(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2013-161476
(P2013-161476A)
(43)公開日　平成25年8月19日(2013.8.19)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 3/06 (2006.01)		G06F	3/06	306Z
		G06F	3/06	540
		G06F	3/06	301Z
		G06F	3/06	305C
		G06F	3/06	304R

審査請求 未請求　請求項の数 20　ＯＬ　（全 21 頁）

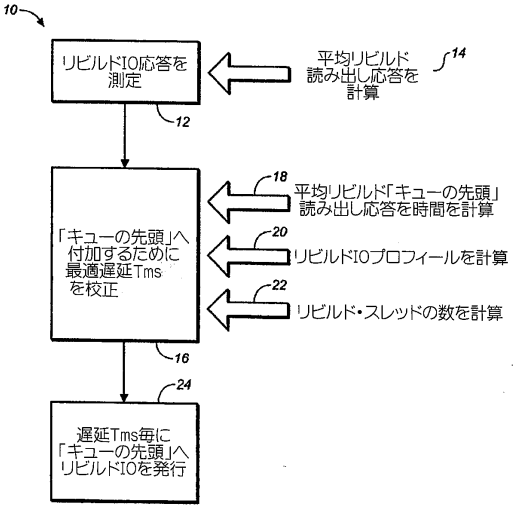
(21)出願番号　特願2013-2710(P2013-2710)
(22)出願日　平成25年1月10日(2013.1.10)
(31)優先権主張番号　13/368,725
(32)優先日　平成24年2月8日(2012.2.8)
(33)優先権主張国　米国(US)

(71)出願人　591007686
エルエスアイ　コーポレーション
アメリカ合衆国カリフォルニア州９５０３５，ミルピタス，バーバー・レーン　１６２１
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100096013
弁理士　富田　博行
(74)代理人　100092967
弁理士　星野　修
(74)代理人　100096068
弁理士　大塚　住江

最終頁に続く

(54)【発明の名称】ＲＡＩＤの再構築を改善するシステム及び方法

(57)【要約】
【課題】ＲＡＩＤシステムにおけるディスクのリビルドの処理を改善する。
【解決手段】ホストＩＯの処理が優先して行われる処理において、ストレージ・システムのドライブのリビルドＩＯ応答を測定して、応答の平均時間を計算し（ステップ１２）、「キューの先頭」に付加すべき遅延時間である最適遅延時間を校正し（ステップ１６）、該最適遅延時間毎に、「キューの先頭」に対してリビルドＩＯ要求を発行する（ステップ２４）。ドライブのキューの先頭へ向けてリビルドＩＯ要求を発行することにより、リビルドＩＯの処理が優先的に行われる。遅延時間は、ディスク・ヘッドの応答時間、リビルドに割り当てられる時間、リビルド・スレッドの数、ドライブの挙動、リビルドＩＯのプロフィール、作業負荷及び局所性を含むホストＩＯのプロファイル、一連のイベントのタイムラインなどに基づいて計算される。
【選択図】図１



【特許請求の範囲】
【請求項１】
複数のドライブを有し、ホストＩＯ条件のもとでのＲＡＩＤのリビルド処理を改善するストレージ・システムであって、
　前記ストレージ・システムの１つのドライブのリビルドＩＯ応答を測定するファームウェアと、
　前記１つのドライブのキューであって、リビルドＩＯ要求及びホストＩＯ要求を含むキューと、
　前記キューに対するリビルドＩＯの発行を遅延させるファームウェアと
を備えることを特徴とするストレージ・システム。
【請求項２】
請求項１に記載のストレージ・システムにおいて、該システムはさらに、
　前記リビルドＩＯの発行の遅延を較正するファームウェア
を備えるストレージ・システム。
【請求項３】
請求項２に記載のストレージ・システムにおいて、前記遅延を較正するファームウェアは、前記キューからの平均リビルド応答時間、全時間のうちのリビルドに割り当てられる時間、及びリビルド・スレッドの数のパラメータに従って、遅延を較正するよう構成されていることを特徴とするストレージ・システム。
【請求項４】
請求項２に記載のストレージ・システムにおいて、該システムはさらに、前記ファームウェアがどれだけ速くリビルドを行うかを示すリビルド・レートを有するファームウェアを備え、前記リビルド・レートは、最大値及び最小値を有し、前記最小値の場合は、前記最大値の場合と比べて、ホストＩＯがリビルドＩＯよりも優先されることを特徴とするストレージ・システム。
【請求項５】
請求項４に記載のストレージ・システムにおいて、前記リビルド・レートを有するファームウェアにおける前記リビルド・レートはコンフィギュレーション可能であり、前記リビルド・レートを有するファームウェアが、前記最小値から前記最大値までの範囲で幾つかの所定の値をとることができるリビルド・レートを有することができるよう構成されていることを特徴とするストレージ・システム。
【請求項６】
請求項２に記載のストレージ・システムにおいて、前記遅延を較正するファームウェアは、読み出し応答時間、リビルドに割り当てられる時間、ユーザ選択のリビルドＩＯプロファイル、コンフィギュレーション可能なリビルドＩＯプロファイル、及びリビルド・スレッドの数を含むグループから選択されたパラメータに従って、前記遅延を較正するよう構成されていることを特徴とするストレージ・システム。
【請求項７】
請求項６に記載のストレージ・システムにおいて、該システムはさらに、周期的に前記キューの先頭へ向けてリビルドＩＯを発行するファームウェアを備え、該ハードウェアにより行われる処理は、１つの周期を構成する複数のステージを有し、前記複数のステージは、
　リビルドＩＯへ遅延を付加せずにリビルドＩＯをキューへ向けて発行し、リビルドを行う第１のステージと、
　リビルド・レートと応答時間とのうちの少なくとも一方が、それぞれに対応して設定された所定の値よりも大きい場合に実行される第２のステージであって、前記遅延の時間であるＴミリ秒を計算する第２のステージと、
　前記Ｔミリ秒毎に前記キューの先頭へ向けてリビルドＩＯを発行する第３のステージとからなることを特徴とするストレージ・システム。
【請求項８】

請求項７に記載のストレージ・システムであって、
　前記第２のステージにおいて、前記遅延の時間であるＴミリ秒は、下記の式
Ｔ＝Ｃ／ｎＲ－ｒｅｓｐ
により計算されるものであり、
　ただし、上記の式において、
　ｎＲ＝ｒＩＯＰＳ／（リビルド・スレッドの数）、
　ｒＩＯＰＳ＝ｒｔ／ｒｅｓｐ
であり、
　前記リビルド・スレッドの数は、前記ＲＡＩＤシステムにおけるリビルド・スレッドの数であり、
　前記ｒｔは、全時間Ｃのうちのリビルドに割り当てられる時間であり、
　前記ｒｅｓｐは、キューの先頭での平均「キューの先頭」リビルド読み出し応答時間であり、
　前記Ｃは時間の定数であり、前記全時間である
ことを特徴とするストレージ・システム。
【請求項９】
請求項７に記載のストレージ・システムにおいて、前記ファームウェアは、ＲＡＩＤ１、ＲＡＩＤ２、ＲＡＩＤ３、ＲＡＩＤ４、ＲＡＩＤ５、ＲＡＩＤ６、ＲＡＩＤ０を含むハイブリッドＲＡＩＤレベル１～６、コピーバック・オペレーションを行うストレージ・システム、及び消失訂正符号を用いるストレージを含むグループから選択されるストレージ・システムにおいて動作するよう構成されていることを特徴とするストレージ・システム。
【請求項１０】
ストレージ・システムの複数のドライブに対してのＲＡＩＤリビルドの方法であって、
　ホストＩＯ条件のもとでリビルドを行うＲＡＩＤシステムにおけるリビルドＩＯ応答を測定するステップと、
　前記ＲＡＩＤシステムのドライブのキューの先頭へ向けてのリビルドＩＯの発行を遅延させる遅延時間を計算するステップと、
　前記キューの先頭へ向けてのリビルドＩＯを発行するステップであって、前記遅延時間により前記リビルドＩＯの発行を遅延させるステップと
を含んでいることを特徴とする方法。
【請求項１１】
請求項１０に記載の方法において、該方法はさらに、
　読み出し応答時間、リビルドに割り当てられる時間、ユーザ選択のリビルドＩＯプロファイル、コンフィギュレーション可能なリビルドＩＯプロファイル、及びリビルド・スレッドの数を含むグループから選択されたパラメータに従って、前記遅延時間を計算するステップ
を含んでいることを特徴とする方法。
【請求項１２】
請求項１１に記載の方法において、該方法はさらに、
　前記リビルドに割り当てられる遅延時間を、最小リビルド・レートから最大リビルド・レートまでの間から選択するステップ
を含んでいることを特徴とする方法。
【請求項１３】
請求項１２に記載の方法において、該方法はさらに、
　前記リビルドに割り当てられる時間を、コンフィギュレーション可能なリビルド・プロフィールから選択するステップ
を備え、前記リビルド・プロフィールにおいてリビルドに割り当てられる前記時間の選択肢は、最小リビルド・レートから最大リビルド・レートまでの間の複数のリビルド・レートを含むことを特徴とする方法。
【請求項１４】

