(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2016-534471
(P2016-534471A)
(43)公表日　平成28年11月4日(2016.11.4)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 11/14	(2006.01)	G06F	11/14	669	5B027
G06F 13/10	(2006.01)	G06F	13/10	340A	
G06F 3/06	(2006.01)	G06F	3/06	304F	
G06F 12/00	(2006.01)	G06F	3/06	306Z	
		G06F	11/14	664	

審査請求 有　予備審査請求 未請求　（全 53 頁）　最終頁に続く

(21)出願番号　特願2016-540866 (P2016-540866)
(86)(22)出願日　平成25年10月18日 (2013.10.18)
(85)翻訳文提出日　平成28年3月2日 (2016.3.2)
(86)国際出願番号　PCT/US2013/065623
(87)国際公開番号　WO2015/057240
(87)国際公開日　平成27年4月23日 (2015.4.23)

(71)出願人　505160898
ヒタチ　データ　システムズ　エンジニアリング　ユーケー　リミテッド
ＨＩＴＡＣＨＩ　ＤＡＴＡ　ＳＹＳＴＥＭＳ　ＥＮＧＩＮＥＥＲＩＮＧ　ＵＫ　ＬＩＭＩＴＥＤ
イギリス国　アールジー１２　８エフゼット　ブラックネル，オールドベリー，キャピトル　ビルディング
(74)代理人　110000279
特許業務法人ウィルフォート国際特許事務所

最終頁に続く

(54)【発明の名称】シェアード・ナッシング分散型ストレージ・システムにおけるターゲットにより駆動される独立したデータの完全性および冗長性のリカバリ

(57)【要約】
ネットワークに接続された独立したノードと、ストレージ・システムに記憶されたデータに対するＩ／Ｏ処理を送るホストとを含む分散型シェアード・ナッシング・ストレージ・システム。各ノードは、部分的に書き込まれたデータのそのノード自体のリカバリを実行し、ストレージ・システムに記憶されたデータの整合性を維持することができる。ノードが、ノード中のすべてのデータの位置を独立して計算し、独立してデータの釣り合いを取り、ストレージ・システムの冗長性のポリシーに基づいて整合性を維持し、位置の変更に応じてデータをマイグレーションする。ノードは、そのノードが不完全なまたは損傷したデータを記憶すると判定する場合、その他のノード上のレプリカ・データからそのノードが記憶するデータを再構築する。ノードの間のデータのマイグレーション中、ホストからのＩ／Ｏ処理は、妨げられない。

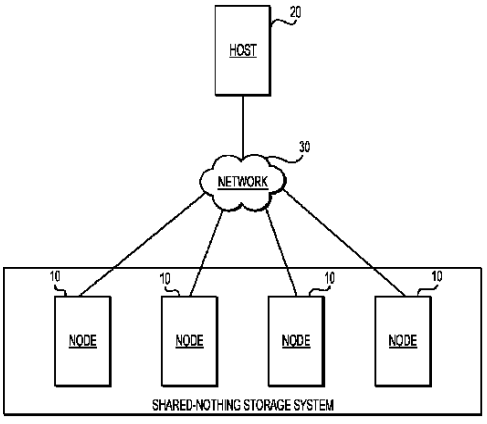


FIG. 1

【特許請求の範囲】
【請求項１】
１つまたは複数のネットワークに接続されたクラスタ内の複数の独立したノードであって、それぞれが、データを記憶するための１つまたは複数のデバイスを含み、
それぞれの独立したノードが、入力／出力（Ｉ／Ｏ）オペレーションを送り、受け取るように構成され、
前記クラスタ内の前記複数のノードに記憶された前記データが、前記データレプリカを含み、各ノードが、位置のマップに基づいて前記クラスタ内のストレージ・デバイス上に記憶された前記データおよび前記データレプリカの位置を知っている、複数の独立したノードと、
前記クラスタ内のすべてのノードが利用可能である、各ノードおよび各ノードの各ストレージ・デバイスの状態を示す状態情報とを含み、
前記クラスタ内でノードがアクティブ化もしくは非アクティブ化されるかまたはいずれかのノードのストレージ・デバイスがアクティブ化もしくは非アクティブ化されると、前記状態情報が、対応するがアクティブ化もしくは非アクティブ化に基づいて更新され、変更され、
前記状態情報の変更に基づいて、新しくアクティブ化されたノードまたは新しくアクティブ化されたストレージ・デバイスを有するノードを含む前記クラスタ内の各ノードが、前記クラスタ内の前記ノード上に記憶されるいずれかのデータまたはデータレプリカの整合性を維持するためにリカバリオペレーションを前記ノードが実行しなければならないかどうかを判定し、
各ノードが、影響を与えられるデータを記憶する前記クラスタ内のその他のノードにメッセージを送信し、受信することによって、前記クラスタ内のその他のノードとは独立して必要に応じて前記リカバリオペレーションを実行して、前記ノードのストレージ・デバイスに記憶された前記データを復元する分散型シェアード・ナッシング・ストレージ・システム。
【請求項２】
前記クラスタ内のデータが、前記クラスタ内の複数のノードにレプリカまたは消失訂正符号化されたセグメントとして冗長的に記憶され、
レプリカまたは消失訂正符号化されたセグメントの数が、前記データの冗長性のポリシーに依存し、
レプリカまたは消失訂正符号化されたセグメントの符号化が、前記データの前記冗長性のポリシーおよび消失訂正符号化アルゴリズムに依存する請求項１に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項３】
各ノードが、イベントの通知を受信し、前記イベントが、前記クラスタ内のノードの前記アクティブ化もしくは前記非アクティブ化またはいずれかのノードのストレージ・デバイスの前記アクティブ化もしくは前記非アクティブ化である請求項２に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項４】
各イベントが、厳密な順番で前記クラスタ内の各ノードに届けられ、前記イベントが、単調な昇順で一意で整合性のある番号を振られ、
各ノードが、前記クラスタ内のすべてのイベントを受信することを保証され、
非アクティブ化の後でアクティブ化された各ノードが、前記ノードがアクティブ化されるときにすべてのイベントを受信することを保証される請求項１に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項５】
前記状態情報が、各ノードのストレージ・デバイスに記憶されたデータおよび各ノードのストレージ・デバイス上のデータレプリカまたは消失訂正符号化されたセグメントの状態、ならびに前記クラスタ内のすべての前記その他のノードのストレージ・デバイス上の

データおよびデータレプリカまたは消失訂正符号化されたセグメントの状態についての情報を含み、
前記データ、前記データレプリカ、または前記消失訂正符号化されたセグメントの前記状態が、
前記データを修正または書き込みする前にデータ、データレプリカ、または消失訂正符号化されたセグメントに関して設定されるＤＩＲＴＹフラグ、
データの書き込みまたは修正がデータレプリカまたは消失訂正符号化されたセグメントの単に一部(less than all)で成功したことを前記ノードが知っているときにデータ、前記データレプリカ、または前記消失訂正符号化されたセグメントに関して設定されるＤＥＧＲＡＤＥＤフラグ、
イベント番号の後に前記データが書き込まれた前記イベント番号を示す情報、および
イベント番号の後に前記ＤＥＧＲＡＤＥＤフラグが設定された前記イベント番号を示す情報を含み、
前記ＤＩＲＴＹフラグが永続的であり、前記ＤＥＧＲＡＤＥＤフラグが永続的であり、
前記ＤＩＲＴＹフラグが、前記データのデータレプリカまたは消失訂正符号化されたセグメントの少なくとも予め決められた最小限の数に対する書き込みが正常に完了したときにクリアされ、
前記データおよび前記データレプリカまたは前記消失訂正符号化されたセグメントの位置が、永続的に記憶される可能性があり、または前記データの前記位置のマップおよびメタデータから必要に応じて計算される可能性があり、
前記メタデータが、一意のＩＤ、前記データの名前空間内の一意の名前属性、または前記データの内容の記述のいずれかを含む請求項３に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項６】
前記クラスタ内の各ノードによって受信される前記イベントが、非アクティブ化の後にアクティブ化された第１のノードのアクティブ化または第１のノードのストレージ・デバイスのアクティブ化であり、
前記判定が、前記位置のマップに基づいていずれかのノードが前記第１のノードに記憶されたいずれかのデータ、データレプリカ、または消失訂正符号化されたセグメントを有するかどうかに基づき、
前記第１のノードが、前記ＤＥＧＲＡＤＥＤフラグ、前記ＤＩＲＴＹフラグ、前記データの最後の修正または書き込みのイベント番号を示す情報、およびイベント番号の後に前記ＤＥＧＲＡＤＥＤフラグが設定された前記イベント番号を示す前記情報を含む情報のリストを受信し、
情報の前記リストに基づいて、前記第１のノードが、どのデータ、レプリカ、または消失訂正符号化されたセグメントが再構築される必要があるかを判定し、
前記状態情報に基づいて、前記第１のノードが、前記第１のノードのストレージ・デバイス上の前記データ、前記データレプリカ、または前記消失訂正符号化されたセグメントを再構築するために再構築される必要がある前記第１のノード上のデータの前記レプリカまたは前記消失訂正符号化されたセグメントを前記クラスタ内のどのノードが記憶するかを判定し、
前記第１のノードが、再構築される必要がある前記データを記憶すると判定された前記クラスタ内の前記ノードからの、再構築される前記データ、前記データレプリカ、または前記消失訂正符号化されたセグメントの内容を要求し、再構築すべき前記データの前記内容を書き込み、完了すると、情報の前記リストを送信したすべてのノードにメッセージを送信し、
前記データのすべてのデータレプリカまたは消失訂正符号化されたセグメントが前記クラスタ内の各ノードでリカバリされると判定される場合に、前記クラスタ内のすべてのノードが、前記データ、前記データレプリカ、または前記消失訂正符号化されたセグメントの前記ＤＥＧＲＡＤＥＤフラグをクリアする請求項５に記載の分散型シェアード・ナッシ

ング・ストレージ・システム。
【請求項７】
ネットワークに接続されたクラスタ内の複数の独立したノードであって、それぞれが、データを記憶するための１つまたは複数のストレージ・デバイスを含み、
それぞれのノードが、入力／出力（Ｉ／Ｏ）オペレーションを送り、受け取るように構成され、
各ノードが、ローカルの処理を行うための、前記ストレージ・デバイスとは別であるメモリを有する、複数の独立したノードを含み、
Ｉ／Ｏオペレーションが、個々のノードで実行され、Ｉ／Ｏオペレーションを受け取る各ノードが、前記ノードのストレージ・デバイスに記憶された前記データに関してのみ前記Ｉ／Ｏオペレーションを実行し、
前記クラスタ内の前記ストレージ・デバイスに記憶されたデータの位置が、すべての前記ノードが利用可能であるレイアウト情報から導出され、
前記データの前記位置が、記憶されるか、前記レイアウト情報から必要に応じて各ノードによって計算され、
イベントの発生を示す第１のイベント・メッセージを受信すると、各ノードが、前記ノードのストレージ・デバイス上に記憶されたすべてのデータを評価し、いずれかのデータが前記第１のイベント・メッセージの情報によって影響を与えられるかどうかを判定する分散型シェアード・ナッシング・ストレージ・システム。
【請求項８】
各ノードおよび各ノードの各ストレージ・デバイスがアクティブ状態であるかどうかを示す、すべてのノードが利用可能な状態情報をさらに含み、
レイアウト情報が、
前記ストレージ・システム内の各ノードおよびストレージ・デバイス上のデータの位置を決定し、Ｉ／Ｏ処理のためにデータの前記位置を判定するために各ノードによって使用される現在のレイアウト・マップと、
前記ストレージ・システム内の各ノードおよびストレージ・デバイス上のデータの提案される位置を決定し、Ｉ／Ｏオペレーションのためにデータの前記位置を決定するために各ノードによって使用され、前記第１のイベント・メッセージを受信すると各ノードにおいて生成される提案されるレイアウト・マップであって、前記現在のレイアウト・マップと比較して、前記イベントに基づいて少なくとも１つのデータの位置を変更する、提案されるレイアウト・マップとを含み、
前記第１のイベント・メッセージが、前記イベントが発生すると現在のマップ・レイアウトの下で前記クラスタ内のすべてのノードに決められた順番に送信され、前記第１のイベント・メッセージの前記情報が、イベントの種類を示し、
各ノードが、メモリに、前記現在のレイアウト・マップおよび前記提案されるレイアウト・マップを記憶し、
すべてのノードが第２のイベント・メッセージを受信すると、前記ノードが、現在のレイアウト・マップの下の位置から前記提案されるレイアウト・マップの下の位置へのデータのマイグレーションを開始し、
前記現在のレイアウト・マップが、第３のイベント・メッセージが受信されると、すべての前記ノードによって前記提案されるレイアウト・マップに置き換えられ、
前記現在のレイアウト・マップおよび前記提案されるレイアウト・マップの下の対応するノードが、マイグレーションされている前記データに関するＩ／Ｏオペレーションを受け付け続ける請求項７に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項９】
前記イベントが、前記クラスタ内のノードのアクティブ化もしくは非アクティブ化またはいずれかのノードのストレージ・デバイスのアクティブ化もしくは非アクティブ化であり、
前記第１のイベント・メッセージが受信されると、各ノードが、前記ノードがソース・

ノードであるかどうかを判定し、各ソース・ノードが、前記提案されるレイアウト・マップの下のデータの前記位置のノードに、マイグレーション要求を送信することによって、前記ノードが前記データの内容を受信すべきであることを知らせ、
マイグレーション要求を受信する各ノードが、前記提案されるレイアウト・マップの下で前記ノードが記憶する前記データの前記内容を獲得するリカバリ要求を送信する請求項８に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項１０】
前記クラスタ内のデータが、前記クラスタ内の複数のノードにレプリカまたは消失訂正符号化されたセグメントとして冗長的に記憶され、
レプリカまたは消失訂正符号化されたセグメントの数が、前記データの冗長性のポリシーに依存し、
前記レプリカまたは前記消失訂正符号化されたセグメントの位置が、前記現在のレイアウト・マップおよび前記提案されるレイアウト・マップに含まれ、
レプリカまたは消失訂正符号化されたセグメントの符号化が、前記ストレージ・システムの前記冗長性のポリシーおよび消失訂正符号化アルゴリズムに依存し、
前記データレプリカおよび消失訂正符号化されたセグメントの位置が、前記レイアウト情報から導出される請求項９に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項１１】
データに関するＩ／Ｏオペレーションが、前記現在のマップ・レイアウトと提案されるマップ・レイアウトとの両方の下で前記データに関する前記レプリカまたは前記消失訂正符号化されたセグメントを記憶するすべてのノードに送られ、
前記提案されるマップ・レイアウトの下のノードが、Ｉ／Ｏオペレーションが影響を与えるデータを記憶する前記現在のマップ・レイアウトの下のノードから前記データに関する前記Ｉ／Ｏオペレーションを受け取る請求項１０に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項１２】
すべてのノードが第２のイベント・メッセージを受信した後、Ｉ／Ｏオペレーションを受け取る前記提案されるマップ・レイアウトの下のノードが、前記Ｉ／Ｏオペレーションが影響を与える前記データが前記提案されるマップ・レイアウトの下の前記ノードにマイグレーションされたか否かを判定し、
前記Ｉ／Ｏオペレーションが影響を与える前記データがマイグレーションされていない場合、前記提案されるマップ・レイアウトの下の前記ノードが、前記オペレーションを実行するために前記データがマイグレーションされるまで待ち、
前記Ｉ／Ｏオペレーションが影響を与える前記データがマイグレーションされた場合、前記提案されるマップ・レイアウトの下の前記ノードが、前記データに対して前記Ｉ／Ｏオペレーションを実行する請求項１１に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項１３】
前記第１のイベント・メッセージによって影響を与えられる判定された各データに関して、前記ノードが、前記ノードが前記現在のマップ・レイアウトの下の前記データのコピーを有するが、前記提案されるマップ・レイアウトの下の前記データのコピーを持たないかどうか、または前記ノードが前記現在のマップ・レイアウトと前記提案されるマップ・レイアウトとの両方の下のデータを記憶するかどうか、または前記ノードが前記提案されるレイアウト・マップの下のノードに、前記ノードが記憶すべきデータを有し、前記データが前記位置にマイグレーションされるべきであることを知らせる役割を担うノードであるかどうかを判断し、
前記ノードが前記提案されるマップ・レイアウトの下のノードに知らせる役割を担うかどうかの判定が、レイアウト情報および状態情報に基づく請求項１２に記載の分散型シェアード・ナッシング・ストレージ・システム。

【請求項１４】
前記データが、データのサブセットであり、有限のサイズであるスライスでマイグレーションされ、
前記Ｉ／ＯオペレーションがＷＲＩＴＥ要求である場合、および前記ＷＲＩＴＥ要求が影響を与えるデータがマイグレーションされていない場合、前記提案されるマップ・レイアウトの下の前記ノードが、前記ＷＲＩＴＥ要求のサイズがスライスのサイズ未満であるかどうかを判定し、
前記ＷＲＩＴＥ要求の前記サイズがスライスの前記サイズ未満である場合、前記ＷＲＩＴＥ要求が当てはまる前記データのスライスが直ちにマイグレーションされ、
前記ＷＲＩＴＥ要求の前記サイズがスライスの前記サイズ未満でないか、または前記スライスの前記サイズに等しい場合、前記ＷＲＩＴＥ要求に関して書き込まれるデータが、マイグレーションされるスライスに関して書き込まれ、前記スライスが、マイグレーションされない請求項１２に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項１５】
分散型シェアード・ナッシング・ストレージ・システムであって、
ネットワークに接続されたクラスタ内の複数の独立したノードであって、それぞれが、データを記憶するための１つまたは複数のストレージ・デバイスを含み、
それぞれのノードが、入力／出力（Ｉ／Ｏ）オペレーションを送り、受け取るように構成される、複数の独立したノードと、
各ノードおよび各ノードの各ストレージ・デバイスがアクティブ状態であるかどうかを示す、すべてのノードが利用可能な状態情報とを含み、
前記クラスタ内のデータが、前記クラスタ内の複数のノードにレプリカまたは消失訂正符号化されたセグメントとして冗長的に記憶され、
レプリカまたは消失訂正符号化されたセグメントの数が、前記データの冗長性のポリシーに依存し、
レプリカまたは消失訂正符号化されたセグメントの符号化が、前記データの前記冗長性のポリシーおよび消失訂正符号化アルゴリズムに依存し、
前記クラスタ内の前記ノードの前記ストレージ・デバイス上の前記データおよび前記データの前記レプリカまたは前記消失訂正符号化されたセグメントの位置が、すべての前記ノードが利用可能であるレイアウト情報から導出され、
前記データの位置が、記憶されるか、前記レイアウト情報から必要に応じて計算され、
ストレージ・システムに関する冗長性のポリシーが、前記データのデータレプリカもしくは消失訂正符号化されたセグメントの数を示すミラー・カウント数を設定し、データレプリカもしくは消失訂正符号化されたセグメントが、前記クラスタ内の前記ストレージ・デバイス上の第１の位置から第２の位置にマイグレーションされ、またはデータ、データレプリカ、もしくは消失訂正符号化されたセグメントが、ストレージ・システムの前記冗長性のポリシーの変更に基づいて前記クラスタから削除される、分散型シェアード・ナッシング・ストレージ・システム。
【請求項１６】
前記レイアウト情報が、
前記ストレージ・システム内の各ノードおよびストレージ・デバイス上のデータの位置を決定し、データ、データレプリカ、または消失訂正符号化されたセグメントの前記位置を判定するために各ノードによって使用される現在のレイアウト・マップと、
前記ストレージ・システム内の各ノードおよびストレージ・デバイス上のデータの提案される位置を決定し、データ、データレプリカ、または消失訂正符号化されたセグメントの前記位置を判定するために各ノードによって使用され、前記現在のレイアウト・マップの下の各ノードが第１のポリシー変更イベント・メッセージを受信すると各ノードで生成される提案されるレイアウト・マップとを含み、前記提案されるレイアウト・マップが、前記現在のレイアウト・マップと比較してポリシー・イベント・メッセージに含まれる前記冗長性のポリシーの変更に基づいて少なくとも１つのデータ、データレプリカ、または

