

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4897885号
(P4897885)

(45) 発行日 平成24年3月14日 (2012. 3. 14)

(24) 登録日 平成24年1月6日 (2012. 1. 6)

(51) Int. Cl.

F I

G 0 6 F 12/00 (2006. 01)

G 0 6 F 12/00 5 1 7

G 0 6 F 17/30 (2006. 01)

G 0 6 F 12/00 5 2 0 A

G 0 6 F 17/30 4 1 9 A

請求項の数 7 (全 27 頁)

(21) 出願番号 特願2009-527124 (P2009-527124)
 (86) (22) 出願日 平成19年9月4日 (2007. 9. 4)
 (65) 公表番号 特表2010-503094 (P2010-503094A)
 (43) 公表日 平成22年1月28日 (2010. 1. 28)
 (86) 国際出願番号 PCT/EP2007/059214
 (87) 国際公開番号 W02008/031741
 (87) 国際公開日 平成20年3月20日 (2008. 3. 20)
 審査請求日 平成22年5月24日 (2010. 5. 24)
 (31) 優先権主張番号 11/531, 130
 (32) 優先日 平成18年9月12日 (2006. 9. 12)
 (33) 優先権主張国 米国 (US)

早期審査対象出願

(73) 特許権者 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレーション
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 アメリカ合衆国10504 ニューヨーク州 アーモンク ニュー オーチャードロード
 (74) 代理人 100108501
 弁理士 上野 剛史
 (74) 代理人 100112690
 弁理士 太佐 種一
 (74) 代理人 100091568
 弁理士 市位 嘉宏

最終頁に続く

(54) 【発明の名称】 データを管理するための方法

(57) 【特許請求の範囲】

【請求項 1】

コントローラと、前記コントローラによってアクセス可能なメモリと、前記コントローラによってアクセス可能なストレージ・ユニットとを備え、前記ストレージ・ユニットが、更新されたデータの複数のバージョンを維持するための継続的データ保護 (CDP) システムを構成するデータ処理システムにおいて、データを管理するための方法であって、
 (a) 前記コントローラによって、前記更新されたデータを前記ストレージ・ユニットに書き込むことを含むそれぞれのデータ更新操作を実行するステップと、
 (b) 前記コントローラによって、前記更新操作を表す第1のデータ構造内の第1のエントリを確立するステップとを含み、

前記第1のエントリは、前記更新操作の更新時間を表すタイムスタンプを含み、前記第1のエントリは、論理ブロック・アドレスを含むキーによってインデックスされ、
 (c) 前記コントローラによって、テーブルを構成する第2のデータ構造内の第2のエントリを確立するステップを含み、

前記第2のエントリの各々は、開始時間と、終了時間と、差し戻し時間と、前記開始時間及び前記終了時間によって定められる時間インターバルとを有し、前記第2のエントリは、親エントリ及び子エントリを含み、前記子エントリの各々ごとに、その前記差し戻し時間はその親エントリの前記時間インターバルの範囲内に含まれ、前記第2のエントリの各々は、その前記時間インターバル中に生じる1組の更新操作に対応し、前記第2のエントリは、初期親エントリを含み、

10

20

(d) 前記コントローラによって、時間の経過に伴う前記更新データの第1の進化を定める、更新操作の初期シーケンスを定めるステップを含み、

その前記更新時間は、前記初期親エントリの前記時間インターバル中に生じ、

(e) 前記コントローラによって、所与の時点において開始する前記初期シーケンスを更新するための差し戻し要求に応答して、初期子エントリを前記テーブルに追加するステップを含み、

その前記差し戻し時間は、前記所与の時点にセットされ且つ前記初期親エントリの前記時間インターバルの範囲内に含まれ、

(f) 前記コントローラによって、前記第1のデータ構造を再帰的に走査することにより、前記初期シーケンスを更新操作の差し戻しシーケンスに差し戻すステップを含み、

前記差し戻しシーケンス内の更新時間は、前記初期親エントリの前記開始時間に開始し且つ前記初期子エントリの前記差し戻し時間に終了する第1の差し戻し時間インターバル中に生じるか、又は前記初期子エントリの前記開始時間に開始する第2の差し戻し時間インターバル中に生じ、

(g) 前記コントローラによって、前記テーブル内に新しい子エントリを確立することができる時間の関連ウインドを定めるステップを含み、

その前記差し戻し時間は、前記関連ウインドの範囲内に含まれ、

(h) 前記コントローラによって、更新時間が前記関連ウインドの範囲内に含まれる第1の更新操作を特定することにより、前記ストレージ・ユニットにおけるデータのアクセス可能性を判定するとともに、前記差し戻しシーケンス内の第2の更新操作を特定するステップと、

(i) 前記コントローラによって、前記第1の更新操作及び前記第2の更新操作の前記更新されたデータをアクセス可能なデータとして指定するステップと、

(j) 前記コントローラによって、前記アクセス可能なデータ以外の前記更新されたデータを前記ストレージ・ユニットから削除するステップと、

(k) 前記ストレージ・ユニット内のデータを前記差し戻しシーケンスでアクセスするステップとをさらに含む、方法。

【請求項2】

前記ステップ(b)が、B木のノードとしてB木のエントリを確立することを含む、請求項1に記載の方法。

【請求項3】

前記ステップ(k)が、

所与のキーと関連する前記更新されたデータの一部をアクセスするための検索タイムスタンプを有する検索要求を受信することと、

前記第1のエントリのうち、前記初期親エントリの前記時間インターバル中に生じるか、又は前記初期子エントリの前記時間インターバル中に生じる更新操作を有するエントリを参照することにより、前記検索タイムスタンプにおける有効な更新されたデータを特定することを含む、請求項1に記載の方法。

【請求項4】

前記ステップ(h)が、前記第2のエントリのうち1つのエントリの前記時間インターバルの終了部分を特定するための粗分析を実施することを含み、

前記終了部分が、前記関連ウインドウの外部にあり且つ前記1つのエントリの子エントリの最後の差し戻し時間において開始する、請求項1に記載の方法。

【請求項5】

前記コントローラによって、前記第1のエントリのうち少なくとも1つのサブセットを走査することにより、特定のキーの組み合わせを前記第2のエントリのうち前記1つのエントリと関連付け且つ当該1つのエントリの子孫エントリを再帰的に走査するための精密分析を実施するステップをさらに含む、請求項4に記載の方法。

【請求項6】

前記コントローラによって、前記第2のエントリのうち特定のエントリであって、その

10

20

30

40

50

前記時間インターバルが前記関連ウィンドウの外部にあり且つ子エントリを持たない、特定のエントリを識別するステップと、

前記コントローラによって、前記テーブルから前記識別された特定のエントリを削除し且つ前記ストレージ・ユニットから前記識別された特定のエントリに対応する前記第 1 のエントリのうち特定のエントリによって表される前記更新されたデータを削除するステップとをさらに含む、請求項 1 に記載の方法。

【請求項 7】

前記ストレージ・ユニットが、前記複数のバージョンを格納するように編成された第 1 及び第 2 のストレージ・モジュールから成り、

前記ステップ (d) が、

前記複数のバージョンのうち前記第 1 のストレージ・モジュールから前記第 2 のストレージ・モジュールに転送することが必要とされる特定のバージョンを表すための送りデータ構造を維持することと、

前記複数のバージョンのうち最近更新されたバージョンが前記第 1 又は第 2 のストレージ・モジュールのどちらに存在するかを表すための戻しデータ構造を維持することと、

前記送りデータ構造及び前記戻しデータ構造内の情報に応答して、前記複数のバージョンのうち選択されたバージョンを前記第 1 のストレージ・モジュールから前記第 2 のストレージ・モジュールに転送することと、

前記複数のバージョンのうち前記選択されたバージョンに対応する前記更新時間を反映するように、前記第 1 のエントリのうち対応するエントリの前記キーを更新することによって前記第 1 のデータ構造をリフレッシュすることとを含む、請求項 1 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データを管理するための方法に関する。

【背景技術】

【0002】

データは、時間の経過と共に進化する場合がある。多くの用途において、データの以前のバージョンを取得する必要がある。1つの解決策は、継続的データ保護 (CDP) である。特許文献 1 及び特許文献 2 は、継続的データ保護についての従来技術の装置及び方法を説明している。

【0003】

時間の経過に伴うデータの進化を追跡するために、種々のデータ構造が提案された。以下の論文、すなわち非特許文献 1 及び非特許文献 2 は、B T 木及び B T R 木として知られる 2 つのデータ構造を説明する。

【先行技術文献】

【特許文献】

【0004】

【特許文献 1】米国特許出願公開第 2 0 0 5 / 0 0 6 6 1 1 8 号

【特許文献 2】米国特許出願公開第 2 0 0 5 / 0 1 9 3 2 7 2 号

【非特許文献】

【0005】

【非特許文献 1】"The BT-Tree: A Branched and Temporal Access Method", Jiang, Salzberg, Lomet, Barrena, Proceedings of the 26th VLDB Conference, 2000

【非特許文献 2】"The BTR-Tree: Path-Defined Version-Range Splitting in a Branched and Temporal Structure", Jiang, Salzberg, Lomet, Barrena, Symposium for Advances in Spatial and Temporal Databases, 2003

【発明の概要】

【発明が解決しようとする課題】

【0006】

これらの分岐時間インデックス (branched temporal indexes) は、特定の前提が満たされる場合には維持することができる。第 1 の前提によれば、分岐時間インデックスのエントリは、タイムスタンプの増加する順序で挿入される。

【 0 0 0 7 】

さらに、これらのデータ構造は標準的なものではなく、これらのデータ構造の更新は、比較的複雑であり、データのみならずメタデータを複製することを含む。さらに、これらのデータ構造の維持には、参照カウンタを必要とする場合がある。

上述された問題を解決する必要性はますます高まっている。

【課題を解決するための手段】

【 0 0 0 8 】

データを管理するための方法は、キーによって特定されるエントリへの挿入操作又は更新操作を表す第 1 のデータ構造を準備するステップと、分岐作成操作を表す第 2 のデータ構造を準備するステップとを含む。第 2 のデータ構造は分岐識別子及び分岐タイミング情報を格納する。本方法は、読み取りタイムスタンプにおけるデータのバージョンを検索する要求を受信するステップと、データのそのバージョンを探し出すために第 1 のデータ構造及び第 2 のデータ構造を走査するステップと、要求されたタイムスタンプにおけるデータのバージョンによって開始する分岐を作成する要求を受信し、それに応じて第 2 のデータ構造を更新するステップとをさらに含む。

【 0 0 0 9 】

好ましくは、第 1 のデータ構造は B 木である。

好ましくは、本方法は、所与のキーと関連する値を更新する要求を受信するステップと、第 1 のデータ構造を更新するステップとを含む。

好ましくは、更新するステップは、所与のキー及び関連するデータ並びに現在のタイムスタンプを持つ新しいエントリを追加することによって実施される。

好ましくは、キーは論理ブロック・アドレスである。

好ましくは、分岐を作成する要求は、差し戻し操作 (revert_operation) であり、特定のタイムスタンプは、差し戻し時間 (revert-to time) である。