請求項１３に記載の方法において、該方法はさらに、周期的に前記キューの先頭へ向けてリビルドＩＯを発行するステップを含み、該ステップは、
リビルドＩＯへ遅延を付加せずにリビルドＩＯをキューへ向けて発行し、ＲＡＩＤシステムのリビルドを行う第１のステージと、
リビルド・レートと応答時間とのうちの少なくとも一方が、それぞれに対応して設定された所定の値よりも高い場合に行われる第２のステージであって、前記遅延時間であるＴミリ秒を計算する第２のステージと、
前記Ｔミリ秒毎に前記キューへ向けてリビルドＩＯを発行する第３のステージと
を含むことを特徴とする方法。
【請求項１５】
請求項１０に記載の方法において、前記遅延時間は、読み出し応答時間、リビルドに割り当てられる時間、ユーザ選択のリビルドＩＯプロファイル、コンフィギュレーション可能なリビルドＩＯプロフィール、及びリビルド・スレッドの数を含むパラメータに従って計算されることを特徴とする方法。
【請求項１６】
請求項１５に記載の方法において、前記リビルドＩＯ応答を測定するステップと、前記キューに関しての遅延時間を計算するステップと、前記キューに対してリビルドＩＯを発行するステップとは、前記ＲＡＩＤシステムにおいて周期的に行われることを特徴とする方法。
【請求項１７】
請求項１６に記載の方法において、周期的に行われるステップは、
リビルドＩＯをドライブのキューの先頭へ向けて発行せずに、リビルドＩＯをキューへ向けて発行し、ＲＡＩＤシステムのリビルドを行う第１のステージと、
リビルド・レートと応答時間とのうちの少なくとも一方が、それぞれに対応して設定された所定の値よりも高い場合に行われる第２のステージであって、遅延時間であるＴミリ秒を計算する第２のステージと、
前記Ｔミリ秒毎に前記ドライブのキューの先頭へ向けてリビルドＩＯを発行する第３のステージと
を含むことを特徴とする方法。
【請求項１８】
請求項１７に記載の方法において、該方法は、ＲＡＩＤ１、ＲＡＩＤ２、ＲＡＩＤ３、ＲＡＩＤ４、ＲＡＩＤ５、ＲＡＩＤ６、ＲＡＩＤ０を含むハイブリッドＲＡＩＤレベル１～６、コピーバック・オペレーションを行うストレージ・システム、消失訂正符号を用いるストレージ、ＩＯを物理デバイスへ向けて発行するオペレーションを行うストレージ・システム、メディア・スキャンを行うストレージ・システム、整合性の検査を行うストレージ・システム、初期設定を行うストレージ・システム、及びフォーマッティングを行うストレージ・システムを含むグループから選択されるストレージ・システムにおいて実行されることを特徴とする方法。
【請求項１９】
ホストＩＯ条件のもとでのＲＡＩＤのリビルド処理のための装置であって、
ＲＡＩＤストレージ・システムにおけるリビルドＩＯ応答を測定する手段と、
前記ＲＡＩＤストレージ・システムに対するリビルドＩＯ要求及びホストＩＯ要求を記憶するためのキューを形成する手段と、
前記キューの先頭へ向けての前記リビルドＩＯ要求の発行を遅延させるためのミリ秒単位の遅延時間を計算する手段と、
前記キューの先頭へ向けてのリビルドＩＯを発行する手段であって、前記遅延時間だけ前記リビルドＩＯの実行を遅延させる手段と、
前記ＲＡＩＤストレージ・システムに対するリビルド・レートを設定する手段と
を備え、
前記ＲＡＩＤストレージ・システムの、前記リビルドＩＯ応答を測定する手段と、前記

遅延時間を計算する手段と、前記リビルドＩＯを発行する手段とは、周期的に動作するものであり、
前記リビルドＩＯ応答を測定する手段は、第１の時間間隔において、従来の様式で動作しているときに平均リビルド応答時間を計算するものであり、この計算は、前記ＲＡＩＤストレージ・システムのリビルド・レートと応答時間とのうちの少なくとも一方が、それぞれに対応して設定された所定の値よりも高い値となる時まで行われるものであり、
前記遅延時間を計算する手段は、第２の時間間隔において、前記ＲＡＩＤストレージ・システムが、前記ドライブのキューの先頭へ向けて発行されるリビルドＩＯ要求に関して発する前記遅延の命令を用いて動作しているときの平均リビルド応答時間と、前記リビルド・レートを設定する手段から得られるリビルド・レートと、前記ＲＡＩＤストレージ・システムにおけるスレッドの数とを用いて計算を行うことにより、最適な遅延時間を求めるものであり、
前記リビルドＩＯを発行する手段は、第３の時間間隔において、前記第２の時間間隔において計算された前記最適な遅延時間だけ、前記ドライブのキューの先頭へ向けてのリビルドＩＯ要求の発行を遅延させる
よう構成されていることを特徴とする装置。
【請求項２０】
請求項１９に記載の装置であって、前記第２の時間間隔において遅延時間を計算する手段は、前記最適な遅延時間であるＴを、下記の式
Ｔ＝Ｃ／ｎＲ－ｒｅｓｐ
により計算するものであり、上記の式において、
ｎＲ＝ｒＩＯＰＳ／（リビルド・スレッドの数）、
ｒＩＯＰＳ＝ｒｔ／ｒｅｓｐ
であり、
前記リビルド・スレッドの数は、前記ＲＡＩＤストレージ・システムにおけるリビルド・スレッドの数であり、
前記ｒｔは、全時間Ｃのうちのリビルドに割り当てられる時間であり、
前記ｒｅｓｐは、キューの先頭での平均「キューの先頭」リビルド読み出し応答時間であり、
前記Ｃは時間の定数であり、前記全時間である
ことを特徴とする装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ＲＡＩＤ（Redundant Array of Independent Disks）の分野における再構築（リビルド：rebuild）に関連する。
【背景技術】
【０００２】
本発明は、ＲＡＩＤに関し、ＲＡＩＤは、特にネットワークにおいて、複数のディスク・ドライブ・コンポーネントを仮想化形態の論理ユニットへと組み合わせる記憶技術であり、主に、ディスクの障害により発生するエラーを低減するためのものである。データは、複数のブロックへと分割されてシーケンシャルに幾つかのディスクへ記憶され、この様式はデータ・ストライピングとして知られている。パリティ・ブロックは、通常、エラー検査のための手段、及びディスク・ドライブの１つに障害が生じた際のデータの再構築のための手段を形成し、パリティ冗長性を形成する。
【０００３】
ＲＡＩＤは、適切に構成された場合には幾つかの利点を有する。ＲＡＩＤの利点として、冗長性を通じてデータ・セキュリティが高くなること（セーブＲＡＩＤ０コンフィギュレーション（save RAID 0 configuration））、障害を許容すること、データへのアクセスが改善されること（データ・アベラビリティ）、大量の隣接したディスク空間を作るた

めの統合された容量が増加すること、及び性能が改善されること等をあげることができる。ＲＡＩＤに関するコストには、より高価なハードウェアと、必要となる特別なメンテナンスとが含まれる。

【０００４】

特に、ハイブリッド（ネスト型）ＲＡＩＤシステムを含めた場合には、多種のＲＡＩＤが存在するが、ＲＡＩＤレベル０～６を用いれば、ディスク・ベースのストレージ・システムに関する全ての典型的なデータ・マッピング及び保護のスキームを定義することができる。ＲＡＩＤに関するシステムの他の分類としては、障害耐性（ドライブ障害に起因するデータの喪失に対する保護を行うシステム）、障害許容性（１又は複数のコンポーネントに起因するデータ・アクセスの喪失に対する保護を行うシステム）、及び災害許容（２以上の独立の地域（ゾーン）を用いる構成を備え、何れのゾーンも記憶されたデータに対するアクセスを提供するシステム）などがある。

【０００５】

通常用いられるＲＡＩＤレベルは、ＲＡＩＤ０、ＲＡＩＤ５、及びＲＡＩＤ６である。ＲＡＩＤ０は、パリティやミラーリングを用いず、冗長性ゼロでのブロック・レベルのストライピングを行う。ＲＡＩＤ０は、性能を向上させ且つストレージを増加させるが、障害許容性ではない。ＲＡＩＤ０では、ブロックは、個々のドライブへ同じセクタ上で同時に書き込まれ、小さいセクションのデータは各ドライブから並列に読み出され、それにより帯域幅が増大される。ＲＡＩＤ０は誤り検出を実行しないので、エラーは修正されない。ＲＡＩＤ０は、ハイブリッドＲＡＩＤシステムで、性能を向上させるために使用されることが多い。ＲＡＩＤ５は、データと共にパリティが分散される分散型パリティを用いるブロック・レベルのストライピングを行うものであり、１つのドライブ以外の全てのドライブが使用され、その１つのドライブは、１ドライブ障害（single drive failure、１つのドライブの障害）が発生したときに使用するリザーブ用である。１ドライブ障害が発生した場合、アレイは破壊されず、後続のデータ読み出しは、分散型パリティに基づいて計算され行われるので、エンド・ユーザにはドライブ障害を認識させない。しかし、１ドライブ障害は、障害したドライブを取り替えて関連するデータをリビルド（再構築）するまで、アレイ全体の性能を低下させる。ＲＡＩＤ６は、二つの分散型パリティを用いるブロック・レベルのストライピングを行うものであり、２ドライブ障害に対する障害許容性を提供する。アレイは、２個のドライブが障害するまで動作を継続できる。ＲＡＩＤ６の利点は、大きいＲＡＩＤグループを、より好適に作動するようにできることであり、この利点が重要なのは、大きい容量のドライブは、リビルドするため及び１つのドライブの障害から回復するために必要とする時間が長くなることと関連する。

