連付けられた記憶クラスタ３０３を表す。図３～４は、ノード３１１に関連付けられた記憶クラスタ３２５と、ノード３２９に関連付けられた記憶クラスタ３３１とを表す。図３～４は、ネットワーク３０９を介して通信するノードを表す。ノード３０１は、図１のノード１０１と同様に動作し、ゆえにその動作は、例示詳細について、図１内と同じレベルでは表されない。同様に、ノード３２９は、図１及び２のセカンダリノード１２５と同様に動作し、ゆえにその例示的な動作もまた、この例示に対して全体では繰り返されない。

【００３３】

図３～４は、図１に表されるモジュールのうちいくつかを有するノード３１１を表す。再びになるが、繰り返しを回避するため、モジュールのすべてが繰り返されてはいない。図３～４において、ＮＯＤＥ＿２として識別されるノード３１１が、セカンダリライタ３１５、ファイルシステム３２１、及び記憶クラスタ同期エンジン３１３を含む。図３～４は、例示的な同期関係を有する同期マッピング３１７としてノード３１１内に同期マッピングデータをさらに表している。さらに、ノード３１１は、追跡データ３１９を有する。しかし、追跡データ３１９は、ノード３１１からホスト記憶要素に送信される要求の状態と、ノード３１１から別のノードに送信される要求の状態とを示す。追跡データ３１９は、図１～２の飛行中追跡データと同様である。図１～２における変更伝搬器及びセカンダリライタの表現とは違って、図３～４は、変更伝搬器を、別の変更伝搬器からの変更要求に応答し、別の場所におけるセカンダリ論理記憶オブジェクトに変更を伝搬する機能を有するものとして表している。図３～４は、オブジェクト場所データ３２７をさらに表す。異なる例示的なエントリが図３～４に表されているが、こうしたエントリがオブジェクト場所データ内にあり、ノード識別子に対する論理記憶オブジェクト識別子を解決する。図１～２内と同様に、図３～４内の段階は、段階識別子と共に例示的な動作を表す。段階識別子は動作におけるシーケンスを示すが、その表される順序は請求項の範囲を限定するのに使用されるべきではなく、なぜならば、上記順序は例示目的のものだからである。

【００３４】

段階Ａ～Ｃは、図１における段階Ｃ、Ｈ、及びＩと同様である。段階Ａにおいて、ノード３０１は、変更要求を受信した後、同期マッピング３０５にアクセスする。該変更要求は、表されていない。同期マッピング３０５を用いて、ノード３０１は、ＯＢＪ３３として識別される論理記憶オブジェクトがＯＢＪ４４として識別されるオブジェクトとの完全

同期関係を有すると決定する。論理記憶オブジェクトＯＢＪ３３は、上記関係においてプライマリ論理記憶オブジェクトであり、記憶クラスタ３０３内にホストされている。記憶クラスタ３０３は、ノード３０１に関連付けられている。ノード３０１は、段階Ｂにおいて、ＯＢＪ３３をホストする記憶クラスタ３０３のメンバに変更要求を送信する。段階Ｃにおいて、ノード３０１は、オブジェクト場所データ３０７にアクセスし、ＯＢＪ４４がＮＯＤＥ＿２とＮＯＤＥ＿２のアドレスとに関連付けられていることを決定する。ＮＯＤＥ＿２はノード３１１である。ノード３０１は、それから、セカンダリ変更要求を送信する。セカンダリ変更要求は、（ファイルシステムの観点での）対象としてのＯＢＪ４４と、要求元としてのノード３０１内の変更伝搬器とを、ネットワーク３０９を介したノード間の接続を通じてノード３１１に対して示す。

【００３５】

段階Ｄ～Ｇにおいて、ノード３１１は、ノード３０１からのセカンダリ要求を処理する。段階Ｄにおいて、記憶クラスタ同期エンジン３１３は、接続のプロトコルに従ってノード３０１からのセカンダリ要求を処理する。記憶クラスタ同期エンジン３１３は、それから、セカンダリ変更要求をセカンダリライタ３１５に渡す。セカンダリライタ３１５は、段階Ｅにおいて同期マッピング３１７にアクセスする。セカンダリライタ３１５は、論理記憶オブジェクトＯＢＪ４４が論理記憶オブジェクトＯＢＪ５２との半同期関係を有すると決定する。段階Ｆにおいて、セカンダリライタ３１５は、追跡データ３１９を更新する。セカンダリライタ３１５は、決定された半同期関係に基づいて作成されることになるセカンダリ変更要求について、及びノード３０１から受信されるセカンダリ変更要求について、追跡データ３１９を更新する。セカンダリライタ３１５は、追跡データ３１９内に、応答をどこにルーティングするかの指標を維持する。この例において、セカンダリライタ３１５は、追跡データ３１９を更新して、オブジェクトＯＢＪ４４及びＯＢＪ５２が半同期関係にあることを示す。論理記憶オブジェクト識別子の各々が、状態指標に関連付けられる。この例示について、“０”の値は、飛行中又は待ちを示し、“１”の値は、対象論理記憶オブジェクトに対して変更要求が実行されたことを示す。この時点で、双方の状態指標は“０”に設定される。セカンダリライタ３１５は、追跡データ３１９をさらに更新して、要求元を“ＮＯＤＥ＿１＿ＯＢＪ３３”として示す。この値は単に、同期関係のノード及びプライマリ論理記憶オブジェクトの一例示的な指標である。段階Ｇにおいて、セカンダリライタ３１５は、さらなるセカンダリ変更要求を作成し、変更要求をその対応するハンドラに渡す。セカンダリライタ３１５は、ＯＢＪ４４を対象にする変更要求をファイルシステム３２１に転送する。変更伝搬器３１５は、ＯＢＪ５２としての対象と、セカンダリライタ３１５及びノード３１１を示す要求元識別子とを用いて、さらなる変更要求を作成する。例えば、さらなる要求は、ノード３１１と、セカンダリライタ３１５に結び付けられたポート又はソケットとを示してもよい。セカンダリライタ３１５は、さらなる変更要求を記憶クラスタ同期エンジン３１３に渡す。

【００３６】

段階Ｈにおいて、ファイルシステム３２１は、セカンダリ要求に従って、ＯＢＪ４４をホストするクラスタ３２５内のホスト記憶要素にアクセスする。

【００３７】

段階Ｉにおいて、記憶クラスタ同期エンジン３１３は、セカンダリライタ３１５からのさらなる変更要求をどこに送信するかを決定する。記憶クラスタ同期エンジン３１３は、オブジェクト場所データ３２７にアクセスし、ＯＢＪ５２がＮＯＤＥ＿３に関連付けられていることを示すエントリを発見する。ＮＯＤＥ＿３はノード３２９である。記憶クラスタ同期エンジン３１３は、オブジェクト場所データ３２７からノード３２９のアドレスを決定し、接続プロトコルに従ってさらなる変更要求を処理し、段階Ｊにおいてさらなる変更要求をネットワーク３０９を介してノード３２９に送信する。段階Ｋにおいて、ノード３２９は、ＯＢＪ５２に対してさらなる変更要求を実行する。

【００３８】

図４は、ノード３１１による、異なる同期関係のための応答の扱いを表す。段階Ａ～Ｄ

は、ノード３１１がＯＢＪ４４を対象にした変更要求に対する応答を処理する例示的な動作を表す。段階Ａにおいて、ＯＢＪ４４をホストするホスト記憶要素が、応答をファイルシステム３２１に送信する。段階Ｂにおいて、ファイルシステム３２１が応答をセカンダリライタ３１５に転送し、なぜならば、セカンダリライタ３１５が変更要求の発信元（originator）であったことを上記応答が示すからである。段階Ｃにおいて、セカンダリライタ３１５は、追跡データ３１９にアクセスする。セカンダリライタ３１５は、追跡データ３１９を更新して、ＯＢＪ４４に対する変更が実行されたことを示す。セカンダリライタ３１５は、ＯＢＪ４４のためのエントリが半同期関係を示し、要求元ＮＯＤＥ＿１＿ＯＢＪ３３を示すと決定する。上記が半同期関係であるため、変更伝搬器３１５は、要求元ＮＯＤＥ＿１＿ＯＢＪ４４に対する応答の提供を進めることができる。セカンダリライタ３１５は、段階Ｄにおいて応答を要求元識別子の指標と共に記憶クラスタ同期エンジン３１３に送信する。

【００３９】

ＯＢＪ３３とＯＢＪ４４との間の同期関係は完全同期関係であるため、ＯＢＪ４４に対する変更は、ＯＢＪ３３に関連付けられたノードに逆向きに（back）即座に通信されることができる。段階Ｅにおいて、記憶クラスタ同期エンジン３１３は、オブジェクト場所データ３２７にアクセスして、ＯＢＪ３３に関連付けられたノードを決定する。記憶クラスタ同期エンジン３１３は、セカンダリライタ３１５によって提供される要求元識別子からオブジェクト識別子を抽出するようにプログラムされることができる。しかしながら、オブジェクト識別子は異なる仕方で通信されてもよい。例えば、セカンダリライタ３１５又は記憶クラスタ同期エンジン３１３が同期マッピングにアクセスして、ＯＢＪ３３のためのプライマリ論理記憶オブジェクトを決定することができる。オブジェクト識別子が如何にして決定されるかにかかわらず、記憶クラスタ同期エンジン３１３は、ＯＢＪ３３がＮＯＤＥ＿１に関連付けられていると決定する。ＮＯＤＥ＿１はノード３０１である。

【００４０】

段階Ｆにおいて、応答がノード３０１に送信される。記憶クラスタ同期エンジン３１３は、段階Ｅにおいて逆向きに宛先をＮＯＤＥ＿１として決定した後（又は決定する間）、接続プロトコルに従って応答を処理する。記憶クラスタ同期エンジン３１３は、それから、ネットワーク３０９を横断する接続をとおして応答を伝達する。ノード３０１は、それから、応答を作成し、ネットワーク４０１を通じて最初の要求４０３（「クライアント」）に送信する。このことは、要求される変更がＯＢＪ３３においてすでに実行されていることを仮定し、なぜならば、ＯＢＪ３３がＯＢＪ４４との完全同期関係を有するからである。

【００４１】

段階Ｈ～Ｊにおいて、ＯＢＪ５２に対する変更を確認する応答が、逆向きにセカンダリライタ３１５に進む。段階Ｈにおいて、ＯＢＪ５２をホストする記憶クラスタ３３１のメンバが、ノード３２９に、ＯＢＪ５２に対して変更が実行されたとの応答を提供する。したがって、ノード３２９は、段階Ｉにおいて、応答を記憶クラスタ同期エンジン３１３に送信する。応答がセカンダリライタ３１５を示すので、記憶クラスタ同期エンジン３１３は、接続プロトコルに従って応答を処理した後、段階Ｊにおいてセカンダリライタ３１５に応答を渡す。

【００４２】

段階Ｋにおいて、変更伝搬器３１５は、追跡データ３１９を更新して、ＯＢＪ５２に対する更新が完了したことを示す。同期が完了したとのこの指標は、一貫性についての他の態様に対して使用されることができ、例えば、シーケンシング、フェイルオーバー、及びロードバランシングなどである。

【００４３】

図１～４はアーキテクチャの一部を表して例示的な動作を示しているが、図５は、１つ又は複数のクラスタのノードにわたり論理記憶オブジェクト粒度において完全同期と半同期との双方を提供する一例示的なアーキテクチャを表す。図５は、ファイルシステム要求

生成器５０１、変更伝搬エンジン５０３、ファイルシステム５０５、及び記憶クラスタ同期エンジン５０７を表している。ファイルシステム要求生成器５０１は、図１～４のファイルシステム要求生成器１０３と同様である。ファイルシステム要求生成器５０１は、ネットワークインターフェース又はシリアルインターフェース（例えば、ネットワークモジュール／スタック又はＳＣＳＩモジュール）を通じて受信される通信を処理するモジュールから受信される記憶プロトコル固有Ｉ／Ｏ要求を処理する。ファイルシステム５０５は、図１のファイルシステム１１１と同様とすることができ、ファイルシステム又はファイルシステムレイヤを実装する。ファイルシステム又はファイルシステムレイヤの例には、Ｗｒｉｔｅ　Ａｎｙｗｈｅｒｅ　Ｆｉｌｅ　Ｌａｙｏｕｔ及びＵＮＩＸ（登録商標）ファイルシステムが含まれる。ファイルシステム５０５は、ファイルシステム要求に従って下層のホスト記憶要素に要求を供給する。記憶クラスタ同期エンジン５０７は、同期関係のセカンダリ論理記憶オブジェクトをホストするクラスタノードにおけるカウンターパートの記憶クラスタ同期エンジンに、変更要求を供給する。

