

【特許請求の範囲】**【請求項 1】**

(a) ストレージデバイスが一つ以上の第一の契機を受信して、ソースデータをプライマリストレージに、更新データに対応するセカンダリストレージに保存するステップと、

(b) 前記契機に応答して、前記セカンダリストレージデータを一つ以上の対応するバーチャルストレージデータスペースにレプリケートし、それにより前記セカンダリストレージが該一つ以上のバーチャルストレージデータスペースにレストア(“ロールバック”)されることを可能とするステップと、
から成ることを特徴とする方法。

【請求項 2】

前記ストレージデバイスは、ディスクアレイと多重アクセスストレージデバイスの少なくとも一つを含むことを特徴とする請求項 1 の方法。

【請求項 3】

前記プライマリストレージ、セカンダリストレージ及びバーチャルストレージデータスペースは、プライマリボリューム、セカンダリボリューム及び一つ以上のバーチャルボリュームを含むことを特徴とする請求項 1 の方法。

【請求項 4】

前記ディスクアレイは、“RAID”(Redundant Array of Independent Disks)及び“JBOD”(Just a Bunch of Disks)の少なくとも一つとして構成可能であることを特徴とする請求項 2 の方法。

【請求項 5】

前記一つ以上の契機は、前記プライマリストレージに対応するデータアクセスコマンド、前記セカンダリストレージに対応するデータアクセスコマンド、及びデータレプリケーションコマンドの少なくとも一つを含むことを特徴とする請求項 1 の方法。

【請求項 6】

前記データレプリケーションコマンドは、チェックポイントコマンドを含むことを特徴とする請求項 5 の方法。

【請求項 7】

前記一つ以上の契機は、少なくともその一部が前記ストレージデバイスによって実行されるアプリケーションを開始するコマンドを含むことを特徴とする請求項 1 の方法。

【請求項 8】

前記アプリケーションは、データバックアップ、ソフトウェアテスト、及びバッチ処理の少なくとも一つを含むことを特徴とする請求項 7 の方法。

【請求項 9】

前記受信は、前記ストレージデバイスの中で、アレイコントローラに向けられたコマンドをバーチャルボリュームマネージャがモニタすることにより行われることを特徴とする請求項 1 の方法。

【請求項 10】

前記レプリケートは、前記ストレージデバイスの中で、バーチャルボリュームマネージャがアレイコントローラに前記セカンダリストレージのデータをコピーさせることにより行われることを特徴とする請求項 1 の方法。

【請求項 11】

前記ストレージデバイスが一つ以上の第二の契機を受信するステップと、前記バーチャルストレージデータスペースの少なくとも一つを前記セカンダリストレージにレプリケートするステップと
をさらに含むことを特徴とする請求項 1 の方法。

【請求項 12】

前記レプリケートは、データバックアップアプリケーションに関連してバックアップされるべきデータを保存し、それにより前記セカンダリストレージのデータを検証可能とすることを特徴とする請求項 1 の方法。

10

20

30

40

50

【請求項 13】

前記レプリケートは、前記セカンダリストレージに保存されているソフトウェアプログラムのソフトウェアテストと関連して、ソフトウェアプログラム及び環境の少なくとも一つを保存することを特徴とする請求項 1 の方法。

【請求項 14】

前記レプリケートは、セカンダリストレージデータのバッチ処理と関連して少なくとも一つのバッチサブプロセスの更新データを保存することを特徴とする請求項 1 の方法。

【請求項 15】

前記ステップ (b) のレプリケートするステップを、
対応するデータアクセスが十分可能なセキュリティ状態にあることを表示子が示しているか判定するステップと、 10

可能な状態であれば、前記契機に応答して、前記セカンダリストレージのデータを一つ以上の対応するバーチャルストレージデータスペースにレプリケートし、それにより前記セカンダリストレージが該一つ以上のバーチャルストレージデータスペースにレストア (“ロールバック”) されることを可能とするステップと、
に置き換えることを特徴とする請求項 1 の方法。

【請求項 16】

(a) ストレージデバイスが一つ以上の第一の契機を受信して、ソースデータをプライマリストレージに、更新データに対応するセカンダリストレージに保存する手段と、

(b) 前記契機に応答して、前記セカンダリストレージデータを一つ以上の対応するバーチャルストレージデータスペースにレプリケートし、それにより前記セカンダリストレージが該一つ以上のバーチャルストレージデータスペースにレストア (“ロールバック”) されることを可能とする手段と、 20
を備えることを特徴とするシステム。

【請求項 17】

前記ストレージデバイスは、ディスクアレイと多重アクセスストレージデバイスの少なくとも一つを含むことを特徴とする請求項 16 のシステム。

【請求項 18】

前記プライマリストレージ、セカンダリストレージ及びバーチャルストレージデータスペースは、プライマリボリューム、セカンダリボリューム及び一つ以上のバーチャルボリュームを含むことを特徴とする請求項 16 のシステム。 30

【請求項 19】

前記一つ以上の契機は、前記プライマリストレージに対応するデータアクセスコマンド、前記セカンダリストレージに対応するデータアクセスコマンド、及びデータレプリケーションコマンドの少なくとも一つを含むことを特徴とする請求項 16 のシステム。

【請求項 20】

前記データレプリケーションコマンドは、チェックポイントコマンドを含むことを特徴とする請求項 19 のシステム。

【請求項 21】

前記一つ以上の契機は、少なくともその一部が前記ストレージデバイスによって実行されるアプリケーションを開始するコマンドを含むことを特徴とする請求項 16 のシステム。 40

【請求項 22】

前記受信する手段は、前記ストレージデバイスの中で、アレイコントローラに向けられたコマンドをバーチャルボリュームマネージャがモニタすることで与えられることを特徴とする請求項 16 のシステム。

【請求項 23】

前記レプリケートする手段は、前記ストレージデバイスの中で、バーチャルボリュームマネージャがアレイコントローラに前記セカンダリストレージのデータをコピーさせることで与えられることを特徴とする請求項 16 のシステム。 50

【請求項 24】

前記ストレージデバイスが一つ以上の第二の契機を受信する手段と、前記バーチャルストレージデータスペースの少なくとも一つを前記セカンダリストレージにレプリケートする手段と、
を備えることを特徴とする請求項 16 のシステム。

【請求項 25】

前記レプリケートする手段は、データバックアップアプリケーションに関連してバックアップされるべきデータを保存し、それにより前記セカンダリストレージのデータを検証可能とすることを特徴とする請求項 16 のシステム。

【請求項 26】

前記レプリケートする手段は、前記セカンダリストレージに保存されているソフトウェアプログラムのソフトウェアテストに関連して、ソフトウェアプログラム及び環境の少なくとも一つを保存することを特徴とする請求項 16 のシステム。

【請求項 27】

前記レプリケートする手段は、セカンダリストレージデータのバッチ処理に関連して少なくとも一つのバッチサブプロセスの更新データを保存することを特徴とする請求項 16 のシステム。

【請求項 28】

(a) ストレージデバイスが一つ以上の第一の契機を受信して、ソースデータをプライマリストレージに、更新データに対応するセカンダリストレージに保存するステップと、

(b) 前記契機に応答して、前記セカンダリストレージデータを一つ以上の対応するバーチャルストレージデータスペースにレプリケートし、それにより前記セカンダリストレージが該一つ以上のバーチャルストレージデータスペースにレストア(“ロールバック”)されることを可能とするステップと、
をコンピュータに実行させることを特徴とするコンピュータ保存プログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は一般的にコンピュータシステムに係わり、特にデータの複製即ちレプリケーションを行う方法とシステムに関連する。

【背景技術】**【0002】**

今日のエンタープライズコンピューティングシステムは、多重アクセス要求に対応できるものの、小型コンピューティングシステムが行ってきた方法と同じ動作を踏襲している。

【0003】

例えば、図 1 に示すエンタープライズコンピューティングシステムでは、ストレージ 102 に格納されたデータにアクセスするアプリケーションプログラムを実行するアプリケーションサーバ 101 を有する。

【0004】

ストレージ 102 は更に、例えば“RAID”(Redundant Array of Independent Disk)として構成されるディスクアレイを有する。

【0005】

ディスクアレイ 102 は、アプリケーションサーバ 101 がデータボリューム 122 a ~ c、123 に発行するリード/ライト要求を、以下に記述する方法で実行するアレイコントローラ 121 を有する。

【0006】

このようなデータアクセスの実行の為に、アレイコントローラ 121 はデータルーティング、キャッシング、冗長制御、パリティチェック等の制御動作を行う。

【0007】

10

20

30

40

50

本アレイコントローラ 1 2 1 は、データ発生サーバ 1 1 1 a ~ c が記録したデータボリュームを、一つ以上のデータ利用アプリケーションサーバ 1 1 2 で稼動しているアプリケーションが独立に利用できる様に、データボリュームアクセスを実行する。

【 0 0 0 8 】

本アレイコントローラ 1 2 1 は、データ発生サーバ 1 1 1 a から要求された “ オリジナル ” データをオリジナル又は “ プライマリ ” ボリュームに書き込む。

【 0 0 0 9 】

次に、本アレイコントローラ 1 2 1 は、データ利用サーバ 1 1 2 からの最初のデータ要求に対して、オリジナルボリュームのデータを “ セカンダリ ” ボリューム 1 2 3 にコピーし、その後のサーバ 1 1 2 からのデータ更新要求では、同じセカンダリボリュームが専ら用いられる。 10

【 0 0 1 0 】

この結果、対応するプライマリボリュームはサーバ 1 1 2 からは更新されず、他のデータ利用アプリケーションサーバからも同様に使用できる。

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 1 1 】

このような多重アクセス構成は、利用は拡大されているものの、格納データの利用法次第では不効率である事がわかってきた。

【 0 0 1 2 】

例えば、慣用的なデータバックアップでは、ローカルデータからレストア可能なバックアップコピーを生成する。システム 1 0 0 では、アプリケーションサーバ 1 1 1 はプライマリボリューム 1 2 2 a にデータベースを生成して格納する。 20

【 0 0 1 3 】

バックアップサーバ、例えばサーバ 1 1 2 はデータベースのデータをデバイス 1 1 3 (例えばテープバックアップ) にバックアップすることを要求し、データをセカンダリボリューム 1 2 3 にコピーさせる。

【 0 0 1 4 】

然しながら、プライマリボリュームからのコピー中にも、データベースは使用されるために、セカンダリボリュームを検証することが望まれる。不運にも、検証中にもセカンダリデータが入れ換えられることはかなり起るので、セカンダリボリュームをバックアップしたデータに不整合が生じる。 30

【 0 0 1 5 】

ソフトウェアのテストでは、対象ソフトウェアは指定条件に従ったテスト環境でテストされる。

【 0 0 1 6 】

ここで、アプリケーションサーバ 1 0 1 は、ソフトウェアをプライマリボリューム 1 2 2 a に格納する開発システム 1 1 1 a、テスト環境をプライマリボリューム 1 2 2 b に格納する環境生成器 1 1 1 b、テスト条件をプライマリボリューム 1 2 2 c に格納する条件生成器 1 1 1 c、及びテスト 1 1 2 の構成となり、テストはアレイコントローラ 1 2 1 にソフトウェア、環境および条件をセカンダリボリューム 1 2 3 にコピーさせて、テストに使用できるようにする。テスト中は、テストにより継続的に生じる更新環境が、セカンダリボリューム中の環境を置き換える結果になる。 40

【 0 0 1 7 】

かくして、途中でテストに失敗すると、大量のテストコンポーネントはしばしばリモートソースから再ロードすることが必要になる。

【 0 0 1 8 】

バッチ処理に於いては、対象データセットはサブプロセスにより継続的に更新される。ここで、アプリケーションサーバ 1 0 1 はデータベースをプライマリボリューム 1 2 2 a に格納するデータベース生成サーバ 1 1 1、データベースをセカンダリボリュームにコピ 50

ーさせるバッチプロセッサ 1 1 2 の構成となる。

【 0 0 1 9 】

バッチ処理では、セカンダリボリュームデータを各サブプロセスによって継続的に更新して、全ての完了プロセスの合成結果を反映させる事になる。不運にも、サブプロセスが途中でエラーになると、ソースデータはそのソースから再ロードする事が余儀なくされる。

【 0 0 2 0 】

現在のこれらアプリケーションを実行する構成では、格納データが格納アプリケーションによりアクセス可能で、最終結果が使用できるのみで、途中の特定結果データをチェックする機能が用意されていない問題点も存在する。したがって、更なる問題点として、データの正常性が容易に確認できない問題がある。

【 0 0 2 1 】

したがって、多重アクセス可能なデータについて、現行システムに於けるデータの再ロードその他の不都合を解決するシステムと方法が必要である。更に又、生成データの特殊処理をしたいアプリケーションに役立つシステムと方法が必要である。

【課題を解決するための手段】

【 0 0 2 2 】

本発明は、現行システムの欠点を解消したデータの多重アクセスを可能にする。本発明は更に、アプリケーションでの中間データの再使用を可能にする。

【 0 0 2 3 】

本発明は、各種のアプリケーションの走行結果として生じた最終データのみならず、更に中間データのアーカイブ、レストア、再使用又はそのデータの管理を可能にする。

【 0 0 2 4 】

本発明の一実施例では、プライマリ又はセカンダリボリュームに加えて、中間的に生成されたデータが、一つ以上のバーチャルボリュームに選択的に保存、抽出できるように管理される。

【 0 0 2 5 】

他の実施例では、バーチャルボリュームの管理はアプリケーション/ユーザコントロールによって、自動的に即ちプログラマ的に行われ、更にディスクアレイ等のマストレージによって管理される。

【 0 0 2 6 】

本発明でのレプリケーション方法では、ストレージデバイスは一つ以上の第一の契機を受信して、ソースデータをプライマリストレージに、更新データに対応するセカンダリストレージに格納する。

【 0 0 2 7 】

本発明のレプリケーション方法では更に、本契機に応答してセカンダリストレージデータに対応するバーチャルデータスペースにレプリケートし、それによって本セカンダリストレージが本一つ以上のバーチャルデータスペースにレストア(“ロールバック”)されることを可能にする。

【発明の効果】

【 0 0 2 8 】

本発明の効果として、継続する更新データで置き換えられてしまうセカンダリストレージデータを、時刻指定して途中結果を更に保存できることがあげられる。

【 0 0 2 9 】

本発明は更に、一つ以上の保存データセットに対して選択的に保存、抽出ができるようにし、これは更に単純なコマンドを用いて又は現存ストレージデバイスと共に達成可能になる。その他の効果については、後続する明細書と図面によって明確になる。

【発明を実施するための最良の形態】

【 0 0 3 0 】

本発明のレプリケーションシステム及び方法では、アプリケーションデータを、その中

10

20

30

40

50

間生成データを含めて“ R A I D ” (Redundant Array of Independent Disk)等のディスクアレイを含めたストレージデバイスに保存できるようにする。

【 0 0 3 1 】

本発明では、更にデータへの限定的又は選択的アクセスを可能にし、以前の結果データの再構築を可能にし、セキュリティ又はサーバベースのアプリケーションの実行が、ストレージデバイスレベルで可能になる。

【 0 0 3 2 】

ここでは、“ 又は ” は特に注記されない限り、“ 及び / 又は ” の意味で使用されることに注意願いたい。

【 0 0 3 3 】

更にアプリケーションサーバは“ 発生 ” 又は“ 更新 ” サーバとして参照され、レプリケーション、ロールバック等のシステム / 処理は、特定のデバイス又はデバイスタイプに適用可能とする事で本発明の理解を容易にしている。

【 0 0 3 4 】

然しながら、サーバや他のデバイスは多様な又は多重の処理を実施可能であり、又データを生成し処理できる事にも注意願いたい。本発明は具体的な実装要求に従って置換可能な多様なデバイス又はデバイスタイプに適用可能である。以上の用語は限定的なものと看做してならない。

【 0 0 3 5 】

図 2 に、ネットワーク 2 1 3 , 2 0 2 に結合した一つ以上のコンピューティングデバイスより構成される、データレプリケーション可能なシステム結合が示されている。

【 0 0 3 6 】

レプリケーションシステム 2 0 0 は、イントラネット 2 1 3 で結合したデバイス群 2 0 1 を含み、本デバイス群にはレプリケーション可能ディスクアレイ 2 1 1、アプリケーションサーバ 2 1 2、2 1 4 a ~ b、2 1 5 a ~ b 及びネットワークサーバ 2 1 6 を含む。

【 0 0 3 7 】

システム 2 0 0 は更に、具体的実装要求に従い、同様に結合したアプリケーションサーバ 2 0 3、その他コンピューティングシステム 2 0 4、更には一つ以上のファイヤウォール 2 1 7、ルータ、キャッシュ、冗長 / 負荷バランスシステム、バックアップシステム又はその他のネットワーク要素等(図示していない)が含まれる。

【 0 0 3 8 】

データレプリケーションはシステム 2 0 0 中の典型的には R A I D 等のディスクアレイ 2 1 1 で実行される。(レプリケーション可能デバイスは、一般的には本発明での説明に矛盾しない方法でデータレプリケーションを実行できる単機能または多機能のストレージデバイスで構成できることに注意願いたい。)

ディスクアレイ 2 1 1 はアレイコントローラ 2 1 1 a、バーチャルボリュームマネージャ 2 1 1 b 及びストレージ媒体のアレイ 2 1 1 c を有する。

【 0 0 3 9 】

ディスクアレイ 2 1 1 は更に、具体的実装に従ってキャッシュ機能、冗長制御、パリティ検査その他のストレージ又はサポート機能(図示していない)を持つその他のコンポーネント 2 1 1 d を有する。それらのコンポーネントは慣用的ディスクアレイその他のストレージデバイスで使用されるものでよく、その構成も具体的要求に従った慣用的なものでよい。

【 0 0 4 0 】

アレイコントローラ 2 1 1 a は一般的にディスクアレイの管理を行う。そのような管理には、システム 2 0 0 の他のデバイスと通信して、ストレージ媒体 2 1 1 c とのデータの保存、抽出を行うのに必要な制御機能が含まれる。(アレイコントローラ 2 1 1 a のコンポーネントと動作は、以下の説明又は具体的実装に従って構成の修正を受ける以外は慣用的方法で提供される。)

アレイコントローラ 2 1 1 a は、例えばファイル、フォルダ、ディレクトリ、ボリュ

10

20

30

40

50

ーム等のデータスペース関連事項の生成、維持又は削除等のサポート機能を実行する。

【0041】

アレイドントローラ211aは、ソース又は“オリジナル”データを保存するために“オリジナルボリューム”を生成し、多重アクセス機能を可能にする。

【0042】

アレイドントローラ211aは更に、更新アプリケーションサーバの為にアプリケーションデータスペース即ち“セカンダリボリューム”を生成して、対応するプライマリボリュームからデータをコピーする。

その後、アレイドントローラ211aは、典型的には同じセカンダリボリュームに対して、対応するアプリケーションサーバによる継続する更新データの保存要求を受信して、データの保存を行う。 10

【0043】

かくして、オリジナルボリュームは不変のままで、他のアプリケーションサーバからの更新に使用でき、対応するセカンダリボリュームは各アプリケーションによってこれまで継続的に更新され保存されたデータを含むことになる。(今後、こうして更新され保存されたデータを“更新データ”と呼ぶ。)

アレイドントローラ211aは、既に述べた、又はこれに限らず、その他のディスクアレイドコンポーネントも管理する。

【0044】

アレイドントローラ211aは、バーチャルボリュームマネージャ211bと相互作用する多少なりとも統合した構成が可能で、又バーチャルボリュームマネージャ211bは具体的実装に従って独立に動作する構成も可能である。 20

【0045】

より以上に統合化された構成では、アレイドントローラ211aはアプリケーションからのアクセス要求を、例えば、バーチャルボリュームマネージャ211bを通して実行してもよい。

