(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-272138
(P2010-272138A)
(43)公開日　平成22年12月2日(2010.12.2)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 3/06	(2006.01)	G06F	3/06	301Z	5B065

審査請求 有　請求項の数 23　ＯＬ　（全 42 頁）

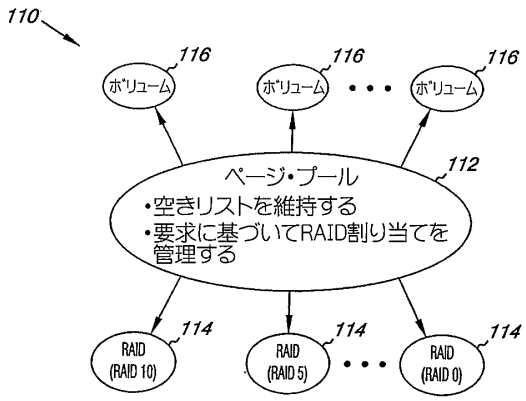
(21)出願番号　特願2010-186320 (P2010-186320)
(22)出願日　平成22年8月23日 (2010.8.23)
(62)分割の表示　特願2006-523434 (P2006-523434)の分割
原出願日　平成16年8月13日 (2004.8.13)
(31)優先権主張番号　60/495,204
(32)優先日　平成15年8月14日 (2003.8.14)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　10/918,329
(32)優先日　平成16年8月13日 (2004.8.13)
(33)優先権主張国　米国 (US)

(71)出願人　506051681
コンペレント・テクノロジーズ
アメリカ合衆国ミネソタ州５５３４４，エデン・プレイリー，ヴァリー・ビュー・ロード　１２９８２
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100089705
弁理士　社本　一夫
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100080137
弁理士　千葉　昭男
(74)代理人　100096013
弁理士　富田　博行

最終頁に続く

(54)【発明の名称】仮想ディスク・ドライブのシステムおよび方法

(57)【要約】
【課題】
動的にデータを割り当てる改善されたシステムを提供する。
【解決手段】
システムは、ストレージのプール（ＲＡＩＤのフリーリストを維持するストレージのページプール等）又はＲＡＩＤのヌルリストを維持するディスクストレージブロックのマトリックスを持つＲＡＩＤサブシステムと、ディスク記憶システム制御装置を持つディスクマネジャとを含む。サブシステム及びマネジャは、ストレージのプール及びディスクドライブにわたりデータを動的に割り当て、また、ディスクドライブが更に必要かを判定して必要な場合に通知を送る。
【選択図】図２



【特許請求の範囲】
【請求項１】
ストレージのプールにおいてデータを動的に割り当てる能力を有するディスク・ドライブ・システムであって、
前記ストレージのプールを有するＲＡＩＤサブシステムと、
少なくとも１つのディスク・ストレージ・システム・コントローラを有するディスク・マネージャと
を含み、
前記ＲＡＩＤサブシステムおよび前記ディスク・マネージャは、ＲＡＩＤ－ツー－ディスク・マッピングに基づいて、前記ストレージのプールおよび複数のディスク・ドライブにわたってデータを動的に割り当てる、
システム。
【請求項２】
請求項１に記載のシステムであって、前記ＲＡＩＤサブシステムおよび前記ディスク・マネージャは、更に別のディスク・ドライブが必要であるかを判定し、前記更に別のディスク・ドライブが必要である場合には通知が送られる、システム。
【請求項３】
請求項１に記載のシステムであって、前記ディスク・マネージャは複数のディスク・ストレージ・システム・コントローラを管理する、システム。
【請求項４】
請求項３に記載のシステムであって、動作されるディスク・ストレージ・システム・コントローラの障害をカバーするために複数の冗長なディスク・ストレージ・システム・コントローラを更に備える、システム。
【請求項５】
請求項１に記載のシステムであって、前記ＲＡＩＤサブシステムは、ＲＡＩＤ－０、ＲＡＩＤ－１、ＲＡＩＤ－５、ＲＡＩＤ－１０などのようなＲＡＩＤタイプのうちの少なくとも１つを含む組合せを更に備える、システム。
【請求項６】
請求項５に記載のシステムであって、ＲＡＩＤ－３、ＲＡＩＤ－４、ＲＡＩＤ－６、およびＲＡＩＤ－７を含むＲＡＩＤタイプを更に備える、システム。
【請求項７】
請求項１に記載のシステムであって、前記ストレージのプールは、ＲＡＩＤのフリー・リストを維持するストレージのページ・プールである、システム。
【請求項８】
請求項１に記載のシステムであって、前記ストレージのプールは、ＲＡＩＤのヌル・リストを維持するディスク・ストレージ・ブロックのマトリックスである、システム。
【請求項９】
動的なデータの割り当ての方法であって、
ストレージのプールを形成するＲＡＩＤサブシステムのディスク空間のデフォルトのサイズを定義するステップと、
データを書き込むステップと、
前記データを前記ストレージのプールにおいて割り当てるステップと、
前記ＲＡＩＤサブシステムの前記ディスク空間の占有率を、前記ＲＡＩＤサブシステムの前記ディスク空間の占有率の履歴に基づいて、決定するステップと、
更に別のディスク・ドライブが必要であるかを判定するステップと、
前記更に別のディスク・ドライブが必要である場合には、通知を前記ＲＡＩＤサブシステムへ送るステップと
を備える方法。
【請求項１０】
請求項９に記載の方法であって、前記通知を電子メールによって送ることを更に含む、

方法。
【請求項１１】
請求項９に記載の方法であって、前記ディスク・ストレージ・ブロックのサイズを、デフォルトとして設定し、ユーザによって変更可能であるように設定するステップを更に備える、方法。
【請求項１２】
請求項９に記載の方法であって、ＲＡＩＤのフリー・リストを維持するステップを更に備え、前記ストレージのプールは、前記ＲＡＩＤのフリー・リストを維持するストレージのページ・プールである、方法。
【請求項１３】
請求項９に記載の方法であって、ＲＡＩＤのヌル・リストを維持するステップを更に備え、前記ストレージのプールは、前記ＲＡＩＤのヌル・リストを維持するディスク・ストレージ・ブロックのマトリックスである、方法。
【請求項１４】
データ・インスタント・リプレイの方法であって、
ストレージのプールを形成するＲＡＩＤサブシステムのディスク空間のデフォルトのサイズを定義するステップと、
前記ストレージのプールのスナップショットを所定の時間間隔で自動的に生成するステップと、
前記スナップショットまたはデルタのアドレス・インデックスを前記ストレージのプールに保存して、前記ストレージのプールの前記スナップショットまたはデルタが、保存した前記アドレス・インデックスを介して直ちに見つけ出せるようにするステップと
を備える方法。
【請求項１５】
請求項１４に記載の方法であって、前記ＲＡＩＤサブシステムの前記スナップショットを所定の時間間隔で自動的に生成する前記ステップは、前記ＲＡＩＤサブシステムのスナップショットをユーザの定義する時間間隔で自動的に生成するステップを含む、方法。
【請求項１６】
請求項１５に記載の方法であって、前記時間間隔は数分毎から数時間毎の範囲にある、方法。
【請求項１７】
請求項１４に記載の方法であって、前記スナップショットはローカルのＲＡＩＤサブシステムに保存される、方法。
【請求項１８】
請求項１４に記載の方法であって、前記スナップショットはリモートのＲＡＩＤサブシステムで保存され、重大なシステム・クラッシュが発生した場合に、データ・インテグリティに対する影響を受けないようにし、前記データが直ちにリカバリできるようにする、方法。
【請求項１９】
請求項１４に記載の方法であって、前記ストレージのプールは、ＲＡＩＤのフリー・リストを維持するストレージのページ・プールである、方法。
【請求項２０】
請求項１４に記載の方法であって、前記ストレージのプールは、ＲＡＩＤのヌル・リストを維持するディスク・ストレージ・ブロックのマトリックスである、方法。
【請求項２１】
データ・インスタント・リプレイのシステムであって、
ストレージのプールを有するＲＡＩＤサブシステムと、
少なくとも１つのディスク・ストレージ・システム・コントローラを有するディスク・マネージャと
を含み、

前記ＲＡＩＤサブシステムおよび前記ディスク・マネージャは、ＲＡＩＤ－ツー－ディスク・マッピングに基づいて、データを前記ストレージのプールおよび複数のディスク・ドライブにわたって動的に割り当てる、
システム。
【請求項２２】
請求項２１に記載のシステムであって、前記ストレージのプールのスナップショットを所定の時間間隔で自動的に生成するため、およびスナップショットまたはデルタのアドレス・インデックスを前記ストレージのプールに保存して、前記ストレージのプールの前記スナップショットまたはデルタが、保存した前記アドレス・インデックスを介して直ちに見つけ出せるようにするためのディスク・ストレージ・システム・コントローラを更に備えるシステム。
【請求項２３】
請求項２２に記載のシステムであって、前記ディスク・ストレージ・システム・コントローラは、前記ストレージのプールの前記スナップショットからのデータ使用の頻度を監視し、使用またはアクセスの頻度の低いデータを費用の低いＲＡＩＤサブシステムへ移動させるエージング規則を適用する、システム。
【請求項２４】
請求項２３に記載のシステムであって、費用の低いＲＡＩＤサブシステム内のデータがより頻繁に使用され始めると、前記コントローラは、前記データを、より費用の高いＲＡＩＤサブシステムへ移動させ、それにより、前記ディスク・ドライブ・システムのコストを著しく低減させる、システム。
【請求項２５】
請求項２１に記載のシステムであって、前記ストレージのプールは、ＲＡＩＤのフリー・リストを維持するストレージのページ・プールである、システム。
【請求項２６】
請求項２１に記載のシステムであって、前記ストレージのプールは、ＲＡＩＤのヌル・リストを維持するディスク・ストレージ・ブロックのマトリックスである、システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般には、ディスク・ドライブのシステムおよび方法に関し、より詳細には、動的データ割り当てやディスク・ドライブ仮想化などの機能を有するディスク・ドライブ・システムに関する。
【背景技術】
【０００２】
既存のディスク・ドライブ・システムは、仮想ボリューム（virtual volume）・データ・ストレージ空間が、データの格納のための特定のサイズおよび場所を有する物理ディスクと静的に関連するように、設計されている。こうしたディスク・ドライブ・システムでは、データを格納するために、データ・ストレージ空間の仮想ボリュームの厳密な場所およびサイズを知ること及び監視／制御を行うことが必要である。更に、このシステムでは、しばしば、より大きなデータ・ストレージ空間が必要であり、そのため、ＲＡＩＤ装置が追加される。しかし、しばしば、こうした追加のＲＡＩＤ装置は、高価であり、また、余分なデータ・ストレージ空間が実際に必要となるまでは、必要ではない。
【０００３】
図１４のＡに、データの格納、読み出し／書き込み、および／またはリカバリ（復旧）のための特定のサイズおよび場所を有する物理ディスクと関連する仮想ボリューム・データ・ストレージ空間を有する従来の既存のディスク・ドライブ・システムを示す。このディスク・ドライブ・システムでは、データの割り当てを、データ・ストレージ空間の仮想ボリュームの特定の場所およびサイズに基づいて、静的に行う。その結果、空のデータ・ストレージ空間は使用されず、このシステムでデータを格納、読み出し／書き込み、およ

び／またはリカバリするために、余分であり時には高価なデータ・ストレージ装置、例えばＲＡＩＤ装置が前もって獲得される。こうした余分なデータ・ストレージ空間は、必要でなく、かつ／または、後の時まで使用されない可能性がある。

【発明の概要】

【０００４】

従って、改善されたディスク・ドライブのシステムおよび方法が必要とされている。効率のよい、動的なデータ割り当てならびにディスク・ドライブの空間および時間管理のシステムおよび方法が更に必要とされている。

【０００５】

本発明は、動的にデータを割り当てる能力を有する改善されたディスク・ドライブのシステムおよび方法を提供する。このディスク・ドライブ・システムは、ディスク・ストレージ・ブロックのマトリックスを有するＲＡＩＤサブシステムと、少なくとも１つのディスク・ストレージ・システム・コントローラを有するディスク・マネージャとを含むことができる。ＲＡＩＤサブシステムおよびディスク・マネージャは、ＲＡＩＤからディスクへのマッピング（RAID-to-disk mapping、ＲＡＩＤ－ツー－ディスク・マッピング）に基づいて、ディスク・ストレージ・ブロックのマトリックスおよび複数のディスク・ドライブを通して、データを動的に割り当てる。ＲＡＩＤサブシステムおよびディスク・マネージャは、更に別のディスク・ドライブが必要かどうかを判定し、更に別のディスク・ドライブが必要である場合には、通知を送る。動的なデータの割り当て（アロケーション）により、ユーザは、ディスク・ドライブを、後の時間に、必要となったときに得ることができる。また、動的データ割り当てにより、ディスク・ストレージ・ブロックの仮想ボリューム・マトリックスまたはプールのスナップショット／ポイント・イン・タイム（時）・コピーの効率のよいデータ格納や、データのバックアップ、リカバリなどのためのインスタント・データ・リプレイおよびデータ・インスタント・フュージョン（fusion）や、リモート・データ・ストレージや、データ・プログレッション（Data Progression）などが、可能になる。また、データ・プログレッションにより、より安価なディスク・ドライブは、後の時間に購入されるため、延期が可能になる。

【０００６】

一実施形態では、仮想ボリュームまたはディスク・ストレージ・ブロックのマトリックスまたはプールが、物理ディスクと関連付けるために提供される。仮想ボリュームまたはディスク・ストレージ・ブロックのマトリックスまたはプールの監視／制御は、複数のディスク・ストレージ・システム・コントローラによって動的に行われる。一実施形態では、それぞれの仮想ボリュームのサイズはデフォルトまたはユーザの予め定義したものとすることができ、それぞれの仮想ボリュームの場所はデフォルトでヌル（ｎｕｌｌ）となっている。仮想ボリュームは、データが割り当てられるまでヌルである。データは、マトリックスまたはプールの任意のグリッドに割り当てることができる（例えば、データがそのグリッドに割り当てられた後は、そのグリッド内の「ドット」（dot）になる）。そのデータが削除された後、その仮想ボリュームは再び使用可能になり「ｎｕｌｌ」として示される。このため、余分なデータ・ストレージ空間や、時には高価なデータ・ストレージ装置、例えばＲＡＩＤ装置を、後の時間に、必要となるたびに得ることができる。

【０００７】

一実施形態では、１つのディスク・マネージャが複数のディスク・ストレージ・システム・コントローラを管理することができ、複数の冗長ディスク・ストレージ・システム・コントローラを実装して、動作させているディスク・ストレージ・システム・コントローラの障害をカバーすることができる。

【０００８】

一実施形態では、ＲＡＩＤサブシステムは、ＲＡＩＤ－０、ＲＡＩＤ－１、ＲＡＩＤ－５、ＲＡＩＤ－１０などのＲＡＩＤタイプのうちの少なくとも１つからなる組合せを含む。代替のＲＡＩＤサブシステムでは、ＲＡＩＤ－３、ＲＡＩＤ－４、ＲＡＩＤ－６、ＲＡＩＤ－７などのような他のＲＡＩＤタイプを使用できることも理解されよう。

【０００９】
また、本発明は、動的データ割り当ての方法を提供するものであり、この方法は、論理ブロックまたはディスク・ストレージ・ブロックのデフォルトのサイズを与えて、ＲＡＩＤサブシステムのディスク空間がディスク・ストレージ・ブロックのマトリックスを形成するようにするステップと、データを書き込み、そのデータをディスク・ストレージ・ブロックのマトリックスに割り当てるステップと、ＲＡＩＤサブシステムのディスク空間の占有率の履歴に基づいて、ＲＡＩＤサブシステムのディスク空間の占有率の決定するステップと、更に別のディスク・ドライブが必要かどうかを決定するステップと、更に別のディスク・ドライブが必要である場合に、ＲＡＩＤサブシステムに通知を送るステップとを含む。一実施形態では、通知は電子メールによって送られる。

【００１０】
本発明のディスク・ドライブ・システムの利点の１つは、ＲＡＩＤサブシステムが、仮想的な数のディスクにわたってＲＡＩＤ技法を使用できることである。残りのストレージ空間は自由に使用可能である。ＲＡＩＤサブシステムのストレージ空間を監視し、そのストレージ空間の占有率を決定することにより、ユーザは、高価であるが購入時点では用途のない大量のドライブを得なくて済む。こうして、ドライブの追加を、ストレージ空間の増大する需要を満たすために実際に必要であるときに行うことにより、ディスク・ドライブの全体のコストが著しく低減される。その一方で、ドライブの使用の効率は著しく改善される。

【００１１】
本発明の別の利点は、ディスク・ストレージ・システム・コントローラが、特定のコンピュータ・ファイル・システムだけでなく、どのようなコンピュータ・ファイル・システムに対しても普遍的であることである。

【００１２】
また、本発明は、データ・インスタント・リプレイ（data instant replay）の方法を提供する。一実施形態では、データ・インスタント・リプレイの方法は、論理ブロックまたはディスク・ストレージ・ブロックのデフォルトのサイズを与えて、ＲＡＩＤサブシステムのディスク空間がストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスを形成するようにするステップと、ストレージのページ・プールのボリュームのスナップショットまたはディスク・ストレージ・ブロックのマトリックスのスナップショットを、所定の時間間隔で自動的に生成するステップと、スナップショットまたは変分（delta、デルタ）のアドレス・インデックスを、ストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスに保存して、ディスク・ストレージ・ブロックのマトリックスのスナップショットまたはデルタを、その保存したアドレス・インデックスを介して直ちに見つけ出せるようにするステップとを含む。