【００４４】

変更伝搬エンジン５０３は、傍受器５０９、シーケンサ（sequencer）５１１、伝搬器５１３、及びセカンダリライタ５１５を含む。ファイルシステム要求生成器５０１は、ファイルシステム要求を、対応する記憶Ｉ／Ｏ要求（例えば、ＳＡＮ又はＮＡＳ要求）内に示された論理記憶オブジェクト対象の指標と共に、変更伝搬エンジン５０３に渡す。ファイルシステム要求生成器５０１の観点から、ファイルシステム要求生成器５０１は、ファイルシステム要求をファイルシステム５０５に渡している。このことは、ファイルシステム要求生成器５０１の修正を回避し又は最小化することに役立つ可能性がある。しかし、ファイルシステム要求生成器５０１によって起動される関数又はプロシージャコールが傍受器５０９を実際には起動し、ゆえに、傍受器５０９がファイルシステム要求を「傍受する」ことが可能になる。応答が伝搬器５１３から受信されるとき、傍受器５０９は、応答を逆向きにファイルシステム要求生成器５０１に渡し、ファイルシステム要求生成器５０１は、それから、対応する記憶プロトコルＩ／Ｏ応答を作成する。

【００４５】

ファイルシステム要求は、最初、傍受器５０９に渡る。傍受器５０９は、最初、ファイルシステム要求が如何にして変更伝搬エンジンを通って流れるかを決定する。ファイルシステム要求が変更要求（例えば、書込み、ゼロ設定（zero）等）である場合、傍受器５０９は、同期関係の中の論理記憶オブジェクトを示す同期関係データにアクセスする（同期関係を有する論理記憶オブジェクトは、以降、エンドポイントといわれる）。同期関係データが、変更要求の対象（すなわち、プライマリ論理記憶オブジェクトであり、以降、「プライマリエンドポイント」といわれる）とセカンダリ論理記憶オブジェクト（すなわち、プライマリエンドポイントと同期する論理記憶オブジェクトであり、以降、「セカンダリエンドポイント」といわれる）との間の完全同期関係を示す場合、傍受器５０９は、変更要求、同期関係の指標、及びセカンダリエンドポイントの指標を伝搬器５１３に渡す。傍受器５０９は、この情報を、伝搬器５１３のインスタンス化を結果としてもたらす関数を呼び出すことによって渡すことができる。同期関係が半同期関係である場合、傍受器５０９は、この情報をシーケンサ５１１に渡す。変更伝搬エンジン５０３は、ファイルシステム要求と対応する半同期関係情報とを、シーケンサ５１１と伝搬器５１３との双方に対して、同時に又は時間において互いに近接して渡す傍受器を備えて設計されることができる。伝搬器５１３と同様に、傍受器５０９は上記情報をシーケンサ５１１に関数コールを用いて渡すことができ、上記関数コールがシーケンサ５１１をインスタンス化することができる。シーケンサ５１１及び伝搬器５１３は、プライマリ及びセカンダリエンドポイントの各ペアについてインスタンス化される。

【００４６】

シーケンサ５１１は、半同期関係におけるエンドポイントのためのファイルシステム要求で、又は、特定の記憶管理動作がトリガされるとき、動作する。上記管理動作は、例えば、スナップショット又は重複排除（deduplication）などである。シーケンサ５１１は

、依存関係（dependencies）を有する要求の順序を保存する。依存関係は、重なった書き込み間で、書き込みの間の読み出し要求から、指定された依存関係から等で生じることがある。シーケンサ５１１は、ＲＰＯなどの構成の制限内で変更要求を追跡する。例えば、ＲＰＯが１０秒として定義されるとき、シーケンサが５秒間隔で要求を追跡してもよい。個別の構成が何であれ、シーケンサ５１１は、構成された境界の範囲内で変更要求を蓄積し、上記境界は、時間、要求の数、又は双方の観点におけるものとすることができる。本説明は、有界の蓄積された変更要求を変更セットという。シーケンサ５１１は、蓄積された変更要求間における依存関係を決定し、依存関係に基づいてシーケンスを示す。シーケンサ５１１は、読み出し要求と変更要求とについての可視性を有して、変更要求間における依存関係を決定する。変更セットの中の各要求のメタデータの中に、シーケンサ５１１は、変更セット内におけるシーケンシングと要求の総数とを示す。例えば、シーケンサ５１１は、変更セット内の５つの変更要求のうちの第１の変更要求のためのメタデータを“１／５”として書き込む。変更セットの境界が到達されるとき、シーケンサ５１１は、次の変更セットのための要求を蓄積することと、セカンダリエンドポイントをホストするノードに対して通信するために同期エンジン５０７に現在の変更セットを通信することとを開始する。本説明は、上記処理を、現在の変更セットのクローズ又は現在の変更セットログのクローズ、及び、次の変更セット又は変更セットログのオープンという。シーケンサ５１１は、示された順序で変更セットログを横断する別のスレッド又はプロセス（例えば、バックグラウンドプロセス）を立ち上げることができ、各変更要求を同期エンジン５０７にサブミットする（submits）。シーケンサ５１１（又は、シーケンサ５１１により起動されるスレッド／プロセス）は、変更セットの中の変更要求を個々に送信し、このことは、要求が順序不同で受信ノードに到着することを可能にする。変更セットについて、成功の応答が同期エンジン５０７から受信されるとき、シーケンサ５１１は、この変更セットを完了としてマーク付けする。それから、変更セットログは破棄され、あるいは上書きされることができる。失敗の応答が受信され、あるいはタイムアウトが発生するとき、シーケンサ５１１は、同期が失敗したとの通知を生成し、あるいは再試行することができる。

【００４７】

伝搬器５１３は、変更要求の状態を追跡するデータを維持し、セカンダリエンドポイントのための要求を同期エンジン５０７に渡し、応答を逆向きに傍受器５０９に渡す。伝搬器５１３が傍受器５０９から変更要求を受信するとき、伝搬器５１３は、要求元の指標を記録し、それから、変更要求を修正して伝搬器５１３を要求元として示す。このことは、ファイルシステム５０５が伝搬器に応答を返すのを容易にするが、必要ではない。要求元のアイデンティティを変更することに代わって、ファイルシステムからの応答を傍受するようにアーキテクチャが設計されることができる。要求元のアイデンティティを変更するアーキテクチャにおいて、伝搬器５１３は、傍受器に応答を渡す前に、ファイルシステム５０５からの応答の中の要求元のアイデンティティを復元する。変更要求を扱うことに戻ると、伝搬器５１３は、完了していない変更要求（すなわち、飛行中変更要求）を示すデータを記録する。完全同期について、伝搬器５１３は、プライマリエンドポイント及びセカンダリエンドポイントのための変更要求に関するデータを記録する。伝搬器５１３は、このデータを使用して、いつ双方が完了するかと応答が要求元に提供されることができるかとを決定する。半同期について、伝搬器５１３は、プライマリエンドポイントのためのデータを記録し、なぜならば、シーケンサ５１１が半同期関係においてセカンダリエンドポイントのための要求を扱うからである。しかし、伝搬器５１３は、シーケンサ５１１に、プライマリエンドポイント上でいつ変更が完了されるかを通知する。こうした変更のすべてがプライマリエンドポイント上で成功裏に完了するまで、シーケンサ５１１はセカンダリエンドポイントのための変更セットを送出しない。

【００４８】

セカンダリライタ５１５は、セカンダリエンドポイントを対象にする変更要求を扱う。セカンダリライタ５１５は、変更要求を複製動作の形式で同期エンジン５０７から受信す

る。プライマリエンドポイントノードにおいて、同期エンジン５０７は、伝搬器５１３又はシーケンサ／シーケンサによりスポーンされた（spawned）スレッド５１１から供給される変更要求から、複製動作を生成する。複製動作は、プライマリエンドポイントノード（例えば、プライマリエンドポイントノード上の伝搬器インスタンス）を複製動作のソースとして示し、セカンダリエンドポイントを示す。複製動作は、同期関係のタイプをさらに示す。複製動作は、変更要求とは異なる要求元及び異なる対象を示し、ファイルシステム５０５のプロトコルから独立したプロトコルに従う、再形成された変更要求であってもよい。例えば、伝搬器からの変更要求が、要求元として伝搬器を、並びに、ＷＡＦＬ（ｗｒｉｔｅ　ａｎｙｗｈｅｒｅ　ｆｉｌｅ　ｌａｙｏｕｔ）に準拠する要求の中の対象として個別のファイル名及びファイル領域（例えば、ブロック）を示してもよい。複製動作は、上記情報を変更要求から抽出し、これを個別のファイルシステム又はプロトコルから独立して示すことができる。セカンダリライタ５１５が上記情報を受信するとき、セカンダリライタ５１５は、セカンダリエンドポイントノードのファイルシステムにより実装されるプロトコルに従って適切な要求を生成する。セカンダリライタ５１５は、ファイルシステム５０５にサブミットされる要求の状態を追跡し、応答を逆向きに同期エンジン５０７に渡す。半同期関係について、セカンダリライタ５１５は、変更セットが完了するまで、変更セットの要求を蓄積する（「ステージする（stages）」）。セカンダリライタ５１５は、要求のメタデータを読み出して、変更セットをいつ作成するかと変更セットがいつ完了するかとを決定する。変更セットが完了するとき、セカンダリライタ５１５は、プライマリエンドポイントノードに対して、変更セット内の各要求について個々の通知を送信することに代わって、変更セットが成功裏に完了したとの通知を生成することになる。セカンダリライタ５１５は、失敗の変更セットについて、通知をさらに生成することになる。いくつかの場合、セカンダリエンドポイントは、別の同期関係におけるプライマリエンドポイントである（「カスケード的（cascading）同期構成」）。セカンダリライタ５１５がインスタンス化されるとき、このセカンダリライタインスタンスは、同期関係データにアクセスして、セカンダリエンドポイントがカスケード的同期構成にあるかを決定する。そうである場合、セカンダリライタ５１５は、カスケード的関係のために伝搬器インスタンス及び／又はシーケンサインスタンスを起動することになる。セカンダリライタ５１５は、それ自体が変更要求の要求元を有すると示すことになる。

【００４９】

図５は、一例示的な論理オブジェクト粒度完全同期及び半同期アーキテクチャの一般的説明を提供するが、以降の図は、例示的な動作についてのさらなる例示を提供する。図６～１３は、クラスタノード内のエンドポイントについての例示的な完全同期及び半同期動作のフローチャートを表す。これら図は、図５に表される例示的なアーキテクチャからのアクタを参照して説明されるが、指定されるアクタは動作の理解を助けるためのものである。前述されたとおり、プログラム構造又は設計が変動する可能性があり、アクタを指定する例は請求項の範囲を限定するのに使用されるべきではない。

【００５０】

図６は、変更要求の受信を扱い、完全同期関係におけるプライマリエンドポイントを対象にする変更要求を扱う、例示的な動作のフローチャートを表す。傍受器が、ブロック６０１、６０３、６０５、及び６０７の動作を実行することができ、伝搬器が、ブロック６０８、６０９、６１１、及び６１３の動作を実行することができる。

【００５１】

ブロック６０１において、傍受器が、記憶プロトコルＩ／Ｏ要求から導出されるファイルシステム要求（以降、「要求」）を受信する。例えば、情報が記憶プロトコルＩ／Ｏ要求から抽出されて、ファイルシステム要求が生成されている。

【００５２】

ブロック６０３において、傍受器は、要求の対象が同期関係にあるかどうかを決定する。要求の対象が同期関係にない場合、制御はブロック６１７に流れる。要求の対象が同期関係にある場合、制御はブロック６０４に流れる。

【００５３】
ブロック６０４において、傍受器は、同期関係情報を読み出す。傍受器は、情報を別のデータ構造にコピーすることと、そのデータ構造を変更要求に関連付けることとによって、上記情報を「読み出す」ことができる。傍受器は、さらに、情報を含むデータ構造の中のエントリに対する参照（例えば、ポインタ、索引等）を記録することによって、上記情報を「読み出す」ことができる。同期関係情報は、同期関係におけるエンドポイントをホストする（１つ以上の）クラスタのメンバ間で広められる（circulated）データ構造の中に維持されることができる。このデータ構造は、個々のノードにおいて構成されることができる。構成は、コミットされた後、（１つ以上の）クラスタ内のノードにわたる更新をトリガすることができる。

【００５４】
ブロック６０５において、傍受器は、要求が変更要求又は読み出し要求であるかを決定する。傍受器が、要求が変更要求であると決定する場合、制御はブロック６１１に流れる。そうでなければ、制御はブロック６０７に流れる。

