(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5426380号
(P5426380)
(45)発行日　平成26年2月26日(2014.2.26)　(24)登録日　平成25年12月6日(2013.12.6)

(51)Int.Cl.　F I
G06F 12/00 (2006.01)　G06F 12/00 531D
G06F 12/00 517

請求項の数 14　(全 37 頁)

(21)出願番号	特願2009-525761 (P2009-525761)
(86)(22)出願日	平成19年8月22日 (2007.8.22)
(65)公表番号	特表2010-501942 (P2010-501942A)
(43)公表日	平成22年1月21日 (2010.1.21)
(86)国際出願番号	PCT/US2007/076535
(87)国際公開番号	WO2008/024850
(87)国際公開日	平成20年2月28日 (2008.2.28)
審査請求日	平成22年8月23日 (2010.8.23)
(31)優先権主張番号	11/508,129
(32)優先日	平成18年8月22日 (2006.8.22)
(33)優先権主張国	米国(US)

(73)特許権者　506329306
アマゾン　テクノロジーズ　インコーポレイテッド
アメリカ合衆国　８９５０７　ネバダ州　レノ　ピーオー　ボックス　８１０２
(74)代理人　100061815
弁理士　矢野　敏雄
(74)代理人　100094798
弁理士　山崎　利臣
(74)代理人　100099483
弁理士　久野　琢也
(74)代理人　100112793
弁理士　高橋　佳大
(74)代理人　100128679
弁理士　星　公弘

最終頁に続く

(54)【発明の名称】高可用性データを提供するためのシステム及び方法

(57)【特許請求の範囲】
【請求項１】
プロセッサと、
メモリと、
を備えているコンピュータ実施データ記憶システムにおいて、
前記メモリは、データ記憶装置を提供するように前記システムを構成する一つ又は複数のコンピュータ実行可能ロジックコンポーネントを備え、
前記コンピュータ実行可能ロジックコンポーネントは、
リング接続構成に対応する複数のデータセンタに及び前記複数のデータセンタ内の複数のホストに複数のデータセットを記憶するための分担をマッピングするように構成されたマッピングロジックと、
前記複数のデータセンタの第１のサブセット内の前記複数のホストの第１のサブセットに、データセットの第１の複数のコピーを書き込むように構成されたデータセット複製ロジックであって、前記複数のホストの前記第１のサブセットは前記データセットのための優先リストに対応し、前記複数のデータセンタの前記第１のサブセットは前記リング接続構成上における所定の位置により識別され、複数部分キーの関数として識別される前記所定の位置は前記データセットに対応する、データセット複製ロジックと、
前記複数のデータセンタの第２のサブセット内の前記複数のホストの第２のサブセットで、前記データセットの第２の複数のコピーを読み出すことにより前記データセットの一つのコピーの供給の要求に応答するように構成されたデータセット検索ロジックであって

、前記複数のホストの前記第２のサブセットは前記データセットのための優先リストに対応する、データセット検索ロジックと、
前記データセットの一つのコピーを供給するために、前記データセットの前記第２の複数のコピー間の因果関係を評価するようにデータセット比較ロジックと、
を備え、
因果関係の前記評価は、
前記データセットの前記第２の複数のコピーのうちのコピーであって、前記データセットの前記第２の複数のコピーのうちの他のコピーの先祖でないコピーに規定の順序を適用すること、
前記データセットの前記第２の複数のコピーのうちのコピーであって、前記データセットの前記第２の複数のコピーのうちの他のコピーの先祖でないコピーを併合すること、及び、
前記データセットの前記第２の複数のコピーのうちのコピーであって、前記データセットの前記第２の複数のコピーのうちの他のコピーの先祖でないコピーの間の矛盾に特定の調整アルゴリズムを適用すること、
のうちの少なくとも一つを含み、
前記データセットの書込のための前記複数のホストの前記第１のサブセットと前記データセットの読出のための前記複数のホストの前記第２のサブセットとは、別個に決定されることを特徴とするコンピュータ実施データ記憶システム。
【請求項２】
前記マッピングロジックは、ハッシュ関数に基づいてハッシュ値を生成するように構成されたロジックを含み、個々の前記データセンタが、前記ハッシュ範囲の複数の異なる部分にある前記複数のデータセットのサブセットを記憶する分担を有するように、前記複数のデータセンタのそれぞれは、前記ハッシュ関数のハッシュ範囲内にある複数の位置を有することを特徴とする請求項１に記載のシステム。
【請求項３】
前記ハッシュ値は第１のハッシュ値であり、前記ハッシュ関数は第１のハッシュ関数であり、前記ハッシュ範囲は第１のハッシュ範囲であり、前記マッピングロジックは、第２のハッシュ範囲における第２のハッシュ関数に基づいて第２のハッシュ値を生成するように構成されたロジックを含むことを特徴とする請求項２に記載のシステム。
【請求項４】
個々の前記データセンタが、前記第２のハッシュ範囲の複数の異なる部分にある前記複数のデータセットのサブセットを記憶する分担を有するように、前記データセンタのそれぞれは、前記第２のハッシュ範囲内にある複数の位置を有することを特徴とする請求項３に記載のシステム。
【請求項５】
データ貸出の終了後に前記データセットの他のコピーを更新するように構成されている貸出ロジックをさらに備えていることを特徴とする請求項１に記載のシステム。
【請求項６】
前記データセット検索ロジックは、前記データセットの前記第２の複数のコピーをプリフェッチするように構成されていることを特徴とする請求項１に記載のシステム。
【請求項７】
前記データセット比較ロジックは、前記データセットの前記第２の複数のコピーのそれぞれに関連して記憶されたバージョン履歴に基づいて前記因果関係を評価するように構成されており、前記バージョン履歴は、それぞれのハッシュ履歴及びそれぞれのベクトルクロックのうちの少なくとも一方を含むことを特徴とする請求項１に記載のシステム。
【請求項８】
リング接続構成に対応する複数のデータセンタに及び前記複数のデータセンタ内の複数のホストに複数のデータセットを記憶するための分担を、各データセットに対応する複数部分キーのための第１及び第２のハッシュ関数を用いてマッピングするステップであって

、前記第１のハッシュ関数は、前記複数のデータセンタのうちの選択されたデータセンタに前記複数のデータセットを記憶するための分担をマッピングし且つ前記複数部分キーの第１の部分を入力として使用し、前記第２のハッシュ関数は、前記選択されたデータセンタ内の前記複数のホストのうちの選択されたホストに前記複数のデータセットを記憶するための分担をマッピングし且つ前記複数部分キーの第２の部分を入力として使用する、ステップと、

データセットの複数のコピーのそれぞれについてのバージョン履歴の書込を含めて、前記複数のデータセンタのサブセットに、前記複数のデータセンタの前記サブセットの所定の位置に基づいて前記データセットの前記複数のコピーを記憶するステップと、

前記データセットの前記複数のコピーについてのバージョン履歴に基づいて、前記データセットのコピー間の因果関係を評価するステップと、

を含み、

因果関係の前記評価は、

前記データセットの前記複数のコピーのうちのコピーであって、前記データセットの前記複数のコピーのうちの他のコピーの先祖でないコピーに規定の順序を適用すること、

前記データセットの前記複数のコピーのうちのコピーであって、前記データセットの前記複数のコピーのうちの他のコピーの先祖でないコピーを併合すること、及び、

前記データセットの前記複数のコピーのうちのコピーであって、前記データセットの前記複数のコピーのうちの他のコピーの先祖でないコピーの間の矛盾に特定の調整アルゴリズムを適用すること、

のうちの少なくとも一つを含む、

ことを特徴とする、

各ステップをプロセッサに実行させるためのコンピュータ実施データ処理方法。

【請求項９】

前記バージョン履歴の書込は、前記複数のデータセンタの前記サブセットの優先リストに従って実行され、前記優先リストは、ハッシュ関数に基づくものであることを特徴とする請求項８に記載の方法。

【請求項１０】

ハッシュキー及び前記ハッシュ関数に基づいてハッシュ値を生成するステップをさらに含み、前記ハッシュキーは、前記データセットに関連するものであり、前記ハッシュ関数に入力として適用され、前記ハッシュ関数は、前記複数のデータセットを記憶するための分担をマッピングし、前記複数のデータセンタの前記サブセットは、前記ハッシュ値に基づいて、及び、他のデータセンタが利用不可能であるか否かに基づいて、前記データセットを記憶するために選択されることを特徴とする請求項９に記載の方法。

【請求項１１】

個々の前記データセンタが、前記リング接続構成内にある複数の位置に対応する前記ハッシュ範囲の複数の異なる部分にある前記複数のデータセットのサブセットを記憶する分担を有するように、前記データセンタのそれぞれは、前記ハッシュ範囲内にある前記ハッシュ値に基づく所定の位置を有することを特徴とする請求項１０に記載の方法。

【請求項１２】

前記データセットの前記複数のコピーの前記記憶は、優先リストに従って実行され、前記優先リストは、前記データセットの前記複数のコピーが記憶されるべきデータセンタの順位を規定し、予め割り当てられた値が、書込が実行されることとなる、前記優先リスト内にあるデータセンタの数を決定し、

前記方法は、前記データセットの前記複数のコピーのうちの一つを第１のデータセンタから第２のデータセンタへ、前記第２のデータセンタが利用可能になった後に移動するステップをさらに含み、前記第２のデータセンタは、前記優先リストにおいて前記第１のデータセンタよりも上位にあることを特徴とする請求項８に記載の方法。

【請求項１３】

前記複数のデータセンタは、ネットワークを通じてユーザがアクセス可能なネットワー

クサービスシステムを実現し、前記ネットワークサービスシステムは、ユーザがアクセス可能なウェブサイトを提供することを特徴とする請求項８に記載の方法。
【請求項１４】
前記ウェブサイトは、商業ウェブサイトであり、前記データセットは、ユーザのうちの一人のためのショッピングカートについてのショッピングカートデータを含むことを特徴とする請求項１３に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
関連出願の相互参照
この出願は、２００６年３月３１日に出願され、参照することによってここに組み込まれる「高可用性データを提供するためのシステム及び方法」という名称の米国特許出願第１１／３９４，６４８号の一部継続出願である。
【背景技術】
【０００２】
事業コンピュータ環境は、特定のビジネスアプリケーションに関連するデータにアクセスすることをしばしば必要とする。単一障害点（single point of failure）を回避するために、データは、しばしば、異なるロケーション（例えば、所定のデータセンタ、異なるデータセンタ等における異なるロケーション）の複数のホストに記憶される。従って、例えば、特定のデータセットが一つのホストから利用不可能になったとしても（例えば、ホスト障害のために、ネットワークパーティション又は他のネットワーク障害等のために）、クライアントプロセスは、他のホストの当該データにアクセスすることができる。個々のホストは高可用性ではないこともあり得るが、個々のホストの組合せは、より高度の可用性の解決策を提供する。
【０００３】
同一データを複数のロケーションに記憶する場合に直面する問題は、当該データの種々のコピー間における一貫性の保持である。一つのホストに存在するままのデータセットの状態は、他のホストに存在するままのデータセットの状態と一致しないことがある。例えば、クライアントプロセスが一つのホストにおけるあるデータセットに変更を発生させたとすると、当該データセットはその後当該ホストからは利用不可能となり、当該ホストにおける当該データセットのコピーになされた変更は、少なくとも一時的に喪失されることがある。当該データセットの最近のバージョンは、別のホストから得られる可能性がある。しかしながら、クライアントプロセスが他のホストからの当該データセットにおける動作を開始すると、他のデータセットに反映されていない変更をそれぞれ伴う当該データセットの二つのバージョンが潜在的に生成され得るという更なる問題が発生する。
【０００４】
従って、高可用性データを提供することが可能なシステム及び方法に対する継続的な要求が存在する。ある特徴及び利点が議論されているとはいえ、ここでの教示は、それらの特徴及び利点のいずれも必ずしも達成しないシステム及び方法を達成するためにも適用され得るということは理解されるべきである。
【図面の簡単な説明】
【０００５】
【図１】実施の一形態に係るデータ処理システムのブロック図である。
【図２】実施の一形態に係る他のデータ処理システムのブロック図である。
【図３】実施の一形態に係る図１のデータセットサービスを詳細に示すブロック図である。
【図４】実施の一形態に係る図１のシステムにより実行される書込動作のフローチャートである。
【図５】実施の一形態に係る図１のシステムにより実行される読出動作のフローチャートである。

【図６】実施の一形態に係る図１のシステムにより実行されるデータ調整及び更新動作のフローチャートである。
【図７】実施の一形態に係る図１のシステムにおけるデータ複製及び負荷平衡との関連において用いられるハッシュ動作の図である。
【図８】実施の一形態に係る図７に示されるハッシュ動作の他の図である。
【図９】実施の一形態に係る図１のシステムの増加スケーラビリティ機構を示す図である。
【図１０】実施の一形態に係る図１のシステムにおいて用いられるデータ複製構成の図である。
【図１１】実施の一形態に係る図１のシステムにおいて用いられるホスト優先リストの図である。
【図１２】実施の一形態に係る図１のシステムにおいて用いられる負荷平衡構成の図である。
【図１３Ａ】実施の一形態に係る図１のシステムにより実行される書込動作のフローチャートである。
【図１３Ｂ】実施の一形態に係る図１のシステムにより実行される書込動作のフローチャートである。
【図１４Ａ】実施の一形態に係る図１のシステムにより実行されるハンドオフ（hand-off）動作を含む書込動作のフローチャートである。
【図１４Ｂ】実施の一形態に係る図１のシステムにより実行されるハンドオフ（hand-off）動作を含む書込動作のフローチャートである。
【図１５Ａ】実施の一形態に係る図１のシステムにより実行される読出動作のフローチャートである。
【図１５Ｂ】実施の一形態に係る図１のシステムにより実行される読出動作のフローチャートである。
【図１６】実施の一形態に係る図１のシステムにおいて用いられるデータバージョニング（versioning）構成の図である。
【図１７】実施の一形態に係る図１のデータセットサービスを詳細に示すブロック図である。
【図１８】実施の一形態に係る図１７のシステムにおけるデータ複製及び負荷平衡との関連において用いられるハッシュ動作の図である。
【図１９】実施の一形態に係る図１７のシステムにおいて用いられるデータセンタ及びホスト優先リストの図である。
【図２０】実施の一形態に係る図１７のシステムにより実行されるアクセス動作のフローチャートである。
【図２１】実施の一形態に係る図２０のアクセス動作の一局面を詳細に示す図である。
【図２２】実施の一形態に係る図２０のアクセス動作の一局面を詳細に示す図である。
【図２３】実施の一形態に係る図２０のアクセス動作の一局面を詳細に示す図である。
【図２４】実施の一形態に係る図２０のアクセス動作の一局面を詳細に示す図である。
【図２５】実施の一形態に係る図１７のシステムにおいて用いられるメッセージフィルタである。
【発明を実施するための形態】
【０００６】
実施の一形態は、複数のデータセンタ内のホストにデータセットを記憶することを含むコンピュータ実施（computer-implemented）データ処理方法に関する。データセンタ及びデータセンタ内のホストは、例えば、複数段（multi-tiered）リング構成に従って構成され得る。実施の一形態において、ハッシュ構成は、データセットの書込及び読出が発生するデータセンタ及びホストを選択するためのリング構成を実施するために用いられる。他の実施の形態において、バージョン履歴は、ホストにおける書込及び読出も行われ、また、バージョン履歴は、読出が発生した後のデータセット間の一時的な関係の評価に用いら

れる。

【０００７】

発明の詳細な説明及び特定の実施例は、本発明の好適な実施の形態を示すと同時に、限定ではなく例示として与えられるということは理解されるべきである。本発明の範囲内において、その精神から離れることなく、多くの変形及び変更が可能であり、本発明は、総てのそのような変形を包含する。

【０００８】

Ｉ．システム構成（System Architecture）

図１を参照すると、実施の一形態に係るデータ処理システム１００が示されている。データ処理システム１００は、ユーザコンピュータ１０２と、通信ネットワーク１０４と、ネットワークサービスシステム１０６とを含んでいる。ユーザコンピュータ１０２は、通信ネットワーク１０４を介してネットワークサービスシステム１０６にアクセスし得る。ネットワークサービスシステム１０６は、ネットワークインタフェース１１０と、データセットサービス１１２と、一つ又は複数の他のサービス１１４とを含んでいる。ネットワークインタフェース１１０は、通信ネットワーク１０４を介してユーザからデータを受信し、ユーザにデータを供給する。例えば、ネットワークインタフェース１１０は、データセットサービス１１２により保持されているデータセットへのアクセスに加えて、他のサービス１１４により生成され及び／又は保持されている他のデータへのアクセスを、ユーザコンピュータ１０２に提供し得る。