消失訂正符号化されたセグメントの前記位置を変更する請求項１５に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項１７】
前記冗長性のポリシーの前記変更が、前記ミラー・カウントの増加であるかまたは前記ミラー・カウントの減少かのどちらかであり、
前記変更がミラー・カウントの減少である場合、前記変更が影響を与える前記データ、前記データレプリカ、または前記消失訂正符号化されたセグメントを記憶する各ノードが、前記ノードが前記提案されるレイアウト・マップの下の前記データを記憶するかどうかを判定し、前記ノードが前記提案されるレイアウト・マップの下の前記データ、前記レプリカ、または前記消失訂正符号化されたセグメントを記憶しない場合、前記ノードが、前記ノードのストレージ・デバイスから前記データ、前記レプリカ、または前記消失訂正符号化されたセグメントを削除し、
前記変更がミラー・カウントの増加である場合、前記提案されるレイアウト・マップが、前記データ、前記レプリカ、または前記消失訂正符号化されたセグメントの追加のミラーに関する新しい位置を決定し、
前記変更が影響を与える各データ、レプリカ、消失訂正符号化されたセグメントに関して、各ノードが、前記新しい位置のノードに、マイグレーション要求を送信することによって、前記ノードが前記変更が影響を与える前記データ、前記レプリカ、または前記消失訂正符号化されたセグメントの内容を受信すべきであることを知らせる請求項１６に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項１８】
マイグレーション要求を受信する各ノードが、前記提案されるレイアウト・マップの下で前記ノードが記憶する前記データ、前記データレプリカ、または消失訂正符号化されたセグメントの前記内容を獲得するリカバリ要求を送信し、
前記マイグレーションが完了すると、前記提案されるレイアウト・マップが前記現在のレイアウト・マップを置き換える請求項１７に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項１９】
前記データ、前記データレプリカ、または前記消失訂正符号化されたセグメントの前記新しい位置が、前記現在のレイアウト・マップに基づく前記データ、前記レプリカ、または前記消失訂正符号化されたセグメントの位置と前記提案されるレイアウト・マップに基づく前記データの位置とを比較することによって各ノードにより計算される請求項１７に記載の分散型シェアード・ナッシング・ストレージ・システム。
【請求項２０】
前記現在のレイアウト・マップおよび前記提案されるレイアウト・マップの下の対応するノードが、マイグレーションされている前記データに関するＩ／Ｏオペレーションを受け付け続け、
データに関するＩ／Ｏオペレーションが、現在のマップ・レイアウトと提案されるマップ・レイアウトとの両方の下で前記データに関する前記レプリカまたは前記消失訂正符号化されたセグメントを記憶するすべてのノードに送られ、
前記提案されるマップ・レイアウトの下のノードが、Ｉ／Ｏオペレーションが影響を与えるデータを記憶する前記現在のマップ・レイアウトの下のノードから前記データに関する前記Ｉ／Ｏオペレーションを受け取る請求項１５に記載の分散型シェアード・ナッシング・ストレージ・システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ネットワークによって接続された独立したコンピュータ・システム（ノード）のクラスタからなるシェアード・ナッシング分散型ストレージ・システムに関する。
【背景技術】

【０００２】
ストレージ・システムに記憶されるデータは、高い信頼性で保有されなければならず、入力／出力（Ｉ／Ｏ）処理は、ストレージ・システムにおけるデータのマイグレーション中に妨げられてはならない。たとえば、書き込みオペレーションに関して、マイグレーション中に、ストレージ・システムは、データ・オブジェクトの状態を高い信頼性で追跡しなければならない。

【０００３】
１つの知られている方法は、修正されるべきデータの領域が、たとえば、データを書き込む前に共通の／共有された「スコアボード（ｓｃｏｒｅｂｏａｒｄ）」上で「ダーティ」フラグによって印を付けられる書き込みのマーキングを使用する。この手法では、データを書き込む要求をログに記録するステップと、データを記憶する各ターゲットにメッセージを送信するステップと、書き込みおよび応答を待つステップと、それから、実際の書き込みオペレーションを送るステップとを含むいくつかのステップが、必要とされる。上述の方法は、書き込みオペレーションに関するネットワーク・レイテンシの増加につながる。

【０００４】
別の知られている記憶方法は、高レベルのデータ・ストレージ・エリア全体にダーティとして印を付ける。しかし、そのような手法は、データの大きな集合体全体のリカバリを必要とするので、大量のデータに対して実用的ではない。知られているストレージ・システムは、修正を示すためにファイル・システム・レベルでファイルにダーティとして印を付ける可能性もある。しかし、ファイル・システム・レベルのマーキングは、非常に大きなデータ・ファイルに関して有効であり得ないほど粗い粒度を有する印を付けられたデータをもたらし、その結果、完了し得ないほど長い期間を必要とするリカバリをもたらす。さらに、Ｐａｒａｓｃａｌｅ　Ｉｎｃ．のスケールアウト・ストレージ・プラットフォーム・ソフトウェアなど、集中型のデータベースにおいてデータのチャンクにダーティとして印を付けることも、当技術分野で知られている。

【０００５】
知られているストレージ・システムにおける同様の機能は、たとえば、参照により本明細書に援用される米国特許第６，９０７，５０７号、第６，９１０，１１１号、第７，０８９，３８５号、および第６，９７８，３５４号に記載されているＶＥＲＩＴＡＳ　Ｖｏｌｕｍｅ　Ｍａｎａｇｅｒ（ＶｘＶＭ）のＦａｓｔ　Ｍｉｒｒｏｒ　Ｒｅｓｙｎｃ（ＦＭＲ）の特徴をさらに含む。米国特許第６，９０７，５０７号、第６，９１０，１１１号、第７，０８９，３８５号、および第６，９７８，３５４号は、多カラムのビットマップ、アキュムレータのマップ、およびミラー毎のマップを使用する。Ｉ／Ｏエラーからのリカバリに関して、従来技術のストレージ・システム（ボリューム・マネージャおよび多コピー・ファイル・システム）は、データを直接読み取るかもしくは書き込むかのどちらかによってリカバリを実行するための中央マネージャを必要とするか、またはリカバリ・プロセスを管理するためのコーディネータを必要とする。そのような構成の欠点は、コーディネータが障害にあうときに集中的に管理されたリカバリが止まり、それがリカバリ・プロセスのさらに困難な状況につながることである。したがって、コーディネータの障害の可能性を考慮して、大量のメタデータが共有されたストレージに高い信頼性で保有されることが求められる。

【０００６】
部分的に書き込まれたデータのリカバリの場合、従来技術は、多くのボリューム・マネージャの実装で使用されるミラー再接続およびミラー「リシルバリング（ｒｅｓｉｌｖｅｒｉｎｇ）」手法からなり、それらの手法は、中央データベースまたは何らかの種類のボリューム・レベルのビットマップを使用する。その他の実装は、１つの中央の位置（すべてのボリューム・マネージャ）からの直接読み取りおよび書き込みを行う中央リカバリ・マネージャを使用するか、または（たとえば、Ｐａｒａｓｃａｌｅ　Ｉｎｃ．のスケールアウト・ストレージ・プラットフォーム・ソフトウェアのように）リカバリを駆動するた

めの中央コーディネータを有する。
【０００７】
ノードまたはノードのディスクがストレージ・システムにおいて追加または削除されるデータのマイグレーションを伴う場合、従来技術は、信頼性の高いハッシュおよびマップの生成に基づくＣＥＰＨファイル・システムの再レイアウトの特徴を含む。ＰＩＣＡＳＳＯシステムとＣＥＰＨシステムとの両方は、「ＣＲＵＳＨ」アルゴリズムとしてよく知られている配置アルゴリズムを用いて、ストレージ・クラスタ全体にまたがるストレージ構成のバージョン情報に基づいてデータ・チャンクの適切な配置を決定的に計算する。参照により本明細書に援用されるＳａｇｅ Ａ．Ｗｅｌｉ、Ｓｃｏｔｔ Ａ．Ｂｒａｎｄｔ、Ｅｔｈａｎ Ｌ．Ｍｉｌｌｅｒ、Ｃａｒｌｏｓ Ｍａｌｔｚａｈｎ、「ＣＲＵＳＨ：Ｃｏｎｔｒｏｌｌｅｄ，Ｓｃａｌａｂｌｅ，Ｄｅｃｅｎｔｒａｌｉｚｅｄ Ｐｌａｃｅｍｅｎｔ ｏｆ Ｒｅｐｌｉｃａｔｅｄ Ｄａｔａ」、Ｐｒｏｃｅｅｄｉｎｇｓ ｏｆ ｔｈｅ ２００６ ＡＣＭ／ＩＥＥＥ Ｃｏｎｆｅｒｅｎｃｅ ｏｎ Ｓｕｐｅｒｃｏｍｐｕｔｉｎｇ、ｐ．３１、２００６を参照されたい。ＣＥＰＨシステムにおいて、再レイアウトは、中央メタデータ・エンジンによって実行される。さらに、Ｐａｒａｓｃａｌｅシステムにおいて、データの再レイアウトは、中央データベースによって駆動され、配置は、アドホックにチャンク毎に行われる。Ｐａｒａｓｃａｌｅシステムにおける再レイアウトが妨げられるとき、データのレイアウトは過渡的であるが整合性のある状態で残され、再レイアウト・プロセスが再開すると、データの配置が再計算される。
【先行技術文献】
【特許文献】
【０００８】
【特許文献１】米国特許第６，９０７，５０７号
【特許文献２】米国特許第６，９１０，１１１号
【特許文献３】米国特許第７，０８９，３８５号
【特許文献４】米国特許第６，９７８，３５４号
【非特許文献】
【０００９】
【非特許文献１】Ｓａｇｅ Ａ．Ｗｅｌｉ、Ｓｃｏｔｔ Ａ．Ｂｒａｎｄｔ、Ｅｔｈａｎ Ｌ．Ｍｉｌｌｅｒ、Ｃａｒｌｏｓ Ｍａｌｔｚａｈｎ、「ＣＲＵＳＨ：Ｃｏｎｔｒｏｌｌｅｄ，Ｓｃａｌａｂｌｅ，Ｄｅｃｅｎｔｒａｌｉｚｅｄ Ｐｌａｃｅｍｅｎｔ ｏｆ Ｒｅｐｌｉｃａｔｅｄ Ｄａｔａ」、Ｐｒｏｃｅｅｄｉｎｇｓ ｏｆ ｔｈｅ ２００６ ＡＣＭ／ＩＥＥＥ Ｃｏｎｆｅｒｅｎｃｅ ｏｎ Ｓｕｐｅｒｃｏｍｐｕｔｉｎｇ、ｐ．３１、２００６
【非特許文献２】Ｌｅｓｌｉｅ Ｌａｍｐｏｒｔによる「Ｐａｘｏｓ Ｍａｄｅ Ｓｉｍｐｌｅ」、ＡＣＭ ＳＩＧＡＣＴ Ｎｅｗｓ（Ｄｉｓｔｒｉｂｕｔｅｄ Ｃｏｍｐｕｔｉｎｇ Ｃｏｌｕｍｎ）３２、４（通巻号数１２１、２００１年１２月）５１～５８
【非特許文献３】Ｔｕｓｈａｒ Ｄｅｅｐａｋ Ｃｈａｎｄｒａ、Ｒｏｂｅｒｔ Ｇｒｉｅｓｅｍｅｒ、およびＪｏｓｈｕａ Ｒｅｄｓｔｏｎｅによる「Ｐａｘｏｓ Ｍａｄｅ Ｌｉｖｅ－Ａｎ Ｅｎｇｉｎｅｅｒｉｎｇ Ｐｅｒｓｐｅｃｔｉｖｅ（２００６ Ｉｎｖｉｔｅｄ Ｔａｌｋ）」、Ｐｒｏｃｅｅｄｉｎｇｓ ｏｆ ｔｈｅ ２６ｔｈ Ａｎｎｕａｌ ＡＣＭ Ｓｙｍｐｏｓｉｕｍ ｏｎ Ｐｒｉｎｃｉｐｌｅｓ ｏｆ Ｄｉｓｔｒｉｂｕｔｅｄ Ｃｏｍｐｕｔｉｎｇ、ＡＣＭ ｐｒｅｓｓ（２００７）
【発明の概要】
【発明が解決しようとする課題】
【００１０】
データの冗長性に対してポリシーの変更が行われる場合、データの移動は、集中的に管理され、中央管理ノードから実行される。Ｐａｒａｓｃａｌｅ Ｉｎｃ．のシステムは、新しいデータの位置として決定される位置が冗長性の変更を満たすために既存の記憶位置からデータを「引き出す」ことを求められるマイグレーションの中央管理を行っていた。

【課題を解決するための手段】
【００１１】
本発明の実施形態は、概して、前記コンピュータ・システムを制御するコンピュータ・システムおよび対応する方法に関し、より詳細には、シェアード・ナッシング分散型ストレージ・システムにおけるデータ管理およびマイグレーションのための技術に関する。下で説明される以下の実施形態は、例示的であり、本発明が本発明の精神および範囲内に入るその他の修正、代替的な構造、および均等物によって実施される可能性があることと、さらに、本発明の１つまたは複数の実施形態が一緒に組み合わされる可能性があることとを当業者が理解するような性質を持つ。
【００１２】
一実施形態においては、シェアード・ナッシング分散型ストレージ・システムにおいてデータの完全性および冗長性を保ち、データの完全性および冗長性をリカバリする方法が提供される。シェアード・ナッシング分散型ストレージ・システムにおいて、ストレージ・システムは、ストレージ・システム（本明細書においてはクラスタと呼ばれる）を形成するために相互に接続される複数の独立したシステム（本明細書においてはノードと呼ばれる）を含む。クラスタの各ノードは、ファイルの名前空間、メタデータ、および位置情報の管理を扱う。さらに、各ノードは、クラスタ内のその他ノードと独立しているストレージ・エリアを備え、クラスタの１つまたは複数のファイル・システムに記憶されるファイルへの物理的な記憶およびアクセスの責任を負う。ローカルであるかまたはストレージ・エリア・ネットワーク（ＳＡＮ）などを通じて取り付けられるかに関わらず、各ストレージ・ノードは１つまたは複数の永続的なストレージ・デバイスを備えており、通常の状況で、各ノードは、そのノード自体のストレージ・デバイスにのみアクセスすることができ、概して、ノードは、その他のノードのストレージ・デバイスにアクセスすることができず、したがって、システム全体のストレージが、システムのノードの間に分散されるので、このクラスタ・アーキテクチャは、シェアード・ナッシング分散型ストレージ・システムと呼ばれる。
【００１３】
データは、ストレージのステータスおよびクラスタ全体の活動の情報を考慮に入れるアルゴリズムを用いて、または何らかのその他のマッピング機能を用いて、クラスタ内のストレージ・デバイスに分散される可能性がある。例示的な実施形態においては、トポロジーおよびクラスタ内のストレージの負荷要因を用いて、ならびにデータに関連する一意の「キー」を用いてデータが配置される位置を計算してクラスタ内のデータの配置を決定する擬似ランダムなコンシステント・ハッシュ（ＣＲＵＳＨアルゴリズム）が使用される。例示的な実施形態において、データは、別個の有限なサイズのコンテナ（本明細書においては「チャンク」と呼ばれる）にマッピングされる。これらのチャンクは、チャンクまたはチャンクのグループの何らかの一意の属性にＣＲＵＳＨアルゴリズムを適用することによって配置される。
【００１４】
ノードおよびストレージの障害に対する回復力を提供するために、データは、冗長的に記憶される。データまたはデータの消失訂正符号化（ｅｒａｓｕｒｅ－ｃｏｄｅ）されたフラグメントのコピーが、ノードまたはストレージ・デバイスのサブセットが障害にあうときにデータがアクセス可能およびリカバリ可能なままであることができるような方法でクラスタ内の複数の位置に分散される。データのリカバリ可能な冗長な配置のためのさまざまな技術が、存在する。例示的な実施形態は、ＣＲＵＳＨアルゴリズムの特徴を利用し、ＣＲＵＳＨアルゴリズムは、クラスタ中に擬似ランダムに分散された１組の位置を計算し、同じストレージ・デバイスまたはノードへのデータの複数のコピーの配置を避けるように制約される。オブジェクト毎にオブジェクトの配置およびオブジェクトのメタデータを追跡することは、クラスタに大きな処理負荷をかける。たとえば、数百万のオブジェクトを有するストレージ・システムは、オブジェクト毎に配置を追跡するとき、過度の負荷に晒される。例示的な実施形態において、チャンクは、より大きな論理的なグループ（本

明細書においては「整合性グループ（ｃｏｎｓｉｓｔｅｎｃｙ　ｇｒｏｕｐ）」または「ＣＧ」と呼ばれる）に集約され、ＣＧ内のすべてのチャンクの配置のための位置が、同じ一意の属性を用いて計算される。例示的な実施形態において、ＣＧは、ＣＧの「名前」にＣＲＵＳＨアルゴリズムを適用することによって配置される。ＣＧは、データを複数のオブジェクトの集合体として追跡することによって上述の問題に対処し、集約されたデータがクラスタ内に冗長的に記憶されるという付け加えられた利点をさらに提供する。その上、整合性グループは、計算の量と、データを記憶し、取り出すときにシステムが追跡しなければならないオブジェクト毎のメタデータの量とを削減する利点を有する。たとえば、メタデータは、ファイル名、ＩＤ、パーミッション、サイズ、種類、データが生成されたときに関する情報（生成時間）、そのイベント番号の後にデータが書き込まれたイベント番号、そのイベント番号の後にデータの状態を示すために使用されるフラグが設定されるイベント番号、データのアクセス時間などのファイル属性に関する１つまたは複数の情報を含む可能性がある。

【００１５】

一部のその他の実施形態においては、ストレージ・デバイスが、各ノードのローカルに取り付けられ、クラスタ内のその他のノードによってアクセスされ得ない。さらに、データ処理は、各ノードがローカルで利用可能であり、クラスタ内のその他ノードによってアクセスされ得ない。そのようなストレージ・システムに記憶されるデータは、ノードのうちの１つもしくは複数の障害またはノードのうちの１つのローカルに取り付けられたストレージの障害の場合にデータの可用性および回復力を提供するためにノードの間で複製される。クラスタ内のいずれかの場所に記憶されるデータは、ノード間通信接続を介してクラスタ内の他のいずれかの場所からアクセス可能であり、データの位置およびデータ・アクセス機構は、クラスタに記憶されるデータを生成し、修正し、読み取るクライアントには見えない。

【００１６】

一部の実施形態において、各ノードのローカル・ストレージ・エリアは、インテリジェントなオブジェクト・ストレージ（本明細書においてはオブジェクトベース・ストレージ・デバイス（Ｏｂｊｅｃｔ－ｂａｓｅｄ　Ｓｔｏｒａｇｅ　Ｄｅｖｉｃｅ）またはＯＳＤと呼ばれるが、ＳＮＩＡおよびＴ１０　ＯＳＤ規格に合致しない）として構成される。ＯＳＤは、ディスク・ストレージと同様であるが、記憶空間のより抽象化されたビュー（ｖｉｅｗ）を提供する。データの決まったサイズのブロックとして読み取りおよび書き込みを処理する代わりに、ＯＳＤは、データをオブジェクト・データ・コンテナへと編成する。各オブジェクトは、データと、オブジェクトを記述する属性情報などのメタデータとの両方を有する。

【００１７】

一部の実施形態において、ノードまたはディスクの障害などの出来事がクラスタ内で起こるとき、システムは、影響を与えられたデータ・アイテムを修復しなければならない。たとえば、リカバリの場合、リカバリ・プロセスにかけられる各ノードが、そのノード自体のリカバリの管理を課せられる。リカバリは、データ修正オペレーションの信頼性の高い追跡と、ストレージ・クラスタおよびストレージ・クラスタ内のデータの状態をすべてのノードに知らせることと、リカバリオペレーションがローカルのデータに対して実行されなければならないかどうかおよびそのようなオペレーションが何を引き起こすかを独立して判定することとによって実現される。一部のその他の実施形態において、ディスクまたはノードが追加または削除されたためにクラスタの構成が変わる場合、データは、新しい記憶容量を利用するためまたは記憶容量が削減されることに適応するために再配分または再配置される必要がある可能性がある。さらにその他の実施形態においては、例えば、記憶されなければならないデータのコピーの数を増やすことによって冗長性のポリシーが変わる場合、新しいデータのコピーが生成されなければならない。そのような出来事は、本明細書においては「リカバリ」と呼ばれるカテゴリの下にグループ化される。本発明の１つの目的は、クラスタ内のさらなる障害がデータの損失をもたらす可能性があるので、

速く、効率的で、信頼性が高いリカバリを提供することであり、データの適時の効率的なリカバリを実行することによって、さらなる障害が原因であるデータの損失の可能性が、大幅に小さくされ得る。

【００１８】

その他の実施形態においては、データ・オブジェクトの状態が、進行中の書き込みおよび欠けているコピーであるデータ・オブジェクトへの書き込み、「良好な」ノードが「壊れている」ノードに書き込む「プッシュ」の手法の代わりに、影響を与えられるノードからの「プル」の手法を用いることによる、同期的に複製されるデータへの書き込みの結果として生じるＩ／Ｏエラーからのリカバリ、グローバルな状態の依存性の連鎖を生み出すことのない、書き込み側の障害によって引き起こされた同期的に複製されるデータへの部分的な書き込みからのリカバリ、ストレージおよび／またはノードが追加または削除される場合のデータのマイグレーション、ならびにデータの冗長性のポリシーが変わるときの同期的に複製されるデータの冗長なコピーの追加および／または削除によって高い信頼性で追跡される。

【００１９】

さらにその他の実施形態においては、ノードの間でイベントおよび処理を調整するために、分散型の合意アルゴリズムがクラスタで実施される。合意アルゴリズムの例は、当技術分野で知られているＰａｘｏｓファミリーのアルゴリズムである。Ｐａｘｏｓアルゴリズムにおいては、ノードのうちのいずれかが、「リーダー」として働き、システム内のあらゆるノードによって実行するためのコマンドを提案しようとする可能性がある。あらゆるそのような提案は、提案をより簡単に追跡するための対応する提案番号とともに送信される可能性がある。そのような提案番号は、ノードが実行するためにコマンドについて合意しようと試みている特定のステップといかなる関係も持つ必要がない。最初に、リーダーが、リーダーが提出しようと意図する提案に関する提案番号を示唆する可能性がある。そして、残りのデバイスのそれぞれは、それらのデバイスが賛成する最後の提案の指示、またはそれらのデバイスがいかなる提案にも賛成しないという指示によって提案番号のリーダーの示唆に応答する可能性がある。さまざまな応答を通じて、リーダーは、デバイスによって賛成されたいかなるその他の提案も知らない場合、以前のメッセージで示唆された提案番号を用いて、所与のクライアントのコマンドがデバイスによって実行されることを提案する可能性がある。各デバイスは、その段階で、アクションに賛成すべきかまたはそのアクションを拒否すべきかを判定する。デバイスは、そのデバイスがより大きな提案番号の別のリーダーの示唆に応答した場合にだけ、アクションを拒否するべきである。最小限の数として知られる十分な数のデバイスが提案に賛成する場合、提案されるアクションは、合意されたと言われ、各デバイスが、アクションを実行し、結果を送信することができる。そのようにして、Ｐａｘｏｓアルゴリズムは、デバイスのそれぞれが同じ順序でアクションを実行することを可能にし、ノードのすべての間で同じ状態を維持する。