【００５５】
ブロック６０７において、傍受器は、要求の同期関係が完全同期関係又は半同期関係であるかを決定する。関係が完全同期関係である場合、制御はブロック６１７に流れ、なぜならば、読み出しは、対応する完全同期動作をトリガしないからである。関係が半同期関係である場合、制御はブロック６０９に流れ、なぜならば、読み出しが、変更セットの中の変更要求間における依存関係を作り出す可能性があるからである。

【００５６】
ブロック６０９において、要求は、同期関係のプライマリエンドポイント及びセカンダリエンドポイントペアのためのシーケンサインスタンスに渡される。シーケンサは、上記ペアについてすでにインスタンス化されていてもよく、あるいは、要求を渡すことと同時にインスタンス化されてもよい。例えば、傍受器が、各々の一意のエンドポイントペアリングについてインスタンス化されたシーケンサを示すデータをチェックすることができる。傍受器がエントリを発見する場合、傍受器は、変更要求及び同期関係情報に対する参照を、上記エントリの中のスレッド識別子を用いてスレッドに渡す。エントリが存在しない場合、傍受器は関数を呼び出し、変更要求及び同期関係情報に対する参照が該関数コールのパラメータとして渡される。制御はブロック６０９から図８のブロック８０１に流れる。

【００５７】
傍受器がブロック６０５において、要求が変更要求であったと決定した場合、制御はブロック６１１に流れている。ブロック６１１において、要求及び同期関係情報が、同期関係情報の中に示されるプライマリエンドポイント及びセカンダリエンドポイントペアのための伝搬器インスタンスに渡される。シーケンサと同様に、伝搬器インスタンスは、要求及び同期関係情報を渡すことと同時にインスタンス化されてもよい。さらに、変更要求と同期関係情報とのうちいずれか又は双方が、参照で（referentially）又はリテラルで（literally）渡されることができる。

【００５８】
ブロック６１３において、伝搬器インスタンスは、要求元の指標を記録し、伝搬器インスタンス自体を要求元として示す。伝搬器インスタンスはそれ自体を要求元として示して、ファイルシステムに、応答を伝搬器インスタンスへ返させる。このことは、伝搬器が既存のファイルシステムとインターフェースをとることを容易にする。伝搬器インスタンスは、少なくとも伝搬器インスタンスの観点から実際の要求元を記録し、したがって、下層のファイルシステムからの応答は実際の要求元を示すように更新されることができる。同期関係が半同期である場合、制御はブロック６０９に流れる。関係が完全同期である場合、制御はブロック６１５に流れる。

【００５９】
ブロック６１５において、伝搬器インスタンスは、データを記録して飛行中要求を追跡

する。「飛行中」要求といわれるが、要求はまだ飛行中でなく、なぜならば、上記データの記録の後まで伝搬器インスタンスが要求を渡さないからである。伝搬器インスタンスは、少なくとも要求の指標と、プライマリエンドポイントと、セカンダリエンドポイントと、応答がプライマリエンドポイント又はセカンダリエンドポイントのいずれかについて受信されたかの指標とを記録する。伝搬器インスタンスは、要求のメタデータから決定される要求の識別子を記録することができる。伝搬器インスタンスは、プライマリエンドポイント識別子及びセカンダリエンドポイント識別子を用いて識別子を生成することができる。伝搬器インスタンスは、このデータを記録して、変更がプライマリエンドポイント及びセカンダリエンドポイントの双方において成功裏に実行されたときを判断する。双方のエンドポイントにおける成功の後、応答が実際の要求元に伝えられることができる。要求を追跡するデータを記録した後、伝搬器インスタンスは、ブロック６１７及び６１９を同時に又は順々に実行することができる。順々の場合、伝搬器インスタンスは上記ブロックのうちいずれかを順繰りに実行することができる。

【００６０】

ブロック６１７において、伝搬器インスタンスは、下層のファイルシステムに要求を供給する。

【００６１】

ブロック６１９において、伝搬器インスタンスは、セカンダリエンドポイント上で実行するための要求を示す。例えば、伝搬器インスタンスは、要求及び同期関係情報を、セカンダリエンドポイントに関連付けられたノードに変更を通信するモジュールに渡す。

【００６２】

図７は、論理記憶オブジェクト粒度半同期動作についての例示的な動作のフローチャートを表す。図７は、図６のブロック６０９からの続きである。

【００６３】

ブロック７０１において、伝搬器インスタンスが、データを記録して飛行中要求を追跡する。ブロック７０１はブロック６１５と同様の言語において表現されるが、ブロック７０１の例示的な動作は、セカンダリエンドポイントに関連付けられたノードに送信される要求の状態を追跡しない。セカンダリエンドポイントノードに送信される要求の状態は、半同期関係に関しては伝搬器インスタンスによっては追跡されず、なぜならば、追跡はシーケンサによって扱われるからである。半同期関係において、伝搬器インスタンスは、プライマリエンドポイントについて下層のファイルシステムに送信される要求を追跡することを回避し、下層のファイルシステムの管理メカニズムに依存することができる。この場合、伝搬器インスタンスは、下層のファイルシステムからの成功又は失敗の応答を渡すことができる。伝搬器インスタンスは、応答の中に実際の要求元のアイデンティティを単に復元することになる。

【００６４】

ブロック７０３において、伝搬器インスタンスは、要求をファイルシステムに供給する。

【００６５】

図６のブロック６０９において、シーケンサは、要求及び同期関係情報を渡されている。ブロック７０５において、シーケンサは、オープン変更セットログの境界が到達されたかを決定する。例えば、境界は、構成されたＲＰＯの断片として定義されてもよい。一例として、シーケンサは、１２秒ＲＰＯに基づいて、４秒境界上で変更セットログを管理する。変更セットログがオープンされるとき、この変更セットログはシステム時間を用いてスタンプされることができる。境界が到達されるたび、変更セットログはクローズされ、処理のためにサブミットされる。１２秒ＲＰＯに対する４秒境界の上記例において、合計３つの変更セットログがＲＰＯ時間期間に及ぶ。変更セットログのうち１つがオープンであることになり、その他の２つはクローズされることになる。変更セット境界が到達された場合、制御はブロック７０９に流れる。変更セット境界が到達されなかった場合、制御はブロック７０７に流れる。

【００６６】
ブロック７０７において、シーケンサは、オープン変更セットログの中の要求を示す。シーケンサは、要求の識別子、要求のタイプ、及び要求に対する参照を記録することができる。シーケンサは、要求に対する参照を記録することができる。
【００６７】
ブロック７０９において、変更セットログはクローズされ、なぜならば、ブロック７０５において決定されたとおり境界が到達されたからである。例えば、シーケンサは、オープン変更セットログポインタと１つ以上のクローズされた変更セットログポインタとを維持することができる。境界が到達されるとき、シーケンサは、ポインタを更新して、ログのオープン及びクローズを反映することができる。シーケンサは、さらに、ログが失敗の変更セットについてである場合でさえ、クローズされたログをバッファの中に維持することができる。このことは、さらなるメモリを消費する可能性があり、なぜならば、ログが、オープン及びクローズ済みの双方、依然として処理中である変更セットに対して十分なメモリ空間に制約されないからである。しかし、さらなるメモリが、失敗した変更セットの調査を可能にし、あるいは、失敗した変更セットのより速い再試行を容易にすることができる。
【００６８】
ブロック７１１において、シーケンサは、クローズされたログを処理のために渡す。例えば、シーケンサは、新しい変更セットログの維持を継続すると同時に、クローズされたログを処理するスレッド又はプロセスを立ち上げることができる。クローズされたログを処理することには、変更要求間における依存関係を決定して変更セットの中の要求間における予期されるシーケンスを満足することが含まれる。
【００６９】
ブロック７１３において、シーケンサは、異なる変更セットログをオープンし、初期化する。ログをオープンするために、シーケンサは、異なるメモリ空間を割り振り、あるいは、（成功裏に又は不成功に）完了したクローズされたログのメモリ空間にアクセスすることができる。シーケンサは、オープン変更セットログを初期タイムスタンプを用いて初期化する。シーケンサは、さらに、いかなるデータも上書きして変更セットログをクリアし、あるいは、オープンされる前に別のプロセスがログのクリアを扱うことを可能にすることができる。
【００７０】
ブロック７１５において、シーケンサは、初期化されたオープン変更セットログの中の要求を示す。
【００７１】
図８は、クローズされた変更セットログを処理する例示的な動作のフローチャートを表す。上記処理は、変更セットの中の要求を解析し、任意の順序付けを決定して、正しい及び一貫したデータのビューを維持する。論理記憶オブジェクト粒度が、変更セットをアトミックな仕方で実装し、指定されたＲＰＯに準拠するように、変更セットを維持する。図７の例示的な動作は、スポーンされたスレッド又はプロセスを、クローズされた変更セットログを処理するものとして説明したが、図８は、シーケンサを、クローズされた変更セットログを処理するものとして説明する。
【００７２】
ブロック８０１において、シーケンサが、クローズされた変更セットログの中の変更要求間における任意の依存関係を決定し、該依存関係に従ってシーケンシングを示す。シーケンサは、変更要求の的であるエンドポイントの領域を示すデータを維持する。例えば、シーケンサは、ファイルについての領域のビットマップを維持することができる。ビットマップの第１の次元は、ファイルシステム及び／又は記憶プロトコルに依存して、ｘバイトのブロックを表現することができる。ビットマップの別の次元が、変更要求の各々を表現することができる。上記ビットマップを用いて、シーケンサは、変更要求がいつ重なるかを決定することができる。変更要求が重なる場合、シーケンサは、変更要求が互いに依

存すると決定し、そのシーケンスを保存して上記依存関係を満足する。シーケンサは、論理記憶オブジェクトを対象にする任意の変更要求が、該論理記憶オブジェクトを対象にする介在の読み出し要求を有するかをさらに決定する。その場合、シーケンサは、依存関係が存在すると決定し、周りの更新要求のシーケンス又は順序を保存する。シーケンサは、シーケンシング情報を変更要求の各々のメタデータに書き込む。例えば、シーケンサは、シーケンシング情報を変更要求のヘッダに書き込む。

【００７３】

ブロック８０３において、シーケンサは、変更セットの各変更要求の中の変更セット情報を示す。シーケンシング情報と同様に、シーケンサは、変更要求の各々のメタデータの中の変更セット情報を示す。シーケンシング情報には、変更セットの識別子と変更セット内の変更要求の数とが含まれる。このことは、セカンダリエンドポイントに関連付けられたノードにおけるセカンダリライタが、セカンダリライタが変更セットのすべての変更要求をいつ受信したかを決定するのを助ける。シーケンサは、変更要求の各々のメタデータの中の変更セットの開始時間をさらに示すことができる。このことは、セカンダリライタが、ＲＰＯ制約がいつ違反されたかを決定するのを助けることができる。

【００７４】

ブロック８０５において、シーケンサは、冗長な変更要求を消去する。シーケンサは、変更要求が後の変更要求によってさらに対象にされる同じプライマリエンドポイント及び同じ領域又はブロックを対象にする場合、変更要求が冗長であると決定する。換言すると、シーケンサは、変更セットを超えて存続するのでない変更を行う変更要求を決定する。

【００７５】

ブロック８０７において、シーケンサは、各変更要求をシーケンシング及び変更セットの指標と共にクラスタ同期エンジンに供給する。シーケンサは、シーケンシング及び変更セット情報の指標を用いて修正された変更要求に対する参照を渡すことができる。こうした変更をセカンダリエンドポイントノードに通信することを担うモジュールが、渡された参照を介して実際のデータを取得することができる。

【００７６】

図９は、同期関係におけるプライマリエンドポイントのファイルシステムからの応答を扱う例示的な動作のフローチャートを表す。これら例示的な動作は、伝搬器インスタンスにより実行される場合として説明される。

【００７７】

ブロック９０１において、伝搬器インスタンスが、プライマリエンドポイントのファイルシステムから応答を受信する。伝搬器インスタンスは、変更要求をファイルシステムに事前に渡している。変更要求は、プライマリエンドポイント（すなわち、ファイルハンドル及びファイルブロック番号などのファイル場所情報を有する論理記憶オブジェクト）と、要求のソースとしての伝搬器インスタンスとを示していた。今度は、ファイルシステムが、変更要求をサービス提供した（又は、サービス提供しようと試みた）後、応答を提供する。応答は、成功又は失敗のいずれかを示すことになる。

【００７８】

ブロック９０３において、伝搬器インスタンスは、応答が成功又は失敗を示すかを決定する。応答が成功を示す場合、制御はブロック９０９に流れる。応答が失敗を示す場合、制御はブロック９０５に流れる。