好ましくは、分岐を探すことは、分岐タイミング情報を走査するステップを含む。

好ましくは、本方法は、データ・ストレージ・ポリシーに応じて第 1 のデータ構造及び第 2 のデータ構造をリフレッシュするステップを含む。

好ましくは、リフレッシュするステップは、第 2 のデータ構造のエントリの粗分析 (coarse_analysis) を実施することを含む。

好ましくは、リフレッシュするステップは、第 1 のデータ構造及び第 2 のデータ構造のエントリの精密分析 (fine analysis) を実施することを含む。

好ましくは、リフレッシュするステップは、少なくとも 1 つの関連ウィンドウ (relevance_window) に応じて行われる。

好ましくは、リフレッシュするステップは、少なくとも 1 つのデータ・ストレージ粒度 (data_storage granularity) に応じて行われる。

好ましくは、リフレッシュするステップは、現時点からの時間経過に対応する多数のデータ・ストレージ粒度に応じて行われる。

好ましくは、リフレッシュするステップは、第 1 のデータ構造のエントリと第 1 のデータ構造のエントリのサブセットとの間の関係に応じて行われる。

好ましくは、リフレッシュするステップは、第 1 のデータ構造の 1 つのエントリと第 1 のデータ構造のエントリ群との間の関係に応じて行われる。

好ましくは、分岐タイミング情報は、分岐開始時間と、分岐終了時間と、要求されたタイムスタンプとを含む。

【図面の簡単な説明】

【 0 0 1 0 】

【図 1】本発明の 1 つの実施形態による、データを管理するためのシステムを示す。

【図 2】本発明の 1 つの実施形態による、ステージング操作及びデステージング操作の例

10

20

30

40

50

示的なシーケンスを示す。

【図 3】本発明の別の実施形態による、データを管理するためのシステムを示す。

【図 4】本発明の別の実施形態による、ステージング操作及びデステージング操作の例示的なシーケンスを示す。

【図 5】本発明の更なる実施形態による、データを管理するためのシステムを示す。

【図 6】本発明の別の実施形態による、読み書きステージング操作及びデステージング操作の例示的なシーケンスを示す。

【図 7】本発明のさらに別の実施形態による、読み書きステージング操作及びデステージング操作の例示的なシーケンスを示す。

【図 8】本発明の更なる実施形態による、読み書きステージング操作及びデステージング操作の例示的なシーケンスを示す。 10

【図 9】本発明のさらにまた別の実施形態による、読み書きステージング操作及びデステージング操作の例示的なシーケンスの図である。

【図 10】本発明の 1 つの実施形態による、第 1 のデータ構造の一部及び第 2 のデータ構造の一部を示す。

【図 11】本発明の 1 つの実施形態による、更新操作及び差し戻し操作の例示的なシーケンスを示す。

【図 12】本発明の 1 つの実施形態による、更新操作及び差し戻し操作の例示的なシーケンスを示す。

【図 13】本発明の別の実施形態による、更新操作及び差し戻し操作の例示的なシーケンスを示す。 20

【図 14】本発明の 1 つの実施形態による、データ・バージョンを取得するための方法を示す。

【図 15】本発明の 1 つの実施形態による粗分析を示す。

【図 16】本発明の 1 つの実施形態による全体精密分析を示す。

【図 17】本発明の 1 つの実施形態による全体精密分析を示す。

【図 18】本発明の 1 つの実施形態による、データを管理するための方法を示す。

【発明を実施するための形態】

【0011】

本発明は、添付の図面と併せて、以下の詳細な説明から、より完全に理解され認識される。 30

【0012】

本発明は、データの多数のバージョンを格納及び取得するための方法、システム、及びコンピュータ・プログラムを提供する。

【0013】

本発明の 1 つの実施形態によれば、データ及びデータと関連するメタデータが提供される。データは、データの少なくとも 1 つの現在のバージョンと、データの少なくとも 1 つの以前のバージョンとを含むことができる。メタデータは、2 つのデータ構造に編成されることが好ましい。第 1 のデータ構造は、データ更新操作に関する情報を含む。第 1 のデータ構造は、拡張された B 木データ構造であることが好ましい。第 2 のデータ構造は、差し戻し操作に関する情報を含む（さらにクローン作成情報を含む場合もある）。第 2 のデータ構造は、分岐テーブルとも呼ばれる。これらの 2 つのデータ構造が用いられるときに、効率的なデータ更新操作、検索操作、及び差し戻し操作を適用することが可能となる。 40

【0014】

好ましくは、更新操作は、B 木データ構造の中のエントリによって表され、一方、差し戻し操作は、第 2 のデータ構造の中のエントリによって表される。検索操作は、要求されたデータを見つけるために、両方のデータ構造を調べることを含むものとして行うことができる。

【0015】

本発明の 1 つの実施形態によれば、第 1 のデータ構造は、第 2 のデータ構造よりはるか 50

に大きい。第1のデータ構造のエントリは、書き込み操作ごとに、又はデステージング操作（destage operation）ごとに、（書き込み又はデステージングのアドレスに応じて）割り当てられ、一方、第2のデータ構造のエントリは、差し戻し操作ごとに割り当てられる。典型的には、差し戻し操作の数は、書き込み操作の数よりはるかに小さい。差し戻し操作によって開始された分岐は、書き込み／デステージング操作に応じて、第1のデータ構造の中の多数のエントリと関連付けることができる。2つの異なるデータ構造を用いることによって、第1のデータ構造（B木）の更新を簡略化することができ、標準的なB木更新アルゴリズムを用いることができる。好ましくは、B木は、タイムスタンプの順序から外れたエントリを挿入することによって更新することができる。

【0016】

10

時間の経過に伴うデータの進化、特に更新操作及び差し戻し操作は、グラフによって表される。グラフは、1つ又は複数の分岐を含むものとすることができるが、（親分岐とも呼ばれる）最初の分岐は、更新操作の第1のシーケンスを示す。各々の差し戻し操作は、多数のエントリを含む場合がある分岐を開始し、各々のエントリは、キーへの更新操作を表す。1つの分岐は、新しい差し戻し操作が実行されたときに終了することができる。グラフは、第1のデータ構造及び第2のデータ構造の内容を示す。

【0017】

好ましくは、2つのデータ構造は、継続的データ保護（CDP）システムに実装され、それによりCDPシステムは、格納されたデータを特定の時点から容易に再構築することができる。差し戻し操作は迅速であり、他の差し戻し操作の実行を妨げない。

20

【0018】

メタデータ及びデータは、検索操作、更新操作、及び差し戻し操作の際には複製されない。このことは、データ及びメタデータのサイズを大きく低減する。好ましくは、分岐間におけるデータの共有は最大となり、参照カウントを回避することができる。

【0019】

本発明の1つの実施形態によれば、いかなる時点においてもアクティブな分岐が1つだけ存在することができる。本発明の別の実施形態によれば、多数のアクティブな分岐が同時に存在することができる。データを管理するのに必要なメタデータは、前者の場合と後者の場合とで異なるものとすることができる。

【0020】

30

本発明の1つの実施形態によれば、第1のデータ構造の各々のエントリは、論理ブロック・アドレス（L）と、書き込み操作が実行された時点を表す書き込みタイムスタンプ（WTS）と、論理ブロック・アドレスと関連する物理アドレス（P）と、を含む。

【0021】

本発明の別の実施形態によれば、第1のデータ構造の各々のエントリは、論理ブロック・アドレス（L）と、現在のデータを上書きする後続の書き込み操作が実行された時点を表す上書きタイムスタンプ（OTS）と、論理ブロック・アドレスと関連する物理アドレス（P）と、を含む。

【0022】

本発明の別の実施形態によれば、各々のエントリは、論理ブロック・アドレスと、オフセットと、長さ、を含む。

40

【0023】

好ましくは、論理ブロック・アドレスは、第1のデータ構造にアクセスするのに用いられるキーを形成する。好ましくは、各々のエントリは、特に多数のアクティブな分岐が同時に存在することができる場合には、分岐識別子（BID）も含む。

【0024】

第1のデータ構造には、キーによってアクセスすることができる。1つの実施形態においては、キーは、論理ブロック・アドレスを含む。強化されたキーは、BID及び／又はタイムスタンプを含むこともできる。

【0025】

50

本発明の別の実施形態によれば、第2のデータ構造の各々のエントリは、分岐識別子（BID）と、分岐開始時間と、分岐終了時間と、分岐差し戻し時間と、を含む。好ましくは、各々のエントリは、親分岐の分岐識別子と、分岐識別子も含む。

【0026】

更新要求は、キーと、書き込まれるデータと、を含む。好ましくは、多数のアクティブな分岐が同時に共存しうる場合には、強化されたキーが必要となることがある。1つの実施形態について、書き込み操作は、第1のデータ構造を更新することと、そのデータを、キーに含まれる論理ブロック・アドレスと関連付けられた物理ブロック・アドレス（P）に書き込むこととを含む。

【0027】

差し戻し要求は、分岐開始時間（これは、差し戻し要求が実行された時間とすることも、差し戻し要求が受信された時間とすることもできる）と、差し戻し時間と、必要に応じて親分岐識別子と、を含む。好ましくは、所与の時点においてアクティブな分岐が1つしか存在できない場合には、親分岐識別子を設ける必要はない。差し戻し操作は、第2のデータ構造を更新することを含む。

【0028】

検索要求は、キーと、検索タイムスタンプと、を含む。多数のアクティブな分岐が同時に共存しうる場合には、強化されたキーが必要となることがある。検索操作は、所与の検索タイムスタンプにおいて有効なエントリを探すことを含む。これは、適切なエントリが見つかるか、又は、親を持たない分岐が見つかり、未検出との回答が得られるまで、要求された分岐から開始し、その親分岐に続くなどといった通常のB木検索のシーケンスを含むことができる。

【0029】

本発明の種々の実施形態によれば、データ・ストレージ・ポリシーを適用することができる。第1の例示的なデータ・ストレージ・ポリシーは、関連ウィンドウを定める。関連ウィンドウ内にあるタイムスタンプにのみ差し戻すことが可能である。関連ウィンドウ内においてアクセス可能なデータと、そのデータに関連するメタデータとは、維持されるべきであり、一方、関連ウィンドウにおいてアクセス可能でないデータと、関連するメタデータとは、削除することができる。タイムスタンプが関連ウィンドウの外部にあるデータが、差し戻し操作及び検索操作のシーケンスの結果としてアクセス可能となることもあるので、データ及び関連するメタデータを削除する前に、検索操作及び差し戻し操作に対するそのデータのアクセス可能性がチェックされることに留意されたい。さらに別のデータ・ストレージ・ポリシーは、多数の関連ウィンドウを定めることができる。

【0030】

好ましくは、データの全ての古いバージョンが格納されるとは限らない。例えば、データ・ストレージ粒度を定めることができる。データ・ストレージ粒度は、データの1つのバージョンによって表される（又は、データのいかなるバージョンによっても表されない）期間を定める。その期間中にデータの多数のバージョンが生成された場合には、これらのバージョンのうちの1つが格納される。本発明の別の実施形態によれば、データ・バージョンは、イベントが生成されたいずれかのタイムスタンプについて格納されることに留意されたい。好ましくは、イベントが生成されたいずれかのタイムスタンプについて1つのバージョンが格納される。

【0031】

本発明の1つの実施形態によれば、データ・ストレージ・ポリシーは、多数のデータ・ストレージ粒度を定める。典型的には、データ・ストレージ粒度は、データ・バージョンが古くなるに従って粗くなる。データ・ストレージ粒度は、各々の関連ウィンドウごとに固定することができるが、必ずしもそうしなくてもよい。

説明を簡単にするために、単一の関連ウィンドウが定められると仮定する。

【0032】

本発明の1つの実施形態によれば、データ・ストレージ・ポリシーは、差し戻し操作及

10

20

30

40

50

び検索操作のいくつかのシーケンスの後で、どのエントリがアクセス可能であることを判定する。種々のアクセス可能性検出プロセスを適用することができ、各々のプロセスは、範囲と複雑度（又は実行時間）との間のトレードオフを与える。

【 0 0 3 3 】

好ましくは、アクセス可能性検出プロセスは、粗分析と精密分析とを含む。好ましくは、粗分析は分岐全体を処理することを含み、一方、精密分析はエントリを処理することを含む。