【０００６】

コピーバックは、機能しているアレイ・メンバのディスクの内容を置換用ディスクへコピーすることにより、そのアレイ・メンバのディスクを別のアレイ・メンバのディスクと入れ替えることである。コピーバックは、障害しかけているコンポーネントが完全に障害してアレイの性能を低下させる前に置換するためや、アレイの特定の物理的コンフィギュレーションをレストア（復帰）するために用いられることが多い。

【０００７】

従来のスピンドル・ベースのハード・ドライブに格納される二次メモリは、アーマチュアにより保持される磁気ヘッドにより読み出されるデータを記憶する複数の回転ディスク（プラッター）を有する。現在のドライブは、通常、幾つかのヘッド及びプラッターを有する。１つのＩＯオペレーションを完了するために、アーマチュアは、ヘッドを、データを保持するプラッターのトラックのセクタへと移動させる必要がある。このプロセスはシーキング（シーク）と呼ばれるものであり、シーク時間と呼ばれる時間を必要とする。次に、このＩＯオペレーションでは、プラッターが回転されてセクタがヘッドの下へ来るまで待つ必要がある。この待ち時間は回転遅延（rotational latency）と呼ばれる。これらの時間及びその他の遅延は、ファームウェアや、ソフトウェアや、ドライブの応答に関連する他のハードウェアに起因するものである。

【０００８】
ＩＯＰＳ（「アイオプス」：１秒当たりのＩＯオペレーションの回数）は、ハード・ディスク・ドライブ（ＨＤＤ）やストレージ・エリア・ネットワーク（ＳＡＮ）等のコンピュータ・ストレージ・デバイスのベンチマークのために使用される一般的な性能尺度である。ストレージ・デバイスの製造者が発表しているＩＯＰＳの数値は、Ｉｏｍｅｔｅｒ（当初はインテル（登録商標）社により開発さけたソフトウェア）などのようなアプリケーションを用いて測定できる実際の応用での性能を保証するものではない。システム構成におけるＩＯＰＳの考えられる数値は、様々な変量（変数）に依存して大きく変わるものである。様々な変量としては、例えば、読み出し動作及び書き込み動作のバランス、シーケンシャル・アクセスとランダム・アクセスとのパターンの混合、ワーカー・スレッド（worker thread）の数及びキューの深さ、データ・ブロックのサイズ、そして、システム構成における他のファクタ、ストレージ・デバイス、ＯＳのバックグラウンドのオペレーションなどの要因が含まれる。

【０００９】
ＲＡＩＤアレイの中の１以上のハード・ドライブが障害した場合、その障害が生じたハード・ドライブをリビルドする必要がある。リビルドを行うためのＩＯオペレーションは、リビルドＩＯと呼ばれ、リビルド用ではない通常のＩＯオペレーション、例えば、ＲＡＩＤグループのハード・ドライブの通常動作用のＩＯオペレーションなどは、ホストＩＯと呼ばれる。ＲＡＩＤシステムでは、リビルド性能は、ホストＩＯ状態のもとでは強い影響を受ける。これは、リビルド動作では、ディスク・グループ中の残りの全てのディスクの読み出しを行う必要があり、各ディスクに対してシークを行う必要があるからである。それに加えて、それぞれのドライブ・モデルは、ＩＯキューを最適化するための個々の方法を有し（その方法はプロプリエタリである場合が多い）、その方法によりＩＯキューを整理し直してドライブのシークを最小化する。その結果、リビルドＩＯは、大きい影響を受け、大きい遅延（レイテンシ）に苦しむこととなる。なぜなら、通常、リビルドＩＯは最も整理し直されやすい（順序変更されやすい）ＩＯだからである。これは、リビルド性能に直接的に影響するものであり、システムは、例えば１ＴＢのデータをリビルドするために８～３０日という時間を必要とし得る。このようにリビルド時間が長くなると、ＲＡＩＤグループは、ホストＩＯの性能が低下した状態のもとに長い期間置かれることになり、これは、ＲＡＩＤグループ全体をオフラインとしてしまうようなデータ喪失をさせ得る第２や第３のドライブ障害を生じさせ得る機会を与えることになる。

【００１０】
全てのＲＡＩＤシステムは、通常、多くのＩＯキューをコントローラにより管理及び制御しており、それぞれのドライブはそれ自体のＩＯキューも有する。本発明は、後者の方、即ち、個々のドライブ内のキューを考慮するものであり、それは３２コマンド又は６４コマンドの深さである。リビルドＩＯは、大きい遅延や応答時間に苦しめられることが多い。これは、リビルドＩＯは、通常、ドライブのキュー内のホストＩＯと局所性が同じではなく、同じ箇所に存在していないからである。局所性に関しては、ディスク上のグループ化されたセクタの共通の領域又はクラスタという定義があり、従って、局所性がある場合、ドライブのヘッドは、１つのＬＢＡ（論理ブロック・アドレス）を得てから次のＬＢＡを得る際に、大きく離れた位置をシークする必要はない。従って、リビルドＩＯに関する動作の際に悪影響を及ぼす可能性がある。リビルド動作では、データを再構築するためにドライブ全体の読み出しを行うので、殆どのリビルド動作では、リビルドＩＯは、ホストＩＯと局所性を共有しない。典型的に、全てのシステムは、ドライブに対するリビルドＩＯとホストＩＯとのレートを制御するが、それらがドライブへ渡されると、ドライブはそれらを引き継いで、ここで説明したように、ＲＡＩＤコントローラにより計算されたレートを変え得るので、リビルドＩＯが枯渇させられる結果となる。

【００１１】
ドライブは、一連の１又は複数の同時のスレッド又はプロセスでリビルドすることができ、それらは、インプリメンテーションに依存するものであり、ＲＡＩＤシステムのファ

ームウェアにより決定されるものであり、使用可能なシステム資源や、ディスク・グループのＩＯサイズの粒度（細分性）、例えば、リビルドされる仮想グループのストライプのサイズなどに基づくものである。

【００１２】

本発明は、ホストＩＯ動作の処理を行いつつリビルドを行うための新規でヒューリスティック（発見的）な方法であり、それにより、リビルドＩＯに関する遅延が大きいことでリビルド時間が長くなるという問題に対処するものである。

【発明の概要】

【００１３】

従って、本発明の１つの態様は、決定性の様式でリビルド時間を大幅に改善するための方法であり、該方法においては、或るＲＡＩＤレベルのストレージ・システムがホストＩＯの状態のもとでリビルドを行っているときに、リビルドＩＯの長い遅延が起こらないようにし、この方法により、システムが、リビルドを行うと同時に、リビルドが行われていないときに行われる通常（従来）の動作を行うようにする。

【００１４】

本発明の別の態様では、ホストＩＯの状態のもとで、ｈｅａｄ－ｏｆ－ｑｕｅｕｅ（キューの先頭）ＳＣＳＩタグを用いて、リビルド動作と関連するＩＯを発行する。

【００１５】

本発明の別の態様では、リビルドのレートを制御して、ホストＩＯの窮乏(starvation)が起きないようにする（ホストＩＯの実行を極度に妨げないようにする）。この方法は、リビルドのレートが正しく制御され継続的に調節されて、最適なリビルドのレートが提供されることを保証し、かつ、ユーザの要求に応じてホストＩＯの処理が行われて、ＩＯホストの窮乏が生じないことを保証する。

【００１６】

本発明の更に別の態様では、ＲＡＩＤストレージ・システムで用いるファームウェアにより、リビルド・キューに対して遅延を故意にもたらすようにする。これはリビルドのレートを制御するものである。遅延は、ディスク・ヘッドの応答時間、リビルドに割り当てられる時間、リビルド・スレッドの数、及び一連のイベントのタイムライン（time line、時間線、スケジュール）の発見的な関数として計算することができる。１つの実施形態では、一連のイベントのタイムラインは７０秒毎のサイクルとなる。ドライブの挙動と、ホストＩＯ及びリビルドＩＯの作業負荷及び局所性を含むホストＩＯプロフィールとは、ドライブの順序変更アルゴリズムに影響を及ぼす重要なファクタ（要因）であり、リビルドＩＯの遅延の原因ともなる。これらは、リビルドＩＯの平均遅延を測定することにより、発見的なアルゴリズムへ組み入れられる。

【００１７】

本発明の別の態様は、ユーザによる選択又はユーザによるコンフィギュレーション（環境設定）が可能なリビルド・レートに関するものであり、このリビルド・レートは、オプションとしてオペレーション・メニューで提示することができる。メニューのオプションは、ホストＩＯと相対してどの程度速くリビルドが行われるかに関してや、ホストＩＯ動作に対してリビルドがどの程度の影響（インパクト）を及ぼすかに関して、例えば、「無インパクト（no impact、無影響）」、「低インパクト（低影響）」、「高インパクト（高影響）」、「最大インパクト（最大影響）」などとすることができる。