【００７９】

失敗の場合、伝搬器インスタンスは、ブロック９０５においてセカンダリエンドポイントへの対応する変更の中止（abort）を開始する。プライマリエンドポイントが完全同期関係にあるか又は半同期関係にあるかにかかわらず、セカンダリエンドポイントに対する変更は、プライマリエンドポイントとセカンダリエンドポイントとの間の同期外れ状態（out of sync state）を回避するために、成功裏に完了すべきでない。完全同期関係について、伝搬器インスタンスは、要求を同期エンジンにサブミットして、セカンダリエンドポイントノードに通信される変更要求を中止する。同期エンジンは、セカンダリエンド

ポイントに対する変更を中止する動作を実行し、エンドポイント間の同期を保存することになる。半同期関係について、伝搬器インスタンスは、変更セットを中止する。変更セットを中止することには、プライマリエンドポイントノードにおける変更セットログを失敗又は中止としてマーク付けすることと、同期エンジンがセカンダリエンドポイントノードに変更セットの要求を失敗し又は中止するように要求することを要求することとを含むことができる。

【００８０】

ブロック９０７において、伝搬器インスタンスは、変更要求が失敗したと要求元が通知されることができることを示す。伝搬器インスタンスは、例えば、プライマリエンドポイントのファイルシステムからの失敗応答を変更して実際の要求元を示し、変更された応答をネットワークモジュールに渡すことができる。ネットワークモジュールは、それから、失敗を実際の要求元に通信することができる。

【００８１】

変更要求が成功した場合、伝搬器インスタンスは、ブロック９０９において追跡データを更新して成功を示す。伝搬器インスタンスは、追跡データを更新して、要求がプライマリエンドポイントにおいて完了したことを示す。

【００８２】

ブロック９１１において、伝搬器インスタンスは、追跡データを用いて、セカンダリエンドポイントに対する変更が完了したかを決定する。そうでない場合、制御はブロック９１３に流れる。セカンダリエンドポイントに対する変更が成功裏に完了した場合、制御はブロック９２１に流れる。

【００８３】

ブロック９１３において、伝搬器インスタンスは、タイムアウトが到達されたかを決定する。タイムアウトは構成されることができる。上記タイムアウトは、タイムアウトが満了する前に応答が受信されるべきであることを仮定する。そうでなければ、要求か又は要求に対するセカンダリエンドポイントノードからの応答が、失われたとみなされることができる。タイムアウトが到達された場合、制御はブロック９１７に流れる。タイムアウトが到達されていない場合、伝搬器インスタンスはブロック９１５において、定義された待ち期間の間待つ。制御はブロック９１５から戻ってブロック９１１に流れる。

【００８４】

ブロック９１７及び９１９は、タイムアウトシナリオにおける動作を表す。ブロック９１７において、伝搬器インスタンスは、セカンダリエンドポイントがプライマリエンドポイントとの同期から外れていることを示す。ブロック９１９において、エンドポイント間における同期外れ状態は、構成されたとおりに処理される。例えば、再試行が許容される場合、同期外れ状態が再試行を引き起こしてもよい。同期外れ状態が、管理モジュールに対する通知をトリガしてもよい。

【００８５】

セカンダリエンドポイントにおける更新要求が成功裏に完了するとき、伝搬器インスタンスは、ブロック９２１において、実際の要求元が変更要求の成功裏の完了を通知されることができることを示す。伝搬器インスタンスは、応答を、実際の要求元に通信するネットワークモジュールに供給する。

【００８６】

ブロック９２３において、追跡データがクリアされる。伝搬器インスタンスは、このデータをクリアし、あるいは、ガーベッジコレクションスレッドによってクリアされるように追跡データをマーク付けすることができる。

【００８７】

図９は、プライマリエンドポイントに対する変更の応答を扱う例示的な動作を表すが、図１０は、クラスタベースの同期エンジンが伝搬器とカウンターパートの同期エンジンとからの要求を処理する例示的な動作のフローチャートを表す。図１０の説明は、アクタを同期エンジンという。

【００８８】
ブロック１００１において、同期エンジンが、セカンダリエンドポイント上で実行されるべき変更要求の指標を受信する。変更要求は、参照で又はリテラルで同期エンジンに渡されることができる。変更要求は、完全同期関係のための変更セットのメンバ、又はスタンドアロンの変更要求であり得る。同期エンジンは、変更要求のメタデータの中、又は変更要求に関連付けられた別個の構造の中に、セカンダリエンドポイントの指標を受信することができる。
【００８９】
ブロック１００３において、同期エンジンは、セカンダリエンドポイントに関連付けられたクラスタノードを決定する。同期エンジンは、クラスタにわたり維持されるデータにアクセスする。このデータは、エンドポイント及びノードのためのディレクトリ（directory）として使用されることができる。データは、いずれのノードがいずれの論理記憶オブジェクトに関連付けられる（すなわち、いずれの論理記憶オブジェクトをホストし及び／又はいずれの論理記憶オブジェクトに対するアクセスを管理する）かを示す。上記データは、データベースとして実装されることができる。同期エンジンは、データをセカンダリエンドポイントのアイデンティティを用いて読み出し、該アイデンティティは、論理記憶オブジェクト識別子である。
【００９０】
ブロック１００５において、同期エンジンは、通信セッションがセカンダリエンドポイントノードにおける同期エンジンとの間ですでに確立されているかを決定する。同期エンジンは、通信セッションを維持して、各要求のための通信セッションを確立するオーバーヘッドを回避する。しかしながら、このことは必要ではない。同期エンジンは、エンドポイントペアごとにセッション又は接続を確立することができる。セッションがすでに確立されてはいない場合、制御はブロック１００７に流れる。そうでなければ、制御はブロック１００９に流れる。
【００９１】
ブロック１００７において、同期エンジンは、セカンダリエンドポイントに関連付けられたクラスタノードにおける同期エンジンとの通信セッションを確立する。
【００９２】
ブロック１００９において、同期エンジンは、変更要求に従ってセカンダリエンドポイントを対象にする複製要求を作成し、伝搬器インスタンスを複製要求のソースとして示す。同期エンジンは、セカンダリエンドポイントを要求の対象として示す要求を作成する。同期エンジンは、セカンダリエンドポイントに書き込まれるべきデータの指標又はデータを有する要求を作成する。同期エンジンは、クラスタノードの指標と受信した変更要求のメタデータとを有する要求をさらに作成する。
【００９３】
ブロック１０１１において、同期エンジンは、セカンダリエンドポイントに関連付けられたクラスタノードに上記セッションを通じて複製要求を通信する。ブロック１０１１からブロック１０１３への破線は、複製要求の送信と応答の受信との間の時間の経過を表す。
【００９４】
ブロック１０１３において、同期エンジンは、セカンダリノードに関連付けられたクラスタノードから、複製要求に対する応答を受信する。同期エンジンは、応答から伝搬器インスタンスを決定し、上記応答は、伝搬器インスタンスを要求元として示している。同期エンジンは、ブロック１０１５において、複製応答を、応答の中に示される適切な伝搬器インスタンスに渡す。同期エンジンは、伝搬器インスタンス識別子を複製要求識別子（例えば、エンドポイント識別子に基づいて生成される識別子）に関連付けたデータを維持することによって要求元を決定するように設計されることができる。
【００９５】
図１１は、伝搬器インスタンスがセカンダリエンドポイントに対する変更要求への応答

を扱う例示的な動作のフローチャートを表す。図１１は、例示的な動作のアクタとして伝搬器インスタンスを参照して説明される。前に説明されたとおり、伝搬器インスタンスは、セカンダリエンドポイントに対して作られることになる変更要求を同期エンジンに渡し、上記変更要求は、複製要求といわれている。同期エンジンは、こうした変更を、セカンダリエンドポイントに関連付けられたクラスタノードに通信する。

【００９６】

ブロック１１０１において、伝搬器インスタンスは、同期エンジンから複製応答を受信する。複製応答は、応答がセカンダリエンドポイント及びプライマリエンドポイントに対応することを示す。

【００９７】

ブロック１１０３において、伝搬器インスタンスは、エンドポイント間における同期関係が完全同期又は半同期であったかを決定する。同期関係が完全同期である場合、制御はブロック１１０５に流れる。同期関係が半同期である場合、制御はブロック１１２３に流れる。

【００９８】

ブロック１１０５において、伝搬器インスタンスは、セカンダリエンドポイントに対する変更要求が成功であったかを、複製応答に基づいて決定する。成功の場合、制御はブロック１１１３に流れる。そうでなければ、制御はブロック１１０７に流れる。

【００９９】

ブロック１１０７において、伝搬器インスタンスは、プライマリエンドポイントに対する要求された変更が成功裏に完了したかを決定する。伝搬器インスタンスは、飛行中追跡データを読み出して、プライマリエンドポイント変更が成功裏に完了したかを決定する。プライマリエンドポイントに対する変更が成功裏に完了しており、セカンダリエンドポイントに対する変更が成功しなかった場合、エンドポイントは同期から外れる。プライマリエンドポイントに対する変更が成功裏に完了した場合、制御はブロック１１２１に流れる。プライマリエンドポイントに対する変更が成功裏に完了しなかった場合、制御はブロック１１０９に流れる。

【０１００】

ブロック１１２１において、プライマリエンドポイントに対する変更はロールバックされる。プライマリエンドポイントに対する変更をロールバックすることは、要求元が失敗の応答を与えられることにつながる。要求元は、それから、変更を再度要求することができる。伝搬器は、プライマリエンドポイントに対する変更をロールバックすることに追加で又は代わって、同期外れ状態を示すようにプログラムされることができる。制御はブロック１１２１からブロック１１２７に流れる。

【０１０１】

ブロック１１０９において、プライマリエンドポイントに対する変更は中止される。まれと見込まれるが、セカンダリエンドポイントに対する変更に関連付けられたノードが、伝搬器インスタンスがプライマリエンドポイントの下層の記憶要素から応答を受信する前に、変更要求をサービス提供する可能性がある。

【０１０２】

ブロック１１１１において、伝搬器インスタンスは、変更要求が失敗したと実際の要求元が通知されることができることを示す。例えば、伝搬器インスタンスは、プライマリエンドポイントの下層の記憶要素からの応答に基づいて、失敗応答を作成する。伝搬器インスタンスは、傍受器から渡された変更要求から事前に記録された要求元を用いて、失敗応答を作成する。

【０１０３】

セカンダリエンドポイントに対する変更が完全同期関係において成功した場合、伝搬器インスタンスは、ブロック１１１３において、要求された変更がプライマリエンドポイントにおいて成功裏に完了したかを決定する。伝搬器インスタンスは、飛行中追跡データにアクセスして、プライマリエンドポイント変更がすでに完了したかを決定する。プライマ

リエンドポイント変更がすでに完了している場合、制御はブロック１１１７に流れる。プライマリエンドポイント変更がまだ完了していない場合、制御はブロック１１１５に流れる。

【０１０４】

ブロック１１１５において、伝搬器インスタンスは、飛行中追跡データを更新して、セカンダリエンドポイント変更が完了したことを示す。

【０１０５】

ブロック１１１７において、伝搬器インスタンスは、変更が双方のエンドポイントにおいて成功したとき、要求が完了したと要求元が通知されることができることを示す。伝搬器インスタンスは、プライマリエンドポイントの下層の記憶要素からの応答に基づいて、応答を生成する。応答は、要求についての成功裏のサービス提供を示す。伝搬器インスタンスは、さらに、要求元としてのそれ自体の指標を、実際の要求元の指標で置換する。伝搬器インスタンスは、それから、応答を傍受器又は通信モジュールに渡す。

【０１０６】

ブロック１１１９において、要求のための追跡データがクリアされる。伝搬器インスタンスは追跡データをクリアすることができ、あるいは、ガーベッジコレクティングスレッド（又は別のデータメンテナンススレッド）が追跡データをクリアすることができる。

【０１０７】

応答が、半同期関係におけるセカンダリエンドポイントについてである場合、制御はブロック１１２３に流れている。ブロック１１２３において、伝搬器インスタンスは、複製応答がセカンダリエンドポイントに対する変更セットの成功裏の完了を示すかを決定する。そうである場合、制御はブロック１１２５に流れる。そうでない場合、制御はブロック１１２７に流れる。

【０１０８】

ブロック１１２５において、伝搬器インスタンスは、エンドポイントが同期から外れていることを示す。伝搬器インスタンスは、クラスタノード間で広められる同期関係データにアクセスすることができる。この同期関係データは、伝搬器インスタンスにより設定されることが可能な単一ビットフィールドを含んで、対応するエンドポイントが同期から外れているか又は同期しているかを示すことができる。伝搬エンジンが、同期外れのエンドポイントを伴う要求を、構成されたとおり処理することになる。例えば、伝搬エンジンは、同期外れであると示されるプライマリエンドポイントを対象にするすべての要求を、同期が上記セカンダリエンドポイント又は代替的セカンダリエンドポイントとの間で復元されるまで、さえぎる（fence）ように構成されることができる。伝搬エンジンは、対象にされたエンドポイントが同期外れとして示されるとき、失敗又はサービス外（out of service）タイプの応答で応答するように構成されることができる。プライマリエンドポイントが成功裏に変わり、変更セットがセカンダリエンドポイントにおいて成功裏に完了した場合、制御はブロック１１２９に流れる。そうでなければ、制御はブロック１１２７に流れる。

