(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7253007号
(P7253007)

(45)発行日　令和5年4月5日(2023.4.5)

(24)登録日　令和5年3月28日(2023.3.28)

(51)国際特許分類

		FI		
G06F 3/06 (2006.01)		G06F	3/06	301Z
G06F 11/07 (2006.01)		G06F	3/06	301X
G06F 11/34 (2006.01)		G06F	11/07	140M
G06F 13/10 (2006.01)		G06F	11/07	193
		G06F	11/34	133

請求項の数　15　（全46頁）　最終頁に続く

(21)出願番号　特願2021-90224(P2021-90224)
(22)出願日　令和3年5月28日(2021.5.28)
(65)公開番号　特開2022-182584(P2022-182584A)
(43)公開日　令和4年12月8日(2022.12.8)
審査請求日　令和3年12月23日(2021.12.23)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　110002365
弁理士法人サンネクスト国際特許事務所
(72)発明者　山本 貴大
東京都千代田区丸の内一丁目６番６号
株式会社日立製作所内
(72)発明者　坂下 悠貴
東京都千代田区丸の内一丁目６番６号
株式会社日立製作所内
(72)発明者　伊藤 晋太郎
東京都千代田区丸の内一丁目６番６号
株式会社日立製作所内
(72)発明者　揚妻 匡邦

最終頁に続く

(54)【発明の名称】　ストレージシステム

(57)【特許請求の範囲】

【請求項１】

複数の領域を含むボリュームを１以上のホストに提供するための処理を行うプロセッサを備える複数のノードと、前記プロセッサと接続され、前記ボリュームのデータを記憶する１以上の記憶デバイスとを備えるストレージシステムであって、

前記複数のノードの各々は、自ノードが提供するボリュームの負荷および前記ボリュームの領域を複数に分割した領域の負荷を監視し、

監視している一のボリュームの負荷が閾値以上であると判定した第１のノードは、前記一のボリュームの領域を複数に分割した領域の負荷と負荷分散のポリシとに応じて、前記一のボリュームに含まれる一部の領域を前記第１のノードとは異なる第２のノードのボリュームに移動する、

ストレージシステム。

【請求項２】

前記第１のノードは、前記一のボリュームの負荷が閾値未満であると判定した場合、前記第２のノードのボリュームから、前記一部の領域を前記一のボリュームに移動する、

請求項１に記載のストレージシステム。

【請求項３】

前記第１のノードは、前記第２のノードとして、前記一部の領域を移動した後のボリュームの負荷が閾値を超えないノードを選択する、

請求項１に記載のストレージシステム。

【請求項４】
前記複数のノードの各々には、前記１以上の記憶デバイスの少なくとも１つが対応して設けられ、
前記複数のノードの各々は、自ノードに割り当てられている領域のデータを、自ノードに設けられている記憶デバイスに記憶し、
前記第１のノードは、前記一のボリュームの容量が提供可能な容量を超えると判定した場合、前記一部の領域を前記第２のノードのボリュームに移動する、
請求項１に記載のストレージシステム。
【請求項５】
前記複数のノードの各々は、自ノードが提供するボリュームに対するリードの負荷と、前記ボリュームに対するライトの負荷とを監視し、
前記第１のノードは、前記一のボリュームに対するリードの負荷が第１の閾値以上であると判定した場合、前記一部の領域を前記第２のノードのボリュームに移動し、前記一のボリュームに対するライトの負荷が前記第１の閾値とは異なる第２の閾値以上であると判定した場合、前記一部の領域を前記第２のノードのボリュームに移動する、
請求項１に記載のストレージシステム。
【請求項６】
前記第１のノードは、前記複数のノードに対して前記一のボリュームに含まれる領域を均等に割り振り、前記第１のノードとは異なるノードに割り振った領域を、前記ノードのボリュームに移動する、
請求項１に記載のストレージシステム。
【請求項７】
前記第１のノードは、前記一のボリュームの負荷が前記閾値を下回るまで、前記一のボリュームに含まれる領域を１つずつ、前記第１のノードとは異なるノードのボリュームに移動する、
請求項１に記載のストレージシステム。
【請求項８】
前記一のボリュームが、複数のホストに提供されている場合、前記第１のノードは、前記複数のホストの各々がアクセスする領域をまとめてホストごとの移動対象とし、ホストごとの移動対象の領域を、前記第１のノードとは異なるノードのボリュームに移動する、
請求項１に記載のストレージシステム。
【請求項９】
前記第１のノードは、自ノードが提供しているボリュームのうち前記一のボリュームとは異なる他のボリュームを前記第１のノードとは異なるノードに移動し、前記第２のノードのボリュームから、前記一部の領域を前記一のボリュームに移動する、
請求項１に記載のストレージシステム。
【請求項１０】
前記一部の領域が移動された前記第２のノードのボリュームと、前記一部の領域にアクセスするホストとの間にパスが設定されていない場合、前記第２のノードおよび前記ホストは、前記パスを設定する、
請求項１に記載のストレージシステム。
【請求項１１】
前記複数のノードの各々は、自ノードのボリュームの負荷が特定のホストに偏っていると判定した場合、前記特定のホストとのパスが最適属性であることを前記特定のホストに通知する、
請求項１に記載のストレージシステム。
【請求項１２】
前記複数のノードの各々は、自ノードのボリュームの負荷が前記特定のホストに偏っていないと判定した場合、前記ボリュームに定義されている全てのパスが最適属性であることを前記パスが設定されている全てのホストに通知する、

請求項１１に記載のストレージシステム。
【請求項１３】
前記１以上の記憶デバイスは、前記複数のノードに共通して設けられている、
請求項１に記載のストレージシステム。
【請求項１４】
前記第１のノードは、前記第２のノードに前記一部の領域を移動する際、前記一部の領域を管理するためのデータを更新し、前記１以上の記憶デバイスに記憶される前記一部の領域のデータを移動しない、
請求項１３に記載のストレージシステム。
【請求項１５】
前記複数のノードが提供するボリュームについて、当該ボリュームの領域の移動先ノードの数に応じたスループットと応答時間とを計算して出力する計算機を備える、
請求項１に記載の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、概して、ボリュームの負荷を分散する技術に関する。
【背景技術】
【０００２】
特許文献１は、複数のサーバをネットワークで接続して、各サーバのローカルストレージをストレージ制御ソフトウェアにより統合して、１つのストレージプールとして提供し、当該ストレージプールからボリュームを提供するストレージシステムについて開示している。上記ストレージシステムは、当該ボリュームにデータを書き込む際に、異なるサーバに格納されるデータを組み合わせ、パリティを計算し、データとは異なるサーバに格納することで、サーバ障害からデータを保護する。上記ストレージシステムでは、サーバの追加により、ストレージの容量と性能とをスケールアウトできることが特徴となっている。
【０００３】
また、特許文献１は、ボリュームへ書き込まれたデータについてアクセス頻度を採取し、アクセス頻度の高いデータは、ローカルストレージに格納し、アクセス頻度の低いデータは異なるサーバのストレージ（ここでは、リモートストレージと呼ぶ）に格納するようにデータ配置を変更する技術についても開示している。ホストがボリュームのデータにアクセスする際は、ボリュームのローカルストレージを割り当てたサーバにアクセスし、当該サーバにて、データがローカルストレージにあるか、リモートストレージにあるかを判定し、リモートストレージにある場合は、異なるサーバにアクセスを転送し、データへのアクセスを行う。このように、アクセス頻度の高いデータをローカルストレージに格納しておくことで、ネットワークを介することなくデータへアクセスできるため、ホストに対して高速に応答することができる。
【先行技術文献】
【特許文献】
【０００４】
【文献】米国特許出願公開第２０１６／０３７１１４５号明細書
【発明の概要】
【発明が解決しようとする課題】
【０００５】
特許文献１に記載の技術に基づいてホストにボリュームを提供する場合、ボリュームへのアクセスは、必ずローカルストレージを有するサーバを経由してからアクセスする。このため、１つのボリュームの性能がローカルストレージを有するサーバの性能が上限となる。上記ストレージシステムは、サーバの追加により、性能がスケールアウトすることが特徴である。特許文献１で開示されている技術では、システム性能（複数のボリュームの合計性能）は、サーバを追加することで、スケールアウトできるが、１つのボリュームの

性能は、サーバを追加してもスケールアウトすることができない。
【０００６】
本発明は、以上の点を考慮してなされたもので、サーバ（ノード）を追加したときにシステム性能がスケールアウトすると共に、１つのボリュームについても性能をスケールアウトし得るストレージシステム等を提案しようとするものである。
【課題を解決するための手段】
【０００７】
かかる課題を解決するため本発明においては、複数の領域を含むボリュームを１以上のホストに提供するための処理を行うプロセッサを備える複数のノードと、前記プロセッサと接続され、前記ボリュームのデータを記憶する１以上の記憶デバイスとを備えるストレージシステムであって、前記複数のノードの各々は、自ノードが提供するボリュームの負荷および前記ボリュームの領域を複数に分割した領域の負荷を監視し、監視している一のボリュームの負荷が閾値以上であると判定した第１のノードは、前記一のボリュームの領域を複数に分割した領域の負荷と負荷分散のポリシとに応じて、前記一のボリュームに含まれる一部の領域を前記第１のノードとは異なる第２のノードのボリュームに移動するようにした。
【０００８】
上記構成では、一のボリュームの負荷が高まったときに、当該ボリュームの一部の領域が他のノードのボリュームに移動されるので、例えば、ノードを追加した場合、一のボリュームについても性能をスケールアウトすることができるようになる。
【発明の効果】
【０００９】
本発明によれば、一のボリュームの性能をスケールアウトし得るストレージシステムを実現することができる。
【図面の簡単な説明】
【００１０】
【図１】第１の実施の形態によるボリュームのデータの割り当て変更の概要を示すイメージ図である。
【図２】第１の実施の形態によるストレージシステムに係る物理構成の一例を示す図である。
【図３】第１の実施の形態によるストレージシステムに係る論理構成の一例を示す図である。
【図４】第１の実施の形態によるメモリ内の情報の一例を示す図である。
【図５】第１の実施の形態によるクラスタ管理テーブルの一例を示す図である。
【図６】第１の実施の形態によるデータ保護セット管理テーブルの一例を示す図である。
【図７】第１の実施の形態によるストレージプール管理テーブルの一例を示す図である。
【図８】第１の実施の形態によるボリューム管理テーブルの一例を示す図である。
【図９】第１の実施の形態によるモニタ情報管理テーブルの一例を示す図である。
【図１０】第１の実施の形態によるフロントエンドパス管理テーブルの一例を示す図である。
【図１１】第１の実施の形態によるリード処理に係るフローチャートの一例を示す図である。
【図１２】第１の実施の形態によるライト処理に係るフローチャートの一例を示す図である。
【図１３Ａ】第１の実施の形態によるモニタ情報採取処理に係るフローチャートの一例を示す図である。
【図１３Ｂ】第１の実施の形態によるモニタ情報採取処理に係るフローチャートの一例を示す図である。
【図１４】第１の実施の形態によるリバランス要否判定処理に係るフローチャートの一例を示す図である。

【図１５】第１の実施の形態によるリソース割当決定処理に係るフローチャートの一例を示す図である。
【図１６】第１の実施の形態によるリソース割当決定処理に係るフローチャートの一例を示す図である。
【図１７】第１の実施の形態によるリソース移動処理に係るフローチャートの一例を示す図である。
【図１８】第１の実施の形態によるフロントエンドパス設定処理に係るフローチャートの一例を示す図である。
【図１９】第１の実施の形態によるフロントエンドパス設定処理に係るフローチャートの一例を示す図である。
【図２０】第１の実施の形態によるクラスタ構成変更処理に係るフローチャートの一例を示す図である。
【図２１Ａ】第１の実施の形態によるＧＵＩの一例を示す図である。
【図２１Ｂ】第１の実施の形態によるＧＵＩの一例を示す図である。
【図２２】第２の実施の形態によるストレージシステムに係る構成の一例を示す図である。
【図２３】第２の実施の形態によるリソース移動処理に係るフローチャートの一例を示す図である。
【発明を実施するための形態】
【００１１】
（Ｉ）第１の実施の形態
以下、本発明の一実施の形態を詳述する。ただし、本発明は、実施の形態に限定されるものではない。
【００１２】
本実施の形態に係るストレージシステムにおいては、ボリュームの領域を複数のスライスと呼ぶ領域に分割し、スライス単位で複数のサーバ計算機に領域を割り当て、ボリュームへのアクセス負荷をモニタリングする。本ストレージシステムでは、主にスライスが割り当てられたサーバ計算機に負荷が生じ、アクセス負荷が低く１つのサーバ計算機でボリュームが要求する性能を提供できる場合は、ボリュームを構成するスライスを１つのサーバ計算機に集約するように割り当てを制御する。また、ストレージシステムでは、アクセス負荷が高く１つのサーバ計算機でボリュームが要求する性能を提供できない場合は、ボリュームを構成するスライスを複数のサーバ計算機に分散して割り当てるように制御する。また、本ストレージシステムでは、ホストがボリュームのデータにアクセスする際は、各サーバ計算機にてアクセス先のスライスがどのサーバ計算機に割り当たっているか判定することで、アクセス時の負荷が特定のサーバ計算機に偏らないようにする。
【００１３】
これにより、ボリュームのアクセス負荷が１つのサーバ計算機で充足する場合、必ずローカルストレージのデータに対してアクセスできるため、ホストに対して、高速に応答することができる。また、ボリュームのアクセス負荷が１つのサーバ計算機では充足しない場合、複数のサーバ計算機でアクセスを処理することでホストに対して高いスループット（ＩＯＰＳ：Ｉｎｐｕｔ／Ｏｕｔｐｕｔ Ｐｅｒ Ｓｅｃｏｎｄｓ）を提供できる。また、これらの制御は、ユーザが意識することなく、ストレージシステムが自動で行うため、ユーザは、特許文献１に記載のストレージシステムと変わらない運用負荷で上記の利益を得ることができる。
【００１４】
本ストレージシステムによれば、１つのボリュームに対してもサーバ計算機の追加に合わせて、容量および性能をスケールアウトし、ボリュームのアクセス負荷に応じて、応答時間とスループットとを自動で好適な状態に変更できる。
【００１５】
次に、本発明の実施の形態を図面に基づいて説明する。以下の記載および図面は、本発明を説明するための例示であって、説明の明確化のため、適宜、省略および簡略化がなさ

れている。本発明は、他の種々の形態でも実施することが可能である。特に限定しない限り、各構成要素は、単数でも複数でも構わない。

【００１６】

なお、以下の説明では、同種の要素を区別しないで説明する場合には、枝番を含む参照符号のうちの共通部分（枝番を除く部分）を使用し、同種の要素を区別して説明する場合は、枝番を含む参照符号を使用することがある。例えば、物理領域を特に区別しないで説明する場合には、「物理領域１２１」と記載し、個々の領域を区別して説明する場合には、「物理領域１２１－１」、「物理領域１２１－２」のように記載することがある。

【００１７】

本明細書等における「第１」、「第２」、「第３」等の表記は、構成要素を識別するために付するものであり、必ずしも、数または順序を限定するものではない。また、構成要素の識別のための番号は、文脈毎に用いられ、１つの文脈で用いた番号が、他の文脈で必ずしも同一の構成を示すとは限らない。また、ある番号で識別された構成要素が、他の番号で識別された構成要素の機能を兼ねることを妨げるものではない。

【００１８】

図１は、ストレージシステムにおいて、ボリュームのデータの割り当て変更の概要を示すイメージ図である。ストレージシステム１１０からストレージシステム１２０に構成を変更する場合を例に挙げて説明する。

【００１９】

ストレージシステム１１０では、ホスト１０１にボリューム１０２が接続されており、ボリューム１０２内のデータ１０５Ａ，１０５Ｂ，１０５Ｃが、ノード１００Ａ内のストレージプール１０３Ａへ割り当てられている。ノード１００Ａは、ボリューム１０２に割当たったデータ１０５へのアクセス負荷をモニタリングする。モニタリングの結果、ストレージシステム１１０は、ノード１００Ａで提供できる性能を超える負荷を検出した場合、ボリューム１０２のデータ１０５Ｂ，１０５Ｃをノード１００Ｂ，１００Ｃに移動し、負荷を分散し、ストレージシステム１２０の状態に遷移する。本制御により、ボリューム１０２の高負荷時は、多数のノード１００に処理を分散することで単体のボリューム１０２について高い性能を提供できるようにする。

【００２０】

ストレージシステム１２０では、ホスト１０１にボリューム１０２が接続されている。ボリューム１０２内のデータ１０５Ａがノード１００Ａ内のストレージプール１０３Ａに割り当てられている。データ１０５Ｂがノード１００Ｂ内のストレージプール１０３Ｂに割り当てられている。データ１０５Ｃがノード１００Ｃ内のストレージプール１０３Ｃに割り当てられている。ノード１００Ａ，１００Ｂ，１００Ｃは、ボリューム１０２に割当たったデータ１０５Ａ，１０５Ｂ，１０５Ｃへのアクセス負荷をモニタリングする。モニタリングの結果、ストレージシステム１２０は、各データ１０５のアクセス負荷がノード１００Ａで提供できる性能を超えない負荷であることを検出した場合、ボリューム１０２のデータ１０５Ｂ，１０５Ｃをノード１００Ａへ移動し、負荷を集約し、ストレージシステム１１０の状態に遷移する。本制御により、ボリューム１０２の低負荷時は、単一ノード１００で処理を集約することでネットワークの利用効率を高め、ストレージシステム全体について、高い性能を提供できるようにする。

【００２１】

図２は、ストレージシステム２００に係る物理構成の一例を示す図である。

【００２２】

ストレージシステム２００には、１以上のサイト２０１が設けられてもよい。各サイト２０１は、ネットワーク２０２を介して通信可能に接続される。ネットワーク２０２は、例えば、ＷＡＮ（Ｗｉｄｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）であるが、ＷＡＮに限定するものではない。