【００１８】

本発明の更に別の態様は、発見的なリビルドを行うものであり、この発見的なリビルドは、何れのハードウェアをも適応させるために絶えず再較正（再校正）を行う。１つの好適な実施形態では、この方法は７０秒毎に反復して行われる。

【００１９】

即ち、本発明は、ホストＩＯの状態のもとでのＲＡＩＤのリビルドを改善するシステム及び方法であり、リビルド時間を大幅に改善し、且つホストＩＯが窮乏しないようにする（ホストＩＯが極度に妨げられないようにする）。ＲＡＩＤの一部であるドライブにおけ

るキューは、リビルドＩＯ及びホストＩＯのリクエスト（要求）を記憶するために用いられ、リビルドＩＯは、ドライブのキューの先頭に向けて発行される。ドライブにおけるリビルドの要求は、遅延時間により遅延される。この遅延は、ＲＡＩＤシステムにホストＩＯの窮乏（ホストＩＯ要求が処理されない状態）をもたらすような、本発明の意図しない副作用が無いことを保証する。遅延は、複数の変数から導き出された関数として計算され、変数は、例えば、ディスク・ヘッドの応答時間、リビルドに割り当てられる時間、リビルド・スレッドの数、ドライブの挙動、リビルドＩＯプロフィール、作業負荷及び局所性を含むホストＩＯプロフィール、及び一連のイベントのタイムラインなどである。一連のイベントのタイムラインは、１つの実施形態では７０秒毎のサイクルである。リビルドＩＯプロフィールは、複数のオプションとして表されるリビルド・レートを有し、それらのオプションは、ホストＩＯと関連してどの程度速くリビルドが行われるかに関してや、ホストＩＯ動作に対してリビルドがどの程度の影響（インパクト）を及ぼすかに関しての、例えば、「無インパクト（無影響）」、「低インパクト（低影響）」、「高インパクト（高影響）」、「最大インパクト（最大影響）」などである。

【００２０】

ホストＩＯによる負荷が大きい状態で、或る構成では、本発明の方法及び装置を用いた場合のリビルド時間は、１０倍まで改善することができる。

上記の全ての利点を足し合わせて得られる利点や、本願で開示された様々な他の利点や本発明に内在する利点は、従来技術を超える改善をもたらす。

本発明の上記及び他の多くの特徴及び利点は、以下の詳細な説明を添付の図面とともに考慮すれば明らかになるであろう。

【００２１】

本発明の好適な実施形態の詳細な説明は、添付の図面を参照しつつ行われる。ここでは、本発明を実施するための現在において最適と考えられる様式を詳細に説明する。以下の説明は、発明を制限するものと判断すべきではない。以下の説明は、単に、本発明の一般的な原理を例示する目的でなされるものである。この説明における各章の名称や全体的な編成は利便性を考慮してのものであり、本発明を限定するためのものではない。

【００２２】

当業者であれば、本発明の教示を用いて、図面に示す実施形態を、本発明の精神から外れることなく変更することが可能であり得ることを、理解すべきである。図面では、１つの図面におけるエレメントに付された参照番号と同様の参照番号が、別の図面におけるエレメントに付されている場合、その別の図面のエレメントは、先の図面のエレメントと同様のエレメントである。

【図面の簡単な説明】

【００２３】

【図１】本発明の一般的な全体の流れを示すフローチャートである。

【図２】リビルドＩＯ対ホストＩＯの、ユーザにより定義可能又はコンフィギュレーション可能なリビルド・レートを示す。

【図３】動作中の本発明の好適な実施形態を示す更に詳細なフローチャートである。

【発明を実施するための形態】

【００２４】

本発明の方法及び装置は、ここで述べた機能を実施する有線又はソフトウェアでプログラムされるデバイス（例えば、ＡＳＩＣや、ＦＰＧＡなどのようなプログラマブル論理デバイス（ＰＬＤ）など）や、ファームウェアを実行するハードウェアや、ソフトウェアを実行するハードウェアなどにより構成することができ、ソフトウェアはメモリに記憶される。また、ここで用いる「ファームウェア」という用語は一般的な表現であり、類義語に置き換えることができ、又はドウェアやソフトウェアの組み合わせとして表すことができる。このハードウェアやソフトウェアは、例えば、ＡＳＩＣ、ＰＬＤ、コントローラ、プロセッサ、コンピュータ・システムなどであり、それらは、コンピュータ・プログラムを記憶するコンピュータ読取可能な記憶媒体を含み、コンピュータ・プログラムは、コンピ

ュータやソフトウェア・プログラマブル・デバイスと組み合わせて用いられるものであり、それらを動作させるための命令を含む。本発明の実施に用いるコンピュータ・システムは、典型的には、１以上のプロセッサと、プロセッサと共に働く一次メモリ及び二次メモリ（プロセッサはこれらのメモリに記憶された命令を実行する）と、モニタやマウスやキーボードなどのような入力／出力手段と、他の必要とされる別のハードウェアやファームウェアとを有する。本発明を構成するために使用されるソフトウェアは、そのソース・コードや機械語の中に、幾つかのクラス、機能、サブルーチン、オブジェクト、変数、テンプレート、モジュール、コードの行、コードの一部、及び本発明をここで説明し教示する連続するステージ（段）で実行するための構成（ここでは集合的に又は一般的に構成と記載したが、この説明に関連するフローチャートでは「プロセス・ステップ」、「ステップ」、「プロセス」、「ブロック」、「ブロック・ステップ」、「アプリケーション」、「命令」、「プログラム命令」、「モジュール」などのように示されている）を有することができ、このソフトウェアは、スタンドアローンのソフトウェア・アプリケーションとすることも、別のソフトウェア・アプリケーションの内部で用いるもとすることも、別のソフトウェア・アプリケーションにより呼び出されるものとすることもできる。

【００２５】

発明の詳細な説明の一部は、プロセス、プロシージャ（手順）、論理ブロック、機能ブロック、及び、他の象徴的な表現、例えば、コンピュータやプロセッサやコントローラやメモリの中のデータ・ビットやデータ・ストリームや波形に対するオペレーションの象徴的な表現に関するものである。ここで説明するプロセス、プロシージャ、ボックス、論理ブロック、機能ブロック、オペレーション（動作、演算）などは、一般に、物理量の物理的操作を含むものと考えられ、その物理量は、例えば、電気的信号、磁気的信号、光学的信号、そして、コンピュータやデータ処理システムや論理回路において記憶、転送、組み合わせ、比較、及び他の操作が可能なその他の信号などの形態である。これらの信号は、一般的な用法を考慮した場合、ビット、波、波形、ストリーム、値、エレメント、記号／符号、特徴、項、数などと呼ぶことが好都合である。

【００２６】

以下で更に説明するが、本発明に関して、前述のように、従来から考えられているリビルドは、ドライブのキューの順序変更（re-ordering）に起因する大きい遅延（レイテンシ）が問題である。ここで教示する解法は、リビルドＩＯに、ＳＣＳＩ　ＩＯタグ「キューの先頭（Head-of-Q）」を発行することであり、これは、そのＩＯをドライブのキューの先頭へ置き、そのＩＯを強制的に実行させる。これにより、リビルドの遅延を大幅に減少させ、リビルド性能を劇的に改善させることができる。ＩＯ（リビルドＩＯ）を発行する際には、ＳＣＳＩ　ＩＯタグ「キューの先頭」を用いるようにし、そのＩＯを、好ましくはキューの前部（先頭）以外の場所ではなくキューの前部（先頭）に置くようにする。この解法は、大きい遅延の問題を解決するが、二次的な問題を生じさせる。その第１の問題は、キューの先頭でリビルドの要求を強力に推し進めると、リビルドは速い速度で行われるが、システムの他の部分に関する命令／要求が欠乏（窮乏）して作動しない状態となることである。この第１の問題を解決するために、ここで教示するように、キューに対して遅延を適用する。しかしながら、この遅延も問題を生じさせ得る。例えば、固定の静的な遅延を用いた場合、全ての状態のもとにおいて最適の結果を得ることができず、状態が変わると結果も変わり、リビルドに１～２日を要する結果となる場合もある。従って、固定の静的な遅延の値に頼るのではなく、ここで教示するように、最適な遅延の値を計算する必要がある。ここで教示するように、最適な遅延は、発見的な方法に基づくものであり、様々な状態における多くの変数や作業に対処できる。ＲＡＩＤのユーザは、単に、ホストＩＯのインパクト（影響）に関して、性能を快適と感じるレベル（許容できる影響のレベル）をシステムへ入力するだけでよく、ここで教示するように、その入力に対応する残りの処理はシステムが行う。遅延は、性能を向上させるものではないが、システムが許容できる最低限の性能を有しつつリビルドが十分速く行われるようにするために、必要なものである。ここで教示する遅延を用いない場合、リビルドは速い速度で行われるが、以下

でも説明するようにシステムの性能は低下する。

【００２７】

ここで図１を参照する。図１は、ＲＡＩＤシステムがホストＩＯ状態のもとでリビルドされるときに用いられる本発明に係るＲＡＩＤシステムに関しての、本発明の全体の流れを概略的に示すフローチャートを示す。本発明に係るシステムは、発見的な様式でリビルドを行う。概略的には、本発明を実現するプログラム１０の流れは３つの段階（ステップ）を備える。本発明は、ファームウェア、ハードウェア、ソフトウェア、又はこれらを組み合わせたもの（以下では、単に「ファームウェア」という）に備えられることができる。