【０１０９】

ブロック１１２９において、伝搬器インスタンスは、変更セットログをクリアする。要求元がプライマリエンドポイントに対する成功裏の変更についてすでに通知されているので、半同期関係におけるセカンダリエンドポイントに対する成功裏の変更は、要求元に対する通知をトリガしない。変更セットログをクリアすることは、変更セットが成功裏に完了したことを暗に示す。伝搬器インスタンスは、クリア又は除去の前、変更セットログを成功裏に完了されたとしてマーク付けするようにプログラムされることができる。

【０１１０】

図１２は、セカンダリライタが複製要求を扱う例示的な動作のフローチャートを表す。前に説明されたとおり、セカンダリライタが同期エンジンから複製要求を受信し、その双方がセカンダリエンドポイントに関連付けられたノード上で稼働している。セカンダリエンドポイントとの同期関係にあるプライマリエンドポイントに関連付けられたノード上で

稼働する同期エンジンが、セカンダリエンドポイントノードにおける同期エンジンに複製要求を通信している。

【０１１１】

ブロック１２０１において、セカンダリエンドポイントに関連付けられたクラスタノードにおける同期エンジンが、複製要求を受信する。複製要求は、プライマリエンドポイント及びセカンダリエンドポイントを示す。プライマリエンドポイント又はプライマリエンドポイントノードにおける伝搬器インスタンスが、複製要求のソースとして示される。複製要求は、同期関係のタイプをさらに示すことができる。

【０１１２】

ブロック１２０３において、同期エンジンは、セカンダリライタがプライマリ及びセカンダリエンドポイントについてすでにインスタンス化されているかを決定する。例えば、上記エンドポイントペアのためのセカンダリライタが、変更セットについてのより早い要求に対してインスタンス化されている可能性がある。セカンダリライタがすでにインスタンス化されている場合、制御はブロック１２０７に流れる。そうでない場合、制御はブロック１２０５に流れる。

【０１１３】

ブロック１２０７において、同期エンジンは、複製応答と、別個に通信される何らかがある場合には関連するメタデータとを、セカンダリライタインスタンスに渡す。メタデータは、複製要求の中に示されてもよい。

【０１１４】

ブロック１２０５において、同期エンジンは、示されるエンドポイントペアに基づいてセカンダリライタをインスタンス化する。制御は、ブロック１２０５及び１２０７のいずれかからブロック１２０９に流れる。

【０１１５】

ブロック１２０９において、セカンダリライタインスタンスは、変更要求がステージされる（staged）べきかを決定する。変更セットのための変更要求のステージングは、変更要求又は変更要求の指標を制限まで蓄積することを指す。セカンダリライタは、変更要求のメタデータを読み出して、変更要求が変更セット内にあるかを決定することができる。メタデータは、変更セットを示してもよい。セカンダリライタインスタンスは、さらに、完全同期の代わりに半同期の指標に基づいて、変更要求が変更セットの中にある場合として進めることができる。変更要求がステージされることになる場合、制御はブロック１２１５に流れる。そうでない場合、制御はブロック１２１１に流れる。

【０１１６】

ブロック１２１１において、セカンダリライタインスタンスは、データを記録して複製要求を追跡する。セカンダリライタインスタンスは、追跡データを使用して、要求が下層のファイルシステムに渡されていることを記録する。セカンダリライタインスタンスは、データを記録して複製要求を追跡することに代わって、下層のファイルシステムに依存することができる。

【０１１７】

ブロック１２１３において、セカンダリライタインスタンスは、複製要求を下層の記憶要素アクセスモジュールに供給する。伝搬器インスタンスと同様に、セカンダリライタインスタンスは、複製要求を下層のファイルシステムに供給する前に、要求元の指標を記録し、それをセカンダリライタインスタンスの指標で置換することができる。

【０１１８】

セカンダリエンドポイントが半同期関係にある場合、セカンダリライタインスタンスは、ブロック１２１５において、エンドポイントペアのための複製要求をステージすることについて変更セットログがすでに作成されているかを決定することになる。変更セットログがすでに作成されている場合、制御はブロック１２１７に流れる。変更セットログがエンドポイントペアについてすでに作成されてはいない場合、制御はブロック１２２５に流れる。

【０１１９】
ブロック１２２５において、セカンダリライタインスタンスは、ステージングログ（staging log）（すなわち、セカンダリエンドポイントノードにおける変更セットログ）を作成する。セカンダリライタインスタンスは、複製要求を用いてステージングログを初期化する。
【０１２０】
ブロック１２１７において、セカンダリライタインスタンスは、すでに作成されたステージングログの中の複製要求を示す。
【０１２１】
ブロック１２１９において、セカンダリライタインスタンスは、変更セットログが完了であるかを決定する。セカンダリライタインスタンスは、複製要求のうちいずれかについてのメタデータにアクセスして、変更セットの中の複製要求の総数を決定することができる。セカンダリライタインスタンスは、それから、メタデータから決定された上記数を、変更セットログ又はステージングログの中に示される複製要求の数に対して比較することができる。ステージングログが完了でない場合、セカンダリライタインスタンスは、さらなる複製要求が受信されるのを待つ。
【０１２２】
ステージングログが完了であるとの決定に対して、セカンダリライタインスタンスは、ブロック１２２１においてステージングログを横断する（traverses）。セカンダリライタインスタンスは、ステージングログの中に示される第１のマーク付けされていない複製要求を選択し、この選択された複製要求を下層のファイルシステムに供給する。セカンダリライタインスタンスは、対応する応答が受信されるとき、ステージングログ内の次のマーク付けされていない複製要求に進む。セカンダリライタインスタンスは、ステージングログが横断されるまで上記処理を継続し、このことは、図１３においてより詳細に説明される。
【０１２３】
図１３は、セカンダリライタインスタンスが下層のファイルシステムからの応答を扱う例示的な動作のフローチャートを表す。図１３は、変更セットの中の要求の応答を扱う例示的な動作を単に表している。換言すると、図１３は、半同期関係におけるセカンダリエンドポイントの例示的な動作を単に表している。完全同期関係のための応答を扱うとき、セカンダリライタインスタンスは、応答を同期エンジンに渡す。セカンダリライタインスタンスは、最初、元の要求元の指標を復元することになる。
【０１２４】
ブロック１３０１において、セカンダリライタインスタンスが、下層のファイルシステムから複製要求に対する応答を受信する。
【０１２５】
ブロック１３０３において、セカンダリライタインスタンスは、応答がセカンダリエンドポイントに対する成功裏の変更を示すかを決定する。応答がセカンダリエンドポイントに対する成功裏の変更を示す場合、制御はブロック１３０４に流れる。応答がセカンダリエンドポイントに対する失敗した変更を示す場合、制御はブロック１３０５に流れる。
【０１２６】
ブロック１３０５において、セカンダリライタインスタンスは、再試行が構成されているかを決定する。セカンダリライタインスタンスは、ＲＰＯ適合構成に依存して、変更セット内の要求を再試行するように構成されることができる。例えば、セカンダリライタインスタンスは、所定量の時間がＲＰＯ時間期間内で依然として残っている場合、変更セットを再試行するように構成されることができる。再試行が構成され、許容される場合、制御はブロック１３０７に流れる。そうでなければ、制御はブロック１３１１に流れる。
【０１２７】
ブロック１３０７において、セカンダリライタインスタンスは、再試行カウンタを更新する。とり得るリソースの無駄を回避するために、再試行は、構成された回数に制限され

る。
【０１２８】
ブロック１３０９において、セカンダリライタインスタンスは、要求を下層のファイルシステムに再度供給する。
【０１２９】
再試行が構成されていなかったか、あるいは許容されていなかった場合、セカンダリライタインスタンスは、ブロック１３１１において、データを記録して、変更セットが失敗したことを示す。セカンダリライタインスタンスは、失敗の指標を、ステージングログのメタデータに書き込むことができる。失敗指標は、失敗がプライマリエンドポイントノードに逆向きに通信されることができないか又は通信されない場合、変更セットについての上記失敗状態を保存するのに役立つ可能性がある。
【０１３０】
ブロック１３１３において、セカンダリライタインスタンスは、変更セットが失敗したとの通知を生成する。セカンダリライタインスタンスは、変更セットを識別する応答と、失敗の指標とを生成することができる。失敗通知は、それから、同期エンジンを介して要求ノード（すなわち、プライマリエンドポイントノード）に供給される。ブロック１３１３からブロック１３１５への点線は、時間の経過を示す。より後の時間に、セカンダリライタインスタンスは、変更セットログを破棄のためにマーク付けすることができる。セカンダリライタは、失敗の指標を用いて変更セットログを破棄するようにプログラムされることができる。
【０１３１】
複製要求応答が成功を示した場合、セカンダリライタインスタンスは、ブロック１３０４において、対応する変更が完了したかを決定する。変更セット全体が完了している場合、制御はブロック１３１７に流れる。変更セット全体が完了してはいない場合、制御はブロック１３１９に流れる。
【０１３２】
ブロック１３１７において、セカンダリライタインスタンスは、変更セットが成功裏に完了したとの通知を生成する。セカンダリライタインスタンスは、変更セットを識別する応答と、成功の指標とを生成することができる。成功通知は、それから、同期エンジンを介して要求ノード（すなわち、プライマリエンドポイントノード）に供給される。セカンダリライタは、変更セット成功通知を生成することに代わって、変更セット内の複製要求のうち１つに対する応答を渡して戻すようにプログラムされることができる。セカンダリライタは、変更セットのシーケンシング情報に従って、最後の要求に対する応答を要求ノードに返すことができる。変更セットの最後の変更要求に対するこの応答が、プライマリエンドポイントノードにおける伝搬器インスタンスに対して変更セット全体の成功通知として動作することができる。
【０１３３】
ブロック１３１９において、セカンダリライタインスタンスは、ステージングログの中の個別の要求を成功裏に完了したとしてマーク付けする。
【０１３４】
ブロック１３２１において、セカンダリライタインスタンスは、ステージングログの横断を継続する。セカンダリライタインスタンスは、ステージングログ内の要求について示される順序付けに従って、ステージングログ内の次のマーク付けされていない要求を選択する。セカンダリライタインスタンスは、この選択された要求を下層のファイルシステムに供給する。
【０１３５】
例示からのバリエーション
フローチャートは、例示の理解の助けとなるように提供されており、請求項の範囲を限定するのに使用されるべきではない。フローチャートは、開示の態様間で変動し得る例示的な動作を表す。さらなる動作が実行されてもよく、より少ない動作が実行されてもよく

、動作が並列に実行されてもよく、動作が異なる順序で実行されてもよい。例えば、傍受器が、要求を伝搬器に単に渡してもよい。伝搬器は、変更要求の対象が同期関係にあるかを決定するようにプログラムされることができる。バリエーションの別の例として、ブロック６１５が、同期構成にかかわらず実行されることができる。追跡データの欠如が半同期関係を暗に示すこと及び応答が逆向きに要求元に供給されることに代わって、各要求の追跡データが維持されることができる。フローチャートの動作のいくつかは、スレッド又はプロセスがエンドポイントペアについてすでにインスタンス化されているかを決定することを説明した。存続スレッドを使用しないアーキテクチャが、設計されることができる。代わりに、状態データがエンドポイントペアごとに記憶される。この状態データは、対応する（１つ以上の）要求が完了し又は失敗した後にクリアされるまで存続する。このことは、リソースを消費する待ち状態のスレッドを回避する。別の例として、伝搬器インスタンスが、シーケンサに代わって又はシーケンサに対して追加で、セカンダリエンドポイントに対する要求の状態を追跡することができる。変更セットの状態を追跡する個別のアクタにかかわらず、変更セット内の個々の要求の状態は、変更がプライマリエンドポイントにおいて実行された後に要求元に応答することを妨げない。ゆえに、変更セット内の個々の要求の状態は、追跡される必要がない。図７において、ブロック７１１は、クローズされた変更セットログをプロセスのスポーンされたスレッドに渡すことを説明している。オープンログを処理するアーキテクチャがプログラムされ、あるいは設計されることができる。アーキテクチャは、要求が追加されるたび変更セットログを処理し、シーケンシング情報を更新し、冗長な変更を消去するなどすることができる。変更セットログがクローズされるとき、該ログはすでに順序付けられており、セカンダリエンドポイントに関連付けられたノードに通信する準備ができている。