【0046】

アレイドントローラ211aは更にコマンド通訳者になって、バーチャルボリュームマネージャ211bの要求に従って、プライマリ/セカンダリボリュームに実施するのと同様に、バーチャルボリュームに対するデータの保存、抽出その他の動作を実行してもよい。(以下に明らかになるが、バーチャルボリューム管理処理との統合を深めると、より一般的なストレージデバイス制御に重複を排除する事ができるが、これにはトレードオフが存在して、より統合を深くすると、既存のストレージ又はホストデバイス実装との互換性を損なう結果となる。) 30

バーチャルボリュームマネージャ211bは、バーチャルボリュームを生成し、情報を移植し、削除その他の管理を実行し、典型的にはストレージ媒体211c内にバーチャルボリュームデータを選択的に保存し、抽出しその他の管理業務を実行する。

【0047】

バーチャルボリュームは他のボリュームタイプと同様に、ストレージ媒体211c内にデータストレージエリアを定義し、他のシステム要素から参照可能なアプリケーションデータを保存できるようにする。 40

【0048】

然しながら、プライマリ又はセカンダリボリュームとは異なって、バーチャルボリュームの管理ではアプリケーションデータを選択的に保存、抽出をする“スナップショット”が可能で、異なったアプリケーションからのプライマリデータへの単なる多重アクセス以上の使い方が可能になる。オリジナル又は更新データの最終結果のみならず、中間的結果をバーチャルボリュームに保存することが出来る。

【0049】

バーチャルボリュームは更に、自動的に(即ちプログラマ的に)又はアプリケーション、ユーザ選択又はセキュリティ操作その他の具体的アプリケーションに応じて選択的に管 50

理することが出来る。

【 0 0 5 0 】

バーチャルボリュームマネージャ 2 1 1 b はアプリケーションサーバからの要求即ち “ コマンド ” に応じてバーチャルボリュームの管理を行う。それらのコマンドはアレイコントローラ 2 1 1 a のコマンドと同様に構成され、広範な互換性をもたらしている。

【 0 0 5 1 】

バーチャルボリュームマネージャ 2 1 1 b はバーチャルボリュームの生成、保存データのバーチャルボリュームへのレプリケーション、バーチャルボリュームからのデータのレストア及びバーチャルボリュームの削除を含む、かなり限定されたコマンドセットに対して応答する。

10

【 0 0 5 2 】

然しながら、処理能力を増大させるとコマンド又は機能範囲は拡大されるが、本明細書でははその一部しか紹介できない事に注意願いたい。

【 0 0 5 3 】

具体的実装次第で、バーチャルボリュームマネージャ 2 1 1 b は、アプリケーションサーバと直接交信する構成も可能であるし、例えばアレイコントローラ 2 1 1 a を通して管理を実行する間接的な実装も可能である

。例えば、より以上に統合化された構成では、バーチャルボリュームマネージャ 2 1 1 b は、アプリケーションサーバからのコマンドを、アレイコントローラ 2 1 1 a を通して間接的に受信し、又本コントローラ 2 1 1 a を通して応答する事も出来る。

20

【 0 0 5 4 】

バーチャルボリュームマネージャ 2 1 1 b はより自主的にアレイコントローラ 2 1 1 a が保存、抽出、その他の動作をするのをモニタし、コマンドをアレイコントローラ 2 1 1 a に（アプリケーションから来たかの如く）与えて、ボリューム生成、データ保存、データ抽出その他のアクセス処理を実行させてもよい。

【 0 0 5 5 】

バーチャルボリュームマネージャ 2 1 1 b は更に、データアクセスに関しては典型的な場合であって、そうでなければ慣用的な方法となるが、キャッシュその他のディスクアレイ 2 1 1 のコンポーネントを利用する事が出来る。（アレイコントローラ 2 1 1 a 又はバーチャルボリュームマネージャ 2 1 1 b には、定常的にもダイナミックにも、実装又はアプリケーションに対応した別の選択肢が存在する事に注意願いたい。）

30

残りのディスクアレイ 2 1 1 のコンポーネントについては、ストレージ媒体 2 1 1 c はデータ保存可能な物理媒体で、ハードディスク、書き換え可能光、可搬 / 非可搬媒体、キャッシュメモリ、その他アプリケーションが要求する相応しい記憶媒体を含む。その他コンポーネントとして、エラーチェック、キャッシング、その他ストレージやアプリケーション関連のコンポーネントを含む。

【 0 0 5 6 】

アプリケーションサーバ 2 1 4 a ~ b、2 1 5 a ~ b、2 0 3、2 0 4 はユーザ / システムにシステム 2 0 0 内での処理を可能にし、ストレージ 2 1 1 へのデータ保存機能を持つデバイスを含み、アプリケーションに対応してバーチャルボリュームマネージャ 2 1 1 b と連携する。

40

【 0 0 5 7 】

そのようなデバイスには、一つ以上のワークステーション、パーソナルコンピュータ “ P C ”、可搬コンピュータ、セットトップボックス、パーソナルデータアシスタンス “ P D A ”、パーソナル情報端末 “ P I M ”、セルフォン、コントローラ、所謂 “ スマート ” デバイスやその他のエレクトロメカニカルデバイスが含まれる。

【 0 0 5 8 】

その他システム 2 0 0 に残っているコンポーネントとしては、ネットワーク 2 1 3 と 2 0 2 は定常的又は再構築可能な “ L A N ”、“ W A N ”、バーチャルネットワーク V P N、その他アプリケーションに対応した結合が含まれる。

50

【 0 0 5 9 】

ネットワークサーバ 2 1 6 は、ネットワークサーバオペレーション（例えば、ネットワークアクセス、e - M a i l、システム管理等）の為に慣用的に構成された、又はストレージデバイスホストとして動作する、一つ以上のアプリケーションサーバで成り立つ。

【 0 0 6 0 】

図 3 では、システム 2 0 0（図 2）の一つ以上の要素から成る処理システムを示す。他の代替構成が可能かも知れないが、簡単の為にシステム 2 0 0 の要素は、特に注記されない限り、ハードウェア、ソフトウェア又はそれらと整合性ある一つ以上の処理システムの組合せとする。

【 0 0 6 1 】

処理システム 3 0 0 は、一つ以上の汎用又は専用のプロセッサ 3 0 2、例えば P e n t i u m、P o w e r P C、D S P (Digital Signal Processor)等を含めた通信チャネル（例えばバス 3 0 1）に結合した要素で成り立つ。（P e n t i u m、P o w e r P C は登録商標です）

【 0 0 6 2 】

システム 3 0 0 は更に、アプリケーションの要求に対応して、一つ以上の入力装置 3 0 3（マウス、キーボード、マイクロフォン、ペン等）と出力装置 3 0 4（ディスプレイ、スピーカ、アクチュエータ等）を有する。

【 0 0 6 3 】

システム 3 0 0 は更に、ストレージ/メモリデバイス、又はハード又はリムーバブルストレージ/メモリ媒体等のコンピュータ読み取り可能ストレージ媒体 3 0 6 とその読み取り装置 3 0 5 を有する。

【 0 0 6 4 】

それらのデバイス又は媒体は、独立のストレージ 3 0 8 やメモリ 3 0 9 としても使用され、アプリケーションの要求に従ってハードディスク、フロッピー/コンパクトディスク、デジタル多用途ディスク（“ D V D ”）、スマートカード、R O M、ランダムアクセスメモリ、キャッシュメモリ等を含む。モデム、D S L、赤外線又は他の適当なトランシーバ等の一つ以上の通信デバイス 3 0 7 が更に含まれ、直接のデバイス間通信又はプライベート又はパブリックネットワーク（既に述べたものを含み、更にこれに限定される事なく）を介した交信を可能にする。

【 0 0 6 5 】

ワーキングメモリ 3 1 0（メモリ 3 0 9 の要素でもある）は、更にオペレーティングシステム（“ O S ”）3 1 1 とアプリケーションプログラム、モバイルコード、データ等のその他のプログラム 3 1 2 をシステム 2 0 0 の実装要素として、使用中に読み込み保存する。

【 0 0 6 6 】

特に O S は、アプリケーション（例えば、W i n d o w s、M a c、L i n u x、U n i x、P a l m O S、専用 O S 等）（W i n d o w s、U n i x は登録商標です）に対応したデバイスと機能に依存する。多様なプログラム言語又はツールが利用される。更に又、ワーキングメモリ 3 1 0 の内容である、O S 3 1 1 とアプリケーションプログラム 3 1 2 はユーザのアプリケーションによって大幅に影響を受ける事に注意願いたい。

【 0 0 6 7 】

ソフトウェア（例えば、一部又は全体を含めて、アプリケーションプログラム、オブジェクト、エージェント、ダウンロードブル、サブレット等）で実装されている場合は、システム 2 0 0 の要素はローカル又はリモートストレージからメモリ（又はキャッシュメモリ等）に実行する為のデータを一時的又はより永続的に格納させ、或いは他の適当なメカニズムが使用され、要素はコンパイル状態かインタープリタ方式で実行される

。入力された中間的又は最終的データ又は機能要素は、より一時的又はより永続的にストレージ媒体、キャッシュ又は他の揮発性、不揮発性メモリ（例えばストレージデバイス 3 0 8、メモリ 3 0 9）にアプリケーションに対応して保存される。

10

20

30

40

50

【 0 0 6 8 】

図 4 で、専用ホスト又は他のアプリケーションサーバと共にレプリケーション可能ストレージデバイスを使用して、如何にデータレプリケーションが実行されるかを説明する。図 4 は更に、プロセッサベースのアレイコントローラとバーチャルボリュームマネージャがより以上に統合したアレイマネージャ 4 0 3 を示す。

【 0 0 6 9 】

図示するように、レプリケーションシステム 4 0 0 は、ホスト 4 0 1 とストレージデバイス 4 0 2 及びネットワーク 4 0 6 を含む。

【 0 0 7 0 】

ホスト 4 0 1 は、図 3 のシステム 3 0 0 に対応し、簡単の為に大幅に簡略化して表示しており、他方同様に図 3 のシステム 3 0 0 に対応するプロセッサベースのストレージデバイス 4 0 2 は大いに詳細に示されている。 10

【 0 0 7 1 】

ホスト 4 0 1 は、I/O インターフェース 4 1 1 と 4 3 1 の結合 4 a を通して、ストレージデバイス 4 0 2 に結合し要求を発行する。

【 0 0 7 2 】

結合 4 a は、例えば、“S C S I” (Small Computer System Interface) インターフェース、ファイバチャネル、“E S C O N” (Enterprise System Connection)、“F I C O N” (Fiber Connectivity)、イーサネット (イーサネットは登録商標です) を含み、一つ以上のプロトコル S C S I、i S C S I、E S C O N、ファイバ F I C O N 等を使用する。 20

【 0 0 7 3 】

ホスト 4 0 1 とストレージデバイス 4 0 2 は更に各々のネットワークインターフェース 4 1 2 と 4 3 2 と結合 4 b と 4 c を通してネットワーク 4 0 6 に結合する。本ネットワークはファイバチャネル、イーサネット、インターネットプロトコル (“I P”)、非同期転送モード (“A T M”) 等のプロトコルで実装される。

【 0 0 7 4 】

本ネットワーク結合は、ホスト 4 0 1 とストレージデバイス 4 0 2 がネットワーク 4 0 6 に結合しているその他のデバイス即ち図 2 のアプリケーションサーバ 2 1 2、2 1 4 a ~ b、2 1 5 a ~ b、2 1 6、2 0 3 及び 2 0 4 等と通信することを可能にする。(インターフェース 4 1 1、4 1 2、4 3 1、4 3 2、4 3 3 及び 4 3 4 は図 3 の通信インターフェース 3 0 7 に対応する。) ストレージデバイス 4 0 2 はインターフェース 4 3 1 ~ 4 3 4 に加えてアレイマネージャ 4 0 3 とストレージ媒体 4 0 4 を有する。 30

【 0 0 7 5 】

アレイマネージャ 4 0 3 内の C P U 4 3 5 は、メモリ 4 0 5 とキャッシュメモリ 4 5 1 内に格納されたコントロール情報 4 5 2 に基づき、内部バス 4 3 6 やその他の図示された相互接続を通して、ストレージコントロールとデータレプリケーションを実行する。

【 0 0 7 6 】

キャッシュメモリ 4 5 1 はホスト 4 0 1 との送受信データを一時的に保管するために使用される。キャッシュメモリ 4 5 1 は更にホストから要求されたシーケンシャルなリード / ライトデータの事前格納の為に使用される。 40

【 0 0 7 7 】

ストレージ媒体 4 0 4 は、I/O インターフェース 4 3 3、4 3 4 と結合 4 f を経由して、アレイマネージャ 4 0 3 に結合し通信する。ストレージ媒体 4 0 4 はアプリケーションからの要求に従い、一つ以上の R A I D、J B O D (Just a Bunch of Disk) として構成される、又はその他の相応しい構成となるディスクドライブ 4 4 1 のアレイを含む。

【 0 0 7 8 】

ストレージ媒体 4 0 4 は内部バス 4 3 6 と結合 4 d ~ 4 f を通して C P U 4 3 5 に固有に結合し、本 C P U 4 3 5 は本ディスクの部分ボリューム (例えば、プライマリ、セカンダリ、バーチャルボリューム) として管理し、ホスト 4 0 1 からストレージ媒体への 50

アクセスは本ボリュームを通してのみ（即ち、物理媒体を直接アドレスするのではなく）可能となる。CPU 435は更に既に述べたセキュリティ、アプリケーション又は実装によって必要になる機能を実行する。

【0079】

図5は本発明に従う、エンタープライズ又はその他一般システムに適応する動作特性に従うデータレプリケーションシステム例を説明するフローダイアグラムである。システム500はアプリケーションサーバ501とディスクアレイ502で構成される。

【0080】

アプリケーションサーバ501は更に、データ生成アプリケーションサーバ511a～b、データ更新アプリケーションサーバ512a～b及びその他のデバイス513を有し、ディスクアレイ502は、更に、アレイマネージャ502a、ストレージ媒体502b、及びネットワーク又は入出力インターフェース（“I/O”）502c（例えば、図4のインターフェース431、432に該当する）を有する。

【0081】

アレイマネージャ502aは、ディスクアレイコントローラ521aとバーチャルボリュームマネージャ521bを含み、ストレージ媒体502bは、プライマリボリューム522a～b、セカンダリボリューム523a～b及びバーチャルボリューム524a～dの各々一式以上を有する。

【0082】

明確にするために、システム500内の信号パスは実線の矢印で描き、コンポーネント間の主要なデータの流れは点線の矢印で示す。

【0083】

加えて、アプリケーションサーバ501は、現在の例の目的に対して、他のサーバが利用する為のオリジナルデータを供給するか（オリジナルアプリケーションサーバ1～M、511a～b）、又は他のアプリケーションサーバから供給されたデータを利用する（更新アプリケーションサーバ1～N、512a～b）かのいずれかを専ら行う。

【0084】

各アプリケーションサーバ511a～b、512a～bは、データアクセスを要求する“コマンド”をI/Oバス502cを経由してアレイマネージャ502aに送信する。

【0085】

オリジナルアプリケーションサーバ511a～bのアプリケーションがデータ書き込み要求（“データライト”）をアレイコントローラ521aに発行し、アレイコントローラ521aはプライマリボリューム、例えば522aを生成し（未だ生成されていなかったら）、オリジナルデータを本プライマリボリュームに書き込む。更新アプリケーションサーバはデータ保存コマンドを発行し、アレイコントローラ521aはこれによりセカンダリボリューム、例えば523aを生成（まだ生成されていなかったら）して、プライマリボリュームデータをコピーして、含まれる更新データを書き込む。

【0086】

更新アプリケーションサーバ512a～bのアプリケーションは更にデータ読み出し要求（“データリード”）をアレイコントローラ521aに発行し、アレイコントローラ521aは、要求データを読み出して要求アプリケーションサーバにI/O 502cを通して返送する。（本事例では、一つのデータセットには唯一つのプライマリボリュームが生成され、各アプリケーションに対して唯一つのセカンダリボリュームが生成される。）

オリジナルアプリケーションサーバ511a～bは一般的には、バーチャルボリュームマネージャ521bとは交信する必要はない。プライマリボリュームのデータは、アレイコントローラによるコピーを通して一つ以上のセカンダリボリューム523a～b上でバーチャルボリュームマネージャ521bにより利用できるため、一般に使用される一つ以上のプライマリボリューム522a～bはバーチャルボリュームマネージャ521bに結合する必要はない

かくして、実装上特別な要求が発生しない限り、システム500はディスクアレイ50

2 に、上記した機能を持たせる事無く簡略に構成することが出来る。

【0087】

更新アプリケーションサーバ512a~bは本事例では、チェックポイント、ロールバック、及びバーチャルボリューム削除を含む限定されたバーチャルボリュームコマンドを発行できる。

【0088】

アプリケーションがチェックポイントコマンドを発行すると、バーチャルボリュームマネージャ521bはバーチャルボリューム1-1、524aを生成し、本バーチャルボリュームに要求アプリケーションに対応するセカンダリボリューム-1、523aの現データをレプリケートする

10

。同じアプリケーションが更にチェックポイントコマンドを発行すると、バーチャルボリュームマネージャ521bは同様に要求アプリケーションに対応するバーチャルボリューム1-y、524bを生成して、同じセカンダリボリュームの現データをレプリケートする。

【0089】

然しながら、異なったアプリケーションから更にチェックポイントコマンドが発行されると、バーチャルボリュームマネージャ521bは本異なったアプリケーションに対応するバーチャルボリュームx'-1、524cを生成し、本異なったアプリケーションに対応するセカンダリボリュームx'、523bの現データを本バーチャルボリュームにレプリケートする。

20

【0090】

アプリケーションがロールバックコマンドを発行すると、バーチャルボリュームマネージャ521bは、コマンドで指定するバーチャルボリューム1-1、524aのデータを要求アプリケーションに対応するセカンダリボリューム-1、523aにレストアする(取り戻す)。

【0091】

更に同じアプリケーションからロールバックコマンドを発行すると、バーチャルボリュームマネージャ521bは、同様にコマンドで指定するバーチャルボリューム1-y、524bのデータを要求アプリケーションに対応するセカンダリボリューム-1、523aに取り戻す。

30

【0092】

更に異なったアプリケーションがロールバックコマンドを発行すると、バーチャルボリュームマネージャ521bは、同様にコマンドで指定するバーチャルボリュームx'-1、524cのデータを要求アプリケーションに対応するセカンダリボリュームx'、523bに取り戻す。

【0093】

最後に、アプリケーションがバーチャルボリューム削除コマンドを発行すると、バーチャルボリュームマネージャ521bはコマンドで指定されたバーチャルボリュームを削除する。(チェックポイント、ロールバック及び削除コマンドの例は図6で示される。)

40

アレイマネージャ502a内のアレイコントローラ521aが、オリジナルサーバ511a~bからデータ保存コマンドを受け取ると、オリジナルサーバに対応するプライマリボリューム(例: 522a、もし未だ存在しなければ)を生成し、含まれているデータを本ボリュームに保存する。アレイコントローラ521aは更新サーバ512a~bからデータ保存コマンドを受け取ると、本アプリケーションに対応するセカンダリボリューム523aを今後使用できるように生成しプライマリボリュームのデータをコピーして、本サーバからのデータを本セカンダリボリュームに保存する。

【0094】

アレイコントローラ521aは更に本アプリケーションからの継続する保存コマンドに
応答して、本セカンダリボリュームのデータを本アプリケーションから供給される更新デ

50

ータで置き換える。アレイコントローラ 5 2 1 a はアプリケーションサーバ 5 0 1 から、読み取りコマンドを受け取るとアプリケーションサーバにコマンドで指定されるプライマリ又はセカンダリボリュームからのデータを返送し、又削除コマンドを受け取ると指定されるプライマリ又はセカンダリボリュームを削除する。

【 0 0 9 5 】

現構成では、アレイコントローラ 5 2 1 a はバーチャルボリュームコマンドをアプリケーションサーバ 5 0 1 から受信するが無視する。然しながら、アレイコントローラ 5 2 1 a はバーチャルボリュームマネージャ 5 2 1 b から結合 5 b ~ 5 c を通してデータアクセスコマンドを受け取り応答する。