【００１３】
このデータ・インスタント・リプレイの方法では、ＲＡＩＤサブシステムのスナップショットの自動生成を、ユーザの定義する時間間隔で、または、ユーザのコンフィギュレーションした動的なタイム・スタンプ、例えば、数分または数時間ごとなどに、または、サーバの指示する時刻に、行う。システム障害またはウイルス攻撃が発生した場合、こうしたタイム・スタンプを有する仮想スナップショットにより、データ・インスタント・リプレイおよびデータ・インスンタント・リカバリが、約数分または数時間などで可能になる。この技法は、インスタント・リプレイ・フュージョン（instant replay fusion）とも呼ばれる。即ち、クラッシュまたは攻撃のすぐ前のデータのフュージョンが遅れずに行われ、クラッシュまたは攻撃の前に保存したスナップショットを直ちにそれ以降の動作で使用することができる。

【００１４】
一実施形態では、スナップショットをローカルのＲＡＩＤサブシステムまたはリモートのＲＡＩＤサブシステムで保存し、重大なシステム・クラッシュが例えばテロリスト攻撃などが原因で生じた場合に、データの完全性（integrity）に対しての影響を受けないよ

うにし、データを直ちにリカバリできるようにすることができる。
【００１５】
このデータ・インスタント・リプレイの方法の別の利点は、スナップショットを、システムが動作し続けたままで、検査のために使用できることである。生（live）のデータをリアル・タイムの検査に使用することができる。
【００１６】
また、本発明は、ＲＡＩＤサブシステムと、少なくとも１つのディスク・ストレージ・システム・コントローラを有するディスク・マネージャとを含むデータ・インスタント・リプレイのシステムを提供する。一実施形態では、ＲＡＩＤサブシステムおよびディスク・マネージャは、複数のドライブのディスク空間にわたって、データを、ＲＡＩＤ－ツー－ディスク・マッピングに基づいて動的に割り当てるものであり、ここでＲＡＩＤサブシステムのディスク空間は、ディスク・ストレージ・ブロックのマトリックスを形成する。ディスク・ストレージ・システム・コントローラは、ディスク・ストレージ・ブロックのマトリックスのスナップショットを所定の時間間隔で自動的に生成し、スナップショットまたはデルタのアドレス・インデックスをディスク・ストレージ・ブロックのマトリックスに保存して、ディスク・ストレージ・ブロックのマトリックスのスナップショットまたはデルタを、その保存したアドレス・インデックスを介して直ちに見つけ出せるようにする。
【００１７】
一実施形態では、ディスク・ストレージ・システム・コントローラは、ディスク・ストレージ・ブロックのマトリックスのスナップショットからデータ使用の頻度を監視し、そして、使用またはアクセスの頻度の低いデータほど、より安価なＲＡＩＤサブシステムへと移動させるような、エージング（aging、加齢）規則を適用する。同様に、安価なＲＡＩＤサブシステム中のデータが頻繁に使用され始めると、コントローラは、そのデータを、より高価なＲＡＩＤサブシステムへと移動させる。このため、ユーザは、ユーザ自身のストレージに対する必要性を満たす所望のＲＡＩＤサブシステム・ポートフォリオを選ぶことができる。従って、ディスク・ドライブ・システムのコストを著しく低減させること、およびユーザにより動的に制御することが可能となる。
【００１８】
本発明のこうしたおよび他の特徴および利点は、本発明を実現するために企図された最良の形態を含む本発明の例示的な実施形態を示し説明する以下の詳細な説明から、当業者に明らかとなろう。本発明は、様々な自明な態様における変更が、すべて本発明の趣旨および範囲から逸脱することなく可能であることが理解されよう。従って、図面および詳細な説明は、本来例示的であって限定的でないと見なすべきである。
【図面の簡単な説明】
【００１９】
【図１】図１は、本発明の諸原理に従ったコンピュータ環境におけるディスク・ドライブ・システムの一実施形態を示す。
【図２】図２は、本発明の諸原理に従った、ディスク・ドライブのＲＡＩＤサブシステムのためのストレージのページ・プールを有する動的なデータ割り当ての一実施形態を示す。
【図２Ａ】図２Ａは、ディスク・ドライブ・システムのＲＡＩＤサブシステムにおける従来のデータ割り当てを示す。
【図２Ｂ】図２Ｂは、本発明の諸原理に従った、ディスク・ドライブ・システムのＲＡＩＤサブシステムにおけるデータ割り当てを示す。
【図２Ｃ】図２Ｃは、本発明の諸原理に従った動的なデータ割り当ての方法を示す。
【図３】図３のＡおよびＢは、本発明の諸原理に従った、複数の時間間隔におけるＲＡＩＤサブシステムのディスク・ストレージ・ブロックのスナップショットの概略図であり、Ｃは、本発明の諸原理に従ったデータ・インスタント・リプレイの方法を示す。
【図４】図４は、本発明の諸原理に従った、ＲＡＩＤサブシステムのディスク・ストレー

ジ・ブロックのスナップショットを使用することによるデータ・インスタント・フュージョン機能の概略図である。
【図５】図５は、本発明の諸原理に従った、ＲＡＩＤサブシステムのディスク・ストレージ・ブロックのスナップショットを使用することによるローカル／リモートのデータの複製およびインスタント・リプレイの機能の概略図である。
【図６】図６は、本発明の諸原理に従った、Ｉ／Ｏを行うために同じＲＡＩＤインターフェースを使用し、複数のＲＡＩＤ装置を１つのボリュームへと連結するスナップショットの概略図を示す。
【図７】図７は、本発明の諸原理に従った、スナップショット構造の一実施形態を示す。
【図８】図８は、本発明の諸原理に従った、ＰＩＴＣライフ・サイクルの一実施形態を示す。
【図９】図９は、本発明の諸原理に従った、マルチ・レベル・インデックスを有するＰＩＴＣテーブル構造の一実施形態を示す。
【図１０】図１０は、本発明の諸原理に従った、ＰＩＴＣテーブルのリカバリの一実施形態を示す。
【図１１】図１１は、本発明の諸原理に従った、所有ページ・シーケンスおよび非所有ページ・シーケンスを有する書き込みプロセスの一実施形態を示す。
【図１２】図１２は、本発明の諸原理に従った、例示的なスナップショット動作を示す。
【図１３Ａ】図１３Ａは、データを静的に割り当てるために特定のサイズおよび場所をもつ物理ディスクと関連付けられた仮想ボリューム・データ・ストレージ空間を有する従来の既存のディスク・ドライブ・システムを示す。
【図１３Ｂ】図１３Ｂは、図１３Ａの従来の既存のディスク・ドライブ・システムでのボリューム論理ブロック・マッピングを示す。
【図１４】図１４のＡは、本発明の諸原理に従った、システムにおいてデータを動的に割り当てるための、ディスク・ストレージ・ブロックの仮想ボリューム・マトリックスを有するディスク・ドライブ・システムの一実施形態を示す。図１４のＢは、図１４のＡに示すディスク・ストレージ・ブロックの仮想ボリューム・マトリックスにおける動的なデータ割り当ての一実施形態を示す。図１４のＣは、本発明の諸原理に従った、ストレージの仮想ボリューム・ページ・プールの一実施形態のボリューム－ＲＡＩＤページ再マッピングの概略図を示す。
【図１５】図１５は、本発明の諸原理に従った、ＲＡＩＤサブシステムの複数のディスク・ストレージ・ブロックへマッピングされる３つのディスク・ドライブの例を示す。
【図１６】図１６は、図１５に示す３つのディスク・ドライブへ１つのディスク・ドライブを追加した後のディスク・ドライブ・ストレージ・ブロックの再マッピングの例を示す。
【図１７】図１７は、本発明の諸原理に従った、データ・プログレッション動作におけるアクセス可能なデータ・ページの一実施形態を示す。
【図１８】図１８は、本発明の諸原理に従った、データ・プログレッション・プロセスの一実施形態の流れ図を示す。
【図１９】図１９は、本発明の諸原理に従った、圧縮されたページ・レイアウトの一実施形態を示す。
【図２０】図２０は、本発明の諸原理に従った、高レベル・ディスク・ドライブ・システムにおけるデータ・プログレッションの一実施形態を示す。
【図２１】図２１は、本発明の諸原理に従った、サブシステムにおける外部データ・フローの一実施形態を示す。
【図２２】図２２は、サブシステムにおける内部データ・フローの一実施形態を示す。
【図２３】図２３は、コヒーレンシを独立して維持する各サブシステムの一実施形態を示す。
【図２４】図２４は、本発明の諸原理に従った、混合ＲＡＩＤウォーターフォール・データ・プログレッションの一実施形態を示す。

【図２５】図２５は、本発明の諸原理に従った、ストレージのページ・プールの複数のフリー・リストの一実施形態を示す。
【図２６】図２６は、本発明の諸原理に従った、データベースの例の一実施形態を示す。
【図２７】図２７は、本発明の諸原理に従った、ＭＲＩ画像の例の一実施形態を示す。
【発明を実施するための形態】
【００２０】
本発明は、動的にデータを割り当てる能力を有する改善されたディスク・ドライブのシステムおよび方法を提供する。このディスク・ドライブ・システムは、ＲＡＩＤの空きリスト（フリー・リスト、free list）を維持するストレージのページ・プール、または代替例としてはディスク・ストレージ・ブロックのマトリックスを有するＲＡＩＤサブシステムと、少なくとも１つのディスク・ストレージ・システム・コントローラを有するディスク・マネージャとを含むことができる。ＲＡＩＤサブシステムおよびディスク・マネージャは、データを、ストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスおよび複数のディスク・ドライブにわたって、ＲＡＩＤ－ツー－ディスク・マッピングに基づいて、動的に割り当てる。ＲＡＩＤサブシステムおよびディスク・マネージャは、更に別のディスク・ドライブが必要かどうかを判定し、更に別のディスク・ドライブが必要である場合には通知を送る。動的なデータ割り当てにより、ユーザは、ディスク・ドライブを、後にそれが必要となったときに得ることができる。また、動的なデータ割り当てにより、ディスク・ストレージ・ブロックの仮想ボリュームのマトリックスまたはプールのスナップショット／ポイント・イン・タイム・コピーの効率のよいデータ・ストレージや、データ・バックアップやリカバリなどのためのインスタント・データ・リプレイおよびデータ・インスタント・フュージョン（fusion）や、リモート・データ・ストレージや、データ・プログレッションなどが可能になる。また、データ・プログレッションにより、より安価なディスク・ドライブは後の時間に購入されるため、その延期が可能になる。
【００２１】
図１に、本発明の諸原理に従った、コンピュータ環境１０２におけるディスク・ドライブ・システム１００の一実施形態を示す。図１に示すように、ディスク・ドライブ・システム１００は、ＲＡＩＤサブシステム１０４と、少なくとも１つのディスク・ストレージ・システム・コントローラ（図１６）を有するディスク・マネージャ１０６とを含む。ＲＡＩＤサブシステム１０４およびディスク・マネージャ１０６は、データを、複数のディスク・ドライブ１０８のディスク空間にわたって、ＲＡＩＤ－ツー－ディスク・マッピングに基づいて、動的に割り当てる。更に、ＲＡＩＤサブシステム１０４およびディスク・マネージャ１０６は、更に別のディスク・ドライブが必要かどうかの判定を、ディスク空間全体にわたるデータ割り当てに基づいて行う能力がある。更に別のディスク・ドライブが必要である場合にはその通知がユーザへ送られて、希望される場合には更に別のディスク空間を追加できる。
【００２２】
本発明の諸原理に従った、動的なデータ割り当て（または「ディスク・ドライブ仮想化」と呼ばれる）を有するディスク・ストレージ・システム１００を、図２に一つの実施形態の形で示し、図１４のＡおよび図１４のＢに別の実施形態の形で示している。図２に示すように、ディスク・ストレージ・システム１１０は、ストレージのページ・プール１１２、即ち、自由にデータを保存できるデータ・ストレージ空間のリストを含むデータ・ストレージのプールを含む。ページ・プール１１２は、ＲＡＩＤ装置１１４のフリー・リストを維持し、読み出し／書き込みの割り当ての管理を、ユーザの要求に基づいて行う。ユーザの要求したデータ・ストレージ・ボリューム１１６は、ストレージ空間を得るために、ページ・プール１１２へと送られる。それぞれのボリュームは、同じまたは異なるＲＡＩＤレベル、例えば、ＲＡＩＤ１０、ＲＡＩＤ５、ＲＡＩＤ０などで、同じまたは異なるクラスのストレージ装置を要求することができる。
【００２３】

本発明の動的なデータ割り当ての別の実施形態を、図１４のＡおよび図１４のＢに示しており、この例では、複数のディスク・ストレージ・システム・コントローラ１４０２と、複数のディスク・ストレージ・システム・コントローラ１４０２により制御されるディスク・ストレージ・ブロックのマトリックス１４０４とを有するディスク・ストレージ・システム１４００は、そのシステムにおいて本発明の諸原理に従ってデータを動的に割り当てる。仮想ボリュームまたはブロックのマトリックス１４０４は、物理ディスクと関連づけるように提供される。仮想ボリュームまたはブロックのマトリックス１４０４は、複数のディスク・ストレージ・システム・コントローラ１４０２によって動的に監視／制御される。一実施形態では、それぞれの仮想ボリューム１４０４のサイズは、例えば２メガバイトのように、予め定めることができ、それぞれの仮想ボリューム１４０４の場所はデフォルトでｎｕｌｌとなっている。仮想ボリューム１４０４のそれぞれは、データが割り当てられるまでｎｕｌｌである。データは、そのマトリックスまたはプールの任意のグリッドに割り当てることができる（例えば、データがそのグリッドに割り当てられた後は、そのグリッド内の「ドット」になる）。そのデータが削除された後は、その仮想ボリューム１４０４は再び使用可能になり、「ｎｕｌｌ」と示される。このため、余分な、時には高価なデータ・ストレージ装置、例えばＲＡＩＤ装置を、後に必要となったときに得ることができる。

【００２４】

従って、ＲＡＩＤサブシステムは、仮想的な数のディスクにわたってＲＡＩＤ技法を使用することができる。残りのストレージ空間は自由に使用可能である。ＲＡＩＤサブシステムのストレージ空間を監視し、そのストレージ空間の占有率を決定することにより、ユーザは、高価であるのに購入時点では用途のない大量のドライブを得なくても済む。このように、ドライブの追加を、ストレージ空間の増大する需要を満たすために実際に必要であるときに行うことにより、ディスク・ドライブの全体のコストが著しく低減される。一方、ドライブの使用の効率も著しく改善される。

【００２５】

また、本発明のディスク・ドライブ・システムの動的なデータ割り当てにより、ストレージの仮想ボリューム・ページ・プール、または、ディスク・ストレージ・ブロックの仮想ボリューム・マトリックスの、スナップショット／ポイント・イン・タイム・コピーの効率のよいデータ・ストレージや、データ・リカバリおよびリモート・データ・ストレージのためのインスタント・データ・リプレイおよびデータ・インスタント・フュージョンや、データ・プログレッションが可能になる。

【００２６】

ディスク・ドライブ・システム１００における動的データ割り当てのシステムおよび方法ならびにその実装形態の結果として得られる上述の特徴および利点を、以下で詳細に論じる。

【００２７】

動的なデータ割り当て

図２Ａに、ディスク・ドライブ・システムのＲＡＩＤサブシステムにおける従来のデータ割り当てを示し、ここでは、空けられたデータ・ストレージ空間は専属（captive）であり、データ・ストレージ用に割り当てることができない。

【００２８】

図２Ｂには、本発明の諸原理に従ったディスク・ドライブ・システムのＲＡＩＤサブシステムにおけるデータ割り当てを示し、ここでは、データ・ストレージ用に使用可能である空けられたデータ・ストレージは、一緒にまとめられて、ページ・プール、例えば、本発明の一実施形態では１つのページ・プール、を形成する。

【００２９】

図２Ｃに、本発明の諸原理に従った動的なデータ割り当ての方法２００を示す。動的なデータの割り当ての方法２００は、論理ブロックまたはディスク・ストレージ・ブロックのデフォルトのサイズを定義して、ＲＡＩＤサブシステムのディスク空間がディスク・ス

トレージ・ブロックのマトリックスを形成するようにするステップ２０２と、データを書き込み、そのデータを、マトリックスの、「ｎｕｌｌ」を示すディスク・ストレージ・ブロックに割り当てるステップ２０４とを含む。この方法は、更に、ＲＡＩＤサブシステムのディスク空間の占有率を、そのＲＡＩＤサブシステムのディスク空間の占有率履歴に基づいて決定するステップ２０６と、更に別のディスク・ドライブが必要かどうかを判定し、必要な場合に、ＲＡＩＤサブシステムへ通知を送るステップ２０８とを含む。一実施形態では、通知は電子メールによって送られる。更に、ディスク・ストレージ・ブロックのサイズはデフォルトに設定し、ユーザにより変更可能とすることができる。

【００３０】

一実施形態では、動的なデータ割り当ては、時には「仮想化」または「ディスク空間仮想化」と呼ばれるものであり、数多くの読み出しおよび書き込みの要求を秒単位で効率よく取り扱う。このアーキテクチャでは、キャッシュ・サブシステムを直接に呼び出す割り込みハンドラが必要であることもある。動的なデータの割り当てでは、要求（リクエスト）をキューに入れないので要求の最適化はなされないが、一度に数多くの未解決（pending）の要求を有し得る。

【００３１】

また、動的なデータの割り当てでは、データのインテグリティを維持し、データの内容をどのようなコントローラの障害に対しても保護することができる。そうするために、動的なデータの割り当てでは、状態情報を信頼性をもって記憶するためにＲＡＩＤ装置へ書き込む。

【００３２】

動的なデータの割り当てでは、更に、読み出しおよび書き込みの要求の順序を維持し、読み出しまたは書き込みの要求を、要求を受け取った順序で完了させることができる。動的なデータの割り当ては、最大のシステム可用性（availability）を提供し、異なる地理的場所へのデータのリモートの複製をサポートする。