【００２３】

サイト２０１は、データセンタ等であり、１以上のノード１００を含んで構成される。

【００２４】
ノード１００は、一般的なサーバ計算機の構成を備えてよい。ノード１００は、例えば、プロセッサ２１１、メモリ２１２等を含む１以上のプロセッサパッケージ２１３、１以上のドライブ２１４、１以上のポート２１５を含んで構成される。各構成要素は、内部バス２１６を介して接続されている。
【００２５】
プロセッサ２１１は、例えば、ＣＰＵ（Ｃｅｎｔｒａｌ Ｐｒｏｃｅｓｓｉｎｇ Ｕｎｉｔ）であり、各種の処理を行う。
【００２６】
メモリ２１２は、ノード１００の機能を実現する上で必要な制御用の情報を格納したり、データを格納したりする。また、メモリ２１２は、例えば、プロセッサ２１１により実行されるプログラムを格納する。メモリ２１２は、揮発性のＤＲＡＭ（Ｄｙｎａｍｉｃ Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）であってもよいし、不揮発のＳＣＭ（Ｓｔｏｒａｇｅ Ｃｌａｓｓ Ｍｅｍｏｒｙ）であってもよいし、その他の記憶デバイスであってもよい。
【００２７】
ドライブ２１４は、各種のデータ、プログラム等を記憶する。ドライブ２１４は、ＳＡＳ（Ｓｅｒｉａｌ Ａｔｔａｃｈｅｄ ＳＣＳＩ）またはＳＡＴＡ（Ｓｅｒｉａｌ Ａｄｖａｎｃｅｄ Ｔｅｃｈｎｏｌｏｇｙ Ａｔｔａｃｈｍｅｎｔ）接続のＨＤＤ（Ｈａｒｄ Ｄｉｓｋ Ｄｒｉｖｅ）やＳＳＤ（Ｓｏｌｉｄ Ｓｔａｔｅ Ｄｒｉｖｅ）、ＮＶＭｅ（Ｎｏｎ－Ｖｏｌａｔｉｌｅ Ｍｅｍｏｒｙ Ｅｘｐｒｅｓｓ）接続のＳＳＤの他、ＳＣＭ等であってもよく、記憶デバイスの一例である。
【００２８】
ポート２１５は、ネットワーク２２０に接続され、サイト２０１内の他のノード１００と通信可能に接続されている。ネットワーク２２０は、例えば、ＬＡＮ（Ｌｏｃａｌ Ａｒｅａ Ｎｅｔｗｏｒｋ）であるが、ＬＡＮに限定するものではない。
【００２９】
ストレージシステム２００に係る物理構成は、上述の内容に限定されるものではない。例えば、ネットワーク２２０，２０２については、冗長化されていてもよい。また、例えば、ネットワーク２２０は、管理用のネットワークとストレージ用のネットワークとで分離してもよく、接続規格は、Ｅｔｈｅｒｎｅｔ（登録商標）、Ｉｎｆｉｎｉｂａｎｄ、無線でもよく、接続トポロジも図２に示す構成に限定しない。
【００３０】
なお、ホスト１０１は、ノード１００と同じ構成要素を備えてもよく、ホスト１０１の物理構成については、その説明を省略する。
【００３１】
図３は、ストレージシステム２００に係る論理構成の一例を示す図である。ストレージシステム２００では、ストレージ仮想化が行われ、複数の物理領域が仮想的に統合され、ストレージプール３１２として利用される。さらに、ストレージシステム２００では、シンプロビジョニングにより、各ホスト１０１が現在利用している容量だけが割り当てられている。
【００３２】
より具体的には、図３に示すように、ドライブ２１４は、データ、パリティ等を格納する物理的な領域であるデータ格納領域を有する。データ格納領域のうちの全部または一部の領域であり、連続した領域である論理ドライブ３１８は、ノード１００を跨る複数の論理ドライブ３１８を組み合わせてパリティグループ３１７を構築する。
【００３３】
パリティグループ３１７は、複数のノード１００のドライブ２１４の論理ドライブ３１８から構成される。例えば、データ保護ポリシが２Ｄ１Ｐである場合、異なるノード１００のドライブ２１４から確保した３つの論理ドライブ３１８でパリティグループ３１７が

構成される。
【００３４】
ここで、データ保護ポリシとしては、例えば、ＥＣ（Ｅｒａｓｕｒｅ　Ｃｏｄｉｎｇ）がある。なお、ＥＣとしては、データローカリティを保持しない第１の方式と、データローカリティを保持する第２の方式（例えば、国際公開第２０１６／５２６６５号に記載の方式）とがあるが、ストレージシステム２００には、何れの方式も適用可能である。なお、本実施の形態では、第２の方式を適用したケースを例に挙げて主に説明する。
【００３５】
付言するならば、例えば、第１の方式の２Ｄ１ＰのＥＣでは、ライト要求のデータを第１のデータと第２のデータとに分け、第１のデータを第１のノード１００に格納し、第２のデータを第２のノード１００に格納し、第１のデータおよび第２のデータで計算されたパリティを第３のノード１００に格納することで冗長化が行われる。また、例えば、第２の方式の２Ｄ１ＰのＥＣでは、ライト要求のデータを第１のデータと第２のデータとに分け、第１のデータおよび第２のデータを第１のノード１００（自ノード１００）に格納し、第１のデータのパリティを第２のノード１００に格納し、第２のデータのパリティを第３のノード１００に格納することで冗長化が行われる。
【００３６】
パリティグループ３１７からは、プールボリューム３１６が切り出される。プールボリューム３１６は、各ノード１００のストレージプール３１２に容量を割り当てる単位である。１つのパリティグループ３１７から１つのプールボリューム３１６が切り出されてもよいし、複数のプールボリューム３１６が切り出されてよい。
【００３７】
付言するならば、例えば、データ保護ポリシが２Ｄ１Ｐである場合、データの格納領域として利用できるのは、パリティグループ３１７に割り当てられた論理ドライブ３１８の総量の２／３となり、パリティの格納領域として利用できるのは、パリティグループ３１７に割り当てられた論理ドライブ３１８の総量の１／３となる。つまり、プールボリューム３１６として切り出せる最大の容量は、データ保護ポリシに応じて異なる。
【００３８】
切り出されたプールボリューム３１６は、ストレージプール３１２にアタッチされる。ストレージプール３１２は、１以上のプールボリューム３１６を含んで構成される。ストレージプール３１２からは、アプリケーション３０１により利用される仮想ボリューム３１３が切り出される。つまり、ストレージプログラム３１１は、利用者の要求に応じた容量を、ドライブ２１４に割り当てず、仮想ボリューム３１３として割り当てる。
【００３９】
ストレージプール３１２から仮想ボリューム３１３を切り出す際、複数のストレージプール３１２から仮想ボリューム３１３の領域をスライス３１４として部分的に切り出し、スライス３１４を束ねることで仮想ボリューム３１３を構築する。スライス３１４は、ストレージプール３１２に対して、仮想的に割り当てられた領域であり、仮想ボリューム３１３を作成した時点では、物理的な領域が割り当てられない。スライス３１４には、１以上のページ３１５が割り当てられる。例えば、ストレージプログラム３１１は、アプリケーション３０１からライト要求を受信した場合、新規のライトであるときは、仮想ボリューム３１３のスライス３１４にページ３１５（より詳細には、ページ３１５に紐づく論理ドライブ３１８の物理領域）を割り当てる。なお、ページ３１５には、プールボリューム３１６のページが対応付けられている。更新のライトであるときは、ストレージプログラム３１１は、割り当てたページ３１５に紐づく論理ドライブ３１８の物理領域を特定してデータを更新する。なお、ライト要求のデータ（または後述の中間データ）は、データの冗長化に係る他のノード１００に転送されてパリティが更新される。
【００４０】
仮想ボリューム３１３とアプリケーション３０１とは、フロントエンドパス３２０（以降、単にパスとも記述する）で接続される。フロントエンドパス３２０の接続および設定

は、ストレージプログラム３１１とホスト１０１上で動作するパス設定プログラム３０２とにより制御される。なお、図３では、スライス３１４を第１のノード１００「Ｎｏｄｅ０」から第２のノード１００「Ｎｏｄｅ１」に移動した後、移動先（割当先）の第２のノード１００「Ｎｏｄｅ１」にフロントエンドパス３２０が設定されていない例を示している。この場合、一度、第１のノード１００「Ｎｏｄｅ０」を経由してスライス３１４の割当先の第２のノード１００「Ｎｏｄｅ１」にＩＯコマンドが転送されて処理される。ただし、後述するように、移動先の第２のノード１００「Ｎｏｄｅ１」にフロントエンドパス３２０が設定され、最適化が行われることが好適である。

【００４１】

このように、ストレージプログラム３１１は、ドライブ２１４を共有のストレージプール３１２として管理し、仮想ボリューム３１３に書き込まれたデータ量に応じてドライブ２１４に容量を割り当てる。これにより、使用されないドライブ２１４の無駄をなくし、効率的な運用が行わる。

【００４２】

以下では、データを更新するにあたり、当該データは、ライト要求を受領したノード１００のドライブ２１４（ローカルドライブ）に格納される構成（データローカリティを維持し、リード時のネットワークオーバヘッドを排除する構成）を例に挙げて主に説明する。

【００４３】

なお、データにアクセスするアプリケーション３０１は、ホスト１０１に設けられて動作するものであってもよいし、ストレージプログラム３１１と同一ノード１００に設けられて動作するものであってもよいし、別のノード１００に設けられて動作するものであってもよい。

【００４４】

図４は、メモリ２１２内の情報（ドライブ２１４からメモリ２１２に読み出される情報）の一例を示す図である。なお、制御情報テーブル４１０、各種のプログラム（ストレージプログラム３１１等）は、実行中はメモリ２１２上に展開されるが、停電等に備えてドライブ２１４等の不揮発な領域に格納されている。

【００４５】

制御情報テーブル４１０には、クラスタ構成管理テーブル４１１、データ保護セット管理テーブル４１２、ストレージプール管理テーブル４１３、ボリューム管理テーブル４１４、モニタ情報管理テーブル４１５、およびフロントエンドパス管理テーブル４１６が含まれる。各テーブルについては、図５～図１０を用いて後述する。

【００４６】

ストレージプログラム３１１は、リード処理プログラム４２１、ライト処理プログラム４２２、モニタ情報採取処理プログラム４２３、リバランス要否判定処理プログラム４２４、リソース割当決定処理プログラム４２５、リソース移動処理プログラム４２６、フロントエンドパス設定処理プログラム４２７、およびクラスタ構成変更処理プログラム４２８を備える。なお、パス設定プログラム３０２は、フロントエンドパス設定処理プログラム４２７を備える。

【００４７】

ノード１００の機能（リード処理プログラム４２１、ライト処理プログラム４２２、モニタ情報採取処理プログラム４２３、リバランス要否判定処理プログラム４２４、リソース割当決定処理プログラム４２５、リソース移動処理プログラム４２６、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７、クラスタ構成変更処理プログラム４２８等）は、例えば、プロセッサ２１１がドライブ２１４に格納されたプログラムをメモリ２１２に読み出して実行すること（ソフトウェア）により実現されてもよいし、専用の回路等のハードウェアにより実現されてもよいし、ソフトウェアとハードウェアとが組み合わされて実現されてもよい。また、ノード１００の機能の一部は、ノード１００と通信可能な他のコンピュータにより実現されてもよい。

【００４８】

ホスト１０１の機能（例えば、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７）は、例えば、プロセッサ２１１がドライブ２１４に格納されたプログラムをメモリ２１２に読み出して実行すること（ソフトウェア）により実現されてもよいし、専用の回路等のハードウェアにより実現されてもよいし、ソフトウェアとハードウェアとが組み合わされて実現されてもよい。また、ホスト１０１の機能の一部は、ホスト１０１と通信可能な他のコンピュータにより実現されてもよい。

【００４９】

図５は、クラスタ構成管理テーブル４１１の一例を示す図である。

【００５０】

クラスタ構成管理テーブル４１１は、サイト２０１、ノード１００、ドライブ２１４の構成を管理するための情報を格納する。

【００５１】

クラスタ構成管理テーブル４１１は、サイト構成管理テーブル５１０、ノード構成管理テーブル５２０、およびドライブ構成管理テーブル５３０を含んで構成される。なお、ストレージシステム２００は、サイト構成管理テーブル５１０を管理し、サイト２０１は、サイト２０１内の複数のノード構成管理テーブル５２０を管理し、ノード１００は、ノード１００内の複数のドライブ構成管理テーブル５３０を管理する。

【００５２】

サイト構成管理テーブル５１０は、サイト２０１に係る構成（サイト２０１とノード１００との関係等）を示す情報を格納する。より具体的には、サイト構成管理テーブル５１０は、サイトＩＤ５１１と、状態５１２と、ノードＩＤリスト５１３とが対応付けられた情報を格納する。

【００５３】

サイトＩＤ５１１は、サイト２０１を識別可能な識別情報である。状態５１２は、サイト２０１の状態を示す状態情報（ＮＯＲＭＡＬ、ＷＡＲＮＩＮＧ、ＦＡＩＬＵＲＥ等）である。ノードＩＤリスト５１３は、サイト２０１に設けられるノード１００を識別可能な識別情報のリストである。

【００５４】

ノード構成管理テーブル５２０は、サイト２０１ごとに設けられ、サイト２０１に設けられるノード１００に係る構成（ノード１００とドライブ２１４との関係等）を示す情報を格納する。より具体的には、ノード構成管理テーブル５２０は、ノードＩＤ５２１と、状態５２２と、ドライブＩＤリスト５２３とが対応付けられた情報を格納する。

【００５５】

ノードＩＤ５２１は、ノード１００を識別可能な識別情報である。状態５２２は、ノード１００の状態を示す状態情報（ＮＯＲＭＡＬ、ＷＡＲＮＩＮＧ、ＦＡＩＬＵＲＥ等）である。ドライブＩＤリスト５２３は、ノード１００に設けられるドライブ２１４を識別可能な識別情報のリストである。

【００５６】

ドライブ構成管理テーブル５３０は、ノード１００ごとに設けられ、ノード１００に設けられるドライブ２１４に係る構成を示す情報を格納する。より具体的には、ドライブ構成管理テーブル５３０は、ドライブＩＤ５３１と、状態５３２と、サイズ５３３とが対応付けられた情報を格納する。

【００５７】

ドライブＩＤ５３１は、ドライブ２１４を識別可能な識別情報である。状態５３２は、ドライブ２１４の状態を示す状態情報（ＮＯＲＭＡＬ、ＷＡＲＮＩＮＧ、ＦＡＩＬＵＲＥ等）である。サイズ５３３は、ドライブ２１４の容量を示す情報（ＴＢ（テラバイト）、ＧＢ（ギガバイト）等）である。

【００５８】

図６は、データ保護セット管理テーブル４１２の一例を示す図である。

【００５９】

データ保護セット管理テーブル４１２は、論理ドライブ３１８を組み合わせて構成したパリティグループ３１７の構成を管理するための制御情報を格納する。

【００６０】

データ保護セット管理テーブル４１２は、プールボリューム管理テーブル６１０、パリティグループ管理テーブル６２０、論理ドライブ管理テーブル６３０、およびストライプマッピングテーブル６４０を含んで構成される。

【００６１】

プールボリューム管理テーブル６１０は、パリティグループ３１７から切り出されたプールボリューム３１６に係る情報を格納する。より具体的には、プールボリューム管理テーブル６１０は、プールボリュームＩＤ６１１と、サイズ６１２と、パリティグループＩＤ６１３と、論理ドライブＩＤ６１４とが対応付けられた情報を格納する。

【００６２】

プールボリュームＩＤ６１１は、パリティグループ３１７から切り出されたプールボリューム３１６を識別可能な識別情報である。サイズ６１２は、プールボリューム３１６の容量を示す情報（ＴＢ（テラバイト）、ＧＢ（ギガバイト）等）である。パリティグループＩＤ６１３は、プールボリューム３１６が属するパリティグループ３１７を識別可能な識別情報である。論理ドライブＩＤ６１４は、プールボリューム３１６に格納するデータ領域を提供する論理ドライブ３１８を識別可能な識別情報である。

【００６３】

パリティグループ管理テーブル６２０は、パリティグループ３１７に係る情報を格納する。より具体的には、パリティグループ管理テーブル６２０は、パリティグループＩＤ６２１と、冗長化ポリシ６２２と、論理ドライブＩＤリスト６２３とが対応付けられた情報を格納する。

【００６４】

パリティグループＩＤ６２１は、パリティグループ３１７を識別可能な識別情報である。冗長化ポリシ６２２は、パリティグループ３１７の冗長化方法に関する設定である。論理ドライブＩＤリスト６２３は、パリティグループ３１７に割り当てられた論理ドライブ３１８を識別可能な識別情報のリストである。

【００６５】

論理ドライブ管理テーブル６３０は、論理ドライブ３１８に係る情報（開始オフセットからサイズ分だけドライブ２１４の物理領域を切り出して論理ドライブ３１８として管理するための情報）を格納する。より具体的には、論理ドライブ管理テーブル６３０は、論理ドライブＩＤ６３１と、開始オフセット６３２と、サイズ６３３と、ドライブＩＤ６３４とが対応付けられた情報を格納する。

【００６６】

論理ドライブＩＤ６３１は、論理ドライブ３１８を識別可能な識別情報である。開始オフセット６３２は、ドライブ２１４から論理ドライブ３１８を切り出すときの開始位置を示す情報である。サイズ６３３は、論理ドライブ３１８の容量を示す情報（ブロックの数）である。ここでブロックとは、ドライブ２１４へのアクセス単位を意味しており、典型的には、１ブロックのサイズは、５１２Ｂｙｔｅである。ただし、ブロックのサイズは、５１２Ｂｙｔｅに限定せず、４ＫＢ、８ＫＢ等でもよい。ドライブＩＤ６３４は、論理ドライブ３１８が切り出されている記憶資源を識別可能な識別情報（論理ドライブ３１８がどのドライブ２１４から切り出されているかを示す情報）である。

【００６７】

ストライプマッピングテーブル６４０は、パリティグループ３１７に係る情報（データおよびパリティの格納先アドレスを計算するための情報）を格納する。一例として、ストライプマッピングテーブル６４０が、ＥＣ（２Ｄ１Ｐ）のストライプマッピングテーブル６４１、およびＭｉｒｒｏｒ（２－Ｒｅｐｌｉｃａｔｉｏｎ）のストライプマッピングテーブル６４２の情報を格納するケースについて説明する。２Ｄ１Ｐとは、２つのデータの組み合わせで１つのパリティを算出し、データを保護することを意味する。