【 0 0 9 6 】

より具体的には、バーチャルボリュームマネージャ 5 2 1 b はチェックポイントコマンドに対して、対応するセカンダリボリュームを決めて、コマンド発行アプリケーションサーバに対応してバーチャルボリュームを割り当て、今後の使用に備えてコマンド発行アプリケーションサーバとボリュームの対応関係を保存する。

【 0 0 9 7 】

バーチャルボリュームマネージャ 5 2 1 b は更に本バーチャルボリュームに対してタイムスタンプ（日時も含む）をアサインして再度今後の使用に備えて、本表示子に対応情報と共に保存する。

【 0 0 9 8 】

ここで言う将来使用には、例えば、同じアプリケーションからのロールバックや削除コマンド及び同じ又は他のアプリケーションサーバに対するバーチャルボリュームの割り当てが該当する。

【 0 0 9 9 】

バーチャルボリュームマネージャ 5 2 1 b は更にアレイコントローラ 5 2 1 a にセカンダリボリューム（例：セカンダリボリューム - 1、5 2 3 a）からバーチャルボリューム（例：バーチャルボリューム 1 - 1、5 2 4 a）へのコピーコマンドを発行する。（例えば、バーチャルボリュームマネージャ 5 2 1 b がアレイコントローラ 5 2 1 a にセカンダリボリュームからのデータ読み取りを依頼して、更にバーチャルボリュームへの書き込みを依頼しても同一結果になることに注意願いたい。）

アレイコントローラ 5 2 1 a は上記の読み取りと保存シーケンスやコピーコマンドを、バーチャルボリュームをセカンダリボリュームだと見なして、恰もアプリケーションサーバから発行されたように扱う。

【 0 1 0 0 】

即ち、アレイコントローラ 5 2 1 a は最初にディスクコントローラ（図 7 a を参照）にペアスプリット要求を発行しアレイコントローラ 5 2 1 a がアクセス中にセカンダリボリュームがプライマリボリュームの新データで更新されるのを防止する。

【 0 1 0 1 】

アレイコントローラ 5 2 1 a は次いで、新しいバーチャルボリューム（例：バーチャルボリューム 1 - 1、5 2 4 a）をセカンダリボリュームだと見なして生成して、セカンダリボリュームデータをバーチャルボリュームにコピーして、同期要求を発行して対応するプライマリボリュームのセカンダリボリュームに対するリンクを回復する。

【 0 1 0 2 】

バーチャルボリュームマネージャ 5 2 1 b はロールバックコマンドに対して、コマンド発行アプリケーションサーバに対応してセカンダリボリュームとバーチャルボリュームを決定する。

【 0 1 0 3 】

この例ではセカンダリボリュームはコマンドを参照するだけでは決定できない。更にバーチャルボリューム表示子はコマンドにより提供されるが、本表示子は日時情報や I D でアレイコントローラ 5 2 1 a は認識する事が出来ない。

【 0 1 0 4 】

10

20

30

40

50

従って、バーチャルボリュームマネージャ 5 2 1 b は、アレイコントローラ 5 2 1 a が使用するセカンダリボリュームとバーチャルボリュームの対応関係を、前回のチェックポイントコマンド中にセカンダリボリュームに保存された表示子を参照して決定する。

【0 1 0 5】

バーチャルボリュームマネージャ 5 2 1 b は更にアレイコントローラ 5 2 1 a に対して、決定されたバーチャルボリューム (例：バーチャルボリューム 1 - 1、5 2 4 a) から、セカンダリボリューム (例：セカンダリボリューム - 1、5 2 3 a) へのコピーコマンドを発行する。(バーチャルボリュームマネージャ 5 2 1 b がアレイコントローラ 5 2 1 a に対して、本バーチャルボリュームを指定して読み取り依頼をし、本セカンダリボリュームに保存依頼を発行しても同じ結果になることに注意。)

10

アレイコントローラ 5 2 1 a は本コピーコマンドに対して、恰もアプリケーションサーバから読み取りと保存のコマンドシーケンス又はコピーコマンドを受信したかの如く見なして、バーチャルボリュームをセカンダリボリュームとして扱った場合と同様に動作する。

【0 1 0 6】

即ちアレイコントローラ 5 2 1 a は最初にディスクコントローラにペアスプリット要求を発行しアレイコントローラ 5 2 1 a がアクセス中にセカンダリボリュームがプライマリボリュームの新データで更新されるのを防止する。

【0 1 0 7】

アレイコントローラ 5 2 1 a は次いで、バーチャルボリュームデータをセカンダリボリュームにコピーして、同期要求を発行して対応するプライマリボリュームとセカンダリボリュームのリンクを回復する。

20

【0 1 0 8】

バーチャルボリュームマネージャ 5 2 1 b はバーチャルボリューム削除コマンドに対して、コマンド発行アプリケーションサーバに対応するバーチャルボリュームを決定する。この場合も、コマンドにより提供されるバーチャルボリューム表示子は、バーチャルボリュームマネージャ 5 2 1 b によってアサインされた、日時表示又は ID で、アレイコントローラ 5 2 1 a には再度認識できないものである。

【0 1 0 9】

従って、バーチャルボリュームマネージャ 5 2 1 b は、前回のチェックポイント中に保存された表示子を参照して、アレイコントローラ 5 2 1 a が使用できる、対応するセカンダリボリュームを決定する (図 7 a ~ c を参照)。

30

【0 1 1 0】

更にバーチャルボリュームマネージャ 5 2 1 b は、バーチャルボリューム (例：1 - 1、5 2 4 a) を削除されるべきセカンダリボリュームと見なして、セカンダリボリュームの削除コマンドをアレイコントローラ 5 2 1 a に発行する。アレイコントローラ 5 2 1 a は本削除コマンドを、恰もアプリケーションサーバから受信したセカンダリボリューム削除コマンドと見なして、対応する。

【0 1 1 1】

アレイコントローラ 5 2 1 a は最初にペアスプリット要求を発行して、セカンダリボリュームがプライマリボリュームの新データで更新されるのを防止する。アレイコントローラ 5 2 1 a は、バーチャルボリュームマネージャ 5 2 1 b からの削除コマンドで指示されたバーチャルボリュームを削除する。

40

【0 1 1 2】

例えば、ストレージ媒体 5 0 2 b 内にバーチャルボリューム用のデータスペースを確保して、バーチャルボリュームマネージャ 5 2 1 b がコピーや削除動作を直接実行するより浅い統合方式も可能である事に注意する必要がある。

【0 1 1 3】

例えば、バーチャルボリュームマネージャ 5 2 1 b は、チェックポイントコマンドに対して、対応するセカンダリボリュームを決定し、コマンド発行アプリケーションと時刻

50

ノ日付その他情報に対応する指定バーチャルボリューム名、タイムスタンプその他表示子をアサインして、コマンド発行アプリケーション及びボリュームとの対応関係やアサイン内容を保存する。

【0114】

バーチャルボリュームマネージャ521bは、次いでペアスプリットコマンド（実装に依存する）とセカンダリボリューム（例：セカンダリボリューム-1、523a）からバーチャルボリューム（例：バーチャルボリューム1-1、524a）へのコピーコマンドを発行し、更に同期コマンドを発行する（図7a参照）。

【0115】

定常的にデータスペースをアサインしておく方法は、ストレージスペースを浪費する可能性があり、必要によりダイナミックにアサインする方法は複雑さを増大させる。 10

【0116】

然しながら、本方式ではバーチャルボリュームマネージャ521bに対して高度のバーチャルボリュームのコントロールを可能にして、例えば、アレイコントローラ521aには手に負えないセキュリティ機能等の追加機能をアレイコントローラ521aにインパクトを与えないで実装する事が可能になる。

【0117】

他の機能も具体的アプリケーションに対応して同様に実装する事が出来る。（この他にアレイコントローラとバーチャルボリュームマネージャが共有スペースを使用するメカニズムも同様に使用することができるが、上記のメカニズムは同様な機能性をより単純なシステムで実現する様に見える。） 20

図7a～7cはアレイコントローラ521aとバーチャルボリュームマネージャ521bの内部動作をより詳細に説明する。

【0118】

図7aにて、アレイコントローラ521aは、アレイエンジン701を有し、本エンジンは、多重アクセスボリュームマップ702を用いて、プライマリとセカンダリボリュームをアプリケーションサーバと物理媒体に対応して管理する。

【0119】

バーチャルボリュームマネージャ521bはバーチャルボリュームエンジン703を有し、本エンジンは、バーチャルボリュームマップ704を用いて、更にオプションとして、セキュリティマップ705を用いてバーチャルボリュームの管理動作を実行する。バーチャルボリュームマネージャ521bは更に、時刻と日付情報を保有する日時ソースへの結合7aを有する。 30

【0120】

各チェックポイント、ロールバック及び削除コマンドに対して、バーチャルボリュームマネージャ521bはバーチャルボリュームマップ704を構築維持しながら、対応する参照を決定する。

【0121】

図7bのバーチャルボリューム（及びセキュリティ）マップについては、物理媒体とボリュームのマッピングは、プライマリとセカンダリボリュームの対応及びアクセスするアプリケーションサーバとの対応を含めて、具体的なアレイコントローラの実装にしたがって維持される。 40

【0122】

図7cのアクセスボリュームマップについては、バーチャルボリュームマネージャ521b（図7a）はコマンドを実行する前に本マップ（基本的マップは起動時に参照され、以降の変更がバーチャルボリュームコントローラで維持される）を参照する。

【0123】

バーチャルボリュームマネージャ521bは、バーチャルボリュームを参照するためにアレイコントローラ521aによってなされたセカンダリボリュームの割り当てやセカンダリボリューム対応関係を用いてバーチャルボリュームを決定する事が出来る。 50

【 0 1 2 4 】

バーチャルボリュームマネージャ 5 2 1 b は、例えば本対応をマップ 7 0 6 に加えて、更に固有のバーチャルボリューム ID の割り当てをマップ 7 0 6 に加える。

【 0 1 2 5 】

バーチャルボリュームマネージャ 5 2 1 b はかくして、セカンダリボリュームとバーチャルボリュームの対応をこのような複合マップを使用して（あるいは双方のマップを使用して）決定する事が出来る。その他の決定 / 参照メカニズムは、例えば最初に参照した同時係属中の出願明細書に記したように、具体的実装に従って採用する事が出来る。

【 0 1 2 6 】

図 7 b はバーチャルボリュームマネージャ 5 2 1 b がバーチャルボリュームに対する制限、セキュリティその他の機能を如何に実行するかの例を示す。 10

【 0 1 2 7 】

バーチャルボリュームマネージャ 5 2 1 b は、例えば、バーチャルボリューム数、保存するデータ量その他に対する制限を適用可能な動作情報を含むバーチャルボリュームマップを参照する事により設定する。

【 0 1 2 8 】

バーチャルボリュームマップ 7 0 6 には、例えば、対応するアサインされたバーチャルボリュームの指定 7 6 2 a、7 6 2 b（数値名とタイムスタンプ指定を含む）に加えて、アプリケーションサーバ（のアプリケーション）7 6 1 を含む。バーチャルボリュームマップ 7 0 6 は更に、アレイコントローラ参照 7 6 3 及び、本例ではサーバ固有セキュリティパラメータルール / パラメータ 7 6 5 を含む。 20

【 0 1 2 9 】

バーチャルボリュームマネージャ 5 2 1 b は、アプリケーションサーバ、アプリケーション、ユーザその他からのアクセス要求を、これら各要求元対応にアクセス可否を決めた事前設定ルール / パラメータに従って許諾の可否を判定してセキュリティ処理を行う。

【 0 1 3 0 】

このようなアクセス要求には、一般的又は特定データ又はデータタイプその他具体的特性に関連するロールバック又はバーチャルボリュームの削除要求が含まれる。

【 0 1 3 1 】

以下、図 8 ~ 1 1 に示すように、データレプリケーションは広範なアプリケーションに 30
応用する事が出来る。

【 0 1 3 2 】

例えば、図 8 はデータバックアップシステム、図 9 はソフトウェア開発システム、更に図 1 0 はバッチ処理システムで、各ケースともデータレプリケーションを活用している。

【 0 1 3 3 】

図 1 1 はアプリケーション又は自動オペレーションで如何にデータレプリケーションが使用されるかを示す。（図 1 1 では、一貫性を保つためにデータバックアップを例に使用する。）

図 8 から始めると、レプリケートされたデータベース即ちデータベース管理システムのデータ又はその他の対象コード又はデータが多重アクセスシステムに於いて、完全に矛盾 40
なく更新されたことを確認する事は困難である。

【 0 1 3 4 】

従って、データベースのデータバックアップ開始前にデータベースを検証することが望ましい。レプリケーションを使用することによって、他の方法では対象データベースを変更してしまうかもしれない検証作業を、データベースのバックアップの信頼性を維持しながら実行することが可能になる。

【 0 1 3 5 】

例えば、図 8 に於いて、データベースサーバ 8 1 1 は検証サーバ 8 1 2 が検証のために要求するデータベースをオリジナルボリューム 8 2 2 に保存する。

【 0 1 3 6 】

アレイコントローラ 8 2 1 a はペアスプリットを指定して、データベースサーバ 8 1 1 からのデータベースへの変更要求がセカンダリボリューム 8 2 3 へ同期化されるのを防止して、データベースをセカンダリボリューム 8 2 3 にコピーする。

検証実行に伴う問題点は不運にも依然として存在して、検証作業はセカンダリボリューム 8 2 3 のデータベースを更新するかも知れず、コピー後に於いてオリジナルボリューム 8 2 2 のデータベースは変更されたかもしれない。従って、信頼できるデータベースがバックアップの為に残っていない事になる。

【 0 1 3 7 】

然しながら、データベースが更新される前にレプリケートしておく事によりデータベースの信頼できるバックアップが可能になる。

10

【 0 1 3 8 】

例えば、データベースサーバ 8 1 1 又は検証サーバ 8 1 2 が、検証サーバ 8 1 2 がセカンダリボリューム 8 2 3 のデータベースの変更を開始する前に、チェックポイント（又はその他のバーチャルボリュームレプリケーション）コマンドを使用して、レプリケーションを要求する事により、バーチャルボリュームマネージャ 8 2 1 b はデータベースをバーチャルボリューム 8 2 4 にレプリケートする。

【 0 1 3 9 】

次いで、検証作業に引き続いて、検証サーバ 8 1 2 又はバックアップサーバ 8 1 3 は、バーチャルボリューム 8 2 4（ボリューム名、タイムスタンプ又はその他の ID を使用して指定する）からセカンダリボリューム 8 2 3 へ、ロールバックコマンド（又はその他のバーチャルボリュームレストアコマンド）を発行する事が出来、これを契機にバーチャルボリュームマネージャ 8 2 1 b はバーチャルボリューム 8 2 4 をセカンダリボリューム 8 2 3 にレプリケートする。

20

【 0 1 4 0 】

この結果、バックアップサーバ 8 1 3 はセカンダリボリューム 8 2 3 の信頼できるバックアップを実行でき、更に高信頼に検証することが出来る。

【 0 1 4 1 】

チェックポイント又はロールバックコマンドの実装次第では、バーチャルボリュームマネージャ 8 2 1 b は自動的に（即ちプログラマ的に）前に述べたレプリケーションを実行する可能性がある事に注意が必要である。

30

【 0 1 4 2 】

例えば、バーチャルボリュームマネージャ 8 2 1 b は、結合 8 a ~ 8 b を通して、データベースサーバ 8 1 1、アレイコントローラ 8 2 1 a、又は検証サーバ 8 1 2 によるプライマリボリューム 8 2 2 への格納、セカンダリボリューム 8 2 3 へのコピー、セカンダリボリューム 8 2 3 に対する読み取り又は初期保存等をモニタする。このようなケースで、保存、コピー、読み取り等はバーチャルボリュームマネージャ 8 2 1 b がセカンダリボリューム 8 2 3 からバーチャルボリューム 8 2 4 へのレプリケーションを開始する契機（バーチャルボリュームコマンドで指示されるのと同様に）の役割を果たす可能性がある。

【 0 1 4 3 】

更に又これに関連して、バックアップサーバ 8 1 3 から発行されるデータ読み取りコマンドは、同様に、バーチャルボリュームマネージャ 8 2 1 b にバーチャルボリューム 8 2 4 からセカンダリボリューム 8 2 3 にロールバックさせる契機を与える可能性がある。（各種の契機、即ち、コマンド、非コマンドイベント、時刻その他の繰り返される契機等が組み合わせて使用される可能性があることに注意が必要である。）

40

続いて図 9 のソフトウェア開発では、ソフトウェアのバージョン管理と条件を変えた各種の環境でのテストが課題になる。

【 0 1 4 4 】

ソフトウェアのテスト中では、例えば、ソフトウェアのバグが検出されると、本バグを修正した新たなバージョンのテストが必要になる。

50

【 0 1 4 5 】

不運にも前のソフトウェアバージョンでのテストが環境を変えているかもしれない。更に悪い事には、バグは環境又は条件によって依然として存在するかも知れず、変更して一つ以上の環境又は条件セットの下で再テストすることが望ましい。

【 0 1 4 6 】

図 9 においては、環境生成器 9 1 3 がオリジナルボリューム 9 2 5 に保存した一つ以上の環境又は条件の下で、検証サーバ 8 1 2 が検証を要求するソフトウェアを、開発ワークステーションがオリジナルボリューム 9 2 2 に格納する。

【 0 1 4 7 】

アレイコントローラ 9 2 1 a はソフトウェア及び環境と条件を各々セカンダリボリューム 9 2 3 と 9 2 6 にコピーする。ソフトウェアテストにおける問題は不運にも依然として存在して、テストによってセカンダリボリューム 9 2 6 の環境が更新され、バグが発見されて修正されると、セカンダリボリューム 9 2 3、9 2 6 は変更されその後のテストに影響することになる。

【 0 1 4 8 】

然しながら、ソフトウェア及び環境を変更される前にレプリケートすることによって、本テストは改善する事が出来る。

【 0 1 4 9 】

例えば、環境生成器 9 1 3 又はテストサーバ 9 1 2 は、テストサーバがセカンダリボリューム 9 2 6 の環境を変更する前に、チェックポイント (又はバーチャルボリュームのレプリケーション) コマンドを使用して、レプリケーションを要求して、バーチャルボリュームマネージャ 9 2 1 b に対応する環境に対応するバーチャルボリューム 9 2 7 a、b にレプリケーションを実行させる。

【 0 1 5 0 】

テストに次いで、テストサーバ 9 1 2 は、バーチャルボリューム 9 2 7 a、b の少なくとも何れかを指定してロールバック (又は他のバーチャルボリュームレストア) コマンドを発行して、バーチャルボリュームマネージャ 9 2 1 b が指定されたバーチャルボリュームをセカンダリボリューム 9 2 6 に取り戻す事が出来る。

【 0 1 5 1 】

かくして、環境を再度転送する事無く、テストを繰り返す事が出来る。

【 0 1 5 2 】

チェックポイント又はロールバックコマンドの実装次第では、バックアップの時と同様に、バーチャルボリュームマネージャ 9 2 1 b は上記で述べたソフトウェア開発でのレプリケーションを自動的に開始する事が有り得ることに注意が必要である。

【 0 1 5 3 】

例えば、バーチャルボリュームマネージャ 9 2 1 b は I / O 又はアレイコントローラ結合を通して、プライマリボリューム 9 2 5 への保存、セカンダリボリューム 9 2 6 へのコピー、或いはセカンダリボリューム 9 2 6 に対する読み取り又は初期書き込み等をモニタする。

【 0 1 5 4 】

そのようなケースで、保存、コピー、読み出し等は、バーチャルボリュームマネージャ 9 2 1 b がセカンダリボリューム 9 2 6 からバーチャルボリューム 9 2 7 a、b へのレプリケーションを開始する一つの契機になり得る。更にこれに関連して、テストサーバ 9 1 2 からの読み出しコマンドは、同様にバーチャルボリュームマネージャ 9 2 1 b がバーチャルボリューム 9 2 7 a、b からセカンダリボリューム 9 2 6 にレストアする契機 (或いは多様な契機が組合せで使用されうる) になり得る。