【 0 0 3 4 】

粗分析は、分岐を除去すること、又は、分岐を統合することを含むことができる。粗分析は、完全に関連ウィンドウの外部にあり、その分岐のいずれかの子の最後の差し戻し時間において開始する分岐終了セクションを、除去することを含むこともできる。

10

【 0 0 3 5 】

好ましくは、分岐は、完全に関連ウィンドウの外部にあり、関連する子分岐を持たない場合には、除去される。分岐が、完全に関連ウィンドウの外部にあるが、子分岐を1つ有する場合には、その分岐と子分岐とは、その分岐を子分岐に移動させることによって統合することができる。

【 0 0 3 6 】

粗分析は第2のデータ構造を走査することを含み、一方、精密分析は第1のデータ構造と第2のデータ構造の両方を走査することを含む。

精密分析は、1度に第1のデータ構造の1ページを走査し、キーを1つずつ走査することが好ましい。単一のキーと関連する分岐は、1つのエントリずつ走査される。

20

【 0 0 3 7 】

本発明の1つの実施形態によれば、精密分析には少なくとも2つのタイプが存在する。第1のタイプは全体精密分析と呼ばれ、他のタイプは局所精密分析と呼ばれる。所与のキーについて、全体精密分析は、第1のデータ構造のエントリのすべてをこのキーを用いて走査できる可能性があり、一方、局所精密分析は、これらのエントリの一部（サブセット）のみを走査する。

【 0 0 3 8 】

好ましくは、局所精密分析は、アクセス可能なエントリ数がアクセス不可能なエントリ数を超えることを保証することができる。これらの上述された分析に代わる分析を適用することに留意されたい。

30

【 0 0 3 9 】

好ましくは、全体精密分析は、特定のキーと特定の分岐との組み合わせの各々について、その特定の分岐の子孫分岐における更新操作を表すエントリを再帰的に走査することによって、この組み合わせを含む第1のデータ構造のエントリがアクセス可能なエントリであるかどうかをチェックする。

【 0 0 4 0 】

好ましくは、局所精密分析は、特定のキーと特定の分岐との組み合わせの各々について、その特定の分岐の子分岐における更新操作を表すエントリを走査することによって、この組み合わせを含む第1のデータ構造のエントリがアクセス可能なエントリであるかどうかをチェックする。

40

アクセス不可能なエントリは削除することができるが、現れたエントリは残すべきである。

【 0 0 4 1 】

好ましくは、各々の分岐について第1のデータ構造内のエントリ数は維持され、対応するエントリを持たないときには、その分岐は除去される。

【 0 0 4 2 】

本発明の種々の実施形態によれば、ストレージ空間再利用プロセスは、種々の方法で開始させることができ、即ち、継続的に、周期的に、擬似ランダムに、ランダムに、イベントの発生に応じて（例えばユーザのデータ又はメタデータのサイズ制限や、特定の数の検

50

索操作、更新操作、及び／又は差し戻し操作の実行に応じて）、などといった方法で、実行することができる。ストレージ空間再利用プロセスの１つの実行から別の実行までの時間経過は、１つ又は複数の関連ウィンドウのサイズに対応するものとすることができる。好ましくは、プロセスは、イベントが発生したときに開始される。イベントは、第１のデータ構造のページが満たされるか又は別のページに統合されるなどのときに関連ウィンドウから分岐が除外されることを含むことができる。

【 0 0 4 3 】

説明を簡単にするために、以下の図面は、第２のデータ構造が差し戻し操作を表し、差し戻し操作は差し戻し時間におけるデータ・バージョンに対して行われ、書き込み操作のために用いられるキーは論理ブロック・アドレスである、本発明の特定の実施形態を示す。本方法、システム、及びコンピュータ可読媒体は、第２のデータ構造が（差し戻し、既存の分岐の複製などといった）分岐作成操作を表すときには準用が可能であること、分岐作成操作は、要求されたタイムスタンプにおけるデータ・バージョンによって開始する分岐を作成すること（要求されたタイムスタンプは、データのバージョンと関連付けられ、必ずしも分岐の作成時間を反映する必要はない）、及び、キーが論理ブロック・アドレスと異なるものとすることができることを留意されたい。

【 0 0 4 4 】

図１は、本発明の１つの実施形態による、データを管理するためのシステム１００を示す。システム１００は分散型アーキテクチャを有する。システム１００は、アダプタ１４３、第１のストレージ・ユニット１２１、及び第２のストレージ・ユニット１２２に接続された、第１の書き戻し（write back）キャッシュ・ユニット１３１を含む。

【 0 0 4 5 】

システム１００はコントローラ１５１を含むが、コントローラは、第１の書き戻しキャッシュ・ユニット１３１内に含むこともできる。第１の書き戻しキャッシュ・ユニット１３１は、データを読み取る要求を受信し、データを書き込む要求を受信し、他の装置に接続することができるアダプタ１４３からのデータを受信する。第１の書き戻しキャッシュ・ユニット１３１は、アダプタ１４３にデータを出力する。

第１の書き戻しキャッシュ・ユニット１３１は、多数のインターフェース、アダプタなどに接続できることに留意されたい。

【 0 0 4 6 】

システム１００は、４つのデータ構造、即ち（*i*）書き込み操作を表す第１のデータ構造２００、（*i i*）差し戻し操作を表す第２のデータ構造２５０、（*i i i*）どのデータを第１のストレージ・ユニットから第２のストレージ・ユニットに送るかを示す第３のデータ構造（送りデータ構造（forward data structure）とも呼ばれる）２６０、及び、（*i v*）最近更新されたデータのバージョンが第１のストレージ・ユニット１２１にあるのか、第２のストレージ・ユニット１２２にあるのかを示す第４のデータ構造（戻しデータ構造（backward data structure）とも呼ばれる）２７０を維持する。

【 0 0 4 7 】

これらのデータ構造の各々は、システム１００の種々の位置に格納することができ、例えば、全てのデータ構造２００、２５０、２６０、及び２７０を第２のストレージ・ユニット１２２に格納することができるが、必ずしもそうである必要はない。

【 0 0 4 8 】

第３のデータ構造２６０は、デステージング期間が開始したとき、及び、データ・バージョンが第１のストレージ・ユニットから第２のストレージ・ユニットにコピーされたときに、更新される。

【 0 0 4 9 】

好ましくは、（１つのデータ割り当てユニットごとに）１つのデータ・バージョンのみがコピーされる。現在のキャッシュ済データ・バージョンがコピーされるべきかどうかを知るために、第１のデータ構造が走査される場合がある。この走査を避けるために、第３のデータ構造２６０は、現在のキャッシュ済データがコピーされるべきかどうかを示す。

【 0 0 5 0 】

好ましくは、第3のデータ構造260は、データ割り当てユニットごとに1ビットを含む。このビットは、現在のキャッシュ済データ・バージョンと新しいデータ・バージョンと（どちらも同じキャッシュ・データ割り当てユニットに関連する）が同じデステージング期間に属するかどうかを示す。答えが肯定の場合には、現在のキャッシュ済データ・バージョンはコピーされない。第3のデータ構造のビットは、現在のデステージング期間ごとに最初のキャッシュ操作が生じたときにセットされる。そのデステージング期間の最後に、ビットはリセットされる。

【 0 0 5 1 】

第4のデータ構造270は、差し戻し操作並びに差し戻し操作に続くステージング操作及びデステージング操作への応答として更新される。

10

【 0 0 5 2 】

差し戻し操作は、第2のストレージ・ユニットから第1のストレージ・ユニットにデータが送られることを必要とする。このプロセスは時間がかかるものであり、このプロセス中にシステムがデータ要求に対応できるようにするために、第4のデータ構造が用いられる。

【 0 0 5 3 】

（特定の時点への）差し戻し操作の後には、最近更新されたデータ・バージョンはその時点に対応していることに留意されたい。従って、これらは、デステージングされたデータ・バージョン及びより古いデータ・バージョンへの差し戻し操作の前に考慮されたデータ・バージョンとすることができる。

20

【 0 0 5 4 】

第4のデータ構造270は、データ割り当てユニットごとに1ビットを含むことができる。差し戻し操作が要求された場合には、このビットをセットすることができる。第2のストレージ・ユニットのデータ割り当てユニットに格納されたデータが第1のストレージ・ユニットに格納されると、ビットはリセットされる。

【 0 0 5 5 】

データは、第1の書き戻しキャッシュ・ユニット131にキャッシュし、次いで第1のストレージ・ユニット121にデステージングすることができる。第1のストレージ・ユニット121からのデータは、第1の書き戻しキャッシュ・ユニット131を経由して、第2のストレージ・ユニット122に送ることができる。第1のストレージ・ユニット121は、ステージングされた（現在の）データ・バージョンを格納し、第2のストレージ・ユニット122は、より古いデータ・バージョンを格納する。差し戻しは、第1のストレージ・ユニット121及び/又は第2のストレージ・ユニット122からのデータをステージングすることを要求することができる。

30

【 0 0 5 6 】

コントローラ151は、(i)特定のキャッシュ・データ割り当てユニットに新しいデータ・バージョンを書き込む要求を受信し、(ii)データ・ストレージ・ポリシーに応じて、特定のキャッシュ・データ割り当てユニットに現在キャッシュされているキャッシュ済データ・バージョンを上書きするか、又は、特定のキャッシュ・データ割り当てユニットに新しいデータ・バージョンを書き込む前に、キャッシュ済データ・バージョンの第1のストレージ・ユニットへのデステージングを行うかを判定し、(iii)特定の時点に対応するデータ・バージョンを読み取る要求を受信し、(iv)要求されたデータ・バージョンの位置を求めるために、書き込み操作を表す第1のデータ構造と、差し戻し操作を表す第2のデータ構造とを走査するように、適合させることができる。第1のデータ構造及び第2のデータ構造は、図10に示される。

40

【 0 0 5 7 】

コントローラ151は、システム100の種々の構成要素を制御して、要求されたデータ・バージョンを提供し、（コントローラの判定に応じて）選択的に強制デステージングを実施する。

50

【 0 0 5 8 】

好ましくは、第 1 の書き戻しキャッシュ・ユニット 1 3 1 は、同一のデータの多数のバージョンを格納するようには適合されない。したがって、新しいデータ・バージョンがキャッシュされる前に、コントローラ 1 5 1 は、キャッシュ済データ・バージョンをデータ・ストレージ・ユニットに保存すべきかどうかを決定しなければならない。決定は、データ・ストレージ・ポリシーに応じて行われる。

【 0 0 5 9 】

別の実施形態によれば、第 1 の書き戻しキャッシュ・ユニット 1 3 1 は、同一のデータの多数のバージョンを格納するようには適合される。

【 0 0 6 0 】

システム 1 0 0、特にコントローラ 1 5 1 は、インターセプトを用いることによって、第 1 の書き戻しキャッシュ・ユニット 1 3 1 の操作を制御することができる。

【 0 0 6 1 】

システム 1 0 0 は、少なくとも 1 つのデータの関連ウィンドウを定めるデータ・ストレージ・ポリシーを定めるか、部分的に定めるか、部分的に受信するか、又は受信することができる。関連ウィンドウは、第 1 の書き戻しキャッシュ・ユニット 1 3 1 と、第 1 のストレージ・ユニット 1 2 1 及び第 2 のストレージ・ユニット 1 2 2 との間のステージング操作及びデステージング操作に影響を与える。

【 0 0 6 2 】

図 2 は、本発明の 1 つの実施形態による、単一の割り当てユニットの単一のデータ・バージョンに対するステージング操作及びデステージング操作の例示的なシーケンスを示す。