【０００９】

データセットサービスは、データセットを記憶し得るデータ記憶システム１１８を含んでいる。データ状態は、システム１０６内におけるユーザインタラクション（ユーザ相互作用）に基づいて及び／又は他の変化に基づいて時間と共に変化し得る。ここで、用語「データセット」は、時間と共に変化し得る任意のデータに関連する。例えば、各データセットは、当該データセットから付加、除去及び／又は変更され得る一つ又は複数の項目を含み得る。データ記憶システム１１８は、システム障害（例えば、ホスト障害、ネットワーク障害等）の場合には、後述するように、データセットが高度の一貫性を有して利用可能（可用）に存続するように、高可用性の方法において情報を記憶するように構成されている。実施の一形態において、データ記憶システム１１８は、バークリー（Berkeley）データベーストランザクションデータ記憶システムを用いて実現されている。

【００１０】

ここで図２も参照すると、図２は、データ処理システム１００の他の実施例を提供している。図２の例においては、ネットワークサービスシステム１０６は商業ウェブサイトシステム１１６であり、ネットワークインタフェース１１０はネットワークショッピングインタフェース１２０である。商業ウェブサイトシステム１１６は、例えば、数千又はそれ以上のホストを含む分散型計算方式環境において実現され得る。商業ウェブサイトシステム１１６は、アイテム（例えば、商品、サービス、予約申込等）の買い物をするためにユーザコンピュータ１０２を操作するユーザにとってアクセス可能である商業ウェブサイト（例えば、オンライン小売ウェブサイト）を提供し得る。そのような実施の形態においては、ネットワークショッピングインタフェース１２０は、アイテムの表示及び／又は販売を容易にするために、ウェブサイト上のグラフィックデータ及び／又はテキストデータをユーザに提供する。ユーザに提供されるデータは、価格、寸法、利用可能性、購入のために現在選択されているアイテム等のアイテム情報を含み得る。商業ショッピングインタフェース１２０は、ユーザが関心を持っているアイテムを示すデータ、トランザクションの完了を必要とするデータ等、ユーザからのデータを受信するように構成されることも可能である。

【００１１】

図２の実施例においては、データセットサービス１１２は、ウェブサイトのユーザによって購入のために又は購入可能性のために選択されたアイテムのリストを保持するショッピングカートデータサービス１２２となるべきものとして示されている。そのような実施

例においては、各データセットは、特定の顧客に関連するショッピングカートであり得る。データセットは、ショッピングカート内のアイテムのためのアイテム識別情報、ユーザが選択したかもしれないがまだ購入していないアイテムのためのアイテム情報、ショッピングカート内のアイテムの数量情報等を含み得る。ショッピングカートデータサービス１２２は、ショッピングカートに関連する他のビジネスロジックを含み得るショッピングカートサービス１２４を通じてアクセスされ得る。ウェブサイトシステム１１６は、例えばユーザのショッピングカートの全部又は一部を表示するウェブページ等、データセットの全部又は一部を含むウェブページを、ウェブサイトのユーザのために表示し得る。他の例示的な実施の形態においては、データセットは、ユーザのインタラクション（相互作用）に基づいて、又は、訪問者の便宜のために、又は、ウェブサイトの操作を容易にするために、ウェブサイトシステム１１６により収集され得る他のデータを含み得る。例えば、データセットサービス１１２は、特定のエンティティ（構成要素、実体）に関連するデータセット（例えば、ウェブサイトの異なるユーザ、ウェブサイト上の異なるセッション、ウェブサイト上で実施される異なるトランザクション、ウェブサイトによって提供される異なるアイテム、ウェブサイトによって提供されるアイテムの異なるカテゴリ、ウェブサイト上で表示される異なる広告、ウェブサイトの異なるページ等に関連するデータセット）も保持し得る。図２はウェブサイトシステムを示しているが、理解されるであろうように、データ処理システム１００は、他のアプリケーションにおいても使用され得る。

【００１２】

再度図１を参照すると、データセットサービス１１２は、ローカル（局所）プロセス及びリモート（遠隔）プロセスの両方との関連において使用され得る。リモートプロセスとの関連においては、データセットサービス１１２に対する読出及び書込要求は、通信ネットワーク１０４経由でリモートプロセスから受信され得る。例えば、ネットワークサービスシステム１０６は、アプリケーションプログラムインタフェース（ＡＰＩ）を介してインタネットを通じてリモートプロセスにアクセス可能なサービスを提供し得る。そのようなサービス要求は、サードパーティによって、例えばそれら自体のデータ処理システムの動作における援助（アシスト）を行うために、行われ得る。

【００１３】

ここで図３乃至図６を参照すると、データセットサービス１１２の構成及び動作が詳細に示されている。図３に示されるように、データセットサービス１１２は、複数のホスト１３０を含み得る。ここで、用語「複数」は、二つ又はそれより多いことを意味する。例えば、データセットサービス１１２は、数十、数百若しくは数千又はそれより多いホストを含み得る。実施の一形態において、各ホスト１３０は、機能的に等価である（例えば、同一コードを実行し、又は、同一コードの関連するバージョンを実行する）。各ホスト１３０は、後述する図３乃至図１６に記載された動作を実行するように構成された、格納されたプログラムロジックを含み得る。以下に述べられるように、データセット記憶システム１１８は、各ホスト１３０がデータセットの一部を記憶するように、各ホスト１３０に亘って分散される。各ホスト１３０は、（キー値ペア（key-value pairs）の）データのサブセットを記憶し、システムは、各データセットのＮの複製（ここで、Ｎは、複製係数、又は、データセットを複製する回数を表す正の整数である）を保持することを試みる。値Ｎは、設定可能であり、データの耐久性、可用性及び一貫性のいずれにも作用する。システム内にＳの物理的なホストがあるとすると、全システム１０６は、Ｓ≧Ｎの物理的なホストを含み（Ｓが小さいほど全システムの可用性は低くなるが）、各ホスト１３０は、データセットの約Ｎ／Ｓを記憶する。代替的に、異種のホスト１３０が使用される場合、各ホスト１３０は、システム１０６内に重み付けをする各ホスト１３０の重みに比例するいくらかのデータセットを記憶する。各ホスト１３０の重みは、各ホスト１３０の資源に基づいて決定され得る。例えば、各ホスト１３０の重みは、より高性能なホスト１３０がより多くのデータセットを記憶するように、各ホスト１３０の相対的な性能に基づいて（例えば、処理能力、記憶容量、及び／又は、ネットワーク容量に基づいて決定されるように）決定され得る。Ｎの値は、例えば、データセットごと又はデータタイプごとの基準で

可用性／耐久性が設定されることを許容するために、データセットごと又はデータタイプごとの基準で設定可能なものとされ得る。

【００１４】

図４に示されるように、クライアントプロセス１３４（例えば、サービス１１４のうちの一つ）から受信されるデータを記憶するために、データセットサービス１１２は、クライアントプロセス１３４から書込要求を受信し（ステップ１５０）、その後、複数のホスト１３０にデータ書込を行うことによって応答する（ステップ１５２）。（この出願の目的のため、用語「クライアントプロセス」とは、任意の他のプログラムロジックからの、例えばここではデータセットサービス１１２からの、データセットを要求することがある任意のプログラムロジックをいう。）実施の一形態においては、後述するように、データは、優先リストに基づいて複数のホストに書き込まれる。データが書き込まれた後、書込動作が実行されたことを確認する応答が、クライアントプロセス１３４に送信される（ステップ１５４）。例示的な書込動作が、図７乃至図１２、図１３Ａ乃至図１３Ｂ及び図１４Ａ乃至図１４Ｂとの関連において詳細に記載されている。

【００１５】

図５に示されるように、クライアントプロセス１３４にデータを供給するために、データセットサービス１１２は、クライアントプロセス１３４から読出要求を受信し（ステップ１６０）、その後、複数のホスト１３０においてデータ読出を行うことによって応答する（ステップ１６２）。データが読み出された後、読出動作が実行されたことを確認し且つ要求されたデータを含む応答が、クライアントプロセス１３４に送信される（ステップ１６４）。例示的な読出動作が、図１５Ａ乃至図１５Ｂとの関連において詳細に記載されている。

【００１６】

図６に関しては、総ての関連するネットワーク接続及びホスト１３０が健全である（例えば、利用可能であり応答する）場合、読出動作に関係するホスト１３０は、典型的には、一貫性のあるデータを供給する。しかし、一つ又は複数のネットワーク接続又はホスト１３０に故障又は障害が発生している場合、ホスト１３０は、同一データセットの異なるバージョンを供給し得る。従って、図６に示されるように、クライアントプロセスにおいてデータセットが受信された後（ステップ１７０）、データセットは、調整されて一致させられ得る（ステップ１７２）。調整されて一致させられたデータセットは、その後、記憶のためにデータセットサービス１１２に送信され得る（ステップ１７４）。以下に詳細に記載されるように、同一データセットの一貫性のないバージョンの存在は、データバージョニング（data versioning）構成を用いて削除され得る。データバージョニング構成は、一貫性のないバージョンを調整するために、バージョン調整ロジック１３６（図３に示されるように、クライアントプロセス１３４の一部として又はクライアントプロセス１３４との関連において提供される）によっても使用され得る。例示的なデータバージョニング構成が、図１６との関連において以下に詳細に記載される。

【００１７】

ＩＩ．読出／書込動作の協調（Coordination of Read/Write Operations）

Ａ．ホスト間のデータセットの区分化（Partitioning Data Sets over Hosts）

図７乃至図８を参照すると、実施の一形態において、データセットサービス１１２は、システム１０６内のホスト間にデータセットを区分化するための機構を含んでいる。実施の一形態において、以下に記載されるように、一貫性ハッシュ構成は、データがホスト１３０間に相対的に均等に分散されるように、データセットを記憶するために使用され得る。他の実施の形態においては、他のデータ区分化構成が使用され得る。

【００１８】

先ず図７を参照すると、実施の一形態において、データセットサービス１１２によって記憶されているデータにアクセスするために（例えば、読出動作又は書込動作を介して）、クライアントプロセスは、各要求が参照する、データセットのためのキーを含むデータ要求を送信する。例えば、ショッピングカートアプリケーションとの関連においては、キ

ーは、ショッピングカートが関係するユーザのユーザＩＤに基づいて生成され得る（例えば、ユーザＩＤがキーとして使用され得る）。キーは、データセットに関連し且つハッシュ関数への入力としての使用に適当な任意のデータ値であり得る。図７に示されるように、キーは、キーの関数としてのハッシュ値ｈを順次生成するハッシュ関数１８２に適用される。実施の一形態において、ハッシュ関数１８２は、ハッシュ範囲に亘るハッシュ値のほぼ均一な分散を達成する。図示された実施の形態においては、ハッシュ値は、ハッシュ範囲｛０，2^{128}｝に亘って分散するように示されているが、ハッシュ値の任意の数、又は、実際上は任意の大きさのハッシュ範囲が使用され得る。

【００１９】

データセットサービス１１２においてアクティブな（活動状態にある）参加者になると、各ホスト１３０は、ハッシュ範囲に亘る位置のセットを割り当てられる。説明の目的のために、ここでの考察の残りの部分では、データセットサービス１１２を実施するホストが、ホストＡ、ホストＢ、ホストＣ、ホストＤ及びホストＥとして示されている五つのホスト１３０であると仮定する。実際には、データセットサービス１１２が、数十、数百若しくは数千又はそれより多いホスト１３０により実施され得ることは、理解されるであろう。

【００２０】

図８を参照すると、図８は、読出動作又は書込動作に対する責任（分担）がハッシュ値に基づいて特定のホスト１３０に割り当てられる方法を示している。各ホスト１３０は、ハッシュ範囲内におけるそれ自体の位置から先行するホスト１３０の位置までに亘るハッシュ値との関連において読出／書込動作に対して責任を有する。例えば、ホストＡ，Ｂ，Ｃ，Ｄ，Ｅがハッシュ値h_1，h_2，h_3，h_4，h_5にそれぞれ位置しているとすると、ホストＢは、ハッシュ値$h_1 \leqq h \leqq h_2$の範囲に対して責任を有し、ホストＣは、ハッシュ値$h_2 \leqq h \leqq h_3$の範囲に対して責任を有し、以下、同様となる。ホストＡに対する責任（分担）「ラップアラウンド（wrap around）」の割当は、即ち、ホストＡが、ハッシュ値$h_5 < h \leqq 2^{128}$及び$0 \leqq h \leqq h_1$の範囲に対して責任を有することである。実施において、例えば、キーk_1及びk_2を有するデータセットは、キーk_1及びk_2をハッシュすることによりリング１８４上におけるそれらの位置を与えて、ホスト１３０に割り当てられ、その後、リング１８４を時計回りに周回して、データセットのハッシュされたキーよりも大きい値の位置を有する最初のホスト１３０を見出す。キーk_1の場合、対応するデータセットが割り当てられる、より大きい位置を有する最初のホストは、ホストＡである。キーk_2の場合、対応するデータセットが割り当てられる、より大きい位置を有する最初のホストは、ホストＢである。

【００２１】

図７乃至図８に示される構成は、各ホスト１３０がリング１８４上におけるそれ自体とその先行ホスト１３０との間のリング１８４の範囲に対して責任を有するという結果になる。例えば、ホストＢは、それ自体とホストＡとの間のリング１８４の部分に対して責任を有する。ある一つのホスト１３０が参加又は離脱したとすると、そのことは、リング１８４上におけるその直近の後続ホストの責任（分担）に影響を与えるだけであり、他の総てのホスト１３０は影響を受けない。これが図９に示されており、そこでは、ホストＦの追加が、リング１８４上におけるその直近の後続ホストであるホストＢの責任（分担）に影響を与えるが、ホストＡ等の他のホストの責任（分担）には影響を与えない。従って、個々のホスト１３０は、各ホスト１３０に対するデータセットの区分化の全体的な再配置を伴うことなく、追加又は除去されることが可能であり、それにより、増加スケーラビリティ（拡大縮小可能性）（scalability）が増進される。

【００２２】

１．データ複製（Data Replication）

ここで図１０乃至図１１を参照すると、図７乃至図８のハッシュ構成は、データ複製の支援のために使用され得る。図１０においては、リング１８４上における直近の後続ホスト１３０に対して単にデータセットが割り当てられるよりむしろ、データセットは、最初

のＮの後続ホスト１３０に割り当てられている。後述するように、データセットサービス１１２は、ホスト１３０間にデータのＮの複製があり、各ホスト１３０がそれ自体とそのＮ番目の先行ホストとの間のリング１８４の範囲に対して責任を有するということを保証するために機能する。

【００２３】

図１１に示されるように、このような構成においては、各キーは、当該キーに基づいて生成されたハッシュ値からリング１８４を時計回りに周回したときにホスト１３０のそれぞれが最初に遭遇を受ける順序である、ホスト１３０の優先リスト１９０を有している。優先リスト１９０は、データセットにアクセスするために（例えば、読出又は書込のために）使用されるホスト１３０の優先順序を表している。総てのホスト１３０が健全である場合、優先リスト１９０における上位Ｎのホストがデータセットを記憶する。特定のホスト１３０に障害が発生した場合、又は、ネットワークパーティションが発生した場合、データセットは、優先リスト１９０においてより低いランク付けのホスト１３０に一時的に記憶され得る。複数のホスト１３０に障害が発生した場合、データセットは、優先リスト１９０においてより低いランク付けの複数のホスト１３０に記憶され得る。Ｎ＝３のとき、キーk_1に関連するデータセットにアクセスしているクライアントプロセス１３４は、キーk_1の位置からリング１８４を時計回りに周回することによって分かるように、ホストＡ，Ｂ及びＤに（より上位のホストのいずれかが利用不可能である場合には次にホストＥにさらにホストＣに）この順序で読出又は書込を行う。キーk_2に関連するデータセットにアクセスしているクライアントプロセス１３４は、キーk_2のハッシュ位置からリング１８４を時計回りに周回することによって分かるように、ホストＢ，Ｄ，Ｅに（より上位のホストのいずれかが利用不可能である場合には次にホストＣにさらにホストＡに）この順序で読出又は書込を行う。以上に述べたように、値Ｎは設定可能な値であり、従って、データセットのより多くの複製を許容するために、より多くのホスト１３０がシステム１０６に付加され得る。従って、データセットの可用性のレベルは設定可能であり、適当な数のホスト１３０を使用して所望するだけの高さに設定され得る。