【 0 1 5 5 】

データレプリケーションに対しては上記したセキュリティを使用することは利益になりえる。例えば、サーバによる環境の削除、変更を禁じる事により、ソフトウェアバージョン、環境、パラメータその他の動作情報 (テスト管理、ユーザ、施設等) の検証可能な結

10

20

30

40

50

果を保護することが可能になる。

【0156】

更にその様な情報の安全なバックアップが実行され、バックアップされた情報は削除され、ストレージスペースは開放される。(バックアップは外部コントロール又はテスト、時刻決めバックアップ、データ量等を契機に自動的に実行されることにも注意が必要である。例えば、図7a～c、9を参照。)

異なったアプリケーションを含む自動動作が望ましい事に注意願いたい。あとでわかるように、定常的又はダイナミックなデータレプリケーションが可能になる。(図16の例を参照)。

【0157】

図10にて、サブプロセスのセットより構成されるバッチ処理が実行され、各サブプロセスが前のサブプロセスの更新データをソースデータとして使用して最終結果が生成される。

【0158】

不運にも、一つの間サブプロセスでの誤動作又はエラーがバッチ処理全体の再試行を余儀なくしてしまうという問題が生ずる。

【0159】

更に悪い事に、バッチ処理が複雑化しサブプロセス数が増大し扱うデータ量が増大するほど、途中での誤動作の可能性が増大するのみならず、データを特にリモートサイトから大量に再ロードしたり再スタートしたりする時間、コストも増大する。(一つのソースデータのデータベース上で異なったバッチ処理を実行する場合も同様な問題が発生する。)

然しながら、図10は、データレプリケーションを使用して上記のバッチ処理で遭遇する問題が如何に解決されるかを示す。

【0160】

図示するように、バッチサーバ1012がバッチ処理のために要求するデータベースを、生産サーバ1011がオリジナルボリューム1022に保存する。アレイコントローラ1021aがデータベースをセカンダリボリューム1023にコピーする。

バッチプロセッサ1012は、バッチサブプロセスが更新を開始して、前のバッチサブプロセスのオリジナル又は更新データが各対応するセカンダリボリューム内で変更される前に、チェックポイント(又は他のバーチャルボリュームレプリケーション)コマンドを発行して、バーチャルボリュームマネージャ1021bに対応するサブプロセスの更新データを各対応するバーチャルボリューム1024a～cにレプリケートさせる。

【0161】

その後、一つのサブプロセスで誤動作が発見された場合には、バッチサーバ1012は、関連するバーチャルボリューム1024a～cを指定してロールバック(又はその他のバーチャルボリュームのレストア)コマンドを発行する事により、バーチャルボリュームマネージャ1021bが指定されたバーチャルボリューム1024a～cのデータをセカンダリボリューム1023に取り戻す。(典型的には、誤動作のあった直前の更新データのみが取り戻され、誤動作以降の全てのバッチ処理は削除される。

然しながら、各種の結果が誤動作を特定する為に利用される。)バッチ処理はかくして、誤動作の発生したプロセスから再実行され、バッチ処理全体の再処理は防止する事が出来る。

【0162】

チェックポイント又はロールバックコマンドの実装次第では、前のアプリケーションと同様にバーチャルボリュームマネージャ1021bはバッチ処理でのレプリケーションを自動的に開始する事があることに注意が必要である。

【0163】

例えば、バーチャルボリュームマネージャ1021bはI/O又はアレイコントローラ結合を通して、セカンダリボリューム1023への書き込みをモニタして、これを契機に対応するセカンダリボリュームをバーチャルボリューム1024a～cにレプリケート

10

20

30

40

50

トする。反対にバッチサーバ 1012 からの読み取り要求を契機（あるいは多様な契機が使用されうる）にバーチャルボリュームマネージャ 1021b がロールバックを実行することが有り得る。セキュリティ/バックアップその他動作が再度実装されうる（図 8、9 を参照）。

【0164】

図 11 は、ストレージデバイスここではディスクアレイによりアプリケーションを実行して、データレプリケーションが以上で述べた効果を達成するために如何に活用されるかを示す。この例では、検証及びバックアップサーバ業務の何れか又は双方がバーチャルボリュームマネージャ 1121b によって実行される。

【0165】

例えば、図 8 のデータベースサーバ 811 及び検証サーバ 812 と同様に、データベースサーバ 1111 及び検証サーバ 1112 は、ディスクアレイコントローラ 1121a とバーチャルボリュームマネージャ 1121b と共同して動作する。

【0166】

検証サーバ 1112 は、次いで、バーチャルボリュームマネージャ 1121b に要求し、本バーチャルボリュームマネージャ 1121b は、ロールバックされたセカンダリボリューム 1123 又は更には、バーチャルボリューム 1124 を自動的又はコマンドを契機にバックアップして、検証されレプリケートばかりのバーチャルボリュームをセカンダリボリューム 1123 にロールバックする必要をなくしてくれる。

【0167】

図 12 ~ 16 は各種アプリケーションでデータレプリケーションを活用する方法を示す。本発明のよりよき理解の為に、データバックアップ、ソフトウェア開発及びバッチ処理の例を再度取り上げる。然しながら本発明のレプリケーションは、各種のアプリケーションで実質的に同様に活用できる事に注意頂きたい。

【0168】

図 12 について、本事例のデータバックアップでは、先ずオリジナルデータベースのコピーを生成する（ステップ 1201）。

【0169】

これにより、データベースのコピーについて検証することが可能になり、オリジナルデータベースをそのままにして、オリジナルデータベースを同様なコピーを通して他のアプリケーションが使用可能になる。

【0170】

データベースは典型的にはオリジナルよりはコピーをバーチャルボリュームにレプリケートし（ステップ 1203）、検証の結果オリジナル又はコピーの更新の整合性が確認された場合には、正常性が検証されたバーチャルボリュームが残る事になる。コピーは検証のためにマウントされ（ステップ 1205）、コピーの整合性が検証される（ステップ 1207）。

【0171】

ステップ 1209 の結果整合性が確認されたら、バーチャルボリュームのデータをコピーボリュームにレストア又はロールバックして（ステップ 1211）、本ロールバックデータをバックアップする（ステップ 1213）。

【0172】

ステップ 1209 の結果不整合が確認されたら、本バーチャルボリュームは削除され（ステップ 1215）、コピーはオリジナルデータベースで再同期され（ステップ 1217）、再度 1201 より開始する。

【0173】

図 13 a - 15 b でソフトウェア開発、バージョン管理及びテスト方法が例示される。

【0174】

図 13 a ~ b でのソフトウェアバージョン方法はステップ 1301（図 13 a）のソフトウェア開発で始まる。次にオリジナルソフトウェアのコピーを作成する（ステップ 130

10

20

30

40

50

3)。これにより、ソフトウェアテストは本コピーに対して実施され、オリジナルは他のアプリケーションが同様にコピーを作成して使用すること又は本ソフトウェアの一つ以上のバージョンを作成するため（例えばテストの為）に利用することが可能になる。

【0175】

ステップ1305にて、ソフトウェアは典型的にはオリジナルよりコピーがバーチャルボリュームにレプリケートされ、オリジナル又はコピーに対する更新が本バージョンのバーチャルボリュームに影響を及ぼないようにする（例えばバグを除去するための更新で新しいバージョンが発生する）。

【0176】

ステップ1307にて、ソフトウェアバージョン表示子と対応するバーチャルボリュームIDがソフトウェアバージョンテーブルに加えられる。最後にコピーは新バージョンのソフトウェア（もし存在すれば）に再同期される（ステップ1309）。 10

【0177】

図13bに、ソフトウェアのバージョンと対応するバーチャルボリュームID（例えば、名前タイプのID）を対比するソフトウェアバージョンテーブルを示す。

【0178】

図14a～bへと続き、テスト環境バージョン方法はテスト環境の開発で始まる（図14aステップ1401）。

【0179】

次いでステップ1403に於いて、オリジナル環境のコピーを作成する。これによりテストはコピー環境によって進められ、オリジナルは他のアプリケーションが同様にコピーして使用でき、又は複数の環境バージョンのテスト用として、例えばさまざまなソフトウェアや環境のテストに使用できる。 20

【0180】

ステップ1405にて、テスト環境は典型的には、オリジナルよりコピーがバーチャルボリュームにレプリケートされ、オリジナル又はコピーに対する更新が本バージョンのバーチャルボリュームに影響を及ぼないようにする（例えばバグを除去するための更新で新しいバージョンが発生する）。

【0181】

ステップ1407にて、環境バージョン表示子と対応するバーチャルボリュームIDが環境バージョンテーブルに加えられる。最後にコピーは新バージョンの環境に（もし存在すれば）再同期される（ステップ1409）。 30

【0182】

図14bに、環境のバージョンと対応するバーチャルボリュームID（例えば名前タイプのID）を対比する環境バージョンテーブルを示す。

【0183】

図15は更に図13、14aの方法を組み合わせたソフトウェアのテスト方法を説明する。

【0184】

同図ステップ1501に示す通り、本方法は、ソフトウェアと環境の1～Nの不成功な組合せiに対してステップ1503～1513を繰り返す。現ソフトウェアバージョンのバーチャルボリュームIDがソフトウェアバージョンテーブルから抽出され（ステップ1503）、バーチャルボリュームに保存されていたバージョンのソフトウェアがコピーの形でロールバックされる（ステップ1505）。 40

【0185】

次いで、現バージョンの環境に対応するバーチャルボリュームIDが図14bと同様の環境テーブルから読み取られ（ステップ1507）、指定されたコピーにロールバックされる（ステップ1509）。

【0186】

次いで、対応するバージョンのソフトウェアと環境がロードされ（ステップ1511） 50

、当該ソフトウェアは当該環境でテストされる（ステップ1513）。ソフトウェアバージョンと環境バージョンの組合せに対するテスト結果が図15bのバージョンテスト結果テーブルの如く纏められる。

【0187】

図16はバッチ処理方法の一例を示し、ステップ1601にて、ソースデータのオリジナルボリュームをセカンダリボリュームにコピーし、バッチ処理をコピーで実行できるようにし、オリジナルは他のアプリケーションが同様にコピーを作成して利用できるようにする。

【0188】

本方法ではステップ1605～1619をステップ1603に示すようにバッチ処理のサブプロセスiの1～Nについて繰り返す。 10

【0189】

ステップ1605にて、セカンダリボリュームはバーチャルボリュームにレプリケートされる。

【0190】

これにより、途中のサブプロセスが不成功になった場合にその前の状態に戻して、当該不成功サブプロセスは再実行できるようになる。次いでサブプロセス番号と対応するバーチャルボリュームIDがバッチサブプロセス又はバージョンマップに加えられ（ステップ1607）、現サブプロセスがセカンダリボリュームのコピーを使用して実行される（ステップ1609）。 20

【0191】

もし、ステップ1611にて、現サブプロセスが成功だった場合には、ステップ1603に戻り、次のサブプロセスが実行される。反対に現サブプロセスが不成功だった場合には、最初にステップ1613にて、不成功だったサブプロセスを、ステップ1607にて使用したマップを参照し特定し、バーチャルボリュームIDを取得する。

次いで本バーチャルボリュームからコピーストレージにレストアレプリケート即ち“ロールバック”する（ステップ1615）。次にステップ1617で不成功なサブプロセスの後に生成されたサブプロセスに関連するバーチャルボリュームが削除され、現サブプロセスは不成功サブプロセスにセットされ、本不成功原因を修正した後に本サブプロセスを再開始する。 30

【0192】

図17a、17bはバーチャルボリュームマネージャ1700とアレイコントローラ1720の構成を示す。

【0193】

図17aから始めると、バーチャルボリュームマネージャ1700は、バーチャルボリュームエンジン1701、参照エンジン1703、アレイコントローラインターフェース1705、アプリケーションインターフェース1707、コマンドエンジン1709、アプリケーションエンジン1711、モニタ1713、セキュリティエンジン1715、バーチャルボリュームマップ1717、及びセキュリティマップ1719を有する。

【0194】 40

バーチャルボリュームエンジン1701は、バーチャルボリューム要求に対する契機を受信し、バーチャルボリュームのコンポーネントを初期化する。

【0195】

参照エンジン1703は、各実装に於いて使用される、バーチャルボリュームIDと例えばセカンダリボリューム、アプリケーションサーバ、アプリケーション、ユーザ等の参照関係を管理する。

【0196】

既に述べた様に、このような参照はダウンロード可能で、参照エンジンによりアサインされ、又はバーチャルボリューム契機の一部として提供されアレイコントローラにより保存され、バーチャルボリュームマップ1717の中に全体又はその一部が保存される 50

。

【0197】

参照エンジン1703は更に、既に説明したように参照の抽出と決定を行う。アレイコントローラインターフェース1705は、バーチャルボリュームマネージャ1700がアレイコントローラと通信出来る様にし、例えば、バーチャルボリュームコマンドを受信しコマンドをアレイコントローラに発行して、データアクセスやサポート機能（キャッシング、エラー訂正等）を実行させる。

【0198】

コマンドエンジン1707はバーチャルボリュームコマンドを解釈し実行する（例えば、参照エンジン1703、アレイコントローラインターフェース1705、アプリケーションエンジン1711又はセキュリティエンジン1715を起動して）。 10

【0199】

アプリケーションエンジン1711は、外部コントロール又はバーチャルボリュームマネージャ1700からの要求による特殊アプリケーションの実行を推進する。

【0200】

アプリケーションエンジン1711は更に、J A V Aバーチャルマシーン、（J A V Aは登録商標です）アクティブ-Xその他の特定アプリケーションからの要求に従う為のインターフェースを提供する。そのようなアプリケーションにはデータバックアップ、ソフトウェア開発又はバッチ処理等も含まれる。

【0201】

その他のバーチャルボリュームコンポーネントとして、モニタエンジン1713は、ホストデバイス、アプリケーションサーバ、アレイコントローラ等を含むストレージ動作をモニタする。 20

【0202】

セキュリティエンジン1715はセキュリティマップ1719と共に許可、認証等のセキュリティオペレーションを実施する。

【0203】

バーチャルボリュームマップ1717とセキュリティマップ1719は、具体的実装に対応して、バーチャルボリューム参照とセキュリティ情報を提供する。

【0204】

アレイコントローラ1720（図17b）は、既に説明したアレイコントローラ機能を有するアレイエンジン1721を有する。 30

【0205】

アレイコントローラ1720は更にバーチャルボリュームインターフェース1723とセキュリティエンジン1725を有する。

【0206】

バーチャルボリュームインターフェース1723は、バーチャルボリュームマネージャとのインターフェースを提供し、例えば、本バーチャルボリュームマネージャがデータスペース共有、コマンド解釈、バーチャルボリュームキャッシングを行い、エラー訂正等その他のサポート機能を実行できるようにする。 40

【0207】

最後にセキュリティエンジン1725はセキュリティマップ1727と共に、既に図17aのバーチャルボリュームマネージャ1700の対応する要素について述べたと同様な動作を、プライマリとセカンダリボリュームのアレイデータスペースに関して実施する。

【0208】

特定の実施例に基づいて本発明を説明してきたが、ある範囲の改変、及び各種の変更と置き換えは既に述べた開示の範囲に含まれる。

【0209】

本発明のある事例でのある機能は、対応する他の機能を使用する事無く、又本発明の範 50

囲と精神から離れる事無く採用できる事に注意願いたい。

【図面の簡単な説明】

【0210】

【図1】は現行データストレージを例示するフローダイアグラムである。

【図2】は本発明の一実施例による、データレプリケーションを採用する結合システムを示すフローダイアグラムである。

【図3】は本発明の一実施例による、図2のデータレプリケーションシステム又はその要素を構成できる処理システムを示すフローダイアグラムである。

【図4】は本発明の一実施例による、データレプリケーションシステムに基づく典型的な処理システムを示すフローダイアグラムである。

【図5】は本発明の一実施例による、汎用的なデータレプリケーションシステム処理を示すフローダイアグラムである。

【図6】は本発明の一実施例による、バーチャルボリュームコマンドのコマンド構成例を示す。

【図7a】は本発明の一実施例による、アレイコントローラとバーチャルボリュームの相互動作を示すフローダイアグラムである。

【図7b】は本発明の一実施例による、バーチャルボリュームとセキュリティを組み合わせたマッピングテーブルを図示する。

【図7c】は本発明の一実施例による、アレイボリュームのマッピングテーブルを図示する。

【図8】は本発明の一実施例による、データバックアップで活用するデータレプリケーション例を示すフローダイアグラムである。

【図9】は本発明の一実施例による、ソフトウェア開発で活用するデータレプリケーション例を示すフローダイアグラムである。

【図10】は本発明の一実施例による、バッチ処理で活用するデータレプリケーション例を示すフローダイアグラムである。

【図11】は本発明の一実施例による、データレプリケーション可能デバイスがアプリケーションで活用される様子を示すフローダイアグラムである。

【図12】は本発明の一実施例による、データバックアップ方法を示すフローチャートである。

【図13a】は本発明の一実施例による、ソフトウェアバージョン方法を示すフローチャートである。

【図13b】は図13aで生成されるソフトウェアバージョンテーブルを示す。

【図14a】は本発明の一実施例による、環境バージョン方法を示すフローチャートである。

【図14b】は図14aで生成される環境バージョンテーブルを示す。

【図15a】は本発明の一実施例による、ソフトウェアテスト方法を示すフローチャートである。

【図15b】は図15aで生成されるソフトウェアテストテーブルを示す。

【図16】は本発明の一実施例による、バッチ処理方法を示すフローチャートである。

【図17a】は本発明の一実施例による、バーチャルボリュームマネージャの構成を示す。

【図17b】は本発明の一実施例による、アレイコントローラの構成を示す。

【符号の説明】

【0211】

200・・・システム

202・・・ネットワーク（インターネット）

203・・・アプリケーションサーバ（オリジナル／更新）

204・・・その他のコンピューティングシステム

211・・・ディスクアレイ

10

20

30

40

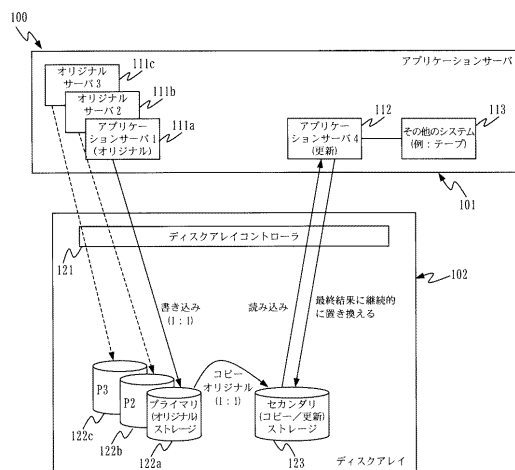
50

- 2 1 1 a . . . アレイコントローラ
- 2 1 1 b . . . バーチャルボリューム (V . V o l) マネージャ
- 2 1 1 c . . . ストレージ媒体
- 2 1 2 . . . アプリケーションサーバ 1 (管理)
- 2 1 3 . . . ネットワーク (企業内イントラネット)
- 2 1 4 a . . . アプリケーションサーバ 2 (オリジナル)
- 2 1 4 b . . . O A S (オリジナルアプリケーションサーバ) - m
- 2 1 5 a . . . アプリケーションサーバ 3 (更新)
- 2 1 5 b . . . M A S (更新アプリケーションサーバ) - n
- 2 1 6 . . . ネットワークサーバ