シーケンスは、新しいデータ・バージョンを書き込む要求をアダプタ 1 4 3 から第 1 の書き戻しキャッシュ・ユニット 1 3 1 の特定のキャッシュ・データ割り当てユニットに送るステップ（文字 A により示される）によって開始する。その特定のキャッシュ割り当てユニットに現在キャッシュされているキャッシュ済データ・バージョンを（データ・ストレージ・ポリシーに従って）デステージングすべき場合には、そのバージョンは、第 1 のデータ・ストレージ・ユニット 1 2 1 にデステージングされ（文字 B によって示される）、デステージングされたデータ・バージョンとなる。このデステージング・ステップの後、新しいデータ・バージョンが特定のキャッシュ・データ割り当てユニットにキャッシュされる。

【 0 0 6 3 】

デステージングされたデータ・バージョンは、特により新しいデータ・バージョンがキャッシュされ、その後デステージングされた場合には、より古いデータ・バージョンとなることがある。より古いデータ・バージョンは、上書きされる前に、第 1 のストレージ・ユニット 1 2 1 から第 1 の書き戻しキャッシュ・ユニット 1 3 1 にステージングし（文字 C により示される）、次いで第 2 のストレージ・ユニット 1 2 2 にデステージングすることができる（文字 D により示される）。

【 0 0 6 4 】

特定の時点に対応するデータ・バージョンを読み取る要求が受信されると、要求されたデータ・バージョンは、第 1 のストレージ・ユニット 1 2 1 からステージングする（文字 E により示される）か、又は、第 2 のストレージ・ユニット 1 2 2 からステージングする（文字 E' により示される）ことができる。次いで、要求されたデータ・バージョンは、第 1 の書き戻しキャッシュ・ユニット 1 3 1 からアダプタ 1 4 3 に送られる。

【 0 0 6 5 】

本発明の 1 つの実施形態によれば、第 1 のストレージ・ユニット 1 2 1 は書き込みタイムスタンプを格納しない。第 2 のストレージ・ユニットは、より古いデータ・バージョンがいつ受信されたかを示すダミー・エントリを第 1 のデータ構造の中に関くことができる。このダミー・エントリは、デステージングされたデータ・バージョンが第 2 のストレージ・ユニット 1 2 2 に送られると、通常のエントリに変換される。

【 0 0 6 6 】

図 3 は、本発明の別の実施形態による、データを管理するためのシステム 1 0 0 ' を示す。図 3 は、集中型アーキテクチャを示す。

【 0 0 6 7 】

システム 1 0 0 ' は、アダプタ 1 4 3 と第 1 のストレージ・ユニット 1 2 1 とに接続された第 1 の書き戻しキャッシュ・ユニット 1 3 1 を含む。

システム 1 0 0 ' は、第 2 のストレージ・ユニット 1 2 2 を含まない点で図 1 のシステム 1 0 0 と異なる。従って、1 つのストレージ・ユニットしか存在しないので、送りデータ構造 2 6 0 及び戻しデータ構造 2 7 0 は必要とされない。

【 0 0 6 8 】

図 4 は、本発明の 1 つの実施形態によるステージング操作及びデステージング操作の例示的なシーケンスを示す。

【 0 0 6 9 】

シーケンスは、新しいデータ・バージョンを書き込む要求をアダプタ 1 4 3 から第 1 の書き戻しキャッシュ・ユニット 1 3 1 の特定のキャッシュ・データ割り当てユニットに送るステップ（文字 A により示される）で開始する。その特定のキャッシュ割り当てユニットに現在キャッシュされているキャッシュ済データ・バージョンを（データ・ストレージ・ポリシーに従って）デステージングすべき場合には、そのバージョンは、第 1 のデータ・ストレージ・ユニット 1 2 1 にデステージングされ（文字 B によって示される）、デステージングされたデータ・バージョンとなる。このデステージング・ステップの後、新しいデータ・バージョンが特定のキャッシュ・データ割り当てユニットにキャッシュされる。

【 0 0 7 0 】

デステージングされたデータ・バージョンは、特により新しいデータ・バージョンがキャッシュされ、その後デステージングされた場合には、より古いデータ・バージョンとなることがある。より古いデータ・バージョンもまた、第 1 のストレージ・ユニット 1 2 1 に格納される。

【 0 0 7 1 】

特定の時点に対応するデータ・バージョンを読み取る要求が受信されると、要求されたデータ・バージョンは、第 1 のストレージ・ユニット 1 2 1 からステージングされる（文字 C により示される）。次いで、要求されたデータ・バージョンは、第 1 の書き戻しキャッシュ・ユニット 1 3 1 からアダプタ 1 4 3 に送られる。

【 0 0 7 2 】

図 5 は、本発明の更なる実施形態による、データを管理するためのシステム 1 0 0 ' ' を示す。

【 0 0 7 3 】

システム 1 0 0 ' ' は、第 1 の書き戻しキャッシュ・ユニット 1 3 1 と、第 2 の書き戻しキャッシュ・ユニット 1 3 2 と、第 1 の不揮発性メモリ 1 4 1 と、第 2 の不揮発性メモリ 1 4 2 と、コントローラ 1 5 1 と、アダプタ 1 4 3 と、第 1 のストレージ・ユニット 1 2 1 と、第 2 のストレージ・ユニット 1 2 2 と、を含む。不揮発性メモリは、図 1、図 2、図 3、及び図 4 に示される構成のような、システムの他の構成に含めることができることに留意されたい。

【 0 0 7 4 】

システム 1 0 0 ' ' は、実質的に互いに独立した 2 つの部分を含む。1 つの部分は他の部分をバックアップする。好ましくは、各々の部分は、自身の電源供給ユニット、自身のコントローラなどを有する。説明を簡単にするために、単一のコントローラ 1 5 1 及び単一のアダプタ 1 4 3 が示されるが、これらは二重化することもできる。

【 0 0 7 5 】

システム 1 0 0 ' ' の第 1 の部分は、第 1 の不揮発性メモリ 1 4 1 と、第 1 の書き戻しキャッシュ・ユニット 1 3 1 と、第 1 のストレージ・ユニット 1 2 1 と、を含む。システ

10

20

30

40

50

ム 1 0 0 ' ' の第 2 の部分は、第 2 の不揮発性メモリ 1 4 2 と、第 2 の書き戻しキャッシュ・ユニット 1 3 2 と、第 2 のストレージ・ユニット 1 2 2 と、を含む。

【 0 0 7 6 】

コントローラ 1 5 1 は、第 1 の書き戻しキャッシュ・ユニット 1 3 1 と、第 2 の書き戻しキャッシュ・ユニット 1 3 2 と、第 1 の不揮発性メモリ 1 4 1 と、第 2 の不揮発性メモリ 1 4 2 と、に接続される。アダプタ 1 4 3 は、第 1 の書き戻しキャッシュ・ユニット 1 3 1 と、第 2 の書き戻しキャッシュ・ユニット 1 3 2 と、第 1 の不揮発性メモリ 1 4 1 と、第 2 の不揮発性メモリ 1 4 2 と、に接続される。

【 0 0 7 7 】

第 1 のストレージ・ユニット 1 2 1 は、第 1 の書き戻しキャッシュ・ユニット 1 3 1 と、第 2 の書き戻しキャッシュ・ユニット 1 3 2 と、に接続される。第 2 のストレージ・ユニット 1 2 2 は、第 1 の書き戻しキャッシュ・ユニット 1 3 1 と、第 2 の書き戻しキャッシュ・ユニット 1 3 2 と、に接続される。

10

【 0 0 7 8 】

第 1 の書き戻しキャッシュ・ユニット 1 3 1 に送られるデータは、第 2 の不揮発性メモリ 1 4 2 にも送られる。第 2 の書き戻しキャッシュ・ユニット 1 3 2 に送られるデータは、第 1 の不揮発性メモリ 1 4 1 にも送られる。

【 0 0 7 9 】

さらに、第 1 のストレージ・ユニット 1 2 1 と第 2 のストレージ・ユニット 1 2 2 の各々は、第 1 の書き戻しキャッシュ・ユニット 1 3 1 からのデータ及び第 2 の書き戻しキャッシュ・ユニット 1 3 2 からのデータを格納することができる。

20

【 0 0 8 0 】

上述されたように、システム 1 0 0 ' ' は、2 つの部分の有する。説明を簡単にするために、図 6 - 図 1 0 は、第 1 の部分（特に第 1 の書き戻しキャッシュ・ユニット 1 3 1 ）への読み取り操作及び書き込み操作を示す。当業者であれば、第 2 の部分への読み取り操作及び書き込み操作は、対称に実行されることが分かるであろう。

【 0 0 8 1 】

図 6 は、本発明の別の実施形態による読み取り及び書き込みのステージング操作及びデステージング操作の例示的なシーケンスを示す。

図 6 は、第 1 のストレージ・ユニット 1 2 1 がデステージングされたデータ・バージョンとより古いデータ・バージョンの両方を格納する集中型アーキテクチャを示す。

30

【 0 0 8 2 】

図 6 に示されるシーケンスは、新しいデータ・バージョンを第 1 の書き戻しキャッシュ・ユニット 1 3 1 にキャッシュすることに加えて、それを第 2 の不揮発性メモリ 1 4 2 に書き込む（文字 A ' によって示される）という点で、図 4 のシーケンスと異なる。従って、第 1 の部分が故障した場合には、そのデータ・バージョンを第 2 の不揮発性メモリ・ユニット 1 4 2 から取得する（文字 D ' によって示される）ことができる。

【 0 0 8 3 】

図 7 は、本発明のさらに別の実施形態による、読み取り及び書き込みのステージング操作及びデステージング操作の例示的なシーケンスを示す。

40

図 7 は、第 1 のデータ・ストレージ・ユニット 1 2 1 がデステージングされたデータ・バージョンを格納し、第 2 のストレージ・ユニット 1 2 2 がより古いデータ・バージョンを格納する、分散型アーキテクチャを示す。

【 0 0 8 4 】

図 7 に示されるシーケンスは、新しいデータ・バージョンを第 1 の書き戻しキャッシュ・ユニット 1 3 1 にキャッシュすることに加えて、それを第 2 の不揮発性メモリ 1 4 2 に書き込む（文字 A ' によって示される）という点で、図 2 のシーケンスと異なる。従って、第 1 の部分が故障した場合には、そのデータ・バージョンを第 2 の不揮発性メモリ・ユニット 1 4 2 から取得する（文字 F ' によって示される）ことができる。

【 0 0 8 5 】

50

図 8 は、本発明の更なる実施形態による、読み取り及び書き込みのステージング操作及びデステージング操作の例示的なシーケンスを示す。

図 8 は、キャッシュ・レベル分割アーキテクチャを示す。キャッシュ・レベル分割アーキテクチャ・システムは、新しいデータ・バージョンを、第 1 の書き戻しキャッシュ・ユニット 1 3 1 と第 2 の書き戻しキャッシュ・ユニット 1 3 2 の両方にキャッシュする。キャッシュ・レベル分割アーキテクチャにおいては、キャッシュ済データ・バージョンは、第 1 のストレージ・ユニット 1 2 1 から（キャッシュ・ユニットを経由して）第 2 のデータ・ストレージ・ユニット 1 2 2 には送られない。キャッシュ済データ・バージョンは、第 1 の書き戻しキャッシュ・ユニット 1 3 1 から第 1 のストレージ・ユニット 1 2 1 に送られ（文字 A によって示される）、第 2 の書き戻しキャッシュ・ユニット 1 3 2 から第 2 のストレージ・ユニット 1 2 2 に送られる（文字 A' によって示される）。より古いデータ・バージョンは、第 1 のストレージ・ユニット 1 2 1 に格納されるのではなく、第 2 のストレージ・ユニット 1 2 2 に格納される。第 2 のストレージ・ユニット 1 2 2 は、デステージングされたデータ・バージョン及びより古いデータ・バージョンを受信し、格納する。