【００３３】

更に、動的なデータの割り当ては、データ破壊からのリカバリ機能を提供する。スナップショットにより、ユーザは、過去におけるディスクの状態を閲覧することができる。

【００３４】

動的なデータの割り当てでは、ＲＡＩＤ装置を管理し、大規模な装置を作成し拡張するためのストレージ抽象化を提供する。

【００３５】

動的なデータの割り当てでは、サーバに対して仮想ディスク装置を提示するものであり、この装置がボリューム（volume）と呼ばれる。サーバに対して、ボリュームは同じ働きをする。ボリュームは、シリアル番号に関して異なる情報を返し得るが、本質的にはディスク・ドライブのような働きをする。ボリュームは、より大きい動的なボリューム装置を作成するために、複数のＲＡＩＤ装置のストレージ抽象化を提供する。ボリュームは複数のＲＡＩＤ装置を含み、ディスク空間の効率的な使用を可能にする。

【００３６】

図２１に、従来の既存のボリューム論理ブロック・マッピングを示す。図１４のＣには、本発明の諸原理に従った、ストレージの仮想ボリュームのページ・プールの一実施形態のボリューム－ＲＡＩＤページ再マッピングを示す。それぞれのボリュームは１組のページ、例えば、１、２、３などへと分割され、それぞれのＲＡＩＤは１組のページへと分割される。ボリュームのページ・サイズと、ＲＡＩＤのページ・サイズとは、一実施形態では同じとすることができる。従って、本発明のボリューム－ＲＡＩＤページ再マッピングの一例では、ＲＡＩＤ－２を使用するページ＃１がＲＡＩＤページ＃１へとマッピングされる。

【００３７】

動的なデータの割り当てでは、ボリュームのデータ・インテグリティを維持する。データは、ボリュームへと書き込まれ、サーバへの確認が行われる。データ・インテグリティ

は、コントローラの障害を通して、スタンド・アロンや冗長なものを含めての様々なコントローラのコンフィギュレーションに適用される。コントローラの障害としては、電源障害、パワー・サイクル（power cycle）、ソフトウェア例外、およびハード・リセットがある。動的なデータの割り当てでは、一般には、ＲＡＩＤでカバーされるディスク・ドライブ障害を取り扱わない。

【００３８】

動的なデータの割り当ては、コントローラに対する最も高レベルのデータ抽象化を提供する。動的なデータの割り当ては、フロント・エンドから要求を受理し、最終的にはＲＡＩＤ装置を使用してデータをディスクへ書き込む。

【００３９】

動的なデータの割り当ては、幾つもの内部サブシステムを含む。

・キャッシュ　－　ボリュームへの読み出しおよび書き込みの動作を、サーバへの迅速な応答時間を提供し、データ・プラグインに対して書き込みをまとめることによって、スムーズにする。

・コンフィギュレーション　－　データ割り当てオブジェクトの作成、削除、取り出し、および変更の方法を含む。より高レベルのシステム・アプリケーションに対するツールボックスを作成するためのコンポーネントを提供する。

・データ・プラグイン　－　ボリュームの読み出しおよび書き込みの要求を、ボリューム・コンフィギュレーションに応じて、様々なサブシステムへ配布する。

・ＲＡＩＤインターフェース　－　より大きいボリュームを作成するためのＲＡＩＤ装置抽象化を、ユーザおよび他の動的なデータの割り当てサブシステムに提供する。

・コピー／ミラー／スワップ　－　ボリュームのデータを、ローカルおよびリモートのボリュームへと複製する。一実施形態では、サーバの書き込まれたブロックのコピーのみを行うことができる。

・スナップショット　－　データの増分的（incremental）ボリューム・リカバリを提供する。過去のボリューム状態の閲覧ボリューム（View Volume）を即座に作成する。

・プロキシ・ボリューム　－　リモートの宛先ボリュームへの要求の通信を行えるようにして、リモート複製（Remote Replication）をサポートする。

・請求（billing）　－　割り当てたストレージ、活動、パフォーマンス、およびデータのリカバリに対してユーザに料金請求する機能。

【００４０】

また、動的なデータの割り当てでは、エラーおよびコンフィギュレーションにおける顕著な変更のログ記録も行う。

【００４１】

図２１に、サブシステムにおける外部データ・フローの一実施形態を示す。外部要求はフロント・エンドから来る。要求には、ボリューム情報取得（get volume information）、読み出し（read）、および書き込み（write）がある。すべての要求はボリュームＩＤをもつ。ボリューム情報は、ボリューム・コンフィギュレーション・サブシステムによって取り扱われる。読み出しおよび書き込みの要求はＬＢＡを含む。また、書き込み要求はデータを含む。

【００４２】

動的なデータの割り当ては、ボリューム・コンフィギュレーションに応じて、要求を幾つもの外部レイヤへ渡す。リモート複製は、要求をフロント・エンドへ渡すが、あて先はリモートの宛先ボリュームである。ＲＡＩＤインターフェースは要求をＲＡＩＤへ渡す。コピー／ミラー／スワップは、要求を、宛先ボリュームに対する動的データ割り当てへと戻す。

【００４３】

図２２に、サブシステムにおける内部データ・フローの一実施形態を示す。内部データ・フローはキャッシュを行うことから始まる。キャッシュを行うと、書き込み要求がキャッシュに入れられるか、または、その要求が直接にデータ・プラグインへ渡される。キャ

ッシュは、フロント・エンドのＨＢＡ装置からの直接ＤＭＡをサポートする。要求は迅速に完了され、応答がサーバに返される。データ・プラグイン・マネージャが、キャッシュより下の要求フローの中心である。それぞれのボリュームに対して、データ・プラグイン・マネージャは、要求ごとに、登録されたサブシステム・オブジェクトを呼び出す。

【００４４】

データ・インテグリティに影響を及ぼす動的データ割り当てサブシステムでは、コントローラの一貫性（コヒーレンシ）に対するサポートが必要となることがある。図２３に示すように、それぞれのサブシステムは、独立してコヒーレンシを維持する。コヒーレンシの更新により、コヒーレンシ・リンクを通じてのデータ・ブロックのコピーを回避する。キャッシュのコヒーレンシでは、データをピア（peer）コントローラにコピーする必要があり得る。

【００４５】

ディスク・ストレージ・システム・コントローラ

図１４のＡに、本発明の諸原理に従った、システムにおいて動的にデータを割り当てるための、複数のディスク・ストレージ・システム・コントローラ１４０２と、複数のディスク・ストレージ・システム・コントローラ１４０２により制御されるディスク・ストレージ・ブロックまたは仮想ボリュームのマトリックス１４０４とを有するディスク・ストレージ・システム１４００を示す。図１４のＢには、ディスク・ストレージ・ブロックまたは仮想ボリュームの仮想ボリューム・マトリックス１４０４における動的なデータの割り当ての一実施形態を示す。

【００４６】

ある動作では、ディスク・ストレージ・システム１４００は、ディスク・ストレージ・ブロックまたは仮想ボリュームのマトリックス１４０４のスナップショットを、所定の時間間隔で時間的に生成し、スナップショットまたはデルタのアドレス・インデックスを、ディスク・ストレージ・ブロックまたは仮想ボリュームのマトリックス１４０４に保存して、ディスク・ストレージ・ブロックまたは仮想ボリュームのマトリックス１４０４のスナップショットまたはデルタが、保存したアドレス・インデックスによって直ちに見つけ出せるようにする。

【００４７】

更にある動作では、ディスク・ストレージ・システム・コントローラ１４０２は、ディスク・ストレージ・ブロックのマトリックス１４０４のスナップショットからデータ使用の頻度を監視し、そして、使用またはアクセスの頻度の低いデータほど、より安価なＲＡＩＤサブシステムへと移動させるようなエージング規則を適用する。同様に、より安価なＲＡＩＤサブシステム中のデータがより頻繁に使用され始めると、コントローラは、そのデータを、より高価なＲＡＩＤサブシステムへと移動させる。従って、ユーザは、ユーザ自身のストレージの必要性を満たすように所望のＲＡＩＤサブシステム・ポートフォリオを選ぶことができる。従って、ディスク・ドライブ・システムのコストを著しく低減させ、また、ユーザによる動的な制御を可能とする。

【００４８】

ＲＡＩＤ－ツー－ディスク・マッピング（RAID-to-Disk Mapping）

ＲＡＩＤサブシステムおよびディスク・マネージャは、データを、複数のディスク・ドライブのディスク空間にわたって、ＲＡＩＤ－ツー－ディスク・マッピングに基づいて、動的に割り当てる。一実施形態では、ＲＡＩＤサブシステムおよびディスク・マネージャは、更に別のディスク・ドライブが必要かどうかを判定し、更に別のディスク・ドライブが必要である場合には通知を送る。

【００４９】

図１５に、本発明の諸原理に従った、ＲＡＩＤ－５サブシステム１５００内の複数のディスク・ストレージ・ブロック１５０２～１５１２へとマッピングされる３つのディスク・ドライブ１０８（図１）の例を示す。

【００５０】

図１６は、図１５に示す３つのディスク・ドライブ１０８にディスク・ドライブ１６０２を追加した後の、ディスク・ドライブ・ストレージ・ブロックの再マッピング１６００の例を示す。

【００５１】

ディスク・マネージャ

図１に示すディスク・マネージャ１０６は、一般には、ディスクおよびディスク・アレイの管理を行うものであり、それには、グループ化／資源プーリング、ディスク属性の抽象化、フォーマット、ディスクの追加（addition）／削減（subtraction）、およびディスク・サービス時間およびエラー率の追跡を含む。ディスク・マネージャ１０６は、ディスクの様々なモデルの間の違いを区別せず、ＲＡＩＤコンポーネントについて包括的なストレージ装置を提示する。また、ディスク・マネージャ１０６はグループ化機能を提供し、これは、例えば１００００ＲＰＭのディスクなどのような特定の特徴をもつＲＡＩＤグループの構築を容易にする。

【００５２】

本発明の一実施形態では、ディスク・マネージャ１０６は、少なくとも３重のものとなっている。その３重のものとは、抽象化、コンフィギュレーション、およびＩ／Ｏ最適化である。ディスク・マネージャ１０６は、「ディスク」を上位レイヤに提示し、上位レイヤは、例えば、ローカルまたはリモートに取り付けられた物理ディスク・ドライブや、リモートに取り付けられたディスク・システムなどとすることができる。

【００５３】

共通の基礎にある特徴は、こうした装置の何れもがＩ／Ｏ動作のターゲットであり得ることである。抽象化サービスは、特にＲＡＩＤサブシステムのような上位レイヤに対して、一様なデータ・パス・インターフェースを提供し、そして、管理者に対して、ターゲット装置の管理のための包括的な機構を提供する。

【００５４】

また、本発明のディスク・マネージャ１０６は、ディスクのグループ化機能を提供して管理およびコンフィギュレーションを簡素化する。ディスクには名前を付け、グループに入れることができ、グループにも名前を付けることができる。グループ化は強力な機能であり、これにより、ボリュームを或るディスクのグループから別のグループへと移行させるタスクや、或るグループのディスクを特定の機能専用にするタスクや、或るグループのディスクをスペアに指定するタスクなどのタスクを簡単にする。

【００５５】

また、ディスク・マネージャは、外部装置の存在を検出する役目をもつＳＣＳＩ装置サブシステムなどのような装置とインターフェースを行う。ＳＣＳＩ装置サブシステムは、少なくともファイバ・チャネル／ＳＣＳＩタイプの装置については、ブロック型のターゲット装置である装置のサブセットを判定する能力をもつ。ディスク・マネージャが管理および抽象化を行うのはこうした装置である。

【００５６】

更に、ディスク・マネージャは、ＳＣＳＩ装置レイヤからのフロー制御に応答する役目をもつ。ディスク・マネージャは、キューイング機能を有し、これにより、ディスク・ドライブ・システムのスループットを最適化する一方法として、Ｉ／Ｏ要求を集合させる機会が与えられる。

【００５７】

更に、本発明のディスク・マネージャは、複数のディスク・ストレージ・システム・コントローラを管理する。また、複数の冗長ディスク・ストレージ・システム・コントローラを実装して、動作させられるディスク・ストレージ・システム・コントローラの障害をカバーすることができる。この冗長ディスク・ストレージ・システム・コントローラもディスク・マネージャによって管理される。

【００５８】

ディスク・マネージャと他のサブシステムとの関係

ディスク・マネージャは、他の幾つかのサブシステムとやり取りを行う。ＲＡＩＤサブシステムは、データ・パス活動に関してディスク・マネージャが提供するサービスの重要なクライアントである。ＲＡＩＤサブシステムは、ディスク・マネージャを、Ｉ／Ｏのためのディスクへの独占的なパスとして使用する。また、ＲＡＩＤシステムでも、ディスク・マネージャからのイベントについて聴取を行い、ディスクの存在および動作状態の判定を行う。また、ＲＡＩＤサブシステムはディスク・マネージャと協働して、ＲＡＩＤ装置の構築のための範囲（extent）を割り当てる。管理コントロールは、ディスクの存在の情報を得るため、また動作状態の変化について情報を得るために、ディスク・イベントについて聴取を行う。本発明の一実施形態では、ＲＡＩＤサブシステム１０４は、ＲＡＩＤ－０、ＲＡＩＤ－１、ＲＡＩＤ－５、ＲＡＩＤ－１０などのＲＡＩＤタイプのうちの少なくとも１つからなる組合せを含むことができる。代替のＲＡＩＤサブシステムでは、ＲＡＩＤ－３、ＲＡＩＤ－４、ＲＡＩＤ－６、ＲＡＩＤ－７などのような他のＲＡＩＤタイプを使用できることも理解されよう。

【００５９】

本発明の一実施形態では、ディスク・マネージャは、コンフィギュレーション・アクセスのサービスを使用して、持続的なコンフィギュレーションを保存し、また、統計などのような過渡的な読み出し専用情報をプレゼンテーション・レイヤへ提示する。ディスク・マネージャは、こうしたパラメータへのアクセスのためにコンフィギュレーション・アクセスへのハンドラの登録を行う。

【００６０】

また、ディスク・マネージャは、ブロック・デバイスの存在および動作状態についての情報を得るためにＳＣＳＩ装置レイヤのサービスを使用するものであり、また、こうしたブロック・デバイスへのＩ／Ｏパスをもつ。ディスク・マネージャは、ディスクを一意に識別するための補助的な方法として、装置についてＳＣＳＩ装置サブシステムに問い合わせを行う。

【００６１】

データ・インスタント・リプレイおよびデータ・インスタント・フュージョン

また、本発明は、データ・インスタント・リプレイ（data instant replay）およびデータ・インスタント・フュージョン（data instant fusion）の方法を提供する。図３のＡおよびＢに、本発明の諸原理に従った、複数の時間間隔でのＲＡＩＤサブシステムのディスク・ストレージ・ブロックのスナップショットの概略図を示す。図３Ｃには、データ・インスタント・リプレイの方法３００を示し、この方法は、論理ブロックまたはディスク・ストレージ・ブロックのデフォルトのサイズを定義して、ＲＡＩＤサブシステムのディスク空間がストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスを形成するようにするステップ３０２と、ページ・プールのボリュームのスナップショットまたはディスク・ストレージ・ブロックのマトリックスのスナップショットを所定の時間間隔で時間的に生成するステップ３０４と、スナップショットまたはデルタ（delta）のアドレス・インデックスを、ストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスに保存して、ディスク・ストレージ・ブロックのマトリックスのスナップショットまたはデルタをその保存したアドレス・インデックスによって直ちに見つけ出せるようにするステップとを含む。

【００６２】

図３Ｂに示すように、所定の各時間間隔、例えば、Ｔ１（午後１２：００）、Ｔ２（午後１２：０５）、Ｔ３（午後１２：１０）、Ｔ４（午後１２：１５）などのような５分毎に、ストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスのスナップショットが自動的に生成される。ストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスの中のスナップショットまたはデルタのアドレス・インデックスは、ストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスに保存されて、ストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスのスナップショットまたはデルタを、その保存したアドレス・

インデックスによって直ちに見つけ出せるようにされる。

【００６３】

従って、このデータ・インスタント・リプレイの方法では、ＲＡＩＤサブシステムのスナップショットを、ユーザの定義する時間間隔や、ユーザのコンフィギュレーションする例えば数分や数時間ごとなどのような動的なタイム・スタンプや、サーバの指示する時刻に、時間的に生成する。システム障害やウイルス攻撃にあった場合、こうしたタイム・スタンプを有する仮想スナップショットにより、データ・インスタント・リプレイおよびデータ・インスンタント・リカバリが、約数分または数時間などで可能になる。この技法は、インスタント・リプレイ・フュージョンとも呼ばれる。即ち、クラッシュまたは攻撃のすぐ前のデータのフュージョンが遅れずに行われ、クラッシュまたは攻撃の前に保存したスナップショットを直ちにそれ以降の動作で使用することができる。

【００６４】

図４に、本発明の諸原理に従った、ＲＡＩＤサブシステムのディスク・ストレージ・ブロックの複数のスナップショットを使用することによるデータ・インスタント・フュージョン機能４００の概略図を更に示している。Ｔ３で、スナップショットの並列チェインＴ３’～Ｔ５’が生成され、それにより、フュージョンされたデータＴ３’によってフュージョンおよび／またはリカバリされたデータを、Ｔ４でフュージョンされるデータと置き換えるために使用することができる。同様に、スナップショットの複数の並列チェインＴ３’’、Ｔ４’’’を生成して、Ｔ４’～Ｔ５’およびＴ４’’～Ｔ５’’でフュージョンされるデータと置き換えることができる。一代替実施形態では、Ｔ４、Ｔ４’～Ｔ５’、Ｔ５’’でのスナップショットは、なおもページ・プールまたはマトリックスに保存することができる。

【００６５】

スナップショットをローカルのＲＡＩＤサブシステムまたはリモートのＲＡＩＤサブシステムに保存することができ、それにより、重大なシステム・クラッシュが、例えば、テロリスト攻撃などが原因で生じた場合にも、データ・インテグリティについて影響を受けず、データを直ちにリカバリできる。図５に、本発明の諸原理に従った、ＲＡＩＤサブシステムのディスク・ストレージ・ブロックのスナップショットを使用することによる、ローカル－リモートのデータ複製およびインスタント・リプレイ機能５００の概略図を示す。