10

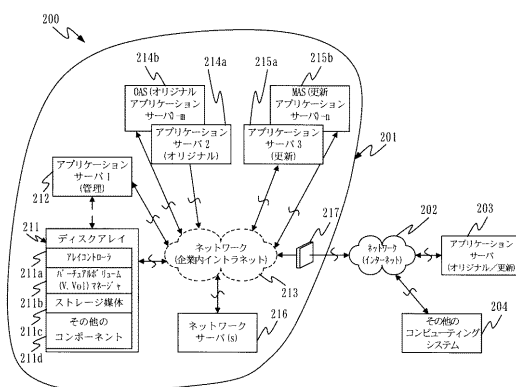
【 図 1 】

図 1 従来技術



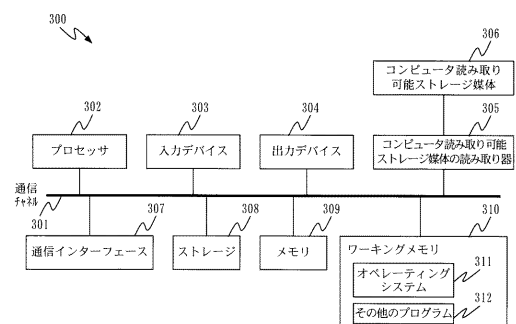
【 図 2 】

図 2

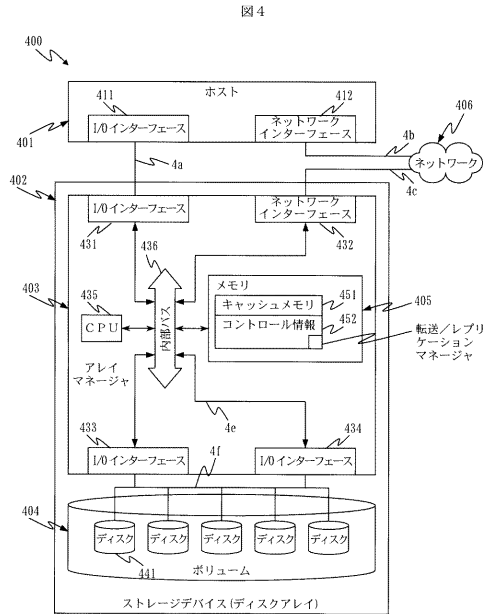


【 図 3 】

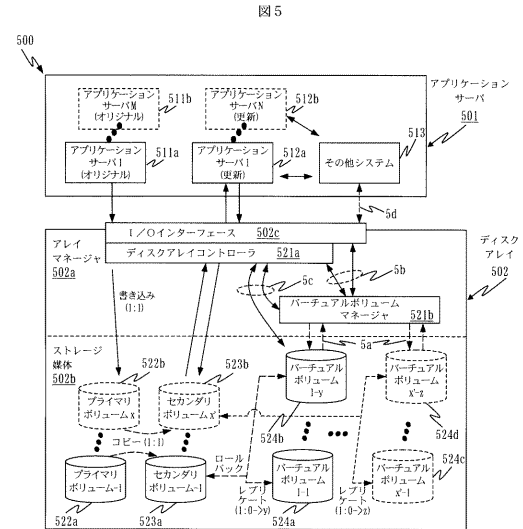
図 3



【 図 4 】



【 図 5 】



【 図 6 】

图 6

コマンドタイプ	チェックポイント
セカンダリボリューム のボリューム ID	vol23

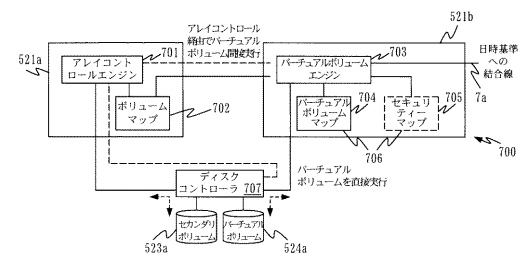
コマンドタイプ	ロールバック又は削除
セカンダリボリュームのボリューム ID	vol123
タイムスタンプ	1/22/03 8:00AM

コマンドタイプ	ロールバック又は削除
セカンダリボリュームのボリューム ID	vol123
バーチャルボリュームのボリューム ID	vol192

チェックポイントコマンドの結果として、バーチャルボリュームのどの表示をディスプレイが返送するかによる。

【 図 7 a 】

图 7 a



【 図 7 b 】

アプリケーションサーバ	名前	時刻	レイアウトロール	セカンダリボリュームID	(サーバセキュリティ) ルール/パラメータ
...
...

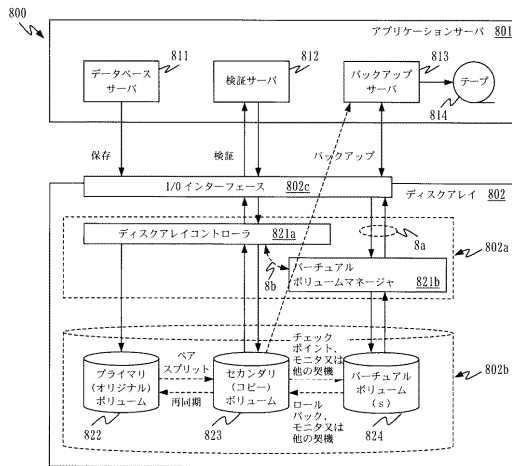
【 図 7 c 】

图 7 c

702		プライマリ	セカンダリ		
↙	アプリケーションサーバポート	ボリュームID	ボリュームID	物理ロケーション	...
	⋮	⋮	⋮	⋮	⋮

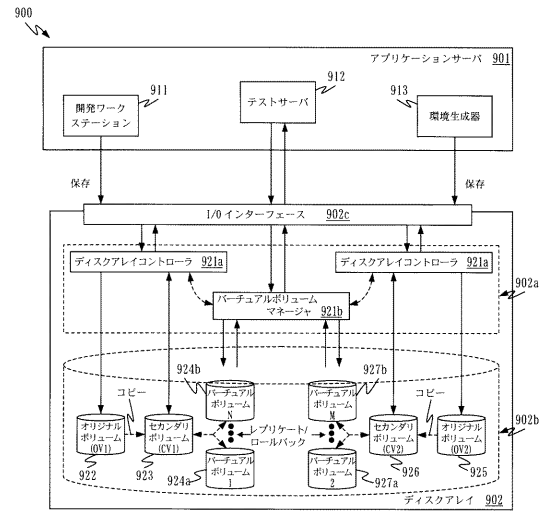
【図 8】

図 8



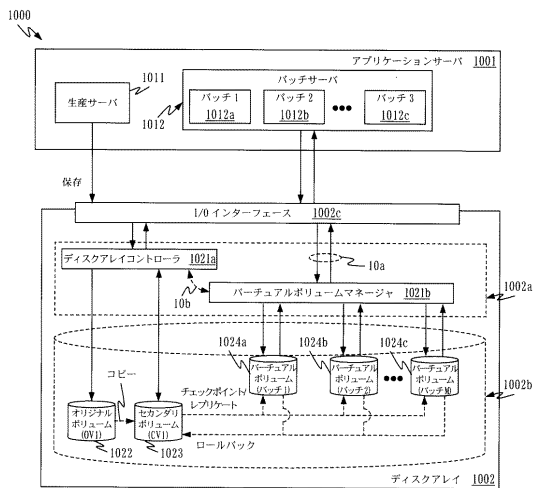
【図 9】

図 9



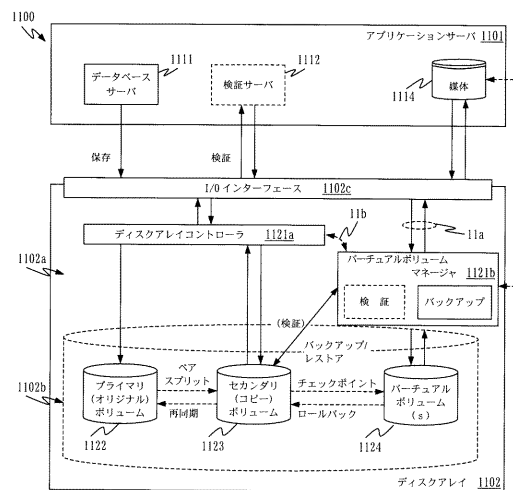
【図 10】

図 10



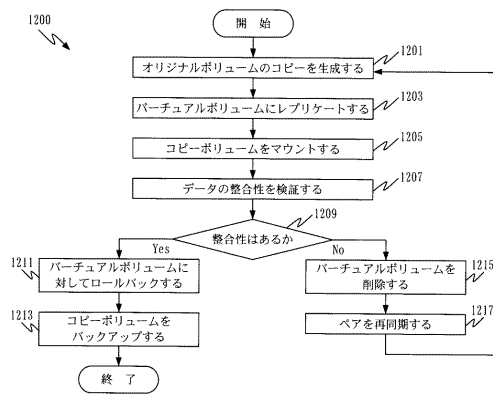
【図 11】

図 11



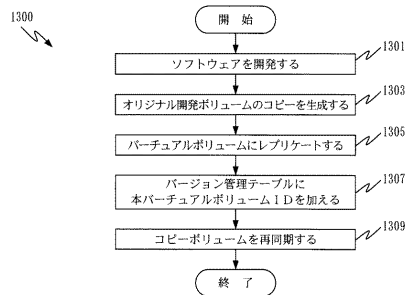
【図 1 2】

図 1 2



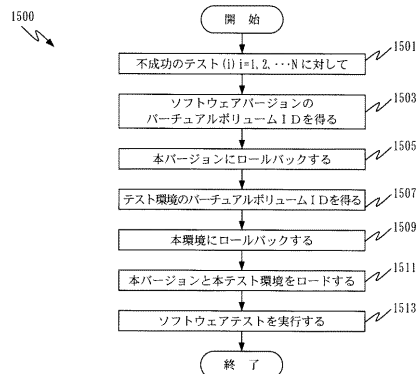
【図 1 3 a】

図 1 3 a



【図 1 5 a】

図 1 5 a



【図 1 5 b】

図 1 5 b

1510

テスト番号	バージョン	テスト環境	成功か
1	1.01	EV1	No
2	1.01	EV2	Yes
3	1.01	EV3	No
4	1.02	EV1	No
5	1.02	EV2	No
6	1.02	EV3	Yes
4	1.03	EV1	No
5	1.03	EV2	No
6	1.03	EV3	Yes

【図 1 3 b】

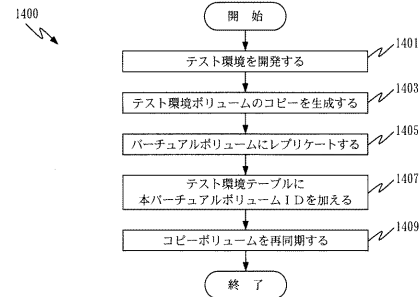
図 1 3 b

1310

バージョン	バーチャルボリューム ID
1.01	DV1
1.02	DV2
1.03	DV3
⋮	⋮

【図 1 4 a】

図 1 4 a



【図 1 4 b】

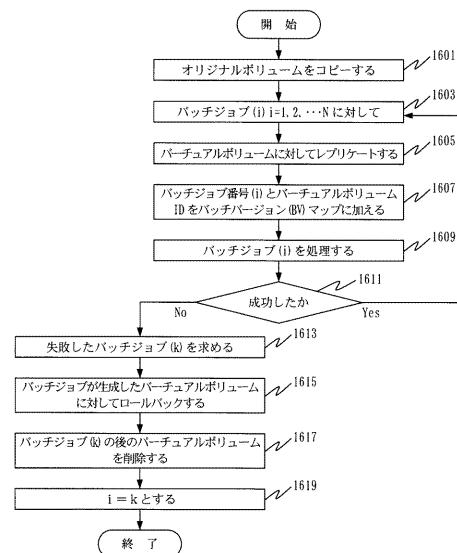
図 1 4 b

1400

テスト環境	バーチャルボリューム ID
1	EV1
2	EV2
3	EV3
⋮	⋮

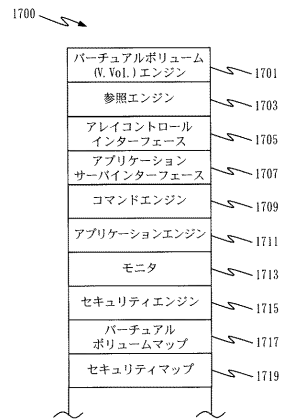
【図 1 6】

図 1 6



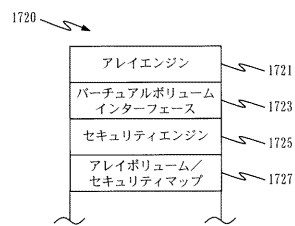
【図 17 a】

図 17 a



【図 17 b】

図 17 b



フロントページの続き

【要約の続き】

する。

【選択図】図2

【 外国語明細書 】

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
PATENT
HAL ID 248

DATA REPLICATION
FOR ENTERPRISE APPLICATIONS

REFERENCES TO OTHER APPLICATIONS

- 5 This application hereby claims priority to and incorporates by reference co-
pending application serial no. _____, entitled Method And Apparatus For
Managing Replication Volumes, filed on June 12, 2003 by Shoji Kodama and
Kenji Yamagami.

10 BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates generally to computer systems, and more particularly
provides a system and methods for data replication.

15 Background

- Today's enterprise computing systems, while supporting multiple access
requirements, continue to operate in much the same way as their small computing
system counterparts. The enterprise computing system of FIG. 1, for example,
includes application servers 101, which execute application programs that access
20 data stored in storage 102. Storage 102 includes a disk array that can, for example,
be configured as a redundant array of independent disks or "RAID".

- Disk array 102 includes an array controller 121a for conducting application
server 112a-c access to data volumes 122, 123a-c stored on storage media 121b in
the manner next described. Array controller 121 might also provide for data
25 routing, caching, redundancy, parity checking, and the like, in conjunction with the
conducting of such data access.

- Array controller 121 more specifically provides for data volume access
such that a data volume stored by a data-originating server 111a-c can be used
independently by a requesting data-utilizing application server application, e.g. of
30 server 112. Array controller 121 first responds to a request from a data-originating
application server 111a by storing original data to an original or "primary" volume.

PATENT

Express Mail Label No. EL 701375704US

Attorney Docket No. 36992.00104

HAL ID 248

Array controller 121 then responds to an initial application request from a data-utilizing application by copying the original volume data to a "secondary" volume 123; the same secondary volume is then used exclusively by that application to store successive modifications of the data, leaving the primary volume intact for
5 similar use by a different application server application.

Unfortunately, while proliferated, such multiple access configurations can nevertheless become inefficient with regard to special uses of the data.

Conventional data backup applications, for example, provide for producing a restorable backup copy of data from locally-stored data. In system 100, an
10 application server 111 application might produce a database that is stored in primary volume 122a. A backup server, e.g., server 112, might then request the database data for conducting a backup to devices 113 (e.g., a tape backup), causing the data to be copied to secondary volume 123. Since the database might be in use
15 during primary volume copying, however, verifying the secondary volume might be desirable. Unfortunately, the verification may well cause the secondary data to be replaced, causing inconsistencies in the backed up secondary volume data.

In software testing, subject software is to be tested in a test environment according to specified conditions. Here, application servers 101 might depict a development system 111a storing software in primary volume 122a, an
20 environment creator 111b storing a test environment in primary volume 122b, a condition creator 111c storing test conditions in primary volume 122c, and a tester 112 for which array controller 102 copies the software, environment and conditions to secondary volume 123 for use during testing. During testing, successively modified environment data produced by the testing replaces the
25 environment stored in the secondary volume. Thus, if the testing fails, then the potentially voluminous test components unfortunately must be re-loaded from their often remote sources.

In batch processing, a subject dataset is to be modified by successive sub-processes. Here, application servers 101 might depict a database-creating server
30 111 storing database data in primary volume 122a, and a batch processor 112 for which the database is copied to secondary volume 123. Batch processing then

PATENT
Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

causes the secondary volume data to be successively replaced by each sub-process, such that the data reflects the combined result of all completed processes.

Unfortunately, if a sub-process produces an error, then the source data must be re-loaded from its source.

5 Current configurations conducting these and other applications can also be problematic in that the stored data is accessible by the storing application, and no mechanism provides for checking that the resultant data reflects particular processing rather than mere storage of fabricated end results. Therefore, among still further disadvantages, data integrity cannot be readily assured.

10 Accordingly, there is a need for systems and methods that enable multiple-accessing of data while avoiding the data re-loading or other disadvantages of conventional systems. There is also a need for systems and methods capable of facilitating applications for which special processing of generated data might be desirable.

15

SUMMARY OF THE INVENTION

Aspects of the invention enable multiple accessing of data while avoiding conventional system disadvantages. Aspects also enable the facilitating of applications capable of reusing intermediate data results. Aspects still further
20 enable the archival, restoration, reuse and/or management of data that is intermediately or otherwise produced by various applications to be conducted.

In one aspect, embodiments enable one or more "virtual volumes" to be managed, such that intermediately produced or other application data can be selectively stored or retrieved from one or more virtual volumes, or further, in
25 addition to data stored by primary or secondary volumes. Other aspects enable the managing of virtual volumes to be conducted automatically, e.g., programmatically, with application/user control, or further, for such managing to be conducted by a mass storage, such as a disk array, among still further aspects.

In a replication method example according to the invention, a storage
30 device receives one or more first triggers, and stores source data in a primary storage and resultant data in a corresponding secondary storage. The replicating

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

method also includes replicating, responsively to the triggers, the secondary storage data to one or more corresponding virtual storage dataspace, thereby enabling the secondary storage to be restored ("rolled back") to the one or more virtual storage dataspace.

5 Advantageously, aspects of the invention enable a timed further storage of secondary storage data that might otherwise be replaced by successive resultant data. Aspects further enable selective storage and retrieval of more than one stored dataset, that can further be accomplished using simple commands or in conjunction with an existing storage device. Other advantages will also become apparent by
10 reference to the following text and figures.

15

20

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a flow diagram a prior art data storage example;

FIG. 2 is a flow diagram illustrating an interconnected system employing an exemplary data replication system, according to an embodiment of the invention;

5 FIG. 3 is a flow diagram illustrating a processing system capable of implementing the data replication system of FIG. 2 or elements thereof, according to an embodiment of the invention;

FIG. 4 is a flow diagram illustrating an exemplary processing system based data replication system, according to an embodiment of the invention;

10 FIG. 5 is a flow diagram illustrating examples of generalized data replication system operation, according to an embodiment of the invention;

FIG. 6 illustrates a exemplary command configurations for virtual volume commands, according to an embodiment of the invention;

15 FIG. 7a is a flow diagram illustrating examples of array control and virtual volume inter-operation, according to an embodiment of the invention;

FIG. 7b illustrates a combined virtual volume and security mapping table, according to an embodiment of the invention;

FIG. 7c illustrates an array volume mapping table, according to an embodiment of the invention;

20 FIG. 8 is a flow diagram illustrating an example of how data replication can be used to facilitate data backup applications, according to an embodiment of the invention;

FIG. 9 is a flow diagram illustrating an example of how data replication can be used to facilitate software development applications, according to an
25 embodiment of the invention;

FIG. 10 is a flow diagram illustrating an example of how data replication can be used to facilitate batch processing applications, according to an embodiment of the invention;

30 FIG. 11 is a flow diagram illustrating examples of how a data replication enabled storage device can be used to conduct one or more applications, according to an embodiment of the invention;

PATENT

Express Mail Label No. EL 701375704US

Attorney Docket No. 36992.00104

HAL ID 248

FIG. 12 is a flowchart illustrating a data backup method, according to an embodiment of the invention;

FIG. 13a is a flowchart illustrating a software versioning method, according to an embodiment of the invention;

5 FIG. 13b illustrates a software versioning table capable of being produced according to the method of FIG. 13a;

FIG. 14a is a flowchart illustrating an environment versioning method, according to an embodiment of the invention;

10 FIG. 14b illustrates an environment versioning table capable of being produced according to the method of FIG. 14a;

FIG. 15a is a flowchart illustrating a software testing method, according to an embodiment of the invention;

FIG. 15b illustrates a software testing table capable of being produced according to the method of FIG. 15a;

15 FIG. 16 is a flowchart illustrating a batch processing method, according to an embodiment of the invention;

FIG. 17a illustrates an exemplary virtual volume manager, according to an embodiment of the invention; and

20 FIG. 17b illustrates an exemplary array controller, according to an embodiment of the invention.

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

DETAILED DESCRIPTION

In providing for data replication systems and methods, aspects of the invention enable data, including one or more intermediately produced application data results, to be stored onto a redundant array of independent disks ("RAID"),
5 other disk array or other storage device(s) or storage device configuration(s). Aspects further enable limited or selectable access to the data, re-establishing prior data results, or conducting of security or server-based applications by a storage device, among still further combinable aspects.