10

【 0 0 8 6 】

好ましくは、デステージングされたデータ・バージョンは、第 1 のストレージ・ユニット 1 2 1 から提供される（文字 C によって示される）。より古いデータ・バージョンは、第 2 のストレージ・ユニットから提供される（文字 C' によって示される）。デステージングされたデータ・バージョンは、第 2 のストレージ・ユニット 1 2 2 から提供できることに留意されたい。第 2 の不揮発性メモリ 1 4 2 もまた、新しいデータ・バージョンを受信し（文字 A' ' によって示される）、これらのデータ・バージョンをアダプタ 1 4 3 に提供することができる（文字 D' ' によって示される）。

20

【 0 0 8 7 】

キャッシュ・レベル分割アーキテクチャにおいては、データは、第 1 のストレージ・ユニット 1 2 1 から第 2 のストレージ・ユニットに送られない。好ましくは、データ・ストレージ粒度が非常に細かい場合には、キャッシュ・レベル分割アーキテクチャは、多数のステージング操作及びデステージングを省くことができる。その一方で、データ・ストレージ粒度が低い場合には、このアーキテクチャは、非分割アーキテクチャにおいて必要とされる可能性のある操作より多くのデステージング操作を実施する。

30

【 0 0 8 8 】

図 9 は、本発明の更なる実施形態による、読み取り及び書き込みのステージング操作及びデステージング操作の例示的なシーケンスを示す。

図 9 は、ストレージ・ユニット・レベル分割アーキテクチャを示す。ストレージ・ユニット・レベル分割アーキテクチャ・システムは、新しいデータ・バージョンを単一の書き戻しキャッシュ・ユニットにキャッシュするが、キャッシュ済データ・バージョンは、第 1 のストレージ・ユニット 1 2 1（文字 B によって示される）及び第 2 のストレージ・ユニット 1 2 2（文字 B' によって示される）に送る。

【 0 0 8 9 】

第 1 のストレージ・ユニット 1 2 1 は、より古いデータ・バージョンを格納しない。デステージングされたデータ・バージョンは、第 1 のストレージ・ユニット 1 2 1 から第 1 の書き戻しキャッシュ・ユニットに提供される（文字 C によって示される）。より古いデータ・バージョンは、第 2 のストレージ・ユニット 1 2 2 に格納され、後で第 2 のストレージ・ユニット 1 2 2 から第 1 の書き戻しキャッシュ・ユニット 1 3 1 に提供することができる（文字 C' によって示される）。デステージングされたデータ・バージョンは、第 2 のストレージ・ユニット 1 2 2 から提供できることに留意されたい。第 2 の不揮発性メモリ 1 4 2 もまた、新しいデータ・バージョンを受信し（文字 A' によって示される）、これらのデータ・バージョンを提供する（文字 D' によって示される）ことができる。

40

【 0 0 9 0 】

ストレージ・ユニット・レベル分割アーキテクチャにおいては、データは、第 1 のスト

50

レージ・ユニット 1 2 1 から第 2 のストレージ・ユニットに送られない。好ましくは、データ・ストレージ粒度が非常に細かい場合には、ストレージ・ユニット・レベル分割アーキテクチャは、多数のステージング操作及びデステージングを省くことができる。

【 0 0 9 1 】

本発明の別の実施形態によれば、システム 1 0 0 ' ' は、選択的ストレージ・ユニット・レベル分割アーキテクチャを適用することもできる。従って、システム 1 0 0 ' ' は、キャッシュ済データ・バージョンをデステージングする前に、そのデータ・バージョンを第 1 のストレージ・ユニットに送るか、又は、第 1 のストレージ・ユニット 1 2 1 及び第 2 のストレージ・ユニット 1 2 2 の両方に送るかを決定することができる。

【 0 0 9 2 】

好ましくは、システムが、キャッシュ済データ・バージョンがデステージング期間の経過の間にキャッシュされることになるデータ・バージョンのうちの最後のものであることを判別できる場合には、このキャッシュ済データ・バージョンは、第 1 のストレージ・ユニット及び第 2 のストレージ・ユニットに送ることができる。キャッシュ済データ・バージョンは、デステージング期間の途中で、又はそのデステージング期間の最初に、デステージングすべき場合には、第 1 のストレージ・ユニット 1 2 1 にデステージングすることができる。

キャッシュ済データ・バージョンは、デステージング期間の所定部分が経過したことに応答して、両方のデータ・ストレージ・ユニットに送ることができることに留意されたい。

【 0 0 9 3 】

選択的ストレージ・ユニット分割アーキテクチャは、データ・ストレージ粒度が粗いときには実質的に分散型アーキテクチャとして作動し、データ・ストレージ粒度が細かいときには実質的にストレージ・ユニット・レベル分割アーキテクチャとして作動する。

【 0 0 9 4 】

図 1 0 は、本発明の 1 つの実施形態による、第 1 のデータ構造 2 0 0 の一部 2 0 1 と、第 2 のデータ構造 2 5 0 の一部 2 5 1 とを示す。この例においては、キーはタイムスタンプを持つ L B A であり、データは物理アドレスである。しかしながら、他の例を用いることもできる。説明を簡単にするために、これらの図は、同じ論理ブロック・アドレス（例えば L B A = 1 2 ）に対する書き込み操作及び差し戻し操作を示す。

第 1 のデータ構造 2 0 0 は、他の L B A に対する書き込み操作に関するメタデータを含むことに留意されたい。

また、一部 2 0 1 は、特定の L B A に対する更新操作を表すものとして示されるが、他の L B A に関する操作を表すこともできることに留意されたい。代替的に、他のグラフが他の L B A に関わる操作を表すこともできる。

【 0 0 9 5 】

第 1 のデータ構造 2 0 0 は、3 つの列 2 0 0 (1)、2 0 0 (2)、及び 2 0 0 (3) を含む。各々のエントリは、書き込み操作に関する情報を含む。第 1 列 2 0 0 (1) は書き込み操作の論理ブロック・アドレスを含み、第 2 列 2 0 0 (2) は書き込みタイムスタンプを含み、第 3 列 2 0 0 (3) は書き込み操作の物理アドレスを含み。論理ブロック・アドレス及び書き込み時間のフィールドは、第 1 のデータ構造 2 0 0 に対するキー（インデックス）として用いることができる。

特定の時点において、多数の L B A に対する書き込み操作が生じた場合には、第 1 のテーブルは、この書き込み操作を反映する多数のエントリを含むことになることに留意されたい。

【 0 0 9 6 】

第 2 のデータ構造 2 5 0 は、4 つの列 2 5 0 (1) - 2 5 0 (4) を含む。第 1 列 2 5 0 (1) は分岐識別子を含み、第 2 列 2 5 0 (2) は分岐開始時間を含み、第 3 列 2 5 0 (3) は分岐終了時間を含み、第 4 列 2 5 0 (4) は各々の分岐の差し戻し時間を含む。

【 0 0 9 7 】

データ構造 2 0 0 及び 2 5 0 は、所与の時点の各々において 1 つの分岐のみがアクティブである書き込み操作及び差し戻し操作のシーケンスを制御するように適合される。同時にアクティブな多数の分岐をサポートするためには、これらのテーブルは、分岐識別情報を含むように修正しなければならない。

【 0 0 9 8 】

図 1 1 は、本発明の 1 つの実施形態による書き込み操作及び差し戻し操作の例示的なシーケンス 1 0 1 を示す。シーケンス 1 0 1 は、時間 1 0、3 0、4 0、6 0、9 0、及び 1 1 0 における論理ユニット (L U N) などの仮想アドレス空間への書き込み操作と、時間 3 5 及び 7 0 における L U N の内容を差し戻すための (時間 8 0 及び 1 0 0 において受信された) 要求と、を含む。

10

破線は差し戻し操作を表す。所与の時点のいずれにおいても 1 つの分岐のみがアクティブであることに留意されたい。

【 0 0 9 9 】

書き込み操作は、論理ブロック・アドレス 1 2 と関連付けられ、これらの書き込み操作と関連付けられる物理アドレスは、これに応じて、a、b、c、d、e、及び f とされたと仮定する。

【 0 1 0 0 】

第 1 のデータ構造 2 0 0 の第 1 列 2 0 0 (1) は、書き込み操作が論理ブロック・アドレス 1 2 に対するものであったことを示す。第 1 のデータ構造 2 0 0 の第 2 列 2 0 0 (2) は、書き込み操作が、時間 1 0、3 0、4 0、6 0、9 0、及び 1 1 0 において生じたことを示す。第 1 のデータ構造 2 0 0 の第 3 列 2 0 0 (3) は、これらの書き込み操作と関連付けられた物理アドレスが、a、b、c、d、e、及び f であったことを示す。

20

【 0 1 0 1 】

第 2 のデータ構造 2 5 0 の第 1 のエントリは、第 1 の分岐が時間 0 において開始し、時間 8 0 において終了したことを示す。第 1 の分岐は、L U N のコンテンツを差し戻す第 1 の要求が受け入れられたときに終了した。

【 0 1 0 2 】

第 2 のデータ構造 2 5 0 の第 2 のエントリは、第 2 の分岐が時間 8 0 において開始し、時間 1 0 0 において終了したことを示す。第 2 の分岐は、L U N のコンテンツを差し戻す第 2 の要求が受け入れられたときに終了した。

30

【 0 1 0 3 】

第 2 のデータ構造 2 5 0 の第 3 のエントリは、第 3 の分岐が時間 1 0 0 において開始し、終了しなかったことを示す。

【 0 1 0 4 】

本発明の別の実施形態によれば、第 2 のデータ構造は、分岐の統計値などの付加的なメタデータを含む。分岐の統計値は、例えば、その分岐に属する第 1 のデータ構造のエントリの数、その分岐の持続時間の間に書き込まれた異なる論理ブロック・アドレスの数などを含むことができる。好ましくは、分岐の統計値は、特に読み書きの操作が頻繁に生じる場合に、どの分岐を削除するかを判定を補助することができる。

【 0 1 0 5 】

40

図 1 2 は、本発明の 1 つの実施形態による書き込み操作及び差し戻し操作の例示的なシーケンス 3 0 0 を示す。

【 0 1 0 6 】

第 1 の L B A への書き込み操作 (W 1、W 3、W 5、及び W 6) は三角形で示される。第 2 の L B A への書き込み操作 (W 4 及び W 1 0) は円形で示される。第 3 の L B A への書き込み操作 (W 2、W 7、W 8 及び W 9) は四角形で示される。

【 0 1 0 7 】

シーケンス 3 0 0 は、4 つの分岐 B 1 3 0 1 - B 4 3 0 4 を含み、T 1 4 から現時点 (T _ C U R R E N T) までの間にわたる関連ウィンドウ 3 1 0 を定める。

【 0 1 0 8 】

50

第1の分岐(B1)301は、時間T0において開始し(S1)、時間T8において終了する(E1)。第1の分岐B1は、以下の書き込み操作、すなわち、T1における(第1のLBAへの)書き込み操作W1と、T2における(第3のLBAへの)書き込み操作W2と、T3における(第1のLBAへの)書き込み操作W3と、T4における(第2のLBAへの)書き込み操作W4と、T6における(第1のLBAへの)書き込み操作W5と、を含む。B1は、T8において終了する(E1)。

【0109】

第2の分岐(B2)302はB1 301の子であり、時間T5への差し戻し操作(RV1)によって時間T8において開始する(S2)。第2の分岐B2 302は、以下の書き込み操作、すなわち、T9における(第1のLBAへの)書き込み操作W6と、T11における(第3のLBAへの)書き込み操作W7と、を含む。B2は、T12において終了する(E2)。

10

【0110】

第3の分岐(B3)303はB2 302の子であり、時間T10への差し戻し操作(RV2)によって時間T12において開始する(S3)。第3の分岐B3 303は、T13における(第3のLBAへの)単一の書き込み操作W8を含む。B3は、T15において終了する(E3)。