【００２０】

概して、Ｐａｘｏｓアルゴリズムは、上述のように、リーダーがデバイスによって賛否の意思表示をされた前の提案を知ることを可能にする最初のフェーズと、リーダーが実行するためのクライアントのコマンドを提案することができる第２のフェーズとを有する２つのフェーズにあると考えられる可能性がある。リーダーは、前の提案を知ると、第１のフェーズを繰り返す必要がない。その代わりに、リーダーは、第２のフェーズを継続して繰り返し、複数のステップで分散型コンピューティング・システムによって実行され得る一連のクライアントのコマンドを提案する可能性がある。そのようにして、各ステップに関して分散型コンピューティング・システムによって実行されるそれぞれのクライアントのコマンドは、Ｐａｘｏｓアルゴリズムの１つのインスタンスと考えられ得るが、リーダーは、次のステップに関する別のクライアントのコマンドを提案する前に所与のステップのための提案されたクライアントのコマンドに対してデバイスが賛否の意思表示をするのを待つ必要がない。

【００２１】

実施形態の一部においては、クラスタ内の各ノードが適切な順番でデータを処理していることと、各ノードが同じ状態を維持することとを保証するためにＰａｘｏｓアルゴリズムが使用される。このようにして、Ｐａｘｏｓアルゴリズムに従う個々のノードは障害にあう可能性があるが、障害の間にクラスタ内でいかなるオペレーションも失われない。Ｐａｘｏｓアルゴリズムを実施する実施形態においては、個々のノードのいずれかが、リーダーとして働き、クラスタ内のあらゆるその他のノードに実行するためのオペレーションを提案しようとする可能性がある。あらゆるノードが同じ順序で同じコマンドを実行することを必要とすることによって、Ｐａｘｏｓアルゴリズムは、クラスタのノードの間の同期化を実現する。加えて、クラスタ内のノードが障害にあうかまたはそうではなくクラッシュする場合、クラスタは、Ｐａｘｏｓのメッセージを用いて知らされる。参照により本明細書に援用される、Ｌｅｓｌｉｅ Ｌａｍｐｏｒｔによる「Ｐａｘｏｓ Ｍａｄｅ Ｓｉｍｐｌｅ」、ＡＣＭ ＳＩＧＡＣＴ Ｎｅｗｓ（Ｄｉｓｔｒｉｂｕｔｅｄ Ｃｏｍｐｕｔｉｎｇ Ｃｏｌｕｍｎ）３２、４（通巻号数１２１、２００１年１２月）５１～５８を参照されたい。参照により本明細書に援用される、Ｔｕｓｈａｒ Ｄｅｅｐａｋ Ｃｈａｎｄｒａ、Ｒｏｂｅｒｔ Ｇｒｉｅｓｅｍｅｒ、およびＪｏｓｈｕａ Ｒｅｄｓｔｏｎｅによる「Ｐａｘｏｓ Ｍａｄｅ Ｌｉｖｅ－Ａｎ Ｅｎｇｉｎｅｅｒｉｎｇ Ｐｅｒｓｐｅｃｔｉｖｅ（２００６ Ｉｎｖｉｔｅｄ Ｔａｌｋ）」、Ｐｒｏｃｅｅｄｉｎｇｓ ｏｆ ｔｈｅ ２６ｔｈ Ａｎｎｕａｌ ＡＣＭ Ｓｙｍｐｏｓｉｕｍ ｏｎ Ｐｒｉｎｃｉｐｌｅｓ ｏｆ Ｄｉｓｔｒｉｂｕｔｅｄ Ｃｏｍｐｕｔｉｎｇ、ＡＣＭ ｐｒｅｓｓ（２００７）も参照されたい。

【図面の簡単な説明】

【００２２】

【図１】本発明の一実施形態によるクラスタ化されたシェアード・ナッシング・ストレージ・システムの図である。

【図２】本発明の一実施形態によるノードのブロック図である。

【図３】本発明の一実施形態による整合性グループと、チャンクと、スライスとの間の関係を示すブロック図である。

【図４】本発明の一実施形態によるシェアード・ナッシング・ストレージ・システム上のデータ・オブジェクトの配置を決定するためのストレージ構成マップの例を示す図である。

【図５】本発明の一実施形態によるＣＲＵＳＨによって生成されるディスク・リストの例を示す図である。

【図６】本発明の一実施形態によるクラスタのすべてのノードの状態情報の例を示す図である。

【図７】本発明の一実施形態によるマップ変更プロセスに関する高レベルの流れ図である。

【図８】本発明の一実施形態による各ノードが記憶するイベント・リストの例を示す図である。

【図９】本発明の一実施形態による図７のマップ変更プロセスのフェーズ１のプロセスの流れ図である。

【図１０】本発明の一実施形態によるフェーズ１中にマイグレーション要求を受信するノードのプロセスの流れ図である。

【図１１】本発明の一実施形態による図７のマップ変更プロセスのフェーズ２のプロセスの流れ図である。

【図１２】本発明の一実施形態によるフラグ・テーブルの例を示す図である。

【図１３】本発明の一実施形態によるチャンク・スタブ（ｓｔｕｂ）管理情報の例を示す図である。

【図１４】本発明の一実施形態によるノードのメモリに記憶されるビットマップの例を示す図である。

【図１５】本発明の一実施形態によるマイグレーション中のＷＲＩＴＥ要求のプロセスの

流れ図である。
【図１６】本発明の一実施形態によるマイグレーション中のＷＲＩＴＥ要求のプロセスの流れ図である。
【図１７】本発明の一実施形態による、通常の条件の間にＩ／Ｏオペレーションを受け取るノードのプロセスの流れ図である。
【図１８】本発明の一実施形態による、マイグレーション中のＩ／Ｏオペレーションが完了した後にデータ・チャンクをクリーンに設定するプロセスの流れ図である。
【図１９】本発明の一実施形態による、障害の後、オンラインに復帰するときのターゲット・ノードのプロセスの流れ図である。
【図２０】本発明の一実施形態によるマイグレーション中のソース・ノードの障害のリカバリのプロセスの流れ図である。
【図２１】本発明の一実施形態による図７のマップ変更プロセスのフェーズ３のプロセスの流れ図である。
【図２２】本発明の一実施形態による整合性グループに関するポリシー・テーブルを示す図である。
【図２３】本発明の一実施形態による冗長性のレベルのポリシーの変更のプロセスの流れ図である。
【図２４】本発明の一実施形態による、クラスタに再び加わった後の部分的に書き込まれたデータのノードのリカバリのプロセスの流れ図である。
【図２５】本発明の一実施形態による、クラスタに再び加わった後の部分的に書き込まれたデータのノードのリカバリのプロセスのメッセージ・シーケンスを示す図である。
【図２６】本発明の一実施形態による（Ｉ／Ｏ）オペレーションの受け取りを扱うためのホストとノードとの間の通信フローを示す図である。
【図２７】本発明の一実施形態によるＩ／Ｏオペレーションを扱うためのホストとノードとの間の代替的な通信フローを示す図である。
【発明を実施するための形態】
【００２３】
本明細書において検討される実施形態は、本発明の１つまたは複数の例を示す。本発明のこれらの実施形態が図を参照して説明される説明されるとき、説明される方法および／または特定の構造のさまざまな修正または適応が、当業者に明らかになる可能性がある。本発明の教示に依拠し、これらの教示が当技術分野を進歩させたすべてのそのような修正、適応、または変形は、本発明の範囲内にあると考えられる。したがって、本発明は本明細書において示される実施形態のみにまったく限定されないことが理解されるので、この説明および図面は限定的な意味にとられるべきでない。別途示されない限り、同様の部分および方法のステップは、同様の参照番号によって参照される。
【００２４】
導入
本発明は、ネットワーク接続ストレージ（ＮＡＳ）のアーキテクチャの基盤として実装される可能性があり、分散型データ・プラットフォームで使用される可能性がある。概して、図１に示されるように、ネットワーク３０に接続されたノード１０上のファイル・システムにおいてデータ・オブジェクトを分散させ、記憶するシェアード・ナッシング分散型ストレージ・システムが、提供される。
【００２５】
図１に示されるように、クラスタの各ノードは、ネットワーク３０を通じて接続される。１つまたは複数のクライアントまたはホスト２０が、シェアード・ナッシング・ストレージ・システムに結合される。ホスト２０は、概してコンピュータであるが、ネットワークを介して接続された別のストレージ・システムまたはサーバである可能性もある。ホスト２０は、たとえば、Ｉ／Ｏオペレーションをストレージ・システムに送る。Ｉ／Ｏオペレーションは、例えば、書き込みオペレーション（ＷＲＩＴＥ）、読み取りオペレーション（ＲＥＡＤ）、削除オペレーション（ＲＥＭＯＶＥ）、またはトランケート（ＴＲＵＣ

ＡＴＥ）オペレーションである。チャンク・データを修正するＩ／Ｏオペレーション（そのようなオペレーションはＷＲＩＴＥである）は、そのチャンクを記憶するノードにグローバル一意キーを供給する。図２に示されるように、各ノード１０は、そのノード１０が記憶する各チャンクに関するＩ／Ｏキーを保有するＩ／Ｏキー・リスト９５をメモリ５０に保有する。加えて、Ｉ／Ｏオペレーションがダウンしているノード１０に対して要求される場合、ノード１０は、クラスタ内のその他のノードにそれらのその他のノードの最新のＩ／Ｏキー・リストに関して問い合わせることによって、オンラインに復帰するときにＩ／Ｏオペレーションを受け取る。アクティブなノード１０またはディスク６５は、本明細書においてオンラインと呼ばれ、一方、非アクティブなノード１０またはディスクは、本明細書においてオフラインと呼ばれる。ノード１０のいずれかが、必要に応じて手動でオフラインもしくはオンラインにされる可能性があり、またはそれらのノードは、ソフトウェアもしくはハードウェアの障害によってオフラインになる可能性がある。

【００２６】

典型的には、ホスト２０は、パーソナル・コンピュータ（ＰＣ）、ワークステーション、ラップトップ、携帯情報端末（ＰＤＡ）、サーバ、メインフレームなどのコンピュータ・システムである。ホスト２０は、ＮＦＳ、ＣＩＦＳ、ＨＴＴＰ、ＦＴＰなどのファイル・アクセス・プロトコルを用いて遠隔のファイルおよびファイル・システムにアクセスするように構成される。

【００２７】

ノード１０は、ＰＣ、ワークステーション、サーバ、メインフレームなどである可能性があり、またはディスク・アレイもしくはストレージ・アプライアンスなどの、本明細書において説明されるシステムを実装するのに十分な組み込まれたコンピューティング能力を有するストレージ・デバイスである可能性がある。ノードは、ローカル・ファイル・システム、ネットワーク接続ストレージ（ＮＡＳ）、ストレージ・エリア・ネットワーク（ＳＡＮ）、データベースなどの上のファイル・システム内のファイルに関連する情報を記憶し得る。さらに、ノードは、ファイル・システムにファイルを記憶するために任意のハードウェアおよび／またはソフトウェア要素から構成される可能性があり、ＮＴＦＳ、ＥＸＴ、ＸＦＳ、ＧＦＳなどの、ファイルを記憶するための１つまたは複数のファイル・システムを実装する可能性がある。

【００２８】

クラスタにおいて、ファイル・システム自体は、特定のノードに物理的に結びつけられないので、１つもしくは複数のノードまたはクラスタ全体に広がる可能性がある。データは、ノード１０に記憶され、ノード間通信接続を介してクラスタ内のその他のノード１０によってアクセスされ得る。ネットワーク・インターフェース４０および基礎を成すネットワーク通信の技術的な態様は、本発明の範囲内になく、したがって、それらの態様は、詳細に説明されない。クラスタ内のノード１０の間のデータのマイグレーション中、ホスト２０からのＩ／Ｏ処理は、妨げられない。

【００２９】

ストレージ・システム内のノード１０に記憶されたデータは、ノード１０のうちの１つもしくは複数の障害またはノード１０に取り付けられたディスク６５の障害の場合にデータの可用性および回復力を提供するためにノード１０の間で複製される。すべてのノード１０は、その他のノード１０のステータスと、そのノードのディスク６５およびその他のノード１０のディスク６５上のデータの状態とを状態情報１００を介して知らされる。消失訂正符号化方法が、ストレージ・システムにおいて使用される可能性がある。たとえば、リード・ソロモン方式においては、データのブロックに対する計算が、消失訂正符号化されたブロックを生成する。数多くのその他の好適な冗長なまたは消失訂正符号化されたストレージ方式が、当業者に明らかであろう。各ノード１０は、処理能力と記憶能力との両方を有する可能性がある。クラスタ内のノード１０は、ローカル・ファイル・システム、ＮＡＳ、ＳＡＮ、データベースなどの上のファイル・システム内のファイルに関連する情報を記憶し得る。クラスタに記憶されたファイルを提供するための容量および帯域幅を

スケーリングするために、ノード１０が追加されるかまたはクラスタから削除される可能性があり、ノード１０のディスク６５が追加または削除される可能性がある。

【００３０】

図４のように、ノード１０のクラスタにわたるストレージ・システム上のデータの分散された記憶に関するレイアウトを決定するために、ストレージ構成マップ７０、７１が、すべてのノード１０によって利用される。たとえば、ノード１０またはディスク６５が追加されるとき、クラスタにおけるノードまたはディスクの追加に基づく新しいマップ７１をコミットするために、マップ変更提案プロセスが開示される。古いおよび新しいマップ７０、７１に基づいて、各ノード１０は、データ・オブジェクトの配置または位置を知る。古いマップ７０と新しいマップ７１との両方は、ノード１０におけるＩ／Ｏ処理およびレイアウトの参照のために利用され得る。

【００３１】

図４に示されるマップ７０、７１は、整合性グループＩＤ（ＣＧＩＤ７７）に基づいてＣＲＵＳＨアルゴリズムがデータのチャンクの配置を決定するノード１０およびディスク６５またはボリュームの集合である。ストレージ・システムは、図２に示された論理デバイス（ＬＤＥＶ）６６を用いて構成された１つまたは複数の論理ストレージ・エリアまたは論理ボリュームを提供する。ストレージ・デバイス６７は、論理ボリュームとＬＤＥＶ６６との間の対応（関係）を管理する。ディスク６５または論理ドライブ上のボリュームが、マップ７０、７１上のＣＧのチャンクに関する位置として提供され得る。したがって、マップ７０、７１における所与のＣＧのチャンクの配置または位置は、ディスク６５の代わりにボリュームまたは論理ボリュームを用いて定義される可能性もある。

【００３２】

各ＣＧは、ＣＧを特定する一意識別子、ＣＧＩＤ７７を有する。ＣＧが生成されると、生成するノード１０が、生成されたＣＧに関するＣＧＩＤ７７を生成する。マップ７０、７１は、データ・ストレージ・システム上のデータ・オブジェクトの分散された記憶に関するレイアウトを決定する際に利用される。ＣＲＵＳＨは、ノード１０およびディスク６５のＣＧおよびそのＣＧがマップ７０、７１に基づいてクラスタ内に含むストレージ・オブジェクトの配置を選択する。ＣＧは、ＣＧＩＤ７７を共有し、したがって、ノード１０およびディスク６５上の同じ配置を有するデータ・チャンクからなるファイルなどのデータ・オブジェクトのグループである。データ・オブジェクトは、その他のデータ・オブジェクトと別々に識別可能であり、ストレージ・システムに転送可能である有限長のデータであり、テキスト・ファイル、画像ファイル、またはプログラム・ファイルなどである可能性がある。ＣＧおよびそれらのＣＧの対応する複製が、クラスタ内の複数のノード１０にわたって記憶される。図３は、ＣＧと、チャンクと、スライスとの間の関係の図を示す。図３に示されるように、ＣＧは、複数のチャンクから成り、各チャンクは、限られた最大サイズ（たとえば、６４ＭＢ）のデータ領域である。チャンクは、信頼性のためにノード１０中に複数複製されて記憶されるかまたは消失訂正符号化される。各チャンクは、スライスと呼ばれるデータの領域にさらに分割され得る。各スライスは、決まったサイズである可能性がある。

【００３３】

ＣＲＵＳＨアルゴリズムは、マップ７０、７１からデータが記憶されるディスク６５のリストを生成するために使用され得る。任意のノード１０が、ＣＲＵＳＨアルゴリズムを実行することができる。ノード１０は、ＣＧＩＤ７７およびいくつかの必要とされる一意の配置位置を含む情報をＣＲＵＳＨアルゴリズムに渡す。ＣＲＵＳＨは、例えば、ディスク６５の明示的に順序付けられたリストによって応答する。ＣＲＵＳＨは、同じマップ７０、７１内の同じＣＧＩＤ７７に関していつもディスク６５の同じシーケンスが生成されることを保証する。つまり、ＣＲＵＳＨは、３つのディスク｛Ａ，Ｂ，Ｃ｝またはディスク・リスト７６｛Ａ，Ｂ，Ｃ，Ｄ｝を生じる４つのディスクのディスク・リスト７６を返す可能性がある。リスト７６のディスク６５は、アクセス可能である可能性があり、またはソフトウェアもしくはハードウェアの問題、たとえば、特定のディスク６５が物理的に

取り付けられているノード１０のクラッシュが原因でアクセス不可能である可能性がある。クラッシュによって生成されるリスト７６の第１の使用可能なディスク６５は、「ファースト・アップ（ｆｉｒｓｔ　ｕｐ）」ディスク（ボリューム）と呼ばれる可能性がある。

【００３４】

図４は、古いマップ７０および新しいマップ７１の例を示す。古いマップ７０は、以前のマップ７０のバージョンに関するＣＲＵＳＨマップを含む１組のノード１０およびディスク６５（ボリューム）である。一方、新しいマップ７１は、提案される新しいマップ７１のバージョンに関するＣＲＵＳＨマップを含む１組のノード１０およびボリューム６５として定義され得る。古いマップ７０は、新しいマップ７１がノード１０によってコミットされるまでデータ・オブジェクトのレイアウト構成として使用される。言い換えると、マップ７０は、クラスタ内のデータの配置のために使用される現在のマップに対応し、一方、マップ７１は、クラスタ内のデータの新しい提案される配置である。図４のキーで示されるように、ノード１０は、ノード識別子、たとえば、番号１、２、３、４を用いて示され、ディスク６５は、大文字Ａ、Ｂ、Ｃ、Ｄ、Ｅを用いて示される。各ディスク６５は、対（ノードの番号，ディスクの文字）によって特定される。たとえば、１Ａは、ノード１のディスクＡを表す。

【００３５】

図５は、ＣＧＩＤ７７「１」に関する古いマップ７０および新しいマップ７１を用いてＣＲＵＳＨによって生成された例示的なディスク・リスト７６を示す。図４および５に示されるように、ＣＧＩＤ７７「１」に関する古いマップ７０の下のディスク・リスト７６は、１Ａ、３Ｃ、および４Ｆを含む。一方、新しいマップ７１の下では、ＣＧＩＤ７７「１」は、ディスク２Ｂ、３Ｃ、および５Ａに記憶される。提案される（新しい）マップ７１の下で、ＣＧＩＤ７７「１」に関するＣＲＵＳＨによって生成されたディスク・リスト７６は、２Ｂ、３Ｃ、および５Ａを含む。

【００３６】

ノード１０は、ファイルシステム、ネットワーク接続ストレージ（ＮＡＳ）、ストレージ・エリア・ネットワーク（ＳＡＮ）、データベースなどの内のデータに関連するメタデータを記憶する。メタデータは、ファイル名、ＩＤ、パーミッション、サイズ、種類、データが生成されたときに関する情報（生成時間）、そのイベント番号の後にデータが書き込まれたイベント番号、そのイベント番号の後にデータの状態を示すために使用されるフラグが設定されるイベント番号、データのアクセス時間などのファイル属性を含む。図２に示されるように、各ノード１０は、オブジェクト・ストレージ・デバイス（ＯＳＤ）６０、ネットワーク・インターフェース４０、プロセッサ４５、（本明細書においてはＳＡと呼ばれる）管理ユーティリティ５５、メモリ５０、および複数のディスク６５を有する。ネットワーク・インターフェース４０は、ネットワーク上のホストおよびその他のノードと通信し、ホスト２０からＩ／Ｏ処理を受信し、そのＩ／Ｏ処理に応答する役割を担う。加えて、ＰａｘｏｓおよびＣＲＵＳＨアルゴリズム、マップ７０、７１、メタデータ、ならびにビットマップ７５が、メモリ５０において構成され、管理される。

【００３７】

各ノードのメモリは、ＣＲＵＳＨおよびＰａｘｏｓアルゴリズムを実行し、各ＣＧおよびそのＣＧのチャンクのメタデータに関連する情報、フラグ・テーブル１２０、Ｉ／Ｏキー・リスト９５、状態情報１００、ビットマップ７５、チャンク・スタブ管理情報１０５、イベント・リスト１２５、ならびにチャンク・コピー・リスト１１５を記憶し、管理するために使用される。

【００３８】

各ノード１０は、独立したストレージ・デバイス６７を有する。ストレージ・デバイス６７は、複数のストレージ・ドライブまたはディスク６５を含む。各ディスク６５は、たとえば、ＳＡＳ（シリアル・アタッチドＳＣＳＩ）、ＳＡＴＡ（シリアルＡＴＡ）、ＦＣ（ファイバ・チャネル）、ＰＡＴＡ（パラレルＡＴＡ）、およびＳＣＳＩなどの種類のハ