Note that the term "or", as used herein, is intended to generally mean
10 "and/or", unless otherwise indicated. Reference will also be made to application servers as "originating" or "modifying", or to replicating, rollback or other system/processing aspects as being applicable to a particular device or device type, so that the invention might be better understood. It will be appreciated, however, that servers or other devices might perform different or multiple operations, or
15 might originate and process originated data. It will further become apparent that aspects might be applicable to a wide variety of devices or device types, among other permutations in accordance with the requirements of a particular implementation. Such terms are not intended to be limiting.

Turning now to FIG. 2, an interconnected system example is illustrated that
20 is configured to provide for data replication in conjunction with one or more computing devices coupled via an interconnected network 201, 202. Replication system 200 includes interconnected devices 201 coupled via intranet 213, including data replication enabled disk array 211, application servers 212, 214a-b, 215a-b and network server 216. System 200 also includes similarly coupled
25 application servers 203 and other computing systems 204, and can further include one or more firewalls (e.g., firewall 217), routers, caches, redundancy/load balancing systems, backup systems or other interconnected network elements (not shown), according to the requirements a particular implementation.

Data replication is conducted in system 200 by a disk array or other shared
30 storage, and more typically a RAID device, such as disk array 211. (Note, however, that a replication-enabled device can more generally comprise one or

PATENT

Express Mail Label No. EL 701375704US

Attorney Docket No. 36992.00104

HAL ID 248

more unitary or multiple function storage or other devices that are capable of providing for data replication in a manner not inconsistent with the teachings herein.)

5 Disk array 211 includes disk array controller 211a, virtual volume manager 211b and an array of storage media 211c. Disk array 211 can also include other components, 211d such as for enabling caching, redundancy, parity checking, or other storage or support features (not shown) according to a particular implementation. Such components can, for example, include those found in conventional disk arrays or other storage system devices, and can be configured in
10 an otherwise conventional manner, or otherwise according to the requirements of a particular application.

Array controller 211a provides for generally managing disk array operation. Such managing can include, for example communicating with other system 200 devices in conjunction with storage and retrieval of data to/from storage media
15 211c, or causing such storage, retrieval and support functions to occur. (Array controller 211a components and operation can be provided in an otherwise conventional manner subject to configuration modifications, such as in the examples that follow, or in accordance with a particular implementation.) Array controller 211a support functions can, for example, include creating, maintaining
20 or deleting dataspace references, such as files, folders, directories, volumes, and so on.

Array controller 211a provides for multiple access by creating an "original volume" for storing source or "original" data. Array controller 211a further creates an application dataspace or "secondary volume" for each modifying application
25 server application, and copies thereto data from a corresponding primary volume. Thereafter, array controller 211a receives and stores, in typically the same secondary volume, successive data modifications made and requested for storage by a corresponding application server application. Thus, the original volume remains unaltered and available to other modifying application server applications,
30 the secondary volume contains data reflecting the original data as successively

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

modified and saved by a corresponding application thus far (hereinafter referred to as "resultant data").

Array controller 211a also provides for management of other disk array components, that can include but are not limited to those already noted. Array controller 211a might further be configured in a more or less integrated manner with or to otherwise inter-operate with virtual volume manager 211b operation to various extents, or virtual volume manager 211b might be configured to operate independently, in accordance with a particular implementation.

In a more integrated configuration, array controller 211a might, for example, provide for passing application server access requests or responses from/to virtual volume manager 211b. Array controller 211a might further provide for command interpretation, or respond to virtual volume manager 211b requests by conducting storage, retrieval or other operations, e.g., operating in a similar manner as with primary/secondary volumes. (It will become apparent that a tradeoff exists in which greater integration of virtual volume management might avoid duplication of more generalized storage device control functionality, while lesser integration might enable greater compatibility with existing storage or host device implementations.)

Virtual volume ("V.Vol") manager 211b provides for creating, populating, deleting or otherwise managing one or more virtual volumes, or for enabling the selective storing, retrieving or other management of virtual volume data, typically at least within storage media 211c.

Virtual volumes, as with other volume types, provide designations of data storage areas within storage media 211c for storing application data that can be referenced by other system elements. However, unlike primary or secondary volumes, management of virtual volumes also enables one or more "snapshots" of ongoing application data of one or more applications to be selectively stored or retrieved, and for uses other than for merely providing multiple access to the same primary data by different applications. Intermediate as well as original or resultant application data can also be stored in virtual volumes. Virtual volumes can further be managed automatically (e.g., programmatically) or selectively in conjunction

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

with application, user selection or security operations, or otherwise in accordance with a particular application.

Virtual volume manager 211b provides for managing virtual volumes in response to application server application requests or "commands". Such
5 commands can, for example, be generally configured as with array controller 211a commands, thereby facilitating broader compatibility. Virtual volume manager 211b responds to a more limited set of commands, including those for creating a virtual volume, replicating stored data to a virtual volume, restoring data from a virtual volume, or deleting a virtual volume. It will be appreciated, however, that
10 greater processing capability may enable a broader range of commands or features, only some of which might be specifically referred to herein.

Virtual volume manager 211b can also be configured to communicate more directly with application server applications, or conduct management aspects more indirectly, e.g., via array controller 211a, in accordance with the requirements of a
15 particular implementation. For example, virtual volume manager 211b might, in a more integrated implementation, receive application server commands indirectly via array controller 211a or respond via array controller 211a. Virtual volume manager 211b might also operate more automatically in conjunction with monitoring array controller store, load, or other operations, or provide commands
20 to array controller 211a (e.g., as with an application) for conducting volume creation, data-storing, data-retrieving or other data access operations.

Virtual volume manager 211a might further utilize a cache or other disk array 211 components, though typically in an otherwise conventional manner in conjunction with data access. (It will be appreciated that virtual array controller
25 211a or virtual volume manager 211b might also be statically or dynamically configurable for providing one or more of such implementation alternatives, or otherwise vary in accordance with a particular application.)

Of the remaining disk array 211 components, storage media 211c provides the physical media into which data is stored, and can include one or more of hard
30 disks, rewriteable optical or other removable/non-removable media, cache memory or any other suitable storage media in accordance with a particular application.

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

Other components can, for example, include error checking, caching or other storage or application related components in accordance with a particular application.

Application servers 214a-b, 215a-b, 203, 204 provide for user/system
5 processing within system 200 and can include any devices capable of storing data to storage 211, or further directing or otherwise inter-operating with virtual volume manager 211b in accordance with a particular application. Such devices, for example, might include one or more of workstations, personal computers ("PCs"), handheld computers, settop boxes, personal data assistants ("PDAs"), personal
10 information managers ("PIMs"), cell phones, controllers, so-called "smart" devices or even suitably configured electromechanical devices, among other devices.

Of the remaining system 200 components, networks 213 and 102 can include static or reconfigurable LANs, WANs, virtual networks (e.g., VPNs), or other interconnections in accordance with a particular application. Network
15 server(s) 216 can further comprise one or more application servers configured in an otherwise conventional manner for network server operation (e.g., for conducting network access, email, system administration, and so on), or for operating as a storage device host.

Turning now to FIG. 3, an exemplary processing system is illustrated that
20 can comprise one or more of the elements of system 200 (FIG. 2). While other alternatives might be utilized, it will be presumed for clarity sake that elements of system 200 are implemented in hardware, software or some combination by one or more processing systems consistent therewith, unless otherwise indicated.

Processing system 300 comprises elements coupled via communication
25 channels (e.g., bus 301) including one or more general or special purpose processors 202, such as a Pentium®, Power PC®, digital signal processor ("DSP"), and so on. System 300 elements also include one or more input devices 303 (such as a mouse, keyboard, microphone, pen, etc.), and one or more output devices 304, such as a suitable display, speakers, actuators, etc., in accordance with a particular
30 application.

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104

HAL ID 248

System 300 also includes a computer readable storage media reader 305 coupled to a computer readable storage medium 306, such as a storage/memory device or hard or removable storage/memory media; such devices or media are further indicated separately as storage device 308 and memory 309, which can
5 include hard disk variants, floppy/ compact disk variants, digital versatile disk ("DVD") variants, smart cards, read only memory, random access memory, cache memory, and so on, in accordance with a particular application. One or more suitable communication devices 307 can also be included, such as a modem, DSL, infrared or other suitable transceiver, etc. for providing inter-device
10 communication directly or via one or more suitable private or public networks that can include but are not limited to those already discussed.

Working memory 310 (e.g. of memory 309) further includes operating system ("OS") 311 elements and other programs 312, such as application programs, mobile code, data, and so on, for implementing system 200 elements that might be
15 stored or loaded therein during use. The particular OS can vary in accordance with a particular device, features or other aspects according to a particular application (e.g. Windows, Mac, Linux, Unix or Palm OS variants, a proprietary OS, and so on). Various programming languages or other tools can also be utilized. It will also be appreciated that working memory 310 contents, broadly given as OS 311
20 and other programs 312 can vary considerably in accordance with a particular application.

When implemented in software (e.g., as an application program, object, agent, downloadable, servlet, and so on in whole or part), a system 200 element can be communicated transitionally or more persistently from local or remote
25 storage to memory (or cache memory, etc.) for execution, or another suitable mechanism can be utilized, and elements can be implemented in compiled or interpretive form. Input, intermediate or resulting data or functional elements can further reside more transitionally or more persistently in a storage media, cache or other volatile or non-volatile memory, (e.g. storage device 308 or memory 309) in
30 accordance with a particular application.

PATENT

Express Mail Label No. EL 701375704US

Attorney Docket No. 36992.00104

HAL ID 248

The FIG. 4 example further illustrates how data replication can also be conducted using a replication-enabled storage device (here, a gate array) in conjunction with a dedicated host or other application server. FIG. 4 also shows an example of a more integrated array controller and virtual volume manager combination, i.e., array manager 403. As shown, replication system 400 includes host 401, storage device 402 and network 406. Host 401, which can correspond to system 300 of FIG. 3, has been simplified for greater clarity, while a processor-based storage device implementation (gate array 402), that can also correspond to system 300 of FIG. 3, is shown in greater detail.

Host 401 is coupled and issues requests to storage device 402 via I/O interfaces 411 and 431 respectively, and connection 4a. Connection 4a can, for example, include a small computer system interface ("SCSI"), fiber channel, enterprise system connection ("ESCON"), fiber connectivity ("FICON") or Ethernet, and interface 411 can be configured to implement one or more protocols, such as one or more of SCSI, iSCSI, ESCON, fiber FICON, among others. Host 401 and storage device 402 are also coupled via respective network interfaces 412 and 432, and connections 4b and 4c, to network 406. Such network coupling can, for example, include implementations of one or more of Fibre Channel, Ethernet, Internet protocol ("IP"), or asynchronous transfer mode ("ATM") protocols, among others. The network coupling enables host 401 and storage device 402 to communicate via network 406 with other devices coupled to network 406, such as application servers 212, 214a-b, 215a-b, 216, 203 and 204 of FIG. 2. (Interfaces 411, 412, 431, 432, 433 and 434 can, for example, correspond to communications interface 307 of FIG. 3.) Storage device 402 includes, in addition to interfaces 431-434, array manager 403 and storage media 404.

Within array manager 403, CPU 435 operates in conjunction with control information 452 stored in memory 405 and cache memory 451, and via internal bus 436 and the other depicted interconnections for implementing storage control and data replication operations. Cache memory 451 provides for temporarily storing write data sent from host 401 and read data read by host 401. Cache

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248.

memory 451 also provides for storing pre-fetched data, such as successive read/write requests from host 401.

Storage media 404 is coupled to and communicates with array manager 403 via I/O interfaces 433, 404 and connection 4f. Storage media 404 includes an
5 array of disk drives 441 that can be configured as one or more of RAID, just a bunch of disks ("JBOD") or other suitable configurations in accordance with a particular application. Storage media 404 is more specifically coupled via internal bus 436 and connections 4d-f to CPU 435, which CPU conducts management of portions of the disks as volumes (e.g., primary, secondary and virtual volumes),
10 and enables host/application server 401 access to storage media via referenced volumes only (e.g., and not by direct addressing of the physical media). CPU 435 can further conduct the aforementioned security, applications or aspects or other features in accordance with a particular implementation.

The FIG. 5 flow diagram illustrates a data replication system example in
15 accordance with operational characteristics for facilitating enterprise or other applications in general. System 500 includes application servers 501, and disk array 502. Application servers 501 further include originating application servers 511a-b, modifying application servers 512a-b and other devices 513, and disk array 502 further includes array manager 502a, storage media 502b and network or input
20 output interface, ("I/O") 502c (which can, for example, correspond with interfaces 431, 432 of FIG 4). Array manager 502a includes disk array controller 521a and virtual volume manager 521b, while storage media 502b includes one or more each of primary volumes 522a-522b, secondary volumes 523a-522b and virtual volumes 524a-b and 524c-d.

25 For greater clarity, signal paths within system 500 are indicated with a solid arrow, while potential data movement between components is depicted by dashed or dotted arrows. Additionally, application servers 501, for purposes of the present example, exclusively provide for either supplying original data for use by other servers (originating application servers 1-M 511a, 511b) or utilizing data supplied
30 by other application servers (modifying application servers 1-n 512a, 512b). Each

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

of application servers 511a-b, 512a-b communicates data access requests or "commands" via I/O 502c to array manager 502a.

Originating application server 511a-b applications issue data storage commands to array controller 521a, causing array controller 521a to create a
5 primary volume, e.g., 522a, if not yet created, and storing therein the included original data. Modifying application server 512a-b applications issue data retrieval commands to array controller 521a, causing array controller 521a to return to the requesting application server the requested data, via I/O 502c. Modifying
10 application server applications can also issue data storage commands, causing array controller 521a to create a secondary volume, e.g., 523a, if not yet created, and replacing the data stored therein, if any, with the included resultant data. (For purposes of the present example, only one primary volume might be created for a corresponding dataset and only one secondary volume might be created for each application server, as in conventional systems).

15 Originating application servers 511a-b generally need not communicate with virtual volume manager 521b. Further, the one or more primary volumes 522a-b that might be used generally need not be coupled to virtual volume manager 521b, since primary volume data is also available for access by virtual volume manager 521b, via array controller copying, from the one or more of
20 secondary volumes 523a-b that might be used. Thus, unless a particular need arises in a given implementation, system 500 can be simplified by configuring disk array 502 (or other storage devices that might also be used) without such capability.

Modifying application server 512a-b applications can, in the present example, issue a limited set of virtual volume commands including checkpoint,
25 rollback and virtual volume delete commands.

An application issuing a checkpoint command causes virtual volume manager 521b to create a virtual volume (e.g., virtual volume 1-1, 524a) and to replicate, to the newly created virtual volume, the current data of the secondary volume that corresponds to the requesting application (e.g. secondary volume-1
30 523a). Further checkpoint commands from the same application would cause virtual volume manager 521b to similarly create further virtual volumes

PATENT

Express Mail Label No. EL 701375704US

Attorney Docket No. 36992.00104

HAL ID 248

corresponding to the application (e.g., virtual volume 1-y 524b) and to replicate the then current data of the same secondary volume. However, a further checkpoint commands from a different application would cause virtual volume manager 521b to create a virtual volume corresponding to the different application (e.g., virtual
 5 volume x'-1) and to replicate the then current data of the secondary volume corresponding to the different application, e.g. secondary volume x' 523b, to the virtual value.

An application issuing a rollback command causes virtual volume manager 521b to restore, to the secondary volume corresponding to the requesting
 10 application (e.g., secondary volume-1523a), the data stored in the virtual volume that is indicated by the command (e.g. virtual volume 1-1 524a). Further rollback commands from the same application would cause virtual volume manager 521b to similarly replace the data of the same secondary volume (e.g., secondary volume-1523a) with the contents of the indicated virtual volume (e.g. virtual volume 1-1
 15 524a), while a further rollback command from a different application would cause virtual volume manager 521b to replace the data of the secondary volume corresponding to the different application (e.g. 523b) with the indicated virtual volume data (e.g., virtual volume x'-z 524d).

Finally, an application issuing a virtual volume delete command causes
 20 virtual volume manager 521b to delete the virtual volume indicated in the command. (Examples of checkpoint, rollback and delete commands are depicted in FIG. 6)

Within array manager 502a, array controller 521a receives and responds to data access store commands from an originating application server 511a-b
 25 application by creating a primary volume corresponding to the originating application (e.g., primary volume 522a) if a corresponding primary volume does not already exist, and storing therein the included data. Array controller 521a further responds to a store command from a modifying application server 512a-b application by creating a secondary volume corresponding to the modifying
 30 application (e.g. secondary volume 523a) for further use by the modifying application, and storing therein the included data. Array controller 521 also

PATENT

Express Mail Label No. EL 701375704US

Attorney Docket No. 36992.00104

HAL ID 248

responds to successive store commands from the application by successively replacing the same secondary volume data with the resultant data provided by the application. Array controller 521a responds to retrieve or delete commands from an application server 501 application by respectively returning to the application
5 server the corresponding primary or secondary volume data of the volume indicated in the command, or by deleting the corresponding primary or secondary volume indicated in the command.

Array controller 521a, in the present configuration, receives but disregards virtual volume commands received from application servers 501. Array controller
10 521a does, however, receive and respond to data access commands from virtual volume manager 521b via interconnections 5b or 5c.

More specifically, virtual volume manager 521b responds to a checkpoint command by determining a corresponding secondary volume and assigning a virtual volume designation corresponding to the command-issuing application
15 server application, and storing the command-issuing application and volume correspondence and assignment for further use. Virtual volume manager 521b also determines and assigns to the virtual volume a time (and date) stamp indicator, and stores the time stamp indicator with the correspondence information, again for further use. Such further use can include, for example, a further rollback or delete
20 volume command by the same application or in assigning a further virtual volume for the same or another application server application.

Virtual volume manager 521b further issues a copy command to array controller 521a including the secondary volume (e.g. secondary volume-1 523a) as the source of the copying and the virtual volume (e.g. virtual volume 1-1 524a) as
25 the destination of the copying. (It will be appreciated that a similar result could also be achieved, for example, by virtual volume manager 521b issuing to array controller 521a a data retrieve command indicating the same source and a store command indicating the same destination.)

Array controller 521a responds to the copy command in much the same
30 manner as with receiving the above retrieve and store command sequence or copy command as if issued by an application server application, treating the virtual

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

volume as a further secondary volume. That is, array controller 521a first issues a pair-split request to a disk controller (e.g., see FIG 7a), thus preventing updating of the secondary volume with any new primary volume data while the secondary volume is being accessed by array controller 521a. Array controller then creates
5 the new virtual volume as a secondary volume (e.g., virtual volume 1-1 524a), conducts the copying of the secondary volume data to the virtual volume, and issues a synchronization request to again link the corresponding primary volume with the secondary volume.

Virtual volume manager 521b responds to a rollback command by
10 determining a secondary volume and a virtual volume corresponding to the command-issuing application server application. In this example, the secondary volume is again unknown by reference to the command. Further, while a virtual volume indicator is provided by the command, the indicator may be a time/date indicator or an ID that will not be recognized by array controller 521a. Therefore,
15 virtual volume manager 521b determines secondary volume and virtual volume corresponding references used by array controller 521a by reference to the indicators stored secondary volume during a prior checkpoint command.