【００６６】

リモート複製（replication）は、リモート・システムに対してボリューム・データを複製するサービスを行う。リモート複製では、ローカルとリモートとのボリュームを、可能な限り同期させた状態に保つことを試みる。一実施形態では、リモートのボリュームのデータが、ローカルのボリュームのデータの完全なコピーをミラーリングしていないこともあり得る。ネットワークの接続性（コネクティビティ）およびパフォーマンスに起因して、リモートのボリュームがローカルのボリュームと同期しない状態が引き起こされる場合もあり得る。

【００６７】

データ・インスタント・リプレイおよびデータ・インスタント・フュージョンの方法の別の特徴は、スナップショットを、システムが動作している状態で検査のために使用できることである。生のデータ（live data）をリアル・タイムの検査に使用することができる。

【００６８】

スナップショットおよびポイント・イン・タイム・コピー（ＰＩＴＣ）

データ・インスタント・リプレイの一例は、本発明の諸原理に従った、ＲＡＩＤサブシステムのディスク・ストレージ・ブロックのスナップショットを使用することである。スナップショットは、ボリュームへの書き込み動作を記録して、過去のボリュームの内容を見るためにビューを作成できるようにする。また、スナップショットは、ボリュームのそれまでのポイント・イン・タイム・コピー（Point-in-Time-Copy（特定の時にとられたコ

ピー）、ＰＩＴＣ）に対するビューを作成することによって、データ・リカバリのサポートを行う。

【００６９】

スナップショットのコアは、スナップショットの作成、合体（coalesce）、管理、およびＩ／Ｏ動作を実現する。スナップショットは、ボリュームへの書き込みを監視し、ボリュームのビューを通してアクセスのためにポイント・イン・タイム・コピー（ＰＩＴＣ）を作成する。スナップショットは、論理ブロック・アドレス（logical block address、ＬＢＡ）再マッピング・レイヤを、仮想化レイヤ内のデータ・パスへ追加する。これは、Ｉ／Ｏパス内の仮想ＬＢＡマッピングの別レイヤである。ＰＩＴＣは、すべてのボリューム情報をコピーしない場合もあり得、単に再マッピングで使用されるテーブルを変更するだけであり得る。

【００７０】

スナップショットは、ボリューム・データについての変更を追跡し、それまでのポイント・イン・タイムからのボリューム・データを見る機能を提供する。スナップショットは、この機能を、それぞれのＰＩＣＴに対するデルタ書き込みのリストを維持することによって、行う。

【００７１】

スナップショットは、ＰＩＴＣプロファイルのための複数の方法を提供するが、これは、アプリケーションにより開始されるもの、および時間により開始されるものを含む。スナップショットは、アプリケーションがＰＩＴＣを作成するようにする機能を提供する。アプリケーションは、作成を、サーバ上のＡＰＩを通して制御し、これはスナップショットＡＰＩへと送られる。また、スナップショットは、時間プロファイルを作成する機能を提供する。

【００７２】

スナップショットは、ジャーナル用システムの実装や、ボリュームへの書き込みのすべてのリカバリを行わない場合もあり得る。スナップショットは、ＰＩＴＣウィンドウ内の一つのアドレスへの最後の書き込みを保持するのみであり得る。スナップショットにより、ユーザは、例えば分や時間などの定められた短期間をカバーするＰＩＴＣを作成することが可能とされる。スナップショットは、障害を取り扱うために、すべての情報をディスクに書き込む。スナップショットは、デルタ書き込み（delta writes）を含むボリューム・データ・ページ・ポインタを維持する。テーブルはマップをボリューム・データへ提供するものであり、これがなければボリューム・データへはアクセス不能であるため、テーブル情報はコントローラ障害の場合を取り扱わなければならない。

【００７３】

ビュー・ボリューム機能（ボリュームを見る機能）により、ＰＩＴＣへのアクセスが提供される。ビュー・ボリューム機能は、アクティブなＰＩＴＣを除き、そのボリューム内の任意のＰＩＴＣに付加させることができる。ＰＩＴＣへの付加は、比較的迅速な動作である。ビュー・ボリューム機能の用途は、検査、訓練、バックアップ、およびリカバリを含む。ビュー・ボリューム機能は、書き込み動作を可能にし、その基礎とされている基礎的ＰＩＴＣの変更は行わない。

【００７４】

一実施形態では、スナップショットは、ディスク空間を使用して、パフォーマンスを最適化し、使用を容易にするように設計されおり、それは下記のようである。

【００７５】

・スナップショットにより、ユーザ要求に対する応答時間が速くなる。ユーザ要求には、Ｉ／Ｏ、ＰＩＴＣの作成、ビュー・ボリュームの作成／削除が含まれる。これを達成するために、スナップショットは、必要最小量よりも多くのディスク空間を使用してテーブル情報を保存する。Ｉ／Ｏについては、スナップショットは、ボリュームの現在の状態を要約して一つのテーブルにして、読み出しおよび書き込みの要求をすべて一つのテーブルで満たすことができるようにする。スナップショットは、通常のＩ／Ｏ動作に対する影響

をできる限り低減する。第２に、ビュー・ボリューム動作については、スナップショットは、メイン・ボリュームのデータ・パスと同じテーブル機構を使用する。

【００７６】

・スナップショットは、コピーされるデータの量を最小化する。このためには、スナップショットは、それぞれのＰＩＴＣについてのポインタのテーブルを維持する。スナップショットはポインタのコピーおよび移動を行うが、ボリューム上のデータの移動は行わない。

【００７７】

・スナップショットは、固定サイズのデータ・ページを用いてボリュームの管理を行う。個々のセクタを追跡するには、一つの手頃なサイズのボリュームでも膨大な量のメモリが必要になる。セクタよりも大きなデータ・ページを使用することにより、ある種のページは、別のページから直接に複製された情報の一部を含むことができる。

【００７８】

・スナップショットは、ボリューム上のデータ空間を使用して、データ・ページ・テーブルを保存する。ルックアップ・テーブルは、コントローラ障害の後に複製される。ルックアップ・テーブルは、ページを割り当て、それをさらに分割する。

【００７９】

・スナップショットがコントローラ障害を取り扱うが、その際、一実施形態では、スナップショットを用いるボリュームが一つのコントローラ上で動作することが必要とされる。この実施形態ではコヒーレンシは必要ない。ボリュームへのすべての変更は、代替のコントローラによるリカバリのために、ディスクまたは信頼できるキャッシュへ記録される。コントローラ障害からのリカバリの際には、一実施形態では、スナップショット情報をディスクから読み取ることが必要である。

【００８０】

・スナップショットは、仮想ＲＡＩＤインターフェースを使用して、ストレージへのアクセスを行う。スナップショットは、複数のＲＡＩＤ装置を単一のデータ空間として使用することができる。

【００８１】

・スナップショットは、ボリュームあたり「ｎ」個のＰＩＴＣおよびボリュームあたり「ｍ」個のビューをサポートする。「ｎ」および「ｍ」に対する制限は、ディスク空間およびコントローラのメモリの関数となる。

【００８２】

ボリュームおよびボリュームの割り当て／レイアウト

スナップショットは、ＬＢＡ再マッピング・レイヤをボリュームに追加する。再マッピングは、Ｉ／Ｏ要求ＬＢＡおよびルックアップ・テーブルを使用して、アドレスをデータ・ページへと変換する。図６に示すように、スナップショットを用いた提示されるボリュームは、スナップショットのないボリュームと同じふるまいをする。それは、線形ＬＢＡ空間を有し、Ｉ／Ｏ要求を取り扱う。スナップショットは、ＲＡＩＤインターフェースを使用してＩ／Ｏを行うものであり、複数のＲＡＩＤ装置を１つのボリュームに含める。一実施形態では、スナップショット・ボリュームに対するＲＡＩＤ装置のサイズは、提示されるボリュームのサイズではない。ＲＡＩＤ装置により、スナップショットは、ボリューム内でデータ・ページ用の空間を拡張することを可能とされる。

【００８３】

新しいボリュームは、初めにスナップショットをイネーブルにしてあると、新しいデータ・ページ用の空間を含めればよいだけである。スナップショットは、ボトム・レベルのＰＩＴＣに置くページのリストを作成しない。ボトム・レベルＰＩＴＣは、この場合には空である。割り当て時に、すべてのＰＩＴＣページはフリー・リスト上にある。初めにスナップショットをイネーブルにしたボリュームを作することにより、ボリュームが提示するよりも少ない物理空間を割り当て得る。スナップショットは、ボリュームへの書き込みを追跡する。本発明の一実施形態では、ヌル（ＮＵＬＬ）・ボリュームは、ページ・プー

ルまたはマトリックスへコピーおよび／または保存されず、これにより、ストレージ空間の使用の効率が高まる。

【００８４】

一実施形態では、どちらの割り当て方式でも、ＰＩＴＣが、仮想ＮＵＬＬボリュームをリストの底部に置く。ＮＵＬＬボリュームに対する読み出しは、ゼロのブロックを返す。ＮＵＬＬボリュームは、サーバがそれまで書き込んでいないセクタを取り扱う。ＮＵＬＬボリュームへの書き込みは起こり得ない。ボリュームは、ＮＵＬＬボリュームを、書き込まれていないセクタに対する読み出しのために使用する。

【００８５】

空きページの数は、ボリュームのサイズ、ＰＩＴＣの数、および予測されるデータ変化レートに依存する。システムは、所与のボリュームに対しての割り当てるページの数を決定する。データ・ページの数は、時間が経つにつれて拡張される可能性がある。拡張により、予測されるよりも急速なデータの変化、より多くのＰＩＴＣ、またはより大きなボリュームを、サポートすることができる。新しいページはフリー・リストへ追加される。フリー・リストへのページの追加は自動的に行ってもよい。

【００８６】

スナップショットは、ボリューム空間の管理のためにデータ・ページを使用する。それぞれのデータ・ページは数メガバイトのデータを含み得る。オペレーティング・システムを使用すると、ボリュームの同じ領域へ多くのセクタが書き込まれる傾向がある。メモリ要件からも、スナップショットがボリュームの管理のためにページを使用することが要求される。１テラバイトのボリュームの各セクタについて一つの３２ビットポインタを維持すると、８ギガバイトのＲＡＭが必要となり得る。異なるボリュームは異なるページ・サイズを有し得る。

【００８７】

図７に、スナップショット構造の一実施形態を示す。スナップショットは、多くのオブジェクトをボリューム構造に付加する。付加されるオブジェクトとしては、ＰＩＴＣ、アクティブなＰＩＴＣへのポインタ、データ・ページ・フリー・リスト、子のビュー・ボリューム、およびＰＩＴＣ合体オブジェクトがある。

【００８８】

・アクティブＰＩＴＣ（ＡＰ）ポインタはボリュームにより維持される。ＡＰは、ボリュームに対する読み出しおよび書き込みの要求のマッピングを取り扱う。ＡＰは、ボリューム内のすべてのデータの現在の場所の要約（一覧、summary）を含む。

【００８９】

・データ・ページ・フリー・リストは、ボリューム上の使用可能なページの追跡を行う。

【００９０】

・オプションの子ビュー・ボリューム（child view volume）は、ボリュームＰＩＴＣへのアクセスを提供する。ビュー・ボリュームは、ＰＩＴＣへの書き込みを記録するためにそれ自体のＡＰを含むが、基礎となるデータの変更は行わない。１つのボリュームが複数の子ビュー・ボリュームをサポートしてもよい。

【００９１】

・スナップショット合体オブジェクトは、以前のＰＩＴＣを取り除く目的で２つのＰＩＴＣを一時的にリンクする。ＰＩＴＣの合体は、データ・ページの所有権を移動し、データ・ページを解放することを含む。

【００９２】

・ＰＩＴＣは、そのＰＩＴＣがアクティブであった間に書き込まれたページに対するテーブルおよびデータ・ページを含む。ＰＩＴＣは、ＰＩＴＣが書き込み要求の受理を停止した時点の凍結タイム・スタンプを含む。また、ＰＩＴＣは、何れの時間にＰＩＴＣが合体することになるかを決めるＴｉｍｅ－ｔｏ－Ｌｉｖｅ（タイム・ツー・ライブ、生きる時間）値を含む。

【００９３】
　また、スナップショットは、ボリューム全体に対するデータ・ページ・ポインタの要約を、ＰＩＴＣが取られた時点で行って、読み出しおよび書き込みの予測可能なパフォーマンスを提供する。他の解決策は、最新のポインタを見つけるために複数のＰＩＴＣを検査するために、読み出しを必要とし得る。こうした解決策では、テーブル・キャッシング・アルゴリズムが必要であるが、パフォーマンスは最低となる場合がある。
【００９４】
　また、本発明におけるスナップショットを要約することにより、テーブルの最悪のメモリ使用が低減される。その場合、テーブル全体をメモリへロードすることを必要とされ得るが、単一のテーブルだけをロードすることを必要とされ得る。
【００９５】
　要約（一覧）は、現在のＰＩＴＣの所有するページを含むが、それまでのすべてのＰＩＴＣからのページを含むことができる。ＰＩＴＣがどのページに書き込みを行うかを決定するために、それぞれのデータ・ページについてのページ所有権を追跡する。また、一覧は、合体プロセスに対する所有権の追跡も行う。これを取り扱うために、データ・ページ・ポインタはページ・インデックスを含む。
【００９６】
　図８に、ＰＩＴＣのライフ・サイクルの一実施形態を示す。それぞれのＰＩＴＣは、読み出し専用として記憶（コミット）される前に、多数の下記の状態を経過する。
【００９７】
　１．テーブル作成　－　作成を行うと、テーブルが作成される。
【００９８】
　２．ディスクへのコミット（記憶、commit）　－　これは、ＰＩＴＣに対してディスク上にストレージを生成する。この時点でテーブルを書き込むことにより、テーブル情報の保存に必要な空間が、ＰＩＴＣが取られる前に割り当てられることが保証される。同時に、ＰＩＴＣオブジェクトもディスクへ記憶（コミット）される。
【００９９】
　３．Ｉ／Ｏの受け入れ　－　これはＰＩＴＣはアクティブＰＩＴＣ（ＡＰ）になっている。　－　ここでは、ＰＩＴＣは、ボリュームに対する読み出しおよび書き込みの要求を取り扱う。これは、テーブルへの書き込み要求を受理する唯一の状態である。ＰＩＴＣは、いまアクティブであるというイベントを生成する。
【０１００】
　４．ディスクへのテーブルの記憶　－　ＰＩＴＣはもはやＡＰではなく、もはやそれ以上のページを受理しない。新しいＡＰによる引き継ぎが行われている。この時点より後では、テーブルは、合体動作中に取り除かれない限り、変更されない。これは読み出し専用である。ＰＩＴＣは、この時点で、凍結され記憶されているというイベントを生成する。何れのサービスもそのイベントを聴取することができる。
【０１０１】
　５．テーブルのメモリの解放（リリース）　－　テーブルが必要とするメモリを解放する。また、このステップは、変更がすべてディスクへと書き込まれていることを示すために、ログをクリアする。
【０１０２】
　ボリュームまたはビュー・ボリュームのためのトップ・レベルＰＩＴＣは、アクティブＰＩＴＣ（ＡＰ）と呼ばれる。ＡＰは、ボリュームへのすべての読み出しおよび書き込みの要求を満たす。ＡＰは、そのボリュームに対しての唯一の、書き込み要求を受理できるＰＩＴＣである。ＡＰは、ボリューム全体に対するデータ・ページ・ポインタの一覧を含む。
【０１０３】
　ＡＰは、合体プロセスに関して、宛先（destination）となることはできるが、ソース（source）となることはできない。宛先である場合、ＡＰは、所有するページの数を増加

するが、データのビューは変更しない。

【０１０４】

ボリュームの拡張に関しては、ＡＰはボリュームとともに直ちに大きくなる。新しいページは、ＮＵＬＬボリュームを指す。ＡＰではないＰＩＴＣは、ボリュームの拡張のために変更は必要ではない。

【０１０５】

それぞれのＰＩＴＣは、入ってくるＬＢＡを、基礎となるボリュームへのデータ・ページ・ポインタへとマッピングするテーブルを維持する。テーブルは、データ・ページへのポインタを含む。テーブルは、提示される論理空間よりも多くの物理ディスク空間をアドレスする必要がある。図９に、マルチ・レベル・インデックスを有するテーブル構造の一実施形態を示す。この構造は、ボリュームＬＢＡを復号化してデータ・ページ・ポインタにする。各レベルでは、図９に示すように、アドレスの増加していくＬＳＢ（less significant bit）を復号化する。このテーブルの構造により、高速のルックアップおよびボリュームの拡張の機能が提供される。高速ルックアップのために、マルチ・レベル・インデックス構造は、テーブルを浅い状態に維持し、各レベルで複数のエントリをもつようにしている。このインデックスは、各レベルでアレイのルックアップを行う。ボリュームの拡張をサポートするため、マルチ・レベル・インデックス構造により、拡張をサポートするための別のレイヤを追加することが可能になっている。この場合のボリュームの拡張は、上位レイヤへ提示されるＬＢＡカウント（計数）の拡張であり、そのボリュームへ割り当てられるストレージ空間の実際の量ではない。

【０１０６】

マルチ・レベル・インデックスは、ボリューム全体のデータ・ページ再マッピングの一覧（要約）を含む。それぞれのＰＩＴＣは、記憶されたポイント・イン・タイム（時（特定の時））の、このボリュームに対する完全な再マッピング・リストを含む。

【０１０７】

マルチ・レベル・インデックス構造は、テーブルのレベルに対して異なるエントリ・タイプを使用する。異なるエントリ・タイプは、情報をディスクから読み出す必要性や、情報をメモリに格納する必要性をサポートする。ボトム・レベルのエントリは、データ・ページ・ポインタのみを含み得る。トップおよび中間のレベルのエントリは２つのアレイを含み、１つは次のレベルのテーブル・エントリのＬＢＡであり、そして、テーブルへのメモリ・ポインタを含む。

【０１０８】

提示されるボリューム・サイズが拡張する際に、以前のＰＩＴＣテーブルのサイズは増やす必要はなく、また、テーブルを変更する必要もない。テーブルの中の情報は、読み出し専用であるので、変わらず、そして、拡張プロセスは、テーブルを、ＮＵＬＬページ・ポインタを末尾に追加することによって、変更する。スナップショットは、以前のＰＩＴＣからのテーブルを直接にユーザに提示しない。