【0111】

第4の分岐(B4)304はB3 303の子であり、時間T14への差し戻し操作(RV3)によって時間T15において開始する(S4)。B4 304は、T16における(第3のLBAへの)書き込み操作W9と、T17における(第2のLBAへの)別の書き込み操作W10と、を含む。第4の分岐B4 304は、T18において終了する(E4)。

20

【0112】

図13は、本発明の別の実施形態による書き込み操作及び差し戻し操作の例示的なシーケンス300'を示す。

シーケンス300'は、第2の差し戻し操作(RV2')の差し戻し時間の点でシーケンス300と異なる。差し戻し時間は、(第1の分岐B1 301に属する)T7であり、(第2の分岐B2 302に属する)T10ではない。従って、シーケンス300では子分岐(B3 303)を有していた第2の分岐B2 302は、シーケンス300'において子を持たない。

30

【0113】

図14は、本発明の1つの実施形態による、データ・バージョンを取得するための方法400を示す。

方法400は、現在の分岐を選択する又は獲得するステージ410で開始する。ステージ410の後に続くステージ412では、その分岐のセクションにおいて、分岐の開始時間において開始し、データ・バージョンの取得要求と関連する要求されたタイムスタンプにおいて終了する、書き込み操作が存在するかどうかをチェックする。答えが否定の場合にはステージ412の後にステージ414が続き、それ以外の場合にはステージ412の後にステージ420が続く。

40

【0114】

ステージ420は、方法400の結果として最後の書き込み操作を戻すことを含む。ステージ414は、その分岐が親分岐を有するかどうかをチェックすることを含む。答えが否定の場合には、方法400は、否定の答え(Null)を戻すステージ416で終了する。答えが肯定の場合には、ステージ414の後に、分岐の親を獲得するステージ422と、その親分岐のセクションにおいて、その分岐の開始時間において開始し、検査される親分岐の子分岐と関連する差し戻し時間において終了する、書き込み操作が存在するかどうかをチェックするステージ424と、が続く。答えが否定の場合にはステージ424の後にステージ414が続き、それ以外の場合にはステージ424の後にステージ420が続く。

50

【 0 1 1 5 】

図 1 5 は、本発明の 1 つの実施形態による粗分析を示す。

粗分析 5 0 0 は、分岐インデックス (J) を 1 にセットするステージ 5 1 0 で開始する。

ステージ 5 1 0 の後に、第 J 番目の分岐と関連ウィンドウとの間の関係を判定するステージ 5 2 0 が続く。

【 0 1 1 6 】

第 J 番目の分岐と関連ウィンドウとが少なくとも部分的に重なる場合には、第 J 番目の分岐は削除されず、ステージ 5 2 0 の後に、J を増加させるステージ 5 3 0 が続き、ステージ 5 2 0 にジャンプする。この反復は、第 2 のデータ構造全体が走査されたときに終了することに留意されたい。

10

【 0 1 1 7 】

第 J 番目の分岐と関連ウィンドウとが重ならない (部分的にも重ならない) 場合には、ステージ 5 2 0 の後に、第 J 番目の分岐が 1 つ又は複数の子分岐を含むかどうかをチェックするステージ 5 4 0 が続く。

【 0 1 1 8 】

第 J 番目の分岐が単一の子分岐を有する場合には、ステージ 5 4 0 の後に、第 J 番目の分岐とその子分岐とを統合するステージ 5 5 0 が続く。本発明の 1 つの実施形態によれば、ステージ 5 5 0 は、分岐を、統合されるべき分岐として (例えば、子分岐の差し戻し時間の前の最後のエントリを子分岐の開始時間に移動させることによって) マークすることを含む。

20

【 0 1 1 9 】

第 J 番目の分岐が関連ウィンドウの外部にあり、いかなる子も持たない場合には、ステージ 5 4 0 の後に、その分岐と関連するメタデータを削除するステージ 5 6 0 が続く。本発明の 1 つの実施形態によれば、ステージ 5 6 0 は、分岐を、削除されるべき分岐としてマークすることを含む。削除は、精密分析の際に行うことができる。

【 0 1 2 0 】

多数の子分岐が存在する場合には、ステージ 5 4 0 の後にステージ 5 3 0 が続く。

ステージ 5 6 0 及び 5 5 0 の後にステージ 5 3 0 が続く。

【 0 1 2 1 】

例えば、図 1 2 を参照すると、第 4 の分岐 B 4 3 0 4 は関連ウィンドウ内にあり、第 1 から第 3 までの分岐は第 4 の分岐の祖先である。従って、どの分岐も削除されない。

30

【 0 1 2 2 】

さらに別の例として、図 1 3 を参照すると、第 2 の分岐 B 2 3 0 2 は、第 4 の分岐 B 4 3 0 4 の祖先ではなく、関連ウィンドウの外部にあり、従って削除することができる。

【 0 1 2 3 】

削除は、分岐を削除候補としてマークし、そのようなマークに対応する精密分析を行い、その後で分岐を削除することを含むことができることに留意されたい。

【 0 1 2 4 】

図 1 6 及び図 1 7 は、本発明の 1 つの実施形態による全体精密分析 6 0 0 を示す。

40

図 1 6 は、分析 6 0 0 の最初の部分 6 0 0 (a) を示し、図 1 7 は分析 6 0 0 の部分 6 0 0 (b) を示す。

【 0 1 2 5 】

全体精密分析は、特定の分岐について実施される。次いで、別の分岐について実施することができる。

全体精密分析 6 0 0 は、その特定の分岐の一部において少なくともアクティブである分岐の各々を探し出すステージ 6 1 0 で開始する。このような分岐の各々は関連分岐と呼ばれる。

【 0 1 2 6 】

50

ステージ 6 1 0 の後に、関連ウィンドウの外部に位置する関連分岐を子分岐の差し戻し時間によって分割されるセクションに分けるステージ 6 2 0 が続く。各々のセクションは子分岐と関係付けられる。

【 0 1 2 7 】

ステージ 6 2 0 の後に、分岐の最後のセクション（子分岐の最後の差し戻し時間において開始し、分岐の終了時間において終了するセクション）が関連ウィンドウの外部にあると仮定して、そのセクションにおけるいずれかの書き込み操作を除去するステージ 6 2 5 が続く。

【 0 1 2 8 】

ステージ 6 2 5 の後に、（関連ウィンドウの外部にある）セクションが多数の書き込みを有する場合には、そのセクションの全ての書き込み操作（最後の書き込みを除く）を除去するステージ 6 3 0 が続く。

【 0 1 2 9 】

ステージ 6 3 0 の後に、多数のセクションのうちの特定のセクションについて、そのセクション及び後続のセクションが書き込み操作を含むかどうかを判定するステージ 6 3 5 が続く。答えが肯定の場合には、ステージ 6 3 5 の後に、書き込みセクションを子分岐の開始時間に移動させるステージ 6 4 2 が続く。答えが否定の場合には、ステージ 6 3 5 の後に、現在のセクションが書き込み操作を有していないか、又は局所精密分析が適用されるかどうかを判定するステージ 6 4 0 が続く。答えがはいであれば、ステージ 6 4 0 の後にステージ 6 5 0 が続き、そうでなければ、多数のセクションの中から次のセクションを選択するステージ 6 4 5 が続き、ステージ 6 3 5 にジャンプする。

【 0 1 3 0 】

ステージ 6 5 0 は、特定の書き込み操作 U と、特定のセクションの子分岐 $C_1, \dots, C_n$ と、を受信することを含む。ステージ 6 5 0 の後に、カウンタ x を 0 にセットし、分岐 b を $null$ にセットし、作業リスト WL を書き込み操作 U と子分岐との対、即ち $WL = \{(U, C_1), \dots, (U, C_n)\}$ に等しくなるように定める（即ちセットする）ステージ 6 5 2 が続く。

【 0 1 3 1 】

ステージ 6 5 2 の後に、 WL が空かどうかを判定するステージ 6 5 4 が続く。答えが肯定の場合にはステージ 6 5 4 の後にステージ 6 5 6 が続き、答えが否定の場合にはステージ 6 5 4 の後にステージ 6 6 4 が続く。

【 0 1 3 2 】

ステージ 6 5 6 は、 $\langle x, b \rangle$ を準備することを含む。ステージ 6 5 6 の後に、 x の値をチェックするステージ 6 5 8 が続く。 $x = 0$ であれば、ステージ 6 5 8 の後にステージ 6 6 0、即ち書き込み操作を除去することが続く。ステージ 6 6 0 の後に、ステージ 6 4 5 が続く。 $x = 1$ であれば、ステージ 6 5 8 の後に、書き込み操作を子分岐に移動させるステージ 6 6 2 が続く。 $x = 2$ であれば、ステージ 6 5 8 の後にステージ 6 4 5 が続く。ステージ 6 6 2 の後にステージ 6 4 5 が続く。

【 0 1 3 3 】

ステージ 6 6 4 は、 WL から第 1 のアイテム（ (w, d) と呼ばれる）を除去することと、分岐 d の最初の書き込み操作が関連ウィンドウの開始後に生じたかどうかをチェックすることと、を含む。答えが否定の場合には、ステージ 6 6 4 の後にステージ 6 7 4 が続き、それ以外の場合にはステージ 6 6 6 が続く。ステージ 6 6 6 は、 x を増加させることを含み、その後に、 x が 1 より大きいかどうかをチェックするステージ 6 6 8 が続く。 x がゼロ又は 1 であれば、ステージ 6 6 8 の後に、 $\langle x, b \rangle$ を $\langle x, null \rangle$ にセットするステージ 6 7 0 が続き、ステージ 6 5 6 にジャンプする。 $x > 1$ であれば、ステージ 6 6 8 の後に、 b を 2 にセットするステージ 6 7 2 が続き、ステージ 6 5 4 にジャンプする。

【 0 1 3 4 】

ステージ 6 7 4 は、作業リストに、分岐 d の開始時間と、 w の時間より小さい差し戻し

10

20

30

40

50

時間を有する分岐 d の子分岐との対のセットを加えることを含む。 $WL = \{ (開始時間(d), d1), \dots, (開始時間(d), dn) \}$ である。ステージ 674 の後にステージ 654 が続く。

【0135】

上述された例の殆どは、いずれの時点においてもアクティブな分岐が一つだけ存在するシナリオについて言及していることに留意されたい。上述されたように、これは必須の条件ではない。共存する多数の分岐をサポートするために、共存する多数の分岐の中から選択するための付加的なメタデータが必要となる。この付加的なメタデータは、分岐識別子、親分岐識別子などを含むものとすることができる。

【0136】

テスト環境、多数のスナップショットをサポートするホスト・アドレス可能な論理ユニットの実装などを含むがこれらに限定されない種々のシナリオにおいて、多数の分岐は同時に存在することができる。

【0137】

多数の分岐が同時に存在する場合には、(例えば、差し戻し操作を実施することによる) 新たな分岐の追加は、必ずしも別の分岐を終了させないことに留意されたい。本発明の1つの実施形態によれば、第3のデータ構造が定められる。

【0138】

第3のデータ構造は、各々のLBAに対する最初の書き込み操作を含む。このようなテーブルは、種々の走査シーケンスを簡略化することができる。さらに別の例について、分岐ごとの最後の書き込み操作を含む別のデータ構造を定めることもできる。本発明の別の実施形態によれば、B木はプレフィクスB木によって置き換えられる。

【0139】

1つの代替的な解決策は、B木のエントリを時間成分のない論理トラック数のみにすること、並びに、全てのLBAエントリ及びその時間情報を伴う可変長の構造を木の葉に格納することである。