ード・ディスク・ドライブ、半導体ストレージ・デバイス（ＳＳＤ）など、またはファイバ・チャネル・ネットワーク上のストレージ・エリア・ネットワーク（ＳＡＮ）を介して遠隔でアクセスされるＳＣＳＩ論理ユニット（ＬＵＮ）もしくはＴＣＰ／ＩＰネットワーク上でアクセスされるｉＳＣＳＩ　ＬＵＮとして与えられるストレージ・デバイス、またはその他の永続的なストレージ・デバイスおよびメカニズムである。

【００３９】

各ノード１０は、Ｉ／Ｏを実行するためにホスト２０が通信するレイヤであるストレージ・アクセス・レイヤ（ＳＡ５５）を提供する管理ユーティリティを有する。ＳＡは、ストレージのミラーリングおよび整合性を扱う。ＳＡは、分散型Ｉ／Ｏシステムの「イニシエータ」部分である。加えて、ＳＡ５５は、各ノード１０のＯＳＤからのデータを要求することができる。各ノード１０のＯＳＤ６０は、ノード１０上のローカルＩ／Ｏを扱い、ストレージ・デバイス６７に対するオペレーションを実行する。ＯＳＤ６０は、たとえば、ストレージ・デバイス６７からデータを読み取るかまたはストレージ・デバイス６７にデータを書き込むときにストレージ・デバイス６７との通信を実行する。ＯＳＤ６０は、ＣＧに関して、そのＯＳＤ６０がＣＧのチャンクをどのディスク６５に記憶するかを決定することができる。マップ７０、７１が、各ノード１０のＳＡ５５において生成される。また、あらゆるノード１０は、マイグレーションが現在実行されているかどうかを示すマップ・マイグレーション状態を知っている。ＣＧは、そのＣＧ自体がリカバリされている途中であるかどうか、またはＣＧがマイグレーション状態であるかどうかをシステムに知らせる状態指示によって印を付けられる。加えて、ＳＡ５５は、マップ７０、７１を管理する。ＳＡ５５のレイヤは、マップ７０、７１が古いマップ７０であるかまたは新しいマップ（提案されるマップ）７１であるかに関わらず、マップ７０、７１を使用する。

【００４０】

各ノード１０は、その他のノード１０によって受信され、ノード１０のクラスタの個々のビューを構築し、維持するために使用される状態情報１００をネットワーク上で提供する。図６は、クラスタ１０のノードがアクティブであるかどうかおよびノードのディスクがデータの記憶のために利用可能であるかどうかについての情報を含む、ノードおよびノードのディスクの状態（ステータス）情報１００の例である。加えて、状態情報１００は、それぞれのローカルに取り付けられたストレージ・デバイス６７内の個々のディスク６５の可用性を含む。当業者は、状態情報１００が異なる属性を含む可能性があり、または更新ステータスもしくはエラー・ステータスなど、異なるように定義される可能性があることを認識する。図６に示される例においては、ノード１がオンラインであり、ディスクＡ～Ｃがオンラインであり、一方、ノード２のディスクＢ、Ｄ、およびＥはオフラインである。状態情報１００は、動的であり、各ノードのメモリに記憶される。

【００４１】

ノード１０のディスク６５または新しいノード１０などの追加的な記憶容量がクラスタに追加されるとき、ディスク６５の削除など、記憶容量が削減されるか、もしくはノード１０が削除される（所与のノード１０に取り付けられたすべてのディスク６５をオフラインにする）とき、または重い負荷がかかっているノード１０もしくはディスク６５から十分に利用されていないノードに割り当てを振り向けるようにディスクの重み係数が変更されるとき、マップ変更が行われなければならず、したがって、クラスタに関するＩ／Ｏ処理がクラスタ構成に対する変更を反映するために新しいマップ７１に従って実行される。マップ変更は、一部のＣＧが新しい位置（ノード１０／ディスク６５）にマッピングされるようにし、それらのＣＧ（すなわち、ＣＧのすべてのチャンク）のコピーが新しいマップ７１がコミットされる前にそれらのＣＧの新しい位置に再配置されることを要求する。マップ変更の下でのマイグレーション中、ホスト２０からのＩ／Ｏ処理は、妨げられない。

【００４２】

マップ変更プロセス

図１のストレージ・システムの分散型シェアード・ナッシング・アーキテクチャにおい

て、各ノード１０は、集中型の調整または監督なしにそのノード１０自体のデータのリカバリ・プロセスを実行する役割を担う。さらに、Ｐａｘｏｓを用いるマップ変更イベント配信アルゴリズムを実施することによって、各リカバリ・イベントが、各ノード１０に順に配信され、各ノード１０が、起こったすべてのイベントを受信する。

【００４３】

図８は、各ノード１０がメモリ５０において管理するイベント・リスト１２５の例を示す。各ノード１０のＳＡ５５がＰａｘｏｓによってイベントを受信するとき、そのイベントは、時間順にイベント・リスト１２５に記憶される。Ｐａｘｏｓは、イベントが処理されるために届けられる順序を管理する。ノード１０は、オンラインに復帰するとき、オフラインであった間に逃した可能性があるイベントおよびタスクのリストを取得するためにクラスタ内のそのノード１０のピアに問い合わせる。ノードは、イベントを受信し、それらのイベントをイベント・リスト１２５に記憶する。図８に示されるように、ノード１は、ＭＡＰ＿ＣＨＡＮＧＥ＿ＰＲＯＰＯＳＡＬイベントを受信し、続いてＭＡＰ＿ＭＩＧＲＡＴＥイベントを受信し、その後の時間にＭＡＰ＿ＣＯＭＭＩＴイベントを受信した。したがって、ノード１０は、順番にイベントに対応するプロセスを行わなければならない。ノード１０は、現在のフェーズが完了するまで図７に示されるように後のフェーズに移ることができない。図７に示される３つのフェーズのそれぞれが、以下の通り詳細に説明される。

【００４４】

上述のように、たとえば、クラスタの記憶容量がクラスタへのノードの追加もしくは削除またはノードへのストレージ・デバイスの追加もしくは削除によって変更されるとき、ＣＲＵＳＨアルゴリズムが、クラスタ中のデータの配置に関する新しいマップを決定するために使用される。既存のマップ（たとえば、古いマップ）から提案されるマップ（たとえば、新しいマップ）への移行は、マップ変更プロセスと呼ばれる。マップ変更プロセス中、データが、新しいマップに合致するために１つまたは複数のノードに再配置される可能性がある。データが再配置される間、データが引き続きクラスタによってホスト２０に利用され得るようにされる、Ｉ／Ｏ処理が移行中に妨げられないことが重要である。マップ変更プロセスは、３つのフェーズで行われる。

【００４５】

図７は、マップ変更プロセスの抽象的な図を示す。概して、フェーズ１　１００は、マップ変更の提案からなる。マップ変更の提案が行われると、各ノード１０は、マップ変更の準備をし、提案に応答する。マップ変更は、ノードによって提案されるか、管理命令によってトリガされる。マップ変更の提案は、クラスタ内のオンラインであるすべてのノード１０にＰａｘｏｓに従って配信される。各ノード１０は、あらゆる継続中のプロセスを終わらせることなどによってそのノード１０自体をマップ変更のために準備する第１のステップを経なければならない。各ノード１０は、そのノード１０自体がフェーズ１に関するそのノード１０のそれぞれの処理を済ませたと考えるときに、マップ変更の提案を受け付けたことをＰａｘｏｓに従って報告する。マップ変更の提案元は、すべての「ダウンしていない（ｕｐ）」ノード（たとえば、オンラインであり、応答可能であるノード１０）がマップ変更の提案に応答し、マップ変更が受け付け可能であることを提案元ノードに示すまでフェーズ２に進むことができない。いずれかのノード１０が提案を拒絶する場合、提案は失敗する。提案がクラスタ内のすべてのダウンしていないノードによって受け付けられないことに応じて、マップ変更プロセスのフェーズ２　２００に入る前に別のマップの提案が生成され、受け付けられる必要がある。

【００４６】

別の実装においては、最大の許容される時間よりも長くダウンしているクラスタのノード１０が、構成から削除される可能性がある。時間の最大の量は、時間の予め決められた量である可能性がある。ノード１０の削除は、マップ変更をトリガし得る。この場合、削除は、マップ変更の提案元によりマップ変更を失敗させる理由であると解釈され得る。マップ変更プロセスのフェーズ２　２００において、マップ変更が行われるために必要とさ

れる処理を勧めるようにすべてのノード１０に命令するメッセージが、Ｐａｘｏｓに従って送信される。つまり、フェーズ２は、クラスタ内でチャンクのマイグレーションおよび書き込みの代理処理（ｐｒｏｘｙｉｎｇ）をトリガする。フェーズ２は、すべてのダウンしていないノード１０がそれらのノード１０が新しいマップを実施するために必要とされるそれらのノード１０のそれぞれのマイグレーション処理を完了したという報告をＰａｘｏｓに従って返すときに終了する。

【００４７】

フェーズ３　３００は、新しいマップ７１が実施されていることをすべてのノード１０に知らせることによってマップ変更を「コミットすること」からなる。これは、すべてのノード１０のＳＡ５５に古いマップ７０を使用することを止めさせ、新しいマップ７１を使用し始めさせる。第３のフェーズにおいて、各ノード１０に関するＯＳＤ６０は、新しいマップ７１の下でもはや記憶されない古いマップの下で記憶されたデータを削除する。

【００４８】

フェーズ１

フェーズ１が、図９（フェーズ１）に示される流れ図を参照して説明される。フェーズ１の初めに、あらゆるノード１０が、マップ変更の提案元ノードからＰａｘｏｓに従ってＭＡＰ＿ＣＨＡＮＧＥ＿ＰＲＯＰＯＳＡＬイベント１０１を受信する。これが、フェーズ１のメッセージである。図９において、提案されるマップは、説明を目的として、新しいマップ７１を指す。各ノード１０によってイベント１０１が受信されると、新しいマップが、ノードのメモリ５０に記憶されるあらゆるノード１０において生成される（ステップ１０５）。

【００４９】

ステップ１１０において、新しいマップ７１に基づいて、各ノード１０が、新しいマップ７１へのマップ変更によって影響を与えられるいずれかのＣＧをそのノードが記憶するかどうかを判定する。判定は、各ＣＧに関して行われる。ノードによって記憶されるＣＧは、古いマップ７０の下での位置と比較されるときに新しいマップ７１の下でのＣＧに関する位置の変化がある場合に、新しいマップ７１によって影響を与えられる。古いマップ７０と新しいマップ７１との間の比較は、各ＣＧに関するＣＧＩＤ７７を調べ、ＣＧＩＤ７７に対応する（古いマップ７０および新しいマップ７１に関する）ディスク・リスト７６を比較することによって行われる。各ノード１０は、新しいマップ７１によって影響を与えられる各ＣＧに関して図９のステップに従う。ノード１０は、新しいマップ７１の下で影響を与えられるＣＧを持っていないか、マップ変更によって影響を与えられるＣＧを持っているかのどちらかである。ノード１０が影響を与えられるいかなるＣＧも持たない場合、ノードは、ステップ１１５において、Ｐａｘｏｓに従ってそのノードの受け付けをポストする、つまり、マップ変更の提案を受け付けた。マップ変更がノード上で今やＣＧに影響を与える場合、処理は、ステップ１４０に続く。

【００５０】

ノード１０は、古いマップ７０でノード１０のローカル・ボリューム６５にコピーを記憶するが、新しいマップ７１の下ではそのコピーをローカルに記憶しない場合、新しいマップの提案のそのノード１０の受け付けをポストし、新しいマップ７１がコミットされるまでＩ／Ｏオペレーションを受け付け続ける。ノード１０は、古いマップ７０と新しいマップ７１との両方の下でそのノード１０のローカル・ボリューム６５にＣＧのコピーを有する場合、新しいマップ７１のそのノード１０の受け付けをポストする。また、ノード１０は、（提案される）新しいマップ７１が有効にされるまでＩ／Ｏ要求を受け付け続け、これは、後でフェーズ２および３において説明される。

【００５１】

フェーズ１中に、ノード１０は、そのノード１０が古いマッピング・シーケンス７０の第１の「ダウンしていない」ボリュームを含むかどうかをやはり判定する。ダウンしていないボリュームは、オンラインであり、応答可能であり、データを新しい位置にマイグレーションすることができるノード１０のボリューム６５と呼ばれる。ダウンしていないボ

リュームは、そこから新しい位置にデータがマイグレーションされるボリュームである可能性がある。ダウンしていない複数もボリュームが存在する可能性があるが、１つが、第１のダウンしていないボリュームとして指定される。ダウンしていないボリュームは、ＣＧＩＤ７７に基づいてＣＧに関してＣＲＵＳＨによって生成されるディスク・リスト７６の第１のディスク６５としてノードによって決定され得る。

【００５２】

ステップ１４０において、ノード１０が第１のダウンしていないボリュームを有するとノード１０が判定する場合、そのノード１０は、新しいマップ７１の下の整合性グループのその他のメンバーに、それらのその他のメンバーが新しいマップ７１の下で記憶されるＣＧを有することを知らせる役割を担うノード１０である。第１のダウンしていないボリュームを有するノードは、新しいマップ７１を用いてＣＧのチャンクに関する新しい位置を特定する。ノード１０は、ステップ１４５において、新しいマップの下のＣＧのメンバーであるノードに、それらのノードが第１のダウンしていないボリュームのノードからＣＧを受信すべきであることを示すＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴメッセージを送信する。フェーズ１の下では、ステップ１４５において第１のダウンしていないボリュームが新しいメンバーのノード１０に知らせない限り、新しいマップの下のＣＧを記憶すべきである新しいマップ７１の下のノードのメンバーが、それらのノードのメンバーがＣＧのチャンクに関する新しい位置であることを知るその他の方法が存在しない。ＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴは、マイグレーションされるべきであるＣＧＩＤ７７のリストからなる。加えて、チャンクに関するフラグ・テーブル１２０を含むＣＧに対応するメタデータが、ＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴとともに送信され得る。ＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴを受信するノードのＯＳＤ６０は、ノードが新しいマップの下の複数のＣＧを記憶すべきである場合、１つまたは複数のノード１０の複数のダウンしていないボリューム６５から複数のそのような要求を受信し得る。ＯＳＤ６０は、ノード１０がＣＧを記憶する位置のリストを生成すれば十分なので、どのノード１０が各要求を送信したかを追跡する必要はない。

【００５３】

図１０は、図９のステップ１４５においてＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴを受信するノード１０に関する処理を示す流れ図を示す。ステップ１６０においてＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴを受信するノードのＯＳＤ６０は、ステップ１６１において、ＣＧが既に存在し、アクティブであるかどうかを判定する。ＣＧが存在するかどうかを判定する際、ノードは、ＣＧが実際に存在するかどうか、またはＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして印を付けられたＣＧがＯＳＤ６０のストレージ・デバイス内に既に存在するかどうかを判定する。ＣＧは、そのＣＧに関連するエラーを持たず、ホストに書き込むことおよび／またはホストから読み取ることが現在可能である場合にアクティブである。ステップ１６１において、既に生成されたＣＧが存在しない、および／またはそのＣＧがエラーを有する場合、ノード１０は、ステップ１６５に進み、ステップ１６５は、新しいマップの下で記憶される各ＣＧに関するスタブおよびＣＧの経路の階層を生成することである。ステップ１６１における判定がはいである場合、このＣＧＩＤに関するＣＧのスタブがノードに既に存在するためノード１０がそのＣＧのスタブを生成する必要が確かにあるので、プロセスはステップ１７５に進む。

【００５４】

また、ＣＧの階層は、そのＣＧの階層が記憶しなければならないＣＧに関するＣＧＩＤ７７を含む。ＣＧのメタデータが生成され、ステップ１７０においてＯＳＤ６０によってＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして永続的に印を付けられる。ＣＧのスタブの階層が生成されると、ノードは、ステップ１７５において、そのノードにＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴを送信したノードにＡＣＫメッセージを送信することによってマイグレーション要求に肯定応答する。ＡＣＫメッセージは、第１のダウンしていないボリュームのノードによって受信され（ステップ１５０）、ノードは、そのノードがマイグレーション要求を送信したすべてのノード１０からＡＣＫメッセージを受信すると、ステ

ップ１５５において、Ｐａｘｏｓに従って提案に対する受け付けをポストすることによってマップ変更提案イベントに応答することができる。フェーズ１は、クラスタ内のすべてのノード１０がマップ変更の提案を受け付けたときに正常に完了する。たとえば、いずれかのノード１０がエラーのために提案を受け付けることができない場合、提案は失敗し、提案元はＭＡＰ＿ＣＨＡＮＧＥ＿ＣＡＮＣＥＬイベントを送出しなければならない。

【００５５】

マップ変更の取り消し

提案が失敗する場合、あらゆるノード１０が、Ｐａｘｏｓに従ってＭＡＰ＿ＣＨＡＮＧＥ＿ＣＡＮＣＥＬイベントを受信する。ＭＡＰ＿ＣＨＡＮＧＥ＿ＣＡＮＣＥＬイベントが起こると、マップ・マイグレーション状態がクリアされ、各ＯＳＤ６０が提案された新しいマップ７１に関して構築されたＣＧを有するかどうかをそのＯＳＤ６０が調べる。ノード１０がＭＡＰ＿ＣＨＡＮＧＥ＿ＣＡＮＣＥＬイベントを受信すると、構築された新しいマップ７１が無効化される。さらに、新しいマップ７１の下のＣＧを受信することを予想するＣＧの階層を生成したノード１０が、ＣＧの階層を削除する。ＭＡＰ＿ＣＨＡＮＧＥ＿ＣＡＮＣＥＬイベントがすべてのノード１０に送信され、ノード１０がクラッシュしたかまたはその他の理由でオフラインであるためにいずれかのノード１０がイベントを受信しない場合、イベントを受信しなかったノード１０は、クラスタのオンラインのノードによって保有されるイベント・リスト１２５に従って、そのノード１０がオンラインに復帰するときにメッセージを受信する。加えて、マップ変更が取り消されると、ＳＡ５５は、新しいマップ７１のメンバーにＩ／Ｏオペレーションの要求を送信することを止める。

【００５６】

フェーズ２

フェーズ２は、Ｐａｘｏｓに従ってクラスタのノード１０に送信されるマップ・マイグレーション指令で始まる。図１１のステップ２０１において、あらゆるノード１０が、フェーズ２のメッセージであるＭＡＰ＿ＭＩＧＲＡＴＥイベント・メッセージを受信する。ステップ２０２において、ノード１０が、マップ・マイグレーション状態を「１」に設定する。マップ・マイグレーション状態は、各ノード１０に関するＯＳＤ６０およびＳＡ５５がマップ・マイグレーションが開始されたことを知るように設定される。

【００５７】

ステップ２０５において、各ノードのＯＳＤ６０が、それらのＯＳＤ６０がＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして印を付けられたいずれかのＣＧを有するかどうかを判定する。ノード１０がそのようなＣＧを持たない場合、ステップ２１０において、ノードが、Ｐａｘｏｓに従ってＡＣＫメッセージを直ちにポストしてマップ・マイグレーション・イベントに肯定応答する。ＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして印を付けられたＣＧを有するノード１０のＯＳＤは、古いマップ７０の下でボリュームに記憶されたチャンクを有するダウンしていないボリューム（ノード１０）からＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして印を付けられたＣＧに属するチャンクをコピーすることによってコピー・プロセスを開始する。ステップ２２０において、受信ノードに関するＯＳＤ６０が、フェーズ１においてＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴとともに送信されたＣＧＩＤ７７のリストを参照することによってＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして印を付けられたＣＧに関するＣＧを有する古いマップ７０のノードのメンバーからのチャンクのリストを要求する。古いマップ７０のメンバー（たとえば、記憶されたチャンクを有するノード）は、チャンクのリスト（すなわち、コピー・リスト）１１５を新しいマップのメンバーに送信する。

【００５８】

代替的に、それぞれの新しいマップのノード１０（要求を受信するノード）が、基礎を成すファイル・システムのキャッシュの効果を利用するために、同じ古いマップのノード１０がチャンクのリスト１１５および各チャンクに関するスライスを送信することを要求する。ある場合、チャンク（スライス）を受信すべきであるそれぞれの新しいマップ７０のノードが、古いマップの下のどのダウンしていないボリュームからそのノードがチャン

クを受信するかを独立に判断する。異なるダウンしていないボリュームが、同じチャンクを受信すべきであるそれぞれの新しいマップのノードによって選択される可能性がある。別の場合、チャンクを受信すべきであるそれぞれの新しいマップのノードが、チャンクを受信するために古いマップの下の同じダウンしていないボリュームのノードを選択する。後者の場合、新しいマップのノードに送信されるチャンク（スライス）が（以前のディスクの読み取りにより）選択されたダウンしていないボリュームのノードのメモリにあることになるためにキャッシュの効果が利用される可能性があり、したがって、転送されるデータが以前の転送によりノードのメモリに既にある場合、ディスクの読み取りが必要とされない。例として、古いマップ７０のノード１および３が新しいマップ７１のノード２および５に転送される同じデータを記憶する場合、ノード２と５との両方が、データを転送するためにノード１にのみＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を送信する。ノード２がノード１からのデータを要求し、それから、ノード５がノード１からの同じデータを要求する場合、ノード１は、データがノード１のメモリにあることになるので、ノード５にデータを送信するためにディスクの読み取りを実行する必要がない。