【００２８】

第１ステップ１２において、「リビルドＩＯ応答を測定」と記載されたボックスでは、プログラムは、解法の基礎となる平均リビルドＩＯ応答時間を計算する。例えば、プログラムは、本発明を含まない従来の技術を用いる通常のリビルド、即ち、「従来の」リビルドが行われる間にデータを読み出すハード・ドライブのヘッドの応答時間を読み出し、ミリ秒（ｍｓ）単位で平均応答時間を計算する。入力１４は、「平均リビルド読み出し応答を計算」と示されている。平均リビルド応答時間は、事実上、任意の時点におけるホストＩＯの作業負荷、局所性、及びドライブのキュー管理効率を測定するものである。ここで説明するように、平均リビルド応答時間が、特定の経験的に決定された閾値よりも下である場合、従来技術によるリビルド機構が、リビルドに関して十分に速く動作していると判断できる。しかし、平均リビルド応答時間が閾値よりも大きい場合、リビルド・コマンドをＲＡＩＤデバイスのキューの先頭（前部）へと動かし、それが即座に処理されるようにして、リビルド性能を向上させる。

【００２９】

第２ステップ１４、すなわち、「『キューの先頭』へ付加するために最適遅延Ｔｍｓを校正」と記載されたボックスでは、プログラムは、発見的な様式で、ミリ秒（ｍｓ）単位で最適遅延Ｔを計算し、この遅延時間により、ドライブのキューへのリビルドＩＯの発生発行を遅延する。この「キューの先頭」キューは、ドライブのベンダにより実装された個々のドライブのファームウェアにおける個々のドライブに関　するキューである。このステップでは、ＲＡＩＤシステムは、ここで教示するように、とりわけキューに関連する従来とは異なる様式でリビルドを行う。ＲＡＩＤシステムに対して、キューは、何れのＩＯプロセスを処理するか、及び、何れの処理のシーケンス（キューの形）で処理するかを示すものであり、処理のシーケンスとは、例えば、ホストＩＯに続いてリビルドＩＯ、リビルドＩＯに続いてホストＩＯ、２つのリビルドＩＯに続いてホストＩＯなどのようなものである。「キューの先頭」とタグ付されたリビルドＩＯは、そのリビルドＩＯを即座に処理すべきこと、即ち、ドライブがキューに対する処理や順序変更の動作に戻る前に処理すべきことを、ドライブに対して示す。遅延Ｔは、ドライブのキューに対してリビルドＩＯ（リビルドＩＯ要求）が発行される前に遅延（ミリ秒単位）される時間である。この遅延（休止）の間、キューの先頭（前部）に対してリビルドＩＯは発行されない。この休止期間があることにより、ドライブはそのキュー（既にキューに含まれているＩＯ要求）を実行でき、そのキューに残った部分を実行すべきときに、「キューの先頭」リビルドＩＯが発行されてドライブがそれに占有されることを防止する。

【００３０】

図１のステップ１６におけるファクタ、即ち、リビルドＩＯコマンド（又はリビルドＩＯ）をドライブのキューの先頭へ発行することを遅延するための、必要とされる最適遅延時間の計算、又は遅延の校正（較正）に用いられるファクタには、幾つかのファクタが含まれる。

第１のファクタは、キューと関連する「キューの先頭」リビルド（即ち、リビルドＩＯをキューの先頭へ置いて行う処理）が行われている間にデータを読み出すハード・ドライブのヘッドの平均リビルド読み出し応答時間である。これは、入力１８において「平均リビルド『キューの先頭』読み出し応答時間を計算」と示されている。

【００３１】
ステップ１６の遅延の校正に用いる第２のファクタは、ユーザ選択の又はコンフィギュレーション可能なリビルドＩＯプロフィールからリビルドへ割り当てられる時間の計算であり、入力２０において「リビルドＩＯプロフィールを計算」と示されている。リビルド・プロフィールは複数のリビルド・レートを含み、これは、どの程度積極的にリビルドを行うかを決定する。リビルド・プロフィールは、また、合計時間と比較しての、リビルドに割り当てられる時間として見ることもできる。ここで教示する１つの実施形態では、ユーザは、メニュー中の一連の選択肢からプロフィールを選択することができ、その選択肢には、リビルドＩＯに対して十分な優先度が与えられずホストＩＯに対して比較的高い優先度が与えられる低インパクト・リビルド（「低」）のような最小リビルド・レート、リビルドＩＯ要求に対してもホストＩＯ要求と同じ優先度が与えられる高インパクト・リビルド（「高」）、ホストＩＯに勝る最高の優先度がリビルドＩＯに対して与えられる最大インパクト・リビルド（「最大」）などがある。この選択は、自動的に行うこともできる。なお、このような等級（選択肢におけるインパクトの段階）は実施の態様に依存するものであり、リビルド・プロフィールにおいて、任意の数の等級を設定することが可能であり、また、ゼロ又は最小値から最大値までの連続的で平滑な関数を用いることも可能である。更に、リビルド・プロフィールのレート及び割り当てられる時間を、ユーザ入力なしで自動的に決定する構成や、ユーザ入力と関連して自動的に決定する構成とすることも可能である。例えば、リビルド・プロフィールのレート及び割り当てられる時間を、エキスパート・システムの場合のように、履歴データを用いてルックアップ・テーブルから自動的に選択するように構成でき、この場合、様々な類似のストレージ・システム及び／又は特定のハード・ドライブ製造者から収集した履歴データに基づいてプロフィールを決定するように構成できる。
【００３２】
ステップ１６の遅延の校正において遅延を決定するために用いる第３のファクタは、リビルド・スレッドの数の計算であり、リビルド・スレッドの数は、ハード・ドライブ製造者の実施の形態に依存するものであり、図１の入力２２において「リビルド・スレッドの数を計算」と示されている。デバイスは、使用可能なシステム・リソース及びディスク・グループのＩＯサイズの粒度（細分性）、例えば、リビルドされている仮想グループのストライプのサイズ、に基づいて、ＲＡＩＤシステムのファームウェアで決定されたように、データの１以上の一連のスレッドでリビルドすることができる。例えば、ストライプ・サイズが１メガの仮想ドライブに対しては、ディスクに対しての各ＩＯに対してキャッシュを使用する必要があるので、一度に１つのスレッドのみが発行される。しかし、ストライプ・サイズが６４キロの仮想ドライブは８個のスレッドを実行することができる。なぜなら、８個のそのようなＩＯが使用するのは、ディスクあたり５１２キロの量のキャッシュのみだからである。これらは本質的に知られていることであり、実施の形態により変わる。
【００３３】
ステップ１６の遅延の校正に用いる第４のファクタは、リビルドに許される時間の長さに関する１以上の時間定数の使用である。１つの実施形態では、その時間定数は１つのＩＯ応答に対して１０００ｍｓであるが、一般に、特定の実施の態様に応じて適宜に決定した値とすることができる。更に、本発明の方法の全てを行うために要する期間もまた別の時間定数であり、１つの実施形態では、後に説明するように７０秒であるが、一般に、その時間長は、普遍性を喪失しない範囲で適宜に決定した長さとすることができる。
【００３４】
最後である第３ステップ２４については、そのボックスに「遅延Ｔｍｓ毎に『キューの先頭』へリビルドＩＯを発行」と記載されており、このステップにおいて、本発明のプログラムは、先行するステップで計算された遅延を実際に実施する。この処理では、本発明を用いるファームウェアにより、キューの前部（キューの先頭）へのリビルドＩＯコマンドの発行を、第２ステップ１６で決定された時間Ｔミリ秒（ｍｓ）だけ遅延する。この遅

延時間の間、キューへリビルドＩＯが送られない。

【００３５】

本発明を用いるＲＡＩＤシステムでは、図１に示す技術を用いると、リビルド処理の間の性能が顕著に向上する。その有効性を示すために、以下の表Ａ及び表Ｂにシミュレーションの結果を示す。幾つかの場合においては、性能における１０倍の向上が見られる。表Ａは、本発明に従って行ったリビルドに関するものであり、表Ｂは、従来のリビルド手法のもとで行ったリビルドに関するものである。

【００３６】

【表１】

表Ａ　－　本発明を用いるリビルド
キューの先頭リビルドＩＯＰＳおよび応答時間（ｍｓ）６４Ｋ

番号	ホストＩＯ	ＩＯＰＳ	平均応答(ms)	ＭＢ／Ｓ
１	ＩＯ無し	２２０	２～４０	１３．７４
２	１ＱＤ近	１５６	４	９．７４
３	１ＱＤ遠	１４６	９	９．１２
４	１６ＱＤ近	１４０	１０	８．７４
５	１６ＱＤ遠	１３０	１７	８．１２
６	２５６ＱＤ近	１２８	１６	７．９９
７	２５６ＱＤ遠	１２３	２０	７．６８

【表２】

表Ｂ　－　本発明を用いないリビルド
リビルドＩＯＰＳおよび応答時間（ｍｓ）６４Ｋ

番号	ホストＩＯ	ＩＯＰＳ	平均応答(ms)	ＭＢ／Ｓ
１	ＩＯ無し	１３９８	０	８７．２９
２	１ＱＤ近	５４０	４	３３．７２
３	１ＱＤ遠	４２２	１２	２６．３５
４	１６ＱＤ近	１０２	７７	６．３７
５	１６ＱＤ遠	１０６	８０	６．６２
６	２５６ＱＤ近	１１	１２００	０．６９
７	２５６ＱＤ遠	９０	９７	５．６２