【０１０９】

Ｉ／Ｏ動作があると、テーブルには、ＬＢＡをデータ・ページ・ポインタへマッピングすることが求められる。その場合、Ｉ／Ｏでは、データ・ページ・ポインタにデータ・ページ・サイズを乗算して、基礎となるＲＡＩＤのＬＢＡを得る。一実施形態では、データ・ページ・サイズは、２のべき乗である。

【０１１０】

テーブルは、ＬＢＡの再マッピング、ページの追加、およびテーブルの合体のためのＡＰＩを提供する。

【０１１１】

スナップショットは、ＰＩＴＣオブジェクトおよびＬＢＡマッピング・テーブルを保存するためにデータ・ページを使用する。テーブルは、そのテーブル・エントリに対するＩ／ＯのためにＲＡＩＤインターフェースへ直接にアクセスする。テーブルは、ＲＡＩＤ装置に対してそのテーブルの読み出しおよび書き込みを行うときの変更を最小にする。変更

がない場合には、テーブルの情報の読み出しおよび書き込みを、テーブル・エントリ構造に対して直接に行うことが可能になる。これにより、Ｉ／Ｏのために必要となるコピーが低減される。スナップショットは、ディスク上のホット・スポットの発生を防ぐために、変化ログを使用することができる。ホット・スポットとは、ボリュームへの更新を追跡するために繰り返し使用される場所のことである。変化ログは、ＰＩＴＣテーブルへの更新、およびボリュームに関するフリー・リストを記録する。リカバリ中には、スナップショットは変化ログを使用して、メモリ内ＡＰ（in-memory AP）およびフリー・リスト（free list）の再作成を行う。図１０にテーブルのリカバリの一実施形態を示し、これにより、メモリ内ＡＰと、ディスク上ＡＰ（on-disk AP）と、変化ログとの間の関係を示している。この図は、フリー・リストに関しても同じ関係を示している。メモリ内ＡＰテーブルは、ディスク上ＡＰテーブルおよびログから再構築することができる。どのようなコントローラ障害の場合でも、ＡＰは、ディスク上ＡＰを読み出してそれにログ変化を適用することによって、再構築される。変化ログは、システム・コンフィギュレーションに応じて、異なる物理資源を使用する。多重コントローラのシステムでは、変化ログは、ストレージのためにバッテリーバックアップされたキャッシュ・メモリを頼りにしている。キャッシュ・メモリを使用することにより、スナップショットは、ディスクへのテーブル書き込みの数を減らし且つデータ・インテグリティを維持することができる。変化ログは、リカバリのためにバックアップ・コントローラに対しての複製を行う。単一コントローラのシステムでは、変化ログは、すべての情報をディスクへ書き込む。これには、ディスク上のログの場所にホット・スポットを発生させるという副作用がある。これにより、幾つもの変化を単一のデバイス・ブロックへ書き込むことが可能とされる。

【０１１２】

スナップショットは、定期的に、ＰＩＴＣテーブルおよびフリー・リストをディスクへ書き込んで、ログにおけるチェックポイントの作成およびそのクリアを行う。この期間は、ＰＩＴＣへの更新の数に応じて変わり得る。合体プロセスは変化ログを使用しない。

【０１１３】

スナップショットのデータ・ページＩ／Ｏは、要求がデータ・ページ境界内に収まることを必要とし得る。スナップショットは、ページ境界にまたがるＩ／Ｏ要求にであった場合に、その要求を分割する。次いで、スナップショットは、その要求を要求ハンドラへと渡す。書き込みおよび読み出しのセクションは、Ｉ／Ｏがページ境界内に収まることを仮定している。ＡＰは、ＬＢＡ再マッピングを提供してＩ／Ｏ要求を満たす。

【０１１４】

ＡＰはすべての書き込み要求を満たす。スナップショットは、所有ページと非所有ページに対する２つの異なる書き込みシーケンスをサポートする。異なるシーケンスにより、テーブルへのページの追加が可能になる。図１１に、所有ページ・シーケンスおよび非所有ページ・シーケンスがある書き込みプロセスの一実施形態を示す。

【０１１５】

所有ページ・シーケンスでは、プロセスは以下のものを含む。

１）テーブル・マッピングを見つける。

２）所有の書き込みをページングする（Page Owned Write）　－　ＬＢＡを再マッピングし、データをＲＡＩＤインターフェースへ書き込む。

【０１１６】

以前に書き込まれたページは、純粋な書き込み要求である。スナップショットは、データをそのページに書き込んで、現在の内容を上書きする。ＡＰの所有するデータ・ページだけが書き込まれる。他のＰＩＴＣの所有するページは、読み出し専用である。

【０１１７】

非所有ページ・シーケンスでは、プロセスは以下のものを含む。

１）テーブル・マッピングを見つける。

２）以前のページを読み出す　－　そのデータ・ページに対する読み出しを行い、書き込み要求と読み取ったデータとでページ全部が出来上がるようにする。これが、コピー・

オン・ライト（copy on write）・プロセスの開始となる。
３）データを組み合わせる　－　データ・ページの読み出しおよび書き込みの要求のペイロードを、単一の連続するブロックに入れる。
４）リスト割り当てを自由にする　－　新しいデータ・ページ・ポインタを、フリー・リストから得る。
５）組み合わせたデータを、新しいデータ・ページへ書き込む。
６）新しいページ情報をログへ記憶する。
７）テーブルを更新する　－　テーブル内のＬＢＡ再マッピングを変更して、新しいデータ・ページ・ポインタを反映させる。これでデータ・ページはＰＩＴＣにより所有される。

【０１１８】
ページを追加するには、そのページがテーブルに追加されるまで、読み出しおよび書き込みの要求を妨げることが必要であり得る。スナップショットは、テーブルの更新をディスクへ書き込み、ログの複数のキャッシュ記憶されたコピーを維持することにより、コントローラのコヒーレンシを達成する。

【０１１９】
読み出し要求に関して、ＡＰはすべての読み出し要求を満たす。読み出し要求は、ＡＰテーブルを用いて、ＬＢＡを、そのデータ・ページのＬＢＡへ再マッピングする。再マッピングされたＬＢＡはＲＡＩＤインターフェースへと渡されて、その要求が満たされる。ボリュームは、それまでにそのボリュームへ書き込まれていないデータ・ページに対する読み出し要求を満たすことができる。こうしたページは、ＰＩＴＣテーブルにおいてＮＵＬＬアドレス（すべて１）でマークされている。このアドレスに対する要求は、ＮＵＬＬボリュームにより満たされ、一定のデータ・パターンを返す。異なるＰＩＴＣテーブルの所有するページは、ページ境界にまたがる読み出し要求を満たすことができる。

【０１２０】
スナップショットは、ＮＵＬＬボリュームを使用して、以前に書き込まれていないデータ・ページに対する読み出し要求を満たす。これは、それぞれのセクタ読み出しに対して、すべてゼロを返す。これは、ＲＡＩＤ装置も、割り当てられた空間もない。ＮＵＬＬボリュームに対する読み出しについてのデータ要件を満たすために、すべてゼロであるブロックをメモリ内に置いておくことが予期される。すべてのボリュームは、読み出し要求を満たすためにＮＵＬＬボリュームを共用する。

【０１２１】
一実施形態では、合体プロセスが、ＰＩＴＣと、その所有ページの一部とをボリュームから取り除く。ＰＩＴＣを取り除くと、新しい差異を追跡するために使用可能な空間がより多く作成される。合体では、２つの隣接するテーブルを差異があるかどうか比較し、新しい差異だけを維持する。合体は、ユーザによるコンフィギュレーションに従って、定期的にまたは手動で起きる。

【０１２２】
このプロセスは、２つのＰＩＴＣ、即ち、ソースと宛先とを含む。一実施形態での、適格なオブジェクトについての規則は、次のようなものである。
１）ソースは、宛先に対して、以前のＰＩＴＣでなければならない　－　ソースは宛先より前に作成されなければならない。
２）宛先は、同時にソースであってはならない。
３）ソースは、複数のＰＩＴＣにより参照されてはならない。複数の参照は、ビュー・ボリュームがＰＩＴＣから作成されたときに、起きる。
４）宛先は、複数の参照をサポートし得る。
５）ＡＰは宛先であり得るが、ソースではない。

【０１２３】
合体・プロセスは、すべての変更をディスクへ書き込むものであり、コヒーレンシを必要としない。コントローラが故障した場合に、ボリュームがＰＩＴＣ情報をディスクから

リカバリし、合体プロセスを再開する。
【０１２４】
このプロセスは、２つのＰＩＴＣを合体のためにマークするものであり、以下の諸ステップを含む。
【０１２５】
１）ソースの状態が合体のソースに設定される　－　この状態は、コントローラ障害のリカバリのために、ディスクへ記憶される。この時点で、ソースはもはやアクセスされないが、それは、そのデータ・ページが無効であるからである。データ・ページがフリー・リストへ返されるか、または所有権が宛先へ移される。
【０１２６】
２）宛先の状態が合体の宛先へ設定される　－　この状態は、コントローラ障害のリカバリのためにディスクへ記憶される。
【０１２７】
３）テーブルのロードおよび比較を行う　－　このプロセスは、データ・ページ・ポインタを移動させる。解放されたデータ・ページは、直ちにフリー・リストに追加される。
【０１２８】
４）宛先の状態が正常に設定される　－　プロセスが完了する。
【０１２９】
５）リストを調整する　－　ソースのｎｅｘｔ（次）ポインタのｐｒｅｖｉｏｕｓ（以前）を、宛先へと変更する。これにより、ソースがリストから効率よく取り除かれる。
【０１３０】
６）ソースを解放する　－　制御の情報に使用された何れのデータ・ページをも、フリー・リストへ戻す。
【０１３１】
上述のプロセスは、２つのＰＩＴＣの組合せをサポートしている。合体を、単一のパスにおいて複数のＰＩＴＣの削除と複数のソースの作成とを行うように設計できることが、当業者には理解されている。
【０１３２】
図２に示すように、ページ・プールは、そのページ・プールに関連するすべてのボリュームが使用するように、データ・ページ・フリー・リストを維持する。フリー・リスト・マネージャは、ページ・プールからのデータ・ページを使用して、フリー・リストを永久的ストレージへ記憶する。フリー・リストの更新を引き起こすものは幾つもあり、書き込みプロセスはページを割り当て、制御ページ・マネージャはページを割り当て、合体用プロセスはページを返す。
【０１３３】
フリー・リストは、それ自体をある閾値のところで自動的に拡張するためのトリガを維持することができる。このトリガは、ページをページ・プールへ追加するためのページ・プール拡張方法を使用する。自動的な拡張は、ボリューム・ポリシの一機能とすることもできる。重要度の高いデータ・ボリュームには拡張を可能にし、重要度の低いボリュームは強制的に合体させる。
【０１３４】
ビュー・ボリュームは、以前のポイント・イン・タイムへのアクセスを提供し、正常のボリュームＩ／Ｏ動作をサポートする。ＰＩＴＣは、ＰＩＴＣ間の差異を追跡し、ビュー・ボリュームは、ＰＩＴＣ内に含まれる情報へユーザがアクセスできるようにする。ビュー・ボリュームはＰＩＴＣから分岐する。ビュー・ボリュームは、リカバリ、検査、バックアップ動作などをサポートする。ビュー・ボリュームの作成は、データ・コピーを必要としないので、ほぼ瞬間的に生じる。ビュー・ボリュームは、ビュー・ボリュームへの書き込みをサポートするために、それ自体のＡＰが必要になり得る。
【０１３５】
ボリュームの現在の状態から取られたビューでは、ＡＰを現在のボリュームのＡＰから

コピーすることができる。ＡＰを用い、ビュー・ボリュームは、ビュー・ボリュームのへの書き込み動作を、基礎となるデータを変更せずに行うことを可能にする。ＯＳは、データを使用するために、ファイル・システムまたはファイルの再構築を必要とし得る。ビュー・ボリュームは、ＡＰおよび書き込まれたデータ・ページに対して親ボリュームから空間を割り当てる。ビュー・ボリュームは、関連するＲＡＩＤ装置情報を有さない。ビュー・ボリュームを削除すると、空間が解放され親ボリュームへ戻される。

【０１３６】

図１２に、スナップショットを用いてのあるボリュームに関する推移を示す例示的なスナップショット動作を示す。図１２は、１０ページをもつボリュームを示す。それぞれの状態（ステート）は、ボリュームに対する読み出し要求遂行（Read Request Fulfillment）リストを含む。影を付けたブロックは、所有データ・ページ・ポインタを示す。

【０１３７】

図の左側（即ち、初期状態）から図の中央への推移は、ページ３および８への書き込みを示している。ページ３への書き込みには、ＰＩＴＣ　Ｉ（ＡＰ）への変更が必要である。ＰＩＴＣ　Ｉは、新ページ書き込み処理に従い、ページ３をテーブルへ追加する。ＰＩＴＣは、変更されていない情報をページＪから読み出し、ドライブ・ページＢを使用してそのページを保存する。このＰＩＴＣのページ３へのこれ以降のすべての書き込みは、ページを移動させずに取り扱われる。ページ８への書き込みでは、ページへの書き込みに関する第２の場合を示す。ＰＩＴＣ　Ｉはすでにページ８を含んでいるため、ＰＩＴＣ　Ｉは、ページ８のそのデータのその部分へ書き込む。この場合、それはドライブ・ページＣに存在する。

【０１３８】

図の中央から図の右側（即ち、最終状態）への推移は、ＰＩＴＣ　ＩＩとＩＩＩとの合体を示す。スナップショットの合体は、それぞれ、古いページを削除すること、および両方のＰＩＴＣにおけるすべての変更を維持することを含む。両方のＰＩＴＣがページ３および８を含む。プロセスは、ＰＩＴＣ　ＩＩからのの新しい方のページを保ち、ＰＩＴＣ　ＩＩＩからのページを解放し、ページＡおよびＤをフリー・リストへ返す。

【０１３９】

スナップショットは、フリー・リストおよびＰＩＴＣテーブル情報を保存するために、ページ・プールからデータ・ページの割り当てを行う。制御ページの割り当ては、データ・ページを、オブジェクトに必要とされるサイズに合うように、更に細かく割り当てる（sub-allocate）。

【０１４０】

ボリュームは、制御ページ情報の先頭に対するページ・ポインタを含む。このページから、他のすべての情報を読み出すことができる。

【０１４１】

スナップショットは、特定の時間間隔で、使用中のページの数を追跡する。これにより、スナップショットは、いつユーザがシステムに更なる物理ディスク空間を追加する必要があるかを予測して、スナップショットが不足するのを防ぐ。

【０１４２】

データ・プログレッション

本発明の一実施形態では、データ・プログレッション（ＤＰ、data progression）を使用して、データを、徐々に、適切なコストのストレージ空間へと移動させる。本発明により、ユーザは、ドライブが実際に必要であるときにドライブの追加を行える。これにより、ディスク・ドライブのコスト全体が著しく低減される。

【０１４３】

データ・プログレッションでは、最近アクセスされていないデータおよび履歴のスナップショット・データを、より安価なストレージへ移動させる。最近アクセスされていないデータについては、最近アクセスされていないページがに対してのストレージのコストを徐々に低減する。データは、コストが最も低いストレージへ直ちに移動させない。履歴の

スナップショット・データについては、読み出し専用ページを、例えばＲＡＩＤ５などのようなより効率のよいストレージ空間へ移動させ、また、そのページにもはやボリュームからアクセス可能でない場合には、最も安価なストレージへ移動される。

【０１４４】

本発明のデータ・プログレッションの他の利点は、現在アクセス中のデータへの高速のＩ／Ｏアクセスを維持すること、および高速であるが高価なディスク・ドライブを購入する必要性を減らすことを含む。

【０１４５】

データ・プログレッションでは、動作時に、ストレージのコストを、物理メディアのコストおよびデータ保護に使用されるＲＡＩＤ装置の効率を用いて、判定する。また、データ・プログレッションでは、ストレージ効率を判定し、それに従ってデータの移動を行う。例えば、データ・プログレッションは、物理ディスク空間をより効率よく使用するために、ＲＡＩＤ１０装置をＲＡＩＤ５装置へと変換することができる。

【０１４６】

データ・プログレッションでは、アクセス可能なデータを、サーバが現在読み出しまたは書き込みできるデータと定義している。データ・プログレッションでは、アクセス可能性（accessibility）を使用して、ページが使用すべきストレージのクラスを決定する。ページは、履歴ＰＩＴＣに属する場合には、読み出し専用である。サーバが、一番最近のＰＩＴＣにおいてそのページを更新していない場合には、そのページはまだアクセス可能である。

【０１４７】

図１７に、データ・プログレッション動作におけるアクセス可能なデータ・ページの一実施形態を示す。アクセス可能なデータ・ページは、以下のカテゴリに分かれる。

【０１４８】

・アクセス可能な、最近アクセスされたもの　－　これらは、ボリュームが最もよく使用しているアクティブ・ページである。

【０１４９】

・アクセス可能な、最近アクセスされていないもの　－　最近使用されていない読み出し／書き込みページ。

【０１５０】

・履歴のアクセス可能なもの　－　ボリュームによる読み出しが行える読み出し専用ページ　－－　スナップショット・ボリュームに当てはまる。

【０１５１】

・履歴のアクセス不能なもの　－　ボリュームによって現在アクセスされていない読み出し専用データ・ページ　－－　スナップショット・ボリュームに当てはまる。スナップショットは、こうしたページをリカバリ目的で維持し、ページは、一般に、可能な限り低コストのストレージに置かれる。

【０１５２】

図１７に、スナップショット・ボリューム用の様々な所有ページを有する３つのＰＩＴＣを示している。動的容量のボリュームは、ＰＩＴＣ　Ｃだけで代表させてある。すべてのページは、アクセス可能であり読み出し／書き込みである。ページは異なるアクセス時間を有し得る。