【００６８】
ストライプマッピングテーブル６４１，６４２は、あるデータ領域の物理ＬＢＡ（Ｌｏｇｉｃａｌ Ｂｌｏｃｋ Ａｄｄｒｅｓｓ）に対して、パリティ領域の物理ＬＢＡ（冗長化先ノード）を特定するために使用する。
【００６９】
ストライプマッピングテーブル６４１，６４２は、表、配列形式等で格納されており、横軸の要素としてノードＩＤに対応する情報を保持し、縦軸の要素としてアドレスに対応する情報を保持している。横軸の情報は、直接的にノードＩＤの情報を格納していてもよいし、ストライプマッピングテーブル６４１，６４２の横軸のＩＤとノードＩＤとを対応づける別のテーブルを介して管理されていてもよい。縦軸の情報は、直接的にＬＢＡの情報を格納している必要はなく、例えばＬＢＡから縦軸のＩＤへは以下のように変換することができる。
【００７０】
ＲｏｗＩＤ ＝ ＬＢＡ ｍｏｄ Ｒｏｗｍａｘ
（Ｒｏｗｍａｘは、ストライプマッピングテーブル６４１，６４２では「６」となる）
【００７１】
図７は、ストレージプール管理テーブル４１３の一例を示す図である。
【００７２】
ストレージプール管理テーブル４１３は、ストレージプール３１２の構成を管理するための制御情報を格納する。ストレージプール管理テーブル４１３は、ストレージプール情報テーブル７１０を含んで構成される。
【００７３】
ストレージプール情報テーブル７１０は、ストレージプール３１２に係る情報を格納する。より具体的には、ストレージプール情報テーブル７１０は、ストレージプールＩＤ７１１と、合計容量７１２と、使用容量７１３と、ノードＩＤ７１４と、プールボリュームＩＤリスト７１５とが対応付けられた情報を格納する。
【００７４】
ストレージプールＩＤ７１１は、ストレージプール３１２を識別可能な識別情報である。合計容量７１２は、ストレージプール３１２に割り当てられた合計容量を示す情報（ＴＢ（テラバイト）、ＧＢ（ギガバイト）等）である。使用容量７１３は、ストレージプール３１２で使用している容量を示す情報（ＴＢ（テラバイト）、ＧＢ（ギガバイト）等）である。ノードＩＤ７１４は、ストレージプール３１２を提供するノード１００を識別可能な識別情報である。プールボリュームＩＤリスト７１５は、ストレージプール３１２に割り当てられたプールボリューム３１６を識別可能な識別情報のリストである。
【００７５】
図８は、ボリューム管理テーブル４１４の一例を示す図である。
【００７６】
ボリューム管理テーブル４１４は、仮想ボリューム３１３の構成情報と、ノード１００間に割り当てられたスライス３１４の構成情報と、シンプロビジョニング機能のための制御情報とを格納する。
【００７７】
ボリューム管理テーブル４１４は、仮想ボリューム管理テーブル８１０、スライス管理テーブル８２０、およびページマッピングテーブル８３０を含んで構成される。
【００７８】
仮想ボリューム管理テーブル８１０は、仮想ボリューム３１３に係る情報（仮想ボリューム３１３と仮想ボリューム３１３に割り当てられたスライス３１４との対応関係を示す情報等）を格納する。より具体的には、仮想ボリューム管理テーブル８１０は、仮想ボリュームＩＤ８１１と、サイズ８１２と、スライスＩＤリスト８１３と、最大分散度８１４とが対応付けられた情報を格納する。
【００７９】

仮想ボリュームＩＤ８１１は、仮想ボリューム３１３を識別可能な識別情報である。サイズ８１２は、仮想ボリューム３１３の容量を示す情報（ＴＢ（テラバイト）、ＧＢ（ギガバイト）等）である。スライスＩＤリスト８１３は、仮想ボリューム３１３に割り当てられたスライス３１４を識別可能な識別情報のリストである。最大分散度８１４は、仮想ボリューム３１３に割り当てるスライス３１４を分散させるノード数の最大値である。これを超えた数のノード１００には、スライス３１４が割り当てられないように制御される。

【００８０】

スライス管理テーブル８２０は、仮想ボリューム３１３に割り当てたスライス３１４に係る情報（スライス３１４とスライス３１４に対応するストレージプール３１２との対応関係を示す情報等）を格納する。より具体的には、スライス管理テーブル８２０は、スライスＩＤ８２１と、サイズ８２２と、ストレージプールＩＤ８２３と、状態８２４とが対応付けられた情報を格納する。

【００８１】

スライスＩＤ８２１は、スライス３１４を識別可能な識別情報である。サイズ８２２は、スライス３１４の容量を示す情報（ＴＢ（テラバイト）、ＧＢ（ギガバイト）、Ｌｏｇｉｃａｌ　Ｂｌｏｃｋ数等）である。ストレージプールＩＤ８２３は、スライス３１４に対応するストレージプール３１２を識別可能な識別情報である。２つのストレージプール３１２間でスライス３１４を移動中の場合、ストレージプールＩＤ８２３は、移動前後のストレージプール３１２を識別可能な識別情報を格納する。状態８２４は、スライス３１４の状態を示す情報である。状態８２４には、正常（Ｎｏｒｍａｌ）、障害状態（Ｆａｉｌｕｒｅ）、２つのストレージプール３１２間を移動中（Ｍｉｇｒａｔｉｎｇ）といった状態がある。

【００８２】

ページマッピングテーブル８３０は、仮想ボリューム３１３に割り当てたページ３１５に係る情報（ページ３１５とプールボリューム３１６との対応関係を示す情報等）を格納する。より具体的には、ページマッピングテーブル８３０は、ページＩＤ８３１と、仮想ボリュームＩＤ８３２と、仮想ボリュームＬＢＡ８３３と、サイズ８３４と、プールボリュームＩＤ８３５と、プールボリュームＬＢＡ８３６とが対応付けられた情報を格納する。

【００８３】

ページＩＤ８３１は、ページ３１５を識別可能な識別情報である。仮想ボリュームＩＤ８３２は、ページ３１５が割り当てられている仮想ボリューム３１３を識別可能な識別情報である。仮想ボリュームＬＢＡ８３３は、仮想ボリューム３１３におけるページ３１５の位置を特定可能な情報であり、例えば、仮想ボリューム３１３の最初のページ３１５から何番目であるかを示す情報である。なお、ページ３１５は、ストレージプログラム３１１が仮想ボリューム３１３にアクセスする単位である。サイズ８３４は、ページ３１５の容量を示す情報（ＴＢ（テラバイト）、ＧＢ（ギガバイト）、Ｌｏｇｉｃａｌ　Ｂｌｏｃｋ数等）である。プールボリュームＩＤ８３５は、ページ３１５に対応するプールボリューム３１６を識別可能な識別情報である。プールボリュームＬＢＡ８３６は、ストレージプール３１２におけるプールボリューム３１６の位置を特定可能な情報であり、例えば、ストレージプール３１２の最初のプールボリューム３１６から何番目であるかを示す情報である。

【００８４】

なお、サイズ８３４は、全てのページ３１５で同じであってもよいし、ページ３１５ごとに異なっていてもよい。

【００８５】

付言するならば、ストレージプログラム３１１は、仮想ボリューム３１３のアドレスからストレージプール３１２のアドレスへの変換を行う際にページマッピングテーブル８３０を参照する。また、ストレージプログラム３１１は、新規ライトを受領する度に、ページ３１５の割当て（ページマッピングテーブル８３０へのレコードの追加）を行う。

【００８６】

図９は、モニタ情報管理テーブル４１５の一例を示す図である。

【００８７】

モニタ情報管理テーブル４１５は、ノード１００で動作するプロセスのプロセッサ２１１、ドライブ２１４、およびポート２１５の使用量と、仮想ボリューム３１３のスライス３１４、およびフロントエンドパス３２０に対するアクセス頻度とを管理するためのするための制御情報を格納する。

【００８８】

モニタ情報管理テーブル４１５は、プロセッサモニタ情報管理テーブル９１０、ドライブモニタ情報管理テーブル９２０、ネットワークモニタ情報管理テーブル９３０、スライスモニタ情報管理テーブル９４０、フロントエンドパスモニタ情報管理テーブル９５０を含んで構成される。

【００８９】

プロセッサモニタ情報管理テーブル９１０は、プロセッサ２１１に係る情報（プロセスとプロセッサ２１１の使用量との関係を示す情報）を格納する。より具体的には、ノードＩＤ９１１と、プロセッサＩＤ９１２と、プロセスＩＤ９１３と、プロセス名９１４と、使用率９１５とが対応付けられた情報を格納する。

【００９０】

ノードＩＤ９１１は、ノード１００を識別可能な識別情報である。プロセッサＩＤ９１２は、ノード１００内に複数のプロセッサコアが搭載されている場合にプロセッサコアを識別可能な識別情報である。プロセスＩＤ９１３は、ノード１００で動作するプログラムを識別可能な識別情報である。プロセス名９１４は、ノード１００で動作するプログラムを識別可能な文字列情報である。使用率９１５は、ノード１００で動作するプログラムが動作するプロセッサコアの占有率を示す。例えば、ストレージプログラム３１１の使用率が５０％である場合、ストレージプログラム３１１が動作するプロセッサコア動作周波数の半分を占有していることを意味する。

【００９１】

ドライブモニタ情報管理テーブル９２０は、ドライブ２１４に係る情報（ドライブ２１４の使用量の関係を示す情報）を格納する。より具体的には、ドライブＩＤ９２１と、リードＩＯＰＳ９２２と、ライトＩＯＰＳ９２３と、リード転送量９２４と、ライト転送量９２５と、使用率９２６とが対応付けられた情報を格納する。

【００９２】

ドライブＩＤ９２１は、ドライブ２１４を識別可能な識別情報である。リードＩＯＰＳ９２２は、当該ドライブ２１４に対してのリードコマンドの秒間あたりの処理数である。ライトＩＯＰＳ９２３は、当該ドライブ２１４に対してのライトコマンドの秒間あたりの処理数である。リード転送量９２４は、当該ドライブ２１４に対してのリードコマンドの秒間あたりのデータ転送量である。ライト転送量９２５は、当該ドライブ２１４に対してのライトコマンドの秒間あたりのデータ転送量である。使用率９２６は、当該ドライブ２１４の負荷度合いを示し、１００％となった場合、当該ドライブ２１４は、それ以上Ｉ／Ｏを処理できず、ドライブ２１４が受領したＩ／Ｏ要求は待たされることになる。

【００９３】

ネットワークモニタ情報管理テーブル９３０は、ネットワーク２２０に接続されているポート２１５に係る情報（ポート２１５の使用量の関係を示す情報）を格納する。より具体的には、ノードＩＤ９３１と、ＮＩＣ（Ｎｅｔｗｏｒｔｋ Ｉｎｔｅｒｆａｃｅ Ｃａｒｄ）ＩＤ９３２と、送信転送量９３３と、受信転送量９３４と、最大転送量９３５とが対応付けられた情報を格納する。

【００９４】

ノードＩＤ９３１は、ノード１００を識別可能な識別情報である。ＮＩＣ ＩＤ９３２は、ノード１００内に複数のＮＩＣ（ポート２１５）が搭載されている場合にＮＩＣを識別可能な識別情報である。なお、本実施の形態では、ＮＩＣが１つのポート２１５を備える場合を例に挙げて説明する。送信転送量９３３は、当該ＮＩＣに対しての送信処理の秒

間あたりの転送量である。受信転送量９３４は、当該ＮＩＣに対しての受信処理の秒間あたりの転送量である。最大転送量９３５は、当該ＮＩＣで処理可能な送受信の秒間あたりの最大転送量である。

【００９５】

スライスモニタ情報管理テーブル９４０は、スライス３１４へのアクセス頻度の情報を格納する。より具体的には、スライスモニタ情報管理テーブル９４０は、スライスＩＤ９４１と、リードカウンタ９４２と、ライトカウンタ９４３と、リード転送量９４４と、ライト転送量９４５と、モニタ開始時刻９４６とが対応付けられた情報を格納する。

【００９６】

スライスＩＤ９４１は、スライス３１４を識別可能な識別情報である。リードカウンタ９４２は、当該スライス３１４をリードした回数を管理するための情報である。ライトカウンタ９４３は、当該スライス３１４に対してライトした回数を管理するための情報である。リード転送量９４４は、当該スライス３１４をリードした転送量を管理するための情報である。ライト転送量９４５は、当該スライス３１４に対してライトした転送量を管理するための情報である。モニタ開始時刻９４６は、当該スライス３１４に対するアクセスの監視が開始された時間を示す情報である。

【００９７】

フロントエンドパスモニタ情報管理テーブル９５０は、フロントエンドパス３２０へのアクセス頻度の情報を格納する。より具体的には、フロントエンドパスモニタ情報管理テーブル９５０は、パスＩＤ９５１と、リードＩＯＰＳ９５２と、ライトＩＯＰＳ９５３と、リード転送量９５４と、ライト転送量９５５とが対応付けられた情報を格納する。

【００９８】

パスＩＤ９５１は、フロントエンドパス３２０を識別可能な識別情報である。リードＩＯＰＳ９５２は、当該フロントエンドパス３２０に対してのリードコマンドの秒間あたりの処理数である。ライトＩＯＰＳ９５３は、当該フロントエンドパス３２０に対してのライトコマンドの秒間あたりの処理数である。リード転送量９５４は、当該フロントエンドパス３２０に対してのリードコマンドの秒間あたりのデータ転送量である。ライト転送量９５５は、当該フロントエンドパス３２０に対してのライトコマンドの秒間あたりのデータ転送量である。

【００９９】

図１０は、フロントエンドパス管理テーブル４１６の一例を示す図である。

【０１００】

フロントエンドパス管理テーブル４１６は、フロントエンドパス３２０の構成を管理するための制御情報を格納する。フロントエンドパス管理テーブル４１６は、フロントエンドパス情報テーブル１０１０を含んで構成される。

【０１０１】

フロントエンドパス情報テーブル１０１０は、フロントエンドパス３２０に係る情報を格納する。より具体的には、フロントエンドパス情報テーブル１０１０は、パスＩＤ１０１１と、仮想ボリュームＩＤ１０１２と、ＩｎｉｔｉａｔｏｒＩＤ１０１３と、ＡＬＵＡ設定１０１４と、接続ノードＩＤ１０１５とが対応付けられた情報を格納する。

【０１０２】

パスＩＤ１０１１は、フロントエンドパス３２０を識別可能な識別情報である。仮想ボリュームＩＤ１０１２は、フロントエンドパス３２０が割り当てられた仮想ボリューム３１３を識別可能な識別情報である。ＩｎｉｔｉａｔｏｒＩＤ１０１３は、フロントエンドパス３２０の接続先であるホスト１０１を識別可能な識別情報である。ＡＬＵＡ設定１０１４は、ストレージシステム２００にとって、対応するフロントエンドパス３２０が好適であるかどうかの設定を示す情報である。ＡＬＵＡ設定１０１４に基づく情報を、ホスト１０１に通知することで、ホスト１０１は、好適なパスへＩ／Ｏリクエストを発行することができ、ストレージシステム２００の処理効率を向上させることができる。接続ノードＩＤ１０１５は、フロントエンドパス３２０を有するノードＩＤを識別可能な識別情報で

ある。
【０１０３】
図１１は、リード処理に係るフローチャートの一例を示す図である。リード処理では、アプリケーション３０１からのデータのリード処理要求を受けて、自ノード１００のドライブ２１４からデータが読み出される。なお、リード処理要求では、リード先（例えば、ＬＵＮ（Ｌｏｇｉｃａｌ Ｕｎｉｔ Ｎｕｍｂｅｒ）のような仮想ボリュームＩＤ、ＬＢＡのようなアドレス等）が指定されている。アクセス先（ドライブ２１４等）が障害状態である場合、冗長データからリード対象のデータが修復されて応答される。以下、詳細について説明する。
【０１０４】
ステップＳ１１０１では、リード処理プログラム４２１は、アクセス先ＬＢＡからスライスＩＤを計算する。より具体的には、リード処理プログラム４２１は、仮想ボリューム管理テーブル８１０を参照し、スライスＩＤリスト８１３の先頭のスライスから仮想ボリューム３１３のＬＢＡが連続的に割り当てられているとき、リストを順に辿ることでアクセス先のＬＢＡに該当するスライスＩＤを取得する。
【０１０５】
ステップＳ１１０２では、リード処理プログラム４２１は、ステップＳ１１０１で取得したスライスＩＤ（対象スライス）が自身のノード１００（自系ノード）に割り当てられているか否かを判定する。より具体的には、リード処理プログラム４２１は、スライス管理テーブル８２０を参照し、該当するスライスＩＤに対応するストレージプールＩＤを取得する。次に、リード処理プログラム４２１は、ストレージプール情報テーブル７１０を参照し、取得したストレージプールＩＤに対応するノードＩＤを取得する。リード処理プログラム４２１は、取得したノードＩＤと自系ノードのノードＩＤとを比較し、同じノードＩＤである場合、対象スライス（アクセス先のスライス３１４）が自系ノードに割り当たっていると判定する。リード処理プログラム４２１は、取得したノードＩＤと自系ノードのノードＩＤとが異なるノードＩＤである場合、対象スライスは、他のノード１００（他系ノード）に割り当たっていると判定する。リード処理プログラム４２１は、対象スライスが自系ノードに割り当てたられていると判定した場合、ステップＳ１１０５に処理を移し、対象スライスが自系ノードに割り当てられていないと判定した場合、ステップＳ１１０３に処理を移す。
【０１０６】
ステップＳ１１０３では、リード処理プログラム４２１は、対象スライスを割り当てた先の他系ノードにリード処理要求を転送する。
【０１０７】
ステップＳ１１０４では、リード処理プログラム４２１は、ステップＳ１１０３にて転送したリード処理要求の実行結果を待ち受け、実行結果を受信し、ステップＳ１１１１に処理を移す。
【０１０８】
ステップＳ１１０５では、リード処理プログラム４２１は、アクセス先の領域に関しての排他制御を取得する。
【０１０９】
ステップＳ１１０６では、リード処理プログラム４２１は、リード処理要求のデータについて、ストレージプール３１２にページ３１５が未割当てであるか否かを判定する。リード処理プログラム４２１は、未割当てであると判定した場合、ステップＳ１１０７に処理を移し、未割当てでないと判定した場合、ステップＳ１１０８に処理を移す。
【０１１０】
ステップＳ１１０７では、リード処理プログラム４２１は、データがないことを示す０データを生成し、ステップＳ１１１０に処理を移す。
【０１１１】
ステップＳ１１０８では、リード処理プログラム４２１は、アクセス先のアドレス（割