【０１３６】

失敗の通信は、フローチャート内に表される仕方とは異なる仕方で通信されることができる。例えば、失敗指標（例えば、ブロック１３１１）は記録されなくてもよく、なぜならば、失敗通知が生成されるからである。失敗した変更セットについて、セカンダリライタインスタンスが、変更セット内の要求のうち１つについての失敗の応答を渡して戻すことができる。要求ノードにおける伝搬器インスタンスは、いずれのクローズされた変更セットログが上記失敗応答に対応するかを決定し、その変更セットログを失敗したとしてマーク付けすることができる。

【０１３７】

さらに、表されていないさらなる動作が実行されることができる。例えば、変更セットログを監視する監視スレッドがスポーンされることができる。監視スレッドは、定義されたＲＰＯに対して、変更セットの寿命を評価することができる。アクティブな変更セットログ又は処理中の変更セットログは、プライマリエンドポイントノード又はセカンダリエンドポイントノードのいずれかからの応答を依然として待つクローズされた変更セットログである。監視スレッドは、変更セット開始時間を評価して、ＲＰＯ時間が経過したかを決定する。そうである場合、監視スレッドは、変更セットを失敗したとしてマーク付けするようにシーケンサスレッドに促し、あるいは、それ自体で変更セットを失敗したとしてマーク付けすることができる。

【０１３８】

本説明は、同期関係についてペアにされている個々の論理記憶オブジェクトを参照するが、同期関係の「エンドポイント」は、論理記憶オブジェクトのグループとすることができる。例えば、ファイルのグループ又はＬＵＮのグループが、論理記憶オブジェクトの別のグループとの同期関係にあることができる。ノードは、さらなるデータを維持して、グループのメンバである論理記憶オブジェクトに対するグループ識別子を解決することができる。

【０１３９】

当業者には十分理解されるであろうとおり、本開示の態様は、システム、方法、又はコンピュータプログラム製品として実施されることができる。したがって、本開示の態様は

、ハードウェア態様、ソフトウェア態様（ファームウェア、常駐ソフトウェア、マイクロコード等を含む）、又はソフトウェア態様とハードウェア態様とを組み合わせた態様の形式をとることができ、これら態様はすべて一般に「回路」、「モジュール」、又は「システム」といわれることがある。さらに、本開示の態様は、コンピュータ読取可能プログラムコードを具現化させた１つ以上のコンピュータ読取可能媒体において具現化されたコンピュータプログラム製品の形式をとってもよい。

【０１４０】

１つ以上のコンピュータ読取可能媒体の任意の組み合わせが利用されてもよい。コンピュータ読取可能媒体は、コンピュータ読取可能信号媒体又はコンピュータ読取可能記憶媒体であり得る。コンピュータ読取可能記憶媒体は、例えば、これらに限られないが、電子的、磁気的、光学的、電磁的、赤外線の、又は半導体の、システム、装置、若しくはデバイス、又は前述のものの任意の適切な組み合わせであり得る。コンピュータ読取可能記憶媒体のさらなる具体的な例（包括的でないリスト）には、下記が含まれる：１つ以上のワイヤを有する電気接続、ポータブルコンピュータディスケット、ハードディスク、ランダムアクセスメモリ（ＲＡＭ）、読取専用メモリ（ＲＯＭ）、消去可能プログラマブル読取専用メモリ（ＥＰＲＯＭ又はフラッシュメモリ）、光ファイバ、ポータブルコンパクトディスク読取専用メモリ（ＣＤ－ＲＯＭ）、光学記憶装置、磁気記憶装置、又は前述のものの任意の適切な組み合わせ。本文献の文脈において、コンピュータ読取可能記憶媒体は、命令実行システム、装置、若しくはデバイスによってか又はこれらに関連して使用されるプログラムを含み又は記憶することができる任意の有形媒体であり得る。

【０１４１】

コンピュータ読取可能信号媒体は、コンピュータ読取可能プログラムコードを例えばベースバンドにおいて又は搬送波の一部として具現化された、伝搬されるデータ信号を含み得る。こうした伝搬信号は、様々な形式のうち任意のものをとることができ、これらに限られないが、電子磁気信号、光信号、赤外線信号、又はこれらのうち任意の適切な組み合わせが含まれる。コンピュータ読取可能信号媒体は、コンピュータ読取可能記憶媒体でなく、コンピュータによってか又はコンピュータと関連して使用されるプログラムを通信し、伝搬し、又は輸送することができる、任意のコンピュータ読取可能媒体であり得る。コンピュータ読取可能信号媒体上に具現化されたプログラムコードは、任意の適切な媒体を用いて伝達されてよく、これらに限られないが、無線、有線、光ファイバケーブル、ＲＦ等、又は前述のものの任意の適切な組み合わせが含まれる。

【０１４２】

本開示の態様の動作を実行するコンピュータプログラムコードは、１つ以上のプログラミング言語の任意の組み合わせにおいて書かれてよく、プログラミング言語には、オブジェクト指向プログラミング言語、例えばＪａｖａ（登録商標）プログラミング言語、Ｃ＋＋など、ダイナミックプログラミング言語、例えばＰｙｔｈｏｎなど、スクリプト言語、例えばＰｅｒｌプログラミング言語又はＰｏｗｅｒＳｈｅｌｌ（登録商標）スクリプト言語など、会話手続型プログラミング言語、例えば“Ｃ”プログラミング言語又は同様のプログラミング言語などが含まれる。プログラムコードは、全体的にスタンドアロンコンピュータ上で実行されてもよく、複数のコンピュータにわたり分散される仕方で実行されてもよく、１つのコンピュータ上で実行されると同時に別のコンピュータ上で結果を提供し又は入力を受け入れてもよい。

【０１４３】

本開示の態様は、本開示の態様に従う方法、装置（システム）、及びコンピュータプログラム製品のフローチャート例示及び／又はブロック図を参照して説明される。フローチャート例示及び／又はブロック図の各ブロック、並びにフローチャート例示及び／又はブロック図内の複数ブロックの組み合わせが、コンピュータプログラム命令によって実装できることが理解されるであろう。上記コンピュータプログラム命令は、汎用目的コンピュータ、特別目的コンピュータ、又は他のプログラマブルデータ処理装置のプロセッサに提供されてマシンを生じさせることができ、したがって、コンピュータ又は他のプログラマ

ブルデータ処理装置のプロセッサを介して実行される上記命令は、フローチャート及び／又はブロック図の１つ又は複数のブロック内に指定される機能／動作を実施する手段を作り出す。

【０１４４】

上記コンピュータプログラム命令は、コンピュータ、他のプログラマブルデータ処理装置、又は他のデバイスに特定の仕方で機能するように指示できるコンピュータ記憶媒体内にさらに記憶されてもよく、したがって、コンピュータ読取可能媒体内に記憶された上記命令は、フローチャート及び／又はブロック図の１つ又は複数のブロック内に指定される機能／動作を実施する命令を含む製造品を生じさせる。

【０１４５】

コンピュータプログラム命令は、さらに、コンピュータ、他のプログラマブルデータ処理装置、又は他のデバイスにロードされて、一連の動作ステップをコンピュータ、他のプログラマブル装置、又は他のデバイス上で実行させて、コンピュータ実施処理を生じさせてもよく、したがって、コンピュータ又は他のプログラマブル装置上で実行される上記命令は、フローチャート及び／又はブロック図の１つ又は複数のブロック内に指定される機能／動作を実施する処理を提供する。

【０１４６】

図１４は、記憶クラスタベースの完全同期及び半同期伝搬エンジンを備えた一例示的なコンピュータシステムを表す。コンピュータシステムは、プロセッサユニット１４０１（可能性として複数のプロセッサ、複数のコア、複数のノードを含み、かつ／あるいはマルチスレッディングを実施するなどする）を含む。コンピュータシステムは、メモリ１４０７を含む。メモリ１４０７は、システムメモリ（例えば、キャッシュ、ＳＲＡＭ、ＤＲＡＭ、ゼロキャパシタＲＡＭ、ツイントランジスタＲＡＭ、ｅＤＲＡＭ、ＥＤＯ　ＲＡＭ、ＤＤＲ　ＲＡＭ、ＥＥＰＲＯＭ、ＮＲＡＭ、ＲＲＡＭ（登録商標）、ＳＯＮＯＳ、ＰＲＡＭ等のうち１つ以上）又は上記ですでに説明されたマシン読取可能媒体のとり得る具現化のうち任意の１つ以上であり得る。コンピュータシステムは、バス１４０３（例えば、ＰＣＩ、ＩＳＡ、ＰＣＩエクスプレス、ＨｙｐｅｒＴｒａｎｓｐｏｒｔ（登録商標）バス、ＩｎｆｉｎｉＢａｎｄ（登録商標）バス、ＮｕＢｕｓ等）と、ネットワークインターフェース１４０５（例えば、ＡＴＭインターフェース、イーサネット（登録商標）インターフェース、フレームリレーインターフェース、ＳＯＮＥＴインターフェース、無線インターフェース、ｉＳＣＳＩ、ファイバチャネル等）とをさらに含む。コンピュータシステムは、記憶クラスタベース粒状（storage cluster based granular）完全同期及び半同期伝搬エンジン１４１１をさらに含む。記憶クラスタベース粒状完全同期及び半同期伝搬エンジン１４１１は、上記で説明されたとおり、同期関係におけるエンドポイントを対象にするファイルシステム変更要求に対応する要求及び応答を扱う。こうした機能性のうち任意のものが、ハードウェアにおいて及び／又は処理ユニット１４０１上で部分的に（又は全体的に）実施されてよい。例えば、上記機能性は、特定用途向け集積回路を用いて、処理ユニット１４０１内に実装されるロジック内で、周辺デバイス又はカード上のコプロセッサ内でなどで実施されてよい。さらに、具現化には、より少ないコンポーネント、又は図１４内に例示されていないさらなるコンポーネント（例えば、ビデオカード、オーディオカード、さらなるネットワークインターフェース、周辺デバイス等）を含んでもよい。処理ユニット１４０１、（１つ以上の）記憶装置１４０９、及びネットワークインターフェース１４０５は、バス１４０３に結合される。メモリ１４０７は、バス１４０３に結合されるものとして例示されているが、処理ユニット１４０１に結合されてもよい。

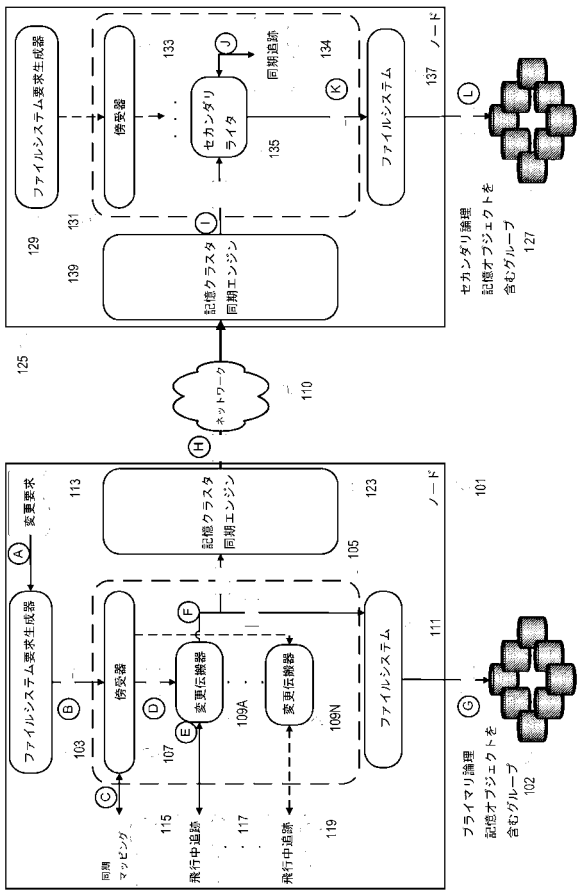
【０１４７】

本開示の態様は、様々な実装及び活用を参照して説明されているが、本開示のこうした態様は例示であり、発明対象事項の範囲はこれらに限定されないことが理解されるであろう。一般に、本明細書に説明されるクラスタノードにわたる論理記憶オブジェクト粒度同期の手法は、任意の１つ又は複数のハードウェアシステムに調和する設備を用いて実装されてよい。多くのバリエーション、修正、追加、及び改善が可能である。