【０１５３】

下記のテーブルは、様々なストレージ装置を、効率の高くなる順または金銭的な支出の減る順に示す。ストレージ装置のリストは、より遅い書き込みＩ／Ｏアクセスの一般的な順序にも従い得る。データ・プログレッションでは、論理保護空間の効率をＲＡＩＤ装置の総物理空間で割ったものを計算する。

【０１５４】

【表１】

表１：ＲＡＩＤタイプ

タイプ	サブタイプ	ストレージ効率	１ブロック書込Ｉ／Ｏカウント	使用
ＲＡＩＤ１０		５０％	２	比較的良い書き込み性能をもつ一次的な読出／書込アクセス可能ストレージ。
ＲＡＩＤ５	３ドライブ	６６．６％	４（２読み出し２書き込み）	ＲＡＩＤ１０を超えての効率の増加は最小であり、ＲＡＩＤ５の書き込みペナルティが生じる。
ＲＡＩＤ５	５ドライブ	８０％	４（２読み出し２書き込み）	読み出し専用の履歴情報用の優れた候補。最近アクセスされていない書き込み可能ページ用の良い候補。
ＲＡＩＤ５	９ドライブ	８８．８％	４（２読み出し２書き込み）	読み出し専用の履歴情報用の優れた候補。
ＲＡＩＤ５	１７ドライブ	９４．１％	４（２読み出し２書き込み）	効率の増加は低減されるが、ＲＡＩＤ装置のフォールト・ドメインは２倍になる。

【０１５５】

ＲＡＩＤ５の効率は、ストライプのドライブ数が増えると高くなる。ストライプのディスク数が増えると、フォールト・ドメインが増大する。ストライプのドライブ数の増加により、ＲＡＩＤ装置の作成に必要なディスクの最小数も増える。一実施形態では、データ・プログレッションは、９ドライブを超えるＲＡＩＤ５ストライプ・サイズを使用しない。なぜなら、フォールト・ドメイン・サイズが増え、効率の増加が限られるからである。データ・プログレッションは、スナップショット・ページ・サイズの整数倍であるＲＡＩＤ５ストライプ・サイズを使用する。これにより、データ・プログレッションは、ページをＲＡＩＤ５へと移動させるときにフル・ストライプ書き込みを行えるようになり、移動の効率を高める。すべてのＲＡＩＤ５コンフィギュレーションは、データ・プログレッションの目的で、同じ書き込みＩ／Ｏ特性を有する。例えば、２．５インチ（６．５ｃｍ）ＦＣディスク上のＲＡＩＤ５は、そうしたディスクの性能を効率よく使用しない可能性がある。この組合せを防ぐために、データ・プログレッションは、あるＲＡＩＤタイプが特定のディスクタイプ上で動作しないようにする機能をサポートする必要がある。また、データ・プログレッションのコンフィギュレーションは、システムがＲＡＩＤ１０またはＲＡＩＤ５の空間を使用しないようにすることもできる。

【０１５６】

ディスクのタイプを次のテーブルに示す。

【０１５７】

【表２】

表２：ディスク・タイプ

タイプ	速度	費用	論点
2.5インチFC	非常に良い	高い	非常に高価
ＦＣ１５ＫRPM	良い	中程度	高価
ＦＣ１０ＫRPM	良い	良い	妥当な価格
SATA	ほどほど	低い	安価／信頼性が低い

【０１５８】

データ・プログレッションは、システム内のドライブに相対的なディスク・ドライブの分類を自動的に行う機能を含む。システムは、ディスクを検査して、そのパフォーマンスをシステム内の他のディスクと比較して決定する。より速いディスクは高評価区分へと分類され、より遅いディスクは低評価区分へと分類される。ディスクがシステムへ追加され

ると、システムは、ディスクの評価分類のバランスを自動的にとり直す。このアプローチでは、決して変化することのないシステムと、新しいディスクが追加されると頻繁に変化するシステムとの双方を取り扱う。この自動分類は、複数のドライブ・タイプを同じ評価区分内に入れることがある。ドライブの評価がかなり近接しているものと判定された場合には、その評価は同じになる。

【０１５９】

一実施形態では、システムは次のドライブを含む。即ち、

高　－　１０Ｋ　ＦＣドライブ

低　－　ＳＡＴＡドライブ

【０１６０】

１５Ｋ　ＦＣドライブを追加すると、データ・プログレッションは、ディスクを自動的に再分類し、１０Ｋ　ＦＣドライブを降格させる。この結果として次の分類となる。

高　－　１５Ｋ　ＦＣドライブ

中　－　１０Ｋ　ＦＣドライブ

低　－　ＳＡＴＡドライブ

【０１６１】

別の実施形態では、システムには次のドライブ・タイプを有し得る。

高　－　２５Ｋ　ＦＣドライブ

低　－　１５Ｋ　ＦＣドライブ

【０１６２】

従って、１５Ｋ　ＦＣドライブは評価の低い区分として分類され、これに対して１５Ｋ　ＦＣドライブは評価の高い区分として分類される。

【０１６３】

ＳＡＴＡドライブがシステムに追加された場合、データ・プログレッションは、ディスクを時間的に再分類する。その結果として次の分類となる。

高　－　２５Ｋ　ＦＣドライブ

中　－　１５Ｋ　ＦＣドライブ

低　－　ＳＡＴＡドライブ

【０１６４】

データ・プログレッションは、ウォーターフォール（waterfall）・プログレッションを含むことができる。典型的に、ウォーターフォール・プログレッションはデータをより安価な資源へ移動させるが、その移動は、その資源が完全に使用されたときにのみ行う。ウォーターフォール・プログレッションは、最も高価なシステム資源の使用を効率的に最大化する。システムのコストの最小化も行う。安いディスクを最も低いプールに追加すると、ボトム（底部）において、より大きなプールが作成される。

【０１６５】

典型的なウォーターフォール・プログレッションは、ＲＡＩＤ１０空間を使用し、次いで、ＲＡＩＤ５空間などのような、ＲＡＩＤ空間の次のものを使用する。このことにより、ウォーターフォールが次のクラスのディスクのうちのＲＡＩＤ１０へ直接向かうように強制される。代替例として、データ・プログレッションは、図２４に示すような混合ＲＡＩＤウォーターフォール・プログレッションを含んでもよい。この代替のデータ・プログレッション方法により、ディスク空間およびパフォーマンスを最大化する問題が解決され、ストレージは同じディスク・クラスにおいて更に効率のよい形式へと変形できる。また、この代替方法では、ＲＡＩＤ１０およびＲＡＩＤ５が或るディスク・クラスの資源全体を共用するという要件をサポートする。これには、あるＲＡＩＤレベルがあるクラスのディスク用に使用する、ディスク空間の一定の割合をコンフィギュレーションすることが必要であることがある。従って、この代替のデータ・プログレッション方法では、高価なストレージの使用が最大化され、かつ、別のＲＡＩＤクラスが共存する余地を残す。

【０１６６】

また、混合ＲＡＩＤウォーターフォールでは、ストレージが限られているときのみ、ペ

ージをより安価なストレージへと移動させる。ディスク空間全体のうちの或るパーセンテージなどのような閾値により、特定のＲＡＩＤタイプのストレージの量が制限される。これにより、システムにおける最も高価なストレージの使用が最大化される。ストレージがその限度に近づくと、データ・プログレッションは、ページを、コストのより低いストレージへと自動的に移動させる。データ・プログレッションは、書き込みスパイクに対するバッファを提供することができる。

【０１６７】

上述のウォーターフォール方法では、一部の場合のように、ページをコストが最も低いストレージへと直ちに移動させ得るということが理解され、履歴およびアクセス不能のページをより安価なストレージ上へ適時に移動させる必要性があり得る。また、履歴ページをより安価なストレージへ直ちに移動させることもできる。

【０１６８】

図１８に、データ・プログレッションのプロセス１８００の流れ図を示す。データ・プログレッションは、システムのそれぞれのページを、そのアクセスのパターンおよびストレージのコストに関して絶えず検査して、移動させるべきデータ・ページがあるかどうか判定する。また、データ・プログレッションは、ストレージがその最大割り当て量に達しているかどうかを判定することもできる。

【０１６９】

データ・プログレッションのプロセスは、そのページがどのボリュームからもアクセス可能かどうかを判定する。プロセスは、履歴に付加されている各ボリュームについてのＰＩＴＣを検査して、そのページが参照されているかどうかを判定する。そのページがアクティブに使用中である場合には、そのページは昇格させるに又は降格を遅くするに望ましい。そのページがどのボリュームからもアクセス可能でない場合には、使用可能な最も低コストのストレージへ移動させられる。また、データ・プログレッションは、ＰＩＴＣが期限切れになる前に時間を計算に入れる。スナップショットで、ＰＩＴＣがまもなく期限切れになるようにスケジュールしている場合には、ページのプログレッションは行わない。ページ・プールがアグレッシブ・モードで動作中の場合には、ページのプログレションが行われ得る。

【０１７０】

データ・プログレッションの最近アクセス検出（recent access detection、最近されたアクセスの検出）では、ページの昇格動作から一塊の活動を取り除く必要がある。データ・プログレッションでは、読み出しと書き込みのアクセス追跡を分離している。これにより、データ・プログレッションは、アクセス可能であるＲＡＩＤ５装置上のデータを保持することが可能とされる。ウイルスのスキャンや報告などのような動作は、データを読み出すのみである。データ・プログレッションは、ストレージが残り少なくなっているときに、最近アクセス（recent access）の資格を変更する。これにより、データ・プログレッションは、ページの降格をよりアグレッシブに行えるようになる。また、これは、ストレージが残り少なくなっているときに、システムをボトム（底）の方から詰めていくのに役立つ。

【０１７１】

データ・プログレッションは、システム資源が少なくなるとデータ・ページをアグレッシブ（積極的）に移動させることができる。より多くのディスクまたはコンフィギュレーションの変更は、こうした場合のすべてでやはり必要である。データ・プログレッションは、時間のない状況においてシステムが動作し得る時間の量を延ばす。データ・プログレッションは、システムをできるだけ長く動作可能に保つように試みる。その時間とは、そのすべてのストレージ・クラスの空間がなくなるときである。

【０１７２】

ＲＡＩＤ１０空間が残り少なくなっており、使用可能なディスク空間全体が残り少なくなっている場合には、データ・プログレッションは、ＲＡＩＤ１０ディスク空間を取り上げて、より効率のよいＲＡＩＤ５へと移動させることができる。これにより、システムの

全体容量が、書き込み性能を犠牲にして、増加する。より多くのディスクはやはり必要である。特定のストレージ・クラスが完全に使用された場合、データ・プログレッションは、受け入れ可能でないページを借りてシステムを動作させ続けることができる。例えば、あるボリュームが、ＲＡＩＤ１０－ＦＣを、そのアクセス可能情報用に使用するようにコンフィギュレーションされている場合に、より多くのＲＡＩＤ１０－ＦＣ空間が使用可能になるまで、ＲＡＩＤ５－ＦＣまたはＲＡＩＤ１０－ＳＡＴＡからページを割り当てる。

【０１７３】

また、データ・プログレッションは、システムの知覚される容量を増加させるための圧縮をサポートする。圧縮が使用されるのは、アクセスされていない履歴ページに対して、またはリカバリ情報の記憶のためにである。圧縮は、ストレージ・コストの最下部近くの別のクラスのストレージとして現れる。

【０１７４】

図２５に示すように、ページ・プールは、本質的に、フリー・リストおよび装置情報を含む。ページ・プールは、複数のフリー・リスト、強化されたページ割り当て方式、およびフリー・リストの分類をサポートする必要がある。ページ・プールは、ストレージのクラスごとに別々のフリー・リストを維持する。この割り当て方式により、ページを多くのプールのうちの１つから割り当て、且つ、許可される最小または最大のクラスを設定する。フリー・リストの分類は、装置のコンフィギュレーションに由来する。それぞれのフリー・リストは、統計の収集および表示のためにそれ自体のカウンタを提供する。また、それぞれのフリー・リストは、ストレージ効率の統計の収集のためのＲＡＩＤ装置効率情報を提供する。

【０１７５】

一実施形態では、装置リストには、ストレージ・クラスのコストを追跡する更なる機能が必要であり得る。組合せにより、ストレージのクラスが決定される。これが起きるのは、ユーザが、コンフィギュレーションしたクラスの粒度がより荒いまたは細かいものであることを希望する場合である。

【０１７６】

図２６に、高性能のデータベースの一実施形態を示しており、ここでは、すべてのアクセス可能なデータは、最近アクセスされていない場合でも、２．５ＦＣドライブ上にある。アクセス不能な履歴データは、ＲＡＩＤ５ファイバ・チャネルへ移動される。

【０１７７】

図２７は、ＭＲＩ画像ボリュームの一実施形態を示しており、ここでは、アクセス可能なストレージは、この動的なボリューム用のＳＡＴＡ　ＲＡＩＤ１０およびＲＡＩＤ５である。画像が最近アクセスされていない場合、画像はＲＡＩＤ５へ移動される。その場合、新しい書き込みは、初めはＲＡＩＤ１０へ向かう。図１９は、圧縮されたページ・レイアウトの一実施形態を示す。データ・プログレッションは、圧縮を、固定サイズのデータ・ページを更に細かく割り当てることによって、実装する。更に細かい割り当て（sub-allocation、サブ・アロケーション）の情報は、ページの空きの部分、およびページの割り当て済みの部分の場所を追跡する。データ・プログレッションは、圧縮の効率を予測することはできないが、そのサブ・アロケーション範囲内で可変サイズのページを取り扱うことができる。

【０１７８】

圧縮されたページは、ＣＰＵの性能に著しく影響することがある。書き込みアクセスの場合、圧縮されたページは、ページ全体の圧縮解除および再圧縮が必要となるはずである。従って、アクティブにアクセスされているページは、圧縮されず、非圧縮の状態に戻される。書き込みは、ストレージが極度に限られた状況で必要となり得る。

【０１７９】

ＰＩＴＣ再マッピング・テーブルは、サブ・アロケーション情報を指し、圧縮されているページを示すようにマークされる。圧縮されたページにアクセスするには、非圧縮のページよりも高いＩ／Ｏカウントが必要になり得る。アクセスには、実際のデータの場所を

取り出すために、サブ・アロケーション情報の読み出しが必要になる。圧縮されたデータは、ディスクから読み出し、プロセッサで圧縮解除することができる。

【０１８０】

データ・プログレッションでは、圧縮は、ページ全体のうちの一部を圧縮解除できるようにすることが必要とされ得る。これにより、データ・プログレッションの読み出しアクセスは、そのページの小さな部分のみの圧縮解除を行える。読み出しキャッシュの先読み機能は、圧縮の遅延に対して役立ち得る。一つの圧縮解除で、幾つものサーバＩ／Ｏを取り扱うことができる。データ・プログレッションは、圧縮の候補として適さないページにマークし、それによって、或るページの圧縮を続けて試みないようにする。

【０１８１】

図２０に、本発明の諸原理に従った、高レベルのディスク・ドライブ・システムにおけるデータ・プログレッションの一実施形態を示す。データ・プログレッションは、ボリュームの外面的な振る舞いやデータ・パスの動作を変更しない。データ・プログレッションは、ページ・プールへの変更を必要とし得る。ページ・プールは、本質的に、フリー・リストおよび装置情報を含む。ページ・プールは、複数のフリー・リスト、強化されたページ割り当て方式、およびフリー・リストの分類をサポートする必要がある。ページ・プールは、ストレージのそれぞれのクラスに対して個々のフリー・リストを維持する。割り当て方式は、許可される最小および最大のクラスを設定しつつ、ページが多くのプールのうちの１つから割り当てられるようにする。フリー・リストの分類は、装置のコンフィギュレーションに由来することができる。それぞれのフリー・リストは、統計の収集および表示のためにそれ自体のカウンタを提供する。また、それぞれのフリー・リストは、ストレージの効率の統計の収集のためにＲＡＩＤ装置効率情報を提供する。

【０１８２】

ＰＩＴＣは、移動の候補を識別し、その移動のときには、アクセス可能なページへのＩ／Ｏを妨げる。データ・プログレッションは、候補に関してＰＩＴＣを継続的に検査する。ページのアクセス可能性は、サーバＩ／Ｏ、新しいスナップショット・ページ更新、およびビュー・ボリュームの作成／削除に起因して、継続的に変化する。また、データ・プログレッションは、ボリュームのコンフィギュレーションの変更を継続的に検査し、ページのクラスおよびカウントの現在のリストをまとめあげる。これにより、データ・プログレッションは、そのまとめ（一覧）を評価し、移動するページがあるかを判定できる。

【０１８３】

それぞれのＰＩＴＣは、ストレージの各クラスに使用されるページの数に関するカウンタを提示する。データ・プログレッションは、この情報を使用して、閾値に達したときにページを移動させるための良い候補になるＰＩＴＣを識別する。

【０１８４】

ＲＡＩＤは、ディスクのコストに基づいて、１組のディスクから装置を割り当てる。また、ＲＡＩＤは、装置または潜在的な装置の効率を取り出すＡＰＩを提供する。また、これは、書き込み動作に必要なＩ／Ｏの数についての情報を返す必要がある。また、データ・プログレッションは、また、サード・パーティのＲＡＩＤコントローラをデータ・プログレッションの一部として使用するために、ＲＡＩＤ　ＮＵＬＬを必要とし得る。ＲＡＩＤ　ＮＵＬＬは、ディスク全体を消費し得、単にパス・スルー・レイヤとして働き得る。