当先アドレス）を取得する。
【０１１２】
ステップＳ１１０９では、リード処理プログラム４２１は、自系ノードのドライブ２１４（ローカルドライブ）からデータを読み出す。
【０１１３】
ステップＳ１１１０では、リード処理プログラム４２１は、取得した排他制御を解放する。
【０１１４】
ステップＳ１１１１では、リード処理プログラム４２１は、ホスト１０１にリード処理結果を応答する。
【０１１５】
ステップＳ１１１２では、リード処理プログラム４２１は、モニタ情報採取処理を実行する。なお、モニタ情報採取処理については、図１３Ｂを用いて後述する。
【０１１６】
図１２は、ライト処理に係るフローチャートの一例を示す図である。ライト処理では、アプリケーション３０１からのライト処理要求を受けて、自系ノードのドライブ２１４にデータが書き込まれ、さらに他系ノードのドライブ２１４に冗長データ（パリティ）が書き込まれる。なお、ライト処理要求では、ライト先（例えば、ＬＵＮのような仮想ボリュームＩＤ、ＬＢＡのようなアドレス等）が指定されている。以下、詳細について説明する。
【０１１７】
ステップＳ１２０１では、ライト処理プログラム４２２は、アクセス先ＬＢＡからスライスＩＤを計算する。より具体的には、ライト処理プログラム４２２は、仮想ボリューム管理テーブル８１０を参照し、スライスＩＤリスト８１３の先頭のスライスから仮想ボリューム３１３のＬＢＡが連続的に割り当てられているとき、リストを順に辿ることでアクセス先のＬＢＡに該当するスライスＩＤを取得する。
【０１１８】
ステップＳ１２０２では、ライト処理プログラム４２２は、ステップＳ１２０１で取得したスライスＩＤ（対象スライス）が自系ノードに割り当てられているか否かを判定する。なお、ライト処理プログラム４２２は、リード処理プログラム４２１で説明した方法と同様に判定する。ライト処理プログラム４２２は、対象スライスが自系ノードに割り当てられていると判定した場合、ステップＳ１２０５に処理を移し、対象スライスが自系ノードに割り当てられていないと判定した場合、ステップＳ１２０３に処理を移す。
【０１１９】
ステップＳ１２０３では、ライト処理プログラム４２２は、対象スライスを割り当てた先の他系ノードにライト処理要求を転送する。
【０１２０】
ステップＳ１２０４では、ライト処理プログラム４２２は、ステップＳ１２０３にて転送したライト処理要求の実行結果を待ち受け、実行結果を受信し、ステップＳ１２２４に処理を移す。
【０１２１】
ステップＳ１２０５では、ライト処理プログラム４２２は、アクセス先の領域に関しての排他制御を取得する。
【０１２２】
ステップＳ１２０６では、ライト処理プログラム４２２は、対象スライスの状態が移動中であるか否かを判定する。より具体的には、ライト処理プログラム４２２は、スライス管理テーブル８２０を参照し、アクセス先となるスライス３１４のスライスＩＤに対応する状態が、Ｍｉｇｒａｔｉｎｇである場合、移動中であると判定し、Ｍｉｇｒａｔｉｎｇでない場合、移動中でないと判定する。ライト処理プログラム４２２は、移動中であると判定した場合、ステップＳ１２０７に処理を移し、移動中でないと判定した場合、ステップＳ１２０９に処理を移す。

【０１２３】
ステップＳ１２０７では、ライト処理プログラム４２２は、対象スライスの移動先のノード１００（移動先ノード）にライト処理要求を転送する。
【０１２４】
ステップＳ１２０８では、ライト処理プログラム４２２は、ステップＳ１２０７にて転送したライト処理要求の実行結果を待ち受け、実行結果を受信し、ステップＳ１２０９に処理を移す。
【０１２５】
ステップＳ１２０９では、ライト処理プログラム４２２は、ライト処理要求のデータについて、ストレージプール３１２にページ３１５が未割当てであるか否かを判定する。ライト処理プログラム４２２は、未割当てであると判定した場合、ステップＳ１２１０に処理を移し、未割当てでないと判定した場合、ステップＳ１２１１に処理を移す。
【０１２６】
ステップＳ１２１０では、ライト処理プログラム４２２は、自系ノードのドライブ２１４の論理ドライブ３１８が関連付けられているプールボリューム３１６（自系プールボリューム）にページ３１５を割り当てる。
【０１２７】
ステップＳ１２１１では、ライト処理プログラム４２２は、アクセス先のアドレス（割当先アドレス）を取得する。
【０１２８】
ステップＳ１２１２では、ライト処理プログラム４２２は、書込み前のデータ（旧データ）を読み込む。ライト処理プログラム４２２は、読み込み先のドライブ２１４またはノード１００が障害状態である場合、リード処理プログラム４２１で説明したようにパリティから読み込み対象のデータを復元して、旧データを読み込む。
【０１２９】
ステップＳ１２１３では、ライト処理プログラム４２２は、中間データを生成する。中間データは、データを部分的に更新するときに作成する一時的なデータであり、新旧の差分を示すデータである。例えば、旧データのストライプが「Ａ１－Ａ２－ＡＰ」である場合、中間データは、次のように求められる。
【０１３０】
ＡＰ（旧パリティ）＝Ａ１（旧データ）　ＸＯＲ　Ａ２（旧データ）
Ａ１（新データ）　ＸＯＲ　Ａ１（旧データ）＝Ｍ（中間データ）
なお、新パリティについては、次のように求められる。
ＡＰ（旧パリティ）　ＸＯＲ　Ｍ（中間データ）＝ＡＰ（新パリティ）
【０１３１】
ステップＳ１２１４では、ライト処理プログラム４２２は、冗長化先のノード１００に中間データ（パリティ更新要求）を送信する。なお、ライト処理プログラム４２２は、冗長度に応じて（冗長度が２以上である場合、２以上のノード１００に）中間データを転送する。
【０１３２】
ステップＳ１２１５では、ライト処理プログラム４２２は、自系ノードのドライブ２１４に新データを書き込む。
【０１３３】
ステップＳ１２１６では、冗長化先のノード１００のライト処理プログラム４２２は、中間データを受信する。
【０１３４】
ステップＳ１２１７では、冗長化先のノード１００のライト処理プログラム４２２は、排他制御を取得する。
【０１３５】
ステップＳ１２１８では、冗長化先のノード１００のライト処理プログラム４２２は、

自系ノードのドライブ２１４から旧パリティを読み出す。
【０１３６】
ステップＳ１２１９では、冗長化先のノード１００のライト処理プログラム４２２は、中間データと旧パリティとから新パリティを計算する。
【０１３７】
ステップＳ１２２０では、冗長化先のノード１００のライト処理プログラム４２２は、自系ノードのドライブ２１４に新パリティを書き込む。
【０１３８】
ステップＳ１２２１では、冗長化先のノード１００のライト処理プログラム４２２は、取得した排他制御を解放し、中間データを転送してきたノード１００にパリティ更新結果を応答する。
【０１３９】
ステップＳ１２２２では、ライト処理プログラム４２２は、冗長化先のノード１００から書込み応答を受信する。
【０１４０】
ステップＳ１２２３では、ライト処理プログラム４２２は、取得した排他制御を解放する。
【０１４１】
ステップＳ１２２４では、ライト処理プログラム４２２は、ホスト１０１にライト処理結果を応答する。
【０１４２】
ステップＳ１２２５では、ライト処理プログラム４２２は、モニタ情報採取処理を実行する。なお、モニタ情報採取処理については、図１３Ｂを用いて後述する。
【０１４３】
図１３Ａは、プロセッサ２１１と、ドライブ２１４と、およびネットワーク２２０に関するモニタ情報採取処理に係るフローチャートの一例を示す図である。
【０１４４】
ステップＳ１３０１では、モニタ情報採取処理プログラム４２３は、プロセッサ２１１のモニタ情報をテーブルに登録する。より具体的には、プロセッサモニタ情報管理テーブル９１０にあるように、モニタ情報採取処理プログラム４２３は、ノード１００ごと、プロセスごと、プロセッサコアごとにプロセッサの使用率の情報を収集し、当該テーブルの情報を更新する。図示はしていないが、使用率以外の情報（ＩＤＬＥ、ＩＯＷＡＩＴ、ハイパーバイザ上の仮想マシンとして実行していれば、ＳＴＥＡＬ等）が取得され、テーブルに加えられてもよい。
【０１４５】
ステップＳ１３０２では、モニタ情報採取処理プログラム４２３は、ドライブ２１４のモニタ情報をテーブルに登録する。より具体的には、ドライブモニタ情報管理テーブル９２０にあるように、モニタ情報採取処理プログラム４２３は、ドライブ２１４ごとに、リードＩＯＰＳと、ライトＩＯＰＳと、リード転送量と、ライト転送量との情報を収集し、当該テーブルの情報を更新する。図示はしていないが、これら以外の情報（リード応答時間、ライト応答時間、キューサイズ等）が取得され、テーブルに加えられてもよい。
【０１４６】
ステップＳ１３０３では、モニタ情報採取処理プログラム４２３は、ネットワーク２２０（ＮＩＣ）のモニタ情報をテーブルに登録する。より具体的には、ネットワークモニタ情報管理テーブル９３０にあるように、モニタ情報採取処理プログラム４２３は、ノード１００ごとのＮＩＣごとに、送信転送量と、受信転送量と、最大転送量との情報を収集し、当該テーブルの情報を更新する。図示はしていないが、これら以外の情報（パケットドロップ数、再送パケット数等）が取得され、テーブルに加えられてもよい。
【０１４７】
ステップＳ１３０４では、モニタ情報採取処理プログラム４２３は、一定時間処理を停

止し、その後、ステップＳ１３０１に処理を移す。つまり、図１３Ａのモニタ情報採取処理は、周期的に実行される。

【０１４８】

図１３Ｂは、スライス３１４のアクセス頻度、およびフロントエンドパス３２０のアクセス頻度に関するモニタ情報採取処理に係るフローチャートの一例を示す図である。

【０１４９】

ステップＳ１３１１では、モニタ情報採取処理プログラム４２３は、Ｉ／Ｏを受信したフロントエンドパス３２０のモニタ情報とアクセス先のスライス３１４のモニタ情報とを取得する。より具体的には、モニタ情報採取処理プログラム４２３は、アクセスを受信したフロントエンドパス３２０に該当するフロントエンドパスモニタ情報管理テーブル９５０のレコード、およびアクセス先のスライス３１４に該当するスライスモニタ情報管理テーブル９４０のレコードを取得する。

【０１５０】

ステップＳ１３１２では、モニタ情報採取処理プログラム４２３は、受信したＩ／Ｏタイプは、リードであるか否かを判定する。モニタ情報採取処理プログラム４２３は、リードであると判定した場合、ステップＳ１３１３に処理を移し、リードでない（ライトである）と判定した場合、ステップＳ１３１５に処理を移す。

【０１５１】

ステップＳ１３１３では、モニタ情報採取処理プログラム４２３は、ステップＳ１３１１で取得したレコードの現行リードカウンタに受領したＩ／Ｏのカウントを加算する。ここで、ＩＯＰＳは、秒間当たりの処理量であるので、モニタ情報採取処理プログラム４２３は、１秒毎にカウンタ値を確定させる、つまり、１秒経過したときに発生したカウンタ値を計算することでＩＯＰＳを求め、ステップＳ１３１１で取得したレコードのリードＩＯＰＳに設定する。

【０１５２】

ステップＳ１３１４では、モニタ情報採取処理プログラム４２３は、ステップＳ１３１１で取得したレコードの現行リード転送量に受領したＩ／Ｏの転送量を加算する。

【０１５３】

ステップＳ１３１５では、モニタ情報採取処理プログラム４２３は、ステップＳ１３１１で取得したレコードの現行ライトカウンタに受領したＩ／Ｏのカウントを加算する。また、モニタ情報採取処理プログラム４２３は、１秒毎にカウンタ値を確定させる、つまり、１秒経過したときに発生したカウンタ値を計算することでＩＯＰＳを求め、ステップＳ１３１１で取得したレコードのライトＩＯＰＳに設定する。

【０１５４】

ステップＳ１３１６では、モニタ情報採取処理プログラム４２３は、ステップＳ１３１１で取得したレコードの現行ライト転送量に受領したＩ／Ｏの転送量を加算する。

【０１５５】

図１４は、リバランス要否判定処理に係るフローチャートの一例を示す図である。本処理は、ストレージシステム２００により周期的に実行されてもよいし、ユーザ（手動）により任意の契機に実行されてもよいし、リード処理またはライト処理の完了後に実行されてもよいし、後述のクラスタ構成変更処理の実行後に実行されてもよい。

【０１５６】

ステップＳ１４０１では、リバランス要否判定処理プログラム４２４は、ストレージプール３１２の使用率が上限閾値以上のノード１００が存在するか否かを判定する。リバランス要否判定処理プログラム４２４は、ストレージプール３１２の使用率が上限閾値以上のノード１００が存在すると判定した場合、ステップＳ１４０５に処理を移し、存在しないと判定した場合、ステップＳ１４０２に処理を移す。

【０１５７】

ステップＳ１４０２では、リバランス要否判定処理プログラム４２４は、プロセッサ２１１の使用率が上限閾値以上のノード１００が存在するか否かを判定する。リバランス要

否判定処理プログラム４２４は、プロセッサ２１１の使用率が上限閾値以上のノード１００が存在すると判定した場合、ステップＳ１４０５に処理を移し、存在しないと判定した場合、ステップＳ１４０３に処理を移す。

【０１５８】

ステップＳ１４０３では、リバランス要否判定処理プログラム４２４は、ドライブ２１４の使用率が上限閾値以上のノード１００が存在するか否かを判定する。リバランス要否判定処理プログラム４２４は、ドライブ２１４の使用率が上限閾値以上のノード１００が存在すると判定した場合、ステップＳ１４０５に処理を移し、存在しないと判定した場合、ステップＳ１４０４に処理を移す。

【０１５９】

ステップＳ１４０４では、リバランス要否判定処理プログラム４２４は、ネットワーク２２０（ＮＩＣ）の使用率が上限閾値以上のノード１００が存在するか否かを判定する。リバランス要否判定処理プログラム４２４は、ネットワーク２２０の使用率が上限閾値以上のノード１００が存在すると判定した場合、ステップＳ１４０５に処理を移し、存在しないと判定した場合、ステップＳ１４０６に処理を移す。

【０１６０】

ステップＳ１４０５では、リバランス要否判定処理プログラム４２４は、リソース割当決定処理（分散ポリシ）を実行する。なお、リソース割当決定処理（分散ポリシ）については、図１５を用いて後述する。

【０１６１】

ステップＳ１４０６では、リバランス要否判定処理プログラム４２４は、プロセッサ２１１の使用率が下限閾値未満のノード１００が存在するか否かを判定する。リバランス要否判定処理プログラム４２４は、プロセッサ２１１の使用率が下限閾値未満のノード１００が存在すると判定した場合、ステップＳ１４０９に処理を移し、存在しないと判定した場合、ステップＳ１４０７に処理を移す。

【０１６２】

ステップＳ１４０７では、リバランス要否判定処理プログラム４２４は、ドライブ２１４の使用率が下限閾値未満のノード１００が存在するか否かを判定する。リバランス要否判定処理プログラム４２４は、ドライブ２１４の使用率が下限閾値未満のノード１００が存在すると判定した場合、ステップＳ１４０９に処理を移し、存在しないと判定した場合、ステップＳ１４０８に処理を移す。

【０１６３】

ステップＳ１４０８では、リバランス要否判定処理プログラム４２４は、ネットワーク２２０（ＮＩＣ）の使用率が下限閾値未満のノード１００が存在するか否かを判定する。リバランス要否判定処理プログラム４２４は、ネットワーク２２０の使用率が下限閾値未満のノード１００が存在すると判定した場合、ステップＳ１４０９に処理を移し、存在しないと判定した場合、処理を終了する。

【０１６４】

ステップＳ１４０９では、リバランス要否判定処理プログラム４２４は、リソース割当決定処理（集約ポリシ）を実行する。なお、リソース割当決定処理（集約ポリシ）については、図１６を用いて後述する。

【０１６５】

なお、ステップＳ１４０２～ステップＳ１４０４、および、ステップＳ１４０６～ステップＳ１４０８は、ノード１００の負荷を判定するものであり、ノード１００に仮想ボリューム３１３が１つ設けられている場合は、仮想ボリューム３１３の負荷を判定するものでもある。

【０１６６】

また、図１４で説明したリソース割当決定処理を実行するか否かを判定するためのメトリクスは、プロセッサ２１１、ドライブ２１４、ポート２１５以外にも仮想ボリューム３１３に対するＩＯＰＳおよび／または転送量を用いてもよい。ＩＯＰＳまたは転送量をメ

トリクスとして用いる場合、リードとライトとで異なる閾値を設けて判定してもよい。
【０１６７】
　また、各ノード１００は、自系ノードの負荷を低減するために、自系ノードが提供している仮想ボリューム３１３のうち、領域を移動していない仮想ボリューム３１３を他系ノードに移動してもよい。これにより、移動している領域がある仮想ボリューム３１３に、当該領域を戻すことが（当該領域を移動）できる場合がある。
【０１６８】
　図１５は、分散ポリシに基づくリソース割当決定処理に係るフローチャートの一例を示す図である。
【０１６９】
　ステップＳ１５０１では、リソース割当決定処理プログラム４２５は、各メトリクス（プロセッサ２１１、ドライブ２１４、ネットワーク２２０等）が上限閾値以上のノード１００を移動元のノード１００（移動元ノード）に選択し、移動対象とする仮想ボリューム３１３を選択する。言い換えると、リソース割当決定処理プログラム４２５は、プロセッサ２１１、ドライブ２１４、ネットワーク２２０の負荷に余裕のないノード１００を選択し、データの移動元（ここでは分散元ともいえる）の仮想ボリューム３１３を選択する。
【０１７０】
　より具体的には、リソース割当決定処理プログラム４２５は、リバランス要否判定処理のステップＳ１４０２、ステップＳ１４０３、またはステップＳ１４０４で選択されたノード１００に定義された仮想ボリューム３１３を選択する。例えば、リソース割当決定処理プログラム４２５は、仮想ボリューム３１３を選択する際、ストレージプログラム３１１と同じノード１００にアプリケーション３０１が動作するＨＣＩ（Ｈｙｐｅｒ－Ｃｏｎｖｅｒｇｅｄ　Ｉｎｆｒａｓｔｒａｃｔｕｒｅ）構成をとっている場合、同じノード１００内のアプリケーション３０１が使用している仮想ボリューム３１３を避けて選択する。これは、同じノード１００内のアプリケーション３０１が使用している仮想ボリューム３１３を選択し、分散させるとアプリケーション３０１は、ネットワーク２２０を介してデータへアクセスすることになり、処理効率が低下するためである。
【０１７１】
　付言するならば、ステップＳ１４０３、またはステップＳ１４０４で選択されたノード１００に定義された仮想ボリューム３１３が１つである場合、リソース割当決定処理プログラム４２５は、当該仮想ボリューム３１３を選択する。
【０１７２】
　ステップＳ１５０２では、リソース割当決定処理プログラム４２５は、仮想ボリューム３１３またはストレージシステム２００に設定された分散ポリシを判定する。リソース割当決定処理プログラム４２５は、分散ポリシがボリューム単位分散ポリシであると判定した場合、ステップＳ１５０３に処理を移し、分散ポリシがスライス単位最大分散ポリシ（スライス単位均等分散ポリシ）であると判定した場合、ステップＳ１５０６に処理を移し、分散ポリシがスライス単位最小分散ポリシであると判定した場合、ステップＳ１５０９に処理を移す。
【０１７３】
　ボリューム単位分散ポリシは、仮想ボリューム３１３単位で負荷が分散されるポリシである。ボリューム単位分散ポリシでは、仮想ボリューム３１３単位でまとめてスライス３１４が移動される。ボリューム単位分散ポリシでは、仮想ボリューム３１３単位でスライス３１４が移動されるため、常にデータの集約が保たれた状態で負荷分散を行うことができる。
【０１７４】
　スライス単位最大分散ポリシは、スライス３１４単位で負荷が分散されるポリシである。スライス単位最大分散ポリシでは、仮想ボリューム３１３に設定された最大分散度のノード数だけスライス３１４が分散される。スライス単位最大分散ポリシでは、最大分散度でスライス３１４が分散されるため、高負荷なノード１００（仮想ボリューム３１３）を