【００５９】

ステップ２２５において、受信ノード１０が、コピー・リスト１１５を受信し、ＯＳＤ６０が、メモリ５０内にコピー・リスト１１５に従ってそのＯＳＤ６０が処理しなければならないチャンクのリストを維持する。ノード１０は、コピー・リスト１１５のチャンクを順番にコピーする。コピー・リスト１１５は、過度に大きいコピー・リストによってノードに負担をかけることを避けるために、おおよそ、ＣＧ毎に１０Ｋのエントリを下回ることが好ましい。ステップ２２６において、ノード１０が、各チャンクに関するフラグ・テーブル１２０を生成する。

【００６０】

すべてのノード１０は、それらのノード１０に記憶される各チャンクに関してメモリ５０に記憶されるデータ構造であるフラグ・テーブル１２０を管理する。クラスタ内の各ノードは、クラスタに記憶される各ＣＧの各チャンクに関するフラグ・テーブルにアクセスすることができる。図１２は、フラグ・テーブル１２０の例を示す。フラグ・テーブル１２０は、各チャンクが設定されたＣＬＥＡＮフラグ、ＤＩＲＴＹフラグ、またはＤＥＧＲＡＤＥＤフラグを有するかどうかに関する情報を含む。加えて、フラグ・テーブル１２０は、ＤＥＧＲＡＤＥＤフラグが設定された後のイベント番号と、そのイベント番号の後にデータが書き込まれたイベント番号とを記憶する。設定されるフラグは、値「１」を有し、一方、値「０」はフラグが設定されなかったことを示す。ステップ２２７において、ノード１０が、コピーされる各チャンクに関して、ＤＩＲＴＹフラグを設定する。ノード１０がクラッシュする場合、リカバリを必要とするチャンクのリストが、ＤＩＲＴＹと印を付けられたチャンクに関してＣＧをスキャンすることによって再生成され、これは、下でより詳細に説明される。

【００６１】

ステップ２２８において、ノード１０が、そのノード１０が受信したチャンクのリストの各チャンクに関するスタブを生成する。図１３は、スタブで管理されるチャンク・スタブ管理情報１０５の例である。各ＯＳＤ６０は、ＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤと印を付けられた各ＣＧの各チャンクに関してメモリ５０にチャンク・スタブ管理情報１０５を記憶する。チャンク・スタブ管理情報１０５は、スタブが対応するチャンク番号を示すチャンク・フィールド１０６と、チャンクが対応するＣＧを示すＣＧＩＤフィールド１０７とを含む。チャンク・スタブ管理情報１０５は、チャンクの古いマップの位置１０８と、このチャンクに関するコピー・プロセスが始まったか否かを示すＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求送信済みフィールド１０９とをさらに含む。コピー・プロセスが始まった場合、ＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求送信済みフィールドは、「１」で印を付けられ、そうでない場合、そのＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求送信済みフィールドは、「０」で印を付けられる。また、チャンク・スタブ管理情報１０５は、チャンクのすべてのスライスがコピーされたかどうかを全スライス・コピー済みフィールド１１０で示し、Ｉ／Ｏキー

・リストが空であるかどうかをキー・リスト空フィールド１１１で示す。ＯＳＤ６０は、すべてのスライスがコピーされたかどうかを確認するためにビットマップ７５を調べる。すべてのスライスがコピーされた場合、値が「１」に設定され、そうでない場合、値は「０」である。ＯＳＤは、Ｉ／Ｏキー・リスト９５が空であるか否かを示すためにチャンクのＩ／Ｏキー・リスト９５にいずれかのＩ／Ｏキーがあるかどうかを判定する。Ｉ／Ｏキー・リスト９５が空である場合、フィールドは「１」で印を付けられる。

【００６２】

図１１のステップ２２９において、ノード１０が、各チャンクに関するスライスのビットマップ７５を構築する。図１４は、ビットマップ７５の例である。ビットマップ７５は、揮発性であり、メモリ５０に記憶される。たとえば、３２Ｋのスライスを用いると、６４ＭＢのチャンクを記載するビットマップは、２０４８ビットまたは２５６バイトになる。ステップ２３０において、新しい位置が、古いマップ７０のメンバーのノード１０からソースの位置（ダウンしていないボリューム）を選択する。新しい位置は、ＣＧＩＤ７７に関してＣＲＵＳＨによって生成されたディスクのリスト７６からソース・ボリューム（第１のダウンしていないボリューム）６５を選択する。たとえば、ＣＲＵＳＨは、古いマップ７０からソースＯＳＤ６０を選択する際にある程度の擬似ランダム性を提供し、すべての新しいマップ７１のターゲットがそれらのターゲットのＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を同じソースに送信するように実装され得る。このプロセスは、基礎を成すファイルシステムのキャッシュを利用して、クラスタの処理の負荷に対するリカバリの全体的影響を小さくする。

【００６３】

図３に示されるように、各ＣＧは、チャンクから構成され、さらに、各チャンクは、データのスライスから構成される。データのマイグレーション中、各チャンクは、チャンクの構成要素のスライスを転送することによってマイグレーションされる。ステップ２３５において、ＯＳＤ６０が、スライスを転送するために、ステップ２３０において決定されたダウンしていないボリュームにＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を送信する。ＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求は、ステップ２２５において受信され、メモリに記憶されたチャンクのリスト１１５にある各チャンクの隠すライスに関して送信される。ステップ２３６において、チャンク・スタブ管理情報１０５のＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求送信済みフィールド１０９が、そのチャンクに関するコピーが始まったことを示すために「１」に変更される。

【００６４】

チャンクの各スライスが転送され、新しいノードに書き込まれるとき、ステップ２４５において、そのスライスに対応するビットが、そのスライスが書き込まれたことを示すためにビットマップ７５内で変更される。ビットマップ７５は、各チャンクのどのスライスが書き込まれたかを示す。チャンクに関する各スライスは、ステップ２４０において、ビットマップ７５に従って順番にＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を用いてコピーされ、受信ノード１０の新しい位置のディスク６５に書き込まれ得る。図１４に示される例示的なビットマップ７５においては、マイグレーション中、スライス１が正常に転送された後、概して、残りのスライスに関する順番に従って、スライス２が次に転送される。図１４のビットマップ７５は、各スライスに対応する複数のビットを示す。少なくとも１つのビットが、スライスがコピーされ、ターゲット・ノードに書き込まれたか否かを表す。

【００６５】

ノード１０は、ビットマップ７５を調べて、それぞれのチャンクのすべての内容がコピーされたかどうかを判定する（２５０）。ノードは、チャンク・コピー・リスト１１５の各チャンクに関するビットマップ７５がすべてのスライスがコピーされたことを示すかどうかを判定する。ステップ２５１において、ＯＳＤが、そのチャンクに関してすべてのスライスがコピーされたことを示すためにチャンク・スタブ管理情報１０５の全スライス・コピー済みフィールド１１０を「１」に設定する。いずれかのスライスがコピーされていない場合、ステップ２３５において、それらのスライスに関してＲＥＣＯＶＥＲＹ＿ＲＥ

ＡＤ要求が送信される。ステップ２５２において、チャンクが、ターゲット・ノードのチャンク・コピー・リスト１１５から削除される。ノードは、チャンク・コピー・リスト１１５の各チャンクに関するスライスのコピーを続ける。ステップ２５３において、すべてのチャンクがコピーされ終わっていない場合、プロセスが、ステップ２２８に戻る。すべてのチャンクの内容のすべてがコピーされたとき、新しい位置（すなわち、新しいノード）は、ステップ２５５において、フェーズ２に肯定応答するＡＣＫメッセージを送信する。フェーズ２は、すべてのノードがそれらのノードのマイグレーション作業を完了したという報告をＰａｘｏｓに従って返すときに終了する。

【００６６】

ステップ２２７において、チャンクがたとえばＷＲＩＴＥオペレーションによって修正されていること、またはチャンクがコピーされるべきであることを示すために、チャンクが、フラグ・テーブル１２０においてＤＩＲＴＹフラグによって印を付けられる。ＤＩＲＴＹフラグは、チャンクが書き込まれるかまたはそれ以外の方法で修正される前に設定される。ＤＩＲＴＹフラグは、ＳＡ５５がＷＲＩＴＥオペレーションを送ったすべてのＯＳＤ６０がＷＲＩＴＥに応答した後に、たとえば、ＳＡ５５によって非同期にクリアされる。チャンクは、ＤＲＩＴＹと印を付けられる場合、チャンク・コピー・リスト１１５上か、ビットマップ７５上か、Ｉ／Ｏキー・リスト９５上か、またはビットマップ７５とＩ／Ｏキー・リスト９５との両方の上かのいずれかにある。各チャンクのリカバリが完了されるとき、そのチャンクのＤＩＲＴＹフラグがクリアされ、このことは下で説明される。

【００６７】

マイグレーション中、ＳＡ５５は、Ｉ／Ｏオペレーションを受け取る場合、ＣＧを記憶する古いマップ７０の下のノード１０と、ＣＧを記憶する新しいマップ７１の下のノード１０との両方に、ＣＧ内の特定のデータを対象とするＩ／Ｏオペレーションを送らなければならない。このプロセスは、マイグレーションするチャンクの内容が、古いマップ７０の下のノード１０へのたとえばホスト２０からの、ＳＡ５５によって処理されたＷＲＩＴＥ要求によって上書きされる場合、チャンクの完全性が維持されることを保証する。

【００６８】

図１５および１６は、本発明の実施形態によるマイグレーション中のＷＲＩＴＥ要求のプロセスの流れ図を示す。図１５のステップ４００において、ノード１０が、ＷＲＩＴＥ要求を受信する。ステップ４０５において、受信ノード１０のＳＡ５５が、現在の（古い）マップ７０の下で記憶されるＣＧにＷＲＩＴＥが影響を与えるかどうかを調べる。ＳＡ５５は、ＳＡ５５が記憶するＣＧにＷＲＩＴＥが影響を与えると判定する場合、ステップ４１０において、マップ・マイグレーション状態を調べる。ＳＡ５５が記憶するＣＧにＷＲＩＴＥが影響を与えないとＳＡ５５が判定する場合、プロセスは始めに戻る。ノード１０は、マップ・マイグレーション状態にない場合、ＷＲＩＴＥ要求をＯＳＤ６０に送信し、ＷＲＩＴＥが通常通り適用される（４２５）。ノード１０がマップ・マイグレーション状態にあり、ＷＲＩＴＥ要求に関するＣＧが影響を与えられる場合、ＳＡ５５は、古いマップ７０と新しいマップ７１との両方の下のＣＧを記憶するノード１０にＷＲＩＴＥ要求を送信しなければならない。ノードは、それぞれ、どのＣＧがマイグレーション状態にあるかを特定することができる。ステップ４１５において、ノードが、そのノードが古いマップ７０と新しいマップ７１との両方の下でそのノードのローカル・ストレージ・デバイス６７に記憶されたＣＧのコピーを有するかどうかを判定する。ノードがＣＧのコピーを記憶する場合、ただ１つのＷＲＩＴＥ要求がＯＳＤ６０に送信され、ＷＲＩＴＥが適用される（４２５）。ＷＲＩＴＥが通常通り適用され、処理が図１７のステップ８００に進む。そうではなく、ノードが古いマップ７０と新しいマップ７１との両方の下で記憶されたＣＧのコピーを持たない場合、ＷＲＩＴＥ要求は、ステップ４２０において、新しいマップの下のＣＧを記憶するノードに送信され、処理は、下で説明される図１６に示されるように進む。

【００６９】

図２６は、Ｉ／Ｏオペレーションの受け取りを扱うためのホスト２０とノード１０ａお

よび１０ｂとの間の通信フローを示す。図２６において、ホスト２０は、クラスタに記憶されたデータに対するＩ／Ｏオペレーションの Ｉ／Ｏ要求２５０１をクラスタ内のノード１０ａに送信する。場合によっては、ホスト２０からＩ／Ｏ要求を受信するノード１０ａが、Ｉ／Ｏ要求に関するデータを記憶しない可能性がある。図２６に示されるように、そのような場合、受信ノード１０ａは、Ｉ／Ｏオペレーションのための適切なノード１０ｂを指定するリダイレクト２５０２をホスト２０に送信することによって適切なノード１０ｂにＩ／Ｏ要求をリダイレクトし得る。リダイレクト２５０２に応答して、ホスト２０は、Ｉ／Ｏ要求を扱う適切なノードであるノード１０ｂにＩ／Ｏ要求２５０３を送信し得る。Ｉ／Ｏオペレーションを完了した後、ノード１０ｂは、応答２５０４をホスト２０に送信し得る。

【００７０】

図２７は、クラスタ内のノード１０ａへのクラスタに記憶されたデータに対するＩ／Ｏオペレーションに関する代替的な通信フローを示す。図２７において、ホスト２０は、クラスタに記憶されたデータに対するＩ／Ｏオペレーションの Ｉ／Ｏ要求２６０１をクラスタ内のノード１０ａに送信する。場合によっては、受信ノード１０ａは、Ｉ／Ｏオペレーションのための適切なノード１０ｂにリダイレクト２６０２を送信することによって適切なノード１０ｂにＩ／Ｏ要求を直接リダイレクトし得る。Ｉ／Ｏオペレーションを完了した後、ノード１０ｂは、応答２６０３をホスト２０に送信し得る。

【００７１】

図１７は、通常の条件の間にＩ／Ｏオペレーションを受け取るノード１０のプロセスの流れ図を示す。通常の条件は、マップ・マイグレーション状態が０であり、マップ変更プロセスの下でのマイグレーションがノードで現在処理されていないことを示すときとして定義され得る。ステップ８００において、ノード１０に関するＳＡ５５が、Ｉ／Ｏオペレーションを受け取る。

【００７２】

Ｉ／Ｏオペレーションを受け取ると、受信ノードが、ノードに関するＯＳＤ６０にＩ／Ｏオペレーションを送り（８０１）、ＯＳＤ６０が、Ｉ／Ｏオペレーションに対応するチャンクに関するＩ／Ｏキー・リスト９５にオペレーションを追加する（８０２）。ステップ８０５において、ノードが、チャンクに関するフラグ・テーブル１２０のＤＩＲＴＹフラグを「１」に設定する。Ｉ／Ｏオペレーションが、ステップ８１０において、チャンクに対してＯＳＤ６０によって実行される。ステップ８１５において、オペレーションが、対応するチャンクのＩ／Ｏキー・リスト９５から削除される。プロセスは、チャンクのＩ／Ｏキー・リスト９５にいずれかのその他のオペレーションが存在するかどうかをノード１０が判定するときステップ８２０に続く。チャンクに関するその他のオペレーションがある場合、プロセスはステップ８１０に進み、オペレーションが実行される。いかなるその他のオペレーションもない場合、チャンクに対応するフラグ・テーブル１２０においてＤＩＲＴＹフラグが「０」に設定される（８２５）。代替的に、プロセスは、ステップ８２５に進む前にステップ８２０においていずれかのその他のオペレーションがあるかどうかを調べることなくステップ８１５からステップ８２５に進む可能性がある。

【００７３】

新しいマップ７１のメンバーのいずれかがＷＲＩＴＥ要求を完了することができない場合、ＳＡ５５は、新しいマップ７１のチャンクに、フラグ・テーブル１２０のＤＥＧＲＡＤＥＤフラグによってＤＥＧＲＡＤＥＤとして印を付ける。加えて、古いマップ７０のノードのＳＡ５５は、古いマップ７０のいずれかがＷＲＩＴＥに失敗する場合、チャンクにＤＥＧＲＡＤＥＤとして印を付ける。チャンクは、チャンクのいずれかのコピー（いずれかのＣＧのコピー）がオンラインおよび応答可能ではないことがノード１０によって知られているとき、チャンクが書き込まれていることを示すためにＤＥＧＲＡＤＥＤとして印を付けられる。チャンクは、ＳＡ５５がＩ／Ｏ要求にＤＥＧＲＡＤＥＤ状態を生じているものとしてフラグを立てるためか、またはＳＡ５５が丁度書き込まれたＤＩＲＴＹチャンクのＤＥＧＲＡＤＥＤの印を要求し、いくつかのミラー（たとえば、ＣＧのコピーを記憶

するノード）がＷＲＩＴＥ中にエラーを返したかもしくはクラッシュしたが、それでも、高い信頼性でデータをリカバリするのに必要な最小限の数を形成する十分なＣＧのコピーが存在することを知っているためかのどちらかでＤＥＧＲＡＤＥＤとして印を付けられる。チャンクにＤＥＧＲＡＤＥＤとして印を付けることは、ＤＩＲＴＹフラグを削除することと、アクセス不可能なコピーがアクセス可能になるとき、たとえば、ノードが障害の後にオンラインに復帰するときにチャンクをリカバリする必要性に関する情報を失わないこととを可能にする。加えて、古いマップに対応するノードのＳＡ５５は、古いマップ７０のいずれかがＷＲＩＴＥに失敗する場合にチャンクにＤＥＧＲＡＤＥＤとして印を付ける。古いマップに対応するノードのすべてのノード１０は、そのノード１０がＣＧのすべてのコピーが存在し、リカバリされると判定する場合、ＤＥＧＲＡＤＥＤフラグをクリアする。

【００７４】

マイグレーション・プロセス全体を通じて、新しいマップ７１の下でデータを記憶しないノード１０を含む、古いマップ７０のすべてのノード１０は、Ｉ／Ｏ要求を受け付け続ける。古いマップ７０のノードの下での各ノード１０に関するＳＡ５５は、図１５を参照して上で説明されたように、ＷＲＩＴＥ、ＴＲＵＮＣＡＴＥ、およびＲＥＭＯＶＥ要求を新しい位置へ転送する。ＳＡ５５は、古い位置への要求と平行してそれぞれの新しい位置にオペレーションを送る。

【００７５】

図１６は、古いマップの下のＣＧのコピーを記憶するＳＡ５５からのＷＲＩＴＥ要求を受信するターゲット・ノードのプロセスの流れ図を示す。ステップ４３５において、新しい位置が、ＷＲＩＴＥを受信する。ステップ４４０において、ノード１０が、全スライス・コピー済みフィールド１１０を参照することによってチャンクのすべてのスライスがコピーされたかどうかを判定するためにチャンク・スタブ管理情報１０５を調べる。チャンクは、ソースから既にコピー済みである（つまり、そのチャンクがＣＬＥＡＮである）場合、ステップ４４５においてＤＩＲＴＹとして印を付けられ、ＷＲＩＴＥが、チャンクに直ちに適用される（ステップ４５０）。チャンクがＤＩＲＴＹとして印を付けられる場合、ＯＳＤ６０は、ステップ４５５において、ビットマップ７５を調べて、ＷＲＩＴＥが当てはまるスライスがコピー済みであるか否かを判定する。ステップ４５５において、ビットマップ７５がスライスがコピー済みであることを示す場合、ステップ４９０において、ＷＲＩＴＥが、対応するスライスに適用される。書き込みが当てはまるスライスがコピー済みでない場合、ステップ４６０において、ノードが、ＷＲＩＴＥオペレーションによって書き込まれるべきであるデータがスライス・サイズ未満であるかどうかを調べる。ＷＲＩＴＥによって書き込まれるべきであるデータがスライス・サイズ以上である場合、ＷＲＩＴＥは、適切なスライスの位置に適用される（４８０）。ビットマップ７５のビットが、スライスがマイグレーション中に新しいマップの位置にコピーされることに晒されないようにそのスライスに関して変更される（ステップ４８５）。ＷＲＩＴＥによって書き込まれるべきであるデータがスライス・サイズ未満である場合、そのスライスに関するデータがマイグレーションされ得るように、そのスライスに関するＲＥＣＯＶＥＲＹ＿ＲＥＡＤ処理が、ビットマップ７５の順序を外れて（つまり、スライスの順序を外れて）送られる（４７０）。つまり、そのスライスに関するＲＥＣＯＶＥＲＹ＿ＲＥＡＤが、ビットマップ７５による順番で通常は次に転送されるスライスの前に送られる。そのスライスに関するデータがマイグレーションされ、コピーされる（４７５）と、ＷＲＩＴＥオペレーションが適用され、ステップ４８５において、ビットマップ７５のビットが、そのデータがコピーされたことを示すためにそのスライスに関して変更される。

【００７６】

（トランケートするための）ＴＲＵＮＣＡＴＥまたは（削除するための）ＲＥＭＯＶＥなどのその他の修正オペレーションが、古いマップ７０と新しいマップ７１との両方の下のノード１０に送られる。概して、ＷＲＩＴＥ要求に関する同じ処理が、ＴＲＵＮＣＡＴＥおよびＲＥＭＯＶＥオペレーションに当てはまる。トランケート処理は、コピー・プロ

セスがどこまで進捗しているかに応じて異なる。チャンクがまだコピーされていない場合、ＴＲＵＮＣＡＴＥオペレーションは、成功する非オペレーション（ｎｏｎ－ｏｐｅｒａｔｉｏｎ）である。チャンクは、既に完全にコピー済みである（そのチャンクがＣＬＥＡＮである）場合、フラグ・テーブル１２０においてＤＩＲＴＹと印を付けられ、ＴＲＵＮＣＡＴＥオペレーションが、通常通り適用される。ＴＲＵＮＣＡＴＥオペレーションがあるときにチャンクがＤＩＲＴＹであり（コピーされており）、ビットマップ７５が、トランケート・ポイント（ｔｒｕｎｃａｔｉｏｎ　ｐｏｉｎｔ）が現在のコピー・オフセット（ｃｏｐｙ　ｏｆｆｓｅｔ）を過ぎている（まだコピーされていない）ことを示す場合、ＴＲＵＮＣＡＴＥオペレーションは、成功する非オペレーションである。ＴＲＵＮＣＡＴＥオペレーションがあるときにチャンクがＤＩＲＴＹであり、ビットが、チャンクのこのスライスが既にコピー済みであることを示す場合、チャンクは、ソースからのさらなるコピーを防ぐためにロックされる。そして、チャンクは、ＴＲＵＮＣＡＴＥオペレーションによってトランケートされ、ソースからさらにコピーされない。