【0140】

図18は、本発明の1つの実施形態による、データを管理するための方法700を示す。

方法700は、例えば論理ブロック・アドレスであるがこれに限定されないキーによって特定される情報への書き込み操作を表す第1のデータ構造を準備し、例えば差し戻し操作、分岐クローニング操作などであるがこれらに限定されない分岐作成操作を表す第2のデータ構造を準備する、ステージ710で開始する。好ましくは、第1のデータ構造は、書き込みタイムスタンプと、論理ブロック・アドレスとその関連する物理アドレスとの間のマッピングとを含む。好ましくは、第2のデータ構造は、分岐識別子、分岐開始時間、分岐終了時間、及び分岐差し戻し時間を含む。第1のデータ構造はB木であることが好ましい。

ステージ710の後には、ステージ720、730、740、及び750が続く。

【0141】

ステージ720は、データを書き込むか又は更新する要求を受信することと、それに応じて第1のデータ構造を更新することと、を含む。

【0142】

ステージ730は、要求されたタイムスタンプにおけるバージョンのデータによって開始する分岐を作成する要求を受信することと、それに応じて第2のデータ構造を更新することと、を含む。例えば、分岐を作成する要求は、データを特定の差し戻し時間に差し戻す要求の結果である場合がある(差し戻し時間は、要求された時間と見なされる)。さらに別の例について、分岐を作成する要求は、データを論理的に複製する(クローニングする)要求の結果である場合があり、この場合には、要求時間が要求された時間となる。

【0143】

ステージ740は、読み取りタイムスタンプにおけるデータのバージョンを読み取る要

10

20

30

40

50

求を受信することと、そのバージョンのデータを探し出すために第1のデータ構造及び第2のデータ構造を走査することと、を含む。ステージ740は、方法400の1つ又は複数のステージを含むことができる。

【0144】

ステージ750は、データ・ストレージ・ポリシーに応じて、第1のデータ構造及び第2のデータ構造を更新することを含む。ステージ750は、粗分析を実施すること、精密分析を実施することなどを含むことができる。ステージ750は、図15、図16、及び図17に示される1つ又は複数のステージを含むことができる。

【0145】

さらに、本発明は、コンピュータ又はいずれかの命令実行システムによって又はこれらと関連して用いるためのプログラム・コードを提供するコンピュータ使用可能媒体又はコンピュータ可読媒体からアクセス可能なコンピュータ・プログラムの形態を取ることができる。本説明の目的上、コンピュータ使用可能媒体又はコンピュータ可読媒体は、命令実行システム、装置、又はデバイスによって又はこれらと関連して用いるためのプログラムを収容し、格納し、通信し、伝播し、又は移動させることが可能なあらゆる装置とすることができる。

【0146】

媒体は、電子式、磁気式、光学式、電磁気式、赤外線式、又は半導体式のシステム（又は、装置若しくはデバイス）であっても、伝播媒体であってもよい。コンピュータ可読媒体の例として、半導体メモリ即ち固体メモリ、磁気テープ、取り外し可能なコンピュータ・ディスク、ランダム・アクセス・メモリ（RAM）、読み取り専用メモリ（ROM）、剛体磁気ディスク、及び光ディスクが挙げられる。光ディスクの現在の例として、コンパクト・ディスク・読み取り専用メモリ（CD-ROM）、コンパクト・ディスク・読み取り/書き込み（CD-R/W）、及びDVDが挙げられる。

【0147】

プログラム・コードを格納及び/又は実行するのに適したデータ処理システムは、システム・バスを経由してメモリ要素に直接的又は間接的に結合された少なくとも1つのプロセッサを含む。メモリ要素は、プログラム・コードの実際の実行中に用いられるローカル・メモリと、バルク・ストレージと、実行中にバルク・ストレージからコードを取得しなければならない回数を減らすために少なくとも幾つかのプログラム・コードを一時的に格納するキャッシュ・メモリと、を含むことができる。

【0148】

入出力デバイス即ちI/Oデバイス（キーボード、ディスプレイ、ポインティング・デバイスなど含むがこれらに限定されない）は、システムに、直接的に、又は中間のI/Oコントローラを介して、結合することができる。

【0149】

データ処理システムが、介在するプライベート・ネットワーク又は公開ネットワークを通じて、他のデータ処理システム又は遠隔プリンタ若しくはストレージ装置に結合されることを可能にするために、ネットワーク・アダプタをシステムに結合することもできる。モデム、ケーブルモデム、及びイーサネット・カードは、現在利用可能なネットワーク・アダプタのタイプのうちのごく一部である。

【0150】

本発明の1つの実施形態によれば、データは書き戻しキャッシュに書き込まれ、データの現在のバージョンも以前のバージョンも、ディスク、ディスク・アレイ、テープなどといった1つ又は複数のストレージ・ユニットに送られる。データ・ストレージ・ポリシーは、データ及びメタデータのリフレッシュに役立つと共に、特定のデータ・バージョンをディスクに送るかどうかの判定を支援する。

【0151】

本発明は、ネットワーク上で、顧客に対するサービスとして提供することができる。特に、サービスはデータの格納及び取得を提供することができる。

10

20

30

40

50

好ましくは、ネットワーク上で顧客にサービスを提供する方法が提供される。本方法は、ネットワーク上で、読み取りタイムスタンプにおけるデータのバージョンを検索する要求を受信するステップと、データのそのバージョンを探し出すために第1のデータ構造及び第2のデータ構造を走査するステップとを含む。第1のデータ構造はキーによって特定されるエントリへの挿入操作又は更新操作を表し、第2のデータ構造は分岐作成操作を表し、第2のデータ構造は分岐識別子及び分岐タイミング情報を格納する。本方法は、読み取りタイムスタンプにおけるデータのバージョンを提供するステップと、ネットワーク上で、要求されたタイムスタンプにおけるデータのバージョンによって開始する分岐を作成する要求を受信するステップと、それに応じて第2のデータ構造を更新するステップとをさらに含む。

10

【0152】

好ましくは、第1のデータ構造はB木である。

好ましくは、本方法は、所与のキーと関連する値を更新する要求をネットワーク上で受信するステップと、第1のデータ構造を更新するステップとを含む。

好ましくは、本方法は、所与のキー及び関連するデータ並びに現在のタイムスタンプを持つ新しいエントリを追加することによって更新するステップを含む。

好ましくは、キーは論理ブロック・アドレスである。

好ましくは、分岐を作成する要求は差し戻し操作であり、特定のタイムスタンプは差し戻し時間である。

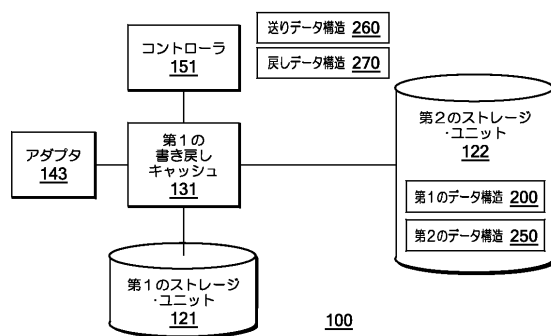
【0153】

20

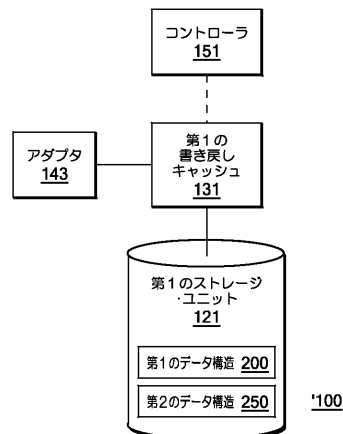
当業者であれば、請求される本発明の趣旨及び範囲から逸脱することなく、ここで説明されたものの変形、修正、及び他の実装を思いつくであろう。