迅速に負荷分散することができる。
【０１７５】
スライス単位最小分散ポリシは、スライス３１４単位で負荷が分散されるポリシである。スライス単位最小分散ポリシでは、仮想ボリューム３１３内のスライス３１４が１つずつ分散されていく。スライス単位最小分散ポリシでは、１つずつスライス３１４が分散されていくことで、データのローカリティを可能な限り保ちつつ、最小限の負荷だけを高負荷なノード１００（仮想ボリューム３１３）から逃がすことで過負荷状態を回避する。
【０１７６】
これらの分散ポリシは、ユーザが仮想ボリューム３１３に対して、事前に設定してもよいし、ストレージシステム２００が状況に応じて分散ポリシを自動で選択してもよい。ストレージシステム２００が自動でポリシを選択する方法の一例として、基本的には、ストレージシステム２００は、ボリューム単位分散ポリシを適用しておき、仮想ボリューム３１３が１つのノード１００の性能で不足する場合に、スライス単位最大分散ポリシまたはスライス単位最小分散ポリシに切り替える。加えて、ストレージシステム２００は、仮想ボリューム３１３の負荷が突発的に高くなった場合は、スライス単位最大散ポリシを適用し、仮想ボリューム３１３の負荷が緩やかに高くなった場合は、スライス単位最小分散ポリシを適用する。
【０１７７】
ステップＳ１５０３、ステップＳ１５０４、およびステップＳ１５０５では、リソース割当決定処理プログラム４２５は、仮想ボリューム３１３単位でスライス３１４を移動するための前処理を行う。
【０１７８】
ステップＳ１５０３では、リソース割当決定処理プログラム４２５は、選択した仮想ボリューム３１３内の全てのスライス３１４をスライスグループとしてグルーピングする。例えば、リソース割当決定処理プログラム４２５は、スライス管理テーブル８２０から移動対象のスライスＩＤをメモリ２１２上にリストとして格納する。
【０１７９】
ステップＳ１５０４では、リソース割当決定処理プログラム４２５は、計算したスライスグループを移動対象として設定する。
【０１８０】
ステップＳ１５０５では、リソース割当決定処理プログラム４２５は、移動先ノード数を「１ノード」に設定する。
【０１８１】
ステップＳ１５０６、ステップＳ１５０７、およびステップＳ１５０８では、リソース割当決定処理プログラム４２５は、仮想ボリューム３１３に設定された最大分散度でスライス３１４を移動するための前処理を行う。
【０１８２】
ステップＳ１５０６では、リソース割当決定処理プログラム４２５は、選択した仮想ボリューム３１３内の全てのスライス３１４を仮想ボリューム３１３に設定された最大分散度で分割し、スライスグループとしてグルーピングする。例えば、リソース割当決定処理プログラム４２５は、グルーピングする際、選択した仮想ボリューム３１３にアクセスするホスト１０１が複数存在する場合で、かつ、ホスト１０１ごとにアクセス対象のスライス３１４に偏り（ローカリティ）がある場合、ホスト１０１ごとのアクセス対象のスライス３１４をグルーピングする。また、例えば、リソース割当決定処理プログラム４２５は、スライスモニタ情報管理テーブル９４０を確認し、仮想ボリューム３１３内の全てのスライス３１４を最大分散度のノード１００で負荷が均等になるようにグルーピングしてもよい。
【０１８３】
ステップＳ１５０７では、リソース割当決定処理プログラム４２５は、計算したスライスグループを移動対象として設定する。

【０１８４】
ステップＳ１５０８では、リソース割当決定処理プログラム４２５は、移動先ノード数を最大分散度と同値に設定する。
【０１８５】
ステップＳ１５０９、ステップＳ１５１０、およびステップＳ１５１１では、リソース割当決定処理プログラム４２５は、最小分散度（つまり１スライス）でスライス３１４を移動するための前処理を行う。
【０１８６】
ステップＳ１５０９では、リソース割当決定処理プログラム４２５は、選択した仮想ボリューム３１３内からスライス３１４を１つ選択する。
【０１８７】
ステップＳ１５１０では、リソース割当決定処理プログラム４２５は、選択したスライス３１４を移動対象として設定する。
【０１８８】
ステップＳ１５１１では、リソース割当決定処理プログラム４２５は、移動先ノード数を「１ノード」に設定する。
【０１８９】
ステップＳ１５１２では、リソース割当決定処理プログラム４２５は、移動対象を１つ選択する。前処理で、スライスグループが作られた場合は、リソース割当決定処理プログラム４２５は、スライスグループを移動対象として選択し、スライス３１４がそのまま選択された場合は、スライス３１４を移動対象として選択する。
【０１９０】
ステップＳ１５１３では、リソース割当決定処理プログラム４２５は、移動先とするノード１００を１つ選択する。リソース割当決定処理プログラム４２５は、移動先ノードの選択方法の一例として、移動元ノードを除き、各メトリクス（プロセッサ２１１、ドライブ２１４、ネットワーク２２０等）の負荷に余裕のあるノード１００を選択する方法がある。
【０１９１】
ステップＳ１５１４では、リソース割当決定処理プログラム４２５は、選択した移動対象を移動先ノードに移動した場合の移動先ノードの各メトリクス（プロセッサ２１１、ドライブ２１４、ネットワーク２２０等）の負荷を計算する。
【０１９２】
ステップＳ１５１５では、リソース割当決定処理プログラム４２５は、ステップＳ１５１４で計算した移動先ノードの各メトリクス（プロセッサ２１１、ドライブ２１４、ネットワーク２２０等）の負荷が閾値を超過していないかを判定する。リソース割当決定処理プログラム４２５は、１つでも閾値を超過しているメトリクスがあると判定した場合、ステップＳ１５１３に処理を移し、全てのメトリクスにおいて閾値を超過していないと判定した場合、ステップＳ１５１６に処理を移す。なお、各メトリクス（プロセッサ２１１、ドライブ２１４、ネットワーク２２０等）の上限閾値とステップＳ１５１５の各メトリクスの閾値とは、同じであってもよいし、異なっていてもよい。
【０１９３】
ステップＳ１５１６では、リソース割当決定処理プログラム４２５は、ステップＳ１５１５で判定したノード１００を移動先ノードとして選択する。
【０１９４】
ステップＳ１５１７では、リソース割当決定処理プログラム４２５は、全ての移動対象に対して判定を完了したか否かを判定する。リソース割当決定処理プログラム４２５は、全ての移動対象に対して判定が完了している場合、ステップＳ１５１８に処理を移し、全ての移動対象に対して判定が完了していない場合、ステップＳ１５１２に処理を移す。
【０１９５】
ステップＳ１５１８では、リソース割当決定処理プログラム４２５は、移動対象として

スライスグループまたはスライス３１４を対象にして、リソース移動処理を実行する。リソース移動処理については、図１７を用いて後述する。

【０１９６】

ステップＳ１５１９では、リソース割当決定処理プログラム４２５は、閾値を超過していたノード１００（仮想ボリューム３１３）の負荷が閾値未満となったか否かを判定する。リソース割当決定処理プログラム４２５は、閾値未満でないと判定した場合、ステップＳ１５０１に処理を移し、閾値未満であると判定した場合、処理を終了する。

【０１９７】

図１６は、集約ポリシに基づくリソース割当決定処理に係るフローチャートの一例を示す図である。

【０１９８】

ステップＳ１６０１では、リソース割当決定処理プログラム４２５は、各メトリクス（プロセッサ２１１、ドライブ２１４、ネットワーク２２０等）が下限閾値未満のノードを移動先ノードに選択する。言い換えると、リソース割当決定処理プログラム４２５は、プロセッサ２１１、ドライブ２１４、ネットワーク２２０等の負荷に余裕のあるノード１００（仮想ボリューム３１３）を選択し、データの移動先（ここでは集約先ともいえる）のノード１００とする。例えば、リソース割当決定処理プログラム４２５は、リバランス要否判定処理のステップＳ１４０６、ステップＳ１４０７、またはステップＳ１４０８で選択されたノード１００を選択する。

【０１９９】

ステップＳ１６０２では、リソース割当決定処理プログラム４２５は、ステップＳ１６０１で選択した移動先ノードに、スライス３１４が分散した仮想ボリューム３１３が存在するか否かを判定する。リソース割当決定処理プログラム４２５は、スライス３１４が分散した仮想ボリューム３１３が存在すると判定した場合、ステップＳ１６０４に処理を移し、スライス３１４が分散した仮想ボリューム３１３が存在しないと判定した場合、ステップＳ１６０３に処理を移す。

【０２００】

ステップＳ１６０３では、リソース割当決定処理プログラム４２５は、ステップＳ１６０１で選択した移動先ノード以外のノード１００に、スライス３１４が分散した仮想ボリューム３１３が存在するか否かを判定する。リソース割当決定処理プログラム４２５は、スライス３１４が分散した仮想ボリューム３１３が存在すると判定した場合、ステップＳ１６０４に処理を移し、スライス３１４が分散した仮想ボリューム３１３が存在しないと判定した場合、集約対象（移動対象）の仮想ボリューム３１３は存在しないため、処理を終了する。

【０２０１】

上述したように、ステップＳ１６０２とステップＳ１６０３とでは、リソース割当決定処理プログラム４２５は、移動対象の仮想ボリューム３１３を選択する。リソース割当決定処理プログラム４２５は、選択する際、移動先ノードとして選択したノード１００上の仮想ボリューム３１３を優先的に選択することで、集約時のスライス３１４の移動量を削減する効果がある。

【０２０２】

ステップＳ１６０４では、リソース割当決定処理プログラム４２５は、選択した仮想ボリューム３１３を移動対象に選択する。

【０２０３】

付言するならば、リソース割当決定処理プログラム４２５は、仮想ボリューム３１３の負荷に余裕のあるノード１００があり、当該仮想ボリューム３１３のスライス３１４が分散している場合、当該ノード１００を移動先ノードとし、当該仮想ボリューム３１３のスライス３１４が分散している仮想ボリューム３１３を移動対象に選択することがある。

【０２０４】

ステップＳ１６０５では、リソース割当決定処理プログラム４２５は、移動対象ボリュ

ーム内のスライス３１４を１つ選択する。選択の方法としては、例えば、スライスモニタ情報管理テーブル９４０を参照し、移動対象ボリューム内のスライス３１４から処理負荷の高いスライス３１４を選択する。

【０２０５】

ステップＳ１６０６では、リソース割当決定処理プログラム４２５は、選択したスライス３１４を移動先ノードに移動した場合の各メトリクス（プロセッサ２１１、ドライブ２１４、ネットワーク２２０等）の負荷を計算する。

【０２０６】

ステップＳ１６０７では、リソース割当決定処理プログラム４２５は、ステップＳ１６０６で計算したスライス３１４を移動後の各メトリクス（プロセッサ２１１、ドライブ２１４、ネットワーク２２０等）の負荷が閾値を超過していないかを判定する。リソース割当決定処理プログラム４２５は、１つでも閾値を超過しているメトリクスがあると判定した場合、処理を終了し、全てのメトリクスにおいて閾値を超過していないと判定した場合、ステップＳ１６０８に処理を移す。

【０２０７】

ステップＳ１６０８では、リソース割当決定処理プログラム４２５は、移動対象のスライス３１４を対象にして、移動対象の仮想ボリューム３１３を提供するノード１００にリソース移動処理の実行を要求する。リソース移動処理については、図１７を用いて後述する。

【０２０８】

ステップＳ１６０９では、リソース割当決定処理プログラム４２５は、移動対象ボリューム内の全てのスライス３１４に対して、移動するか否かの判定を行ったかを判定する。リソース割当決定処理プログラム４２５は、全てのスライス３１４に対して移動するか否かの判定を行った場合、処理を終了し、全てのスライス３１４に対して移動するか否かの判定を行っていない場合、ステップＳ１６０５に処理を移し、まだ未判定のスライス３１４に対して処理を行う。

【０２０９】

図１７は、リソース移動処理に係るフローチャートの一例を示す図である。リソース移動処理では、処理対象のスライス３１４について、現在割当たっているストレージプール３１２から別ノード１００のストレージプール３１２へ割当先が移動される。リソース移動処理プログラム４２６は、スライス３１４の割当先を移動するにあたり、スライス３１４に書き込まれたデータを移動元のストレージプール３１２から読み出し、移動先のストレージプール３１２へ書き出す。

【０２１０】

ステップＳ１７０１では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の状態を移動中に更新する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０から移動対象のスライス３１４のレコードを取得し、状態８２４の値を「Ｍｉｇｇｒａｔｉｎｇ」に更新する。

【０２１１】

ステップＳ１７０２では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の先頭オフセットを取得する。より具体的には、リソース移動処理プログラム４２６は、ページマッピングテーブル８３０を参照し、移動対象のスライス３１４のスライスＩＤに該当するレコードを参照し、プールボリュームＩＤとプールボリュームＬＢＡとを取得する。次に、リソース移動処理プログラム４２６は、プールボリューム管理テーブル６１０を参照し、取得したプールボリュームＩＤに該当するレコードを参照し、論理ドライブＩＤを取得する。次に、リソース移動処理プログラム４２６は、論理ドライブ管理テーブル６３０を参照し、取得した論理ドライブＩＤに該当するレコードを取得し、ドライブＩＤと開始オフセットを取得し、先に取得しているプールボリュームＬＢＡからアクセス先のドライブ２１４のアドレスを求める。例えば、リソース移動処理プログラム４２６は、以下のように計算する。

【０２１２】
アクセス先アドレス ＝ 開始オフセット ＋ プールボリュームＬＢＡ
【０２１３】
ステップＳ１７０３では、リソース移動処理プログラム４２６は、アクセス先の領域の排他制御を取得する。
【０２１４】
ステップＳ１７０４では、リソース移動処理プログラム４２６は、処理対象のオフセットのデータについて、ストレージプール３１２にページ３１５が未割当てであるか否かを判定する。リソース移動処理プログラム４２６は、未割当てであると判定した場合、ステップＳ１７０７に処理を移し、未割当てでないと判定した場合、ステップＳ１７０５に処理を移す。
【０２１５】
ステップＳ１７０５では、リソース移動処理プログラム４２６は、取得したオフセットに該当するスライス３１４の移動元のストレージプール３１２の領域にリードを発行する。より具体的には、リード処理が実行される。なお、リード処理の動作は、図１１で説明したとおりである。
【０２１６】
ステップＳ１７０６では、リソース移動処理プログラム４２６は、取得したオフセットに該当するスライス３１４の移動先のストレージプール３１２の領域にステップＳ１７０５で読み取ったデータでライトを発行する。ライト先として、移動元のスライス３１４に対してライトを発行する。移動元のスライス３１４では、ライト処理プログラム４２２内にてステップＳ１２０６の分岐で、Ｙｅｓの判定となり、移動先のスライス３１４へデータの書き込みがなされる。ただし、ステップＳ１７０６のライト発行先は、移動先のスライス３１４としてもよい。なお、ライト処理の動作は、図１２で説明したとおりである。
【０２１７】
ステップＳ１７０７では、リソース移動処理プログラム４２６は、アクセス先の領域の排他制御を解放する。
【０２１８】
ステップＳ１７０８では、リソース移動処理プログラム４２６は、現在処理対象としているアクセス先のオフセットが移動対象のスライス３１４の終端オフセットであるか否かを判定する。リソース移動処理プログラム４２６は、終端オフセットであると判定した場合、ステップＳ１７１０に処理を移し、終端オフセットでないと判定した場合、ステップＳ１７０９に処理を移す。
【０２１９】
ステップＳ１７０９では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の次のオフセットを取得する。
【０２２０】
ステップＳ１７１０では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の制御情報に関する排他制御を取得する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０のアクセス先のスライス３１４のスライスＩＤに該当するレコードの排他制御を取得する。
【０２２１】
ステップＳ１７１１では、リソース移動処理プログラム４２６は、スライス３１４を割当てるストレージプール３１２の情報を更新する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０のストレージプールＩＤ８２３の情報を、移動元のストレージプールＩＤから移動先のストレージプールＩＤに更新する。
【０２２２】
ステップＳ１７１２では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の制御情報に関する排他制御を解放する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０のアクセス先のスライス３１４のスライス

ＩＤに該当するレコードの排他制御を解放する。
【０２２３】
ステップＳ１７１３では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の状態を正常に更新する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０から移動対象のスライス３１４のレコードを取得し、状態８２４の値を「Ｎｏｒｍａｌ」に更新する。
【０２２４】
ステップＳ１７１４では、リソース移動処理プログラム４２６は、移動対象のスライス３１４を全て移動したか否かを判定する。リソース移動処理プログラム４２６は、全ての移動対象のスライス３１４を移動していない判定した場合、ステップＳ１７０１に処理を移し、全ての移動対象のスライス３１４を移動したと判定した場合、処理を終了する。
【０２２５】
図１８は、フロントエンドパス設定処理に係るフローチャートの一例を示す図である。
【０２２６】
ステップＳ１８０１では、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、ストレージプログラム３１１にスライス３１４の移動があったか否かを問合せる。
【０２２７】
ステップＳ１８０２では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、スライス３１４の移動の有無を判定し、パス設定プログラム３０２に応答する。より具体的には、ストレージプログラム３１１は、現在時刻までのスライス３１４の移動記録と、パス設定プログラム３０２から前回問い合わせられたときの時刻情報とをログとして保存しておき、パス設定プログラム３０２からの前回の問い合わせ時刻以降に、スライス３１４の移動記録が更新されているか確認する。ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、スライス３１４の移動記録が更新されている場合、移動があったと判定し、更新がなかった場合、移動はなかったと判定し、判定結果をパス設定プログラム３０２に応答する。
【０２２８】
ステップＳ１８０３では、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、ストレージプログラム３１１からの応答を受信する。
【０２２９】
ステップＳ１８０４では、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、ストレージプログラム３１１からの応答内容に基づき、スライス３１４の移動があったか否かを判定する。パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、スライス３１４の移動があったと判定した場合、ステップＳ１８０５に処理を移し、スライス３１４の移動がなかったと判定した場合、処理を終了する。
【０２３０】
ステップＳ１８０５では、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、移動したスライス３１４の情報をストレージプログラム３１１に問い合わせる。
【０２３１】
ステップＳ１８０６では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、移動したスライス３１４と、当該スライス３１４を含む仮想ボリューム３１３との情報をパス設定プログラム３０２に応答する。
【０２３２】
ステップＳ１８０７では、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、ストレージプログラム３１１からの応答を受信する。
【０２３３】
ステップＳ１８０８では、パス設定プログラム３０２のフロントエンドパス設定処理プ