【００７７】

マイグレーション中に、Ｉ／ＯオペレーションがＲＥＭＯＶＥである場合、チャンクが、新しい位置から削除され、インメモリ構造からパージされる。この場合、ＯＳＤ６０は、チャンクが削除される前に、もしあれば、チャンクの保留中のコピーがコピーされるまで待つ。チャンクがＣＧから削除されている場合、新しいマップのメンバーは、そのチャンクも削除する。

【００７８】

マイグレーション中のＩ／Ｏ処理中に、新しい位置のＯＳＤは、通常のＷＲＩＴＥオペレーションにおいてＯＳＤ６０が行うのと同じ方法で要求元のＳＡのキーが追跡され続けるようにする。図１８は、本発明による、マイグレーション中のＩ／Ｏオペレーションが完了した後にＣＨＵＮＫをクリーンに設定するプロセスの流れ図を示す。ステップ５０５において、オペレーションが完了されたとき、ＯＳＤ６０が、ＳＡにＡＣＫメッセージによって応答する（ステップ５１０）。その後、ＯＳＤ６０は、古いマップ７０と新しいマップ７１との両方のＯＳＤにおいてノード１０にＣＬＥＡＮ指令を送信する（５２０）。ステップ５２５において、各チャンクに関して、それぞれの新しい位置が、チャンクのＩ／Ｏキー・リスト９５からＩ／Ｏキーを削除する。新しい位置のノード１０は、チャンクをＣＬＥＡＮに設定する前に次のこと、すなわち、チャンクの残りのＩ／Ｏキーのリスト９５が空であるかどうか（ステップ５３０）、ビットマップ７５のビットがすべてのスライスがコピー済みであることを示すかどうか（ステップ５３５）、およびチャンクがチャンク・コピー・リスト１１５にあるかどうか（５４０）を調べる。代替的に、ノード１０は、ステップ５３５において、チャンク・スタブ管理情報１０５を参照して全スライス・コピー済みフィールドが「１」であるかまたは「０」であるかを判定することによって、すべてのスライスがコピー済みであるかどうかを調べる可能性がある。これらの要因のいずれかが満たされない場合、チャンクは、ステップ５５０においてフラグ・テーブル１２０でＤＩＲＴＹのままにされる。これらの要因のすべてが満たされる場合、ＤＩＲＴＹフラグが、そのチャンクに関してクリアされ、チャンクは、チャンクに対応するフラグ・テーブル１２０でステップ５４５においてＣＬＥＡＮとして印を付けられる。ステップ５５５において、ノードが、チャンクがクリーンであると印を付けられるために必要なプロセスを実行する。たとえば、ステップ５３０において、チャンクのＩ／Ｏキー・リストが空でないと判定される場合、ノードは、残りのＩ／Ｏオペレーションを実行する。プロセスが完了した後、チャンクが、チャンクをＣＬＥＡＮに設定するために再び調べられる（つまり、ステップ５３０～５４０）。チャンクをＣＬＥＡＮに設定すべきかどうかを判定する際に考慮される可能性がある要因は、少なくとも最小限の数を示す予め決められた値に関するＣＧに対するＷＲＩＴＥオペレーションが正常に完了されたかどうかを判定することも含む。チャンクは、ＤＩＲＴＹのままにされ、最小限の数があるときにリカバリされ得る。この場合、すべてのＣＧのコピーおよびそれらのＣＧのコピーのチャンクが、Ｐａｘｏｓのイベント番号を考慮に入れることを含めて調べられる可能性がある。換言すると

、システムは、十分なコピーが利用可能であるとき、チャンクに関するすべてのコピーが同一であることを確認する。代替的に、ＤＩＲＴＹフラグは、少なくともＣＧのコピーの最小限の数に対するＷＲＩＴＥオペレーションが正常に完了されたときにクリアされる。最小限の数は、チャンク／ＣＧのＣＬＥＡＮなコピーを有するノードの最少数である。

【００７９】

最小限の数が確立されると、すべてのノードが、チャンクのいずれかのコピーがＤＩＲＴＹフラグを有するかどうかを判定する。ＤＩＲＴＹフラグを有するチャンクがある場合、ノードに記憶されたＤＩＲＴＹなコピーを有するノードが、ダウンしていないボリュームを選択し、チャンクをリカバリするためにＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を送信する。ノードは、新しいマップ７１からダウンしていないボリュームのメンバーを選択する。

【００８０】

ノードのクラッシュ

マイグレーション中、ノード１０またはノード１０のディスク６５がクラッシュする可能性があり、それがマイグレーションを妨げ、この場合、マイグレーション中に継続しているＷＲＩＴＥオペレーションは、ノードがクラッシュする前に処理されるチャンクの一部に過ぎない可能性があり得る。この状況は、そのようなチャンクがフラグ・テーブル１２０でＤＩＲＴＹであり、ＣＧがマイグレーション中にＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして印を付けられ、このことが、対応するデータの内容が信頼できないことを示すために検出され、リカバリされ得る。

【００８１】

図１９は、ノードがクラッシュの後にオンラインに復帰するときに従うプロセスに関する流れ図を示す。ノード１０またはディスク６５がオンラインに復帰するときに、ノードは、ステップ５６０において、どのチャンクがＤＩＲＴＹとして印を付けられているか、およびどのＣＧがＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして印をつけられているかを判定する。代替的に、ノード１０は、ノードのクラッシュまたは障害の後にリストが利用可能である場合、チャンク・コピー・リスト１１５を参照して、どのチャンクがコピーされる必要があるかを判定し得る。この判定から、ノード１０は、ステップ５６５において、コピーされるチャンクのリストを生成する。ノードは、一部のスライスがクラッシュの前にコピーされたものとして示されているとしても、コピーされるチャンクに関してビットマップ７５のビットをクリアする（５７０）。加えて、ステップ５７０において、チャンク・スタブ管理情報１０５がクリアされる。ノード１０は、ステップ５７５においてチャンクをコピーするために妥当なソース・ノード（ダウンしていないボリューム）を選択し、新しく書き込まれるビットを含むチャンク全体が、ＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を用いて妥当なソースからコピーされる。ステップ５８０において、処理が、図１１のステップ２３５に進んで、ステップ５７５からの決定されたダウンしていないボリュームにＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を送信し、データをコピーするためにフェーズ２のステップ２３５に続く処理に従って進行する。

【００８２】

ノードは、通常のオペレーション中に（つまり、マップ変更またはマイグレーション中でないときに）クラッシュする場合、図１１のステップ２５５のようにＰａｘｏｓにＡＣＫメッセージをポストしないことを除いて上記のルーチンに従って回復する。ノードは、ＷＲＩＴＥオペレーションなどのデータを修正するＩ／Ｏオペレーション中にクラッシュする場合、決定されたダウンしていないボリュームから新しく書き込まれたデータをリカバリする。したがって、データの完全性が維持される。

【００８３】

妥当なソースは、マイグレーションを再開する古いマップ７０からのノード、または妨げられたマップ変更を完了するために十分に自己をリカバリした新しいマップ７１のメンバーからのノードである可能性がある。そして、それぞれの障害が起こったノードが、上述のように、チャンクをＣＬＥＡＮとして設定する処理を経る。

【００８４】

マップ・マイグレーション中のソース・ノードの障害

ソース・ノード１０またはノード１０のディスク６５がマイグレーション中にクラッシュする場合、図２０に示されるように以下の処理フローが開始される。ステップ６０１において、クラッシュの前に完全にコピーされなかったそれぞれのＣＧに関して、障害が起こったソース・ノードからチャンクを受信していた新しいマップ７１のメンバーは、ＣＲＵＳＨを用いて、マイグレーションされる必要があるＣＧに関してそれに記憶されたＣＧを有する古いマップ７０の下でディスク・リスト７６を生成する。そして、ノードは、ステップ６０５において、リストの第１のダウンしていないボリュームを選択する。第１のダウンしていないボリュームであるように決定されるボリューム６５は、「有効な」コピーと呼ばれる。有効なコピーを保持すると決定されたノードは、（クラッシュし、今ではオンラインに復帰している同じノードを含む）古いマップ７０と新しいマップ７１との両方の下のピア・ノードをリカバリする。有効なコピーは、ここで、ターゲット・ノードからＲＥＣＯＶＥＲＹ＿ＲＥＡＤを受信する（たとえば、ステップ２３５）。言い換えると、ターゲット・ノードが、データ転送のソースとなるように古いマップのメンバーの中で新しいダウンしていないボリュームを選択する。別の言い方をすれば、上述の通常のリカバリ・プロセスは、同じである。有効なＯＳＤは、マイグレーション中にクラッシュした１つのノードまたは複数のノード１０がオンラインに復帰するときにそれらのノード１０をリカバリする。有効なＯＳＤは、すべてのノードが利用可能な状態情報１００を調べることによって、クラッシュした位置がオンラインに復帰していることを知っている。

【００８５】

本発明のこの態様によれば、ソース・ノード１０が障害から復帰するために単一のルーチンのみが必要とされる。ルーチンは、セルのマップ・マイグレーション状態を調べて、ピア・リストを生成するとき、およびどのＯＳＤが有効なＯＳＤであるかを決定するときに、それらのセルがマップ・マイグレーションのフェーズ２にあるかどうかを判定する。

【００８６】

書き込み不可能と判定されるＣＧ

ターゲット・ノード１０がクラッシュするか、またはターゲット・ノード１０のディスク６５がクラッシュする場合、ＣＧのコピーを記憶するノード１０が、ターゲット・ノードの障害があっても最小限の数が維持されるかどうかを判定する。最小限の数は、ＣＧのコピーを記憶するノード１０またはディスク６５の数が最小限の数の値を超えているときに維持される。最小限の数の値は、各ノード１０が知っている予め決められた値である。代替として、最小限の数は、ＣＧに関するチャンクを含むＯＳＤ６０の数が予め決められた最小限の数の値を超える場合にやはり満たされる可能性がある。新しいマップ７１の下のノード１０が最小限の数を維持する場合、リカバリが継続する。新しいマップ７１のメンバーの数が最小限の数を下回る場合、残りのＯＳＤ６０が、リカバリを完了するが、ＣＧは、書き込み不可能と判定される可能性がある。ＣＧが書き込み不可能であると判定される場合、この状態は、次のマップ変更があると変更される。ＣＧは、最小限の数が現在利用可能でない場合、書き込み不可能であると判定される。

【００８７】

ソース・ノード１０がクラッシュするか、またはソース・ノード１０のディスク６５が障害にあう場合、マイグレーションは、（上で説明されたように）ダウンしていないその他のソースＯＳＤ６０がある限り継続する。ソースのコピーのすべてがダウンしている場合、ターゲット・ノード１０（受信ノード）はマイグレーションを断念する。この場合、ターゲット上で構築されたＣＧのコピーは、ＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤと印を付けられたままであり、ソースのコピーがオンラインに復帰し、利用可能にされるときに可能であればリカバリされる。この場合、ＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤとして印を付けられたＣＧのコピーは、リカバリの目的を除いて読み取り可能または書き込み可能でない。

【００８８】

フェーズ３

図２１は、本発明の実施形態によるフェーズ３のプロセスの流れ図を示す。ステップ３０１において、提案元が、すべてのノード１０がステップ２５５において示されるように肯定応答を生成することによってフェーズ２に肯定応答したかどうかを調べる。すべてのノード１０がフェーズ２に肯定応答したとき、フェーズ２は完了する。提案者は、提案されたマップ７１のコミットである、マップ変更のフェーズ３を開始する。ステップ３０５において、マップ変更に関するＭＡＰ＿ＣＯＭＭＩＴイベント・メッセージが、クラスタのすべてのノードにＰａｘｏｓに従って送信される。ステップ３１０において、コミット・メッセージを受信したすべてのＯＳＤが、それらのＯＳＤが記憶するＣＧを評価する。ステップ３１５において、各ＯＳＤが、そのＯＳＤが記憶する各ＣＧを新しいマップ７１と付き合わせて、そのＯＳＤが記憶するあらゆるＣＧに関して、ノード１０が新しいマップの下のＣＧの適切な保有者であるかどうかを調べる。

【００８９】

ノード１０は、そのノード１０が記憶するＣＧが新しいマップ７１の下でノードによって記憶されるものとして示されていないと判定する場合、ステップ３２０において、永続的なＣＧ＿ＩＮＶＡＬＩＤ＿ＰＬＡＣＥＭＥＮＴフラグによってＣＧに印を付け、ステップ３３０において、そのＣＧのチャンクのすべての非同期の削除および空間の再利用を開始する。そのＣＧに関して記憶されたすべてのチャンクが、ノード１０から削除される。チャンクの削除の後、ステップ３３５において、ＣＧの階層およびチャンク・スタブ管理情報１０５が削除され、さらに、ＣＧがＯＳＤ６０に対応するキャッシュから削除される。ＯＳＤは、そのＯＳＤが新しいマップ７１に従ってＣＧを適切に記憶することを発見する場合、ＢＥＩＮＧ＿ＣＯＮＳＴＲＵＣＴＥＤフラグが設定されている場合、ステップ３２５において、ＣＧからそのフラグをクリアする。ステップ３４０において、新しいマップが、コミットされ、あらゆるＳＡが、今や新しいマップを使用し、あらゆるノードが、今や新しいマップ７１のデータ配置に従ってＩ／Ｏオペレーションを受け付ける。

【００９０】

冗長性のポリシーの変更

当業者は、ストレージ・システムにおいて冗長性のレベルが変更され得ることを理解するであろう。たとえば、ストレージ・システムの冗長性のポリシーが、ＲＡＩＤ１～ＲＡＩＤ６の間で設定される可能性があり、その後、変更される可能性がある。冗長性のレベルは、下のようにクラスタに反映される可能性があり、ＲＡＩＤ１に関しては異なるミラー・カウント（ｍｉｒｒｏｒ　ｃｏｕｎｔ）を用いてポリシー変更に関して説明される。

【００９１】

ミラー（コピー）・ポリシーは、たとえば、管理コマンドによって変更され得る。また、マップ変更の提案が、冗長性のレベルの変更に基づいて行われる可能性がある。ポリシーの変更が増加であるかまたは減少であるかに関わらず、新しいマップ７１が生成される。ミラーの数を増やす場合、冗長なＣＧのコピーの数が新しいマップのレイアウトに従ってクラスタ内で配信されなければならないので、マップ変更が行われる。ミラーの数を減らす場合、クラスタ内のノードからＣＧのコピーを削除し、クラスタ内の残りのデータの配置を再配信するためにマップ変更が行われる。

【００９２】

図２２は、各ＣＧに関連するミラー・カウントを示すポリシー・テーブル８５の例である。この例において、ミラー・カウントは、各ＣＧに関して指定されるが、クラスタ全体またはクラスタのサブセクションに関して指定される可能性もある。ミラー・カウントは、たとえば、整数３であり、管理者によって指定される可能性がある。加えて、ミラー・カウントの既定値が、クラスタ内のすべてのＣＧに関してストレージ・システムによって設定される可能性がある。

【００９３】

ＣＲＵＳＨのシーケンスに従って、ディスク６５（ボリューム）が、シーケンスの終わりに追加および削除される。つまり、ＣＲＵＳＨは、３ディスクに関して求められる場合、ディスク・リスト７６｛Ａ，Ｂ，Ｃ｝を生成する。そして、ＣＲＵＳＨは、４ディスク

に関して求められる場合、ディスク・リスト７６｛Ａ，Ｂ，Ｃ，Ｄ｝を生成し、一方、２ディスクに関して求めることは、ディスク・リスト７６｛Ａ，Ｂ｝を生成する。変更の性質に関係なく、ストレージ・システムは、リストの第１のダウンしていないメンバーが前のディスク・リスト７６にあったか、またはそうではなく、データが完全にアクセス不可能であり、前のＣＧのリストからのディスク６５が利用可能になるまでチャンクに関して何も行われ得ないかのどちらかを保証される。

【００９４】

図２３の流れ図は、ストレージ・システムにおけるポリシー変更のためのプロセスを示す。管理者は、ＣＧに関するミラー・カウントを増やすかまたは減らす。新しいミラー・カウントは、ポリシー・テーブル８５に記憶される。ポリシー・変更が行われるために、ＰＯＬＩＣＹ＿ＣＨＡＮＧＥイベント・メッセージが、Ｐａｘｏｓに従ってクラスタ内のノード１０に送信される。ＰＯＬＩＣＹ＿ＣＨＡＮＧＥイベント・メッセージは、ポリシー変更がＣＧに関するミラー・カウントの増加であるかまたは減少であるかを含む。メッセージは、ステップ７０１において、クラスタ内のＯＳＤ６０によって受信される。ステップ７０５において、ノードが、そのノードが記憶するポリシー変更によってどのＣＧが影響を与えられるかを判定する。ステップ７１０において、影響を与えられる各ＣＧについて、ノードが、変更に基づいて新しいマップ７１を構築する。ポリシー変更が増加である場合、それぞれのＣＧのレプリカ・コピーに関する新しい位置を含む新しいマップ７１が構築される。たとえば、ポリシー変更がミラーの数を２増やした場合、新しいマップ７１は、ＣＧのレプリカ・コピーに関して２つの新しい位置を有する。新しいマップに従って、ポリシー変更に合致するように生成されたＣＧの新しいコピーが、ポリシー変更の前にＣＧを記憶しなかったクラスタ内の位置に記憶される。ポリシー変更が減少である場合、新しいマップ７１は、古いマップ７０の下で含まれたＣＧの（古いポリシーと新しいポリシーとの間のミラー・カウントの差に応じた）１つまたは複数のレプリカ・コピーを含まない。

【００９５】

ステップ７１５において、ノードが、ポリシー変更がＣＧのコピーの数を減らすかまたはＣＧのコピーの数を増やすかを判定する。ポリシーが、コピーの数を減らし、ＯＳＤ６０が、ステップ７２０において、そのＯＳＤ６０が新しいマップの下のＣＧのコピーを記憶するようにもはや求められないことを発見する場合、そのＯＳＤ６０は、ステップ７２５において、そのＯＳＤ６０のローカル・ストレージからのＣＧの非同期の削除を開始する。ステップ７２０中に、ノードが、そのノードが新しいマップ７１の下のＣＧのコピーを記憶すると判定する場合、たとえその他のコピーがクラスタのその他の場所で削除される可能性があるとしても、そのノードは、図９のステップ１３５に進む（つまり、フェーズ１でＰａｘｏｓに受け付けをポストする）。

【００９６】

しかし、ポリシーがコピーの数を増やす場合、現在の（古い）マップ７０からのノード１０のＯＳＤ６０が、ディスク・リスト７６が古いポリシーの下で利用可能であるかどうかに応じて、新しいマップ７１の下のＣＧを記憶するすべてのメンバーか、または新しいマップ７１の下の新しいノード１０のみかのどちらかに、それらのノードが今やＣＧのコピーを記憶しなければならないことを知らせる。ステップ７４０において、ＣＲＵＳＨによって生成されたディスク・リストの第１のダウンしていないボリュームが、新しいマップ７１の下のすべてのターゲット・ノードのメンバーにそれらのメンバーが記憶すべきＣＧを有することを知らせるノード１０であるように決定される。その後、新しいメンバーが、新しいマップ７１の下の各ターゲット・ノード１０にＣＧ＿ＭＩＧＲＡＴＩＯＮ＿ＲＥＱＵＥＳＴを送信することによってステップ７４５において通知される。ステップ７５０において、プロセスは、マップ変更プロセスのフェーズ１のステップ１５０に進む。

【００９７】

リストがポリシー変更のために生成されるときに利用可能でないディスク６５に特定のＣＧが記憶される場合、そのようなＣＧは、ダウンしていないいかなるノードによっても

見つけられず、ポリシー変更のリカバリを経ない。しかし、ポリシー変更イベントは、記録され、ターゲット・ノード１０がオンラインに復帰し、ノードがその時点でリカバリ・プロセスを経るときにＣＧを記憶するＯＳＤ６０にＰａｘｏｓに従って利用され得る。

【００９８】

本発明のストレージ・システムは、ディスク６５およびノード１０の障害を追跡する。予め決められた期間の後、オフラインになった（１つもしくは複数の）ディスク６５または（１つもしくは複数の）ノード１０がオンラインに復帰しない場合、それらのディスク６５またはノード１０は、マップ構成から削除され、新しいマップ７１が、提案される。この場合、新しいマップ７１が、上述のマップ変更プロセスによって、データ・マイグレーション、および新しいマップ７１へのデータの複製をトリガする。結果として、冗長性のポリシーによって指定された冗長性のレベルがリカバリされる。予め決められた時間は、何らかのデータのすべてのコピーの全損失をもたらすその後の障害の可能性を最小化することの考慮に基づく可能性がある。また、システム管理者が、マップ変更コマンドを発する可能性がある。

【００９９】

部分的に書き込まれたデータのリカバリ

以下のルーチンに関するステップの一部の詳細は、既に説明されており、以下で繰り返されない。各ノードが、クラスタ内のイベントの通知を受信する。イベントは、ディスクもしくはノードの障害、またはディスクもしくはノードが再び加わることもしくは追加されることである可能性がある。発生が通知されるとクラスタの各ノードが、そのノードが記憶するＣＧを評価して、そのノードがイベントによって影響を与えられるＣＧのいずれかのコピーを記憶するかどうかを判定する。そのとき、ノードは、発生に基づいて影響を与えられるＣＧに関するチャンクを再構築またはコピーするためのアクションをそのノードが行う必要があるかどうかを判定する。図２４は、本発明の実施形態による、クラスタに再び加わった後の部分的に書き込まれたデータのノードのリカバリのプロセスの流れ図を示す。図２５は、本発明の実施形態による、クラスタに再び加わった後の部分的に書き込まれたデータのノードのリカバリのプロセスのメッセージ・シーケンスを示す。