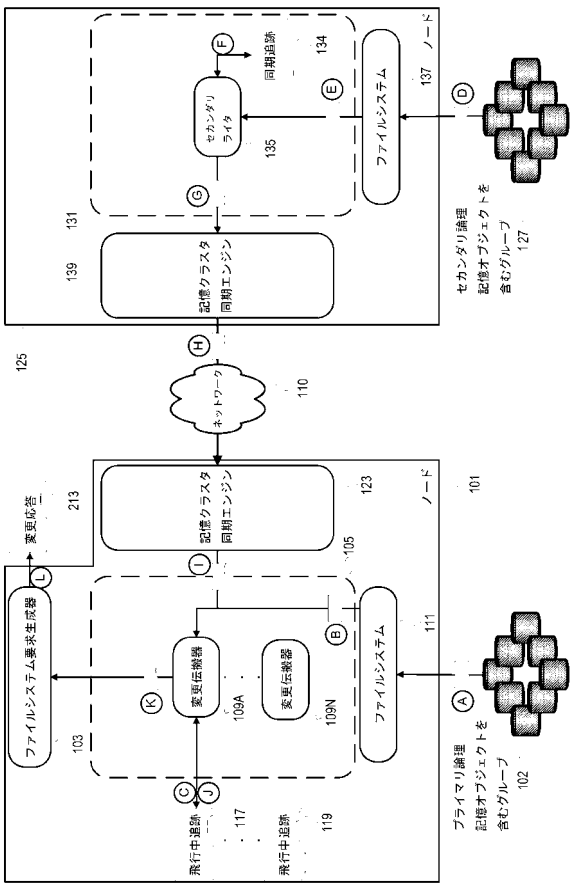
【０１４８】

本明細書において単一のインスタンスとして説明されるコンポーネント、動作、又は構造について、複数のインスタンスが提供されてもよい。最後、様々なコンポーネント、動作、及びデータストア間の境界はいくらか任意的であり、特定の動作が具体的な例示構成の文脈の中で例示されている。機能性についての他の割り振りが想定され、発明対象事項の範囲内に入り得る。一般に、例示構成の中で別個のコンポーネントとして提示される構造及び機能性は、組み合わせられた構造又はコンポーネントとして実装されてもよい。同様に、単一のコンポーネントとして提示される構造及び機能性が、別個のコンポーネントとして実装されてもよい。上記及び他のバリエーション、修正、追加、及び改善が発明対象事項の範囲内に入り得る。

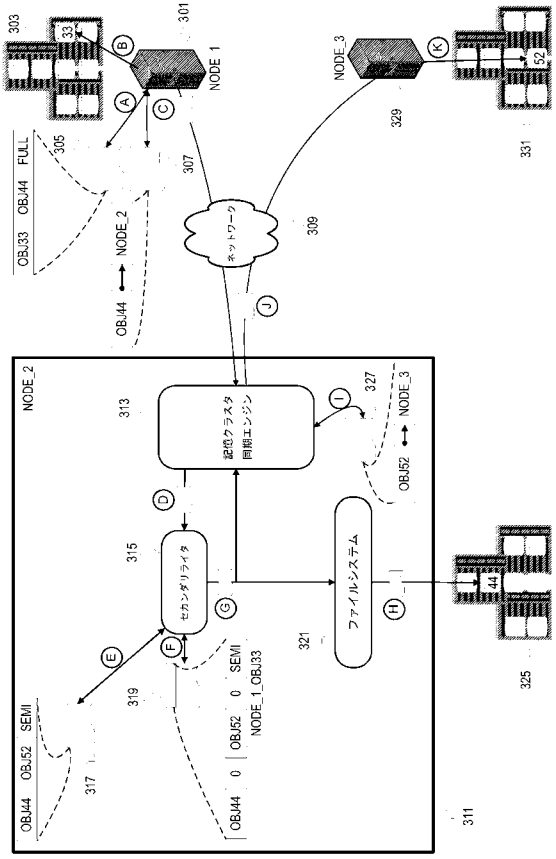
【図１】



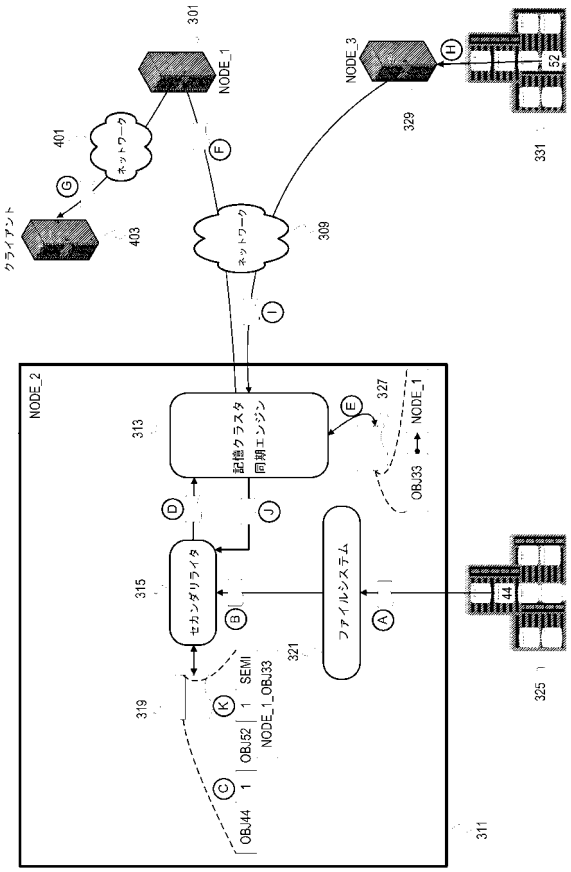
【図２】



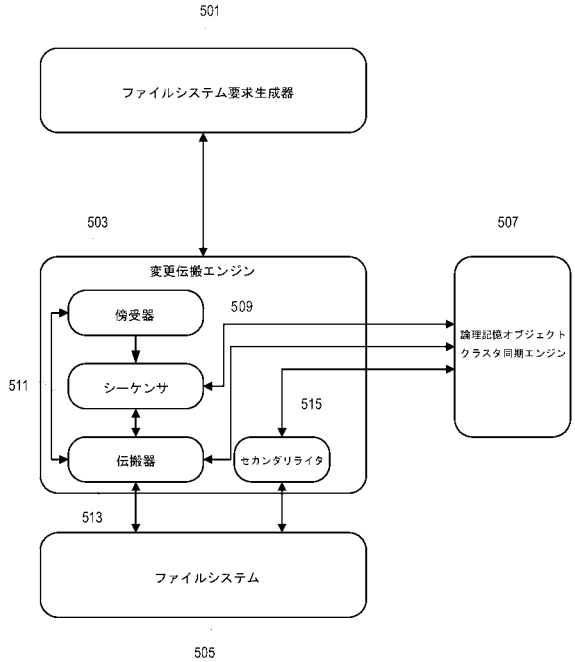
【図３】


NODE_1
NODE_2
NODE_3
ネットワーク
記憶クラスタ同期エンジン
セカンダリライタ
ファイルシステム
OBJ44 OBJ52 SEMI
OBJ44 0 OBJ52 0 SEMI NODE_1_OBJ33
OBJ52
OBJ33 OBJ44 FULL

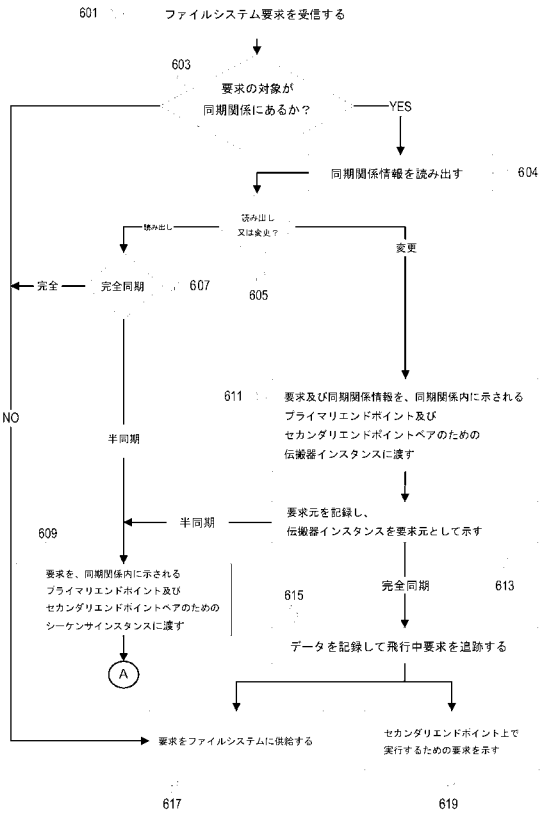
【図４】


NODE_1
NODE_2
NODE_3
ネットワーク
クライアント
記憶クラスタ同期エンジン
セカンダリライタ
ファイルシステム
OBJ44 1 OBJ52 1 SEMI NODE_1_OBJ33
OBJ33

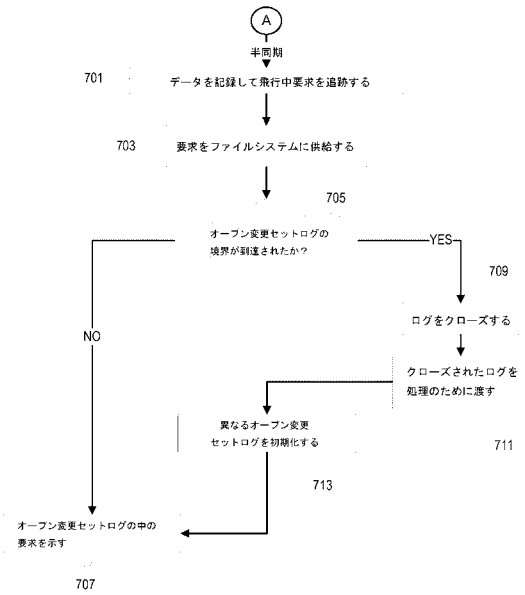
【図５】


ファイルシステム要求生成器
変更伝搬エンジン
傍受器
シーケンサ
伝搬器
セカンダリライタ
論理記憶オブジェクトクラスタ同期エンジン
ファイルシステム

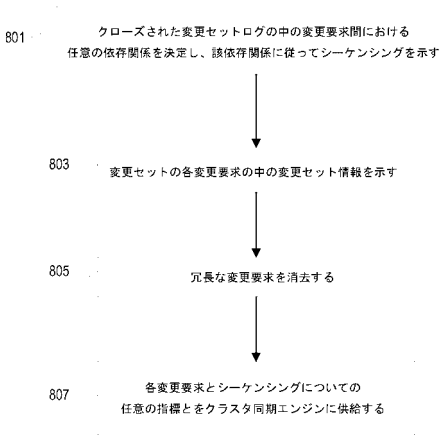
【図６】


ファイルシステム要求を受信する
要求の対象が同期関係にあるか？
YES
NO
同期関係情報を読み出す
読み出し又は変更？
読み出し
変更
完全
完全同期
半同期
要求及び同期関係情報を、同期関係内に示されるプライマリエンドポイント及びセカンダリエンドポイントペアのための伝搬器インスタンスに渡す
要求元を記録し、伝搬器インスタンスを要求元として示す
要求を、同期関係内に示されるプライマリエンドポイント及びセカンダリエンドポイントペアのためのシーケンサインスタンスに渡す
データを記録して飛行中要求を追跡する
要求をファイルシステムに供給する
セカンダリエンドポイント上で実行するための要求を示す

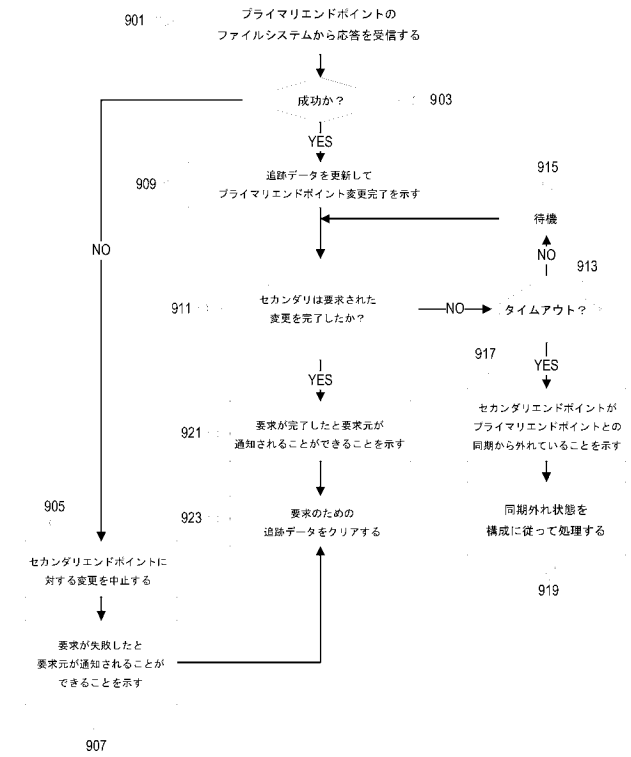
【図７】


A
半同期
701
データを記録して飛行中要求を追跡する
703
要求をファイルシステムに供給する
705
オープン変更セットログの
境界が到達されたか？
YES
709
ログをクローズする
NO
クローズされたログを
処理のために渡す
異なるオープン変更
セットログを初期化する
711
713
オープン変更セットログの中の
要求を示す
707