Virtual volume manager 521b further issues a copy command to array controller 521a including the determined virtual volume reference (e.g. to virtual
20 volume 1-1 524a) as the source of the copying and the secondary volume (e.g. secondary volume-1 523a) as the destination of the copying. (Note that a similar result could also be achieved for example, by virtual volume manager 521b issuing to array controller 521a a data retrieve command indicating the same source followed by a store command indicating the same destination.)

25 Array controller 521a again responds to the copy command in much the same manner as with receiving the above retrieve and store command sequence or copy command as if issued by an application server application, treating the virtual volume as a further secondary volume. That is, array controller 521a first issues a pair-split request, which prevents updating of the secondary volume with any new
30 primary volume data while the secondary volume is being accessed by array controller 521a. (Array controller 521a does not need to create a volume since

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

both the secondary and virtual volumes already exist.) Array controller 521a then conducts the copying of the virtual volume data to the secondary volume, and issues a synchronization request to again link the corresponding primary volume with the secondary volume.

5 Virtual volume manager 521b responds to a delete virtual volume command by determining a virtual volume corresponding to the command-issuing application server application. In this case, a virtual volume indicator provided by the command might again be a time/date indicator or a ID assigned by a virtual volume manager that will not be recognized by array controller 521a. Therefore,
10 virtual volume manager 521b determines the corresponding "secondary volume" reference used by array controller 521a by reference to the indicator stored during a prior checkpoint command (e.g., see FIGS. 7a-7c).

 Virtual volume manager 521b further issues a delete (secondary volume) command to array controller 521a including the virtual volume (e.g. virtual volume
15 1-1 524a) as the (secondary) volume to be deleted. Array controller 521a responds to the delete command as if issued by an application server application, treating the virtual volume as a further secondary volume. That is, array controller 521a may first issue a split-pair request, which would prevent attempted updating of the secondary volume with any new primary volume data. Array controller 521a again
20 does not need to create a volume, and proceeds with conducting the deleting of the virtual volume indicated by the delete command from virtual volume manager 521b.

 Note that a similar result could also be achieved in a less integrated implementation, for example, by reserving a dataspace within storage media 502b
25 for virtual volumes and configuring virtual volume manager 521b to conduct the copying or deleting operations directly. For example, virtual volume manager 521b might respond to a checkpoint command by determining a corresponding secondary volume and assigning a virtual volume name designation, time stamp or other indicator corresponding to the command-issuing application server
30 application and time/date or other information, storing the command-issuing application and volume correspondence and assignments.

PATENT

Express Mail Label No. EL 701375704US

Attorney Docket No. 36992.00104

HAL ID 248

Virtual volume manager 521b might then issue a pair-split command (depending on the implementation) and a copy command indicating the secondary volume (e.g. secondary volume-1 523a) as the source of the copying and the virtual volume (e.g. virtual volume 1-1 524a) as the destination of the copying, and then
5 issue a synchronization command, (e.g., see FIG. 7a), among other alternatives.

Reserving a dataspace might cause storage space to be wasted in a statically assigned dataspace configuration or add complexity in a dynamic dataspace configuration, in which space is allocated as needed. However, such configurations might provide a higher degree of control of virtual volumes by a
10 virtual volume manager, for example, in providing additional features without otherwise impacting array control, such as in providing security that cannot be as easily surmounted via array control. Other features might also be similarly implemented in accordance with a particular application. (While other mechanisms, such as using a shared space for both array control and virtual
15 volume management operations might also be similarly used, the above mechanisms appear to provide similar functionality while imposing less system complexity.)

FIGS. 7a through 7c illustrate examples of the inter-operation of array controller 521a and virtual volume manager 521b in greater detail. Beginning with
20 FIG. 7a, array controller 521a includes array engine 701, which conducts array control operation in conjunction with the mapping of primary and secondary volumes to application servers and physical media provided by multiple-access volume map 702. Virtual volume manager 521b includes virtual volume engine 703, which conducts virtual volume management operation in conjunction with
25 virtual volume map 702, and optionally, further in accordance with security map 705. Virtual volume manager 521b also includes an interconnection 7a to a time and date reference source, which can include any suitable time and date reference.

In each of the checkpoint, rollback and delete commands, virtual volume manager 521b can, for example, determine the corresponding references by
30 building and maintaining virtual volume map 704. Turning also to the exemplary virtual volume (and security) map of FIG. 7b, a mapping of physical media to

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

volumes can be maintained in accordance with a particular array control implementation that includes a correspondence between primary and secondary volumes, and accessing application servers.

Turning further to the exemplary access volume map of FIG. 7c, virtual volume manager 521b (FIG. 7a) can poll the access volume map prior to executing a command (or the basic map can be polled at startup and modifications to the map can be pushed to virtual volume controller, and so on). Virtual volume manager 521b can determine there from secondary volume correspondences, as well as secondary volume assignments made by array controller 521a for referencing virtual volumes (see above). Virtual volume manager 521b can, for example, add such correspondences to map 706 and further add its own virtual volume (ID) assignments to map 706. Virtual volume manager 521b can thus determine secondary volume and virtual volume references as needed by polling such a composite mapping (or alternatively, by reference to both mappings). Other determining/referencing mechanisms can also be used in accordance with a particular implementation such as, for example, that of the above-referenced co-pending application.

FIG. 7b also illustrates an example of how virtual volume manager 521a further enables virtual volume limitations, security or other features to be implemented. A virtual volume manager might, for example implement a limitation on a number of virtual volumes, amount of data that can be stored or other combinable features by polling a virtual volume mapping that includes applicable operational information. Virtual volume map 706, for example, includes application server (applications) 761, as well as corresponding assigned virtual volume reference designations 762a, 762b (e.g., including numerical name and time stamp designations). Virtual volume map 706 also includes array controller references 764 and, in this example, server-specific security parameter rules/parameters 765.

Virtual volume manager 521b can, for example, implement security protocols by comparing an access attempt by an application server, application, user, and so on, to predetermined rules/parameters 765 indicating those access

PATENT
Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

attempts that are or are not allowable by one or more servers, applications, users,
and so on. Such access attempts might, for example, include one or more of
issuing a rollback or deleting virtual volumes generally or further in accordance
with particular data, data types or further more specific characteristics; among
5 other features.

Turning now to FIGS. 8 through 11, data replication can be used to
facilitate a wide variety of applications. FIG. 8, for example, illustrates a data
backup system, while FIG. 9 illustrates a software development system and FIG.
10 illustrates a batch processing system, each of which is facilitated through the
10 use of data replication. FIG. 11 further provides an example of how data
replication can be used to conduct applications or automatic operation. (For
consistency, FIG. 11 again uses data backup as an exemplary application.)

Beginning with FIG. 8, it is difficult to assure that a replicated database, i.e.,
database management system data, or some other subject code or data, in a
15 multiple-access system will be fully updated and consistent. Therefore, it is
desirable to verify the database before conducting a data backup of the database.
The use of data replication enables verification to be conducted that might alter the
subject database as otherwise provided, while still maintaining a reliable database
to backup.

20 In FIG. 8, for example, database server 811 stores a database in original
volume 822 that verification server 812 requests for verification. Array controller
821a initiates a pair-split to prevent further synchronization of changes to the
database by database server 811 with secondary volume 823 and copies the
database to volume 823. A problem in conducting the verification unfortunately
25 still exists, since the verification may alter the secondary volume 823 database and
the original volume 822 database might have been updated after the copying.
Therefore, no reliable database would remain to be backed up.

However, a reliable database can be provided for backup by replicating the
database before it is altered. For example, database server 811 or verification
30 server 812 can initiate replication using checkpoint or other virtual volume
replication command prior to a verification server initiated updating of the

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

database in secondary volume 823, causing virtual volume manager 821b to replicate the database to a virtual volume 824. Then, following verification, verification server 812 or backup server 813 can issue a rollback or other virtual volume restore command indicating virtual volume 824 as the source (e.g. using a
5 volume name, timestamp indicator or other ID) causing virtual volume manager 821 to replicate virtual volume 824 to secondary volume 823. Backup server 813 can then conduct a reliable data backup of secondary volume 823, that can further be reliably restored or verified and restored.

Note that virtual volume manager 821 might also automatically (e.g.,
10 programmatically) initiate the aforementioned replications. For example, virtual volume manager 821b might monitor, via connections 8a or 8b, the storing to primary volume 822, copying to secondary volume 823 or retrieving or initial storing from/to secondary volume 823 respectively of database server 811, array controller 821 or verification server 812. In such cases, the store, copy, retrieve or
15 other store might serve as a triggering event (such as with a virtual volume command or other triggers), causing virtual volume manager 821 to initiate a replication of secondary volume 823 to virtual volume 824. Alternatively or in conjunction therewith, a data retrieval command issued by backup server 813 might similarly trigger a restore replication by virtual volume manager 821b from
20 virtual volume 824 to secondary volume 823. (It will be appreciated that various triggers might also be used in combination, e.g., one or more of command, singular non-command event, timed or other repeated triggers and so on.)

Continuing with FIG. 9, software development raises the problems of managing versions of the software and testing of the software in various
25 environments or according to varying conditions. During testing of the software, for example, software errors or "bugs" might be identified, causing the software to be modified to a new version that might also require testing. Unfortunately, testing of the prior software version might have changed the environment. Worse yet, bugs might also exist with regard to the environment or conditions, requiring
30 modification and re-testing, or testing of the software using more than one environment or set of conditions might be desirable.

PATENT

Express Mail Label No. EL 701375704US

Attorney Docket No. 36992.00104

HAL ID 248

In FIG. 9, for example, development workstation 911 stores software in original volume 922 that verification server 812 requests for verification in conjunction with one or more environments or conditions stored by environment creator 913 in original volume(s) 925. Array controller 821a copies the software and environment plus conditions respectively to secondary volumes 923 and 926. A problem in conducting the software testing unfortunately still exists, since testing would replace the environment of secondary volume 926 and, assuming bugs are found and modifications made, one or more of secondary volumes 923, 926 might also result, interfering with further testing.

However, such testing can be facilitated by replicating the software or environment prior to their modification. For example, environment creator 913 or test server 912 can initiate replication using checkpoint or other virtual volume replication command prior to a test server initiated updating of the environment in secondary volumes 926, causing virtual volume manager 921b to replicate the respective environments to corresponding ones of virtual volumes 927a through 927b. Then, following testing using an environment, test server 912 can issue a rollback or other virtual volume restore command indicating at least one of the virtual volume 927a-b as the source and causing virtual volume manager 921b to replicate the respective virtual volume(s) to secondary volume(s) 926. Testing can then be repeated without a need to again transfer the environment to

Note that, as with data backup, virtual volume manager 821 might also automatically initiate the aforementioned software development replications. For example, virtual volume manager 821b might monitor, via the I/O or array controller connections, the storing to primary volume(s) 925, copying to secondary volume(s) 926 or retrieving or initial storing from/to secondary volume(s) 927a, 927b. In such cases, the store, copy, retrieve or other store might serve as a triggering event, causing virtual volume manager 921b to initiate a replication of secondary volume(s) 926 to virtual volume(s) 927a, 927b. Alternatively or in conjunction therewith, a data retrieval command issued by test server 912 might similarly trigger a restore replication by virtual volume manager 921b from virtual

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

volume(s) 927a, 927b to secondary volume(s) 926 (or various triggers might again be used in combination).

Note also that data replication further utilizing the aforementioned security might be advantageous. For example, by disallowing deletion or modification of
5 an environment by one or more servers or users, a verifiable record can be preserved of the software version, environment, parameters or other operational information (e.g., test administration, user, facility, and so on). A further (secure) backup of such information might then be conducted such that the information backed up might be deleted and storage space freed. (Once again, the backup
10 might also be in conjunction with an external control or automatically conducted in accordance with testing, timed backup, amount of data, and so on. See, for example, FIGS. 7a-c, 9.)

It will still further be appreciated that particularly automatic operation involving different applications might be desirable. As will become apparent,
15 static or dynamic configuration of data replication can also be provided. (See, for example, FIG. 16.

Continuing with FIG. 10, in batch processing a set of sub-processes is conducted in a sequence to produce a final result, with each sub-process using the prior resultant data as source data to be further processed. A problem
20 unfortunately arises in that a misoperation or other "error" during some intermediate sub-process would ordinarily require a re-starting of the entire batch process. To make matters worse, more complex batch processing, more sub-processing or greater amounts of data can not only increase the likelihood of a misoperation, but also increase the time or cost involved in re-starting as well as
25 re-loading of the data, particularly voluminous data or data from a remote site. (A similar problem also arises where different batch processing might be conducted on a database of source data.)

FIG. 10, however, shows an example of how data replication can be used to avoid problems otherwise encountered in batch processing. As shown, production
30 server 1011 stores a database in original volume 1022 that batch server 1012 requests for batch processing. Array controller 1021a copies the database to

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

secondary volume 1023. Batch processor 1012 can initiate replication using a checkpoint or other virtual volume replication command prior to a batch sub-process initiated updating causing the original or resultant data of the prior sub-process to be stored in a respective one of secondary volumes 1024a-c, causing
5 virtual volume manager 921b to replicate the respective resultant data of the sub-process to the virtual volume (e.g., 1024a). Then, following a sub-process disoperation or upon further batch processing and discovery of the misoperation, batch server 1012 can issue a rollback or other virtual volume restore command indicating at least one of the virtual volume 1024a-c as the source, and causing
10 virtual volume manager 921b to replicate the respective virtual volume(s) to secondary volumes 1023. (Typically, only the resultant data immediately prior to the misoperation will be restored, while all batch processes from the misoperation to completion will be deleted. However, the various results might also be used to identify the misoperation or for other purposes.) Batch processing can then be re-
15 started from the process causing the misoperation without a need to re-start the entire batch processing.

Note that, as with the prior applications, virtual volume manager 1021b might also automatically initiate the aforementioned batch processing replications. For example, virtual volume manager 821b might monitor, via the I/O or array
20 controller connections, the storing to secondary volume 1023 which might serve as triggering events, causing virtual volume manager 1021b to initiate a replication of corresponding secondary volume to a respective one of virtual volume(s) 1024a-c. Alternatively or in conjunction therewith, a data retrieval command issued by batch server 1012 might similarly trigger a restore replication by virtual volume
25 manager 1021b (or various triggers might again be used in combination). Security/backup or other operation might also, again, be implemented (e.g., see FIGS. 8 and 9).

FIG. 11 further illustrates how data replication can be used to achieve the above or other advantages while further enabling the application to be conducted
30 by a storage device, here a further disk array. In this example, either one or both of verification and backup server operation can be implemented by virtual volume

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

manager 1121b. For example, database server 1111 and verification server 1112 can operate in conjunction with disk array controller 1121a and virtual volume manager 1121b as discussed with regard to database server 811 and verification server 812 of FIG. 8. Verification server 1112 can then trigger virtual volume manager 1121b or virtual volume manager 1121b can initiate and conduct the backing up of either restored secondary volume data 1123, or further of virtual volume data 1124 automatically or upon a command trigger, thereby avoiding the need to restore the now-verified replicated virtual volume data to secondary storage 1123.

FIGS. 12 through 16 further illustrate methods for using data replication to facilitate various applications. Examples of the specific applications of data backup, software development and batch processing will again be used so that aspects of the invention might be better understood. It will be appreciated, however, that a variety of applications can be facilitated in a substantially similar manner in conjunction with data replication in accordance with the invention.

Beginning with FIG. 12, an exemplary data backup method includes making a copy of the original database in step 1201. This enables verification to be conducted on the copy of the database while preserving the original database for use by other applications via similar copying. In step 1203, the database, and typically the copy rather than the original, is replicated to a virtual volume, such that any updating of the original, or modification of the copy as a result of the verification, will still leave a verified virtual volume if the verification shows that the copy is/was reliable or "consistent". In step 1205, the copy is mounted for verification, and in step 1207, the copy is verified for consistency.

If, in step 1209, the copy is consistent, then a restore replication or "rollback" of the virtual volume data to the copied data storage is conducted in step 1211 and the restored copy is backed up in step 1213. If instead in step 1209, the copy is unreliable or "inconsistent", then the virtual volume is deleted in step 1215, the copy is re-synchronized with the original database in step 1217 and the method returns to step 1201.

PATENT
Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

In FIGS. 12a through 15b, exemplary software development, versioning and testing methods are illustrated.

Beginning with FIGS. 13a-b, a software versioning method includes developing software in step 1301 (FIG. 13a). Then, step 1303 includes making a
5 copy of the original software. This enables testing to be conducted on the copy of the software while preserving the original for use by other applications via similar copying, or for otherwise versioning of more than one version of the software, e.g. for testing. In step 1305, the software, and typically the copy rather than the original, is replicated to a virtual volume, such that any updating of the original, or
10 modification of the copy will still leave the virtual volume of the software version intact (e.g., if modification to remove bugs results in a new version). In step 1307, a software version indicator and its corresponding virtual volume ID are added to a software versioning table. Finally, the copy is re-synchronized with a new version of the software, if any, in step 1309. FIG. 13b further shows how a resulting
15 software versioning table includes the software version indicator and its corresponding virtual volume ID (e.g., a name type ID).

Continuing with FIGS. 14a-b, a test environment versioning method includes developing the test environment in step 1401 (FIG. 14a). Then, step 1403 includes copying the original software. This enables testing to be conducted on the
20 copy of the environment while preserving the original for use by other applications via similar copying, or for otherwise versioning of more than one version of the environment, e.g. for varied software or environment testing. In step 1405, the environment, and typically the copy rather than the original, is replicated to a virtual volume, such that any updating of the original, or modification of the copy
25 will still leave the virtual volume of the environment version intact (e.g., if a modification to remove bugs results in a new version). In step 1407, an environment version indicator and its corresponding virtual volume ID are added to an environment versioning table. Finally, the copy is re-synchronized with a new version of the environment, if any, in step 1309. FIG. 14b further shows how
30 a resulting environment versioning table includes the software version indicator and its corresponding virtual volume ID (e.g., a name type ID).

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

FIG. 15 further illustrates a software testing method operating in conjunction with the prior methods of FIGS. 13 and 14a. As shown, the method includes repeating steps 1503 through 1513 for unsuccessful testing of one to I tests of n software and environment combinations, as given in step 1501. In step 5 1503, the virtual volume ID of the current software version is retrieved, for example from a software version table, such as that of FIG. 13b. In step 1505, a rollback is conducted from the software version in the copy to the software version stored in a virtual volume. In step 1507, a virtual volume ID of a current environment is further similarly retrieved as the ID corresponding to the 10 environment version found, for example, in an environment table, such as that of FIG. 14b, and a rollback is conducted from the environment version in the copy to the environment version stored in the corresponding virtual volume, in step 1509. Then, the software and test environment copies are loaded in step 1511, and a test of the software running with the environment is conducted in step 1513. 15 (Indicators of the testing results can also be stored, such as with the resulting version-test results table of FIG. 15b.)

FIG. 16 illustrates an exemplary batch processing method including copying an original volume of source data to a secondary volume in step 1601. This enables batch processing to be conducted on the copy of the source data while 20 preserving the original for use by other applications via similar copying. The method further includes repeating steps 1605 through 1619 for one to n batch subprocesses or "batch jobs" of the batch process, as given in step 1603. In step 1605, the secondary volume is replicated to a virtual volume. This enables the source data to be returned to a state prior to a batch job that is unsuccessful, so that 25 processing may be started again with the unsuccessful batch job. In step 1607, the batch job number and corresponding virtual volume ID are added to a batch subprocess or "version" map in step 1607, and the current batch job is processed using the secondary volume copy in step 1609.

If, in step 1611, the current batch job is successful, then the method 30 continues with step 1603. If instead the current batch job is unsuccessful, then,

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

first in step 1613 the unsuccessful batch job is identified, e.g., by reference to the mapping of step 1607 and returning of the corresponding virtual volume ID. Next, a restore replication or “rollback” of the virtual volume given by the returned virtual volume ID to the copied data storage is conducted in step 1615. Next, the virtual volumes created in conjunction with batch jobs after the unsuccessful batch job are deleted in step 1617, and, in step 1619, the current batch job is set to the unsuccessful batch job, for example, to continue the batch processing with the unsuccessful batch job after the cause of the error is corrected.