ログラム４２７は、移動したスライス３１４の移動先ノード（移動したスライス３１４を含む仮想ボリューム３１３）にパス３２０が設定されているか否かを確認する。パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、フロントエンドパス３２０を設定済みであると判定した場合、処理を終了し、フロントエンドパス３２０を未設定であると判定した場合、ステップＳ１８０９に処理を移す。

【０２３４】

ステップＳ１８０９では、フロントエンドパス設定処理プログラム４２７は、移動したスライス３１４を含む仮想ボリューム３１３について、当該スライス３１４の移動先ノードとのフロントエンドパス３２０の確立を要求する。パス設定プログラム３０２とストレージプログラム３１１と間での具体的なパス確立の手順は、ｉＳＣＳＩ（Ｉｎｔｅｒｎｅｔ Ｓｍａｌｌ Ｃｏｍｐｕｔｅｒ Ｓｙｓｔｅｍ Ｉｎｔｅｒｆａｃｅ）、Ｆｉｂｅｒ Ｃｈａｎｎｅｌのプロトコル等に基づく。

【０２３５】

ステップＳ１８１０では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、パス設定プログラム３０２から要求された仮想ボリューム３１３とホスト１０１とのパス情報をフロントエンドパス情報テーブル１０１０に登録し、パス設定プログラム３０２に応答する。

【０２３６】

ステップＳ１８１１では、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、ストレージプログラム３１１からパス設定完了の応答を受信する。

【０２３７】

図１９は、ＳＣＳＩ（Ｓｍａｌｌ Ｃｏｍｐｕｔｅｒ Ｓｙｓｔｅｍ Ｉｎｔｅｒｆａｃｅ）規格のＡＬＵＡ（Ａｓｙｍｍｅｔｒｉｃ Ｌｏｇｉｃａｌ Ｕｎｉｔ Ａｃｃｅｓｓ）のメカニズムに基づきフロントエンドパス３２０の設定を最適化する際のフロントエンドパス設定処理に係るフローチャートの一例を示す図である。

【０２３８】

ステップＳ１９０１では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、処理対象の仮想ボリューム３１３を選択する。より具体的には、処理対象の仮想ボリューム３１３については、定義された全ての仮想ボリューム３１３に対して周期的に選択してもよいし、リード処理またはライト処理の実行の完了後に、当該処理においてアクセスのあった仮想ボリューム３１３を選択してもよい。

【０２３９】

ステップＳ１９０２では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、仮想ボリューム３１３に接続されたホスト１０１を処理対象として選択する。より具体的には、処理対象のホスト１０１については、処理対象の仮想ボリューム３１３に定義された全てのホスト１０１を選択してもよいし、リード処理またはライト処理の実行の完了後に、当該処理においてアクセスのあったホスト１０１を選択してもよい。

【０２４０】

ステップＳ１９０３では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、選択した仮想ボリューム３１３と選択したホスト１０１と間のパスの情報を取得する。より具体的には、フロントエンドパス設定処理プログラム４２７は、フロントエンドパス情報テーブル１０１０を参照し、選択した仮想ボリューム３１３の仮想ボリュームＩＤと、選択したホスト１０１のＩｎｉｔｉａｔｏｒＩＤとに該当するレコードを取得する。

【０２４１】

ステップＳ１９０４では、フロントエンドパス設定処理プログラム４２７は、ステップＳ１９０３で取得したフロントエンドパス３２０へのＩ／Ｏの発行比率を計算する。より具体的には、フロントエンドパス設定処理プログラム４２７は、フロントエンドパスモニタ情報管理テーブル９５０のリードＩＯＰＳ９５２とライトＩＯＰＳ９５３とを参照し、

パスＡにリードおよびライトのＩＯＰＳが合計９００ＩＯＰＳ発行されており、パスＢにリードおよびライトのＩＯＰＳが合計１００ＩＯＰＳ発行されている場合、パスＡとパスＢのＩ／Ｏ発行比率を、パスＡ：パスＢ＝９：１と計算する。加えて、フロントエンドパス設定処理プログラム４２７は、同様にリードおよびライトの転送量についても、比率を計算し、ＩＯＰＳと転送量とで各パスのＩ／Ｏの発行比率の分散が大きい方を最終的な比率として採用してもよいし、ＩＯＰＳと転送量との比率の平均値を最終的な比率として採用してもよい。

【０２４２】

ステップＳ１９０５では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、Ｉ／Ｏの発行比率が特定のフロントエンドパス３２０（ホスト１０１）に偏っているか否かを判定する。例えば、フロントエンドパス設定処理プログラム４２７は、各フロントエンドパス３２０へのＩ／Ｏ発行比率の分散を計算し、分散が閾値を上回っている場合に偏りが発生していると判定する。ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、特定のフロントエンドパス３２０にＩ／Ｏの発行比率が偏っていると判定した場合、ステップＳ１９０６に処理を移し、特定のフロントエンドパス３２０にＩ／Ｏの発行比率が偏っていないと判定した場合、ステップＳ１９０７に処理を移す。

【０２４３】

ステップＳ１９０６では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、Ｉ／Ｏの発行比率が高いフロントエンドパス３２０を「Ｏｐｔｉｍｉｚｅ（最適パス）」に設定する。例えば、フロントエンドパス設定処理プログラム４２７は、フロントエンドパス情報テーブル１０１０を参照し、Ｉ／Ｏの発行比率が閾値より高いフロントエンドパス３２０のパスＩＤと一致するレコードのＡＬＵＡ設定１０１４を「Ｏｐｔｉｍｉｚｅ」に更新し、Ｉ／Ｏの発行比率が閾値より高くないフロントエンドパス３２０のパスＩＤと一致するレコードのＡＬＵＡ設定１０１４を、「Ｎｏｎ－Ｏｐｔｉｍｉｚｅ」に更新する。

【０２４４】

ステップＳ１９０７では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、選択した仮想ボリューム３１３に定義された全てのフロントエンドパス３２０について「Ｏｐｔｉｍｉｚｅ（最適パス）」に設定（いわゆるラウンドロビンに設定）する。

【０２４５】

ステップＳ１９０８では、ストレージプログラム３１１のフロントエンドパス設定処理プログラム４２７は、パス設定プログラム３０２に、「Ｏｐｔｉｍｉｚｅ」に設定したフロントエンドパス３２０の情報（最適化情報）を通知する。図１９では、ストレージプログラム３１１からパス設定プログラム３０２へ通知を発行する形式で情報を伝達するが、パス設定プログラム３０２からストレージプログラム３１１に問い合わせる形式で情報を伝達してもよい。

【０２４６】

ステップＳ１９０９では、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、ストレージプログラム３１１からフロントエンドパス３２０の最適化情報を受領する。

【０２４７】

ステップＳ１９１０では、パス設定プログラム３０２のフロントエンドパス設定処理プログラム４２７は、ストレージプログラム３１１から受信したフロントエンドパス３２０を「Ｏｐｔｉｍｉｚｅ（最適パス）」に設定する。以降、アプリケーション３０１は、最適パスに対して優先的にＩ／Ｏを発行するように制御される。

【０２４８】

図２０は、クラスタ構成変更処理に係るフローチャートの一例を示す図である。

【０２４９】

ステップＳ２００１では、クラスタ構成変更処理プログラム４２８は、ユーザからのクラスタ操作要求を受信する。

【０２５０】

ステップＳ２００２では、クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求が仮想ボリューム３１３の作成または削除の要求であるか否かを判定する。クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求が仮想ボリューム３１３の作成または削除の要求であると判定した場合、ステップＳ２００７に処理を移し、受信したクラスタ操作要求が仮想ボリューム３１３の作成または削除の要求でないと判定した場合、ステップＳ２００３に処理を移す。

【０２５１】

ステップＳ２００３では、クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求が仮想ボリューム３１３の拡張または縮小の要求であるか否かを判定する。クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求が仮想ボリューム３１３の拡張または縮小の要求であると判定した場合、ステップＳ２００７に処理を移し、受信したクラスタ操作要求が仮想ボリューム３１３の拡張または縮小の要求でないと判定した場合、ステップＳ２００４に処理を移す。

【０２５２】

ステップＳ２００４では、クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求がドライブ２１４の増設または減設の要求であるか否かを判定する。クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求がドライブ２１４の増設または減設の要求であると判定した場合、ステップＳ２００７に処理を移し、受信したクラスタ操作要求がドライブ２１４の増設または減設の要求でないと判定した場合、ステップＳ２００５に処理を移す。

【０２５３】

ステップＳ２００５では、クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求がノード１００の増設または減設の要求であるか否かを判定する。クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求がノード１００の増設または減設の要求であると判定した場合、ステップＳ２００７に処理を移し、受信したクラスタ操作要求がノード１００の増設または減設の要求でないと判定した場合、ステップＳ２００６に処理を移す。

【０２５４】

ステップＳ２００６では、クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求がサイト２０１の増設または減設の要求であるか否かを判定する。クラスタ構成変更処理プログラム４２８は、受信したクラスタ操作要求がサイト２０１の増設または減設の要求であると判定した場合、ステップＳ２００７に処理を移し、受信したクラスタ操作要求がサイト２０１の増設または減設の要求でないと判定した場合、ステップＳ２００１に処理を移す。

【０２５５】

ステップＳ２００７では、クラスタ構成変更処理プログラム４２８は、ユーザから要求された操作を実行する。

【０２５６】

ステップＳ２００８では、クラスタ構成変更処理プログラム４２８は、リバランス要否判定処理を実行する。なお、リバランス要否判定処理については、図１４を用いて説明したので、その説明を省略する。

【０２５７】

図２１Ａは、ユーザが操作可能なボリュームの設定画面（ＧＵＩ：Ｇｒａｐｈｉｃａｌ Ｕｓｅｒ Ｉｎｔｅｒｆａｃｅ）の一例を示す図である。ボリューム設定画面２１０１は、ストレージシステム２００と通信可能に接続された所定の計算機（例えば、管理サーバ等）に出力される。

【０２５８】

ボリューム設定画面２１０１は、仮想ボリューム３１３ごとの設定が行われる画面である。ボリューム設定画面２１０１は、ボリュームＩＤ２１０２および分散度２１０３の情報を設定可能に構成される。

【０２５９】

ボリュームＩＤ２１０２は、設定対象とする仮想ボリューム３１３を指定する項目である。分散度２１０３は、設定対象の仮想ボリューム３１３のスライス３１４を分散するノード数の最大値の情報（最大分散度）を設定可能な項目である。

【０２６０】

図２１Ｂは、ストレージシステム２００がユーザに提示するボリューム性能予測画面２１１１（ＧＵＩ）の一例を示す図である。ボリューム性能予測画面２１１１は、ストレージシステム２００と通信可能に接続された所定の計算機（例えば、管理サーバ等）に出力される。

【０２６１】

ボリューム性能予測画面２１１１は、ボリューム性能予測情報２１１２とメッセージ２１２０とを含んで構成される。

【０２６２】

ボリューム性能予測情報２１１２は、確認対象の仮想ボリューム３１３について、最大分散度とスループット（ＩＯＰＳ）と応答時間とに関する情報をユーザに提示する。ボリューム性能予測情報２１１２は、分散度情報２１１３、スループット情報２１１４、応答時間情報２１１５、予測スループット情報２１１６、予測応答時間情報２１１７、現行分散度情報２１１８、および目標性能値情報２１１９を含んで構成される。

【０２６３】

分散度情報２１１３は、最大分散度に関する情報を含む。スループット情報２１１４は、ＩＯＰＳに関する情報を含む。応答時間情報２１１５は、応答時間に関する情報を含む。予測スループット情報２１１６は、該当する最大分散度におけるスループットの予測値に関する情報を含む。予測応答時間情報２１１７は、該当する最大分散度における応答時間の予測値に関する情報を含む。現行分散度情報２１１８は、確認対象の仮想ボリューム３１３に現在設定されている最大分散度に関する情報を含む。目標性能値情報２１１９は、確認対象の仮想ボリューム３１３に現在設定されている目標とする（ユーザが期待する）スループットおよび応答時間の性能値に関する情報を含む。

【０２６４】

メッセージ２１２０は、現行分散度情報２１１８に基づく予測応答時間情報２１１７が目標性能値情報２１１９を超過する場合、または、現行分散度情報２１１８に基づく予測スループット情報２１１６が目標性能値情報２１１９未満となる場合、ストレージシステム２００は、ユーザに対して、現在の最大分散度では目標性能値を達成できない旨を提示する。また、メッセージ２１２０は、現在の設定値から最大分散度を変更したときに予測性能が目標性能値を満たせるか否かをユーザに提示する。

【０２６５】

本実施の形態によれば、１つのボリュームについて容量および性能をスケールアウトすることができる。

【０２６６】

（ＩＩ）第２の実施の形態

図２２は、本実施の形態のストレージシステム２２００に係る構成の一例を示す図である。第１の実施の形態と同じ構成については、同じ符号を用いて説明を省略する。

【０２６７】

ストレージシステム２２００は、図２の構成に加えて、ドライブボックス２２０１を含んで構成される。ドライブボックス２２０１は、例えば、プロセッサ２１１、メモリ２１２等を含む１以上のプロセッサパッケージ２１３、１以上のドライブ２１４、１以上のポート２１５を含んで構成される。各構成要素は、内部バス２１６を介して接続されている。

【０２６８】

ポート２１５は、ネットワーク２２０２に接続され、サイト２０１内のノード１００と通信可能に接続されている。ネットワーク２２０２は、例えば、ＬＡＮ、ＳＡＮ（Ｓｔｏｒａｇｅ Ａｒｅａ Ｎｅｔｗｏｒｋ）、ＳＡＳ（Ｓｅｒｉａｌ Ａｔｔａｃｈｅｄ ＳＣＳＩ）であるが、これらに限定するものではない。

【０２６９】

プロセッサ２１１では、ドライブボックス２２０１に対する、Ｉ／Ｏ処理プログラムが動作しており、必要に応じて、データの圧縮処理、ドライブボックス２２０１内のドライブ２１４に対するＲＡＩＤ処理等を実施してもよい。また、ドライブボックス２２０１は、上記に加えて、専用のハードウェアであるＡＳＩＣ（Ａｐｐｌｉｃａｔｉｏｎ Ｓｐｅｃｉｆｉｃ Ｉｎｔｅｇｒａｔｅｄ Ｃｉｒｃｕｉｔ）が搭載されていてもよく、ＡＳＩＣは、データ圧縮処理、パリティ演算処理等を備えていてもよい。

【０２７０】

図２３は、本実施の形態におけるリソース移動処理に係るフローチャートの一例を示す図である。リソース移動処理では、リバランス要否判定処理の結果を受けて、リバランスが必要となった場合にノード１００間でスライス３１４を移動する。

【０２７１】

第１の実施の形態との違いについて説明する。本実施の形態では、スライス３１４内のデータは、ノード１００間で共有されたドライブボックス２２０１に配置される。このため、ノード１００間でスライス３１４を移動する場合も仮想ボリューム３１３とスライス３１４の割り当てとに関する制御情報を更新するだけでよく、データ自体は移動する必要はない。以下、詳細について説明する。

【０２７２】

ステップＳ２３０１では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の状態を移動中に更新する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０から移動対象のスライス３１４のレコードを取得し、状態８２４の値を「Ｍｉｇｇｒａｔｉｎｇ」に更新する。

【０２７３】

ステップＳ２３０２では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の制御情報に関する排他制御を取得する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０のアクセス先のスライス３１４のスライスＩＤに該当するレコードの排他制御を取得する。

【０２７４】

ステップＳ２３０３では、リソース移動処理プログラム４２６は、スライス３１４を割当てるストレージプール３１２の情報を更新する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０のストレージプールＩＤ８２３の情報を、移動元のストレージプールＩＤから移動先のストレージプールＩＤに更新する。

【０２７５】

ステップＳ２３０４では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の制御情報に関する排他制御を解放する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０のアクセス先のスライス３１４のスライスＩＤに該当するレコードの排他制御を解放する。

【０２７６】

ステップＳ２３０５では、リソース移動処理プログラム４２６は、移動対象のスライス３１４の状態を正常に更新する。より具体的には、リソース移動処理プログラム４２６は、スライス管理テーブル８２０から移動対象のスライス３１４のレコードを取得し、状態８２４の値を「Ｎｏｒｍａｌ」に更新する。

【０２７７】

本実施の形態によれば、データを移動することなく、１つのボリュームについて性能をスケールアウトすることができる。

【０２７８】

（ＩＩＩ）付記
上述の実施の形態には、例えば、以下のような内容が含まれる。
【０２７９】
上述の実施の形態においては、本発明をストレージシステムに適用するようにした場合について述べたが、本発明はこれに限らず、この他種々のシステム、装置、方法、プログラムに広く適用することができる。
【０２８０】
また、上述の実施の形態においては、スライスモニタ情報管理テーブル９４０は、スライスＩＤ９４１と、リードカウンタ９４２と、ライトカウンタ９４３と、リード転送量９４４と、ライト転送量９４５と、モニタ開始時刻９４６とが対応付けられた情報を格納する場合について述べたが、本発明はこれに限らない。例えば、スライスモニタ情報管理テーブル９４０は、スライスＩＤ９４１と、リードＩＯＰＳと、ライトＩＯＰＳと、リード転送量９４４と、ライト転送量９４５とが対応付けられた情報を格納するようにしてもよい。
【０２８１】
また、上述の実施の形態においては、フロントエンドパスモニタ情報管理テーブル９５０は、パスＩＤ９５１と、リードＩＯＰＳ９５２と、ライトＩＯＰＳ９５３と、リード転送量９５４と、ライト転送量９５５とが対応付けられた情報を格納する場合について述べたが、本発明はこれに限らない。例えば、フロントエンドパスモニタ情報管理テーブル９５０は、パスＩＤ９５１と、リードカウンタと、ライトカウンタと、リード転送量９５４と、ライト転送量９５５と、モニタ開始時刻とが対応付けられた情報を格納するようにしてもよい。
【０２８２】
また、上述の実施の形態において、各テーブルの構成は一例であり、１つのテーブルは、２以上のテーブルに分割されてもよいし、２以上のテーブルの全部または一部が１つのテーブルであってもよい。
【０２８３】
また、上述の実施の形態において、図示および説明した画面は、一例であり、受け付ける情報が同じであるならば、どのようなデザインであってもよい。
【０２８４】
また、上述の実施の形態において、図示および説明した画面は、一例であり、提示する情報が同じであるならば、どのようなデザインであってもよい。
【０２８５】
また、上述の実施の形態において、統計値として分散および平均値を用いる場合について説明したが、統計値は、分散および平均値に限るものではなく、最大値、最小値、最大値と最小値との差、最頻値、中央値、標準偏差等の他の統計値であってもよい。
【０２８６】
上述した実施の形態は、例えば、以下の特徴的な構成を有する。
【０２８７】
（１）
複数の領域を含むボリューム（例えば、仮想ボリューム３１３）を１以上のホスト（例えば、ホスト１０１）に提供するための処理を行うプロセッサ（例えば、プロセッサ２１１、プロセッサパッケージ２１３）を備える複数のノード（例えば、ノード１００）と、上記プロセッサと接続され、上記ボリュームのデータを記憶する１以上の記憶デバイス（例えば、ドライブ２１４、ドライブボックス２２０１）とを備えるストレージシステム（例えば、ストレージシステム２００、ストレージシステム２２００）は、上記複数のノードの各々は、自ノードが提供するボリュームの負荷および上記ボリュームの領域を複数に分割した領域の負荷を監視し、監視している一のボリュームの負荷が閾値以上であると判定した第１のノードは、上記一のボリュームの領域を複数に分割した領域の負荷と負荷分散のポリシ（ボリューム単位分散ポリシ、スライス単位最大分散ポリシ、スライス単位最