【００３７】

上記の表の用語について説明する。第１コラムの「番号」は、考慮している例におけるケース番号であり、表Ａ及び表Ｂにはそれぞれ７個の例が示されている。第２コラムの「ホストＩＯ」に関して、「近」は、リビルド領域がホスト領域に近いことを表す。ホストＩＯの作業負荷が大きい場合において、リビルド領域がホスト領域に近い場合には、遠い場合よりも、その処理に関する遅延が長くなることが知られており、表Ｂの行６（番号６の行）はその一例である。第２コラムでは、リビルド領域がホスト領域から遠いこと「遠」と示している。多くの製造業者は、ファームウェアに依存するＩＯキューを最適化するためのプロプリエタリの方法を有するので、この「近」及び「遠」に関する現象についての共通する理由付けはできないが、その現象が観察されていることは事実である。第２コラムの「ホストＩＯ」は、ＩＯ要求に応答する指定されたホストＩＯを表す。このコラムにおいて、「ＱＤ」はキューの深さ（Queue Depth）を表し、キューの深さとは、ホスト

ＩＯ要求の間に存在する（ホストＩＯ要求に応じての）ホストＩＯの数を測定したものである。「１ＱＤ」では、非常に軽いホストＩＯドライブ順序変更スキームとなり、システム全体に対するインパクトを実質的には与えない。１ＱＤでは、ドライブのＩＯキューに含まれるホストＩＯは１個のみであるので、そのＩＯキューへ多くのリビルドＩＯを含ませることができる。それに対して、１６ＱＤは、比較的典型的なホストＩＯ作業負荷であり、或る時点で１６個の処理されるべきＩＯが存在することを示す。１ＱＤと対極にあるのは２５６ＱＤであり、これは、或る時点で多数の処理されるべきＩＯが存在することを示すものであり、これは重い作業負荷である。最大が２５６ＱＤというのは典型的ではないが、この例では最も長いリビルド時間を生じさせている。

【００３８】

第３コラムの「ＩＯＰＳ」は、１秒あたりの入力／出力リビルド・オペレーションの数である。第４コラムの「平均応答」は、リビルドＩＯに対するドライブの平均応答の時間をミリ秒単位で示す。最後のコラムの「ＭＢ／Ｓ」は、リビルドＩＯに関してのデータ転送レート（スループット）であり、「メガバイト／１秒」という単位で表している。

【００３９】

表Ａと表Ｂとを比較すると分かるように、本発明を用いることにより、従来のリビルド処理と比較してシステムの性能が顕著に向上する。例えば、番号「５」の行の例では、作業負荷は１６ＱＤという中程度のものであるが、従来のリビルドでの平均応答時間は長く、８０ｍｓである。しかし、本発明の技術を用いた場合、番号「５」の行の例での平均応答時間は１７ｍｓであり、８０ｍｓと比較して７９％短くなっている。ＩＯＰＳは１０６から１３０へ、スループット（ＭＢ／Ｓ）は６．６２ＭＢ／ｓから８．１２ＭＢ／ｓへと向上している。

【００４０】

同様に、重い作業負荷、例えば、番号「７」の行の例で示される「２５６ＱＤ遠」の場合にも、本発明の技術を用いると、従来のリビルド処理の場合よりも性能が改善されている。この場合、ＩＯＰＳでは９０から１２３へと３７％の向上が見られ、スループット（ＭＢ／Ｓ）では５．６２ＭＢ／ｓから７．６８ＭＢ／ｓへと上昇し、更に、平均応答時間は９７ｍｓから２０ｍｓへと７９％低下しており、この２０ｍｓという値は９７ｍｓという値と比べてかなり好適なものである。

【００４１】

ＩＯ作業負荷が軽い場合には、本発明を用いても利益は得られない。このことは認識されており、本発明の方法でも織り込み済みである。例えば、表Ａと表Ｂとの番号「１」の行の例を比較すると、「ＩＯ無し」の場合では本発明による利益はなく、本発明を用いることにより性能は低下している。番号「２」の行の例を比較すると、「１ＱＤ近」の場合でも、従来のリビルド処理と比較しての本発明の利益はない。（なお、表Ａの場合において、番号「２」の行の例に関しては、１つの「キューの先頭」リビルドＩＯ要求と別の「キューの先頭」リビルドＩＯ要求との間に５０ｍｓの最小遅延時間が付加されている。）しかし、番号「３」の行の例では、「１ＱＤ遠」の場合に関して、本発明を用いた場合には、平均応答時間において１２ｍｓから９ｍｓへと低下しており、少しではあるが改善が見られる。しかし、この例では、本発明を用いてもスループット（ＭＢ／Ｓ）に関して利益は得られず、従来のリビルド処理でのスループットは２６．３５ＭＢ／ｓであり、これは、本発明を用いた場合のスループットである９．１２ＭＢ／ｓよりも高い。また、表から理解できるように、一般には「近」の状態のほうが「遠」の状態よりも処理が速くなるが、例外もある。従来のリビルド処理に関する表Ｂにおいて、番号「６」の行、即ち、重い負荷の例である「２５６ＱＤ近」で長い遅延が発生しており、これは、番号「７」の行の「２５６ＱＤ遠」と比較すると差が明確に理解できる。この例では、「近」のＩＯ要求における通常の場合のような速い応答がなされず、反対に応答が遅くなっており、「遠」の平均応答時間は９７ｍｓであるが「近」の平均応答時間は１２００ｍｓであり、「遠」のほうの平均応答時間のほうが１２倍以上速くなっている。この逆転現象は、本発明により対処され解決される１つの問題の例であり、表Ａにおける番号「６」の例と番号「７」

の例とを比較すると本発明の効果が理解できる。表Ａと表Ｂとのそれぞれの「２５６ＱＤ近」に関する性能を比較すると、平均応答時間は１６ｍｓと１２００ｍｓとであり、本発明を用いた場合には平均応答時間が９８．７％減少しており（即ち、７５倍速く（１２００／１６））、表Ａに示す処理の性能が表Ｂに示す従来の処理の性能に勝っていることが理解できる。スループットに関しても、０．６９ＭＢ／ｓ（表Ｂ）から７．９９ＭＢ／ｓ（表Ａ）へと劇的に増加しており、これは１桁異なる増加であり、より正確には１１倍以上の向上である。また、表Ａの番号「７」の例と、表Ｂの番号「５」の例とを比較すると、表Ａの番号「７」の例（本発明を用いた場合）の平均応答時間は２０ｍｓであり、表Ｂの番号「５」の例（従来のリビルド処理を用いた場合）の平均応答時間は８０ｍｓであり、表Ａの番号「７」の例では、その作業負荷が表Ａの番号「５」の例の作業負荷よりも重いにもかかわらず、平均応答時間が良好に（短く）なっている。

【００４２】

作業負荷が軽く又は無く、キューの深さが浅い状態では、従来のリビルドの方法と比較して本発明の方法が利益をもたらさないことが、表Ａ及び表Ｂから理解できる。従って、以下に説明するように、本発明では、リビルドを行っているときに、リビルド・レートと応答時間とを考慮し、１つの実施形態では、遅延を伴う「キューの先頭」リビルド方法（リビルドＩＯをキューの先頭へ向けて発行する方法）のみを実施するが、この方法は、リビルド・レートがベースライン（基準値）の３３％を超える大きさの場合、又は応答時間が４５ｍｓよりも長い場合に実行する。これにより、本発明では、表Ａにおける「番号」の大きい行で示す例、例えば、番号「４」から番号「７」の行に示す例の範囲の状態において、本発明の処理が行われるようにする。

【００４３】

次に、図２を参照する。図２はユーザ定義可能すなわちコンフィギュレーション可能なリビルド・レートを示す。図２では、グラフィカル・ユーザ・インターフェース（ＧＵＩ）２００を用いてリビルド・レートを選択する方法を示すが、本発明では、ユーザにより操作されるＧＵＩは必ずしも必要ではない。従って、ＧＵＩと関連する説明は、コンフィギュレーション可能かつ変更可能な様式でのリビルド・レートの操作を説明するために用いる概念的なツールの説明と理解すべきである。なぜなら、リビルド・レートは、実際には、ＧＵＩや人間の操作者により選択及び変更するのではなく、自動的に選択及び変更することが可能であり、また、固定のシーケンスで選択することも可能であるからである。ＧＵＩ２００は、本発明によると、ユーザが望むＲＡＩＤシステムのリビルド処理を選択するために、様々な強度の度合いに対応する複数のボタン２１０、２１５、２２０、２２５を備える。その度合いは、例えば、「低」、「高」、「最大」、「無し」であり、それぞれ、低い値／低インパクト、高い値／高インパクト、最大値／最大インパクト、リビルド無し／インパクト無し、というリビルド・レートに対応する。例えば、１つの好適な実施形態では、「無インパクト」、「低インパクト」、「高インパクト」、「最大インパクト」という４個のレベルのリビルド・レートがユーザに提示される。なお、インパクトとは、ホストＩＯがリビルドにより受ける影響を意味する。