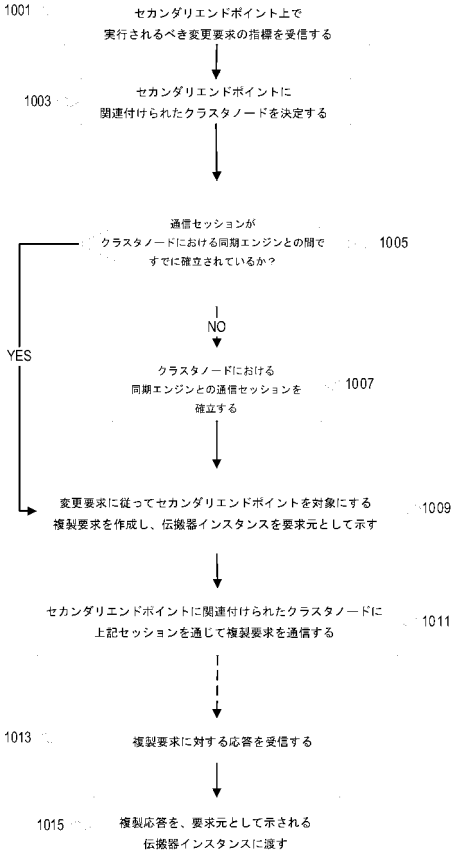
【図８】


801
クローズされた変更セットログの中の変更要求間における
任意の依存関係を決定し、該依存関係に従ってシーケンシングを示す
803
変更セットの各変更要求の中の変更セット情報を示す
805
冗長な変更要求を消去する
807
各変更要求とシーケンシングについての
任意の指標とをクラスタ同期エンジンに供給する

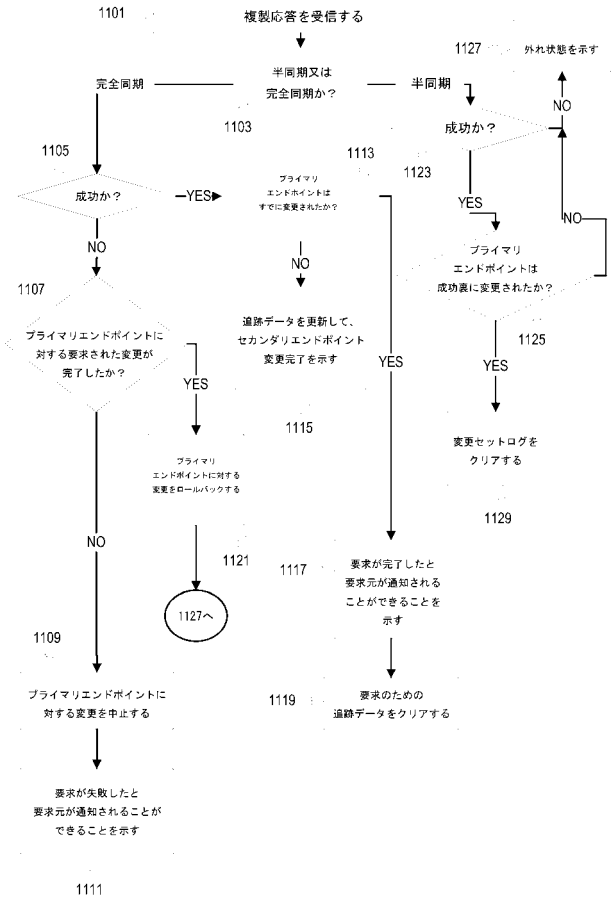
【図９】


901
プライマリエンドポイントの
ファイルシステムから応答を受信する
成功か？
903
YES
909
追跡データを更新して
プライマリエンドポイント変更完了を示す
915
待機
NO
NO
913
911
セカンダリは要求された
変更を完了したか？
NO
タイムアウト？
917
YES
YES
921
要求が完了したと要求元が
通知されることができることを示す
セカンダリエンドポイントが
プライマリエンドポイントとの
同期から外れていることを示す
905
923
要求のための
追跡データをクリアする
同期外れ状態を
構成に従って処理する
919
セカンダリエンドポイントに
対する変更を中止する
要求が失敗したと
要求元が通知されることが
できることを示す
907

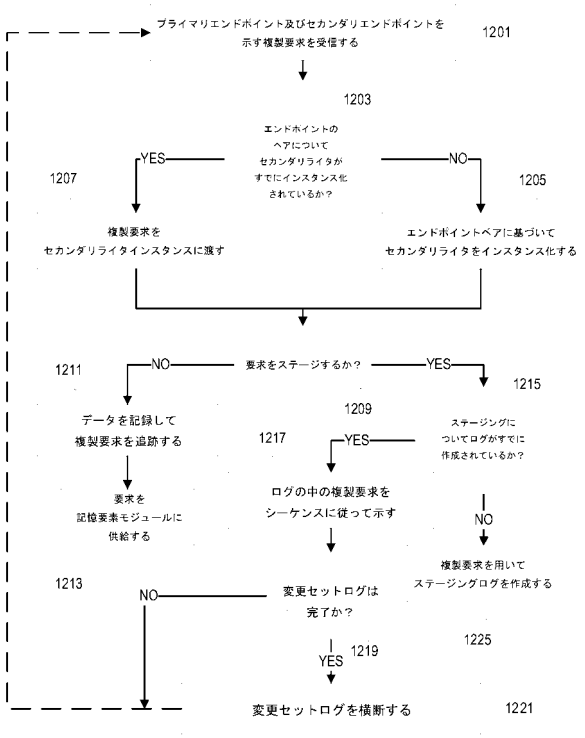
【図１０】


1001
セカンダリエンドポイント上で
実行されるべき変更要求の指標を受信する
1003
セカンダリエンドポイントに
関連付けられたクラスタノードを決定する
通信セッションが
クラスタノードにおける同期エンジンとの間で
すでに確立されているか？
1005
NO
YES
クラスタノードにおける
同期エンジンとの通信セッションを
確立する
1007
変更要求に従ってセカンダリエンドポイントを対象にする
複製要求を作成し、伝搬器インスタンスを要求元として示す
1009
セカンダリエンドポイントに関連付けられたクラスタノードに
上記セッションを通じて複製要求を通信する
1011
1013
複製要求に対する応答を受信する
1015
複製応答を、要求元として示される
伝搬器インスタンスに渡す

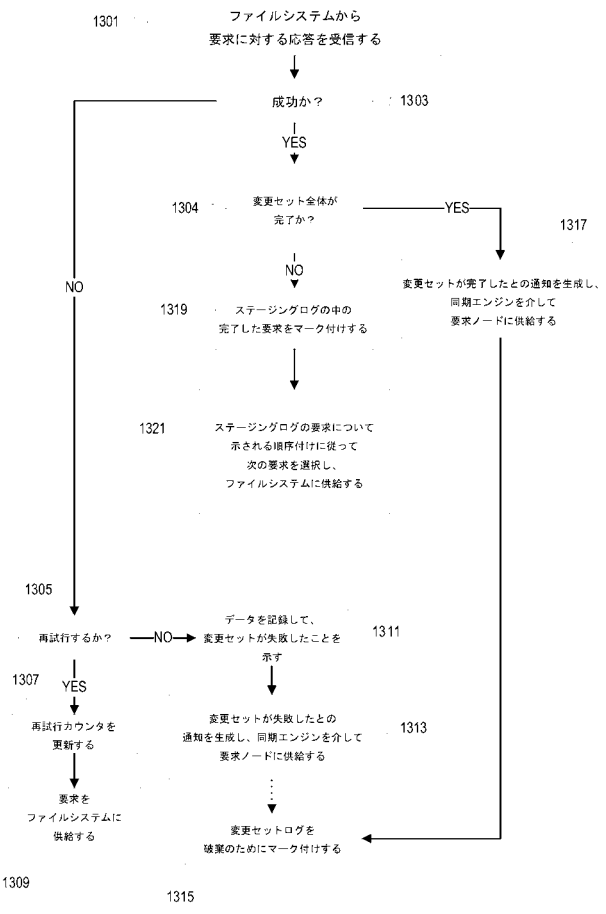
【図１１】



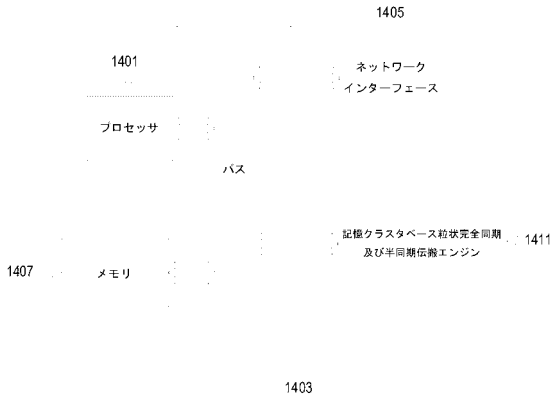
【図１２】



【図１３】



【図１４】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/043159

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30 H04L29/08
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, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 395 352 B1 (LAM SAHN [US] ET AL) 1 July 2008 (2008-07-01) column 1, line 51 - line 62 column 3, line 48 - line 60 -----	1-20
X	US 2011/066592 A1 (NEWPORT WILLIAM T [US] ET AL) 17 March 2011 (2011-03-17) page 1, left-hand column, paragraph 0008 - page 1, left-hand column, paragraph 0008 figure 4 page 3, right-hand column, paragraph 0030 ----- -/--	1-20

[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
20 October 2015	28/10/2015

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	Stan, Johann

4

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/043159

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 7 039 661 B1 (RANADE DILIP M [IN]) 2 May 2006 (2006-05-02) column 3, line 5 - line 13 column 7, line 58 - line 64 column 8, line 29 - line 35 column 5, line 18 - line 21 column 5, line 25 - line 45	2,6,10, 14,17
A	US 8 495 250 B2 (ANANTHANARAYANAN RAJAGOPOL [US] ET AL) 23 July 2013 (2013-07-23) column 7, line 4 - line 26	2,6,10, 14,17
A	US 8 667 236 B2 (PHELPS ADAM M [US] ET AL) 4 March 2014 (2014-03-04) column 2, line 57 - column 3, line 5	2,6,10, 14,17
A	US 2005/193041 A1 (BOURBONNAIS SERGE [US] ET AL) 1 September 2005 (2005-09-01) the whole document	2,6,10, 14,17
A	US 7 571 391 B2 (ROESSLER ANDREAS [DE]) 4 August 2009 (2009-08-04) the whole document	3,11,18
A	"Sun StorageTek (TM) Availability Suite 4.0 Remote Mirror Software Administration Guide", , 30 April 2006 (2006-04-30), XP055220220, Retrieved from the Internet: URL:https://docs.oracle.com/cd/E19359-01/8 19-6148-10/819-6148-10.pdf [retrieved on 2015-10-12] page 19 - page 32	1,8,9,16

4

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

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2015/043159

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7395352 B1	01-07-2008	NONE	
US 2011066592 A1	17-03-2011	JP 5496839 B2	21-05-2014
		JP 2011060292 A	24-03-2011
		KR 20110029071 A	22-03-2011
		US 2011066592 A1	17-03-2011
		US 2013041869 A1	14-02-2013
US 7039661 B1	02-05-2006	US 7039661 B1	02-05-2006
		US 7606841 B1	20-10-2009
US 8495250 B2	23-07-2013	US 2011145499 A1	16-06-2011
		US 2012311065 A1	06-12-2012
US 8667236 B2	04-03-2014	CN 103124961 A	29-05-2013
		EP 2622487 A2	07-08-2013
		US 2012079222 A1	29-03-2012
		WO 2012050904 A2	19-04-2012
US 2005193041 A1	01-09-2005	US 2005193041 A1	01-09-2005
		US 2007288537 A1	13-12-2007
		US 2008163222 A1	03-07-2008
US 7571391 B2	04-08-2009	NONE	

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,ST,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,KM,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,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US

(72)発明者 コートニー，スーザン，エム.
アメリカ合衆国 カリフォルニア州 ９４０８９ サニーベール イースト ジャヴァ ドライヴ ４９５
(72)発明者 ムゥ，ユエドーン
アメリカ合衆国 カリフォルニア州 ９４０８９ サニーベール イースト ジャヴァ ドライヴ ４９５
(72)発明者 ラオ，サントシュ
アメリカ合衆国 カリフォルニア州 ９４０８９ サニーベール イースト ジャヴァ ドライヴ ４９５