【０１００】

ステップ９００において、ノードまたはディスク６５（論理ボリューム）が、そのノードまたはディスク６５がオフラインであった後にクラスタに再び加わる。ディスクがＩ／Ｏオペレーションに失敗し、再試行またはリカバリＩ／Ｏオペレーションも失敗した場合、ディスクはオフラインであると宣言され得る。ノード／ディスク（ボリューム）は、オンラインに復帰し、アクティブ状態であり、ノード１０は、再び加わったノードと呼ばれる。すべてのノードは、通知、および状態情報１００の参照によってノード１０がクラスタに再び参加したことを知っている（９０１）。ステップ９０５において、再び加わったノードを含むすべてのノードが、現在のマップの下で、そのノードが再び加わったノードに記憶されるいずれかのＣＧを有するかどうかを判定する。そのノードが判定する各ＣＧが再び加わったノードにコピーを有するので、ノードは、フラグ・テーブル１２０を調べることによって、ＣＧに関するチャンクのいずれかが設定されたＤＥＧＲＡＤＥＤまたはＤＩＲＴＹフラグを有するかどうかを判定する。

【０１０１】

ステップ９１０においてはいと判定すると、ステップ９１５において、ノード１０が、設定されたＤＥＧＲＡＤＥＤまたはＤＩＲＴＹフラグを有するチャンクのリスト１１５を計算し、再び加わったノードにリストを送信する。再び加わったノードは、ステップ９１５において、複数のノードからリストを受信する。ステップ９１５において各ノードによって送信されたリストは、各チャンクに関する情報を含む。情報は、フラグ・テーブル１２０から決定される。リストの情報は、たとえば、チャンクが設定されたＤＥＧＲＡＤＥＤフラグを有するかまたは設定されたＤＩＲＴＹフラグを有するか、そのイベント番号の後にチャンクへのＷＲＩＴＥオペレーションによってチャンクが書き込まれたイベント番号、およびそのイベント番号の後にＤＥＧＲＡＤＥＤフラグが設定されたイベント番号を

含む。イベントは、Ｐａｘｏｓからのメッセージであり、イベント番号は、クラスタにデバイスの障害を知らせるメッセージに関連するシーケンス番号である。したがって、イベントがディスクの障害である場合、メッセージは、ＷＲＩＴＥが実行された番号よりも大きなシーケンス番号を用いてＰａｘｏｓによって送信される。したがって、ＤＥＧＲＡＤＥＤ状態に関連するＰａｘｏｓのシーケンス番号は、システムがどのコピーが古いかを曖昧性なく判定することを可能にする。

【０１０２】

９１５におけるリストの受信の後、再び参加したノードが、リカバリされる必要があるチャンクを有する各ＣＧに関する現在のマップの下の第１のダウンしていないボリュームを決定する。ノードは、リストの情報に基づいて第１のダウンしていないボリュームを決定する。リストの情報から、ノードは、どのノードがチャンクの最も新しいコピーを記憶するか、およびどのコピーがＣＧに関するチャンクを再構築するのに十分であるかを判定し得る。ダウンしていないボリュームを決定すると、再び加わったノードは、リカバリされるかまたは再び書き込まれる必要がある各ノードに関してダウンしていないボリュームにＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を送信する（９２５）。ステップ９３０において、その他のノード上のチャンクのその他のコピーに基づいて、再び加わったノードが、データを受信し、チャンクを再び書き込むか、またはチャンクを再構築する。ステップ９３５において、再び加わったノードが、現在のマップの下でそのノードが記憶するＣＧに関してデータのすべてがリカバリされたかどうかを判定する。再び加わったノードは、データのすべてをリカバリしていない場合、すべてのチャンクが再構築されるまでＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を送信する。

【０１０３】

再び加わったノードは、ステップ９３５において、すべてのチャンクが再び書き込まれるかまたは再構築されると判定するとき、ステップ９４０において、情報のリストを送信した（９１４）すべてのノードにリカバリが完了していることを知らせるためにそれらのノードにメッセージを送信する。ステップ９４５において、すべてのノードが、チャンクのすべてのコピーが存在し、リカバリされるかどうかを判定する。すべてのチャンクは、チャンクのコピーを記憶する各ノードおよびディスクがオンラインである場合、ならびにすべてのチャンクのコピーが同じであり、そのＤＥＧＲＡＤＥＤ状態に関連する最も大きなイベント番号を有するチャンクと同一であると判定される場合に存在し、リカバリされる。すべてのチャンクが存在し、リカバリされる場合、再び加わったノードを含むすべてのノードが、フラグ・テーブル１２０の各チャンクに関するＤＥＧＲＡＤＥＤフラグをクリアする（ステップ９５５）。ステップ９４５において、ノードは、そのノードの判定中にそのノードのチャンクのいずれも存在せず、リカバリされないことを発見する場合、リカバリされる必要があるチャンクに関してダウンしていないボリュームを計算し、データをリカバリするためにダウンしていないボリュームにＲＥＣＯＶＥＲＹ＿ＲＥＡＤ要求を送信する。

【０１０４】

代替的な方法においては、すべてのノードが、ステップ９０５において、それらのノードが再び加わったノードにＣＧのコピーを記憶すると判定した後、各ノードは、影響を与えられるＣＧのリストおよび関連するメタデータを再び加わったノードに送信する。そして、再び加わったノードは、ＣＧのリストを計算し、ＣＧの独自のリストおよびメタデータを位置の受信されたリストのすべてのノードに送信する。ＣＧのリストおよびメタデータを受信した後、再び加わったノードを含む各ノードは、受信されたＣＧのメタデータを調べ、そのノードが記憶するＣＧのコピーがその他のコピーに存在する欠けているデータであるかどうか、またはノードのコピーがその他のコピーからのリカバリを必要としているかどうかを判定する。ノードのコピーがその他のコピーからのリカバリを必要としている場合、ノードは、図２４で説明されたルーチンのステップ９２０以降に従ってデータをリカバリする。

【０１０５】

本明細書において提供された以上の開示および教示から理解され得るように、本発明の実施形態は、ソフトウェアまたはハードウェアの制御論理またはこれらの組合せの形態で実装され得る。制御論理は、本発明の実施形態で開示された１組のステップを実行するように情報処理デバイスに指示するように適合された複数の命令として情報ストレージ媒体に記憶される可能性がある。本明細書において提供された開示および教示に基づいて、当業者は、本発明を実施するためのその他のやり方および／または方法を理解するであろう。

【０１０６】

上記の説明は、例示的であり、限定的でない。本発明の多くの変形形態が、本開示を検討すると当業者に明らかになるであろう。したがって、本発明の範囲は、上記の説明を参照して決定されるべきでなく、その代わりに、添付の請求項をそれらの請求項の完全な範囲およびそれらの均等物と一緒に参照して決定されるべきである。

【図１】

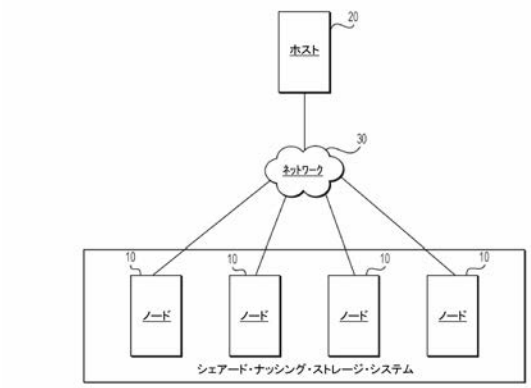


FIG. 1

【図２】

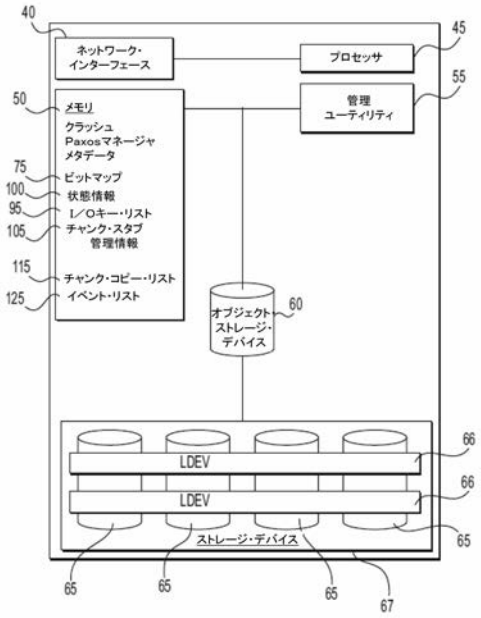


FIG. 2

【図３】

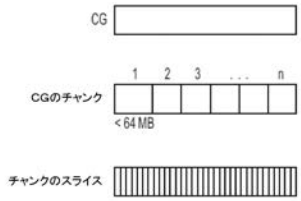


FIG. 3

【図４】

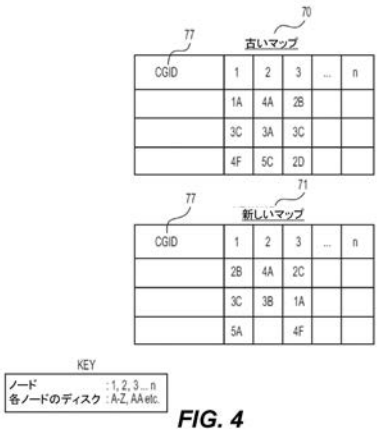


FIG. 4

【図５】

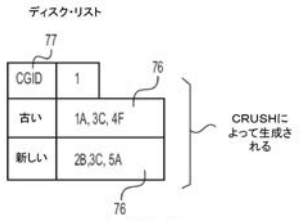
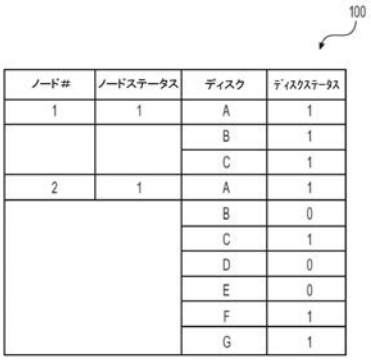


FIG. 5

【図６】



ノード#	ノードステータス	ディスク	ディスクステータス
1	1	A	1
		B	1
		C	1
2	1	A	1
		B	0
		C	1
		D	0
		E	0
		F	1
		G	1

FIG. 6

【図７】

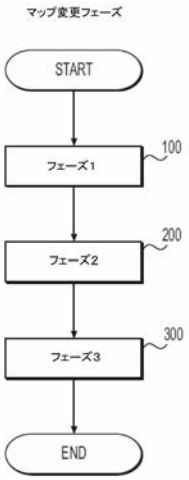


FIG. 7

【図８】

FIG. 8

【図９】

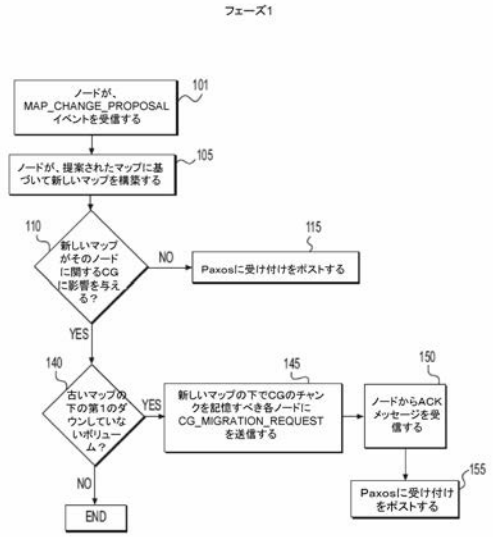


FIG. 9

【図１０】

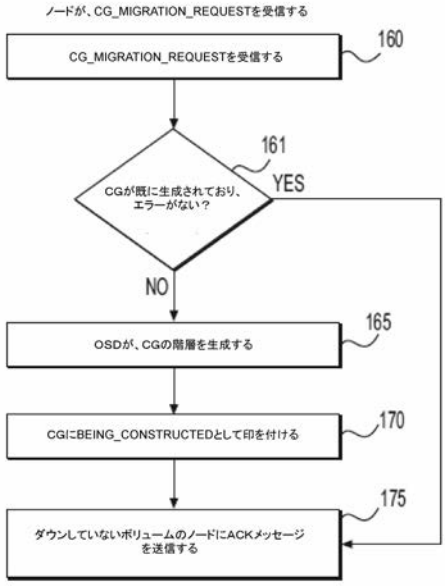


FIG. 10

【図１１】

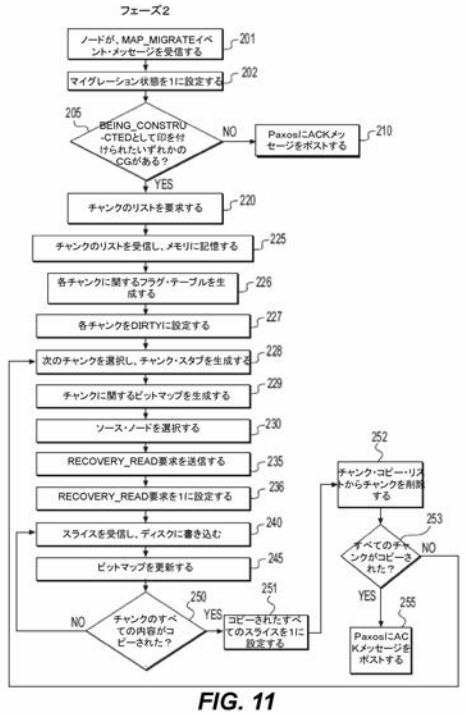


FIG. 11

【図１２】

フラグ・テーブル

	CHUNK 1		CHUNK 2		120
	STATUS	DEGRADEDフラグまたはWRITEの後のイベント番号	STATUS	DEGRADEDフラグまたはWRITEの後のイベント番号	
CLEAN	0		1		
DIRTY	1		0		
DEGRADED	1	3	1	2	
WRITE		2		1	

FIG. 12

【図１３】

チャンク・スタブ管理情報

106	チャンク番号	8	105
107	CG ID	4	
108	古いマップの位置	2A	
109	RECOVERY_READ要求送信済み	0	
110	全スライス・コピー済み	0	
111	I/Oキー・リスト空	0	

FIG. 13

【図１４】

FIG. 14

【図１５】

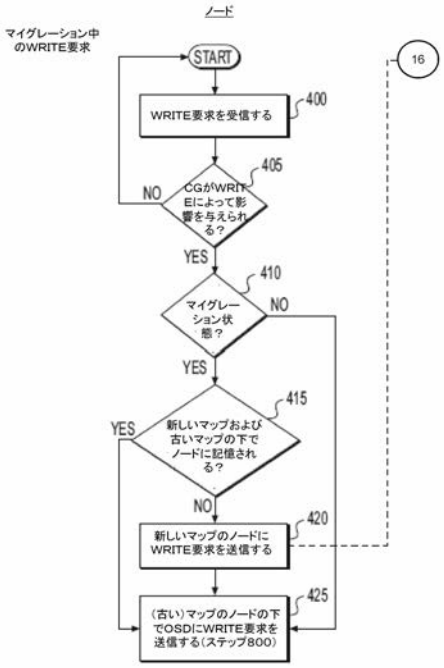


FIG. 15

【図１６】

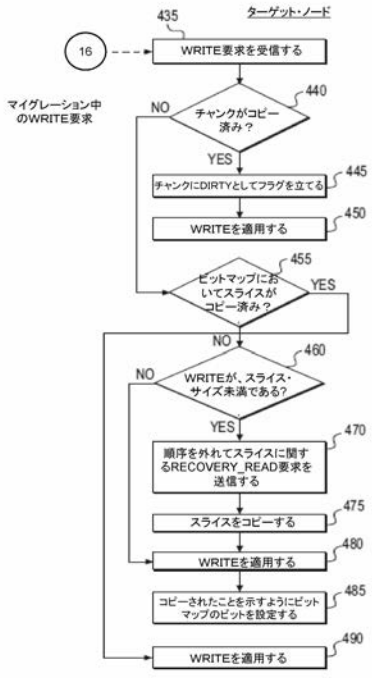


FIG. 16

【図１７】

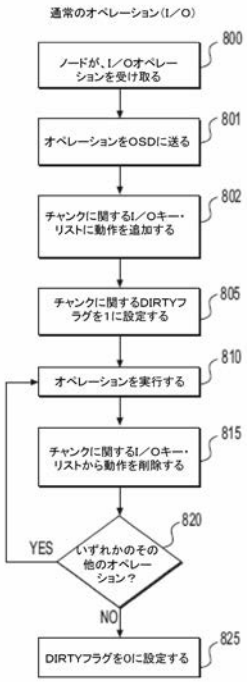


FIG. 17

【図１８】

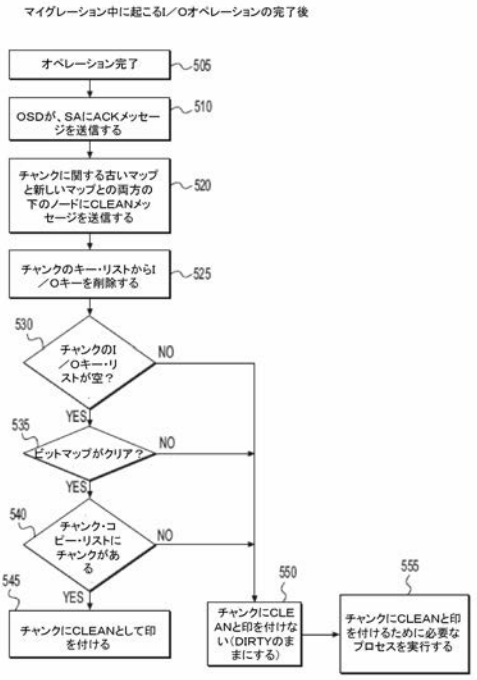


FIG. 18

【図１９】

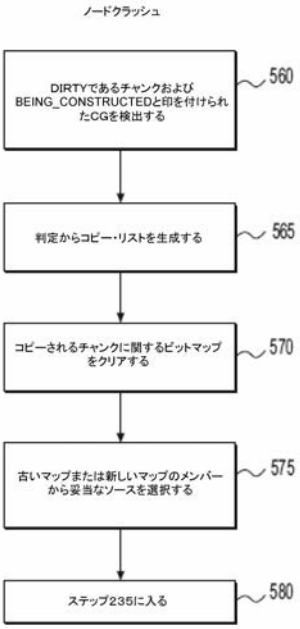


FIG. 19

【図２０】

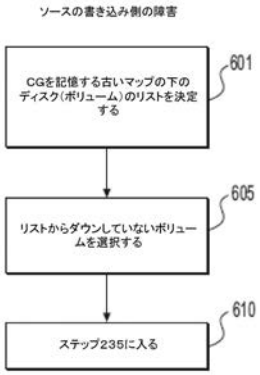


FIG. 20

【図２１】

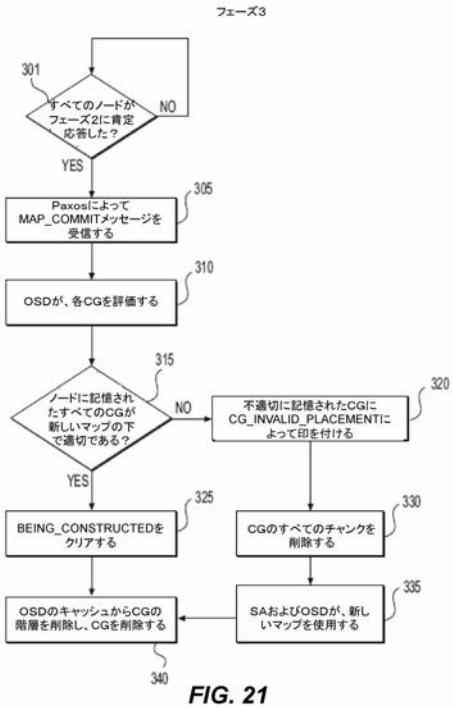


FIG. 21

【図２２】

FIG. 22

【図２３】

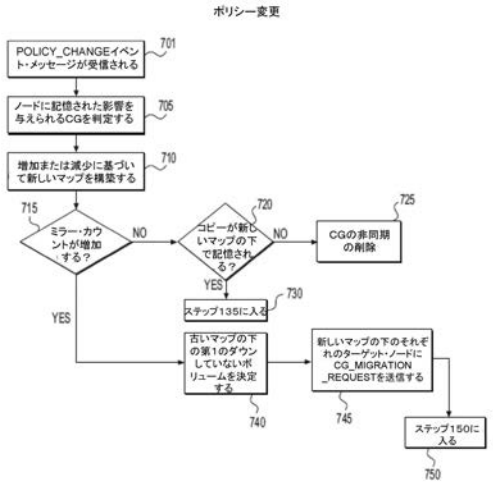


FIG. 23

【図２４】

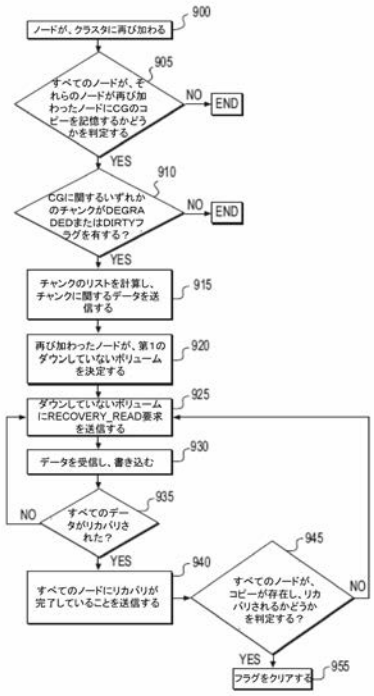


FIG. 24

【図２５】

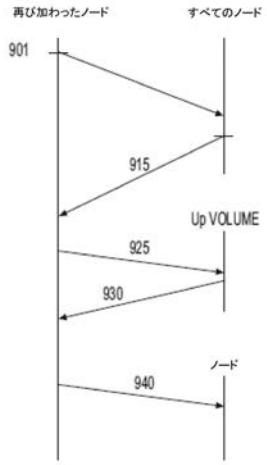


FIG. 25

【図２６】

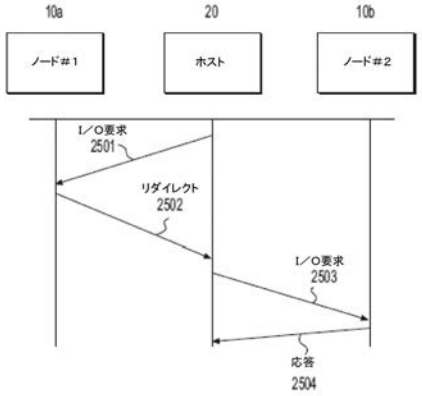


FIG. 26

【図２７】

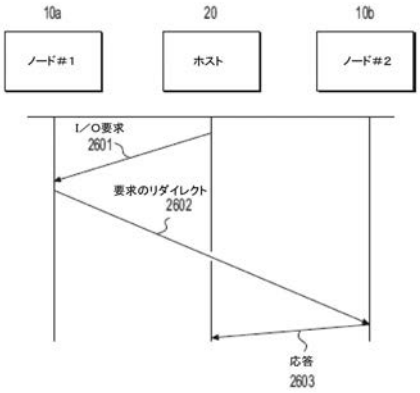


FIG. 27

【手続補正書】
【提出日】平成28年3月2日(2016.3.2)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
１以上のネットワークに接続された複数のノードを備え、
各ノードが、Ｉ／Ｏ（Input/Output）オペレーションを送受信可能であり、
各ノードが、データ及びデータレプリカを記憶するための１以上のストレージ・デバイスを備え、
各ノードが、データ及びデータレプリカの位置を示す位置情報に基づいて、前記複数のノードのストレージ・デバイスに記憶されたデータ及びデータレプリカの位置を特定できるようになっており、
各ノードと各ノードの各ストレージ・デバイスとの状態を示す状態情報が、
前記複数のノードのそれぞれに提供され、
ノード又はそのノードのストレージ・デバイスがアクティブ化されるか又は非アクティブ化されると、そのアクティブ化又は非アクティブ化に基づいて、変更され、
前記アクティブ化が、新しくアクティブ化されたノード又は新しくアクティブ化されたストレージ・デバイスを有するノードを含む場合、各ノードが、前記状態情報の変更に基づいて、前記複数のノードのストレージ・デバイスに記憶されたデータ又はデータレプリカの整合性を維持するためにリカバリオペレーションが必要とされるかどうかを判定し、
必要があれば、各ノードが、影響を与えられるデータを記憶する他ノードとの間でメッ