【００４４】

「無インパクト」の場合、本発明のＲＡＩＤシステムは、従来のリビルド処理を行う。このオプション（「無インパクト」）の場合、本発明の処理は行われないので、ホストＩＯに関する性能に対するインパクトは最小限となる。また、このオプションの場合、ユーザにとって、リビルド時間は問題にはならない。

【００４５】

他の３個のオプションでは、本発明に従ったリビルドの動作を行う。図２に示す実施形態では、「低インパクト」を選択した場合、ホストＩＯに関する性能は、「無インパクト」の場合と比べて３分の１が影響を受ける。より詳細には、図２の棒グラフで示す「『キューの先頭』リビルドＩＯ」において、低インパクトの場合は、割り当てられた１０００ｍｓのうちの３３３ｍｓを使っている。図２の２４０で示すように、キューの先頭へ向けて発行されるリビルドＩＯ（従って、「『キューの先頭』リビルドＩＯ」と示している）

には、動作のために１０００ｍｓのうちの３３３ｍｓが与えられる。残りの６７７ｍｓは本発明のファームウェアに与えられ、ホストＩＯ要求を処理するために使用されるので、この部分に関して図２では「ホストＩＯ」と示している。「高インパクト」を選択した場合、リビルドＩＯがキューの先頭へ向けて発行され、図２の２５０で示すように、本発明におけるリビルドＩＯ処理は５００ｍｓ（これは１０００ｍｓのうちの５００ｍｓであり、半分の時間）にわたって行わる。半分の残った時間は、ホストＩＯに関する処理に対して与えられる。「最大インパクト」を選択した場合、リビルドＩＯがキューの先頭へ向けて発行され、図２の２６０で示すように、本発明におけるリビルドＩＯ処理は６６６ｍｓ（１０００ｍｓのうちの６６６ｍｓであり、約６７％）にわたって行われる。約３３％の残った時間は、ホストＩＯに関する処理に対して与えられる。特定の時間間隔の間には、リビルドＩＯが全く発行されないが、ホストＩＯの間には散在している。リビルドＩＯは、望まれるリビルド・レートを提供するため、及び効果的なサービス時間を達成するために、適切な遅延を伴って発行される。

【００４６】

図２では、性能に関する個別の段階的な等級を選択可能とする４個のオプション・ボタンのみを示しているが、任意の数の選択可能なオプションを設定できる。例えば、ユーザは、スライダ２３０として示すスライダ・バーを用いてリビルド・レートを設定することができ、また、０％～最大％までの間のパーセンテージを設定することによりリビルド・レートを決定することもできる。更に、人工知能手段を用いて、時刻、リビルド動作の過去の経験、製造者のデータなどのような履歴データに基づいて、リビルド・レートを自動的に設定することもできる。

【００４７】

次に、本発明の好適な実施形態の更に詳細なフローチャートを示す図３を参照する。図３のフローチャートは、０秒～７０秒までのタイムラインに沿ったサイクルで、校正、再校正、及びリビルドの作業の行われる態様を示す。タイムラインはＴ０～Ｔ７０までのものとして示され、第１のステージはＴ０～Ｔ５、第２のステージはＴ５～Ｔ１０、第３のステージはＴ１０～Ｔ７０の間となっている。図３に示す本発明の方法は、３ステージに分割されており、処理に要する時間は７０秒であるが、これに限らず、この処理に要する時間はここでの教示に従って任意に設定することができる。「通常のリビルドＩＯを発行」と示された第１のステージは、Ｔ０～Ｔ５の５秒で終了する測定段階である。「『キューの先頭』へリビルドＩＯを発行」と示された第２のステージは、Ｔ５～Ｔ１０の５秒で終了する段階であり、リビルドＩＯ要求をキューの先頭へ向けて発行する間において必要とされる遅延Δ（すなわち、遅延Ｔ）をミリ秒単位で計算する。この遅延は、前述のようにリビルドに関する遅延を軽減するためのものである。この遅延は、ヘッド応答時間、リビルドに割り当てられる時間、リビルド・スレッドの数などのようなパラメータを考慮したものである。「遅延（ｍｓ）毎に『キューの先頭』へリビルドＩＯを発行」と示された最後の段階である第３のステージは、Ｔ１０～Ｔ７０の６０秒で終了する段階であり、Δミリ秒毎に「キューの先頭」へ向けてリビルドＩＯを発行することにより、第２のステージで計算された遅延を導入する。計算された遅延Δは、キューの先頭へ向けてのリビルドＩＯの発行を遅らせる時間である。第３のステージが完了するとサイクルが反復される。このようにして、本発明の方法は、発見的な様式で与えられるパラメータに従って、ＩＯの再発行を動的に再校正する。

【００４８】

第１のステージは、Ｔ０～Ｔ５の間の５秒間（７０秒のうちの５秒、即ち、全時間の約７％）であり、この第１のステージにおいて、本発明に従ってＲＡＩＤシステムのドライブを動作させるファームウェア、ハードウェア、及び／又はソフトウェア（単に「ファームウェア」という）は、「平均リビルド読み出し応答時間を計算」と示された図３のボックス３２５に示すように、平均リビルド読み出し応答時間を計算する。このリビルドが従来の方法により行われる場合、即ち、本発明を用いず従来技術を用いるリビルド（通常リビルドＩＯと言う）が行われる場合、ボックス３１０に示すように「通常のリビルドＩＯ

を発行」となる。次に、「リビルド・レート＜３３％　｜｜　応答時間＜４５ｍｓ？」と示された判断ボックス３３０において、ファームウェアは、この従来のリビルドにおいて、リビルド・レートが３３％未満であるか（即ち、ホストＩＯに対してインパクトが無いことを、ユーザが要求していることを示すか）、リビルドに関する応答時間が４５ｍｓ未満であるか（即ち、システムに対して従来のリビルド方法が最適であることを示しているか）を検査する。判断ボックス３３０におけるこの条件が満たされた場合、ファームウェアは、「ＹＥＳ」と記載された経路をたどり第１のステージの処理を反復する。この処理は、判断ボックス３３０において条件が満たされなくなる時（「ＮＯ」の状態）まで反復される。「ＮＯ」の状態となった時点で、処理はＴ５～Ｔ１０の第２のステージへ移る。

【００４９】

第２のステージは、Ｔ５～Ｔ１０の間の５秒間（７０秒のうちの５秒、即ち全時間の約７％）であり、この第２のステージにおいて、ファームウェアは、キューにおける１つのリビルドＩＯコマンドと別のリビルドＩＯコマンドとの間に必要な最適の遅延を計算する。この遅延は、上記で説明したように、ホストＩＯの窮乏及びリビルドＩＯの大きい遅延を避けるためのものである。この最適の遅延を求めるために、図３のボックス３４０に示すように、ファームウェアは、実際の環境から様々なパラメータを計算し、第２のステージの時間間隔全体にわたってリビルド「キューの先頭」読み出し応答時間の平均値を計算し、第２のステージの最後にこの平均値を用いる。上述のように、この計算には様々なファクタが組み込まれる。

【００５０】

最初に、システムがキューの先頭へリビルドＩＯを発行して、それがパラメータとしてファームウェアにより読みとられると、５秒間に、ハード・ドライブのヘッドの読み出し応答時間の平均値が計算される。キューは、この分野の技術では知られた技術であり、ＲＡＩＤシステムの特定のドライブに対するキューには、リビルドＩＯ及びホストＩＯが記憶される。本発明ではリビルドＩＯをキューの先頭に配置するので、図３では「キューの先頭」と示している。リビルドＩＯ要求は、ホストＩＯ要求とともにキューへ送られ、１つのリビルドＩＯ要求と別のリビルドＩＯ要求との間には遅延時間が与えられる。この遅延時間は、初期的にデフォルトで１００ミリ秒とすることができ、また、前の７０秒サイクルにおいて処理が行われていた場合には、そのサイクルで計算された遅延時間を用いることができる。ハード・ドライブのヘッドの読み出し応答時間は、ファームウェアにより読み出され、このステージ中にその平均が求められる。この処理に関して、図３のボックス３４０では「平均リビルド『キューの先頭』読み出し応答時間（ｒｅｓｐ）を計算」と示し、この平均をとられたパラメータを「ｒｅｓｐ」と表す。

【００５１】

次に、リビルド・プロファイルが決定され、リビルドに割り当てられる時間が確かめられる。リビルドに割り当てられる時間は、ホストＩＯ要求及びリビルドＩＯ要求の双方を含むリビルドの全時間と相対して求められたものである。前述のように、リビルド・レートが３３％～６５％のリビルド・レートで、ユーザ及び／又はシステムが「低」を選択した場合、そのパラメータが、変数「リビルドに割り当てられる時間（ｒｔ）」に関して選択される。様々なリビルド・プロファイルの選択肢（オプション）に対しての「ｒｔ」（ミリ秒で表す）は、（１）３３％～６５％のリビルド・レートで３３３ｍｓ（低のオプション）、（２）６６％～９８％のリビルド・レートで５００ｍｓ（高のオプション）、（３）９９％～１００％のリビルド・レートで６６６ｍｓ（最大のオプション）である。これらのオプションの数は実施形態に依存するものであり、３個のみではなく任意の数のオプションを設けることができ、段階的な等級として設定することも、最小値から最大値まで連続的なものとすることもできる。