従って、本発明は、前述の例示的な説明ではなく特許請求の趣旨及び範囲によって定められることとなる。

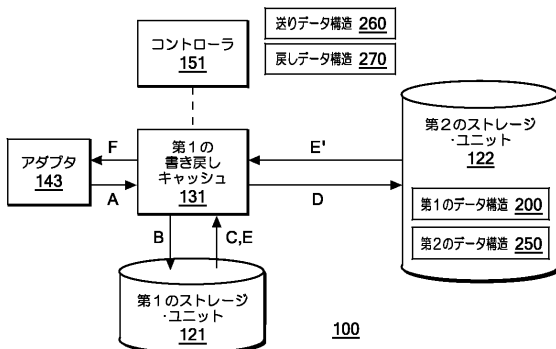
【図1】



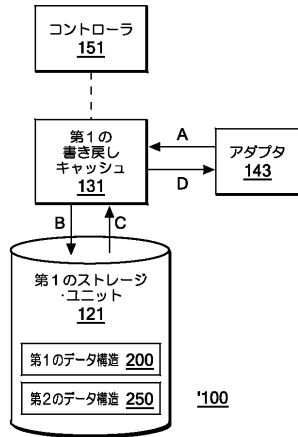
【図3】



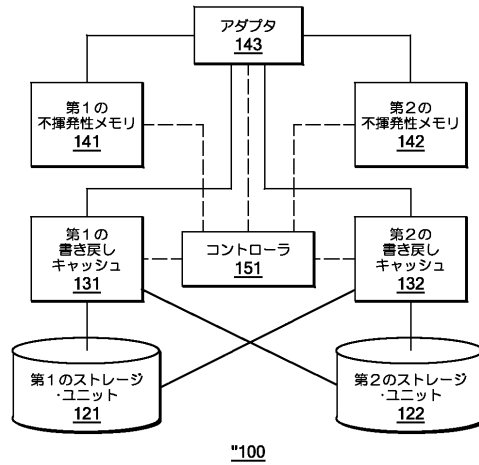
【図2】



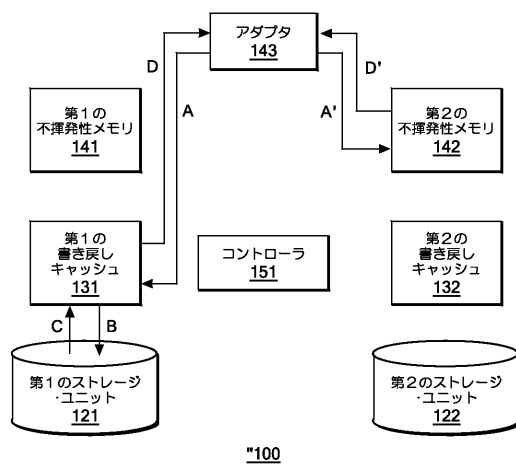
【図 4】



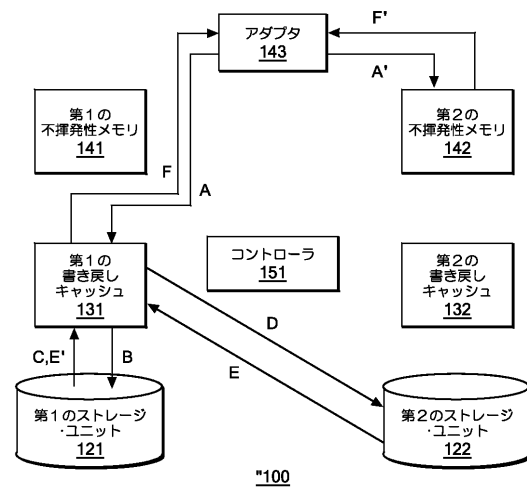
【図 5】



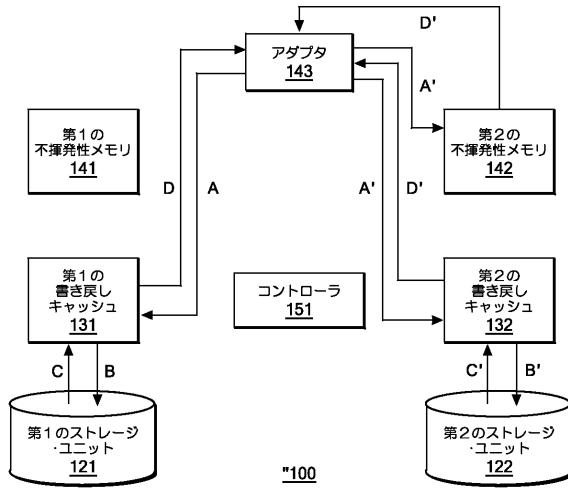
【図 6】



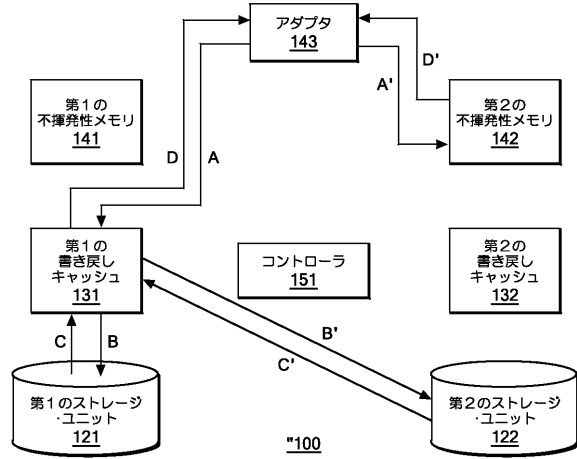
【図 7】



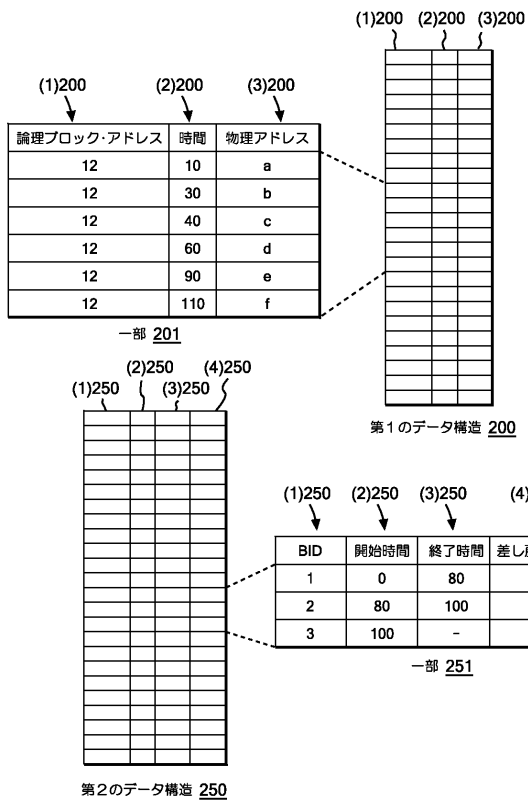
【図 8】



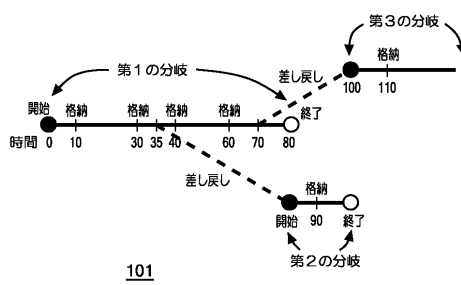
【図 9】



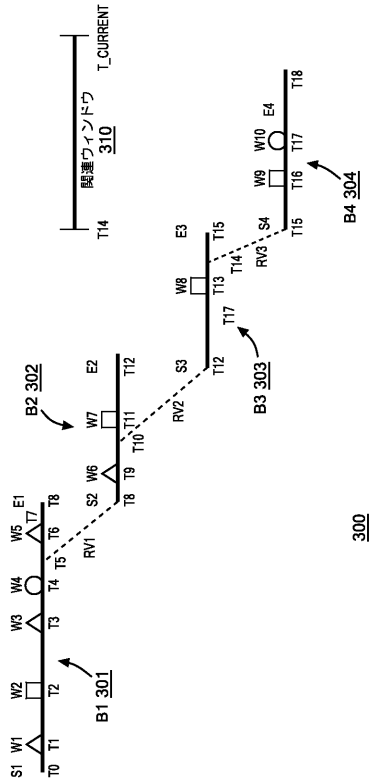
【図 10】



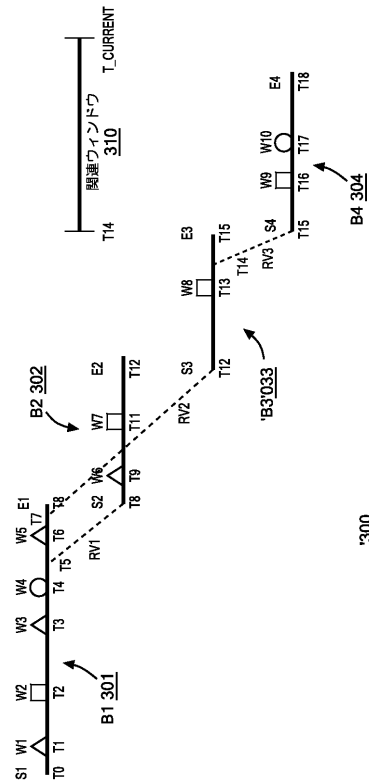
【図 11】



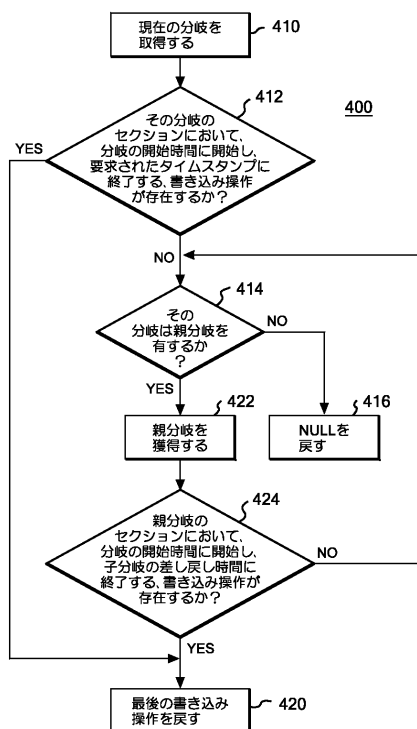
【図 1 2】



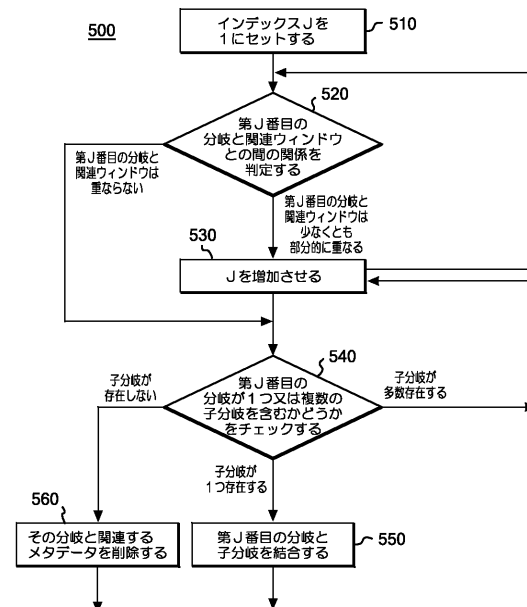
【図 1 3】



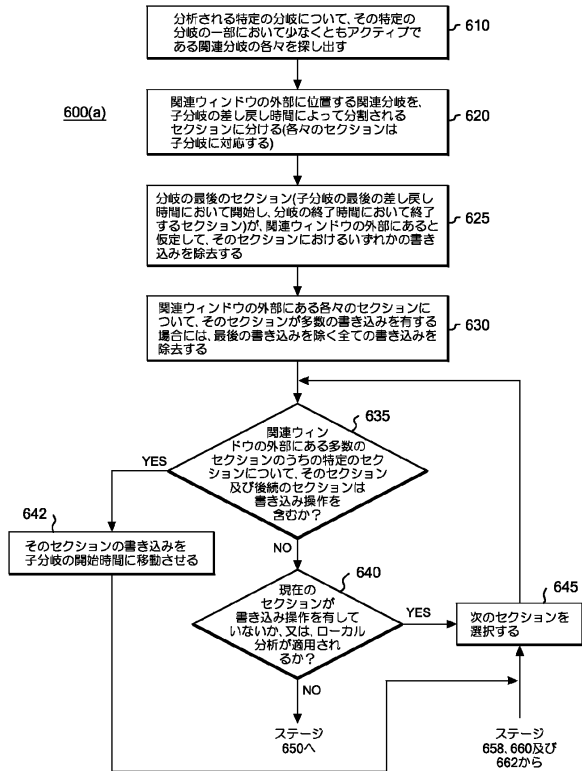
【図 1 4】



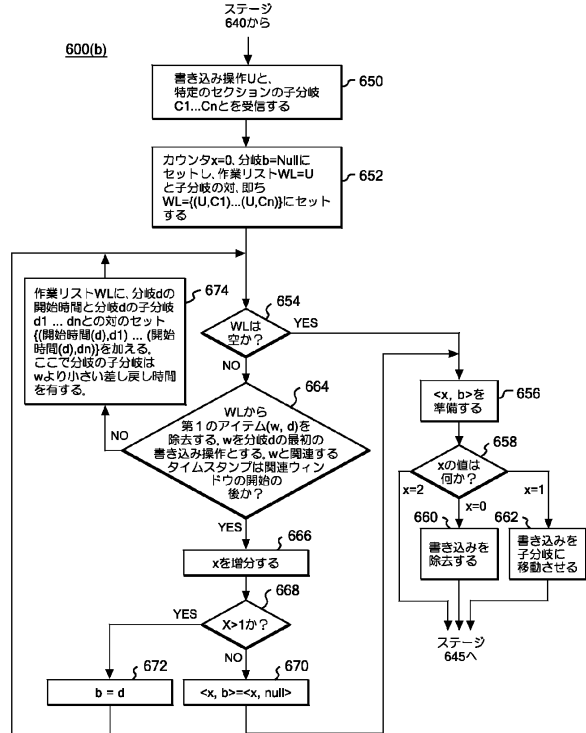
【図 1 5】



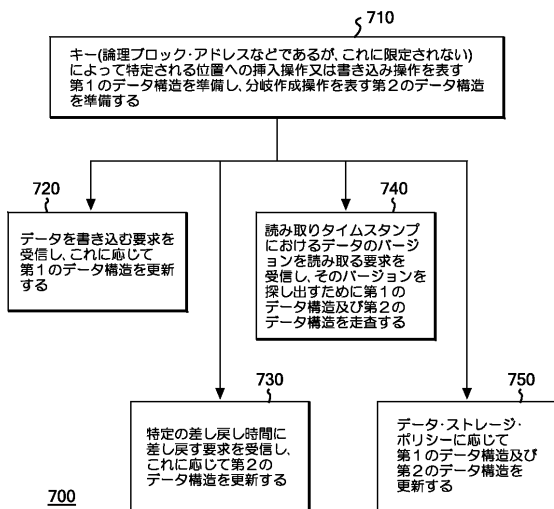
【図 16】



【図 17】



【図 18】



フロントページの続き

- (72)発明者 ファクター、マイケル
イスラエル国 3 4 4 0 2 ハイファ レイチェル・ストリート 3 2
- (72)発明者 フィーンブリット、シャチャール
イスラエル国 3 0 8 2 5 アイン・アヤラ ハリモン 2
- (72)発明者 ラーデン、ガイ
イスラエル国 6 8 1 3 0 ヤッファ エフェット 3 9
- (72)発明者 タシマ、ポーラ、キム
イスラエル国 6 9 1 2 7 テル・アビブ・ヤッファ ボイル・ストリート 3

審査官 田川 泰宏

(56)参考文献 特開2 0 0 4 - 1 8 5 6 1 7 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G06F 12/00

G06F 17/30