FIGS. 17a and 17b illustrate further examples of a virtual volume manager 1700 and an array controller 1720 respectively.

Beginning with FIG. 17a, virtual volume manager 1700 includes virtual volume engine 1701, reference engine 1703, array control interface 1705, application interface 1707, command engine 1719, application engine 1711, monitor 1713, security engine 1715, virtual volume map 1717 and security map 1719. Virtual volume engine 1701 provides for receiving virtual volume triggers and initiating other virtual volume components. Reference engine 1703 provides for managing virtual volume IDs and other references, e.g., secondary volumes, application servers, applications, users, and so on, as might be utilized in a particular implementation. As discussed, such references might be downloadable, assigned by the reference engine or provided as part of a virtual volume trigger or as stored by an array controller, and might be stored in whole or part in virtual volume map 1719.

Reference engine 1703 also provides for retrieving and determining references, for example, as already discussed. Array control interface 1705 provides for virtual volume manager 1700 interacting with an array controller, for example, in receiving virtual volume commands via or issuing commands to an array controller for conducting data access or support functions (e.g., caching, error correction, and so on). Command engine 1707 provides for interpreting and conducting virtual volume commands (e.g., by initiating reference engine 1703, array control interface 1705, application engine 1711 or security engine 1715).

PATENT

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
HAL ID 248

Application engine 1709 provides for facilitating specific applications in response to external control or as implemented by virtual volume manager 1700. Application engine 1709 might thus also include or interface with a java virtual machine, active-X or other control capability in accordance with a particular implementation (e.g., see above). Such applications might include but are not limited to one or more of data backup, software development or batch processing.

Of the remaining virtual volume components, monitor engine 1713 provides for monitoring storage operations, including one or more of a host device, other application server or array controller. Security engine 1715 provides for conducting security operations, such as permissions or authentication, e.g., see above, in conjunction with security map 1719. Virtual volume map 1717 and security map 1719 provide for storing virtual volume reference and security information respectively, e.g., such as that discussed, in accordance with a particular implementation.

Array controller 1720 (FIG. 17b) includes an array engine 1721 that provides for conducting array control operations, for example, in the manner already discussed. Array controller 1720 also includes virtual volume interface 1723 and security engine 1723. Virtual volume interface 1723 provides for inter-operation with a virtual volume manager, for example, one or more of directing commands to a virtual volume manager, conducting dataspace sharing, interpreting commands or conducting virtual volume caching, error correction or other support functions, and so on. Finally, security engine 1705 operates in conjunction with security map 1707 in a similar manner as with corresponding elements of the virtual volume manager 1700 of FIG. 17a, but with respect to array dataspace, such as primary and secondary volumes.

While the present invention has been described herein with reference to particular embodiments thereof, a degree of latitude of modification, various changes and substitutions are intended in the foregoing disclosure, and it will be appreciated that in some instances some features of the invention will be employed without corresponding use of other features without departing from the spirit and scope of the invention as set forth.

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

WHAT IS CLAIMED IS:

1. A method, comprising:

(a) receiving one or more first triggers by a storage device storing source data in a primary storage and resultant data in a corresponding secondary storage;
5 and

(b) replicating, responsively to the triggers, the secondary storage data to one or more corresponding virtual storage dataspace, thereby enabling the secondary storage to be restored ("rolled back") to the one or more virtual storage dataspace.

10

2. The method of claim 1, wherein the storage device includes at least one of a disk array and a multiple-access storage device.

15

3. The method of claim 1, wherein the primary storage, secondary storage and virtual storage dataspace include a primary volume, a secondary volume and one or more virtual volumes.

20

4. The method of claim 2, wherein the disk array is configurable as at least one of a redundant array of independent disks ("RAID") and just a bunch of disks ("JBOD").

25

5. The method of claim 1, wherein the one or more triggers include at least one of a data access command corresponding to the primary storage, a data access command corresponding to the secondary storage and a data replication command.

30

6. The method of claim 5, wherein the data replication command includes a checkpoint command.

7. The method of claim 1, wherein the one or more triggers include a command to initiate an application, the application being conducted, at least in part, by the storage device.

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

8. The method of claim 7, wherein the application includes at least one of data backup, software testing and batch processing.
- 5 9. The method of claim 2, wherein the receiving is conducted, within the storage device, by a virtual volume manager monitoring of commands directed to an array controller.
- 10 10. The method of claim 2, wherein the replicating is conducted, within the storage device, by a virtual volume manager causing an array controller to copy the secondary storage data.
- 15 11. The method of claim 1, further comprising: receiving, by the storage device, one or more second triggers; and replicating at least one of the virtual storage dataspace to the secondary storage.
- 20 12. The method of claim 1, wherein the replicating stores data to be backed up in conjunction with a data backup application, thereby enabling a verifying of the secondary storage data.
- 25 13. The method of claim 1, wherein the replicating stores at least one of a software program and an environment in conjunction with software testing of a software program stored in the secondary storage.
- 30 14. The method of claim 1, wherein the replicating stores a resultant data of at least one batch sub-process in conjunction with batch processing of secondary storage data.
15. The method of claim 1, wherein the step (b) of replicating is replaced by:
determining whether the indicator indicates a security status sufficient for enabling a corresponding data access; and if so, replicating, responsively to the triggers, the

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

secondary storage data to one or more corresponding virtual storage dataspace, thereby enabling the secondary storage to be restored ("rolled back") to the one or more virtual storage dataspace.

- 5 16. A system, comprising:
- (a) means for receiving one or more first triggers by a storage device storing source data in a primary storage and resultant data in a corresponding secondary storage; and
- (b) means for replicating, responsively to the triggers, the secondary
10 storage data to one or more corresponding virtual storage dataspace, thereby enabling the secondary storage to be restored ("rolled back") to the one or more virtual storage dataspace.
- 15 17. The system of claim 16, wherein the storage device includes at least one of a disk array and a multiple-access storage device.
18. The method of claim 16, wherein the primary storage, secondary storage and virtual storage dataspace include a primary volume, a secondary volume and one or more virtual volumes.
- 20 19. The system of claim 16, wherein the one or more triggers include at least one of a data access command corresponding to the primary storage, a data access command corresponding to the secondary storage and a data replication command.
- 25 20. The method of claim 19, wherein the data replication command includes a checkpoint command.
21. The system of claim 16, wherein the one or more triggers include a command to initiate an application, the application being conducted, at least in part, by the
30 storage device.

Express Mail Label No. EL 701375704US Attorney Docket No. 36992.00104
PATENT
HAL ID 248

22. The method of claim 17, wherein the means for receiving provide, within the storage device, for receiving by a virtual volume manager monitoring of commands directed to an array controller.
- 5 23. The method of claim 18, wherein the means for replicating provide, within the storage device, for a virtual volume manager causing an array controller to copy the secondary storage data.
24. The system of claim 16, further comprising: means for receiving, by the
10 storage device, one or more second triggers; and means for replicating at least one of the virtual storage dataspace to the secondary storage.
25. The system of claim 16, wherein the means for replicating store data to be backed up in conjunction with a data backup application, thereby enabling a
15 verifying of the secondary storage data.
26. The system of claim 16, wherein the means for replicating stores at least one of a software program and an environment in conjunction with software testing of a software program stored in the secondary storage.
- 20 27. The system of claim 16, wherein the means for replicating stores a resultant data of at least one batch sub-process in conjunction with batch processing of secondary storage data.
- 25 28. A computer storing program for causing the computer to perform the steps of:
 (a) receiving one or more first triggers by a storage device storing source data in a primary storage and resultant data in a corresponding secondary storage;
 and
 (b) replicating, responsively to the triggers, the secondary storage data to
30 one or more corresponding virtual storage dataspace, thereby enabling the

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

secondary storage to be restored ("rolled back") to the one or more virtual storage
dataspaces.

Express Mail Label No. EL 701375704US PATENT
Attorney Docket No. 36992.00104
HAL ID 248

ABSTRACT OF THE DISCLOSURE

Aspects of the invention provide for the selective replication of data
5 between a secondary volume and one or more virtual volumes of a storage device.
Aspects enable data to be replicated to the virtual volume(s) at one or more
selectable checkpoints, and enable data stored in a secondary volume to be "rolled
back" to prior secondary storage data via replication from the virtual volumes.
Aspects further enable facilitating of various applications or security, among other
10 aspects.

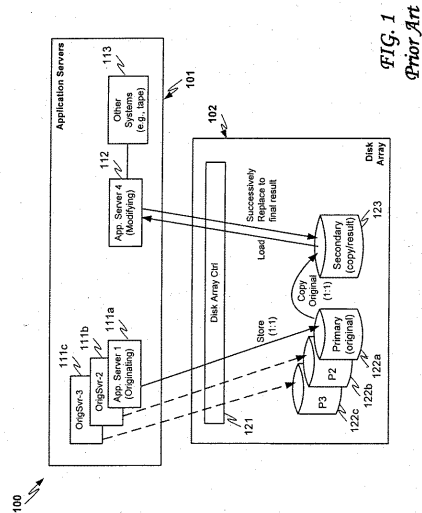
15

20

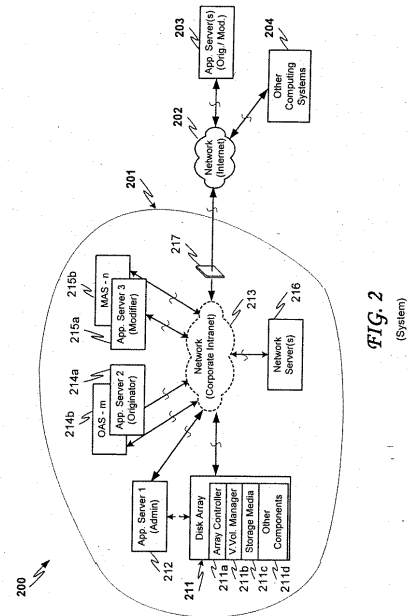
25

30

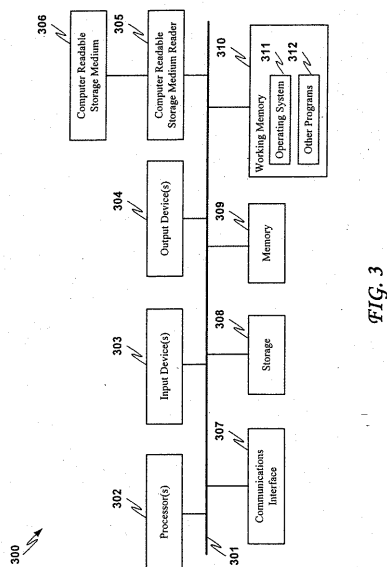
【 図 1 】



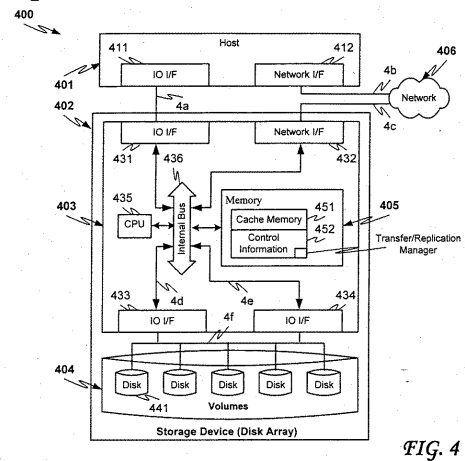
【 図 2 】



【 図 3 】



【 図 4 】



【 図 5 】

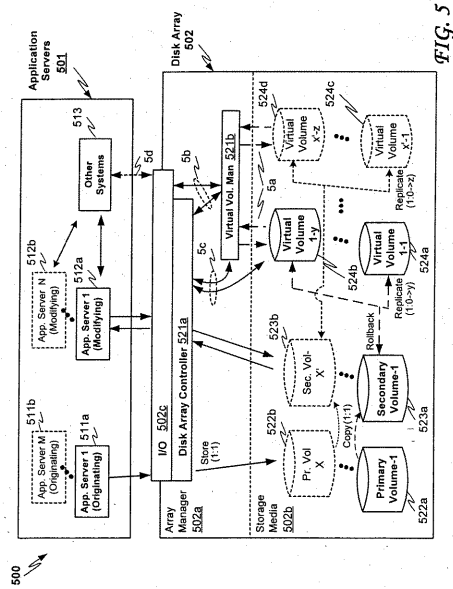


FIG. 5

【 図 6 】

CheckPoint Command	
Command Type	CheckPoint
Volume ID of Secondary Volume	vol23

Rollback and Delete Command (Type1)

Command Type	Rollback or Delete
Volume ID of Secondary Volume	vol23
TimeStamp	1/22/03 8:00AM

Rollback and Delete Command (Type2)

Command Type	Rollback or Delete
Volume ID of Secondary Volume	vol23
Volume ID of Virtual Volume	vol92

It depends on which identification of virtual volume a disk array returns as a result of checkpoint command

FIG. 6

【 図 7 A 】

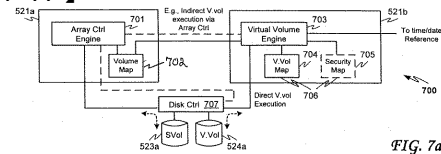


FIG. 7a

【 図 7 B 】

App Svr	Name	Time	Any Crt	2ndary Vol ID	(Srvr Security) Rules/Param.
761	762a	762b	763	764	765

FIG. 7b

【 図 7 C 】

App Svr	Port	Primary Volume ID	Secondary Volume ID	Physical local
702				

FIG. 7c

【 図 8 】

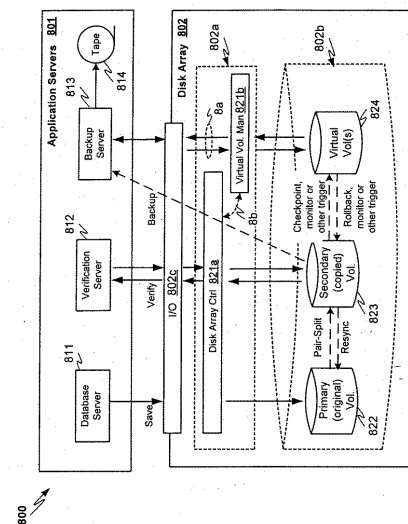


FIG. 8

【 図 9 】

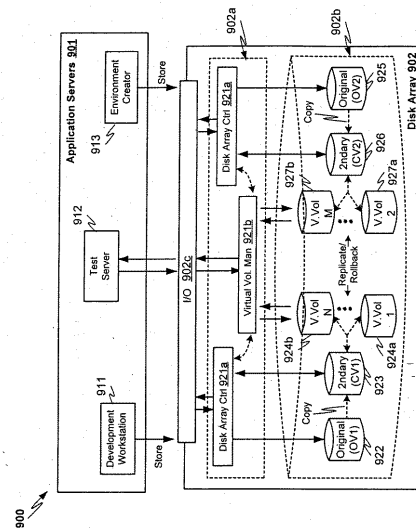


FIG. 9

【 図 10 】

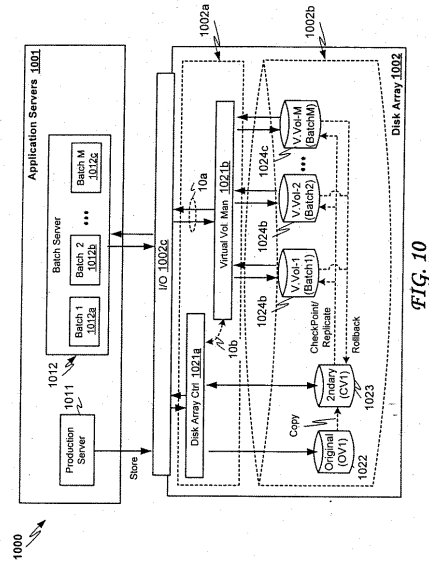


FIG. 10

【 図 11 】

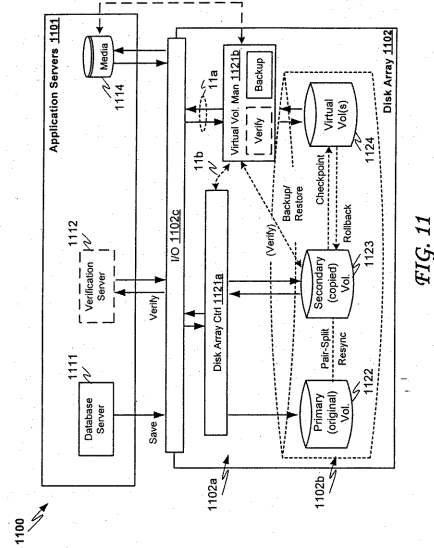


FIG. 11

【 図 12 】

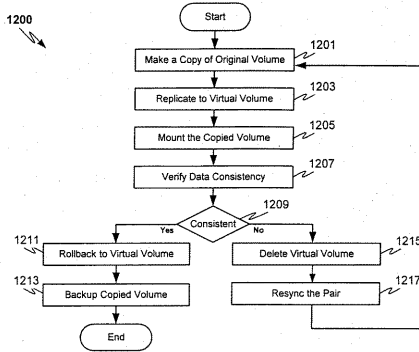


FIG. 12

【 図 13 A 】

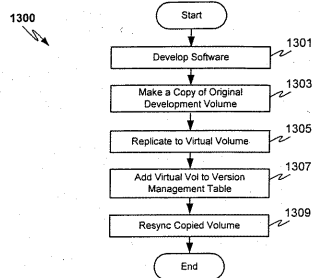


FIG. 13A

【 図 13 B 】

Version	V.Vol ID
1.01	DV1
1.02	DV2
1.03	DV3
⋮	⋮

FIG. 13B

【 図 14 A 】

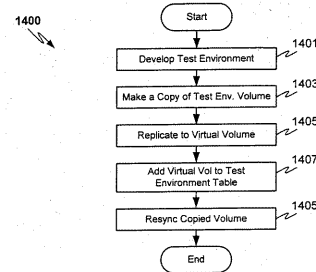


FIG. 14A

【 図 14 B 】

Test Env.	V.Vol ID
1	EV1
2	EV2
3	EV3
⋮	⋮

FIG. 14B

【 図 15 A 】

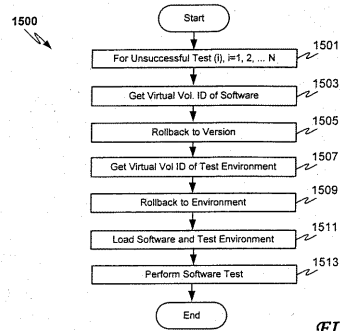


FIG. 15A

【 図 15 B 】

1510

Test No.	Version	Test Env	Successful
1	1.01	EV1	No
2	1.01	EV2	Yes
3	1.01	EV3	No
4	1.02	EV1	No
5	1.02	EV2	No
6	1.02	EV3	Yes
4	1.03	EV1	No
5	1.03	EV2	No
6	1.03	EV3	Yes

FIG. 15B

【 図 16 】

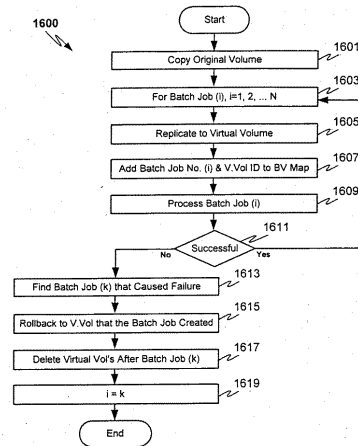


FIG. 16

【 図 17 A 】

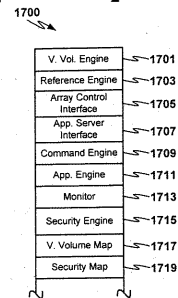


FIG. 17A

【 図 17 B 】

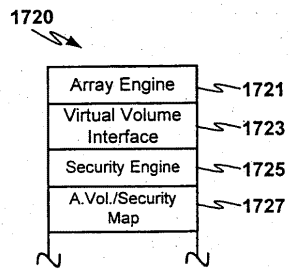


FIG. 17B