【００２４】

総てのホスト１３０が利用可能である場合、同一データセットにおける連続する動作はＮのホストの同一セットにアクセスし、従って一貫性を有する（即ち、動作は、同一キーにおける先行動作により読出／書込が行われた同一データにアクセスする）。ネットワーク又はホストの障害がある場合、同一データセットへの連続する動作は、ホスト１３０の異なるセットにアクセスすることがあるが、動作は、アクセスされるホストのセットにいくらかの重複がある限り、依然として一貫性を有し得る。例えば、キーk_1における第１の動作が、ホストＡ，Ｂ及びＤにアクセスし得る。その後、ホストＢが利用不可能になった場合、キーk_1における第２の動作は、ホストＡ，Ｄ及びＥにアクセスし得る。従って、優先リスト１９０における最上位の利用可能なホスト１３０にアクセスすることにより、動作から動作へのホストの利用可能性の小さい変化は、一貫性に否定的な影響を与えない。その理由は、後続のアクセスが、重複するホストを含み得るからである。（非一貫性という結果になる）ホストセットの間に重複が存在しない状態となるためには、少なくともＮのホストの利用可能性（可用性）は、二つの連続する動作の間に変化しなければならない。以上に述べたように、値Ｎは設定可能な値であり、従って、一貫性の確率保証は設定可能であり、所望するだけ高く設定し得る。これは、大域（global）一貫性（システム応答は、データに対してなされた最新の絶対的な変更を反映する）、及び、主観的一貫性（システム応答は、現在の要求を行っているクライアントによりなされた最新の変更を反映する）の両方の確率保証を含む。

【００２５】

実施の一形態において、データセット上におけるクライアント動作は、複数のロケーション（例えばサーバ）において提供され得る。また、同一データセット上における連続する動作は、異なるサーバによって提供され得る。実施の一形態において、所定のデータセットを記憶するホスト１３０にアクセスするために、サーバは、ホスト１３０の利用可能

性（優先リスト１９０において最上位であるＮの利用可能なホストを選択するために）に加えて、ハッシュ空間におけるホスト位置に関する情報（優先リスト１９０を計算するために）を記憶する。ネットワーク又はホストの障害が存在する際には、異なるサーバが、ホストの利用可能性に関する異なる情報を記憶し得る。システムに結合され又は離脱するホストが存在する際には、異なるサーバが、ハッシュ空間におけるセット位置に関する異なる情報を記憶し得る。例えば、サーバＸは、ホストＡがデータセットサービス１１２に結合したことを認識しないことがある。従って、キーk_1を有するデータセット上における動作の提供において、サーバＸは、ホストＢ、Ｄ及びＥにアクセスし得る。別のサーバＹは、ホストＡとホストＡのハッシュ位置とを既に認識している可能性がある。この情報に基づいて、キーk_1上における後続の動作を提供する際には、サーバＹは、ホストＡ、Ｂ及びＤにアクセスし得る。従って、優先リスト１９０における最上位の利用可能なホスト１３０にアクセスすることにより、書込及び読出動作の間に少なくとも一つのホストにアクセスする確率は増加する。以上に述べたように、一貫性のこの確率保証は、Ｎの値によって決定される。

【００２６】

実施の一形態においては、優先リスト１９０は、ハッシュ関数１８２の動作によって実施され得る（例えば、分離して記憶されることなく）。他の実施の形態においては、優先リスト１９０は、記憶され得る。理解されるであろうように、他の要因が、優先リスト１９０を構成する際に考慮に入れられることがある。優先リスト１９０は、そのような要因を考慮に入れるために、手動で又は自動的に構成され得る。例えば、可用性及び耐久性をさらに改善するために、同一優先リスト１９０内に相互関係障害の確率が比較的低いホスト１３０が含まれるように、優先リスト１９０が構成され得る。例えば、システム１００が複数のネットワークに亘って分散させられている場合、同時に故障することがありそうもないホスト１３０の集合があり得る。従って、システム１００は、データセットのＮの複製のために、障害の相互関係が低くなるようにＮのホストを選択することによって、可用性及び耐久性を最大化することができる。同様に、低い障害相互関係は、ホスト１３０が異なるハードウェア上で運転され、異なるプログラムロジック実行を用い、地理的に異なる領域において運転され、それらが組み合わせられる場合にも存在し得る。例えば、リング１８４を時計回りに周回するときに、遭遇したホスト１３０が、考慮されるべきことが望ましい任意の追加的な標準（criteria）に適合するか否かの評価に、規則の集合が適用され得る。遭遇したホスト１３０が追加的な標準に適合しない場合、利用可能なホストの検索が、追加的な標準に適合するホストに遭遇するまで、リング１８４の周回を前進して継続され得る。

【００２７】

他の構成も、地理的な多様性を実現するために用いられ得る。そのような構成の実施例が、図１７乃至図２５との関連において以下に詳細に記述される。

【００２８】

２．負荷平衡（Load Balancing）

図１２を参照すると、ホスト１３０は、負荷平衡を促進するために、即ち、データ及び負荷の不均一な分散を回避するために、リング１８４上の複数の位置に割り当てられ得る。そうでなければ、リング１８４上における各ホスト１３０の無作為な位置割当によって、データ及び負荷の不均一な分散が発生し得る。従って、図１２において、ホストＡ，Ｂ，Ｃ，Ｄ，Ｅは、リング１８４上の複数の位置に割り当てられている。この複数の位置決めは、各ホスト１３０に割り当てられたデータセットの数の分散を減少させる傾向がある。その理由は、リング１８４上における増加した多数の無作為の配置が、各ホスト１３０に割り当てられたデータセットの数に平均値への集中を引き起こす傾向があるからである。従って、より多くの位置をリング１８４上における各ホスト１３０に割り当てることは、負荷平衡を改善する。実施の一形態において、遭遇した各ホスト１３０の最初の実例だけが優先リスト１９０に配置される。キーk_1の場合、対応するデータセットが割り当てられている、より大きい位置の最初のホストは、ホストＡである。Ｎ＝４の場合、キーｋ

$_1$に関連するデータセットにアクセスするプロセスは、ホストＡ，Ｂ，Ｃ，Ｄに対して読出又は書込を行う。キーｋ$_1$のための優先リスト１９０は、リング１８４上における複数の位置を有するホストのために、及び、異なる順序で遭遇するホストのために、上記とは異なる。キーｋ$_2$の場合、対応するデータセットが割り当てられている、より大きい位置の最初のホストは、ホストＢである。キーｋ$_2$に関連するデータセットにアクセスするクライアントプロセス１３４は、ホストＢ，Ｃ，Ｄ，Ａに対してその順序で読出又は書込を行う。他の例示的な実施の形態においては、遭遇する各ホスト１３０の複数の実例は、例えば、以前は利用不可能であったホスト１３０を再試行するために、優先リスト１９０に配置され得る。

【００２９】

ホスト１３０をリング１８４上の複数の位置に割り当てることは、異種のハードウェアの使用を容易にし、即ち、より性能の高いホスト１３０がリング１８４上のより多くの位置に割り当てられ、より性能の低いホスト１３０がリング１８４上のより少ない位置に割り当てられ得る。例えば、図１２において、ホストＥは、他のどのホストよりも少ない位置を有しており、従って、性能の低いホストであると想定される。理解されるであろうように、あるホストの範囲が使用されることがあり、各ホストは、他のホスト１３０より性能が高いか又は低い。特定のホスト１３０に割り当てられている位置の数は、当該特定のホスト１３０の相対的な性能の関数であり得る。

【００３０】

加えて、十分な数の位置が各ホスト１３０に割り当てられたとすると、各ホスト１３０は、他のホスト１３０のそれぞれと後続／先行関係を有し得る。従って、ホスト１３０のうちの一つが利用不可能になり又は使用解除されたとすると、使用解除されたホストにより処理されていた負荷は、データ可用性を失うことなく残余の利用可能なホスト１３０に亘ってほぼ均等に分散させられ得る。同様に、あるホスト１３０が再度利用可能になった場合又は新たなホスト１３０がデータセットサービス１１２に追加された場合は、その新たに利用可能になったホスト１３０は、他の利用可能なホスト１３０のそれぞれからおおよそ均等な量の負荷を軽減し得る。

【００３１】

Ｂ．読出／書込アクセス動作（Read/Write Access Operations）

ここで図１３Ａ乃至図１３Ｂ、図１４Ａ乃至図１４Ｂ及び図１５Ａ乃至図１５Ｂを参照すると、読出及び書込動作が示されている。読出／書込動作は、クライアントプロセス１３４によりデータセットサービス１１２に対して行われるサービス要求によって引き起こされ得る。サービス要求を受信すると、データセットサービス１１２は、要求された動作を実行し、クライアントプロセス１３４に対して応答を供給する。

【００３２】

データセットサービス１１２においては、ホスト１３０のうちの一つが、読出又は書込要求を調整するための責任を有する。読出又は書込要求を調整するための責任を有するホスト１３０を、ここではコーディネータ（調整者）と称する。実施の一形態において、コーディネータは、優先リスト１９０に載せられた最初のホスト１３０であり、ローカル読出又は書込動作の実行を含む読出又は書込要求を調整する。例えば、サービス要求は、最初は別のホスト１３０によって受信されることがあり、当該ホスト１３０は、コーディネータとして機能するホスト１３０（例えば、優先リスト１９０における最上位ホスト）にそのサービス要求を転送することを決定し得る。他の実施の形態においては、コーディネータは、優先リスト１９０に載っていないホスト１３０等の他のホスト１３０であり得る。例えば、コーディネータは、偶然に最初に読出又は書込要求を受信したが、たまたま優先リスト１９０の最上位付近にはなく、優先リスト１９０の最上位付近にあるホストにそのサービス要求を転送することを決定しないホスト１３０であり得る。実施例を提供する目的のために、ここでは、コーディネータは優先リスト１９０に載せられた最初のホスト１３０であるものとする。

【００３３】

実施の一形態においては、上述のように、読出及び書込動作は、潜在的にダウンしている又はアクセス不可能なホスト１３０をスキップしながら、優先リスト１９０における上位Ｎの健全なホストにアクセスし得る。総てのホスト１３０が健全である場合は、あるキーの優先リスト１９０における上位Ｎのホストがアクセスされ得る。ホスト障害又はネットワークパーティションが存在する場合、優先リスト１９０におけるより下位のホスト１３０が代わりにアクセスされることがあり、それにより高可用性が維持される。

【００３４】

先ず図１３Ａ乃至図１３Ｂを参照すると、例示的な書込動作が示されている。図１３Ａにおいて、バージョンV_{n+1}のための書込要求が、ホストＡによってクライアントプロセス１３４から受信される（上述のように、直接に又は間接に、のどちらでも）。図１２に示されるようなリング１８４上におけるホスト１３０の分散を想定すると、キーk_1のための優先リスト１９０は、P_L＝｛Ａ，Ｂ，Ｃ，Ｄ，Ｅ｝である。この実施例において、ホストＡはコーディネータであり、書込動作をローカルに実行する（ステップ１５０）。ホストＡは、その後、新しいバージョンV_{n+1}を残余の最上位にランクされているＮ－１の到達可能なホストであるホストＢ及びＣにコピーし（例えば、Ｎ＝３のとき）、それはその後に、書込動作の実行及び追加的なコピーの記憶も行う（ステップ１５２）。

【００３５】

データセットが記憶されるとき、データ自体に加えて、当該データに関連するキー、及び、ベクトルクロックも記憶される。キーは、データセットが後に識別されることを可能にする。ベクトルクロックは、同一データセットの異なるバージョン間の因果関係を捕捉するためのデータバージョニングのために使用され、データセットのバージョンに関連する｛ホストＩＤ，カウンタ｝ペアのリストを含む。ベクトルクロックの使用を通じたデータバージョニングは、図１６との関係において詳細に後述される。

【００３６】

図１３Ｂにおいては、ホストＢ及びＣは、書込動作が成功したか否かをホストＡに折り返し報告し、ホストＡは、書込動作が成功したか否かの確認をクライアントプロセス１３４に応答する（ステップ１５４）。実施の形態において、成功と考えられるべき書込動作のために、書込動作は、Ｗが設定可能な値であってＷ≦Ｎであるとすると、Ｗのホストにおいて成功しなければならない。従って、例えば、Ｎ＝３かつＷ＝２とすると、書込動作は、三つのホスト１３０において試行されたとしても、二つのホスト１３０において成功すれば、成功であると考えられる。書込動作が一つ又は複数のホスト１３０において成功したとすると、データセットのコピーは、詳細に後述するように、依然としていつかは優先リスト１９０における上位Ｎのホストに移動し得るということを意味し得る。従って、上述したテストに従って書込動作が成功であると考えられなかったとしても、上位Ｎのホストにおけるデータセットの最終的な一貫性は依然として実現され得る。

【００３７】

図１４Ａ乃至図１４Ｂを参照すると、データハンドオフを伴う例示的な書込動作が示されている。データハンドオフは、あるデータセットのための優先リスト１９０における最上位にランクされているＮのホストに対してデータを移動させることを試行する機構又は手順である。例えば、上述のように、コーディネータは、一般に、優先リスト１９０における上位Ｎのホストに対してデータの送信を試行する。しかしながら、一つ又は複数のホスト１３０がダウンしている場合、コーディネータは、優先リスト１９０におけるより下位のホスト１３０に対してデータを送信する。優先リスト１９０は、書込動作に（及び読出動作に）関係することとなるホスト１３０の適切に規定された順序を提供し、データハンドオフ機構は、優先リスト１９０における最上位にランクされているＮのホストに対してデータを移動させて返すために使用される。

【００３８】

従って、図１４Ａに示されるように、ホストＡは、図１３ＡのようにバージョンV_{n+1}のための書込要求を受信する。ホストＡは、その後、書込動作を実行し、残余の最上位にランクされているＮの到達可能なホストであるホストＢ及びＣに新しいバージョンをコ

ピーすることを試行する。図示された実施例においては、ホストＣが一時的に故障しており、従って、ホストＤにおける書込が試行される。ホストＤにおいて書き込まれたデータは、そのいくらか後の時点でホストＤが当該データをホストＣに転送し得るように、どのホスト１３０がデータを受信して書込を行うべきか（例えば、ホストＣ）を提案するヒントと共にタグを付され得る。図１４Ｂにおいて、ホストＣが健全である場合、データハンドオフが行われ、データはホストＣにコピーし戻される（copied back）。データは、このようにして、優先リスト１９０における最上位にランクされたＮのホストの一つであるホストCに移動し戻される。

【００３９】

実施の一形態においては、関連する技術がデータセットの喪失したコピーを回復するために使用され得る。例えば、ホスト１３０が参加又は離脱して優先リスト１９０に対応する変更があるとすると、データの不適切な配置が発生し得る。例えば、システム１００に追加されたホスト１３０は、優先リスト１９０における他のホストの序列に不適切な配置を発生させる。そのような状況において、データハンドオフを実行するために、ホスト１３０のペアは、それらが共通に共有する領域の比較を定期的に実行し、その後、比較の間に検出されたいかなる差異も調整して一致させるために、必要なデータ転送を実行し得る。例えば、キーの領域を自体のために保持している、上位Ｎのホストの一つではないホスト（送信者）は、上位Ｎのホストの任意の一つ（受信者）を無作為に選択し得る。別の実施例として、上記ホストは、上位Ｎのホストのなかから、例えば、当該データを有している可能性の低いホストを選択し得る。その理由は、上記ホストがデータセットサービス１１２に最近参加したものだからである。上記二つのホスト１３０は、その後、低レベルデータベース比較を継続してもよく、送信者は、比較によって検出されたいかなる差異も調整して一致させるために、受信者が記憶しているものよりも新しいデータセットを転送し得る。データは、優先リスト１９０における少なくとも一つのホスト１３０に移動されることが可能であり、従って、優先リスト１９０における残余のホスト１３０に増殖される。例えば、残余のホスト１３０への増殖は、いくつかのキーのセットに対する優先リスト１９０における上位Ｎのホストに含まれているホスト１３０のペアに記憶されているデータセットを比較することによって実行され得る。実施の一形態においては、二つのホストに記憶されているデータ間の差異を効果的に見出すために、マークルツリー（Merkle tree）が使用され得る。例えば、マークルツリーは、ツリーの各ノードがそのサブツリー（subtree）におけるデータから算出された合計（又はハッシュ値）を含んでいるところで、及び、リーフ（leaves）が一つ又は複数のデータ値（例えば、キー、バージョン及びクロック）のハッシュを含んでいるところで、使用され得る。ツリーのコンテンツにおける差異は、ブランチ（branch）に沿ってデータ合計（ハッシュ値）が異なっている当該ブランチを再帰的に展開する（recursing down）ことによって見出され得る。比較の効果を改善するために、マークルツリーは、ブルームフィルタ（Bloom filter）を使用してコード化（符号化）され得る。