小分散ポリシ等）とに応じて、上記一のボリュームに含まれる一部の領域を上記第１のノードとは異なる第２のノードのボリュームに移動する（例えば、図１５参照）。
【０２８８】
上記ボリュームの負荷は、例えば、自ノードのプロセッサの使用率、自ノードの記憶デバイスへのＩ／Ｏ量、自ノードの記憶デバイスへのＩ／Ｏレスポンス、自ノードのネットワークインターフェースへのＩ／Ｏ量、自ノードのネットワークインターフェースへのＩ／Ｏレスポンス、ボリュームに対するＩＯＰＳ、および、ボリュームに対する転送量の少なくとも１つである。
【０２８９】
上記構成では、一のボリュームの負荷が高まったときに、当該ボリュームの一部の領域が他のノードのボリュームに移動されるので、例えば、ノードを追加した場合、一のボリュームについても性能をスケールアウトすることができるようになる。
【０２９０】
（２）
上記第１のノードは、上記一のボリュームの負荷が閾値未満であると判定した場合（例えば、図１４参照）、上記第２のノードのボリュームから、上記一部の領域を上記一のボリュームに移動する（例えば、図１６参照）。
【０２９１】
上記構成では、例えば、領域が移動されているボリュームの負荷が低い場合、当該ボリュームに当該領域が集約されるので、当該ボリュームのスループットを向上させ、当該ボリュームのレイテンシを低下させることができる。
【０２９２】
（３）
上記第１のノードは、上記第２のノードとして、上記一部の領域を移動した後のボリュームの負荷が閾値を超えないノードを選択する（例えば、ステップＳ１５１４～ステップＳ１５１６参照）。
【０２９３】
上記構成では、例えば、移動先ノードのボリュームが過負荷になってしまい、更に移動しなければならない事態を回避することができる。
【０２９４】
（４）
上記複数のノードの各々には、上記１以上の記憶デバイスの少なくとも１つが対応して設けられ（例えば、図２参照）、上記複数のノードの各々は、自ノードに割り当てられている領域のデータを、自ノードに設けられている記憶デバイスに記憶し（例えば、図１２参照）、上記第１のノードは、上記一のボリュームの容量が提供可能な容量を超えると判定した場合（例えば、ステップＳ１４０１参照）、上記一部の領域を上記第２のノードのボリュームに移動する。
【０２９５】
上記構成によれば、例えば、各ノードが提供するボリュームは、ストレージシステムが備える記憶デバイス分、容量を利用できるようになる。
【０２９６】
（５）
上記複数のノードの各々は、自ノードが提供するボリュームに対するリードの負荷（例えば、リードＩＯＰＳ、リード転送量、リードカウンタ等）と、上記ボリュームに対するライトの負荷（例えば、ライトＩＯＰＳ、ライト転送量、ライトカウンタ等）とを監視し、上記第１のノードは、上記一のボリュームに対するリードの負荷が第１の閾値以上であると判定した場合、上記一部の領域を上記第２のノードのボリュームに移動し、上記一のボリュームに対するライトの負荷が上記第１の閾値とは異なる第２の閾値以上であると判定した場合、上記一部の領域を上記第２のノードのボリュームに移動する。
【０２９７】

ボリュームに対するリードとライトとでは、ノードにかかる負荷が異なるが、上記構成では、それぞれに対して監視を行い、別の閾値を設けることで、例えば、より適切にボリュームの負荷を判定することができる。

【０２９８】

（６）

上記第１のノードは、上記複数のノードに対して上記一のボリュームに含まれる領域を均等に割り振り（例えば、ステップＳ１５０６、ステップＳ１５０７）、上記第１のノードとは異なるノードに割り振った領域を、上記ノードのボリュームに移動する。

【０２９９】

上記構成では、例えば、移動する領域による負荷が均等になるようにボリュームの領域を移動することができる。

【０３００】

（７）

上記第１のノードは、上記一のボリュームの負荷が上記閾値を下回るまで、上記一のボリュームに含まれる領域を１つずつ（例えば、ステップＳ１５０９、ステップＳ１５１０）、上記第１のノードとは異なるノードのボリュームに移動する。

【０３０１】

上記構成では、例えば、データのローカリティを極力保ち、最小限の負荷をボリュームから逃すことができる。

【０３０２】

（８）

上記一のボリュームが、複数のホストに提供されている場合、上記第１のノードは、上記複数のホストの各々がアクセスする領域をまとめてホストごとの移動対象とし（例えば、ステップＳ１５０６、ステップＳ１５０７）、ホストごとの移動対象の領域を、上記第１のノードとは異なるノードのボリュームに移動する。

【０３０３】

上記構成では、例えば、ホストごとにアクセスする領域が違う場合に、ホストがアクセスする領域をまとめて移動することができる。

【０３０４】

（９）

上記第１のノードは、自ノードが提供しているボリュームのうち上記一のボリュームとは異なる他のボリュームを上記第１のノードとは異なるノードに移動し、上記第２のノードのボリュームから、上記一部の領域を上記一のボリュームに移動する。

【０３０５】

上記構成では、例えば、第１のノードは、一のボリュームとは異なる他のボリュームをまるごと別のノードに移動することにより、移動した領域を集めることができる場合がある。

【０３０６】

（１０）

上記一部の領域が移動された上記第２のノードのボリュームと、上記一部の領域にアクセスするホストとの間にパスが設定されていない場合、上記第２のノードおよび上記ホストは、上記パスを設定する（例えば、図１８参照）。

【０３０７】

上記構成では、領域が移動された第２のノードのボリュームと当該領域にアクセスするホストとの間にパスが設定されていない場合、パスが設定されるので、例えば、当該領域に対するアクセスがあった際、第１のノードを介することなくデータをやり取りすることができるようになる。

【０３０８】

（１１）

上記複数のノードの各々は、自ノードのボリュームの負荷が特定のホストに偏っている

と判定した場合、上記特定のホストとのパスが最適属性であることを上記特定のホストに通知する（例えば、ステップＳ１９０４～ステップＳ１９０６、ステップＳ１９０８）。
【０３０９】
上記構成によれば、例えば、ボリュームの負荷が特定のホストに偏っている場合、特定のホストに優先的にＩ／Ｏが発行されるようになる。
【０３１０】
（１２）
上記複数のノードの各々は、自ノードのボリュームの負荷が上記特定のホストに偏っていないと判定した場合、上記ボリュームに定義されている全てのパスが最適属性であることを上記パスが設定されている全てのホストに通知する（例えば、ステップＳ１９０４、ステップＳ１９０５、ステップＳ１９０７、ステップＳ１９０８）。
【０３１１】
上記構成によれば、ボリュームの負荷が特定のホストに偏っていない場合、当該ボリュームにアクセスする全てのホストに均等にＩ／Ｏが発行されるようになる。
【０３１２】
（１３）
上記１以上の記憶デバイス（例えば、ドライブボックス２２０１）は、上記複数のノードに共通して設けられている。
【０３１３】
上記構成では、ノードのコンピュート部分と記憶部分とが分かれたことにより、例えば、プロセッサの使用率が余っているが、記憶デバイスの容量が不足している場合、コンピュート部分を増やすことなく、記憶部分を増やすことができる。
【０３１４】
（１４）
上記第１のノードは、上記第２のノードに上記一部の領域を移動する際、上記一部の領域を管理するためのデータ（例えば、スライス管理テーブル８２０）を更新し、上記１以上の記憶デバイスに記憶される上記一部の領域のデータを移動しない（例えば、図２３参照）。
【０３１５】
上記構成では、どのノードからも、均等にデータにアクセスできるようになっているので、例えば、ノード間でボリュームの負荷を分散するときは、ボリュームのオーナー権（排他権、メタデータ）を移すだけで、ボリュームの負荷を分散できる。
【０３１６】
（１５）
上記複数のノードが提供するボリュームについて、当該ボリュームの領域の移動先ノードの数に応じたスループットと応答時間とを計算して出力する計算機（例えば、管理サーバ）を備える。
【０３１７】
上記構成によれば、例えば、ユーザは、領域の移動先ノードの数（例えば、最大分散度）を容易に決定することができる。
【０３１８】
上記複数のノードの各々は、ボリュームに含まれる領域単位で、ボリュームの負荷を監視する（例えば、図１３Ｂ）。
【０３１９】
上記ストレージシステムは、１つのボリュームが分散可能な最大のノードの数をユーザが指定するためのＧＵＩを出力する計算機を備える（図２１Ａ）。
【０３２０】
また上述した構成については、本発明の要旨を超えない範囲において、適宜に、変更したり、組み替えたり、組み合わせたり、省略したりしてもよい。
【０３２１】

「Ａ、Ｂ、およびＣのうちの少なくとも１つ」という形式におけるリストに含まれる項目は、（Ａ）、（Ｂ）、（Ｃ）、（ＡおよびＢ）、（ＡおよびＣ）、（ＢおよびＣ）または（Ａ、Ｂ、およびＣ）を意味することができると理解されたい。同様に、「Ａ、Ｂ、またはＣのうちの少なくとも１つ」の形式においてリストされた項目は、（Ａ）、（Ｂ）、（Ｃ）、（ＡおよびＢ）、（ＡおよびＣ）、（ＢおよびＣ）または（Ａ、Ｂ、およびＣ）を意味することができる。

【符号の説明】

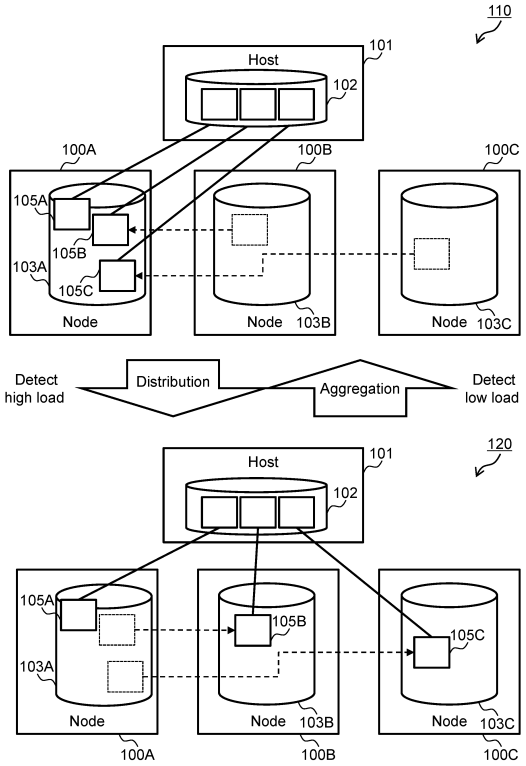
【０３２２】

１００……ノード、１０１……ホスト、２００……ストレージシステム。

【図面】

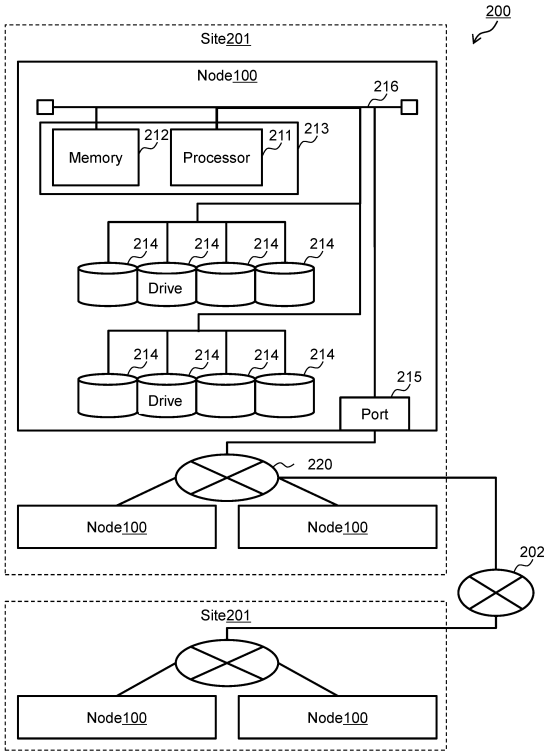
【図１】

図 1



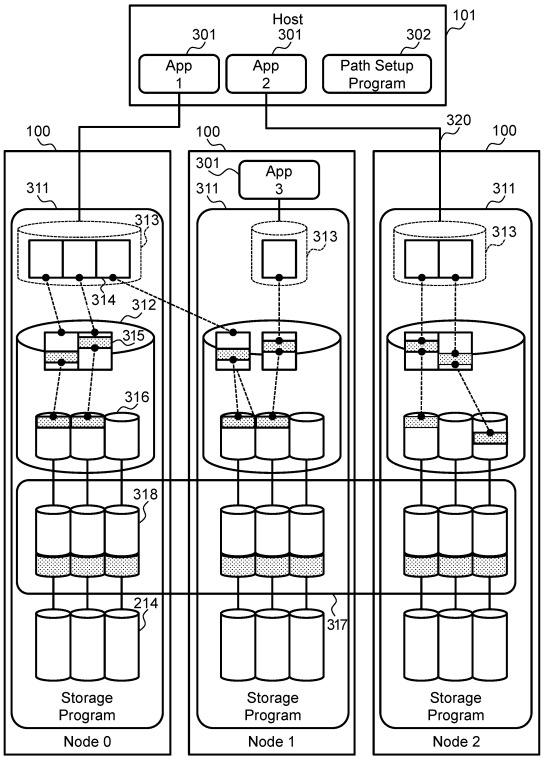
【図２】

図 2



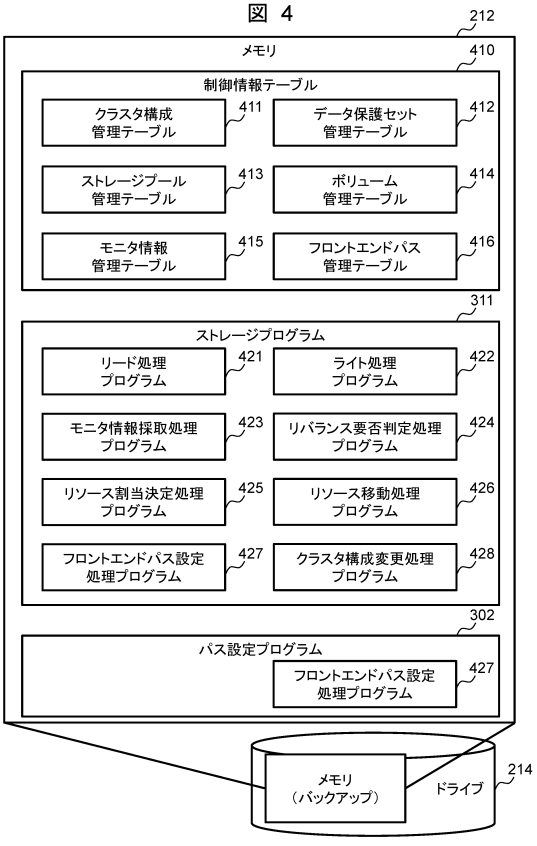
【図３】

図 3



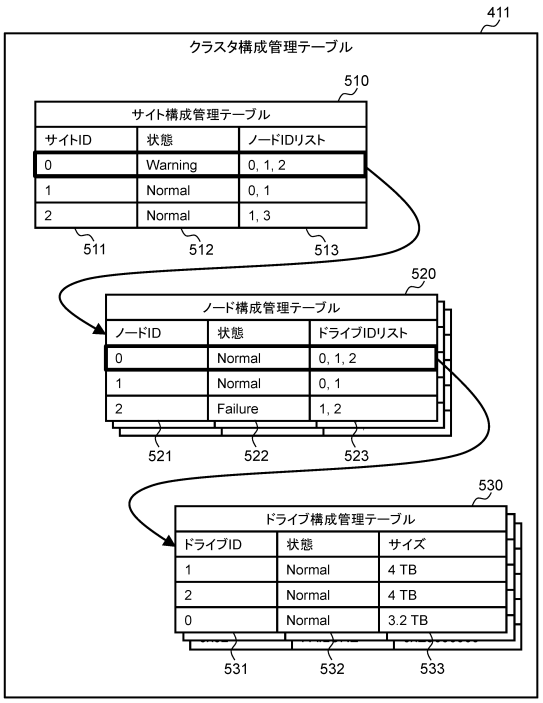
【図４】

図 4



【図５】

図 5

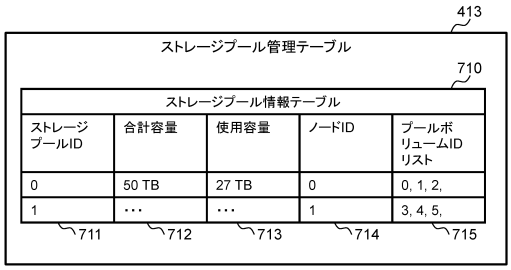


【図６】

図 6

【図７】

図 7



【図８】

図 8

【図９】

図 9

415

モニタ情報管理テーブル 910

プロセッサモニタ情報管理テーブル

ノードID	プロセッサID	プロセスID	プロセス名	使用率
0	0	2304	Storage Program	50%
0	1	2401	App1	20%

911 912 913 914 915

920

ドライブモニタ情報管理テーブル

ドライブID	リード IOPS	ライト IOPS	リード 転送量	ライト 転送量	使用率
0	100	50	800 KB/s	200 KB/s	20%
1	200	600	1.6 MB/s	2.4 MB/s	50%