【０１８５】

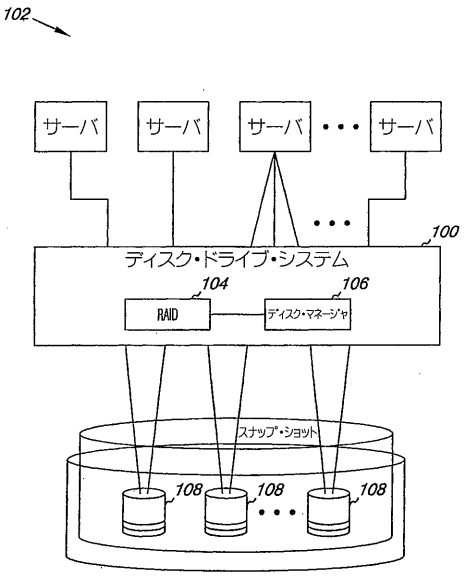
また、ディスク・マネージャは、ディスク分類を自動的に決定し保存することができる。ディスク分類を自動的に決定するには、ＳＣＳＩイニシエータへの変更が必要となり得る。

【０１８６】

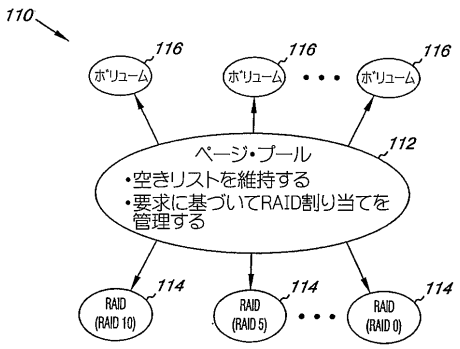
上述の説明および図面から、ここに示し説明した特定の諸実施形態は、もっぱら例示を目的としており、本発明の範囲を限定するものではないことが当業者には理解されよう。本発明を他の特定の形式で実施することは、その趣旨または本質的な諸特徴から逸脱することなく行えることが当業者には理解されよう。特定の諸実施形態の詳細への言及は、本

発明の範囲を限定するものではない。

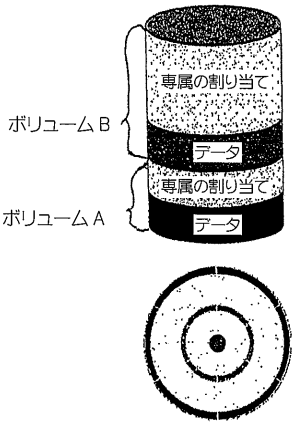
【図１】



【図２】

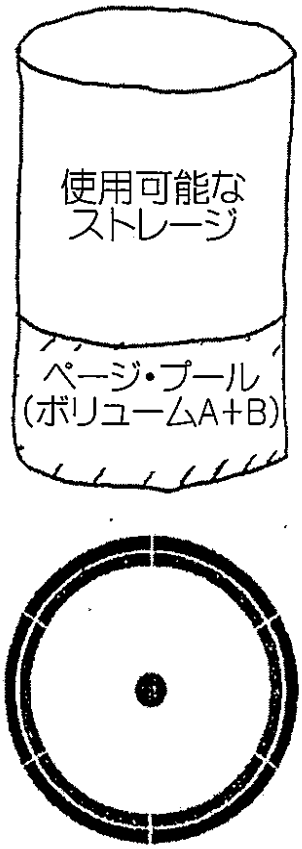


【図２Ａ】

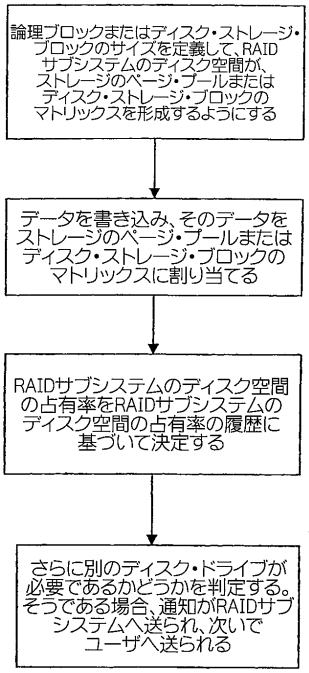


（従来技術）

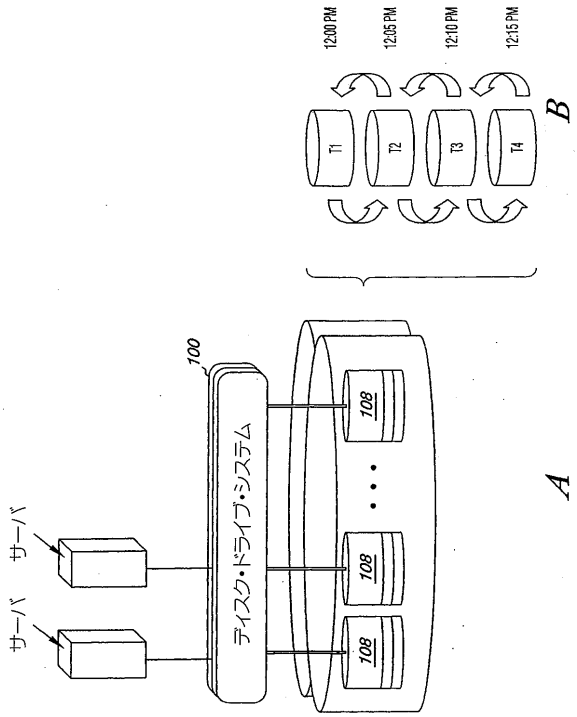
【図２Ｂ】


使用可能な
ストレージ
ページ・プール
(ボリュームA+B)

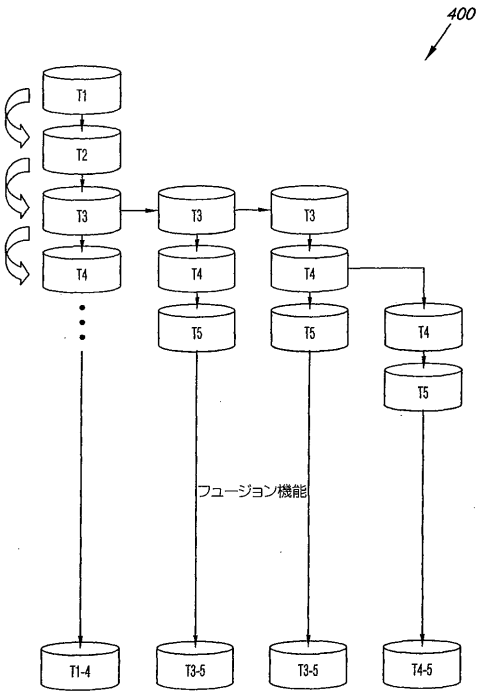
【図２Ｃ】


論理ブロックまたはディスク・ストレージ・ブロックのサイズを定義して、RAIDサブシステムのディスク空間が、ストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスを形成するようにする
データを書き込み、そのデータをストレージのページ・プールまたはディスク・ストレージ・ブロックのマトリックスに割り当てる
RAIDサブシステムのディスク空間の占有率をRAIDサブシステムのディスク空間の占有率の履歴に基づいて決定する
さらに別のディスク・ドライブが必要であるかどうかを判定する。そうである場合、通知がRAIDサブシステムへ送られ、次いでユーザへ送られる

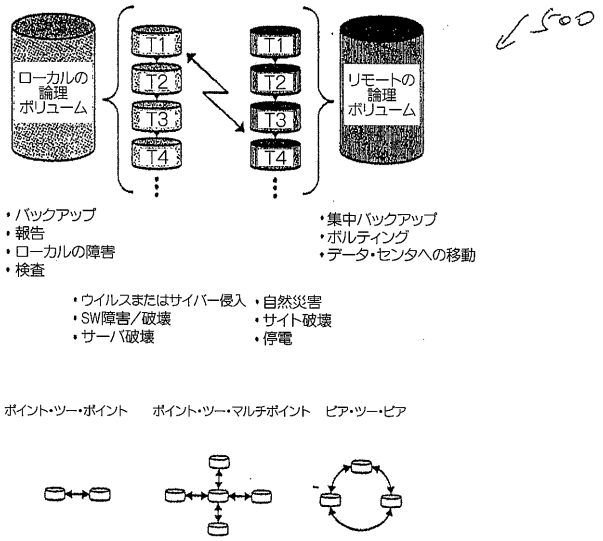
【図３】


12:00 PM
12:05 PM
12:10 PM
12:15 PM
T1
T2
T3
T4
B
100
ディスク・ドライブ・システム
108
サーバ
サーバ
A

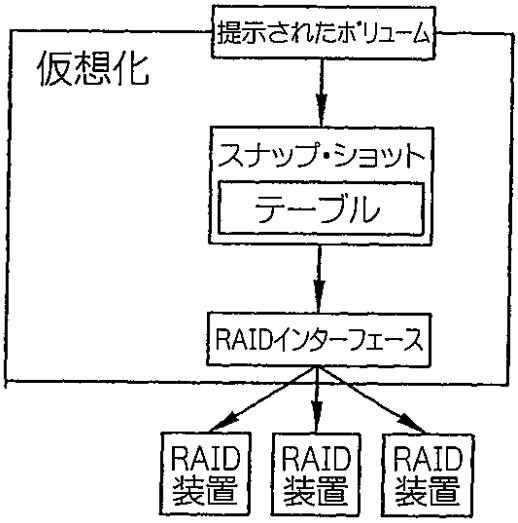
【図４】


400
T1
T2
T3
T4
T5
フュージョン機能
T1-4
T3-5
T3-5
T4-5

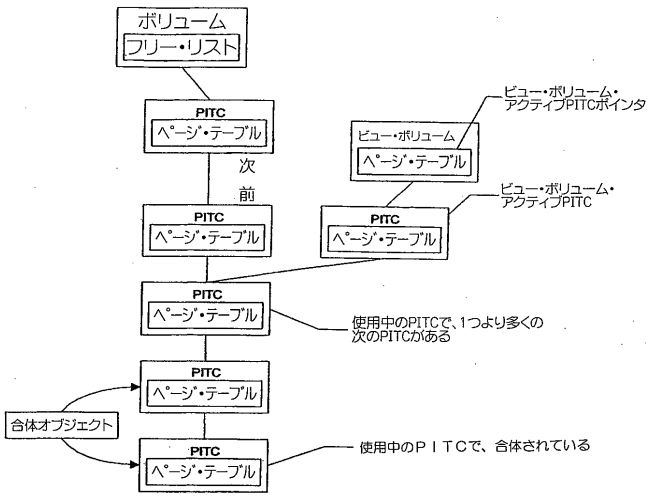
【図５】


500
ローカルの論理ボリューム
リモートの論理ボリューム
T1
T2
T3
T4
・バックアップ
・報告
・ローカルの障害
・検査
・集中バックアップ
・ボルティング
・データ・センタへの移動
・ウイルスまたはサイバー侵入
・SW障害／破壊
・サーバ破壊
・自然災害
・サイト破壊
・停電
ポイント・ツー・ポイント
ポイント・ツー・マルチポイント
ピア・ツー・ピア

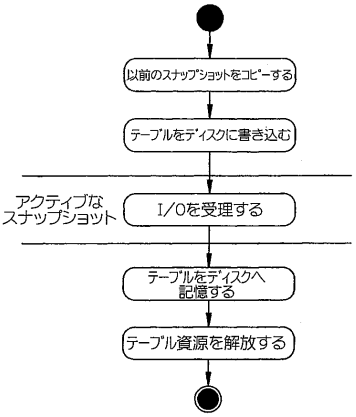
【図６】


仮想化
提示されたボリューム
スナップ・ショット
テーブル
RAIDインターフェース
RAID装置
RAID装置
RAID装置

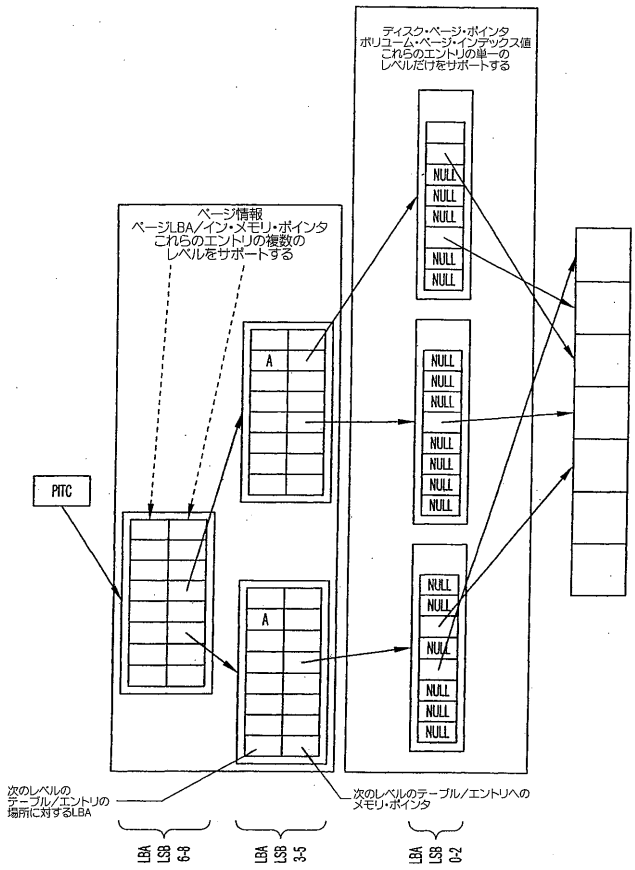
【図７】


ボリューム
フリー・リスト
PITC
ページ・テーブル
次
前
ビュー・ボリューム・アクティブPITCポインタ
ビュー・ボリューム
ビュー・ボリューム・アクティブPITC
使用中のPITCで、1つより多くの次のPITCがある
合体オブジェクト
使用中のＰＩＴＣで、合体されている

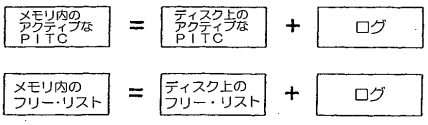
【図８】


以前のスナップショットをコピーする
テーブルをディスクに書き込む
アクティブなスナップショット
I/Oを受理する
テーブルをディスクへ記憶する
テーブル資源を解放する