【００４０】

上述した機構を用いて、データセットサービス１１２は、データセットの最も新しいバージョンのコピーをその優先リスト１９０における上位Ｎのホストに動的に移動するために、進行中の試行を行う。従って、データセットの最も新しいバージョンのコピーが最初はその優先リスト１９０におけるより下位のホスト１３０にコピーされることがあったとしても、当該コピーは最終的には優先リスト１９０における上位Ｎのホストに移動し戻され、上位Ｎのホストにおけるデータセットの最終的な一貫性に帰着する。

【００４１】

図１５Ａ乃至図１５Ｂを参照すると、優先リスト１９０を用いて実行される例示的な読出動作１４８が示されている。図１５Ａにおいて、読出要求は、ホストＡによってクライアントプロセス１３４から受信される（上述のように、直接に又は間接に、のどちらでも）（ステップ１６０）。ホストＡは、ローカル読出を行うためにホストＢ及びＣからのデータを並行に要求することにより読出動作を調整する。ホストＢ及びＣは、要求された読

出動作を実行する。図１５Ｂにおいて、ホストＡは、ホストＢ及びＣから読出結果を受信し（ステップ１６２）、クライアントプロセス１３４に対して応答を供給する。

【００４２】

読出要求を受信するとき、コーディネータは、そのキーに対する優先リスト１９０における最上位にランクされているＮの到達可能なホスト１３０からの、そのキーに対するデータの総ての存在するバージョンを要求することがあり、その後、クライアントプロセス１３４に対して結果を戻す前に、Ｒの応答を待つ（ここで、Ｒは、良好な読出動作に参加することが必要とされるホストの数である）。図１５Ａ乃至図１５Ｂの実施例においては、値Ｒは、３に等しく設定される。

【００４３】

値Ｗと同様に、値Ｒは設定可能である。例えば、Ｒ＝１であるとすると、ホストＡは良好な読出について一度応答し、その読出からのデータは、クライアントプロセス１３４へ使用のために戻される。別の実施例として、Ｒ＝２であるとすると、ホストＡ及びＢの両方において読出が実行されるまでデータが戻されないことがある。二つの読出が実行されると、システム１００は、データが同一バージョンであることを認識し、Ｒ＝１のときと同一のデータを戻す。さらに別の実施例として、Ｒ＝３であるとすると、ホストＡ，Ｂ及びＣにおいて読出が実行されるまでデータが戻されないことがある。

【００４４】

値Ｒ及びＷは、一貫性及び高性能を提供するために、Ｎより小さくなるように設定され得る。Ｒ＋Ｗ＞Ｎとなるような値Ｒ及びＷの設定は、読出及び書込動作に関係するホスト１３０のセットの間における重複の設定可能な高い確率が存在するクォーラムのようなシステム（quorum-like system）をもたらす。より高いＮが設定されると、少なくとも一つの複製が存在する可能性が高いので、システムは可用性及び耐久性を有する可能性が高くなる。一方、データは、ホスト１３０の同一セットに書き込まれ又はホスト１３０の同一セットから読み出される必要がないことは注目され得る。例えば、データセットは、優先リスト１９０におけるより下位のホスト１３０に書き込まれ、優先リスト１９０におけるより上位のホスト１３０にデータハンドオフを通じて移動されることがあり、その後、最終的には、優先リスト１９０におけるより上位のホスト１３０から読み出される。優先リスト１９０における上位Ｎのホストにおけるデータセットの可能性のある一貫性は、達成される。他の実施の形態においては、Ｒ及びＷは、Ｎよりも十分に小さくなるように（例えば、Ｒ＋Ｗ＜Ｎ）設定されることがあり、データセットのコピーは、（コーディネータに加えて）Ｗ－１のホストにのみ送信され得る。そのような実施の形態においては、上述のデータ回復機構は、上位Ｎのホストの残余のものにデータセットを増殖させるために使用され得る。

【００４５】

実施の一形態において、データセットサービス１１２のためのアプリケーションプログラムインタフェースは、以下のように設定され得る。例えば、コマンドは、以下の形態を有し得る。

【００４６】

write(Key, Value, Context)→ResultCode
（書込（キー、値、コンテキスト）→結果コード）
read(Key) →ValueList, Context, ResultCode
（読出（キー）→値リスト、コンテキスト、結果コード）

ここで、キー（Key）は、バイトの非有界シーケンスであり、値（Value）は、データ（バイトの非有界シーケンス）及びメタデータ（値が書き込まれた最終時刻を含む、値についての情報、ダイアグノスティック（診断）及びデバッグ情報等を包含する、読出専用の、任意の、拡張可能なデータセット）を含むオブジェクトであり、値リスト（ValueList）は、値（Value）のリストであり、コンテキスト（Context）は、読出－変更－書込サイクルに対するベクトルクロック状態をトラック（追跡）するために記憶システムによって内部構造において使用されるオペーク（opaque）オブジェクト（不透明なオブジェクト）で

あり、結果コード（ResultCode）は、読出又は書込動作が成功したか否かについてのコード表示である。

【００４７】

書込動作は、介在する書込がそのキーについて既に起こったことを意味するコンテキストの無効が発生していない限り、キーによって識別される値を、値パラメータによって特定される値に変更する。実施の一形態において、クライアントプロセス１３４は、読出－変更－書込サイクル（楽観的ロック（optimistic locking））を再始動する。他の実施の形態においては、クライアントプロセス１３４は、データセットのコンフリクト（矛盾）するバージョンが存在し得る場合に、書込動作が継続することを許可し得る。読出動作は、キーに関連する値のための、データセットサービス１１２内におけるルックアップを実行する。良好に読み出された任意の及び総ての値は、値リストに戻される。オペークコンテキストオブジェクトは、後続の更新動作において使用するために戻される。複数の値が戻される場合、クライアントプロセス１３４は、総ての値に対して調整動作を実行することが期待される。後続の更新が（戻されたコンテキストを使用して）実行される場合、更新された値が値リストに戻された総ての値の調整を表すという仮定は、（もしあれば）その値に任意の追加的な変更を加える。

【００４８】

理解されるであろうように、アプリケーションプログラムインタフェースにおける複雑性のより大きい又は小さいレベルが用いられ得る。例えば、実施の一形態において、値オブジェクトは、どの程度の期間だけデータが維持されるべきか、例えば、その結果、古い／放棄されたデータが最終的に削除されてもよいか、に関する情報が特定されることを可能にするタイプパラメータを含み得る。

【００４９】

他の実施の形態において、キーは、二つの部分（パーティションキー、オブジェクトキー）に分割されて使用され得る。そのような実施の形態においては、パーティションキーは、キーパラメータに関して上述したように、そのキーに対する優先リスト１９０を生成するためにハッシュされ得る。同一のパーティションキーを共有する二つのデータセットは、従って同一の優先リスト１９０を有し、それ故に非常に高い確率でそれらのデータセットのそれぞれのコピーは、ホスト１３０の同一セットに属することとなる。パーティションキーを共有する総てのキーに対する優先リスト１９０の上位Ｎのホストのなかにホストの同一セットがあるので、そのような機構は、数個のデータセットに同時にアクセスすることを最適化として許容する。例えば、図２の商業ウェブサイトの例において、特定のユーザに関連する総てのデータセット（例えば、ショッピングカート、プロファイル、クレジットカード情報等）をホスト１３０の同一セットに記憶することは望ましいことであり得る。それらのデータセットのそれぞれについて同一のパーティションキーを使用することにより、データセットは、ホスト１３０の同一セットに記憶される。（パーティションキー、オブジェクトキー）組合せは、ユーザに対する各個人のデータセットを一意に識別する。この構成によって可能となる他の最適化は、パーティションキーを共有するキーにおける範囲問合せ（range query）である。例えば、そのような範囲問合せは、所定のパーティションキーに対する優先リスト１９０の上位Ｎのホストのうちの単一ホスト１３０にアクセスすることによって、そのパーティションキーに対する総てのオブジェクトキーを通して繰り返すために使用され得る。

【００５０】

他の実施の形態においては、書き込まれているデータのタイプをクライアントプロセス１３４が特定し得るように、タイプパラメータは、書込コマンド（例えば、write(Key, Value, Context, Type)→ResultCode（書込（キー、値、コンテキスト、タイプ）→結果コード））に付加され得る。データセットサービス１１２は、データが最後にアクセスされてから所定量の時間経過後に、（例えば、そのデータが最早必要とされない場合に、記憶領域を再利用するために）そのデータを削除するように設定され得る。削除前に許容される時間は、データのタイプに基づき得る。タイプは、（例えば、データのいくつかのタイ

プは他のタイプのデータよりも重要であり得るということに基づいて）データセットサービス１１２が記憶すべきデータのコピーの数を決定するためにも使用され得る。

【００５１】

他の実施の形態において、読出コンテキストは、読出コマンドに対する入力として通過させられることもあり得る（例えば、read(Key, Context) →ValueList, Context, ResultCode（読出（キー、コンテキスト）→値リスト、コンテキスト、結果コード））。そのような実施の形態においては、読出コマンドに対する入力として通過した読出コンテキストは、先行する読出の結果として得られることがある。読出動作に対する入力として読出コンテキストを通過させ戻すことにより、クライアントプロセス１３４は、先行する読出動作の間にアクセスされたデータセットの特定のバージョンの検索において重要性を指示し得る。理解されるであろうように、アプリケーションプログラムインタフェース上における他のバージョンもあり得る。

【００５２】

ＩＩＩ．データバージョニング（Data Versioning）

Ａ．ベクトルクロックの動作（Operation of Vector Clocks）

図１６を参照して、データバージョニング構成について説明する。前述したように、高可用性を提供するために、データセットサービス１１２は、同一データの複数のバージョンが異なるホスト１３０上に同時に存在することを許容する。データセットの最も新しいバージョンのコピーをその優先リスト１９０における上位Ｎのホストに移動するために、進行中の試行が行われるが、このプロセスは瞬間的なものではない。移動が発生する前には、データセットのより古いバージョンのコピーが、その優先リスト１９０におけるいくつかのホストに、優先リスト１９０における最上位又は最上位に近いホストにさえ、存在し得る。従って、例えば、一つのホスト１３０が一時的に失われた古い変更を反映した一つのバージョンを有し、別のホスト１３０が古い変更が利用不可能になると同時になされた新しい変更を反映した別のバージョンを有することがある。

【００５３】

実施の一形態において、同一のデータセットの二つのコピーがそのデータセットの異なるバージョンであって相互に関して差異を有しているか否かを判定することができるようにすることが望ましい。二つのバージョンが相互に先祖と子孫の関係にある（例えば、一方のバージョンが単に古くなっただけで他方のバージョンに組み込まれている）状況を、二つのバージョンがコンフリクト（矛盾）する（例えば、各バージョンが他方のバージョンに反映されていないデータを含んでいる）状況から区別することができるように、それらの差異にアクセスすることができるようにすることも望ましい。

【００５４】

実施の一形態において、バージョン履歴は、データセットの各コピーと共に記憶される。例えば、バージョン履歴は、同一データセットの異なるバージョン間の因果関係を捕捉するベクトルクロックの形態で記憶され得る。ベクトルクロックは、二つのバージョンがコンフリクトするか否かを判定することが可能となるように、データセットのバージョン履歴に関する十分な情報を簡潔に記憶し得る。実施の一形態において、ベクトルクロックは、データセットのバージョンに関連する｛ホストＩＤ、カウンタ｝（｛host ID, counter｝）ペアのリストを含む。ホストＩＤ（host ID）値は、書込動作を調整（コーディネート）したホストを示す。カウンタ（counter）値は、そのホストがデータセットに書込を行った回数を示している。カウンタ値は、データバージョンについての因果関係、即ち、どのような変更がそのバージョンに先行するかについての要約をコード化（符号化）する。

【００５５】

データセットの二つのバージョンが因果関係を示す順序を有するか（従って一方を無視してもよいか）又は並列のブランチ（分岐）であるか（従って調整が必要であるか）の判定を試みる場合、それらのベクトルクロックを検査するだけで十分である。一つのベクトルクロックが、他のベクトルクロックのなかで総てのホストＩＤに対して最大の又は最大

に等しいカウンタ値を有している場合、前者は後者の子孫であり、後者は無視することができる。従って、ベクトルクロックは、データ展開の複数のブランチをまとめて一つに戻すために、同一データの複数のバージョンを調整することをクライアントプロセス１３４に可能にさせる。

【００５６】

図１６は、データセットサービス１１２により使用され得るデータバージョニングの実施例を示している。ステップ４００において、最初は、データセットは空（empty）である。ステップ４０２において、クライアントプロセス１３４は、ホストＡを使用して空データバージョンV_0を更新する。書込を調整（コーディネート）するホストＡは、先行するバージョンのクロックをコピーし、ホストＡに関連するカウンタ値を増加させ、データバージョンV_1に対するベクトルクロックを生成する。この場合、これが最初の更新なので、カウンタは、１にインクリメントされる。データセットサービス１１２は、データバージョンV_1及びそれに関連するベクトルクロック［（Ａ，１）］を記憶する、例えば、ホストＡが、ローカル書込動作を実行し、さらに、追加的なローカル書込動作を実行して追加的なコピーを記憶するために新しいバージョンをホストＢ及びＣに送信する。データセットサービス１１２が商業ウェブサイトシステム内にショッピングカート情報を記憶する例示的な実施の一形態において、この更新は、アイテムをショッピングカートに追加するビジターのために発生し得る。理解されるであろうように、データセットの新しい「バージョン」を構成するものは、アプリケーションに依存して変化し得る。

【００５７】

図１６において、コーディネータは、優先リスト１９０における最上位にランクされているＮの到達可能なホストの一つである。上述のように、コーディネータは、コーディネータは、優先リスト１９０における最上位にランクされているＮの到達可能なホストの一つではないホスト１３０であり得る。そのような実施例においては、書込要求を受信したとき、コーディネータは、新しいバージョンに対するベクトルクロックを生成してその新しいバージョンをローカルに記憶するために、そのキーに対する優先リスト１９０における最上位にランクされているＮの到達可能なホストの一つを選択し得る。その後、コーディネータは、既に説明したように、残余の最上位にランクされているＮの到達可能なホストにその新しいバージョンを（その新しいベクトルクロックと共に）送信し得る。

【００５８】

ステップ４０４において、同一のクライアントプロセス１３４は、ホストＡを使用してデータバージョンV_1を更新する。書込を調整（コーディネート）するホストＡは、先行するバージョンのクロックをコピーし、ホストＡに関連するカウンタ値を２に増加させ、データバージョンV_2に対するベクトルクロックを生成する。再度、ホストＡは、データバージョンV_2及びその関連するベクトルクロック［（Ａ，２）］をホストＢ及びＣに転送し、追加的なコピーを記憶する。バージョンV_2はバージョンV_1に由来し、従ってバージョンV_1を上書きするが、まだバージョンV_2を認識していないホストパーティションに残存しているバージョンV_1の複製が存在し得る。