921 922 923 924 925 926

930

ネットワークモニタ情報管理テーブル

ノードID	NIC ID	送信転送量	受信転送量	最大転送量
0	0	1000 bps	2000 bps	10 Gbps
1	1	5000 bps	6000 bps	25Gbps

921 932 933 934 935

940

スライスモニタ情報管理テーブル

スライスID	リード カウンタ	ライト カウンタ	リード 転送量	ライト 転送量	モニタ 開始時刻
0	100	500	300 MB	2000 MB	00:00:00
1	1000	20	5000 MB	50 KB	00:00:00

941 942 943 944 945 946

950

フロントエンドパスモニタ情報管理テーブル

パスID	リードIOPS	ライトIOPS	リード転送量	ライト転送量
0	100	50	800 KB/s	200 KB/s
1	200	600	1.6 MB/s	2.4 MB/s

951 952 953 954 955

【図１０】

図 10

416

フロントエンドパス管理テーブル 1010

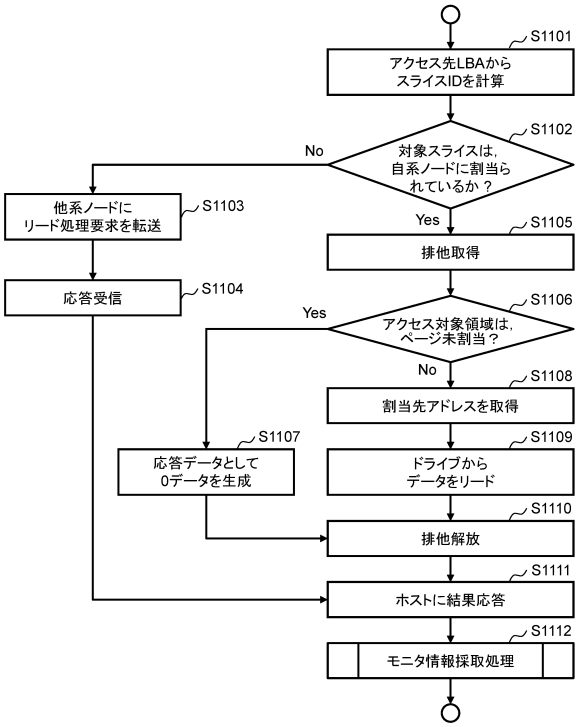
フロントエンドパス情報テーブル

パスID	仮想 ボリュームID	Initiator ID	ALUA設定	接続 ノードID
0	0	iqn-xxxx	Optimize	0
1	0	iqn-yyy	Non-Optimize	1
2	1	iqn-zzz	Optimize	0

1011 1012 1013 1014 1015

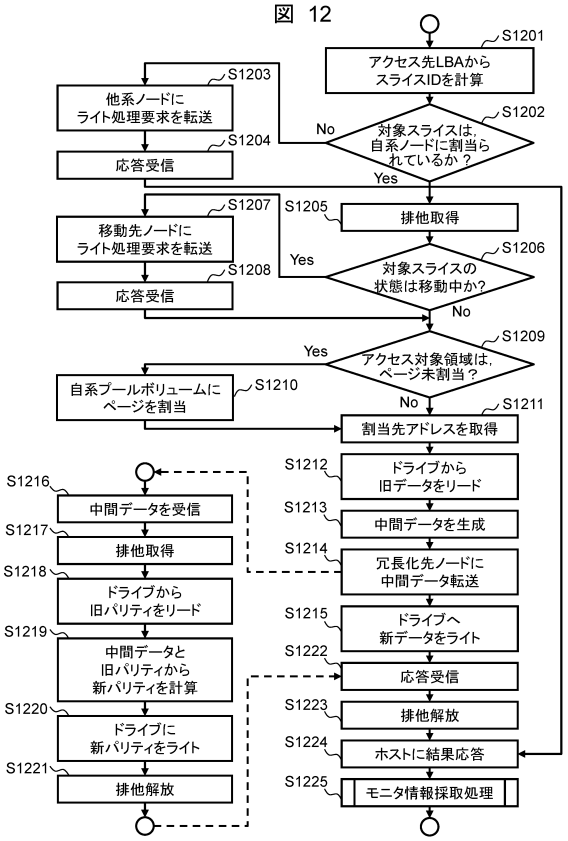
【図１１】

図 11



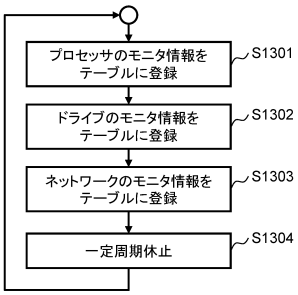
【図１２】

図 12



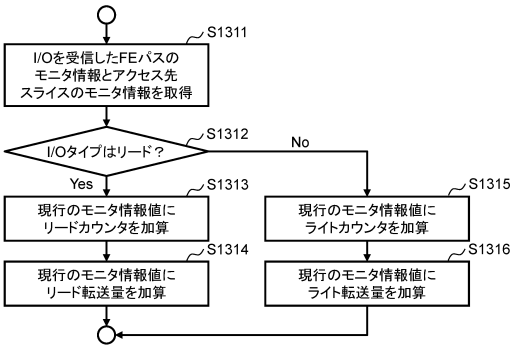
【図１３Ａ】

図 13A



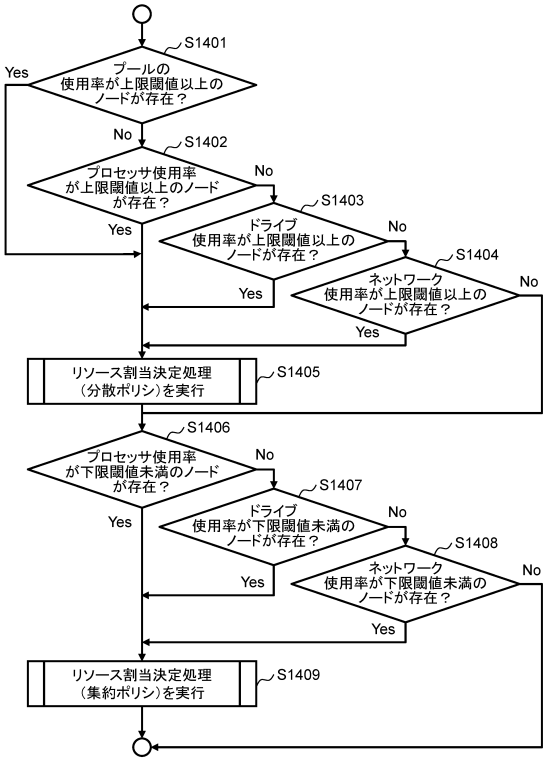
【図１３Ｂ】

図 13B



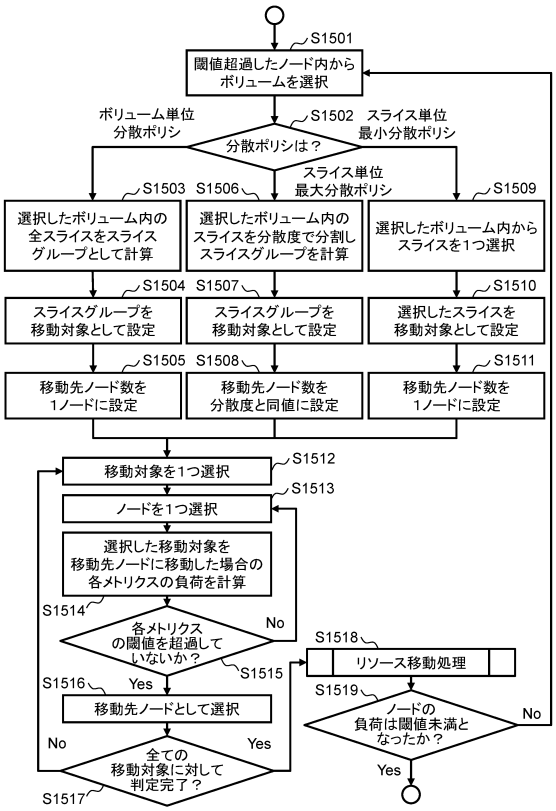
【図１４】

図 14



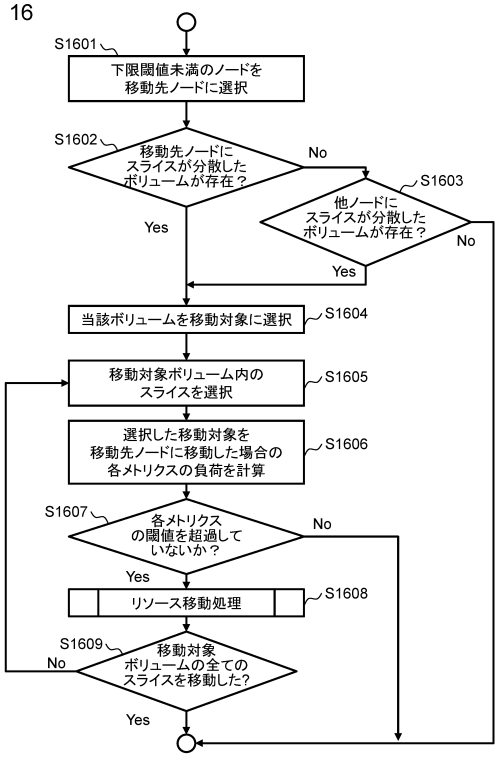
【図１５】

図 15



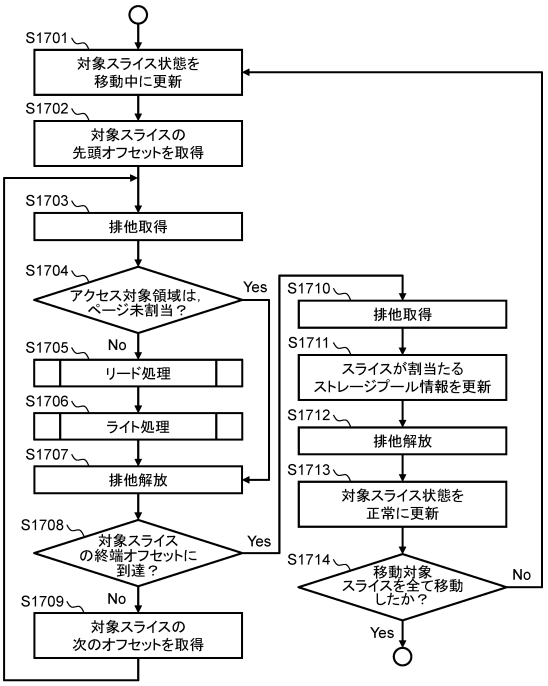
【図１６】

図 16



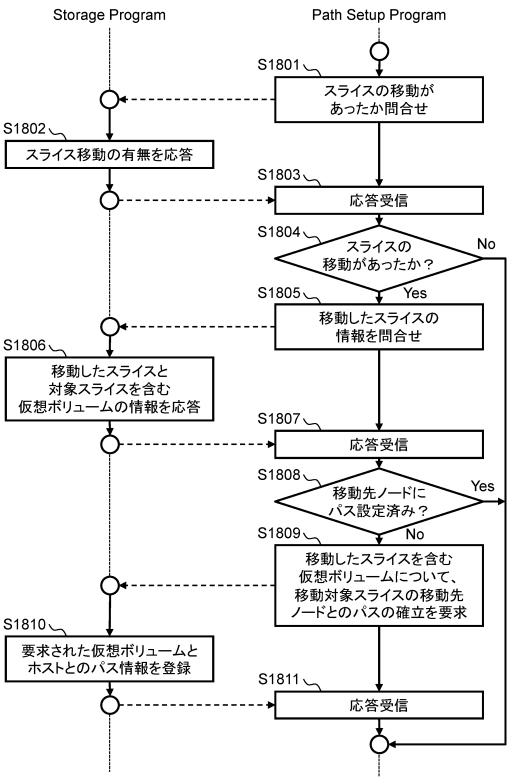
【図１７】

図 17



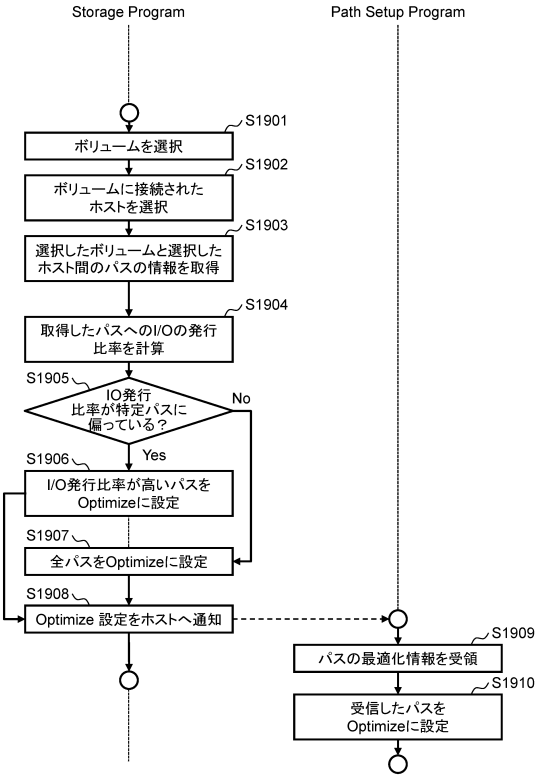
【図１８】

図 18



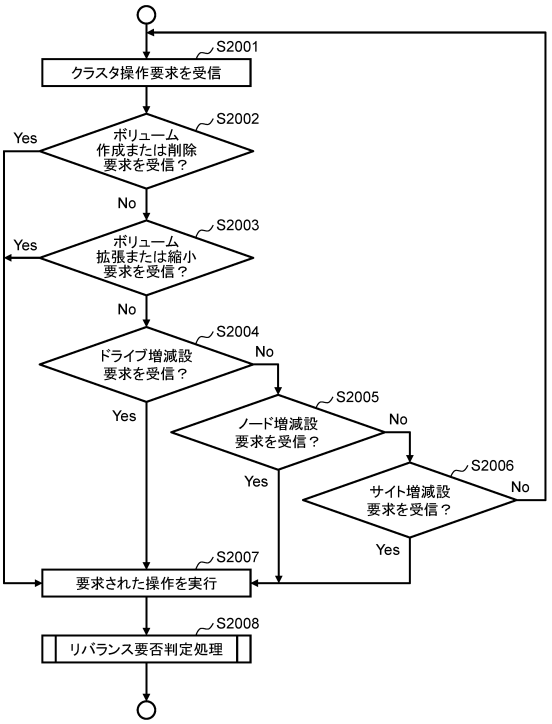
【図１９】

図 19



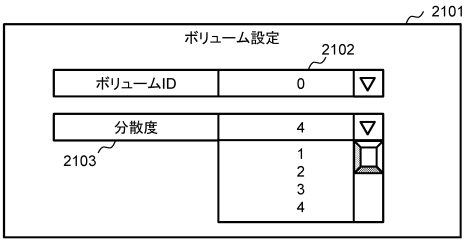
【図２０】

図 20



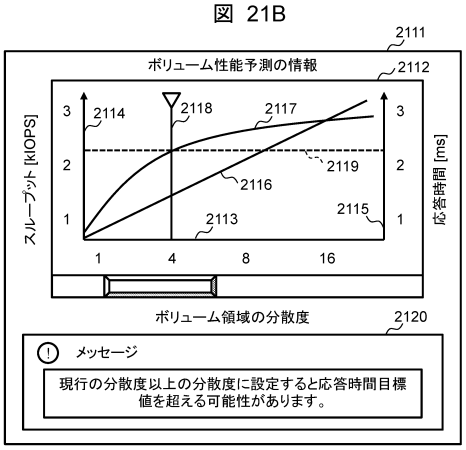
【図２１Ａ】

図 21A



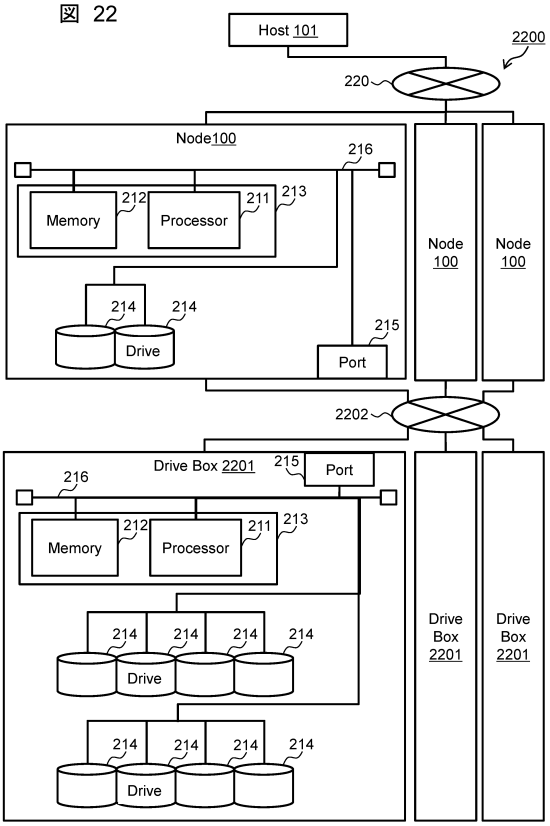
【図２１Ｂ】

図 21B



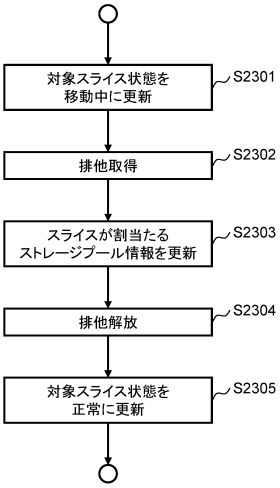
【図２２】

図 22



【図２３】

図 23


対象スライス状態を
移動中に更新
S2301
排他取得
S2302
スライスが割当たる
ストレージプール情報を更新
S2303
排他解放
S2304
対象スライス状態を
正常に更新
S2305

フロントページの続き

(51)国際特許分類　　ＦＩ
　　Ｇ０６Ｆ　　13/10　　３４０Ａ

東京都千代田区丸の内一丁目６番６号 株式会社日立製作所内

審査官　　打出 義尚

(56)参考文献　　特開２０１０－２７７２８９（ＪＰ，Ａ）
特開２０１６－２４５６３（ＪＰ，Ａ）
特開２０２０－４６９２９（ＪＰ，Ａ）
米国特許出願公開第２０１９／０２６５９１５（ＵＳ，Ａ１）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　３／０６
Ｇ０６Ｆ　１３／１０
Ｇ０６Ｆ　１１／０７
Ｇ０６Ｆ　１１／３４