セージを送受信することによって、そのノードのストレージ・デバイスに記憶されたデータを復元するために、他ノードとは独立して前記リカバリオペレーションを実行する、
ストレージ・システム。
【請求項２】
データが、複数のノードにデータレプリカ又は消失訂正符号化セグメントとして冗長的に記憶され、
データレプリカ又は消失訂正符号化セグメントの数が、前記データの冗長性のポリシーに依存し、
データレプリカ又は消失訂正符号化セグメントの符号化が、前記データの前記冗長性のポリシー及び消失訂正符号化アルゴリズムに依存する
請求項１に記載のストレージ・システム。
【請求項３】
各ノードが、イベントの通知を受信するようになっており、
前記イベントが、ノード又はそのノードのストレージ・デバイスのアクティブ化又は非アクティブ化である
請求項２に記載のストレージ・システム。
【請求項４】
各イベントが、所定の順番で各ノードに届けられ、それら１以上のイベントが、一意で整合性のある番号である昇順のイベント番号を振られ、
各ノードが、前記１以上のイベントのそれぞれを受信するようになっており、
非アクティブ化の後でアクティブ化された各ノードが、前記ノードがアクティブ化されるときに前記イベントのそれぞれを受信する、
請求項３に記載のストレージ・システム。
【請求項５】
前記状態情報が、
各ノードのストレージ・デバイスに記憶されたデータと各ノードのストレージ・デバイス上のデータレプリカ又は消失訂正符号化セグメントとの状態、及び、
前記他ノードのそれぞれのストレージ・デバイス上のデータとデータレプリカ又は消失訂正符号化セグメントとの状態
を表す情報を含み、
データ、データレプリカ、又は消失訂正符号化セグメントの前記状態が、
データを修正又は書き込みする前に、データ、データレプリカ、又は消失訂正符号セグメントに関して設定されるＤＩＲＴＹフラグ、
データの修正又は書き込みがデータレプリカ又は消失訂正符号化セグメントの一部で成功したことを前記ノードが知っているときに、データ、データレプリカ、又は消失訂正符号化セグメントに関して設定されるＤＥＧＲＡＤＥＤフラグ、
イベント後にデータが書き込まれたそのイベントの番号を示す情報、及び
イベント後にＤＥＧＲＡＤＥＤフラグが設定されたそのイベントの番号を示す情報、
を含み、
前記ＤＩＲＴＹフラグ及び前記ＤＥＧＲＡＤＥＤフラグは、いずれも永続的であり、
前記ＤＩＲＴＹフラグが、前記データのデータレプリカ又は消失訂正符号化セグメントの少なくとも予め決められた最小限の数に対する書き込みが正常に完了したときにクリアされ、
前記データと前記データレプリカ又は前記消失訂正符号化セグメントとの位置が、永続的に記憶されることと、前記データの位置情報及びメタデータを基に算出されることのうちのいずれかが可能であり、
前記メタデータが、一意のＩＤ、前記データの名前空間内の一意の名前属性、及び、前記データの内容の記述、のうちの少なくとも１つを含む
請求項３又は４に記載のストレージ・システム。
【請求項６】

各ノードによって受信される前記イベントが、非アクティブ化の後にアクティブ化された第１ノードのアクティブ化又は前記第１ノードのストレージ・デバイスのアクティブ化であり、
前記判定が、前記位置情報に基づいて前記ノードのうちの１つが前記第１ノードに記憶されたデータ、データレプリカ、又は消失訂正符号化セグメントを有するかどうかに基づき、
前記第１ノードが、前記ＤＥＧＲＡＤＥＤフラグと、前記ＤＩＲＴＹフラグと、前記データの最後の修正又は書き込みのイベント番号を示す情報と、イベント後に前記ＤＥＧＲＡＤＥＤフラグが設定されたそのイベントの番号を示す情報とを含む情報のリストを受信し、
前記リストに基づいて、前記第１ノードが、どのデータ、データレプリカ、又は消失訂正符号化セグメントが再構築される必要があるかを判定し、
前記状態情報に基づいて、前記第１ノードが、前記第１ノードのストレージ・デバイス上のデータ、データレプリカ、又は消失訂正符号化セグメントを再構築するために再構築される必要がある前記第１ノード上のデータのデータレプリカ又は前記消失訂正符号化セグメントをどのノードが記憶するかを判定し、
前記第１ノードが、判定されたノードからの再構築されるデータ、データレプリカ、又は消失訂正符号化セグメントの内容を要求し、再構築すべきデータの内容を書き込み、完了すると、前記リストを送信した前記複数のノードのそれぞれにメッセージを送信し、
前記データのデータレプリカ又は消失訂正符号化セグメントのそれぞれが各ノードでリカバリされると判定される場合に、前記複数のノードのそれぞれが、前記データと、前記データレプリカと、前記消失訂正符号化セグメントとのいずれかの前記ＤＥＧＲＡＤＥＤフラグをクリアする、
請求項５に記載のストレージ・システム。
【請求項７】
請求項１乃至６のうちのいずれか１項に記載のストレージ・システムにおけるノードであって、
データ及びデータレプリカを記憶するための１以上のストレージ・デバイスと、
前記１以上のストレージ・デバイスに接続され、Ｉ／Ｏ（Input/Output）オペレーションを送受信でき、データ及びデータレプリカの位置を示す位置情報に基づいてデータ及びデータレプリカの位置を特定できる制御手段と
を備え、
前記ストレージ・システムの各ノードと各ノードの各ストレージ・デバイスとの状態を示す状態情報が、
前記複数のノードのそれぞれに提供され、
ノード又はそのノードのストレージ・デバイスがアクティブ化されるか又は非アクティブ化されると、そのアクティブ化又は非アクティブ化に基づいて、変更され、
前記アクティブ化が、新しくアクティブ化されたノード又は新しくアクティブ化されたストレージ・デバイスを有するノードを含む場合、そのノードの前記制御手段が、前記状態情報の変更に基づいて、前記複数のノードのストレージ・デバイスに記憶されたデータ又はデータレプリカの整合性を維持するためにリカバリオペレーションが必要とされるかどうかを判定し、
必要があれば、前記制御手段が、影響を与えられるデータを記憶する他ノードとの間でメッセージを送受信することによって、そのノードのストレージ・デバイスに記憶されたデータを復元するために、他ノードとは独立して前記リカバリオペレーションを実行する、
ノード。
【請求項８】
Ｉ／Ｏ（Input/Output）オペレーションを送受信できデータ及びデータレプリカの位置を示す位置情報に基づいてデータ及びデータレプリカの位置を特定できる各ノードと、各

ノードの各ストレージ・デバイスとの状態を示す状態情報を、複数のノードのそれぞれに提供するステップと、
ノード又はそのノードのストレージ・デバイスがアクティブ化されるか又は非アクティブ化されると、そのアクティブ化又は非アクティブ化に基づいて、前記状態情報を変更するステップと、
前記アクティブ化が、新しくアクティブ化されたノード又は新しくアクティブ化されたストレージ・デバイスを有するノードを含む場合、前記状態情報の変更に基づいて、前記複数のノードのストレージ・デバイスに記憶されたデータ又はデータレプリカの整合性を維持するためにリカバリオペレーションが必要とされるかどうかを判定するステップと、
必要があれば、影響を与えられるデータを記憶する他ノードとの間でメッセージを送受信することによって、ストレージ・デバイスに記憶されたデータを復元するために、他ノードとは独立して前記リカバリオペレーションを実行するステップと
をコンピュータに実行させるコンピュータ・プログラム。
【請求項９】
データが、複数のノードにデータレプリカ又は消失訂正符号化セグメントとして冗長的に記憶され、
データレプリカ又は消失訂正符号化セグメントの数が、前記データの冗長性のポリシーに依存し、
データレプリカ又は消失訂正符号化セグメントの符号化が、前記データの前記冗長性のポリシー及び消失訂正符号化アルゴリズムに依存する
請求項８に記載のコンピュータ・プログラム。
【請求項１０】
各ノードが、イベントの通知を受信するようになっており、
前記イベントが、ノード又はそのノードのストレージ・デバイスのアクティブ化又は非アクティブ化である
請求項９に記載のコンピュータ・プログラム。
【請求項１１】
各イベントが、所定の順番で各ノードに届けられ、それら１以上のイベントが、一意で整合性のある番号である昇順のイベント番号を振られ、
各ノードが、前記イベントのそれぞれを受信するようになっており、
非アクティブ化の後でアクティブ化された各ノードが、前記ノードがアクティブ化されるときに前記イベントのそれぞれを受信する
請求項１０に記載のコンピュータ・プログラム。
【請求項１２】
前記状態情報が、
各ノードのストレージ・デバイスに記憶されたデータと各ノードのストレージ・デバイス上のデータレプリカ又は消失訂正符号化セグメントとの状態、及び、
前記他ノードのそれぞれのストレージ・デバイス上のデータとデータレプリカ又は消失訂正符号化セグメントとの状態
を表す情報を含み、
データ、データレプリカ、又は消失訂正符号化セグメントの前記状態が、
データを修正又は書き込みする前に、データ、データレプリカ、又は消失訂正符号セグメントに関して設定されるＤＩＲＴＹフラグ、
データの修正又は書き込みがデータレプリカ又は消失訂正符号化セグメントの一部で成功したことを前記ノードが知っているときに、データ、データレプリカ、又は消失訂正符号化セグメントに関して設定されるＤＥＧＲＡＤＥＤフラグ、
イベント後にデータが書き込まれたそのイベントの番号を示す情報、及び
イベント後にＤＥＧＲＡＤＥＤフラグが設定されたそのイベントの番号を示す情報、
を含み、
前記ＤＩＲＴＹフラグ及び前記ＤＥＧＲＡＤＥＤフラグは、いずれも永続的であり、

前記ＤＩＲＴＹフラグが、前記データのデータレプリカ又は消失訂正符号化セグメントの少なくとも予め決められた最小限の数に対する書き込みが正常に完了したときにクリアされ、
前記データと前記データレプリカ又は前記消失訂正符号化セグメントとの位置が、永続的に記憶されることと、前記データの位置情報及びメタデータを基に算出されることのうちのいずれかが可能であり、
前記メタデータが、一意のＩＤ、前記データの名前空間内の一意の名前属性、及び、前記データの内容の記述、のうちの少なくとも１つを含む
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】
各ノードによって受信される前記イベントが、非アクティブ化の後にアクティブ化された第１ノードのアクティブ化又は前記第１ノードのストレージ・デバイスのアクティブ化であり、
前記判定が、前記位置情報に基づいて前記ノードのうちの１つが前記第１ノードに記憶されたデータ、データレプリカ、又は消失訂正符号化セグメントを有するかどうかに基づき、
前記第１ノードが、前記ＤＥＧＲＡＤＥＤフラグと、前記ＤＩＲＴＹフラグと、前記データの最後の修正又は書き込みのイベント番号を示す情報と、イベント後に前記ＤＥＧＲＡＤＥＤフラグが設定されたそのイベントの番号を示す情報とを含む情報のリストを受信し、
前記リストに基づいて、前記第１ノードが、どのデータ、データレプリカ、又は消失訂正符号化セグメントが再構築される必要があるかを判定し、
前記状態情報に基づいて、前記第１ノードが、前記第１ノードのストレージ・デバイス上のデータ、データレプリカ、又は消失訂正符号化セグメントを再構築するために再構築される必要がある前記第１ノード上のデータのデータレプリカ又は前記消失訂正符号化セグメントをどのノードが記憶するかを判定し、
前記第１ノードが、判定されたノードからの再構築されるデータ、データレプリカ、又は消失訂正符号化セグメントの内容を要求し、再構築すべきデータの内容を書き込み、完了すると、前記リストを送信した前記複数のノードのそれぞれにメッセージを送信し、
前記データのデータレプリカ又は消失訂正符号化セグメントのそれぞれが各ノードでリカバリされると判定される場合に、前記複数のノードのそれぞれが、前記データと、前記データレプリカと、前記消失訂正符号化セグメントとのいずれかの前記ＤＥＧＲＡＤＥＤフラグをクリアする、
請求項１２に記載のコンピュータ・プログラム製品。
【請求項１４】
Ｉ／Ｏ（Input/Output）オペレーションを送受信できデータ及びデータレプリカの位置を示す位置情報に基づいてデータ及びデータレプリカの位置を特定できる各ノードと、各ノードの各ストレージ・デバイスとの状態を示す状態情報を、複数のノードのそれぞれに提供するステップと、
ノード又はそのノードのストレージ・デバイスがアクティブ化されるか又は非アクティブ化されると、そのアクティブ化又は非アクティブ化に基づいて、前記状態情報を変更するステップと、
前記アクティブ化が、新しくアクティブ化されたノード又は新しくアクティブ化されたストレージ・デバイスを有するノードを含む場合、前記状態情報の変更に基づいて、前記複数のノードのストレージ・デバイスに記憶されたデータ又はデータレプリカの整合性を維持するためにリカバリオペレーションが必要とされるかどうかを判定するステップと、
必要があれば、影響を与えられるデータを記憶する他ノードとの間でメッセージを送受信することによって、ストレージ・デバイスに記憶されたデータを復元するために、他ノードとは独立して前記リカバリオペレーションを実行するステップと
を有する方法。

【請求項１５】
請求項８乃至１３のうちのいずれか１項に記載のコンピュータ・プログラムに基づいて実行される請求項１４に記載の方法。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/065623

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/14 G06F11/20
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

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, IBM-TDB, COMPENDEX, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 8 401 997 B1 (TAWRI DEEPAK [US] ET AL) 19 March 2013 (2013-03-19) column 2, line 7 - line 67 column 3, line 55 - column 6, line 41 column 10, line 29 - column 12, line 24 column 16, line 60 - column 21, line 24 figures 1,4-6,8 -----	1-20
A	US 6 907 507 B1 (KISELEV OLEG [US] ET AL) 14 June 2005 (2005-06-14) column 1, line 7 - column 4, line 58 ----- -/--	1-6

[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
14 January 2015	29/01/2015

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	Johansson, Ulf

2

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/065623

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 7 617 259 B1 (MUTH JOHN A [US] ET AL) 10 November 2009 (2009-11-10) column 1, line 24 - column 2, line 53 column 3, line 48 - column 4, line 29 column 5, line 11 - line 40 column 6, line 58 - column 12, line 58 column 13, line 48 - column 15, line 13 figures 1,2a,3-8	7-14
A	SAGE A WEIL ET AL: "CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data", SUPERCOMPUTING, 2006. SC '06. PROCEEDINGS OF THE ACM/IEEE SC 2006 CONFERENCE, IEEE, PI, 1 November 2006 (2006-11-01), pages 31-31, XP031044223, ISBN: 978-0-7695-2700-0 page 1, line 1 - page 4, line 1 page 5, line 14 - page 6, line 21 page 10, line 1 - page 11, line 3	15-20

2

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

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/065623

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☒ No protest accompanied the payment of additional search fees.

International Application No. PCT/US2013/065623

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-6

Distributed shared-nothing storage system, comprising: a plurality of independent nodes in a cluster connected to one or more networks, each comprising one or more devices for storing data; each independent node is configured to send and receive input/output (I/O) operations; the data stored on the plurality of nodes in the cluster includes replicas of the data, each node is aware of the location of the data and data replicas stored on the storage devices in the cluster based on a location map; and state information, which indicates the status of each node and each storage device on each node, which is available to all nodes in the cluster, upon activation or inactivation of a node or activation or inactivation of a storage device on any node in the cluster, the state information is updated and changed based on the corresponding activation or inactivation, based on the change in the state information, each node in the cluster, including the newly activated node or node that has a newly activated storage device, determines if it must perform a recovery operation to maintain the consistency of any data or data replicas stored on the nodes in the cluster, and each node performs the recovery operation, if needed, independently from the other nodes in the cluster to restore the data stored on its storage devices by sending and receiving messages to other nodes in the cluster that store affected data.

---

2. claims: 7-14

Distributed shared-nothing storage system, comprising: a plurality of independent nodes in a cluster connected to a network, each comprising one or more storage devices for storing data; each node is configured to send and receive input/output (I/O) operations; and each node has memory for conducting local processing and is separate from the storage devices, wherein)/0 operations are performed on individual nodes and each node that receives an)/0 operation performs the I/O operation only for the data stored on its storage devices, wherein a location of data stored on the storage devices in the cluster is derived from layout information available to all the nodes, wherein the location of the data is stored and calculated by each node, as needed, from the layout information, and wherein, upon receiving a first event message indicating the occurrence of an event, each node evaluates all data stored on its storage devices and determines if any data is affected by information in the first event message.

---

3. claims: 15-20

International Application No. PCT/US2013/065623

Distributed shared-nothing storage system, comprising: a plurality of independent nodes in a cluster connected to a network, each comprising one or more storage devices for storing data; each node is configured to send and receive input/output (1/0) operations; and state information available to all nodes which indicates whether each node and each storage device on each node is in an active state, wherein, data in the cluster is redundantly stored as replicas or erasure-coded segments on a plurality of nodes in the cluster, wherein the number of replicas or erasure-coded segments depends on the redundancy policy of the data, wherein the encoding of replicas or erasure-coded segments depends on the redundancy policy and erasure-coding algorithm of the data, wherein a location of the data and replicas or erasure-coded segments of the data on the storage devices of the nodes in the cluster is derived from layout information available to all the nodes, wherein a location of the data is stored and calculated as needed from the layout information, and wherein a redundancy policy for the storage system sets a mirror count number indicating the number of data replicas or erasure-coded segments of the data, data replicas or erasure-coded segments are migrated from a first location to a second location on the storage devices in the cluster or data, data replicas, or erasure-coded segments are removed from the cluster based on a change in the redundancy policy of the storage system.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2013/065623

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8401997 B1	19-03-2013	NONE	
US 6907507 B1	14-06-2005	US 6907507 B1	14-06-2005
		US 7089385 B1	08-08-2006
US 7617259 B1	10-11-2009	NONE	

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

フロントページの続き

(51)Int.Cl.	FI	テーマコード（参考）
	G06F 12/00 531R	

(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,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,LT,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 キセレヴ，オレグ
アメリカ合衆国 ９４３０１ カリフォルニア州，パロ アルト，ブライアント ストリート ＃３３３ ５５５
(72)発明者 ポール，ガウラブ
アメリカ合衆国 ９５０１４ カリフォルニア州，クパチーノ，アパートメント ディー， サウス ブラニー アベニュー １０１０９
(72)発明者 ヤングワース，クリストファー
アメリカ合衆国 ９５１２０ カリフォルニア州，サンノゼ，ガダラジャラ コート １６０４

Ｆターム(参考) 5B027 CC04