【００５９】

ステップ４０６において、要求を調整（コーディネート）するために、同一プロセスがホストＢを使用してデータバージョンV_2を更新する。新しいホストＢが更新を調整（コーディネート）するので、新しいベクトルクロックエントリが、このホストＢとの関連においてカウンタ値１で生成される。データセットサービス１１２は、データバージョンV_3及びその関連するベクトルクロック［（Ａ，２）；（Ｂ，１）］を記憶する。データバージョンV_2に対するベクトルクロックも、バージョン履歴を保持するために又は実行されるべきより複雑な調整を可能とするために、所望される場合には記憶され得る。ステップ４０６の後、バージョンV_1は認識しているがバージョンV_2は認識していないホストが、バージョンV_3及びその関連するベクトルクロックを受信することがある。そのホストは、バージョンV_1及びバージョンV_3のそれぞれのベクトルクロック［（Ａ，１）］及び［（Ａ，２）；（Ｂ，１）］を比較することによって、バージョンV_1が因果関係に

おいてバージョンV_3に先行し従ってバージョンV_3により上書きされるべきことを意味していることを判定することができる。一方、事象の異常なシーケンスが発生して、データバージョンV_3に対するベクトルクロックがバージョンV_1のクロックにおける総てのホストに対して小さい又は等しいカウンタを有する場合、バージョンV_3は、バージョンV_1の先祖であり、抹消され得る。

【００６０】

ステップ４０８において、異なるクライアントプロセス１３４は、バージョンV_2を読み出し、ホストＣを使用してバージョンV_2の更新を試行する。例えば、ホストＢ及びＣは、利用不可能である場合がある。本実施例においては、ホストＣが、ステップ４０６の書込動作に関係しておらず、バージョンV_3を認識していないものとする。新しいホストＣが更新を調整（コーディネート）するので、新しいベクトルクロックエントリは、このホストＣとの関連においてカウンタ値１で生成される。データセットサービス１１２は、データバージョンV_4及びその関連するベクトルクロック［（Ａ，２）；（Ｃ，１）］を記憶する。ステップ４０８において、バージョンV_1又はバージョンV_2を認識しているホストは、バージョンV_4及びその関連するベクトルクロックを受信したときに、バージョンV_1及びバージョンV_2が新しいデータによって上書きされること及び抹消され得ることを、決定し得る。

【００６１】

ステップ４１０において、クライアントプロセス１３４は、バージョンV_3及びバージョンV_4の両方を読み出す。例えば、読出動作は、ホストＡによって調整（コーディネート）されることがあり、かつ、ホストＢ及びＣも関係することがある。ホストＡは、ベクトルクロック［（Ａ，２）］を有するデータセットのそれ自身のコピーと、ベクトルクロック［（Ａ，２）；（Ｂ，１）］を有する、ホストＢからのデータセットのコピーと、ベクトルクロック［（Ａ，２）；（Ｃ，１）］を有する、ホストＣからのデータセットのコピーとを得る。当該読出のコンテキストは、バージョンV_3及びバージョンV_4のクロックの要約、即ち、［（Ａ，２）；（Ｂ，１）；（Ｃ，１）］である。ベクトルクロックの検査からは、バージョンV_3及びバージョンV_4のそれぞれには、相互に反映されていない変更が含まれているので、ホストＡは、バージョンV_3とバージョンV_4との間にいかなる因果関係も存在しないことを見出すことになる。従って、バージョンV_3及びバージョンV_4は調整される。

【００６２】

実施の一形態において、データセットサービス１１２（この例では、ホストＡ）は、調整をどのように実行するかを順番に決定するクライアントプロセス１３４（及び／又はクライアントプロセス１３４に関連するバージョン調整ロジック１３６）に複数のバージョンを提供する。この構成は、調整を実行するために使用される任意のビジネスロジックに、記憶され、又は、データセットサービス１１２よりむしろクライアントプロセス１３４と関連付けられることを可能とする。クライアントプロセス１３４とバージョン調整ロジック１３６とは分離したものとして示されているが、クライアントプロセス１３４とバージョン調整ロジック１３６とが統合された形態で提供され得ることは理解されるであろう。他の実施の形態においては、バージョン調整ロジック１３６は、データセットサービス１１２と共に提供され得る。複数のバージョンは、例えば、いずれを保持するかを決定するためのバージョン上のデフォルト序列を使用することにより、単一の調整されたバージョンを生成するために異なる複数のバージョンを併合することにより、データの分析を実行して矛盾をどのように取り扱うかを矛盾ごとに決定することにより、等々により、調整され得る。理解されるであろうように、アプリケーションに依存して、異なる方策が、異なる状況ではより最適であり得る。

【００６３】

ステップ４１２において、書込要求がクライアントプロセス１３４から受信される。ホストＡは、書込を調整（コーディネート）し、ベクトルクロックにおける対応するカウンタ値を更新する。更新されたバージョンは、調整動作に関係しない、クライアントプロセ

ス１３４により実行された他の変更を含み得る。新しいバージョンV_5は、ベクトルクロック［（Ａ，３）；（Ｂ，１）；（Ｃ，１）］を有することになる。

【００６４】

データセットに対して調整に加えてどのような変更が実行されたかに拘わらず、ステップ４１２において、ホストＡがカウンタ値を［（Ａ，３）；（Ｂ，１）；（Ｃ，１）］に更新することは理解されるであろう。ベクトルクロック［（Ａ，２）；（Ｂ，１）；（Ｃ，１）］を有するいかなるバージョンも存在しないので、ベクトルクロック内のカウンタの更新は、親クロックを新しいクロックから区別する。加えて、複数のクライアントプロセスが同時に調整を試行し得る（例えば、調整のために異なるホストを使用して）が異なる結果に到達し得る（例えば、変更のみならず調整も加える異なる併合ロジック等のために、）ので、カウンタを増加させることが望ましい。もしカウンタが更新されなければ、異なる併合試行が同一クロック即ち［（Ａ，２）；（Ｂ，１）；（Ｃ，１）］に割り当てられることがあり、従って、相互に区別することができなくなる。

【００６５】

Ｂ．ベクトルクロック情報及び切捨（Vector Clock Information and Truncation）

実施の一形態において、｛ホストＩＤ、カウンタ｝（｛host ID, counter｝）ペアのみを含むよりもむしろ、ベクトルクロックは、多数の追加的な値を含み、以下のような形態を有する。

【００６６】

ベクトルクロック
＝｛(＜ホストＩＤ＞＜ホスト情報＞＜キー情報＞),＜カウンタ＞,＜タイムスタンプ＞｝
（ VectorClock
＝｛(＜HostID＞＜host-gen＞＜key-gen＞),＜counter＞,＜time-stamp＞｝）
ホストＩＤ（host ID）は、ホストに対する固有の識別子であり、カウンタパラメータは、データバージョンに対応する因果関係情報をコード化（符号化）し、上述した｛ホストＩＤ、カウンタ｝（｛host ID, counter｝）ペアに対応する。実施の一形態において、(＜ホストＩＤ＞＜ホスト情報＞＜キー情報＞)((＜HostID＞＜host-gen＞＜key-gen＞))パラメータの組合せは、ホストＩＤ単独に関して前述したような態様で機能する。即ち、三つのパラメータ(＜ホストＩＤ＞＜ホスト情報＞＜キー情報＞)のいずれか一つでも異なっていれば、ホストは、異なるホストであると考えられる（即ち、データセットの異なるバージョン間にいかなる因果関係も伴い得ない）。

【００６７】

実施の一形態において、ホスト１３０は、ベクトルクロックを同時にディスクには書き込まない。従って、ホストが各キーに対して生成されたシーケンス番号を見落とすことがあり、その結果そのシーケンス番号が再使用され、それによりベクトルクロックの一貫性が損なわれるという可能性が存在する。見落としの危険性（例えば、ホスト障害後に）が認識されると、ホスト１３０は、その＜ホスト情報＞（＜host-gen＞）パラメータを更新し、総ての未来のベクトルクロックに対して、完全に異なるホストとなるように生成し（任意のキーに対して）、完全に異なるホストとなって見えるようにする。従って、ホスト１３０が再起動されたときに＜ホスト情報＞パラメータをインクリメントすることは、障害の前に生成されたベクトルクロックを、再起動後に生成されたベクトルクロックから区別することを可能にする。理解されるであろうように、各ベクトルクロックに対するカウンタは、無限に単調増加する。実施の一形態において、無限のカウンタ番号を回避するために、各ホストは、例えば、＜ホスト情報＞パラメータをインクリメントすることにより、新しい固有の識別性を強制的に、定期的に選択する。例えば、再起動後に新しい固有の識別性を割り当てられたホストは、それにより、＜カウンタ＞（＜counter＞）パラメータをゼロに設定もする。これは、識別性を変更する前に単一のホスト１３０が調整（コーディネート）できる書込の回数によって、可能性のある最高のカウンタ値が制限される結果をもたらす。他の実施の形態において、識別性の変更は、そのカウンタ値の一つ又は複数が所定の閾値に到達したときに、ホストにおいて自動的にトリガされ得る。

【００６８】
＜キー情報＞（＜key-gen＞）パラメータは、キー生成カウンタをトラック（追跡）するために使用され得る。実施の一形態において、データハンドオフ後に、ホスト１３０は、得られたいかなるデータも削除する。これが、優先リスト１９０において下位にあるホスト１３０のための記憶容量を節約する。同時に、ホスト１３０は、データハンドオフ後にインクリメントされた＜キー情報＞パラメータを保持し、それにより、次回にホスト１３０が書込動作を実行することを要求されたと仮定した場合におけるいかなる因果関係も回避する。例えば、ホストＤが、ベクトルクロック［（Ａ，３），（Ｄ，１）］を有するデータセットのバージョンに対する書込動作を調整（コーディネート）し、データハンドオフを実行し、その後、ベクトルクロック［（Ａ，２）］を有するデータセットのバージョンに対する別の書込動作を調整することを要求される場合、更新されたデータセットがベクトルクロック［（Ａ，３），（Ｄ，２）］を有するようにすることは不適当である。この状況において新しい＜キー情報＞値を割り当てることにより、ホスト１３０は、新しいホストであるかのごとく見えるようになり、それにより、二つのバージョン間の因果関係の出現を回避する。実施の一形態において、各ホスト１３０は、キーごとに別個の＜キー情報＞を保持し、また、ベクトルクロックは変更された識別性（例えば、変更された＜ホストＩＤ＞又は更新されたその＜ホスト情報＞）を保持するので、各ホスト１３０は、対応してベクトルクロックが生成された総てのキーについてキーの世代を記憶する。同様に、各ホスト１３０は、対応する＜キー情報＞パラメータ又は＜ホスト情報＞パラメータのいずれかが更新されてから、キーに対するベクトルクロックにおいて使用された最後の＜カウンタ＞パラメータも記憶し得る。

【００６９】
＜タイムスタンプ＞パラメータは、データセットの世代及びそのベクトルクロック内のエントリを監視するために使用され得る。いくつかのアプリケーションにおいては、データが所定の世代を超えた場合、そのデータを削除することが望ましい。例えば、ショッピングカートアプリケーションにおいて、数日、数週間、数月又は数年等の期間に亘って放置されているショッピングカートは、削除することが望ましい場合がある。タイムスタンプは、この方法におけるデータセットの削除を支援（サポート）するために使用され得る。加えて、タイムスタンプは、ベクトルクロック切捨（Vector Clock Truncation）のためにも使用され得る。理解されるであろうように、データセットとの関係において書込動作が調整（コーディネート）された異なるホスト（又は異なる＜ホスト情報＞若しくは＜キー情報＞を有する同一ホスト）のリストの長さが増加するのに伴い、そのデータセットに対するベクトルクロックの長さも増加する（即ち、ベクトルクロックに含まれる｛ホストＩＤ、カウンタ｝（｛host ID, counter｝）ペアのリストの長さが増加するからである）。従って、所定数だけ世代を重ねたベクトルクロックは、タイムスタンプを使用して、削除され又は切り捨てられ得る。

【００７０】
他の実施の形態においては、データセットにおける変更をトラック（追跡）するために、ベクトルクロックを使用するよりむしろ、他のバージョン履歴機構が使用され得る。例えば、ハッシュ履歴も使用され得る。ここで、用語「バージョン履歴（version history）」とは、データセットにおける変更を経時的にトラック（追跡）するために（即ち、変更が存在することをトラックするために、であって、必ずしも変更の性質をトラックするために、ではない）使用され得る任意のデータ構造をいう。理解され得るように、異なるバージョン履歴機構は、ディスク領域使用量、バンド幅、古いバージョンが削除されたときの一貫性の保持、因果関係における先行の検出の速度及び容易さ、等による異なるトレードオフを提供し得る。実施の一形態において、データセットの二つ又はそれ以上のコピー間の因果関係における先行（又はその不存在、上記ではコンフリクト（矛盾）と称された）の検出を可能とするバージョン履歴機構が使用される。バージョン履歴機構は、バージョンコンフリクトを許容して、データの損失なしに（可用性）を発生させるために、及び、優先リストにおける上位のホストへのデータ移動の際に一貫性の保持を容易にするた

めに、使用され得る。

【００７１】

ＩＶ．複数データセンタ（Multiple Data Centers）

Ａ．複数データセンタ構成の構造（Architecture of Multiple Data Center Arrangement）

図１７乃至図２５を参照すると、データ処理システム１００の他の実施の形態が示されている。図１７乃至図２５において、データセットは、複数段（multi-tiered）リング構成に従ってホスト１３０間に分割されている。複数段リング構成は、例えば、各ホストが異なる地理的ロケーションに（例えば、異なる都市、異なる国、異なる大陸に存在し得る異なるデータセンタに）配置されているデータセット記憶システムを実施するために使用され得る。例えば、ホスト間における相互関係障害の確率を低減するために、データは、そのような異なるデータセンタ間で複製され得る。単一のデータセンタの障害が全システムの可用性に重大な影響を与える可能性は非常に小さい。加えて、クライアント要求をより近いデータセンタに転送することにより（ネットワーク待ち時間（network latency）によって）、端末間（end-to-end）データ検索応答時間は短縮され得る。複数段リング構成は、他の理由のために、例えば、共有データセンタ内に位置するホスト等においても使用され得る。例えば、リングの異なる段は、データセンタ内の領域、データセンタ内のホストの特定のラック（rack）等を特定するために使用され得る。一つの実施例を提供する目的のために、図１７においては、各ホストが異なるデータセンタに配置されているデータセット記憶システムを実施するために、複数段リング構成が使用されることが想定されている。

【００７２】

先ず図１７を参照すると、図１７は、データ処理システム１００が２段又は２レベルリング構成を備えている実施の形態を示している。２段リング構成は、図１及び／又は図２に示されたデータセットサービス１１２を実施するために使用され得る。図１７において、データ処理システム１００は、上位レベルリング５０４上に論理的に配置されている複数のデータセンタ５０２を備えている。四つのデータセンタ５０２が示されているが、実際には、任意の数のデータセンタ５０２が使用され得ることは理解されるであろう。

【００７３】

データセンタ５０２は、通信ネットワーク５０８によって相互に接続され得る（例えば、広域ネットワーク、インタネット等）。データセンタ５０２間のメッセージ通信は、図２５との関連において詳細に後述されるように、メッセージフィルタ５１０を通過し得る。図１乃至図２におけるように、データセンタ５０２のそれぞれは、通信ネットワーク１０４（例えば、インタネット）経由で種々のユーザコンピュータ１０２によってアクセスされ得る。

【００７４】

データセンタ５０２のそれぞれは、各下位レベルリング１８４上に論理的に配置されている複数のホスト１３０をさらに備えている。図示されている実施例においては、各下位レベルリング１８４は、異なるデータセンタ５０２に対応している。各データセンタ５０２内では、各リング１８４上のホスト１３０はまた、図３乃至図１６との関連において上述したように動作し得る。下位レベルリング１８４は、同種又は異種（例えば、異なる数のホスト、異なるハッシュ関数、異なる構成等を有する）であり得る。さらに、以下に説明するように、データセンタ５０２に関する上位レベルリング５０４の動作は、ホスト１３０に関して図３乃至図１６との関係において上述した通りのリング１８４の動作と同一であり得る。

【００７５】

図１８を参照すると、実施の一形態において、データセット記憶システム１１８は、各データセンタ５０２がデータセットの一部を記憶するように、各データセンタ５０２に分散され得る。データセンタ５０２のそれぞれは、上位レベルリング５０４上におけるハッシュ値の一つの範囲（又は、詳細に後述するように、上位レベルリング５０４上における