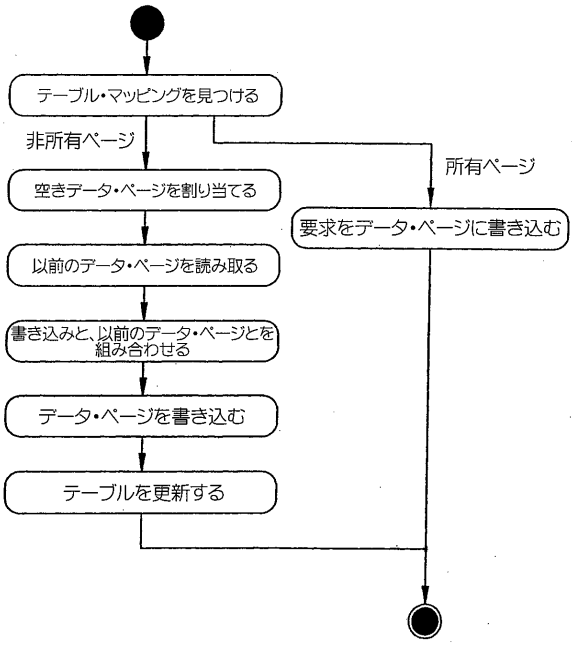
【図９】



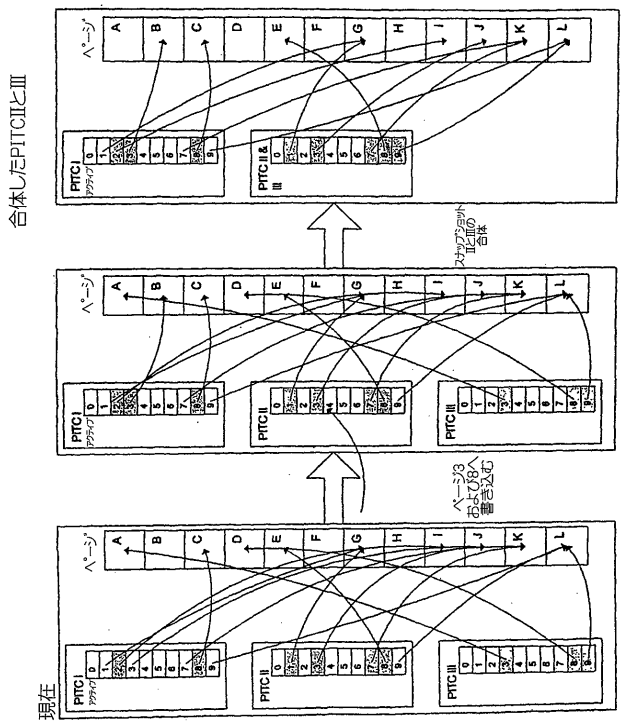
【図１０】



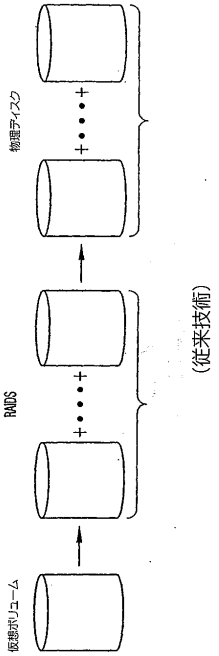
【図１１】



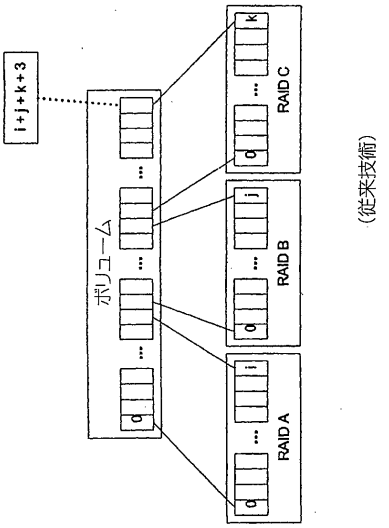
【図１２】



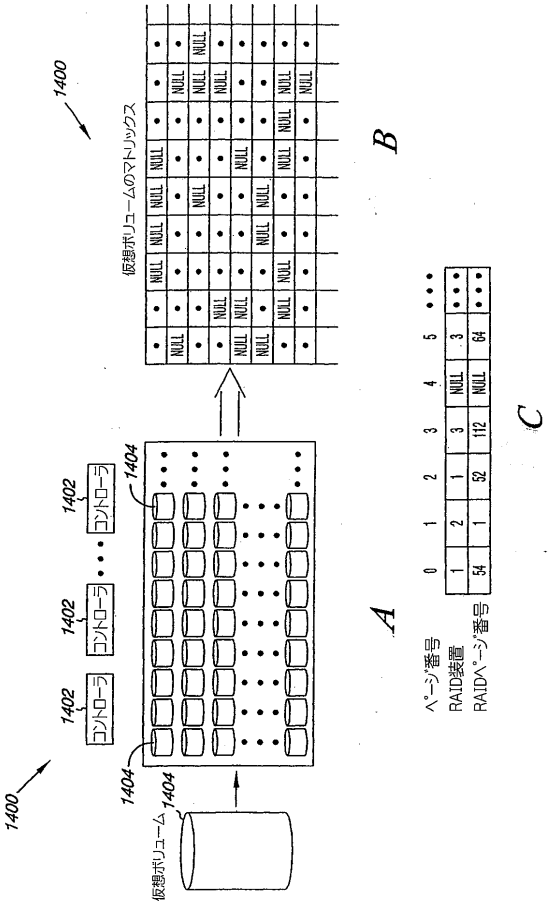
【図１３Ａ】



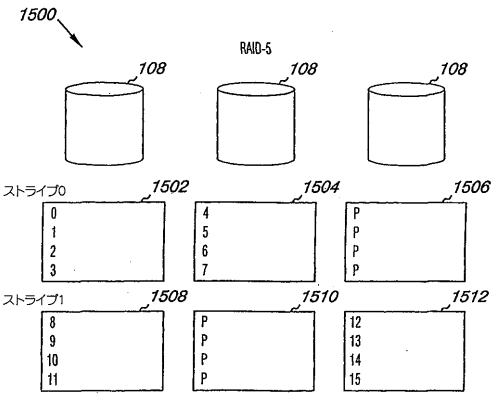
【図１３Ｂ】



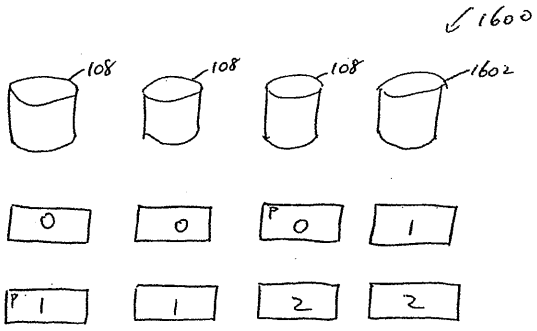
【図１４】



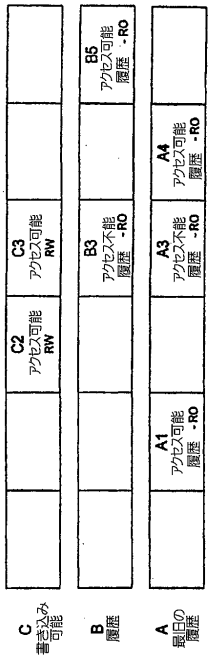
【図１５】



【図１６】



【図１７】



【図１８】

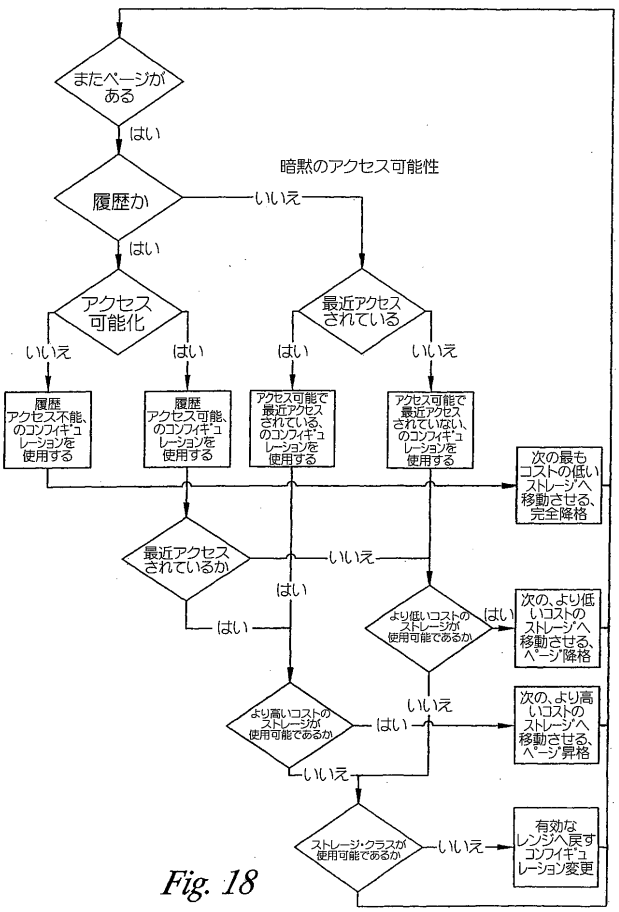
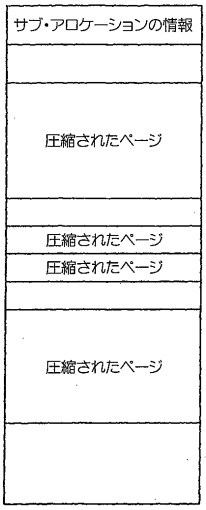
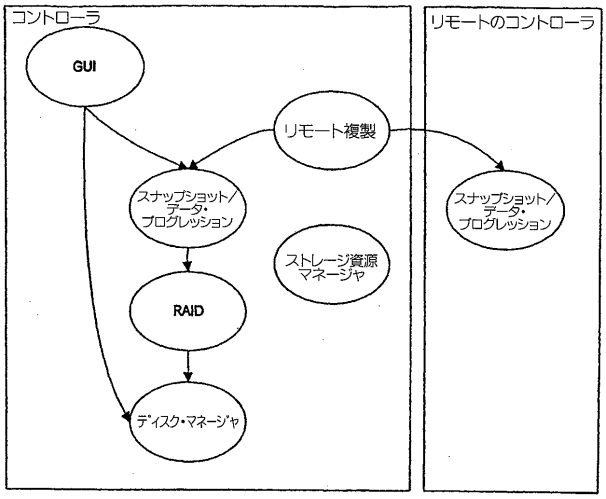


Fig. 18

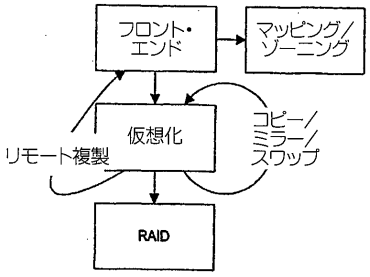
【図１９】



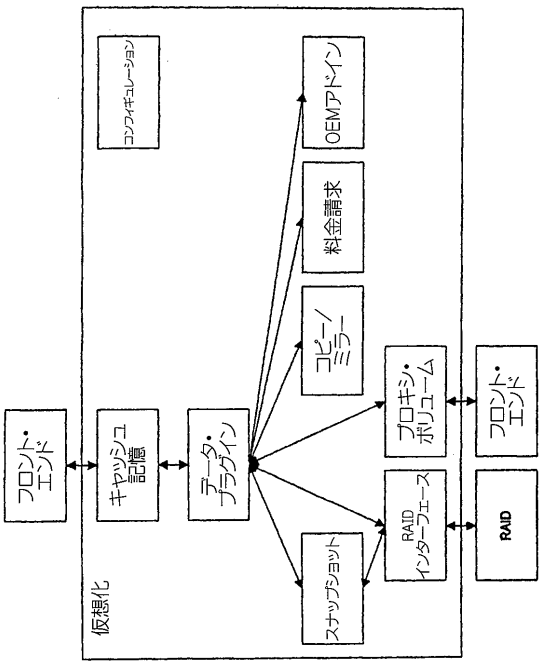
【図２０】



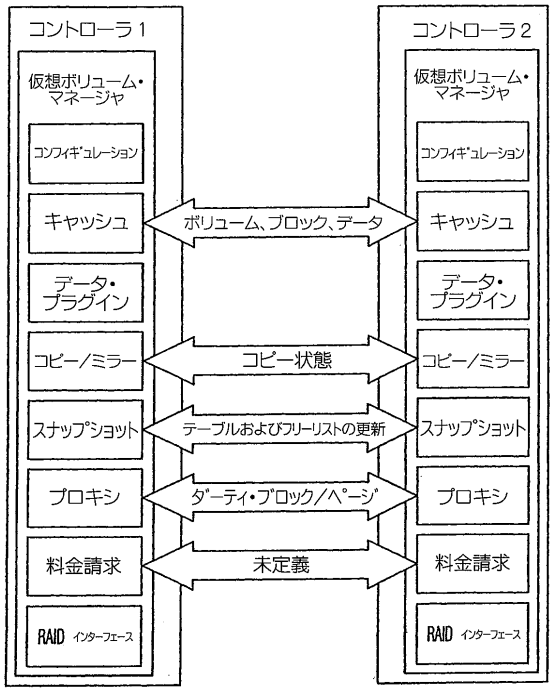
【図２１】



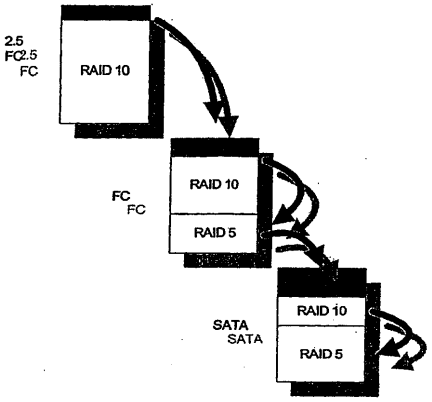
【図２２】



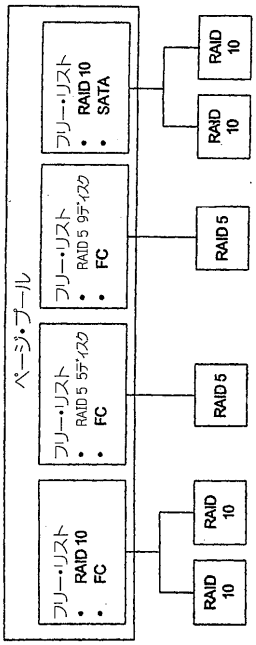
【図２３】



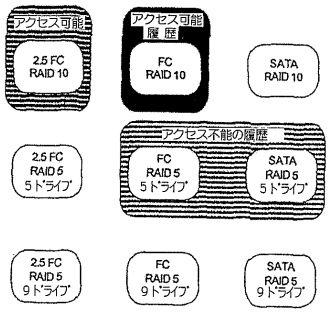
【図２４】



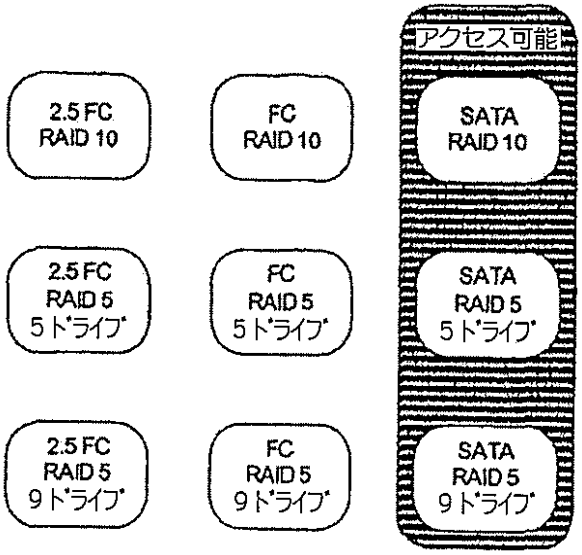
【図２５】



【図２６】



【図２７】



【手続補正書】
【提出日】平成22年9月22日(2010.9.22)
【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】
データ・インスタント・リプレイの方法であって、
複数のデータ・ストレージ・デバイスからデータ空間ブロックの抽象的なボリュームを作るステップと、
前記抽象的なボリュームのデータのスナップショットを生成するステップと、
前記スナップショットのアドレス・インデックスを記憶するステップと、
記憶された前記アドレス・インデックスを介して前記スナップショットにアクセスするステップと
を備える方法。
【請求項２】
請求項１に記載の方法であって、
前記抽象的なボリュームの前記データ空間ブロックを割り当てるステップと、
割り当てられた前記データ空間ブロックへデータを書くステップと
を更に備える方法。
【請求項３】
請求項１または２に記載の方法であって、前記スナップショットは所定の時間間隔において生成される、方法。
【請求項４】
請求項１ないし３の何れかに記載の方法であって、前記スナップショットはアプリケーションにより開始される、方法。
【請求項５】
請求項１ないし４の何れかに記載の方法であって、前記スナップショットを生成するステップは、書き込み動作を前記抽象的なボリュームに記録するステップを備える、方法。
【請求項６】
請求項１ないし５の何れかに記載の方法であって、前記スナップショットを生成するステップは、前記抽象的なボリュームに関しての変化を追跡するステップを備える、方法。
【請求項７】
請求項６に記載の方法であって、前記変化を追跡するステップは、少なくとも、前記抽象的なボリュームへのデルタ書き込みの部分的リストを維持するステップを備える、方法。
【請求項８】
請求項７に記載の方法であって、前記変化を追跡するステップは、少なくとも維持される前記部分的リストは、前のスナップショットの生成以降の、前記抽象的なボリュームの１つのデータ・アドレスへの最も新しい書き込みを含む、方法。
【請求項９】
請求項１ないし８の何れかに記載の方法であって、以前の時点での前記抽象的なボリュームのコンテンツのビューを提供する前記スナップショットに基づいて一時的なボリュームを作るステップを更に備える方法。
【請求項１０】
請求項９に記載の方法であって、前記スナップショットおよび前記抽象的なボリュームにおける基礎となるデータを変更せずに、前記一時的なボリュームを変更するステップを更に備える方法。

【請求項１１】
請求項１０に記載の方法であって、前記一時的なボリュームは検査に使用され、前記検査は、前記抽象的なボリュームにおける基礎となるデータを変えずに、前記以前の時点にあったデータに対して行われ得る、方法。
【請求項１２】
請求項１０または１１に記載の方法であって、前記一時的なボリュームは訓練に使用され、前記訓練は、前記抽象的なボリュームにおける基礎となるデータを変えずに、前記以前の時点にあったデータに対して行われ得る、方法。
【請求項１３】
請求項１０ないし１２の何れかに記載の方法であって、前記一時的なボリュームは、データを複製することにより該データのバックアップに用いられる、方法。
【請求項１４】
請求項１０ないし１３の何れかに記載の方法であって、前記一時的なボリュームは、データのリカバリに用いられる、方法。
【請求項１５】
データを再生可能なデスク・ドライブ・システムであって、
データ空間ブロックの抽象化したものを有するデータ・ストレージ・システムと、
少なくとも１つのデータ・ストレージ・システム・コントローラを有するデータ・マネージャであって、前記データ・ストレージ・システム・コントローラが、自動的に、前記データ空間ブロックの抽象化したもののスナップショットを生成し、前記スナップショットのアドレス・インデックスを記憶し、記憶された前記アドレス・インデックスを介して前記スナップショットにアクセスするものである、データ・マネージャと
を備えるデスク・ドライブ・システム。
【請求項１６】
請求項１５に記載のデスク・ドライブ・システムであって、前記システム・コントローラは、自動的に、前記スナップショットを所定の時間間隔において生成前する、デスク・ドライブ・システム。
【請求項１７】
請求項１５または１６に記載のデスク・ドライブ・システムであって、前記システム・コントローラは、自動的に、アプリケーションの制御に基づいて前記スナップショットを生成する、デスク・ドライブ・システム。
【請求項１８】
請求項１５ないし１７の何れかに記載のデスク・ドライブ・システムであって、前記システム・コントローラは、前記データ空間ブロックの抽象化したものに対する書き込み動作を記憶する、デスク・ドライブ・システム。
【請求項１９】
請求項１５ないし１８の何れかに記載のデスク・ドライブ・システムであって、前記システム・コントローラは、前記データ空間ブロックの抽象化したものに関しての変化を追跡する、デスク・ドライブ・システム。
【請求項２０】
請求項１９の何れかに記載のデスク・ドライブ・システムであって、前記システム・コントローラは、少なくとも、前記データ空間ブロックの抽象化したものに対するデルタ書き込みの部分的リストを維持する、デスク・ドライブ・システム。
【請求項２１】
請求項２０の何れかに記載のデスク・ドライブ・システムであって、少なくとも維持される前記部分的リストは、前のスナップショットの生成以降の、前記データ空間ブロックの抽象化したものの１つのデータ・アドレスへの最も新しい書き込みを含む、デスク・ドライブ・システム。
【請求項２２】
請求項１５ないし２１の何れかに記載のデスク・ドライブ・システムであって、前記シ

ステム・コントローラは、以前の時点での前記データ空間ブロックの抽象化したもののコンテンツのビューを提供する前記スナップショットに基づいて一時的なボリュームを作る、デスク・ドライブ・システム。

【請求項２３】

請求項２２に記載のデスク・ドライブ・システムであって、前記システム・コントローラは、前記一時的なボリュームをデータのリカバリに用いる、デスク・ドライブ・システム。

フロントページの続き

(72)発明者　ソラン，フィリップ・イー
アメリカ合衆国ミネソタ州５５３４７，エデン・プレイリー，エイムズバリー・レイン　９５０１
(72)発明者　ガイダー，ジョン・ピー
アメリカ合衆国ミネソタ州５５１２７，ノース・オークス，キャットバード・レイン　７
(72)発明者　アスズマン，ローレンス・イー
アメリカ合衆国ミネソタ州５５３７２，プライアー・レイク，ショア・トレイル・ノースイースト　５４４５
(72)発明者　クレム，マイケル・ジェイ
アメリカ合衆国ミネソタ州５５３０５，ミネトンカ，リヴェンデル・レイン　２３０１

Ｆターム(参考) 5B065 BA06　CA30　CC03　EA02