【００５２】

そして、図３のボックス３４０に示されるように、リビルドＩＯＰＳ（ｒＩＯＰＳ）が、

リビルドＩＯＰＳ（ｒＩＯＰＳ）＝ｒｔ／ｒｅｓｐ

に従って計算される。上記の式において、「ｒｔ」は、リビルドに割り当てられる時間であり、前の計算ステップで得られたものでる。「ｒｅｓｐ」は、前述のように、「平均リビルド『キューの先頭』読み出し応答時間」、すなわち、キュー先頭読み出し応答時間の平均値である。

【００５３】

そしてさらに、パラメータ「ｎｕｍＲｅｂｕｉｌｄｓ」すなわち「ｎＲ）」が計算される。「ｎＲ」は、リビルドＩＯＰＳ（ｒＩＯＰＳ）を、対象となっているＲＡＩＤシステムにおけるリビルド・スレッドの数（図３のボックス３４０における「リビルド・スレッドの数」）で割ったものである。「ｎＲ」の値は、

ｎｕｍＲｅｂｕｉｌｄｓ（ｎＲ）

＝ｒＩＯＰＳ／（リビルド・スレッドの数）

により計算される。

【００５４】

最後に、Ｔ５～Ｔ１０の間の第２のステージの最終処理で、リビルドＩＯ間で用いる実際の遅延が、

リビルドＩＯ間の遅延＝１０００／ｎＲ－ｒｅｓｐ

に従って計算される。遅延はΔ又はＴと表す場合もある。上記の式において、１０００は時間定数であり、単位はミリ秒である。この時間定数は任意の値とすることができる。なお、ＩＯＰＳは１秒あたりの値なので、使用に好適な定数は、１秒に対応する１０００ｍｓである。しかし、この定数が変えられた場合、ｒＩＯＰＳも変更する必要がある。このことは、当業者であれば、ここでの教示から理解できる。従って、図３に示すＴ５からＴ１０の間の第２のステージの計算は、この時間間隔以外の任意の時間定数を用いて行うことができる。なお、上記の式において、前述のように「ｎＲ」は「ｎｕｍＲｅｂｕｉｌｄｓ」であり、「ｒｅｓｐ」は「平均リビルド『キューの先頭』読み出し応答時間」である。

【００５５】

遅延時間Δ（又はＴ）が計算されると、本発明の方法は第３のステージへ進む。この第３のステージはＴ１０～Ｔ７０までであり、６０秒である。この第３のステージで、ＲＡＩＤシステムの個々のドライブのキューの先頭へ向けて発行されるリビルドＩＯを遅延させるために、前のステージで計算した実際の遅延Δが用いられる。なお、キューは、リビルドＩＯ及びホストＩＯを含むものである。それぞれのリビルドＩＯは、そのリビルドＩＯ要求がドライブへ向けて発行される前に、遅延Δだけ、ファームウェアにより遅延される。このような遅延をキューにおいて用いることにより、上述のように、ホストＩＯを窮乏しないようにする効果がある。この期間の最後である時点Ｔ７０で、即ち、時点Ｔ０で、校正が新たに開始され、処理が反復される。このように、本発明は発見的及び日和見的であり、ハードウェアの実際の状態を考慮に入れるものである。その理由は、遅延を計算する際にも、ハードウェアの状態は時間とともに変化するからである。

【００５６】

ここで説明した本発明の方法及び装置は、ＲＡＩＤ２、３、及び４を含めてのＲＡＩＤ１～６までの任意のＲＡＩＤレベルで使用可能であり、又はハイブリッドＲＡＩＤシステムや、消失訂正符号（ｅｒａｓｕｒｅ　ｃｏｄｅ）などを用いる新しいタイプのフェイルセーフ型及び冗長型のストレージでも使用可能である。更に、本発明の方法及び装置は、コピーバック・オペレーションや、ＲＡＩＤコントローラで用いる必要のある任務で重要なプロセスなどにおいて、用いることができる。この重要なプロセスとは、物理デバイスへＩＯを発行する必要があり、かつ、ホストＩＯに対する管理可能なインパクトで、予測可能に完了せねばならないプロセスであり、例えば、メディアのスキャン、整合性の検査、初期設定、フォーマッティングなどであるが、これらに限定されるものではない。

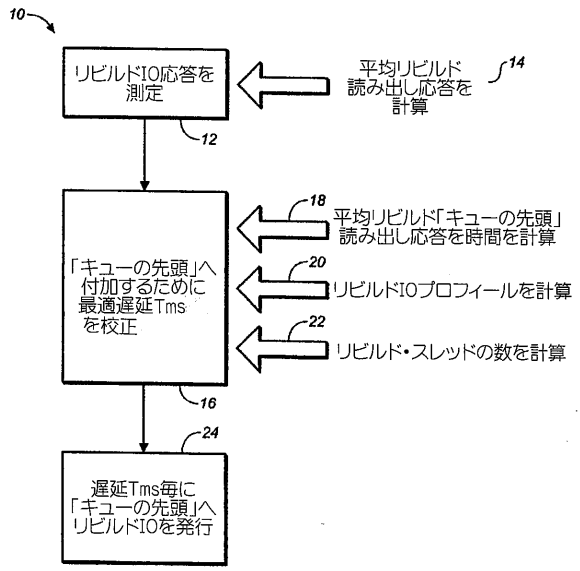
【００５７】

本発明の実施形態の構成に関して、当業者であれば、本発明の精神から離れることなく、変更、構成の部分的な削除、別の構成の付加などを行うことができる。更に、ここでは

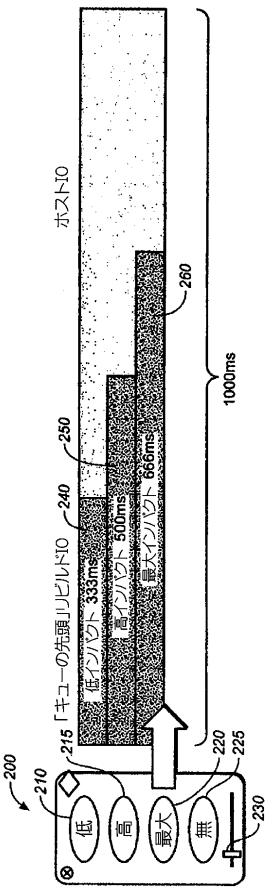
本発明の様々な構成について記載したが、これらの構成の組み合わせを、本発明を解体して構成できる実施形態において用いることもできる。本発明の範囲は、特許請求の範囲により定められる。

本発明の範囲は、変更や付加などがなされた構成の全てにわたることを意図しており、本発明の範囲は、特許請求の範囲によってのみ制限される。

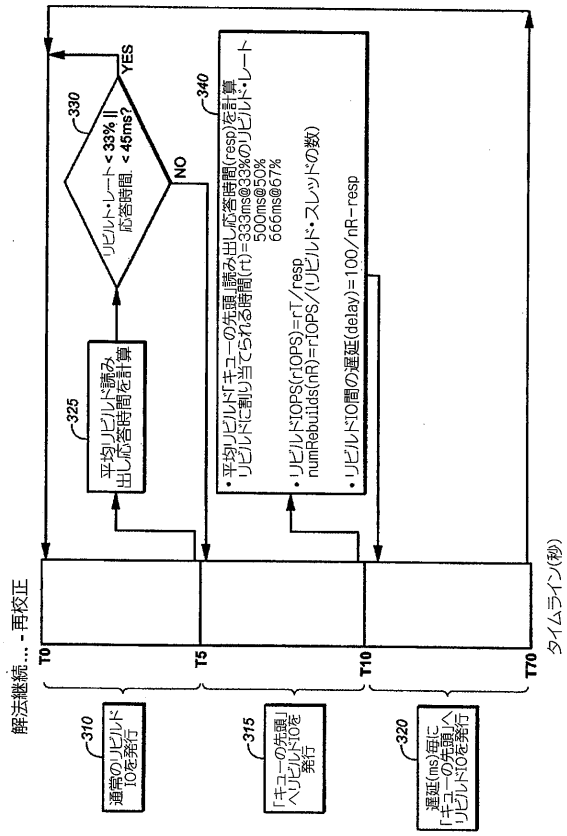
【図１】



【図２】



【図３】


解法継続 ... - 再校正
T0
T5
T10
T70
タイムライン(秒)
310
通常のリビルドIOを発行
315
「キューの先頭」へリビルドIOを発行
320
遅延(ms)後に「キューの先頭」へリビルドIOを発行
325
平均リビルド読み出し応答時間を計算
330
リビルド・レート < 33% || 応答時間 < 45ms?
YES
NO
340
• 平均リビルド「キュー」の先頭/読み出し応答時間(resp)を計算
リビルドに割り当てられる時間(rt)=333ms@33%のリビルド・レート
500ms@50%
666ms@67%
• リビルドIOPS(rIOPS)=rT/resp
numRebuilds(nR)=rIOPS/(リビルド・スレッドの数)
• リビルドIO間の遅延(delay)=100/nR−resp

フロントページの続き

(72)発明者　ナマン・ネア
アメリカ合衆国カリフォルニア州９５０３５，ミルピタス，バーバー・レイン　１６２１，エイディー８００
(72)発明者　カイ・エム・レ
アメリカ合衆国ジョージア州３００９３，ノークロス，シャックルフォード・ロード　４１６５，エイエヌオーアールシー