ハッシュ値の範囲の組（複数の範囲））に対して責任を有するものとすることができ、その場合、ホスト１３０及びリング１８４との関連において上述したのと同様の態様で、各データセンタ５０２は、ハッシュ範囲におけるそれ自体の位置から先行するデータセンタ５０２の位置までに延在するハッシュ値との関係において読出／書込動作に責任を有する。データセットへのアクセスの要求が受信されたとき（例えば、読出動作又は書込動作を通じて）、データセットがアクセスされ得るデータセンタ５０２を決定するために、上位レベルリング５０４のためのハッシュ関数にキーが適用される。（図１８において、符号ＤＣ１－ＤＣ４はそれぞれ、図１７における四つのデータセンタ５０２の異なる一つを示している。）　データセットがアクセスされ得るホスト１３０を、関係するデータセンタ５０２内で決定するために、キーは、下位レベルリング１８４のためのハッシュ関数にも適用される。上位レベルリング５０４のために使用されるハッシュ関数は、下位レベルリング１８４のために使用されるハッシュ関数と同一のもの又は異なるものであり得る。同様に、上述のように、下位レベルリング１８４のそれぞれのために使用されるハッシュ関数は、他の下位レベルリング１８４のために使用されるハッシュ関数と同一のもの又は異なるものであり得る。図１８に示された構成のマッピングでは、個々のデータセンタ５０２は、データセンタ５０２へのデータセットの分割についての全体的な再マッピングを行うことなく、追加し又は除去することが可能であり、それによりスケーラビリティ（拡大縮小可能性）（scalability）が増進される。

【００７６】

実施の一形態において、データセンタ間のデータ複製も、ホスト１３０との関連において図１０について上述したのと同様の態様により、支援（サポート）され得る。従って、図１８に示したように、リング５０４上の直近のデータセンタ５０２に単純にデータセットが割り当てられるよりむしろ、データセットは、最初のＭの後続するデータセンタ５０２に割り当てられるものとしてもよい。データセットサービス１１２は、データセットがＭのデータセンタにおいて複製されることを保証するために動作することが可能であり、各データセンタ５０２は、それ自体とそのＭ番目の先行するデータセンタ５０２との間のリング５０４上の範囲に対して責任を有し得る。

【００７７】

所定のデータセットの複製を記憶するデータセンタ５０２の数は、例えば、データセットごとの基準、データタイプごとの基準等で設定可能であるものとし得る。理解されるであろうように、保持される各データセットの複製の数は、特に、可用性の望ましいレベル及び通信ネットワーク５０８上の更新通信量の望ましいレベルに基づいて決定され得る。即ち、異なるデータセンタに亘ってより多くの複製が記憶されるほど、可用性は増加する。しかし、データセットの複製されたコピーを、一貫性を有するように保持するための更新の間に、通信ネットワーク５０８上のネットワーク通信量も増加する。データセットが一つのデータセンタ５０２内で複製されるべきものと仮定すると、データセットを複製するデータセンタ５０２内のホストの数も、例えば、データセンタごとの基準、データセットごとの基準、データタイプごとの基準等に基づいて、設定可能であるものとし得る。

【００７８】

実施の一形態において、データセンタ間の負荷平衡も、ホスト１３０との関連において図１２について上述したのと同様の態様により、支援（サポート）され得る。例えば、データセンタ５０２は、リング５０４上の複数の位置に割り当てられ得る。そのような構成は、当該構成を使用しなかったとするとリング５０４上における各データセンタ５０２の無作為の位置割り当てによって発生し得る不均一なデータ及び負荷の分散を回避するために使用され得る。そのような複数配置は、各データセンタ５０２に割り当てられるデータセットの数の不均一を低減するように作用する。その理由は、リング１８４上における無作為配置の増加した数は、各データセンタ５０２に割り当てられたデータセットの数を平均値に収束させるように作用するからである。加えて、リング５０４上の複数の位置へのデータセンタ５０２の割当ては、異種のデータセンタの使用を容易にもする、即ち、より性能の高いデータセンタ５０２（例えば、処理能力、記憶容量及び／又はネットワーク容

量に基づいて決定される）がリング５０４上のより多くの位置を割り当てられ、より性能の低いデータセンタ５０２がリング５０４上のより少ない位置を割り当てられるようにし得る。さらに加えて、リング５０４上の複数の位置へのデータセンタ５０２の割当ては、データセンタ間での負荷の移動も容易にする。その理由は、各データセンタ５０２が、他のデータセンタ５０２のそれぞれと後続／先行関係を有し得るからである（リング５０４上の各データセンタ５０２に十分な数の位置が割り当てられているものとする）。従って、例えば、データセンタ５０２のうちの一つが利用不可能又は非動作状態になったとすると、非動作状態になったデータセンタ５０２により処理されていた負荷は、データ可用性を損なうことなく、残余の利用可能な各データセンタ５０２にほぼ均等に分散させられ得る。

【００７９】

図１９を参照すると、各データセットは、各データセンタ５０２がキーに基づいて生成されたハッシュ値からリング５０４上を時計回りに周回するときに最初に遭遇する他のデータセンタ５０２の優先リスト５１９を有し得る。優先リスト５１９は、データセットへのアクセス（例えば、読出、書込等）に使用されるデータセンタ５０２の好適な順序を表している。総てのデータセンタ５０２が利用可能である場合、優先リスト５１９における上位Ｍのデータセンタ５０２がデータセットを記憶する。同一データセットにおける連続する動作がＭのデータセンタの同一セットにアクセスすることがあり、従って一貫性を有するものであり得る（即ち、同一キーにおける先行する動作によって読出／書込が行われた同一データに、動作がアクセスする）。優先リスト５１９における一つ又は複数のデータセンタに故障が発生した場合、又は、ネットワークパーティションが存在する場合、データセットは、優先リスト５１９における下位にランク付けされた一つ又は複数のデータセンタ５０２に一時的に記憶され、それにより高可用性が維持される。加えて、同一データセットへの連続する動作がデータセンタ５０２の異なるセットにアクセスすることがあるが、アクセスされるデータセンタ５０２のセットにいくらかの重複がある限り、動作は依然として一貫性を有し得る。優先リスト５１９において上位の利用可能なデータセンタ５０２にアクセスすることにより、動作から動作へのホストの利用可能性における小さい変化は、一貫性に否定的な影響を与えない。その理由は、後続のアクセスが、重複するデータセンタを含み得るからである。

【００８０】

優先リスト５１９は、例えば、ハッシュ関数に基づいて算定され得る。実施の一形態において、所定のデータセットを記憶しているデータセンタ５０２にアクセスするために、各ホスト１３０は、データセンタ５０２の利用可能性（優先リスト５１９における上位Ｍの利用可能なデータセンタを選択するために）に加えて、ハッシュ空間におけるデータセンタ位置に関する情報（優先リスト５１９を算定するために）も記憶し得る。他の実施の形態においては、例えば、記憶される優先リスト５１９が、ハッシュ関数に基づいて構成されること、及び、優先リスト５１９を構成する際に考慮に入れることが望ましい場合がある他の関数に基づいて構成されることを可能とするために、優先リスト５１９は、記憶され得る。

【００８１】

Ｂ．アクセス動作（Access Operations）

図２０乃至図２４を参照すると、データセンタ５０２に記憶されているデータセットへのアクセスに関連する動作が示されている。図２０は、実施の一形態に係る図１７のシステムにより実行されるアクセス動作のフローチャートである。図２１乃至図２４は、実施の一形態に係る図２０のアクセス動作の態様を詳細に示す図である。

【００８２】

ステップ６０２において、ユーザコンピュータ１０２との接続がデータセンタ５０２により確立される。理解されるであろうように、各データセンタ５０２は、データセットサービス１１２を実現するホスト１３０だけでなく、ネットワークインタフェース１１０及び他のサービス１１４を実現する他のホストも含み得る。従って、図２１を参照すると、

例えば、ネットワークインタフェース１１０を実現するホストの一つであり得るホスト５３２との接続が確立され得る。

【００８３】

実施の一形態において、ユーザコンピュータ１０２との接続は、一つのデータセンタ５０２（例えば、無作為基準による可能性もある）において確立される場合があり、その後、他のデータセンタ５０２に転送される。例えば、図２１において、ユーザコンピュータ１０２との接続は、一つのデータセンタＤＣ４内のホスト５３２によって確立されることがあり（ステップ６０２）、その後、例えば、より近いものであってもよく、より少ない負荷しか掛かっていないものであってもよく、及び／又は、接続を維持するためにより適したものとなるような他の特性を示すものであってもよい、他のデータセンタＤＣ１内の他のホスト５３４に転送される（ステップ６０４）。

【００８４】

ステップ６０６において、データアクセス要求（例えば、読出要求、書込要求等）が受信される。図２２を参照すると、データアクセス要求は、クライアントプロセス１３４を実行しているものであり得るホスト５３６から、データセットサービス１１２内のホスト１３０によって受信され得る（図３参照）。例えば、上述の図２において示した実施例との関係においては、ホスト５３４は、ネットワークインタフェース１１０を実現するホストの一つである場合があり、ユーザコンピュータ１０２に接続されている場合があり、ホスト５３６は、ショッピングカートサービス１２４を実現するホストの一つである場合があり、ホスト５３４から要求を受信する場合があり、ホスト１３０は、データセットサービス１１２を実現するホストの一つである場合があり、ホスト５３６からアクセス要求を受信する場合がある。データセットへのアクセス要求がデータセンタ５０２内のホスト１３０において受信されると、ステップ６０８において、ホスト１３０は、データセットがデータセンタ５０２内にローカルに記憶されているか否かを判定する。データセットは、データセンタ５０２内にローカルに記憶され得る。その理由は、例えば、データセンタ５０２が優先リスト５１９における上位Ｍのデータセンタの一つだからであり、データセンタ５０２が優先リスト５１９における下位のデータセンタであるが優先リスト５１９における上位Ｍのデータセンタの一つであるデータセンタ５０２にデータセットを移動するまで一時的にデータセットを記憶しているからであり、データセンタ５０２がユーザとの接続を確立してデータセットの貸し出された（leased）コピーを一時的に記憶しているからであり（詳細に後述するように）、又は、別の理由があるからである。データセットがローカルに記憶されている場合は、ステップ６１０において、応答は、データセットの一つ又は複数のローカルコピー（場合によっては一つ以上のバージョン）に基づいて供給され得る。そうでない場合は、ホスト１３０は、他のデータセンタ５０２からデータセットの一つ又は複数のコピー（場合によっては一つ以上のバージョン）を取得し得る。データセットのコンフリクト（矛盾）するバージョン（例えば、一つのデータセンタ内からのコンフリクトバージョン、異なるデータセンタからのコンフリクトバージョン、又は、その両方）が存在する場合、そのようないかなるコンフリクトバージョンも、データセットを要求する特定のクライアントプロセス１３４に関連するデータセットバージョン調整ロジック１３６に報告され、上述のように、データセットバージョン調整ロジック１３６により解決され得る。例を挙げる目的のために、データセンタＤＣ１が、データセットの複製を記憶するＭのデータセンタの一つではないものと仮定する。従って、データセンタＤＣ１におけるホスト１３０が、他のデータセンタからデータセットのコピーを得るためにコーディネータとして動作する。

【００８５】

ステップ６１２において、アクセス要求が受信された後、データセットに対するキーが、上位レベルリング５０４及び下位レベルリング１８４に対するハッシュ関数に適用される。ステップ６１４において、図２２を参照すると、データセンタＤＣ１におけるホスト１３０（コーディネータとして動作する）が、優先リスト５１９における一つ又は複数の上位のデータセンタからのデータを要求する。実施の一形態において、ホスト１３０は、

上位レベルリング５０４に対するハッシュ関数にキーを適用して、データセンタＤＣ２及びＤＣ３にアクセス要求を送信する（例えば、データセンタＤＣ２及びＤＣ３がそのデータセットに対する優先リスト５１９の上位にあることが判定された後に）。アクセス要求がデータセンタＤＣ２及びＤＣ３における各ホスト１３０により受信されると、それらのホスト１３０は、下位レベルリング１８４に対するハッシュ関数にキーを適用して、そのデータセットを記憶する各データセンタ内のホスト１３０を判別する。この手法において、データセンタＤＣ１におけるホスト１３０にとって、遠隔のデータセンタＤＣ２及びＤＣ３のリング１８４上におけるホスト１３０の位置に関する情報を記憶することは、必要ではない。他の実施の形態においては、各データセンタ５０２における各ホスト１３０はこの情報を記憶し、データセンタＤＣ１におけるホスト１３０は、上位レベルリング５０４及び下位レベルリング１８４の両方に対してキーを適用し得る。

【００８６】

実施の一形態において、ユーザコンピュータ１０２との接続が確立されたときに、データセットはプリフェッチ（pre-fetch：先取り）され得る。例えば、図２のショッピングカートの実施例との関係においては、ユーザコンピュータ１０２は、ホスト５３４との接続を確立し得るが、それは、ショッピングカートデータセットに対する要求が行われるいくらか前であり得る。例えば、ユーザは、ショッピングカートデータセットに対してアクセスすることを必要とするアクション（動作）が実行される前の時間は買い物を行い得る。従って、ショッピングカートデータセットに対してアクセスすることを必要とするアクションをユーザが実行するのを待たずに、ユーザコンピュータ１０２との接続が確立されると直ちに、データセットは、遠隔のデータセンタ５０２からの読出動作を実行することによりプリフェッチされ得る。この構成は、通信ネットワーク５０８経由でデータセット取得することに関連するネットワーク待ち時間（network latency）を回避するために用いられ得る。

【００８７】

ステップ６１６において、遠隔のデータセンタ５０２は、アクセス要求を処理し、データセンタＤＣ１におけるホスト１３０により受信される応答を送信する。実施の一形態においては、良好な読出動作のために、読出動作は、R_{DC}のデータセンタにおいて良好に行われなければならない。ここで、R_{DC}は、設定可能な値であって、$R_{DC} \leqq M$である。実施の一形態においては、良好と考えられる書込動作のために、書込動作は、W_{DC}のデータセンタにおいて良好に行われなければならない。ここで、W_{DC}は、設定可能な値であって、$W_{DC} \leqq M$である。$R_{DC} + W_{DC} > M$となるようなR_{DC}及びW_{DC}の設定は、読出及び書込動作に関係するデータセンタ５０２のセットの間における重複の設定可能な高い確率が存在するクォーラムのようなシステム（quorum-like system）をもたらす。

【００８８】

理解されるであろうように、データセンタ５０２からのデータセットにアクセスする場合、ホスト１３０に関して上述したように、データセットは、データセンタ５０２の同一のセットに書き込まれる必要はなく、また、データセンタ５０２の同一のセットから読み出される必要もない。例えば、データセットは、優先リスト５１９における下位のデータセンタ５０２に書き込まれて、優先リスト５１９における上位のデータセンタ５０２にデータハンドオフを通じて移動され、その後、最終的に優先リスト５１９における上位のデータセンタ５０２から読み出され得る。この態様において、優先リスト５１９の上位Ｍのデータセンタにおけるデータセットの最終的な一貫性は、実現され得る。データセンタ５０２はまた、それらが共有している範囲の下位レベルデータベース比較を周期的に実行し、その後、比較の間に検出されたいかなる差異も調整するために必要なデータ転送を実行する（例えば、データセットの喪失したコピーのために）。従って、データセットサービス１１２は、それらの優先リスト５１９における上位Ｍのデータセンタにデータセットの最新バージョンのコピーを動的に移動させる進行中の試行を行い得る。たとえデータセットの最新バージョンのコピーが、その優先リスト５１９の下位にあるデータセンタ５０２

に最初はコピーされることがあり、又は、別の理由により上位Ｍのデータセンタの一つにおいて喪失することがあるとしても、そのコピーは、優先リスト５１９における上位Ｍのデータセンタへ最終的に移動し戻され、上位Ｍのデータセンタにおけるデータセットの最終的な一貫性に帰着する。

【００８９】

ステップ６１８において、種々のホスト１３０及びデータセンタ５０２から受信される総てのデータセットについてのバージョン履歴は、異なるデータセンタから受信されるデータセット間の一貫性をチェックするために比較される。実施の一形態において、バージョン履歴はベクトルクロックであり、図１６との関連において上述したベクトルクロック構成は、異なるデータセンタ５０２に記憶されている同一データセットの異なるバージョン間の因果関係を捕捉するために使用される。例えば、いかなる所与の二つのホスト１３０も、それらが異なるセンタ内のものであったとしても、相互に区別され得るように、総てのホスト１３０は、普遍的に固有の＜ホストＩＤ＞（＜Host ID＞）を与えられ得る。そのような構成においては、データバージョニングを実行するために使用されるロジックは、ホスト１３０が複数段（multi-tiered）リング構成に従って組織的構造を与えられているという事実を必ずしも認識している（又は考慮に入れている）必要はない。他の実施の形態において、データバージョニングは、下位レベルリング１８４のレベルにおいて、及び、上位レベルリング５０４のレベルにおいて、分離して実行される。そのような実施の形態においては、ベクトルクロックは、書込動作をコーディネート（調整）したデータセンタを示す＜データセンタＩＤ＞（＜data center ID＞）パラメータを含み得る。

【００９０】

ステップ６２０において、図２４を参照すると、データセットは、遠隔のデータセンタにおいてリフレッシュされる。実施の一形態において、一旦データセットが取得されると、データセットは、ある時間期間（ここでは「貸出時間（lease time）」と称する）に亘ってデータセンタＤＣ１に保持される。その後、貸出（lease）が未だ終了していないという条件で、未来の読出動作がローカルに実行される。例えば、データセンタがデータセットへの読出を受信すると、そのデータセンタは、読出要求を送信することなく、ローカルホスト１３０からのデータセットを遠隔のデータセンタ５０２へ返送する。書込動作に関しては、実施の一形態において、書込動作は、上述したように、メッセージフィルタを使用して制御され得る。他の実施の形態において、書込動作は、読出動作について上述したのと同様の貸出構成を使用して実行され得る。即ち、データセンタがデータセットへの更新を受信すると、ホスト１３０は、その更新をローカルに実行し、貸出の終了時にのみ、その更新を他のデータセンタ５０２に非同期的に伝達する。データセットのローカルコピー上で動作を行うことにより、ユーザコンピュータ１０２において体験される待ち時間は、減少させられる。他の実施の形態においては、メッセージフィルタも貸出構成も使用されず、更新は、直ちに他のデータセンタ５０２に伝達される。

【００９１】

ネットワークサービス、例えば、小売ウェブサイトとの関係において、貸出時間は、ユーザコンピュータ１０２の平均セッション時間に基づいて決定されてもよく、ユーザコンピュータ１０２とのセッションが実行中（アクティブ）であるときは延長され得る。しかし、この貸出時間の間は、データセットが複数のデータセンタにおいて同時にアクセスされる場合（例えば、複数のユーザコンピュータ１０２が、異なるデータセンタにおける同一のデータセットにアクセスしている場合）、コンフリクトがデータセットの異なるコピーにおいて発生する可能性もある。そのような状況においては、ローカルコピーをリフレッシュし（図２４に示されるように）、そのローカルコピーを他のデータセンタに保持されている他の複製コピーと同期させることが望ましいことであり得る。異なるデータセンタ５０２に記憶されているデータセットにコンフリクトが発生するという点については、そのようなコンフリクトは、データセットバージョン調整ロジック１３６によって解決され得る。

【００９２】

Ｃ．メッセージフィルタ動作（Message Filter Operations）

図２５を参照すると、実施の一形態において、例えば、データセンタにデータを記憶するためのコストがネットワーク通信量（例えば、ピークネットワーク通信量、平均ネットワーク通信量等）のレベルによって影響を受ける場合、可用性の望ましいレベルになお適合させながら、データセンタ間の通信オーバヘッドを減少させることは望ましいことであり得る。例えば、他のデータセンタへのデータセットの即時の伝達を回避して、通信（トラフィック）バーストが平滑化されるようにすることは、望ましいことであり得る。

【００９３】

そのために、データセンタ５０２のそれぞれにおけるメッセージフィルタ５１０は、ネットワーク通信量を変調するために使用され得る。各メッセージフィルタ５１０は、対応するデータセンタ５０２内においてロジック的（論理的）に集中化され得る。メッセージフィルタ５１０は、データセンタ５０２内のホスト１３０からの書込要求を受信して、その書込要求を即時に又は遅延した態様で伝達するように構成され得る。図２５に示されるように、データセンタ５０２は、バースト５５２を伴うネットワーク通信量を発生する。メッセージフィルタ５１０は、バーストを平滑化して、ネットワーク通信量が時間期間に亘って分散させられた帯域幅波形５５４を生成するために動作する。他の実施例として、メッセージフィルタ５１０は、データセットのより古いバージョンを廃棄して（ベクトルクロックの分析に基づいて）、データセットの最新のバージョンのみを転送するために構成され得る。メッセージフィルタ５１０はまた、信頼性の高いメッセージ記憶システムとしても構成され得る。例えば、データセンタ５０２がダウンした場合又はメッセージを受信するために利用不可能になった場合、メッセージフィルタ５１０は、メッセージを記憶し、データセンタ５０２がオンラインに復帰したときにそのメッセージを送信するように構成され得る。

【００９４】

本発明は、図面を参照して、以上に説明されている。それらの図面は、本発明に係るシステム、方法及びプログラムを実施する特定の実施の形態についてのある一定の詳細を示している。しかし、図面による本発明の記載は、図面中に存在し得るいかなる限定も本発明に課するものと解釈されるべきではない。本発明は、方法、システム、及び、その動作を達成するための任意の機械読取可能な記録媒体上のプログラム製品を企図している。本発明の実施の形態は、既存のコンピュータプロセッサを使用して、又は、この若しくは他の目的のために組み込まれた専用コンピュータプロセッサにより、又は、結線接続されたシステムにより、実施され得る。

【００９５】

上述したように、本発明の範囲内にある実施の形態は、その上に記録された機械実行可能な命令若しくはデータ構造を運搬又は保持するための機械読取可能な記録媒体を含むプログラム製品を含む。そのような機械読取可能な記録媒体は、汎用若しくは専用のコンピュータ又はプロセッサを有する他の機械によりアクセス可能な任意の利用可能な記録媒体とすることができる。例として、そのような機械読取可能な記録媒体は、ＲＡＭ、ＲＯＭ、ＥＰＲＯＭ、ＥＥＰＲＯＭ、ＣＤ－ＲＯＭ若しくは他の光学ディスク記録媒体、磁気ディスク記録媒体若しくは他の磁気記録装置、又は、機械実行可能な命令若しくはデータ構造の形態でのプログラムコードを運搬若しくは記憶するために使用可能であり、汎用若しくは専用コンピュータ又はプロセッサを有する他の機械によりアクセス可能な任意の他の記録媒体を含み得る。情報が、ネットワーク若しくは他の通信接続（結線接続、無線のいずれか、又は、結線接続若しくは無線の組合せ）を通じて機械に転送され又は供給されると、その機械は、機械読取可能な記録媒体としての接続を適当に調査する。従って、そのような任意の接続は、機械読取可能な記録媒体と適当に称される。上記の組合せも、機械読取可能な記録媒体の範囲内に含まれる。機械実行可能な命令は、例えば、汎用コンピュータ、専用コンピュータ又は専用処理機械に、ある一定の機能若しくは機能のグループを実行させるための命令及びデータを含む。

【００９６】

本発明の実施の形態は、例えば、ネットワーク接続環境における機械によって実行されるプログラムモジュールの形態におけるプログラムコード等の、機械実行可能な命令を含むプログラム製品による実施の一形態において実施され得る方法ステップの一般的な前後関係において記載されている。概して、プログラムモジュールは、特定のタスクを実行し又は特定の抽象データタイプを実施するルーチン、プログラム、オブジェクト、コンポーネント、データ構造等を含む。機械実行可能な命令、関連するデータ構造、及び、プログラムモジュールは、ここに開示されている方法のステップを実行するためのプログラムコードの例を表している。そのような実行可能な命令又は関連するデータ構造の特定のシーケンスは、そのようなステップにおいて説明された機能を実施するための対応する動作の例を示している。

【００９７】

本発明の実施の形態は、プロセッサを有する一つ又は複数の遠隔コンピュータへの論理的（ロジック的）結合を使用するネットワーク接続環境において実施され得る。論理的結合は、ここでは例として示されるものであって限定ではないローカルエリアネットワーク（ＬＡＮ）及びワイドエリア（広域）ネットワーク（ＷＡＮ）を含み得る。そのようなネットワーク接続環境は、職場全域若しくは事業全域コンピュータネットワーク、イントラネット及びインタネットにおいて一般的であり、異なる通信プロトコルの幅広い変形を使用し得る。当該技術分野における通常の知識を有する者は、そのようなネットワーク計算方式環境が典型的には、パーソナルコンピュータ、ハンドヘルド（手持ち式）装置、マルチプロセッサシステム、マイクロプロセッサベースの又はプログラム可能なコンシューマ（消費者）電子機器、ネットワークＰＣ、サーバ、ミニ（小型）コンピュータ、メインフレームコンピュータ等を含む多くの種類のコンピュータシステム構成を包含することを理解するであろう。従って、図１に図示されたユーザコンピュータ１０２は、デスクトップコンピュータ、ラップトップコンピュータ、セットトップボックス、携帯（個人）情報端末（personal digital assistant：ＰＤＡ）、携帯電話、メディアプレーヤ、ウェブパッド、タブレット等を含み得るが、それらには限定されない。本発明の実施の形態は、通信ネットワークを通じて（結線接続リンク、無線リンクのいずれかにより、又は、結線接続リンク若しくは無線リンクの組合せにより）連結（リンク）されている局所（ローカル）及び遠隔（リモート）の処理装置によりタスクが実行される分散型計算方式環境においても実施され得る。分散型計算方式環境においては、プログラムモジュールは、局所（ローカル）及び遠隔（リモート）のメモリ記憶装置に配置され得る。

【００９８】

本発明の全体的なシステム又は部分を実施するための例示的なシステムは、演算処理装置、システムメモリ、及び、当該システムメモリを含む多様なシステム構成要素（コンポーネント）を当該演算処理装置に結合するシステムバスを含む、コンピュータの形態における汎用計算方式装置を含み得る。システムメモリは、読出専用メモリ（ＲＯＭ）及びランダムアクセスメモリ（ＲＡＭ）を含み得る。コンピュータは、磁気ハードディスクからの読出及び磁気ハードディスクへの書込を行うための磁気ハードディスク駆動装置、取り外し可能磁気ディスクからの読出及び取り外し可能磁気ディスクへの書込を行うための磁気ディスク駆動装置、ＣＤ－ＲＯＭ又は他の光学記録媒体等の取り外し可能光学ディスクからの読出及び取り外し可能光学ディスクへの書込を行うための光学ディスク駆動装置を含み得る。駆動装置及びそれらの関連する機械読出可能な記録媒体は、機械実行可能な命令、データ構造、プログラムモジュール、及び、コンピュータのための他のデータの不揮発性記憶装置を提供する。

【００９９】

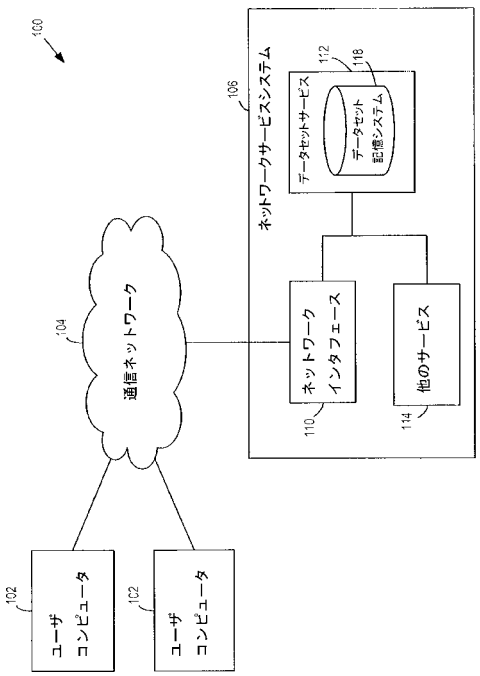
ここで提供されているフローチャートは方法ステップの特定の順序を示しているが、それらのステップの順序は図示されているものとは異なり得ることが理解される点に留意すべきである。また二つ又はそれ以上のステップは、同時に又は部分的に同時に実行され得る。そのような変更は、選択されたソフトウェア及びハードウェアシステム並びに設計者の選択に依存することになる。総てのそのような変更は、本発明の範囲内にあることが理

解される。同様に、本発明のソフトウェア及びウェブ手段は、種々のデータベース検索ステップ、相関ステップ、比較ステップ及び決定ステップを達成するための、規則に基づくロジック及び他のロジックによる標準プログラミング技術によって達成され得る。ここで及び特許請求の範囲で使用される単語「エンジン（engine）」は、ソフトウェアコードの一つ又は複数のラインを使用する手段、及び／又は、ハードウェア手段、及び／又は、手動入力を受信するための装置を包含することが意図されているということも留意されるべきである。エンジン、インタフェース、データベース、ブラウザ等の構成要素（コンポーネント）は、相互に通信し得る。そのような構成要素は、統合された態様で提供されるからであり、それらは、ネットワーク等の通信リンクを通じて相互に通信するからであり、及び／又は、その他の理由によるからである。

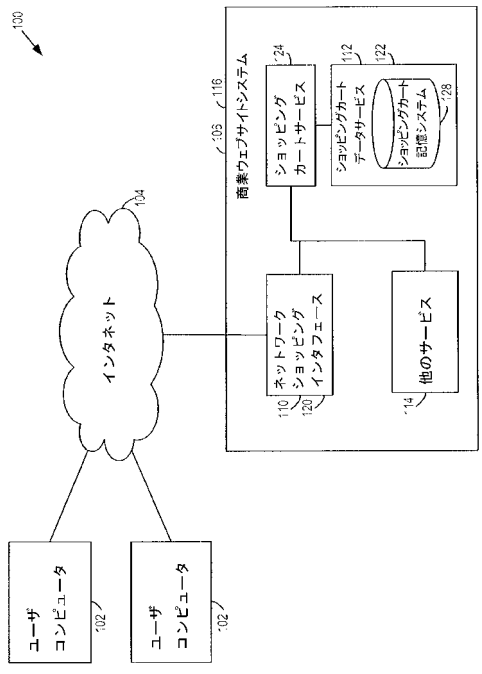
【０１００】

本発明の実施の形態についての以上の記載は、例示及び説明の目的のために提示されているものである。総てを網羅すること又は本発明を開示された通りの正確な形態に限定することは意図されておらず、変形及び変更は、上記教示に鑑みて可能であり、本発明の実施から獲得され得る。実施の形態は、当該技術分野における通常の知識を有する者が本発明を企図された特定用途に適当な種々の態様で及び種々の変形と共に利用することができるように、本発明の原理及びその実施上の応用を説明するために選択されて記載されたものである。

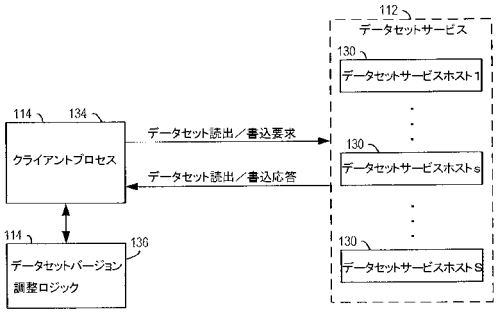
【図１】



【図２】

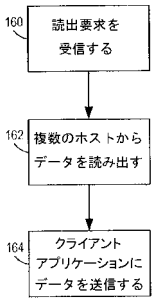


【図３】

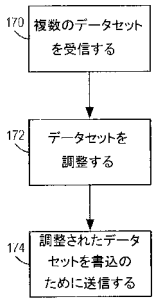


【図４】

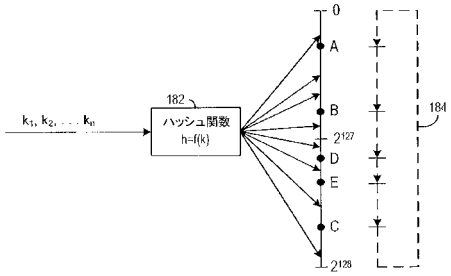
【図５】



【図６】



【図７】



【図８】

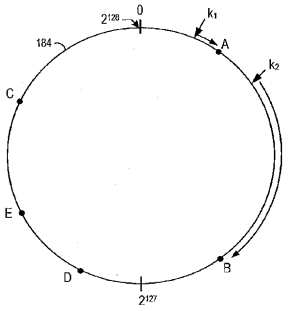


FIG. 8

【図９】

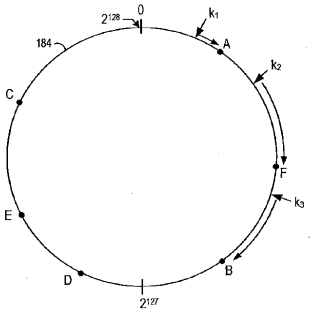


FIG. 9

【図１０】

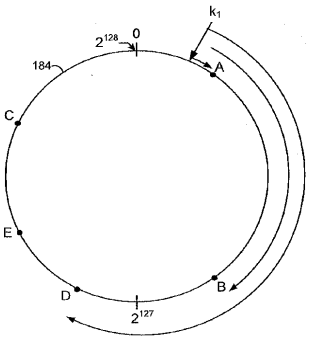
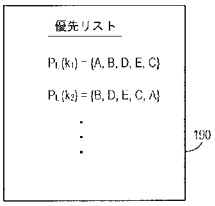


FIG. 10

【図１１】



【図１２】

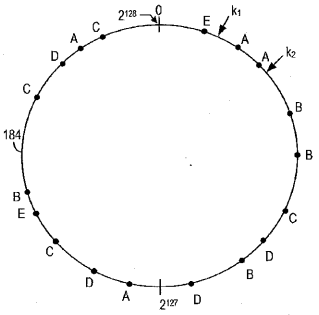
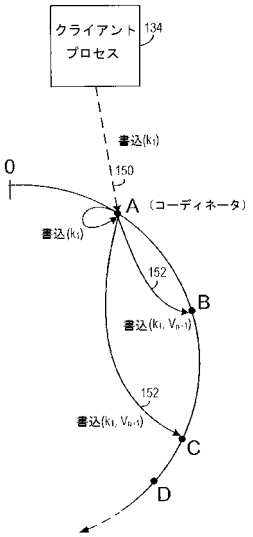
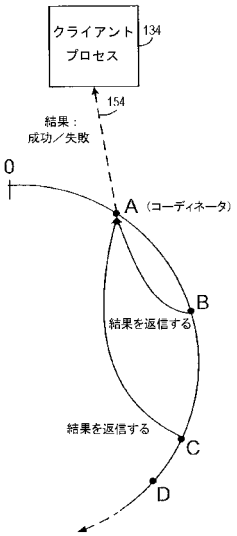


FIG. 12

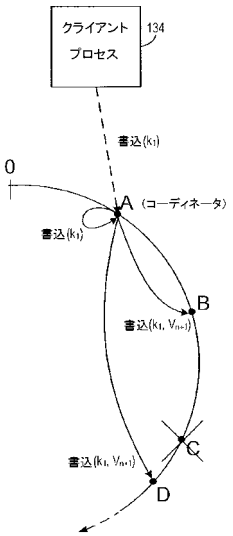
【図１３Ａ】



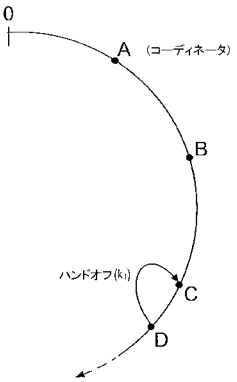
【図１３Ｂ】



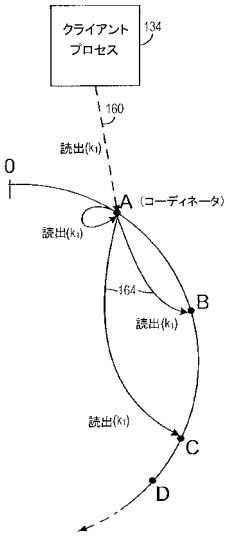
【図１４Ａ】



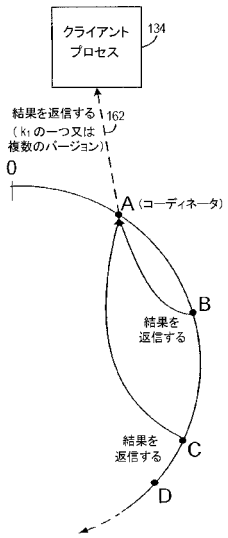
【図１４Ｂ】



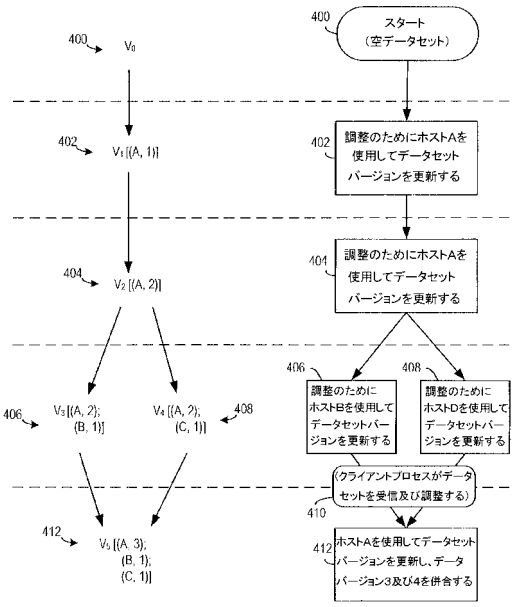
【図１５Ａ】



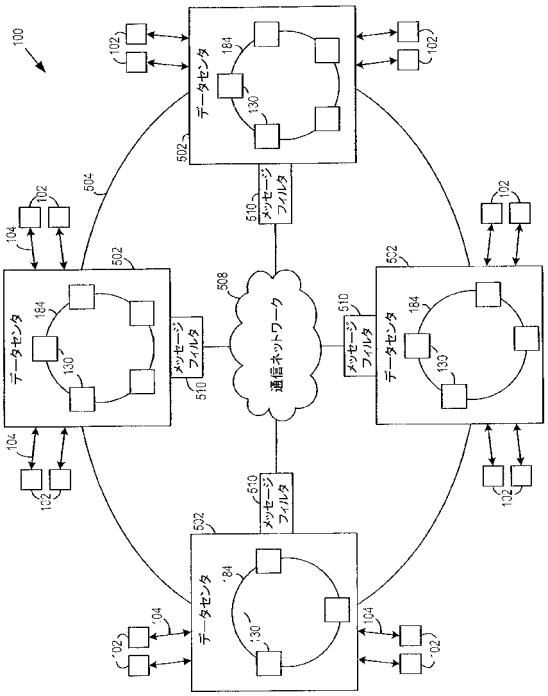
【図１５Ｂ】



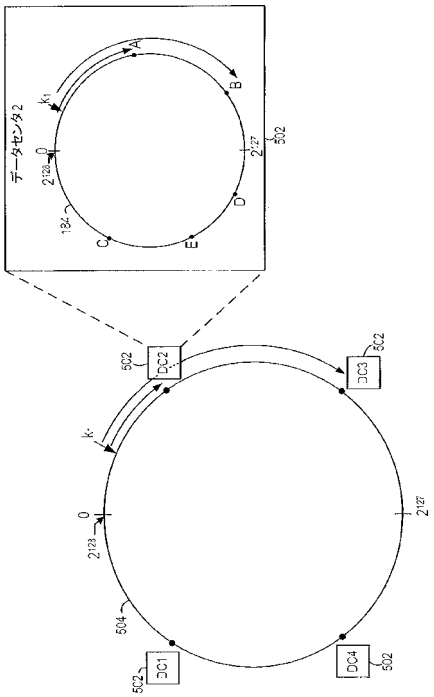
【図１６】



【図１７】



【図１８】



【図１９】

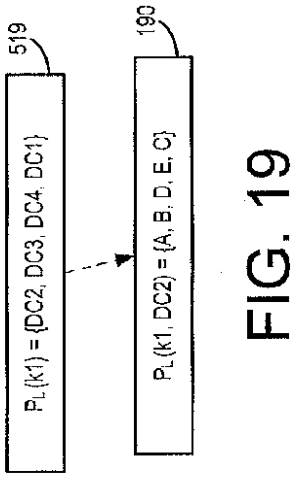
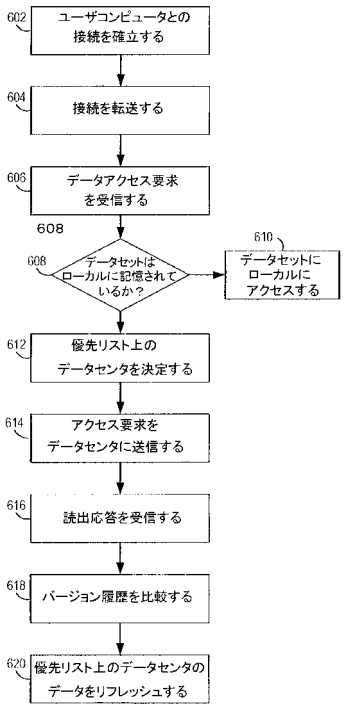
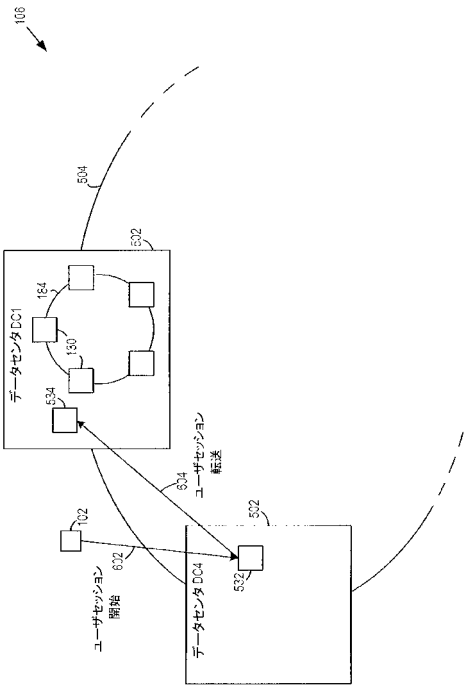


FIG. 19

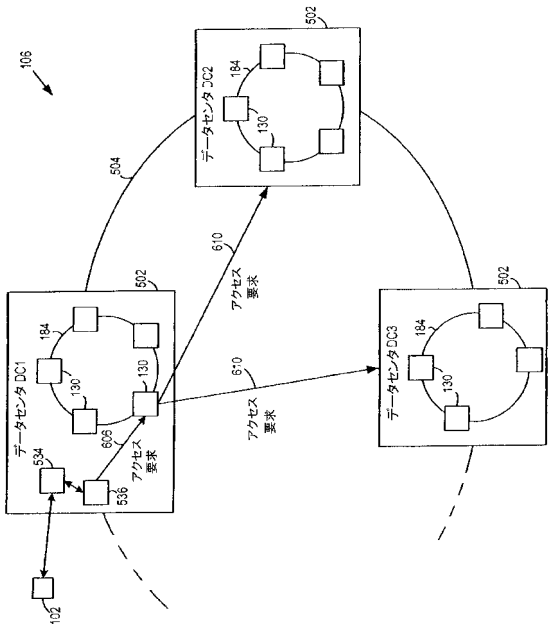
【図２０】



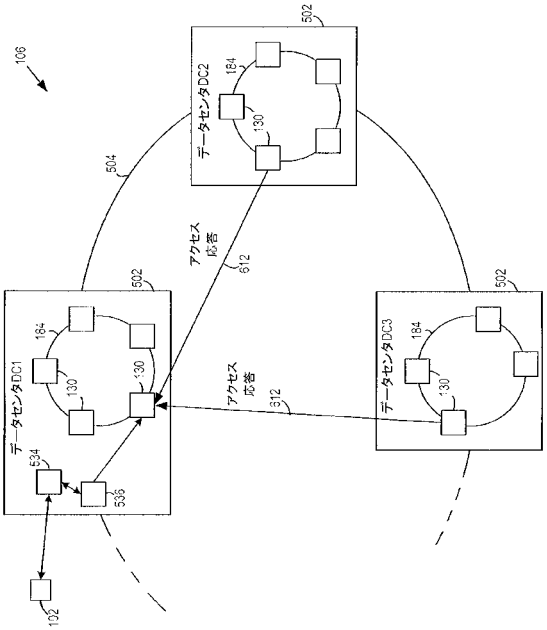
【図２１】



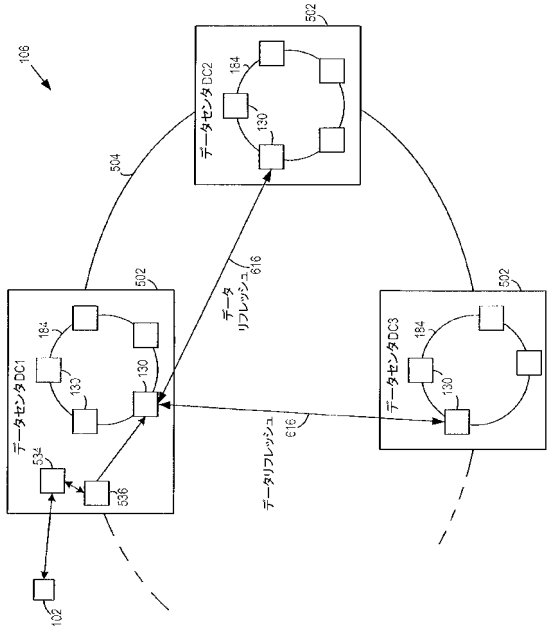
【図２２】



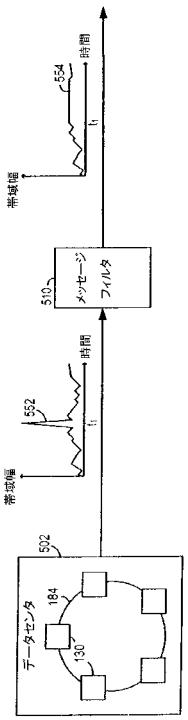
【図２３】



【図２４】



【図２５】



フロントページの続き

(74)代理人 100135633
弁理士 二宮 浩康
(74)代理人 100114890
弁理士 アインゼル・フェリックス＝ラインハルト
(72)発明者 ピーター ヴォスホール
アメリカ合衆国 ワシントン シアトル トゥウェルヴス アヴェニュー サウス １２００ － スイート １２００
(72)発明者 スワミナサン シヴァスブラマニアン
アメリカ合衆国 ワシントン シアトル トゥウェルヴス アヴェニュー サウス １２００ － スイート １２００
(72)発明者 ジョゼッペ デカンディア
アメリカ合衆国 ワシントン シアトル トゥウェルヴス アヴェニュー サウス １２００ － スイート １２００
(72)発明者 デニス ハストルン
アメリカ合衆国 ワシントン シアトル トゥウェルヴス アヴェニュー サウス １２００ － スイート １２００
(72)発明者 アヴィナシュ ラクシュマン
アメリカ合衆国 ワシントン シアトル トゥウェルヴス アヴェニュー サウス １２００ － スイート １２００
(72)発明者 アレックス ピルチン
アメリカ合衆国 ワシントン シアトル トゥウェルヴス アヴェニュー サウス １２００ － スイート １２００
(72)発明者 イヴァン ディー ロセロ
アメリカ合衆国 ワシントン シアトル トゥウェルヴス アヴェニュー サウス １２００ － スイート １２００

審査官 原 秀人

(56)参考文献 特開２００６－１９５７４４（ＪＰ，Ａ）
特開２００４－１５７７９３（ＪＰ，Ａ）
米国特許出願公開第２００５／０１２００２５（ＵＳ，Ａ１）
国際公開第２００６／０５３５８０（WO，Ａ１）
国際公開第２００５／０４３３２３（WO，Ａ１）
米国特許第０５９２４０９６（ＵＳ，Ａ）
米国特許第０６５９７７００（ＵＳ，Ｂ１）
DABEK F ET AL., Wide-area cooperative storage with CFS, ２００１年１０月２１日, ＵＲＬ, http://ftp.lcdf.org/papers/cfs:sosp01/cfs_sosp.pdf

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １２／００
Ｇ０６Ｆ １３／００